

Network Working Group
Request for Comments: 3315
Category: Standards Track

R. Droms, Ed.
Cisco
J. Bound
Hewlett Packard
B. Volz
Ericsson
T. Lemon
Nominum
C. Perkins
Nokia Research Center
M. Carney
Sun Microsystems
July 2003

Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

The Dynamic Host Configuration Protocol for IPv6 (DHCP) enables DHCP servers to pass configuration parameters such as IPv6 network addresses to IPv6 nodes. It offers the capability of automatic allocation of reusable network addresses and additional configuration flexibility. This protocol is a stateful counterpart to "IPv6 Stateless Address Autoconfiguration" ([RFC 2462](#)), and can be used separately or concurrently with the latter to obtain configuration parameters.

Table of Contents

1.	Introduction and Overview	5
1.1.	Protocols and Addressing	6
1.2.	Client-server Exchanges Involving Two Messages	6
1.3.	Client-server Exchanges Involving Four Messages	7
2.	Requirements	7
3.	Background	8
4.	Terminology	8
4.1.	IPv6 Terminology	9
4.2.	DHCP Terminology	10
5.	DHCP Constants	12
5.1.	Multicast Addresses	13
5.2.	UDP Ports	13
5.3.	DHCP Message Types	13
5.4.	Status Codes	15
5.5.	Transmission and Retransmission Parameters	16
5.6.	Representation of time values and "Infinity" as a time value	16
6.	Client/Server Message Formats	16
7.	Relay Agent/Server Message Formats	17
7.1.	Relay-forward Message	18
7.2.	Relay-reply Message	19
8.	Representation and Use of Domain Names	19
9.	DHCP Unique Identifier (DUID)	19
9.1.	DUID Contents	20
9.2.	DUID Based on Link-layer Address Plus Time [DUID-LLT] .	20
9.3.	DUID Assigned by Vendor Based on Enterprise Number [DUID-EN]	22
9.4.	DUID Based on Link-layer Address [DUID-LL]	22
10.	Identity Association	23
11.	Selecting Addresses for Assignment to an IA	24
12.	Management of Temporary Addresses	25
13.	Transmission of Messages by a Client	25
14.	Reliability of Client Initiated Message Exchanges	26
15.	Message Validation	27
15.1.	Use of Transaction IDs	28
15.2.	Solicit Message	28
15.3.	Advertise Message	28
15.4.	Request Message	29
15.5.	Confirm Message	29
15.6.	Renew Message	29
15.7.	Rebind Message	29
15.8.	Decline Messages	30
15.9.	Release Message	30
15.10.	Reply Message	30
15.11.	Reconfigure Message	31
15.12.	Information-request Message	31

15.13.	Relay-forward Message	31
15.14.	Relay-reply Message	31
16.	Client Source Address and Interface Selection	32
17.	DHCP Server Solicitation	32
17.1.	Client Behavior	32
17.1.1.	Creation of Solicit Messages	32
17.1.2.	Transmission of Solicit Messages	33
17.1.3.	Receipt of Advertise Messages	35
17.1.4.	Receipt of Reply Message	35
17.2.	Server Behavior	36
17.2.1.	Receipt of Solicit Messages	36
17.2.2.	Creation and Transmission of Advertise Messages	36
17.2.3.	Creation and Transmission of Reply Messages .	38
18.	DHCP Client-Initiated Configuration Exchange	38
18.1.	Client Behavior	39
18.1.1.	Creation and Transmission of Request Messages .	39
18.1.2.	Creation and Transmission of Confirm Messages .	40
18.1.3.	Creation and Transmission of Renew Messages .	41
18.1.4.	Creation and Transmission of Rebind Messages .	43
18.1.5.	Creation and Transmission of Information-request Messages	44
18.1.6.	Creation and Transmission of Release Messages .	44
18.1.7.	Creation and Transmission of Decline Messages .	46
18.1.8.	Receipt of Reply Messages	46
18.2.	Server Behavior	48
18.2.1.	Receipt of Request Messages	49
18.2.2.	Receipt of Confirm Messages	50
18.2.3.	Receipt of Renew Messages	51
18.2.4.	Receipt of Rebind Messages	51
18.2.5.	Receipt of Information-request Messages	52
18.2.6.	Receipt of Release Messages	53
18.2.7.	Receipt of Decline Messages	53
18.2.8.	Transmission of Reply Messages	54
19.	DHCP Server-Initiated Configuration Exchange	54
19.1.	Server Behavior	55
19.1.1.	Creation and Transmission of Reconfigure Messages	55
19.1.2.	Time Out and Retransmission of Reconfigure Messages	56
19.2.	Receipt of Renew Messages	56
19.3.	Receipt of Information-request Messages	56
19.4.	Client Behavior	57
19.4.1.	Receipt of Reconfigure Messages	57
19.4.2.	Creation and Transmission of Renew Messages .	58
19.4.3.	Creation and Transmission of Information-request Messages	58
19.4.4.	Time Out and Retransmission of Renew or Information-request Messages	58

19.4.5. Receipt of Reply Messages	58
20. Relay Agent Behavior	58
20.1. Relaying a Client Message or a Relay-forward Message	59
20.1.1. Relaying a Message from a Client	59
20.1.2. Relaying a Message from a Relay Agent	59
20.2. Relaying a Relay-reply Message	60
20.3. Construction of Relay-reply Messages	60
21. Authentication of DHCP Messages	61
21.1. Security of Messages Sent Between Servers and Relay Agents	61
21.2. Summary of DHCP Authentication	63
21.3. Replay Detection	63
21.4. Delayed Authentication Protocol	63
21.4.1. Use of the Authentication Option in the Delayed Authentication Protocol	64
21.4.2. Message Validation	65
21.4.3. Key Utilization	65
21.4.4. Client Considerations for Delayed Authentication Protocol	66
21.4.5. Server Considerations for Delayed Authentication Protocol	67
21.5. Reconfigure Key Authentication Protocol	68
21.5.1. Use of the Authentication Option in the Reconfigure Key Authentication Protocol	69
21.5.2. Server considerations for Reconfigure Key protocol	69
21.5.3. Client considerations for Reconfigure Key protocol	70
22. DHCP Options	70
22.1. Format of DHCP Options	71
22.2. Client Identifier Option	71
22.3. Server Identifier Option	72
22.4. Identity Association for Non-temporary Addresses Option	72
22.5. Identity Association for Temporary Addresses Option .	75
22.6. IA Address Option	76
22.7. Option Request Option	78
22.8. Preference Option	79
22.9. Elapsed Time Option	79
22.10. Relay Message Option	80
22.11. Authentication Option	81
22.12. Server Unicast Option	82
22.13. Status Code Option	82
22.14. Rapid Commit Option	83
22.15. User Class Option	84
22.16. Vendor Class Option	85
22.17. Vendor-specific Information Option	86
22.18. Interface-Id Option	87
22.19. Reconfigure Message Option	88

22.20. Reconfigure Accept Option.	89
23. Security Considerations	89
24. IANA Considerations	91
24.1. Multicast Addresses.	92
24.2. DHCP Message Types	93
24.3. DHCP Options	94
24.4. Status Codes	95
24.5. DUID	95
25. Acknowledgments	95
26. References.	96
26.1. Normative References	96
26.2. Informative References	97
A. Appearance of Options in Message Types	98
B. Appearance of Options in the Options Field of DHCP Options	99
Chair's Address	99
Authors' Addresses.	100
Full Copyright Statement.	101

1. Introduction and Overview

This document describes DHCP for IPv6 (DHCP), a client/server protocol that provides managed configuration of devices.

DHCP can provide a device with addresses assigned by a DHCP server and other configuration information, which are carried in options. DHCP can be extended through the definition of new options to carry configuration information not specified in this document.

DHCP is the "stateful address autoconfiguration protocol" and the "stateful autoconfiguration protocol" referred to in "IPv6 Stateless Address Autoconfiguration" [17].

The operational models and relevant configuration information for DHCPv4 [18][19] and DHCPv6 are sufficiently different that integration between the two services is not included in this document. If there is sufficient interest and demand, integration can be specified in a document that extends DHCPv6 to carry IPv4 addresses and configuration information.

The remainder of this introduction summarizes DHCP, explaining the message exchange mechanisms and example message flows. The message flows in sections 1.2 and 1.3 are intended as illustrations of DHCP operation rather than an exhaustive list of all possible client-server interactions. Sections 17, 18, and 19 explain client and server operation in detail.

1.1. Protocols and Addressing

Clients and servers exchange DHCP messages using UDP [15]. The client uses a link-local address or addresses determined through other mechanisms for transmitting and receiving DHCP messages.

DHCP servers receive messages from clients using a reserved, link-scoped multicast address. A DHCP client transmits most messages to this reserved multicast 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.

1.2. Client-server Exchanges Involving Two Messages

When a DHCP client does not need to have a DHCP server assign it IP addresses, the client can obtain configuration information such as a list of available DNS servers [20] or NTP servers [21] through a single message and reply exchanged with a DHCP server. To obtain 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 configuration information for the client.

This message exchange assumes that the client requires only configuration information and does not require the assignment of any IPv6 addresses.

When a server has IPv6 addresses 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 requesting the assignment of addresses and other configuration information. This message includes an indication 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 to the client

immediately responds with a Reply message. The configuration information and the addresses in the Reply message are then immediately available for use by the client.

Each address 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, 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 without interruption.

1.3. Client-server Exchanges Involving Four Messages

To request the assignment of one or more IPv6 addresses, a client first locates a DHCP server and then requests the assignment of addresses and other configuration information from the server. The client sends a Solicit message to the All_DHCP_Relay_Agents_and_Servers 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 other configuration information. The server responds with a Reply message that contains the confirmed addresses and configuration.

As described in the previous section, the client sends a Renew message to the server to extend the lifetimes associated with its addresses, allowing the client to continue to use those addresses without interruption.

2. Requirements

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [1].

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 implementor to study includes the IPv6 Specification [3], the IPv6 Addressing Architecture [5], IPv6 Stateless Address Autoconfiguration [17], IPv6 Neighbor Discovery Processing [13], and Dynamic Updates to DNS [22]. These specifications enable DHCP to build upon the IPv6 work to provide both robust stateful autoconfiguration and autoregistration of DNS Host Names.

The IPv6 Addressing Architecture specification [5] 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 [17] specifies procedures by which a node may autoconfigure addresses based on router advertisements [13], and the use of a valid lifetime to support renumbering of addresses on the Internet. In addition, the protocol interaction by which a node begins stateless or stateful autoconfiguration is specified. DHCP is one vehicle to perform stateful autoconfiguration. Compatibility with stateless address autoconfiguration is a design requirement of DHCP.

IPv6 Neighbor Discovery [13] is the node discovery protocol in IPv6 which replaces and enhances functions of ARP [14]. To understand IPv6 and stateless address autoconfiguration, it is strongly recommended that implementors understand IPv6 Neighbor Discovery.

Dynamic Updates to DNS [22] is a specification that supports the dynamic update of DNS records for both IPv4 and IPv6. DHCP can use the dynamic updates to DNS to integrate addresses and name space to not only support autoconfiguration, but also autoregistration in IPv6.

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 [3], IPv6 Addressing Architecture [5], and IPv6 Stateless Address Autoconfiguration [17] 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); Token Ring; PPP links, X.25, Frame Relay, or ATM networks; 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 in an IA or configuration information explicitly assigned to the client. Configuration information that has been returned to a client through a policy - for example, 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 address 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.
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.
DHCP domain	A set of links managed by DHCP and operated by a single administrative entity.
DHCP realm	A name used to identify the DHCP administrative domain from which a DHCP authentication key was selected.
DHCP relay agent (or relay agent)	A node that acts as an intermediary to deliver DHCP messages between clients and servers, and is on the same link as the client.
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).
DUID	A DHCP Unique IDentifier for a DHCP participant; each DHCP client and server has exactly one DUID. See section 9 for details of the ways in which a DUID may be constructed.
Identity association (IA)	A collection of addresses 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 address; for example, an identity association for temporary addresses (IA_TA) holds temporary addresses (see "identity association for temporary addresses"). Throughout this document, "IA" is used to refer to an identity association without identifying the type of addresses in the IA.

Identity association identifier (IAID) An identifier for an IA, chosen by the client. Each IA has an IAIID, which is chosen to be unique among all IAIIDs for IAs belonging to that client.

Identity association for non-temporary addresses (IA_NA) An IA that carries assigned addresses that are not temporary addresses (see "identity association for temporary addresses")

Identity association for temporary addresses (IA_TA) An IA that carries temporary addresses (see [RFC 3041 \[12\]](#)).

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.

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

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

5. DHCP Constants

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

5.1. 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.

5.2. UDP Ports

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

5.3. DHCP Message Types

DHCP defines the following message types. More detail on these message types can be found in sections 6 and 7. Message types not listed here are reserved for future use. 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 IP addresses, 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 addresses and configuration parameters to extend the lifetimes on the addresses 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 addresses 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 addresses 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 addresses to the client to indicate that the client will no longer use one or more of the assigned addresses.
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 IP addresses 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.

5.4. Status Codes

DHCPv6 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 24.4](#).

5.5. 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
<hr/>		
SOL_MAX_DELAY	1 sec	Max delay of first Solicit
SOL_TIMEOUT	1 sec	Initial Solicit timeout
SOL_MAX_RT	120 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	120 secs	Max Information-request timeout value
REL_TIMEOUT	1 sec	Initial Release timeout
REL_MAX_RC	5	MAX Release attempts
DEC_TIMEOUT	1 sec	Initial Decline timeout
DEC_MAX_RC	5	Max Decline 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

5.6 Representation of time values and "Infinity" as a time value

All time values for lifetimes, T1 and T2 are unsigned integers. The value 0xffffffff is taken to mean "infinity" when used as a lifetime (as in [RFC2461 \[17\]](#)) or a value for T1 or T2.

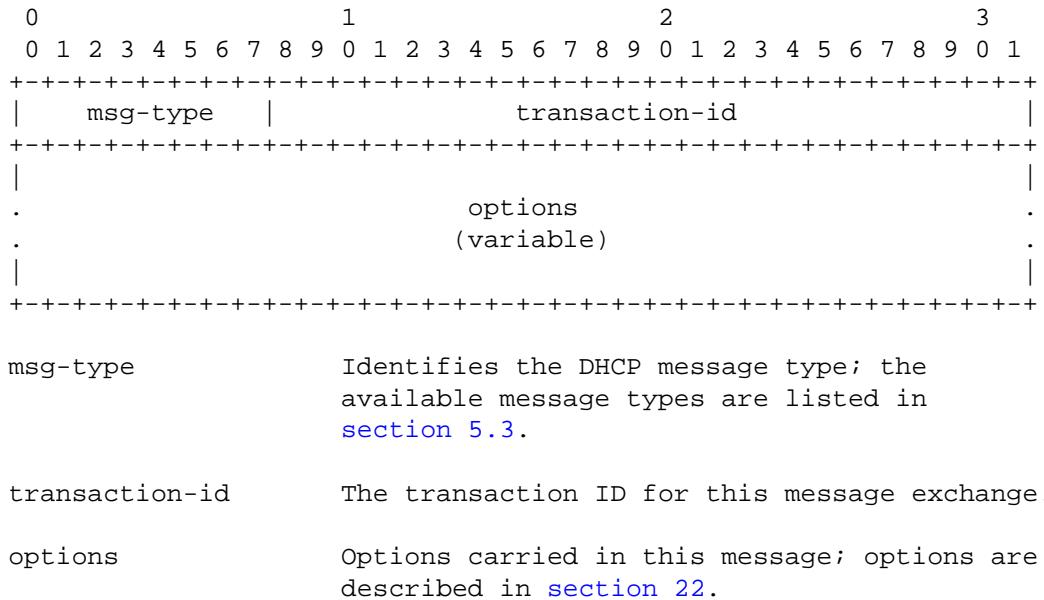
6. Client/Server Message Formats

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:



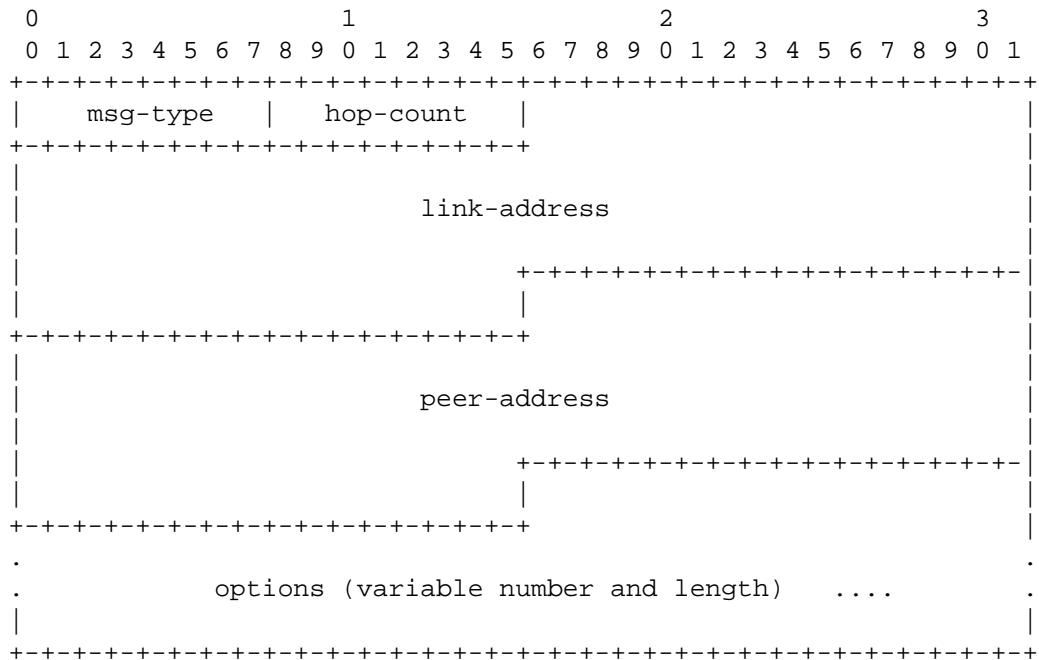
7. Relay Agent/Server Message Formats

Relay agents exchange messages with 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:



The following sections describe the use of the Relay Agent message header.

7.1. Relay-forward Message

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

msg-type	RELAY-FORW
hop-count	Number of relay agents that have relayed this message.
link-address	A global or site-local address that will be used by the server to identify the link on which the client is located.
peer-address	The address of the client or relay agent from which the message to be relayed was received.
options	MUST include a "Relay Message option" (see section 22.10); MAY include other options added by the relay agent.

7.2. Relay-reply Message

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

msg-type	RELAY-REPL
hop-count	Copied from the Relay-forward message
link-address	Copied from the Relay-forward message
peer-address	Copied from the Relay-forward message
options	MUST include a "Relay Message option"; see section 22.10 ; MAY include other options

8. 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 RFC 1035 \[10\]](#). 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 RFC 1035](#).

9. DHCP Unique Identifier (DUID)

Each DHCP client and server has a DUID. DHCP servers use DUIDs to identify clients for the selection of configuration parameters and in the association of IAs with clients. DHCP clients use DUIDs to identify a server in messages where a server needs to be identified. See sections [22.2](#) and [22.3](#) 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 MUST 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.

The DUID is carried in an option because it may be variable 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 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.

9.1. DUID Contents

A DUID consists of a two-octet type code represented in network byte order, followed by a variable number of octets that make up the actual identifier. A DUID can be no more than 128 octets long (not including the type code). The following types are currently defined:

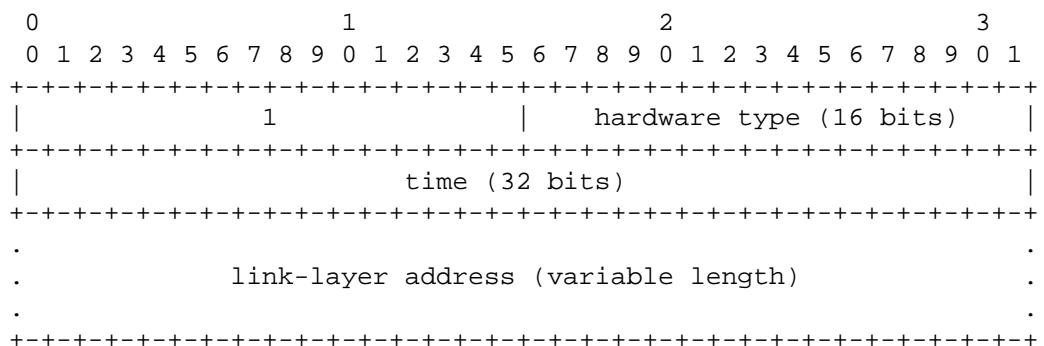
- 1 Link-layer address plus time
 - 2 Vendor-assigned unique ID based on Enterprise Number
 - 3 Link-layer address

Formats for the variable field of the DUID for each of the above types are shown below.

9.2. DUID Based on Link-layer Address Plus Time [DUID-LLT]

This type of DUID consists of a two octet type field containing the value 1, a two octet 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 [RFC 826](#) [14]. 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 [RFC 2464](#) [2].

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



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.

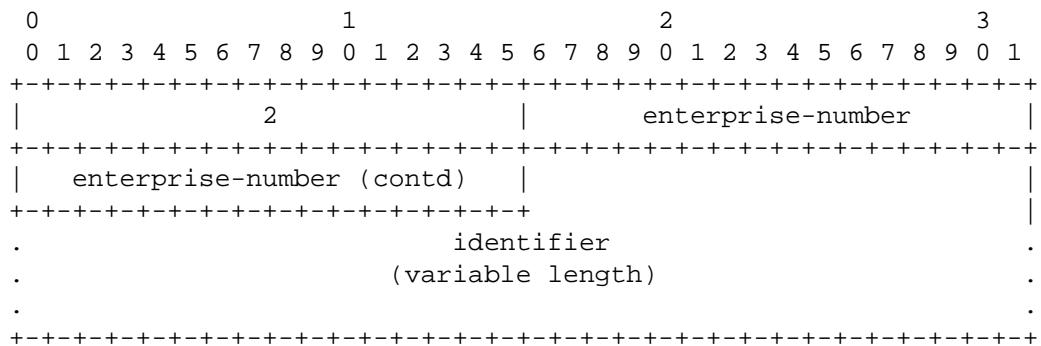
Clients and servers that use this DUID SHOULD attempt to configure the time prior to generating the DUID, if that is possible, and MUST use some sort of time source (for example, a real-time clock) in generating the DUID, even if that time source could not be configured prior to generating the DUID. 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.

Despite our best efforts, 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.

9.3. 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 vendor's registered Private Enterprise Number as maintained by IANA [6] followed by a unique identifier assigned by the vendor. The following diagram summarizes the structure of a DUID-EN:



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 at the time it is manufactured and stored in some form of non-volatile storage. 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 [6]. 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|221| 3 | 0 | 9 | 18|
+-----+-----+-----+

```

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

9.4. DUID Based on Link-layer Address [DUID-LL]

This type of DUID consists of two octets containing the DUID type 3, a two octet network hardware type code, followed by the link-layer address of any one network interface that is permanently connected to the client or server device. For example, a host 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 [RFC 826 \[14\]](#). The hardware type is stored in network byte order. The link-layer address is stored in canonical form, as described in [RFC 2464 \[2\]](#). 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
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           3           |       hardware type (16 bits)   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
.
.
.
link-layer address (variable length)
.
.
.
+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

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 MUST 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.

10. 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. Each IA consists of an IAID and associated configuration information.

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 IAID uniquely identifies the IA and must be chosen to be unique among the IAIDs on the client. 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.

The configuration information in an IA consists of one or more IPv6 addresses along with the times T1 and T2 for the IA. See [section 22.4](#) for the representation of an IA in a DHCP message.

Each address in an IA has a preferred lifetime and a valid lifetime, as defined in [RFC 2462 \[17\]](#). The lifetimes are transmitted from the DHCP server to the client in the IA option. The lifetimes apply to the use of IPv6 addresses, as described in [section 5.5.4 of RFC 2462](#).

11. Selecting Addresses for Assignment to an IA

A server selects addresses to be assigned to an IA 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.
 - * 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.

- Other information in options supplied by the relay agent.

Any address assigned by a server that is based on an EUI-64 identifier MUST include an interface identifier with the "u" (universal/local) and "g" (individual/group) bits of the interface identifier set appropriately, as indicated in section 2.5.1 of [RFC 2373](#) [5].

A server MUST NOT assign an address that is otherwise reserved for some other purpose. For example, a server MUST NOT assign reserved anycast addresses, as defined in [RFC 2526](#), from any subnet.

12. Management of Temporary Addresses

A client may request the assignment of temporary addresses (see [RFC 3041](#) [12] for the definition of temporary addresses). DHCPv6 handling of address assignment is no different for temporary addresses. DHCPv6 says nothing about details of temporary addresses like lifetimes, how clients use temporary addresses, rules for generating successive temporary addresses, etc.

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 22.5](#)). Each IA_TA option contains at most one temporary address for each of the prefixes on the link to which the client is attached.

The IAID number space for the IA_TA option IAID number space is separate from the IA_NA option IAID number space.

The server MAY update the DNS for a temporary address, as described in [section 4 of RFC 3041](#).

13. 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.

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 22.3](#)) 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 22.12](#).

14. Reliability of Client Initiated Message Exchanges

DHCP clients are responsible for reliable delivery of messages in the client-initiated message exchanges described in sections [17](#) and [18](#). If a DHCP client fails to receive an expected response from a server, the client must retransmit its message. This section describes the retransmission strategy to be used by clients in client-initiated message exchanges.

Note that the procedure described in this section is slightly modified when used with the `Solicit` message. The modified procedure is described in [section 17.1.2](#).

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 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.

15. Message Validation

Clients and servers SHOULD discard any messages that contain options that are not allowed to appear in the received message. For example, an IA option is not allowed to appear in an Information-request message. Clients and servers MAY choose to extract information from such a message if the information is of use to the recipient.

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

Message validation based on DHCP authentication is discussed in [section 21.4.2](#).

If a server receives a message that contains options it should not contain (such as an Information-request message with an IA option), is missing options that it should contain, or is otherwise not valid, 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.

[15.1.](#) 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.

[15.2.](#) 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.

[15.3.](#) Advertise Message

Clients MUST discard any received Advertise messages that meet 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.

15.4. Request Message

Clients MUST discard any received Request messages.

Servers MUST discard any received Request message that meet 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.

15.5. 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.

15.6. 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.

15.7. 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.

15.8. 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.

15.9. 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.

15.10. 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.

15.11. Reconfigure Message

Servers and relay agents MUST discard any received Reconfigure messages.

Clients MUST discard any Reconfigure messages 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 contain a Reconfigure Message option and the msg-type must be a valid value.
- the message includes any IA options and the msg-type in the Reconfigure Message option is INFORMATION-REQUEST.
- the message does not include DHCP authentication:
 - * the message does not contain an authentication option.
 - * the message does not pass the authentication validation performed by the client.

15.12. 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.

15.13. Relay-forward Message

Clients MUST discard any received Relay-forward messages.

15.14. Relay-reply Message

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

16. Client Source Address and Interface Selection

When a client sends a DHCP message to the All_DHCP_Relay_Agents_and_Servers address, it SHOULD send the message through the interface for which configuration information is being requested. However, the client MAY send the message through another interface attached to the same link, if and only if the client is certain the two interfaces are attached to the same link. The client MUST use a link-local address assigned to the interface for which it is requesting configuration information as the source address in the header of the IP datagram.

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 IP 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. DHCP Server Solicitation

This section describes how a client locates servers that will assign addresses to IAs belonging to the client.

The client is responsible for creating IAs and requesting that a server assign IPv6 addresses to the IA. The client first creates an IA and assigns it an IAID. The client then transmits a *Solicit* message containing an IA option describing the IA. Servers that can assign addresses to the IA respond to the client with an *Advertise* message. The client then initiates a configuration exchange as described in [section 18](#).

If the client will accept a *Reply* message with committed address assignments and other resources in response to the *Solicit* message, the client includes a Rapid Commit option (see [section 22.14](#)) in the *Solicit* message.

17.1. Client Behavior

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

17.1.1. Creation 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 addresses. The client MAY include addresses in the IAs as a hint to the server about addresses for which the client has a preference. The client MUST NOT include any other options in the Solicit message, except as specifically allowed in the definition of individual options.

The client uses IA_NA options to request the assignment of non-temporary addresses and uses IA_TA options to request the assignment of temporary addresses. Either IA_NA or IA_TA options, or a combination of both, can be included in DHCP messages.

The client SHOULD include an Option Request option (see [section 22.7](#)) to indicate the 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 22.20](#)) if the client is willing to accept Reconfigure messages from the server.

[17.1.2.](#) Transmission of Solicit Messages

The first Solicit message from the client on the interface MUST be delayed by a random amount of time between 0 and SOL_MAX_DELAY. In the case of a Solicit message transmitted when DHCP is initiated by IPv6 Neighbor Discovery, the delay gives the amount of time to wait after IPv6 Neighbor Discovery causes the client to invoke the stateful address autoconfiguration protocol (see section 5.5.3 of [RFC 2462](#)). This random delay desynchronizes clients which start at the same time (for example, after a power outage).

The client transmits the message according to [section 14](#), using the following parameters:

IRT SOL_TIMEOUT

MRT SOL_MAX_RT

MRC 0

MRD 0

If the client has included a Rapid Commit option in its Solicit message, the client terminates the waiting process as soon as a Reply message with a Rapid Commit option is received.

If the client is waiting for an Advertise message, the mechanism in [section 14](#) 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 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 Advertise messages for the first RT seconds, unless it receives an Advertise message with a preference value of 255. The preference value is carried in the Preference option ([section 22.8](#)). Any Advertise that does not include a Preference option is considered to have a preference value of 0. If the client receives an 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](#)) by sending a Request message to the server from which the Advertise message was received. If the client receives an 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 an Advertise message, the client SHOULD continue with a client-initiated message exchange by sending a Request message.

If the client does not receive any Advertise messages before the first RT has elapsed, it begins the retransmission mechanism described in [section 14](#). The client terminates the retransmission process as soon as it receives any 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.

17.1.3. Receipt of Advertise Messages

The client MUST ignore any Advertise message that includes a Status Code option containing the value NoAddrsAvail, with the exception that the client MAY display the associated status message to the user.

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. For example, the client may choose a server that returned an advertisement with configuration options 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 addresses advertised in IAs.

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.

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.

17.1.4. Receipt of Reply Message

If the client includes a Rapid Commit option in the Solicit message, it will expect a Reply message that includes a Rapid Commit option in response. The client discards any Reply messages it receives that do not include a Rapid Commit option. If the client receives a valid Reply message that includes a Rapid Commit option, it processes the message as described in [section 18.1.8](#). If it does not receive such a Reply message and does receive a valid Advertise message, the client processes the Advertise message as described in [section 17.1.3](#).

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 18.1.8](#), 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.

[17.2.](#) Server Behavior

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

[17.2.1.](#) Receipt of Solicit Messages

The server determines the information about the client and its location as described in [section 11](#) 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 indicates with a Reconfigure Accept option in the Solicit message that it will not accept a Reconfigure message, the servers discard the Solicit message.

If the client has included a Rapid Commit option in the Solicit message and the server has been configured to respond with committed address assignments and other resources, the server responds to the Solicit with a Reply message as described in [section 17.2.3](#). Otherwise, the server ignores the Rapid Commit option and processes the remainder of the message as if no Rapid Commit option were present.

[17.2.2.](#) Creation and Transmission 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 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 require that the client accept Reconfigure messages.

The server includes 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. If the client has included an Option Request option in the Solicit message, the server includes options in the Advertise 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 must be aware of the recommendations on packet sizes and the use of fragmentation in [section 5 of RFC 2460](#).

If the Solicit message from the client included one or more IA options, the server MUST include IA options in the Advertise message containing any addresses that would be assigned to IAs contained in the Solicit message from the client. If the client has included addresses in the IAs in the Solicit message, the server uses those addresses as hints about the addresses the client would like to receive.

If the server will not assign any addresses to any IAs in a subsequent Request from the client, the server MUST send an Advertise message to the client that includes only a Status Code option with code NoAddrsAvail and a status message for the user, a Server Identifier option with the server's DUID, and a Client Identifier option with the client's DUID.

If the Solicit message was received directly by the server, the server unicasts the Advertise message directly to the client using the address in the source address field from the IP datagram in which the Solicit message was received. The Advertise message MUST be unicast on the link from which the Solicit message was received.

If the Solicit message was received in a Relay-forward message, the server constructs a Relay-reply message with the Advertise message in the payload of a "relay-message" option. 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.

[17.2.3.](#) Creation and Transmission of Reply Messages

The server MUST commit the assignment of any addresses or other configuration information message before sending a Reply message to a client in response to a Solicit message.

DISCUSSION:

When using the Solicit-Reply message exchange, the server commits the assignment of any addresses before sending the Reply message. The client can assume it has been assigned the addresses in the Reply message and does not need to send a Request message for those addresses.

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 addresses from one of the servers, while the addresses from the other servers will be committed to the client but not used by the client.

The server includes a Rapid Commit option in the Reply message to indicate that the Reply is in response to a Solicit message.

The server includes a Reconfigure Accept option if the server wants to require that the client accept Reconfigure messages.

The server produces the Reply message as though it had received a Request message, as described in [section 18.2.1](#). The server transmits the Reply message as described in [section 18.2.8](#).

[18.](#) DHCP Client-Initiated Configuration Exchange

A client initiates a message exchange with a server or servers to acquire or update configuration information of interest. The client may initiate the configuration exchange as part of the operating system configuration process, when requested to do so by the application layer, when required by Stateless Address Autoconfiguration or as required to extend the lifetime of an address (Renew and Rebind messages).

18.1. Client Behavior

A client uses Request, Renew, Rebind, Release and Decline messages during the normal life cycle of addresses. It uses Confirm to validate addresses when it may have moved to a new link. It uses Information-Request messages when it needs configuration information but no addresses.

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 22.12](#)) from the server, the client SHOULD unicast any Request, Renew, Release and Decline messages to the server.

DISCUSSION:

Use of unicast may avoid delays due to the relaying of messages by relay agents, as well as avoid overhead and duplicate responses by servers due to the delivery of client messages to multiple servers. 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.

18.1.1. Creation and Transmission of Request Messages

The client uses a Request message to populate IAs with addresses and obtain other configuration information. The client includes one or more IA options in the Request message. The server then returns addresses 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 places 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 (if the client is requesting that the server assign it some network addresses).

The client MUST include an Option Request option (see [section 22.7](#)) to indicate the 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 includes a Reconfigure Accept option (see [section 22.20](#)) indicating whether or not the client is willing to accept Reconfigure messages from the server.

The client transmits the message according to [section 14](#), 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 17](#).
- Terminate the configuration process and report failure.

[18.1.2.](#) Creation and Transmission of Confirm Messages

Whenever a client may have moved to a new link, the prefixes from the 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.
- o The client is physically connected to a wired connection.
- o The client returns from sleep mode.
- o The client using a wireless technology changes access points.

In any situation when a client may have moved to a new link, the client MUST 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.

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 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 14](#), 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 14](#), the client SHOULD continue to use any IP addresses, using the last known lifetimes for those addresses, and SHOULD continue to use any other previously obtained configuration parameters.

[18.1.3.](#) Creation and Transmission of Renew Messages

To extend the valid and preferred lifetimes for the addresses associated with an IA, the client sends a Renew message to the server from which the client obtained the addresses in the IA containing an IA option for the IA. The client includes IA Address options in the IA option for the addresses associated with the IA. The server determines new lifetimes for the addresses in the IA according to the administrative configuration of the server. The server may also add

new addresses to the IA. The server may remove addresses from the IA by setting the preferred and valid lifetimes of those addresses to zero.

The server controls the time at which the client contacts the server to extend the lifetimes on assigned addresses through the T1 and T2 parameters assigned to an IA.

At time T1 for an IA, the client initiates a Renew/Reply message exchange to extend the lifetimes on any addresses in the IA. The client includes an IA option with all addresses currently assigned to the IA in its Renew message.

If T1 or T2 is set to 0 by the server (for an IA_NA) or there are no T1 or T2 times (for an IA_TA), the client may send a Renew or Rebind message, respectively, at the client's discretion.

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 places 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 appropriate options, including one or more IA options. The client MUST include the list of addresses the client currently has associated with the IAs in the Renew message.

The client MUST include an Option Request option (see [section 22.7](#)) to indicate the 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 14](#), using the following parameters:

IRT	REN_TIMEOUT
MRT	REN_MAX_RT
MRC	0
MRD	Remaining time until T2

The message exchange is terminated when time T2 is reached (see [section 18.1.4](#)), at which time the client begins a Rebind message exchange.

18.1.4. Creation and Transmission of Rebind Messages

At time T2 for an IA (which will only be reached if the server to which the Renew message was sent at time T1 has not responded), the client initiates a Rebind/Reply message exchange with any available server. The client includes an IA option with all addresses currently assigned to the IA in its Rebind message.

The client sets the "msg-type" field to REBIND. 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 adds any appropriate options, including one or more IA options. The client MUST include the list of addresses the client currently has associated with the IAs in the Rebind message.

The client MUST include an Option Request option (see [section 22.7](#)) to indicate the 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 14](#), using the following parameters:

IRT REB_TIMEOUT

MRT REB_MAX_RT

MRC 0

MRD Remaining time until valid lifetimes of all addresses have expired

The message exchange is terminated when the valid lifetimes of all the addresses assigned to the IA expire (see [section 10](#)), at which time the client has several alternative actions to choose from; for example:

- The client may choose to use a Solicit message to locate a new DHCP server and send a Request for the expired IA to the new server.

- The client may have other addresses in other IAs, so the client may choose to discard the expired IA and use the addresses in the other IAs.

18.1.5. Creation and Transmission of Information-request Messages

The client uses an Information-request message to obtain configuration information without having addresses 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. 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 a Client Identifier option if the Information-Request message will be authenticated.

The client MUST include an Option Request option (see [section 22.7](#)) to indicate the 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 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 14](#), using the following parameters:

IRT INF_TIMEOUT

MRT INF_MAX_RT

MRC 0

MRD 0

18.1.6. Creation and Transmission of Release Messages

To release one or more addresses, 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 address(es) in a Server Identifier option.

The client MUST include a Client Identifier option to identify itself to the server. The client includes options containing the IAs for the addresses it is releasing in the "options" field. The addresses to be released MUST be included in the IAs. Any addresses for the IAs the client wishes to continue to use MUST NOT be added to the IAs.

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 14](#), using the following parameters:

IRT REL_TIMEOUT

MRT 0

MRC REL_MAX_RC

MRD 0

The client MUST stop using all of the addresses being released as soon as the client begins the Release message exchange process. If addresses 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 addresses, each address assigned to the IA will be reclaimed by the server when the valid lifetime of that address expires.

18.1.7. 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 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 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 included in 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 14](#), 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.

18.1.8. 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 configuration

information contained in the Reply. The client MAY choose to report any status code or message from the status code option in the Reply message.

The client SHOULD perform duplicate address detection [17] on each of the addresses in any IAs it receives in the Reply message before using that address for traffic. If any of the addresses are found to be in use on the link, the client sends a Decline message to the server as described in [section 18.1.7](#).

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.
- Add any new addresses in the IA option to the IA as recorded by the client.
- Update lifetimes for any addresses in the IA option that the client already has recorded in the IA.
- Discard any addresses from the IA, as recorded by the client, that have a valid lifetime of 0 in the IA Address option.
- Leave unchanged any information about addresses the client has recorded in the IA but that were not included in the IA from the server.

Management of the specific configuration information is detailed in the definition of each option in [section 22](#).

If the client receives a Reply message with a Status Code containing UnspecFail, the server is indicating that it was unable to process the 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.

When the client receives a Reply message with a Status Code option with the value 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.

When the client receives a NotOnLink status from the server in response to a Confirm message, the client performs DHCP server solicitation, as described in [section 17](#), and client-initiated configuration as described in [section 18](#). If the client receives any Reply messages that do not indicate a NotOnLink status, the client can use the addresses in the IA and ignore any messages that indicate a NotOnLink status.

When the client receives a NotOnLink status from the server in response to a Request, the client can either re-issue the Request without specifying any addresses or restart the DHCP server discovery process (see [section 17](#)).

The client examines the status code in each IA individually. If the status code is NoAddrsAvail, the client has received no usable addresses in the IA and may choose to try obtaining addresses for the IA from another server. The client uses addresses and other information from any IAs that do not contain a Status Code option with the NoAddrsAvail code. If the client receives no addresses in any of the IAs, it 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 examines each IA independently. For each IA in the original Renew or Rebind message, the client:

- sends a Request message if the IA contained a Status Code option with the NoBinding status (and does not send any additional Renew/Rebind messages)
- sends a Renew/Rebind if the IA is not in the Reply message
- otherwise accepts the information in the IA

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. Server Behavior](#)

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

In most instances, the server will send a Reply in response to a client message. This Reply message 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 Reply 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 RFC 2460](#). If the client included an Option Request option in its message, the server includes options in the Reply 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.

18.2.1. Receipt of Request Messages

When the server receives a Request message via unicast from a client to which the server has not sent a unicast option, the server discards the Request message and responds with a Reply message containing a Status Code option with the value UseMulticast, a Server Identifier option containing the server's DUID, the Client Identifier option from the client message, and no other options.

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.

If the server finds that the prefix on one or more IP addresses in any IA in the message from the client 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 cannot assign any addresses to an IA in the message from the client, the server MUST include the IA in the Reply message with no addresses in the IA and a Status Code option in the IA containing status code NoAddrsAvail.

For any IAs to which the server can assign addresses, the server includes the IA with addresses and other configuration parameters, and records the IA as a new client binding.

The server includes a Reconfigure Accept option if the server wants to require that the client accept Reconfigure messages.

The server includes other options containing configuration information to be returned to the client as described in [section 18.2](#).

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 client has resent a Request message for which it did not receive a Reply message. The server either resends a previously cached Reply message or sends a new Reply message.

[18.2.2. Receipt of Confirm Messages](#)

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

The server ignores the T1 and T2 fields in the IA options and the preferred-lifetime and valid-lifetime fields in the IA Address options.

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.

18.2.3. Receipt of Renew Messages

When the server receives a Renew message via unicast from a client to which the server has not sent a unicast option, the server discards the Renew message and responds with a Reply message containing a Status Code option with the value UseMulticast, a Server Identifier option containing the server's DUID, the Client Identifier option from the client message, and no other options.

When the server receives a Renew 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 cannot find a client entry for the IA the server returns the IA containing no addresses with a Status Code option set to NoBinding in the Reply message.

If the server finds that any of the addresses 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 the addresses in the IA for the client then the server sends back the IA to the client with new lifetimes and T1/T2 times. The server may choose to change the list of addresses and the lifetimes of addresses in IAs that are returned to the client.

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 18.2](#).

18.2.4. Receipt of Rebind Messages

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

If the server cannot find a client entry for the IA and the server determines that the addresses in the IA are not appropriate for the link to which the client's interface is attached according to the server's explicit configuration information, the server MAY send a Reply message to the client containing the client's IA, with the lifetimes for the addresses in the IA set to zero. This Reply constitutes an explicit notification to the client that the addresses in the IA are no longer valid. In this situation, if the server does not send a Reply message it silently discards the Rebind message.

If the server finds that any of the addresses are no longer 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 the addresses in the IA for the client then the server SHOULD send back the IA to the client with new lifetimes and T1/T2 times.

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 18.2](#).

[18.2.5.](#) Receipt of Information-request Messages

When the server receives an Information-request message, the client is requesting configuration information that does not include the assignment of any addresses. 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 18.2](#).

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.

[18.2.6.](#) Receipt of Release Messages

When the server receives a Release message via unicast from a client to which the server has not sent a unicast option, the server discards the Release message and responds with a Reply 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, and no other options.

Upon the receipt of a valid Release 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 and makes the addresses available for assignment to other clients. The server ignores addresses not assigned to the IA, although it may choose to log an error.

After all the addresses 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 addresses and IAs after the lifetimes on the addresses have expired to allow the server to reassign the previously assigned addresses to a client.

[18.2.7.](#) Receipt of Decline Messages

When the server receives a Decline message via unicast from a client to which the server has not sent a unicast option, the server discards the Decline message and responds with a Reply message containing a Status Code option with the value UseMulticast, a Server Identifier option containing the server's DUID, the Client Identifier option from the client message, and no other options.

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 and 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 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.

[18.2.8.](#) Transmission of Reply Messages

If the original message was received directly by the server, the server unicasts the 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 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 22.10](#)). 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.

[19.](#) DHCP Server-Initiated Configuration Exchange

A server initiates a configuration exchange to cause DHCP clients to obtain new addresses 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. Other

examples include changes in the location of directory servers, addition of new services such as printing, and availability of new software.

19.1. Server Behavior

A server sends a Reconfigure message to cause a client to initiate immediately a Renew/Reply or Information-request/Reply message exchange with the server.

19.1.1. 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.

The server MAY include an Option Request option to inform the client of what information has been changed or new information that has been added. In particular, the server specifies the IA option in the Option Request option if the server wants the client to obtain new address information. If the server identifies the IA option in the Option Request option, the server MUST include an IA option that contains no other sub-options to identify each IA that is to be reconfigured on the client.

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 22.19](#)) to select whether the client responds with a Renew 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 20.3](#)) 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 or Information-request/Reply message exchange with the server. The server interprets the receipt of a Renew or Information-request message (whichever was specified in the original Reconfigure message) from the client as satisfying the Reconfigure message request.

[19.1.2.](#) Time Out and Retransmission of Reconfigure Messages

If the server does not receive a Renew 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 5.5](#).

[19.2.](#) Receipt of Renew Messages

The server generates and sends a Reply message to the client as described in sections [18.2.3](#) and [18.2.8](#), including options for configuration parameters.

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 Renew message from the client.

[19.3.](#) Receipt of Information-request Messages

The server generates and sends a Reply message to the client as described in sections [18.2.5](#) and [18.2.8](#), including options for configuration parameters.

The server MAY include options containing new values for other configuration parameters in the Reply message, even if those parameters were not requested in the Information-request message from the client.

19.4. Client Behavior

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.

19.4.1. Receipt of Reconfigure Messages

Upon receipt of a valid Reconfigure message, the client responds with either a Renew message or an Information-request message as indicated by the Reconfigure Message option (as defined in [section 22.19](#)). The client ignores the transaction-id field in the received Reconfigure message. While the transaction is in progress, the client silently discards any Reconfigure messages it receives.

DISCUSSION:

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 or Information-request message if necessary); the client ignores any additional Reconfigure messages until the exchange is complete. Subsequent Reconfigure messages cause the client to initiate a new exchange.

How does this mechanism work in the face of duplicated or retransmitted Reconfigure messages? 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 can discontinue retransmission of Reconfigure messages to the client once the server receives the Renew 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.

19.4.2. Creation and Transmission of Renew Messages

When responding to a Reconfigure, the client creates and sends the Renew message in exactly the same manner as outlined in [section 18.1.3](#), with the exception that the client copies the Option Request option and any IA options from the Reconfigure message into the Renew message.

19.4.3. Creation and Transmission of Information-request Messages

When responding to a Reconfigure, the client creates and sends the Information-request message in exactly the same manner as outlined in [section 18.1.5](#), with the exception that the client includes a Server Identifier option with the identifier from the Reconfigure message to which the client is responding.

19.4.4. Time Out and Retransmission of Renew or Information-request Messages

The client uses the same variables and retransmission algorithm as it does with Renew or Information-request messages generated as part of a client-initiated configuration exchange. See sections [18.1.3](#) and [18.1.5](#) for details. If the client does not receive a response from the server by the end of the retransmission process, the client ignores and discards the Reconfigure message.

19.4.5. Receipt of Reply Messages

Upon the receipt of a valid Reply message, the client processes the options and sets (or resets) configuration parameters appropriately. The client records and updates the lifetimes for any addresses specified in IAs in the Reply message.

20. Relay Agent Behavior

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

20.1. 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 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 to 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.

20.1.1. 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 or site-scoped address with a prefix assigned to the link on which the client should be assigned an address in the link-address field. This address will be used by the server to determine the link from which the client should be assigned an address 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 22.18](#)) in the Relay-forward message. The server will include the Interface-id option in its Relay-reply message. The relay agent fills in the link-address field as described in the previous paragraph regardless of whether the relay agent includes an Interface-id option in the Relay-forward message.

20.1.2. 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 client 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 or site-local address (and the device on which the relay agent is running belongs to only one site), the relay agent sets the link-address field to 0; otherwise the relay agent sets the link-address field to a global or site-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.

20.2. Relaying a Relay-reply Message

The relay agent processes any options included in the Relay-reply message in addition to the Relay Message option, and then discards those options.

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.

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.

20.3. 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>
```

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.

21. Authentication of DHCP Messages

Some network administrators may wish to provide authentication of the source and contents of DHCP messages. For example, clients may be subject to denial of service attacks through the use of bogus DHCP servers, or may simply be misconfigured due to unintentionally instantiated DHCP servers. Network administrators may wish to constrain the allocation of addresses to authorized hosts to avoid denial of service attacks in "hostile" environments where the network medium is not physically secured, such as wireless networks or college residence halls.

The DHCP authentication mechanism is based on the design of authentication for DHCPv4 [4].

21.1. Security of Messages Sent Between Servers and Relay Agents

Relay agents and servers that exchange messages securely use the IPsec mechanisms for IPv6 [7]. If a client message is relayed through multiple relay agents, each of the relay agents must have established independent, pairwise trust relationships. That is, if messages from client C will be relayed by relay agent A to relay

agent B and then to the server, relay agents A and B must be configured to use IPsec for the messages they exchange, and relay agent B and the server must be configured to use IPsec for the messages they exchange.

Relay agents and servers that support secure relay agent to server or relay agent to relay agent communication use IPsec under the following conditions:

Selectors	Relay agents are manually configured with the addresses of the relay agent or server to which DHCP messages are to be forwarded. Each relay agent and server that will be using IPsec for securing DHCP messages must also be configured with a list of the relay agents to which messages will be returned. The selectors for the relay agents and servers will be the pairs of addresses defining relay agents and servers that exchange DHCP messages on the DHCPv6 UDP ports 546 and 547.
Mode	Relay agents and servers use transport mode and ESP. The information in DHCP messages is not generally considered confidential, so encryption need not be used (i.e., NULL encryption can be used).
Key management	Because the relay agents and servers are used within an organization, public key schemes are not necessary. Because the relay agents and servers must be manually configured, manually configured key management may suffice, but does not provide defense against replayed messages. Accordingly, IKE with preshared secrets SHOULD be supported. IKE with public keys MAY be supported.
Security policy	DHCP messages between relay agents and servers should only be accepted from DHCP peers as identified in the local configuration.
Authentication	Shared keys, indexed to the source IP address of the received DHCP message, are adequate in this application.
Availability	Appropriate IPsec implementations are likely to be available for servers and for relay agents in more featureful devices used in enterprise and

core ISP networks. IPsec is less likely to be available for relay agents in low end devices primarily used in the home or small office markets.

21.2. Summary of DHCP Authentication

Authentication of DHCP messages is accomplished through the use of the Authentication option (see [section 22.11](#)). 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. Two such protocols are defined here. 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.

21.3. 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 monotonically increasing counter. Using a counter value, such as the current time of day (for example, an NTP-format timestamp [\[9\]](#)), can reduce the danger of replay attacks. This method MUST be supported by all protocols.

21.4. Delayed Authentication Protocol

If the protocol field is 2, the message is using the "delayed authentication" mechanism. In delayed authentication, the client requests authentication in its Solicit message, and the server replies with an Advertise message that includes authentication

information. This authentication information contains a nonce value generated by the source as a message authentication code (MAC) to provide message authentication and entity authentication.

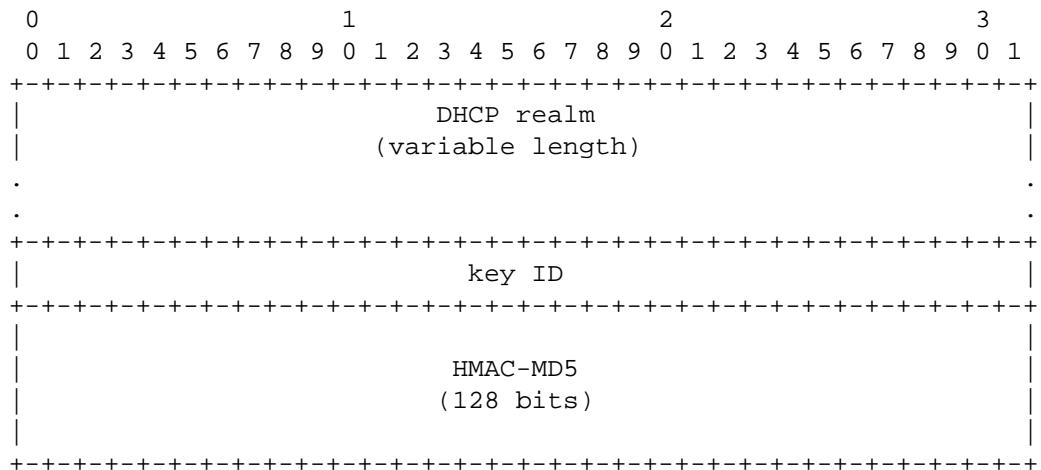
The use of a particular technique based on the HMAC protocol [8] using the MD5 hash [16] is defined here.

21.4.1. Use of the Authentication Option in the Delayed Authentication Protocol

In a Solicit message, the client fills in the protocol, algorithm and RDM fields in the Authentication option with the client's preferences. The client sets the replay detection field to zero and omits the authentication information field. The client sets the option-len field to 11.

In all other messages, the protocol and algorithm fields identify the method used to construct the contents of the authentication information field. The RDM field identifies the method used to construct the contents of the replay detection field.

The format of the Authentication information is:



DHCP realm The DHCP realm that identifies the key used to generate the HMAC-MD5 value.

key ID The key identifier that identified the key used to generate the HMAC-MD5 value.

HMAC-MD5 The message authentication code generated by applying MD5 to the DHCP message using the key identified by the DHCP realm, client DUID, and key ID.

The sender computes the MAC using the HMAC generation algorithm [8] and the MD5 hash function [16]. The entire DHCP message (setting the MAC field of the authentication option to zero), including the DHCP message header and the options field, is used as input to the HMAC-MD5 computation function.

DISCUSSION:

Algorithm 1 specifies the use of HMAC-MD5. Use of a different technique, such as HMAC-SHA, will be specified as a separate protocol.

The DHCP realm used to identify authentication keys is chosen to be unique among administrative domains. Use of the DHCP realm allows DHCP administrators to avoid conflict in the use of key identifiers, and allows a host using DHCP to use authenticated DHCP while roaming among DHCP administrative domains.

21.4.2. Message Validation

Any DHCP message that includes more than one authentication option MUST be discarded.

To validate an incoming message, the receiver first checks that the value in the replay detection field is acceptable according to the replay detection method specified by the RDM field. Next, the receiver computes the MAC as described in [8]. The entire DHCP message (setting the MAC field of the authentication option to 0) is used as input to the HMAC-MD5 computation function. If the MAC computed by the receiver does not match the MAC contained in the authentication option, the receiver MUST discard the DHCP message.

21.4.3. Key Utilization

Each DHCP client has a set of keys. Each key is identified by <DHCP realm, client DUID, key id>. Each key also has a lifetime. The key may not be used past the end of its lifetime. The client's keys are initially distributed to the client through some out-of-band mechanism. The lifetime for each key is distributed with the key. Mechanisms for key distribution and lifetime specification are beyond the scope of this document.

The client and server use one of the client's keys to authenticate DHCP messages during a session (until the next Solicit message sent by the client).

21.4.4. Client Considerations for Delayed Authentication Protocol

The client announces its intention to use DHCP authentication by including an Authentication option in its Solicit message. The server selects a key for the client based on the client's DUID. The client and server use that key to authenticate all DHCP messages exchanged during the session.

21.4.4.1. Sending Solicit Messages

When the client sends a Solicit message and wishes to use authentication, it includes an Authentication option with the desired protocol, algorithm and RDM as described in [section 21.4](#). The client does not include any replay detection or authentication information in the Authentication option.

21.4.4.2. Receiving Advertise Messages

The client validates any Advertise messages containing an Authentication option specifying the delayed authentication protocol using the validation test described in [section 21.4.2](#).

Client behavior, if no Advertise messages include authentication information or pass the validation test, is controlled by local policy on the client. According to client policy, the client MAY choose to respond to an Advertise message that has not been authenticated.

The decision to set local policy to accept unauthenticated messages should be made with care. Accepting an unauthenticated Advertise message can make the client vulnerable to spoofing and other attacks. If local users are not explicitly informed that the client has accepted an unauthenticated Advertise message, the users may incorrectly assume that the client has received an authenticated address and is not subject to DHCP attacks through unauthenticated messages.

A client MUST be configurable to discard unauthenticated messages, and SHOULD be configured by default to discard unauthenticated messages if the client has been configured with an authentication key or other authentication information. A client MAY choose to differentiate between Advertise messages with no authentication information and Advertise messages that do not pass the validation test; for example, a client might accept the former and discard the latter. If a client does accept an unauthenticated message, the client SHOULD inform any local users and SHOULD log the event.

21.4.4.3. Sending Request, Confirm, Renew, Rebind, Decline or Release Messages

If the client authenticated the Advertise message through which the client selected the server, the client MUST generate authentication information for subsequent Request, Confirm, Renew, Rebind or Release messages sent to the server, as described in [section 21.4](#). When the client sends a subsequent message, it MUST use the same key used by the server to generate the authentication information.

21.4.4.4. Sending Information-request Messages

If the server has selected a key for the client in a previous message exchange (see [section 21.4.5.1](#)), the client MUST use the same key to generate the authentication information throughout the session.

21.4.4.5. Receiving Reply Messages

If the client authenticated the Advertise it accepted, the client MUST validate the associated Reply message from the server. The client MUST discard the Reply if the message fails to pass the validation test and MAY log the validation failure. If the Reply fails to pass the validation test, the client MUST restart the DHCP configuration process by sending a Solicit message.

If the client accepted an Advertise message that did not include authentication information or did not pass the validation test, the client MAY accept an unauthenticated Reply message from the server.

21.4.4.6. Receiving Reconfigure Messages

The client MUST discard the Reconfigure if the message fails to pass the validation test and MAY log the validation failure.

21.4.5. Server Considerations for Delayed Authentication Protocol

After receiving a Solicit message that contains an Authentication option, the server selects a key for the client, based on the client's DUID and key selection policies with which the server has been configured. The server identifies the selected key in the Advertise message and uses the key to validate subsequent messages between the client and the server.

21.4.5.1. Receiving Solicit Messages and Sending Advertise Messages

The server selects a key for the client and includes authentication information in the Advertise message returned to the client as specified in [section 21.4](#). The server MUST record the identifier of the key selected for the client and use that same key for validating subsequent messages with the client.

21.4.5.2. Receiving Request, Confirm, Renew, Rebind or Release Messages and Sending Reply Messages

The server uses the key identified in the message and validates the message as specified in [section 21.4.2](#). If the message fails to pass the validation test or the server does not know the key identified by the 'key ID' field, the server MUST discard the message and MAY choose to log the validation failure.

If the message passes the validation test, the server responds to the specific message as described in [section 18.2](#). The server MUST include authentication information generated using the key identified in the received message, as specified in [section 21.4](#).

21.5. 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 are not using any other authentication protocol and the client and server have negotiated to use Reconfigure messages.

21.5.1. 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)		
+-----+			
.	.	.	.
.	.	.	.
	+-----+		
+-----+			

Type Type of data in 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).

Value Data as defined by field.

21.5.2. Server considerations for Reconfigure Key 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.

[21.5.3.](#) Client considerations for Reconfigure Key protocol

The client will receive a Reconfigure Key from the server 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.

[22.](#) 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 22.1](#). 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.

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.

22.1. 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)			
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			

option-code An unsigned integer identifying the specific option type carried in this option.

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

option-data The data for the option; the format of this data depends on the definition of the option.

DHCPv6 options are scoped by using encapsulation. Some options apply generally to the client, some are specific to an IA, and some are specific to the addresses within an IA. These latter two cases are discussed in sections [22.4](#) and [22.6](#).

22.2. Client Identifier Option

The Client Identifier option is used to carry a DUID (see [section 9](#)) 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)	.
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			

option-code OPTION_CLIENTID (1).

option-len Length of DUID in octets.

DUID The DUID for the client.

22.3. Server Identifier Option

The Server Identifier option is used to carry a DUID (see [section 9](#)) identifying a server between a client and a server. The format of the Server Identifier option is:

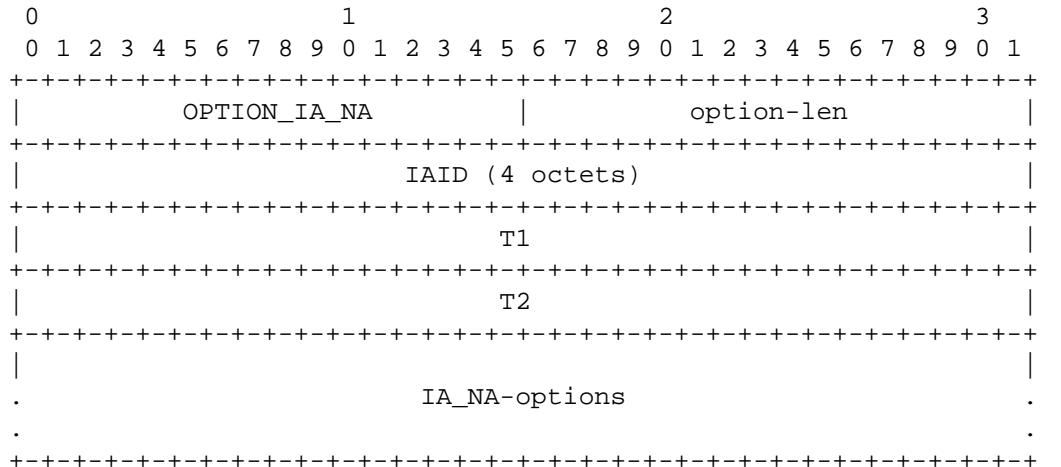
0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1			
+-----+			
OPTION_SERVERID option-len			
+-----+			
.			
DUID			
(variable length)			
.			
.			
+-----+			
option-code OPTION_SERVERID (2).			
option-len Length of DUID in octets.			
DUID The DUID for the server.			

22.4. 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 22.5](#)).

The format of the IA_NA option is:



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

IAID The unique identifier for this IA_NA; the IAIID must be unique among the identifiers for all of this client's IA_NAs. The number space for IA_NA IAIIDs is separate from the number space for IA_TA IAIIDs.

T1 The time at which the client contacts 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.

T2 The time at which the client contacts 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.

IA NA-options Options associated with this IA NA.

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.

An IA_NA option may only appear in the options area of a DHCP message. A DHCP message may contain multiple IA_NA options.

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, values in the T1 and T2 fields indicate the client's preference for those parameters. The client sets T1 and T2 to 0 if it has no preference for those values. 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.

If a server receives an IA_NA with T1 greater than T2, and both T1 and T2 are greater than 0, the server ignores the invalid values of T1 and T2 and processes the IA_NA as though the client had set T1 and T2 to 0.

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.

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.

22.5. 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 [RFC 3041](#) [12]. The format of the IA_TA option is:

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; that is, at most one address from each prefix assigned to the link to which the client is attached.

An IA_TA option may only appear in the options area of a DHCP message. A DHCP message may contain multiple IA_TA options.

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 lifetimes 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 lifetime of a temporary address to allow an application to continue to use an established TCP connection.

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 [RFC 3041](#). 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.

This option MAY appear in a Confirm message if the lifetimes on the temporary addresses in the associated IA have not expired.

22.6. IA Address Option

The IA Address option is used to specify IPv6 addresses 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 Options field 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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
OPTION_IAADDR option-len			
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
IPv6 address			
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
preferred-lifetime			
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
valid-lifetime			
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
.			
.			
IAaddr-options			
.			
.			
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			

option-code OPTION_IAADDR (5).

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

IPv6 address An IPv6 address.

preferred-lifetime The preferred lifetime for the IPv6 address in the option, expressed in units of seconds.

valid-lifetime The valid lifetime for the IPv6 address in the option, expressed in units of seconds.

IAaddr-options Options associated with this address.

In a message sent by a client to a server, values in the preferred and valid lifetime fields indicate the client's preference for those parameters. The client may send 0 if it has no preference for the preferred and valid lifetimes. 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.

A client discards any addresses for which the preferred lifetime is greater than the valid lifetime. A server ignores the lifetimes set by the client if the preferred lifetime is greater than the valid lifetime and ignores the values for T1 and T2 set by the client if those values are greater than the preferred lifetime.

Care should be taken in setting the valid lifetime of an address to 0xffffffff ("infinity"), which amounts to a permanent assignment of an address to a client.

An IA Address option may appear only in an IA_NA option or an IA_TA option. 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.

22.7. 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 2 3 4 5 6 7 8 9 0 1	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+				
OPTION_ORO		option-len		
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+				
requested-option-code-1		requested-option-code-2		
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+				
...				
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+				

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.

A client MAY include an Option Request option in a Solicit, Request, Renew, Rebind, Confirm or Information-request message to inform the server about options the client wants the server to send to the client. A server MAY include an Option Request option in a Reconfigure option to indicate which options the client should request from the server.

22.8. 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   |
+-----+
option-code      OPTION_PREFERENCE (7).

option-len      1.

pref-value      The preference value for the server in this message

```

A server MAY include a Preference option in an Advertise message to control the selection of a server by the client. See [section 17.1.3](#) for the use of the Preference option by the client and the interpretation of Preference option data value.

22.9. 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														
+-----+														
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 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.

22.10. 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:

```

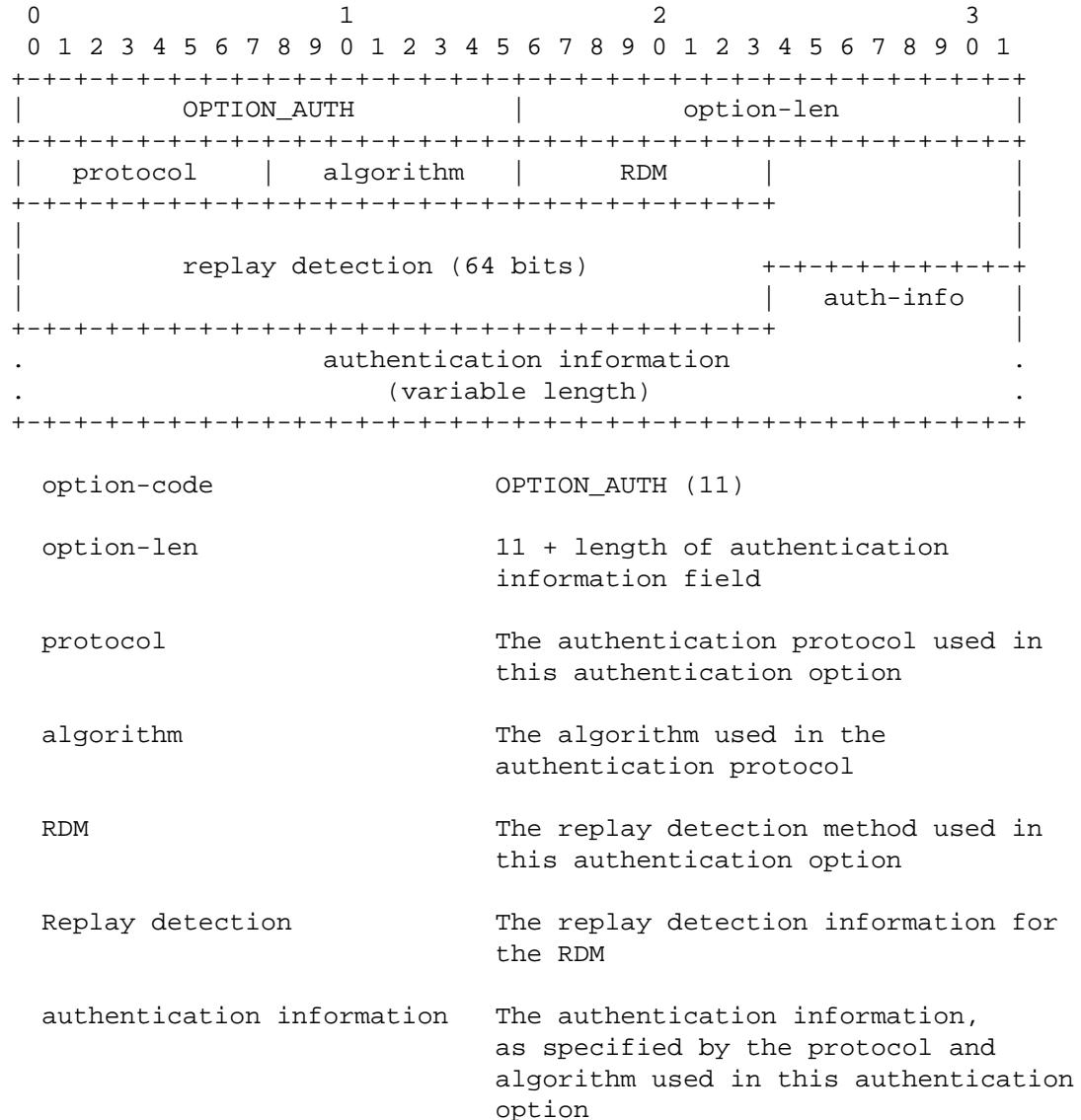
0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|       OPTION_RELAY_MSG      |       option-len      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|
|
.
.
.
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+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

```

22.11. 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 21](#). The format of the Authentication option is:



22.12. 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				
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+				
				option-code OPTION_UNICAST (12).
				option-len 16.
				server-address The IP address to which the client should send messages delivered using unicast.

The server specifies the IPv6 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 IPv6 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](#).

22.13. 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 2 3 4 5 6 7 8 9 0 1	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+				
OPTION_STATUS_CODE option-len				
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+				
status-code				
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+				
.				.
.	status-message			.
.				.
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+				
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. The status codes are defined in section 24.4 .		
status-message		A UTF-8 encoded text string suitable for display to an end user, which MUST NOT be null-terminated.		

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.

22.14. 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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
OPTION_RAPID_COMMIT 0			
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
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 17.1.1](#).

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 assigned addresses 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 addresses that are not actually used by the client.

The problem of unused addresses can be minimized, for example, by designing the DHCP service so that only one server responds to the Solicit or by using relatively short lifetimes for assigned addresses.

22.15. 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 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      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
.
.
.
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
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 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:

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.

22.16. 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 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                    |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.
.
.
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
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 [6].

vendor-class-data The hardware configuration of the host on which the client is running.

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:

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

[22.17.](#) 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:

option-len 4 + length of option-data field

enterprise-number The vendor's registered Enterprise Number as registered with IANA [6].

`option-data` An opaque object of `option-len` octets, interpreted by vendor-specific code on the clients and servers

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 host device's IPv6 stack to be functional.

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

opt-code The code for the encapsulated option.

option-len An unsigned integer giving the length of the option-data field in this encapsulated option in octets.

option-data The data area for the encapsulated option.

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.

22.18. 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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
OPTION_INTERFACE_ID		option-len	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
.			
interface-id		.	
.			
.			
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
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 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.

22.19. 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 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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
OPTION_RECONF_MSG		option-len	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
msg-type		.	
+-----+-----+-----+			

```

option-code      OPTION_RECONF_MSG (19).

option-len      1.

msg-type        5 for Renew message, 11 for
                Information-request message.

```

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

[22.20.](#) 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	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+			
option-code OPTION_RECONF_ACCEPT (20).			
option-len 0.			

[23.](#) Security Considerations

The threat to DHCP is inherently an insider threat (assuming a properly configured network where DHCPv6 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.

Use of manually configured preshared 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 addresses.

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.

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 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.

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 addresses, or the exhaustion of CPU or network bandwidth, and are present anytime there is a shared resource.

In the case where relay agents add additional options to Relay Forward messages, the messages exchanged between relay agents and servers may be used to mount a "man in the middle" or denial of service attack.

This threat model does not consider the privacy of the contents of DHCP messages to be important. DHCP is not used to exchange authentication or configuration information that must be kept secret from other networks nodes.

DHCP authentication provides for authentication of the identity of DHCP clients and servers, and for the integrity of messages delivered between DHCP clients and servers. DHCP authentication does not provide any privacy for the contents of DHCP messages.

The Delayed Authentication protocol described in [section 21.4](#) uses a secret key that is shared between a client and a server. The use of a "DHCP realm" in the shared key allows identification of administrative domains so that a client can select the appropriate key or keys when roaming between administrative domains. However, the Delayed Authentication protocol does not define any mechanism for sharing of keys, so a client may require separate keys for each administrative domain it encounters. The use of shared keys may not scale well and does not provide for repudiation of compromised keys. This protocol is focused on solving the intradomain problem where the out-of-band exchange of a shared key is feasible.

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 21.5](#) 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 to the client.

Communication between a server and a relay agent, and communication between relay agents, can be secured through the use of IPSec, as described in [section 21.1](#). The use of manual configuration and installation of static keys are acceptable in this instance because relay agents and the server will belong to the same administrative domain and the relay agents will require other specific configuration (for example, configuration of the DHCP server address) as well as the IPSec configuration.

24. IANA Considerations

This document defines several new name spaces associated with DHCPv6 and DHCPv6 options:

- Message types
- Status codes
- DUID

- Option codes

IANA has established a registry of values for each of these name spaces, which are described in the remainder of this section. These name spaces will be managed by the IANA and all will be managed separately from the name spaces defined for DHCPv4.

New multicast addresses, message types, status codes, and DUID types are assigned via Standards Action [11].

New DHCP option codes are tentatively assigned after the specification for the associated option, published as an Internet Draft, has received expert review by a designated expert [11]. The final assignment of DHCP option codes is through Standards Action, as defined in RFC 2434 [11].

This document also references three name spaces in section 21 that are associated with the Authentication Option (section 22.11). These name spaces are defined by the authentication mechanism for DHCPv4 in RFC 3118 [4].

The authentication name spaces currently registered by IANA will apply to both DHCPv6 and DHCPv4. In the future, specifications that define new Protocol, Algorithm and RDM mechanisms will explicitly define whether the new mechanisms are used with DHCPv4, DHCPv6 or both.

24.1. Multicast Addresses

Section 5.1 defines the following multicast addresses, which have been assigned by IANA for use by DHCPv6:

All_DHCP_Relay_Agents_and_Servers address: FF02::1:2

All_DHCP_Servers address: FF05::1:3

24.2. DHCP Message Types

IANA has recorded the following message types (defined in [section 5.3](#)). IANA will maintain the registry of DHCP message types.

SOLICIT	1
ADVERTISE	2
REQUEST	3
CONFIRM	4
RENEW	5
REBIND	6
REPLY	7
RELEASE	8
DECLINE	9
RECONFIGURE	10
INFORMATION-REQUEST	11
RELAY-FORW	12
RELAY-REPL	13

24.3. DHCP Options

IANA has recorded the following option-codes (as defined in [section 22](#)). IANA will maintain the registry of DHCP option codes.

OPTION_CLIENTID	1
OPTION_SERVERID	2
OPTION_IA_NA	3
OPTION_IA_TA	4
OPTION_IAADDR	5
OPTION_ORO	6
OPTION_PREFERENCE	7
OPTION_ELAPSED_TIME	8
OPTION_RELAY_MSG	9
OPTION_AUTH	11
OPTION_UNICAST	12
OPTION_STATUS_CODE	13
OPTION_RAPID_COMMIT	14
OPTION_USER_CLASS	15
OPTION_VENDOR_CLASS	16
OPTION_VENDOR_OPTS	17
OPTION_INTERFACE_ID	18
OPTION_RECONF_MSG	19
OPTION_RECONF_ACCEPT	20

24.4. Status Codes

IANA has recorded the status codes defined in the following table. IANA will manage the definition of additional status codes in the future.

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 address.

24.5. DUID

IANA has recorded the following DUID types (as defined in [section 9.1](#)). IANA will manage the definition of additional DUID types in the future.

DUID-LLT	1
DUID-EN	2
DUID-LL	3

25. Acknowledgments

Thanks to the DHC Working Group and the members of the IETF for their time and input into the specification. In particular, thanks also for the consistent input, ideas, and review by (in alphabetical order) Bernard Aboba, Bill Arbaugh, Thirumalesh Bhat, Steve Bellovin, A. K. Vijayabaskar, Brian Carpenter, Matt Crawford, Francis Dupont, Richard Hussong, Kim Kinnear, Fredrik Lindholm, Tony Lindstrom, Josh Littlefield, Gerald Maguire, Jack McCann, Shin Miyakawa, Thomas Narten, Erik Nordmark, Jarno Rajahalme, Yakov Rekhter, Mark Stapp, Matt Thomas, Sue Thomson, Tatuya Jinmei and Phil Wells.

Thanks to Steve Deering and Bob Hinden, who have consistently taken the time to discuss the more complex parts of the IPv6 specifications.

And, thanks to Steve Deering for pointing out at IETF 51 in London that the DHCPv6 specification has the highest revision number of any Internet Draft.

26. References

26.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [2] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", [RFC 2464](#), December 1998.
- [3] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.
- [4] Droms, R., Ed. and W. Arbaugh, Ed., "Authentication for DHCP Messages", [RFC 3118](#), June 2001.
- [5] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 2373](#), July 1998.
- [6] IANA. Private Enterprise Numbers.
<http://www.iana.org/assignments/enterprise-numbers.html>.
- [7] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", [RFC 2401](#), November 1998.
- [8] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", [RFC 2104](#), February 1997.
- [9] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", [RFC 1305](#), March 1992.
- [10] Mockapetris, P., "Domain names - implementation and specification", [RFC 1035](#), November 1987.
- [11] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 2434](#), October 1998.
- [12] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", [RFC 3041](#), January 2001.

- [13] Narten, T., Nordmark, E. and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", [RFC 2461](#), December 1998.
- [14] Plummer, D.C., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, [RFC 826](#), November 1982.
- [15] Postel, J., "User Datagram Protocol", STD 6, [RFC 768](#), August 1980.
- [16] Rivest, R., "The MD5 Message-Digest Algorithm", [RFC 1321](#), April 1992.
- [17] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", [RFC 2462](#), December 1998.

26.2. Informative References

- [18] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", [RFC 2132](#), March 1997.
- [19] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), March 1997.
- [20] R. Droms, Ed. DNS Configuration options for DHCPv6. April 2002. Work in Progress.
- [21] A. K. Vijayabhaskar. Time Configuration Options for DHCPv6. May 2002. Work in Progress.
- [22] Vixie, P., Ed., Thomson, S., Rekhter, Y. and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", [RFC 2136](#), April 1997.

A. Appearance of Options in Message Types

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

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

NOTE:

Only included in Information-Request messages that are sent in response to a Reconfigure (see [section 19.4.3](#)).

	Status Code	Rap. Comm.	User Class	Vendor Class	Vendor Spec.	Inter. ID	Recon. Msg.	Recon. Accept
Solicit		*	*	*	*			*
Advert.	*		*	*	*			*
Request		*	*	*	*			*
Confirm		*	*	*	*			
Renew		*	*	*	*			*
Rebind		*	*	*	*			*
Decline		*	*	*	*			
Release		*	*	*	*			
Reply	*	*	*	*	*			*
Reconf.							*	
Inform.		*	*	*				*
R-forw.		*	*	*		*		
R-repl.		*	*	*	*			

B. Appearance of Options in the Options Field of DHCP Options

The following table indicates with a "*" where options can appear in the options field of other options:

	Option Field	IA_NA/ IA_TA	IAADDR	Relay Forw.	Relay Reply
Client ID	*				
Server ID	*				
IA_NA/IA_TA	*				
IAADDR		*			
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	*				

Note: "Relay Forw" / "Relay Reply" options appear in the options field of the message but may only appear in these messages.

Chair's Address

The working group can be contacted via the current chair:

Ralph Droms
Cisco Systems
1414 Massachusetts Avenue
Boxborough, MA 01719

Phone: (978) 936-1674
EMail: rdroms@cisco.com

Authors' Addresses

Jim Bound
Hewlett Packard Corporation
ZK3-3/W20
110 Spit Brook Road
Nashua, NH 03062-2698
USA

Phone: +1 603 884 0062
EMail: Jim.Bound@hp.com

Bernie Volz
116 Hawkins Pond Road
Center Harbor, NH 03226-3103
USA

Phone: +1-508-259-3734
EMail: volz@metrocast.net

Ted Lemon
Nominum, Inc.
950 Charter Street
Redwood City, CA 94043
USA

EMail: Ted.Lemon@nominum.com

Charles E. Perkins
Communications Systems Lab
Nokia Research Center
313 Fairchild Drive
Mountain View, California 94043
USA

Phone: +1-650 625-2986
EMail: charles.perkins@nokia.com

Mike Carney
Sun Microsystems, Inc
17 Network Circle
Menlo Park, CA 94025
USA

Phone: +1-650-786-4171
EMail: michael.carney@sun.com

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.