Internet Engineering Task Force
INTERNET DRAFT
DHC Working Group
Obsoletes: draft-ietf-dhc-dhcpv6-24.txt

J. Bound
Hewlett Packard
M. Carney
Sun Microsystems, Inc
C. Perkins
Nokia Research Center
Ted Lemon
Nominum
Bernie Volz
Ericsson
R. Droms(ed.)
Cisco Systems
May 24 2002

Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
draft-ietf-dhc-dhcpv6-25.txt

Status of This Memo

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" (RFC2462), and can be used

separately or concurrently with the latter to obtain configuration parameters.

Contents

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. The addresses and additional configuration 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 RFC2462, "IPv6 Stateless Address Autoconfiguration".

The remainder of this introduction summarizes DHCP, explaining the message exchange mechanisms and example message flows. The message flows in sections 1.3 and 1.4 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 [16]. The client uses its link-local address determined through stateless autoconfiguration 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 forward 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 forwarding 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. Protocol implementation

This specification for DHCP includes messages and descriptions of client and server behavior for several different functions. Some clients and servers will be deployed in situations in which not all of the functions will be required. For example, a client that uses stateless autoconfiguration to determine its IPv6 addresses would

use only the Information-request and Reply messages to obtain other configuration information.

Clients and servers that do not use all of the functions of DHCP need not implement processing for those DHCP messages that will not be used. A client or server that receives a message that it is not prepared to process may simply discard that message. For example, a DHCP server that only provides configuration information and does not do IPv6 address assignment can respond to only Information-request messages and discard other messages such as Solicit or Request messages.

1.3. Client-server exchanges involving two messages

A DHCP client can obtain configuration information such as a list of available DNS servers or NTP servers 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. The server responds 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. Because the server need not keep any dynamic state information about individual clients to support this two message exchange, a server that provides just configuration information can be realized with a relatively simple and small implementation.

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.

1.4. 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 messages 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 [2].

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 [6], IPv6 Stateless Address Autoconfiguration [18], IPv6 Neighbor Discovery Processing [14], and Dynamic Updates to DNS [19]. 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 [6] 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. This means that a client can immediately use its link-local address and a well-known multicast address to begin communications to discover neighbors on the link. For instance, a client can send a Solicit message and locate a server or relay agent.

IPv6 Stateless Address Autoconfiguration [18] specifies procedures by which a node may autoconfigure addresses based on router advertisements [14], 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 [14] is the node discovery protocol in IPv6 which replaces and enhances functions of ARP [15]. To understand

IPv6 and stateless address autoconfiguration it is strongly recommended that implementors understand IPv6 Neighbor Discovery.

Dynamic Updates to DNS [19] 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 sections 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 [6], and IPv6 Stateless Address Autoconfiguration [18] 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 link-only scope, indicated by having the prefix (FE80::0000/10), 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.

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 and any other configuration information assigned to the client. A binding is indexed by the tuple <DUID, IA-type, IAID> (where IA-type is the type of address in the IA; for example, temporary)
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 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.

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

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

reconfiguration nonce A 64 bit opaque value used to provide security for Reconfigure messages. A server generates a cryptographically strong random number as a nonce and

sends that nonce value to the client. The server then includes the nonce in any Reconfigure messages it sends to the client.

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 client or relay agent to communicate with servers, either because the client or 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 client or 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 Section 6. Message types not listed here are reserved for future use. The message code for each message type is shown with the message name.

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 from a server.
CONFIRM (4)	A client sends a Confirm message to any available server when it detects that it may have moved to a new link to request that the servers verify that the addresses and current configuration parameters assigned by the server to the client are still valid.
RENEW (5)	A client sends a Renew message to the server that originally provided the client's addresses and configuration parameters to extend the leases 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 leases 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 or Information-request message received from a client. A server sends a Reply message confirming or denying the validity of the client's addresses and configuration parameters in response to a Confirm message. 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 forward client messages to servers. The client message is encapsulated in an option in the Relay-forward message.

RELAY-REPL (13) A server sends a Relay-reply message to a relay agent to send messages to clients through the relay agent. The server encapsulates the client message as an option in the Relay-reply message, which the relay agent extracts and forwards 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. Configuration Parameters

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

Parameter	Default	Description
MIN_SOL_DELAY	1 sec	Min delay of first Solicit
MAX_SOL_DELAY	5 secs	Max delay of first Solicit
SOL_TIMEOUT	500 msecs	Initial Solicit timeout
SOL_MAX_RT	30 secs	Max Solicit timeout value

REQ_TIMEOUT	250	msecs	Initial Request timeout
REQ_MAX_RT	30	secs	Max Request timeout value
REQ_MAX_RC	10		Max Request retry attempts
CNF_TIMEOUT	250	msecs	Initial Confirm timeout
CNF_MAX_RT	1	sec	Max Confirm timeout
CNF_MAX_RD	10	secs	Max Confirm duration
REN_TIMEOUT	10	sec	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_TIMEOUT	500	ms	Initial Information-request timeout
INF_MAX_RT	30	secs	Max Information-request timeout value
REL_TIMEOUT	250	msecs	Initial Release timeout
REL_MAX_RT	1	sec	Max Release timeout
REL_MAX_RC	5		MAX Release attempts
DEC_TIMEOUT	250	msecs	Initial Decline timeout
DEC_MAX_RT	1	sec	Max Decline timeout
DEC_MAX_RC	5		Max Decline attempts
REC_TIMEOUT	1	sec	Initial Reconfigure timeout
REC_MAX_RC	8		Max Reconfigure attempts

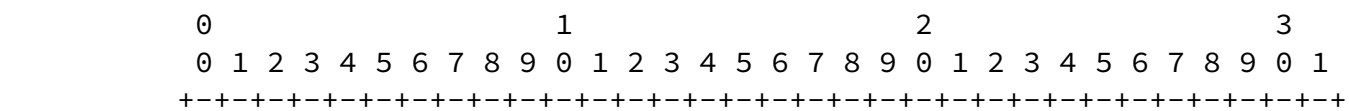
6. Message Formats

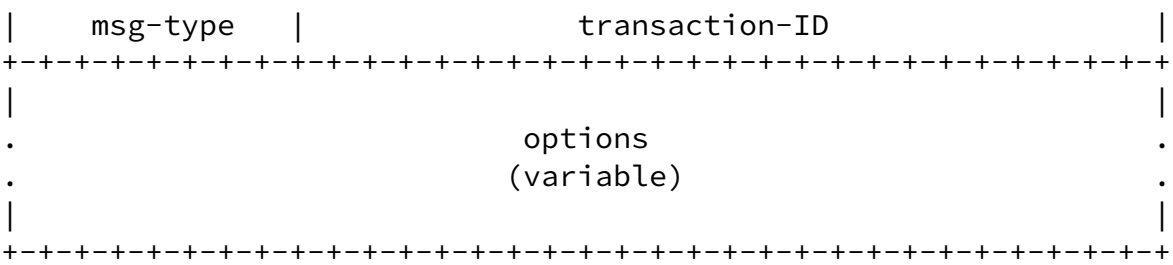
All DHCP messages sent between clients and servers share an identical fixed format header and a variable format area for options.

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:



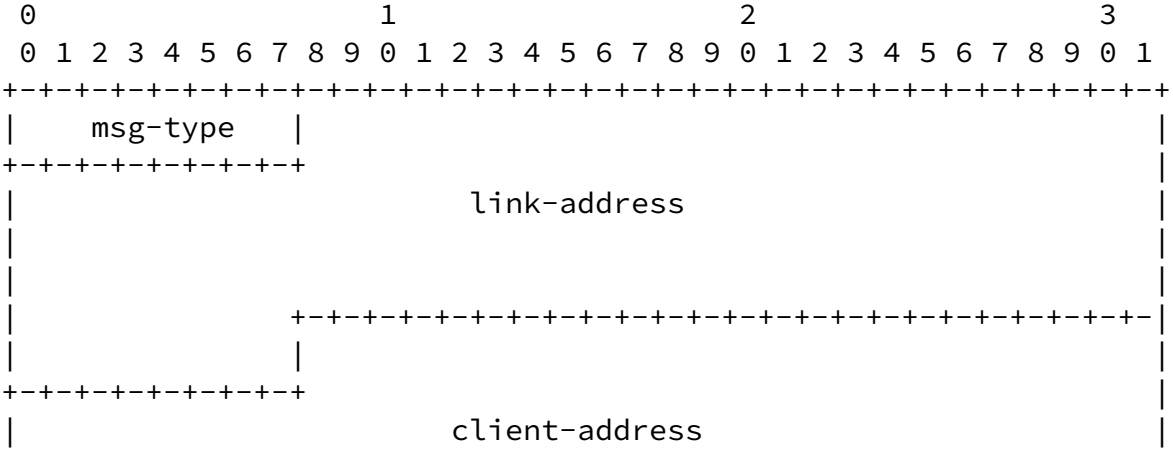


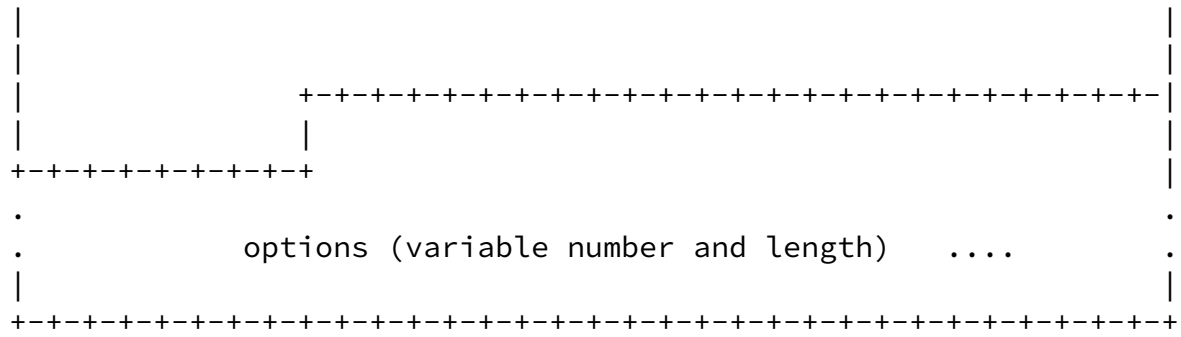
msg-type	Identifies the DHCP message type; the available message types are listed in section 5.3.
transaction-id	An unsigned integer used by a client or server to match a response message to a request message.
options	Options carried in this message; options are described in section 22.

7. Relay agent messages

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

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
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.
client-address	The address of the client from which the message to be forwarded was received
options	MUST include a "Client 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

link-address	The link-address copied from the Relay-forward message
client-address	The client's address, copied from the relay-forward message
options	MUST include a "Server message option"; see section 22.11; 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 RFC1035 [11]. A domain name or list of domain names in DHCP MUST NOT be stored in compressed form as described in section 4.1.4 of RFC1035.

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 consistent 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 256 octets long. The following types are currently defined:

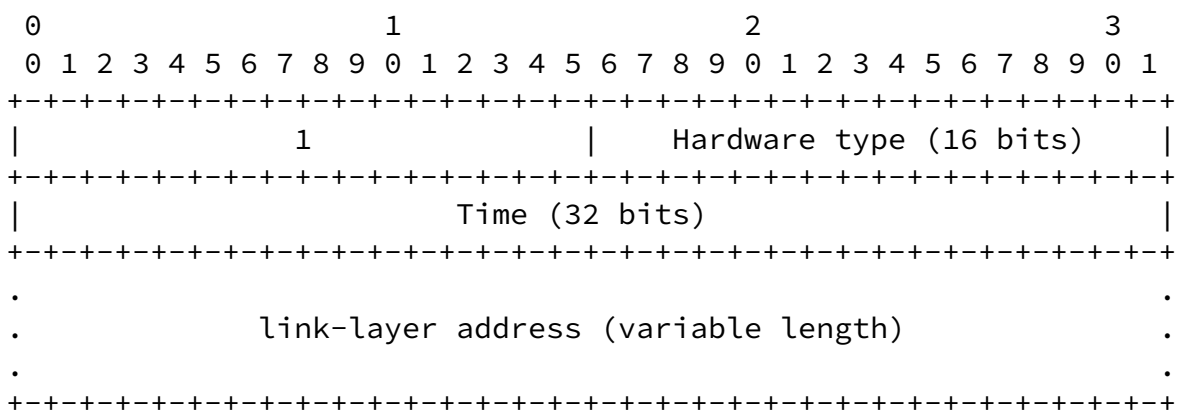
1	Link-layer address plus time
2	Vendor-assigned unique ID based on domain name
3	Vendor-assigned unique ID based on Enterprise Number
4	Link-layer address

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

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 the section on ARP in RFC 826. Both the time and the hardware type are stored in network byte order.

The following diagram illustrates the format of a DUID based on link-layer address plus time:



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 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.

Clients and servers using this type of DUID MUST store the DUID in stable storage, and MUST continue to use this DUID even if the network interface used to generate the DUID 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 DUIDs 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. A DUID collision is very unlikely even if the clocks haven't 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.

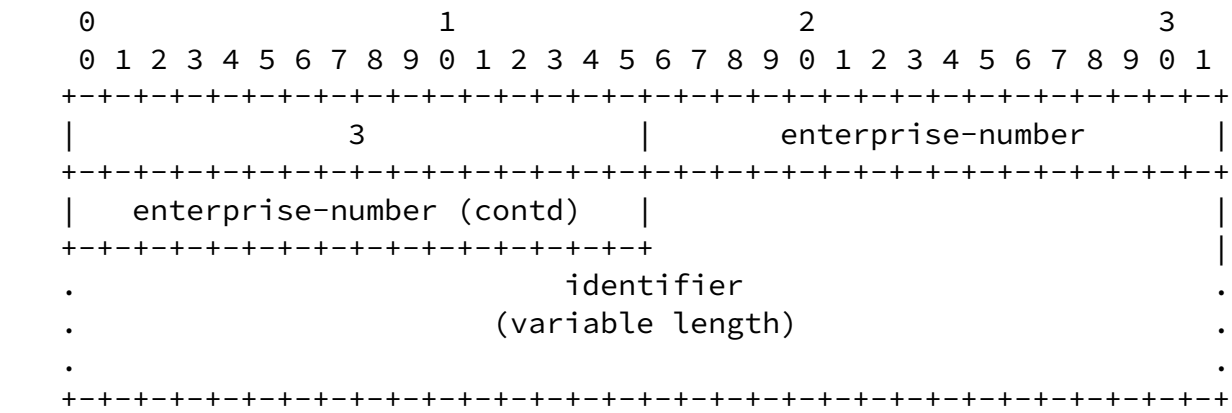
Despite our best efforts, it is possible that this algorithm for generating a DUID could result in a client identifier collision. A DHCP client that generates a DUID using this mechanism MUST provide an administrative interface that replaces the existing DUID with a newly-generated DUID of this type.

9.3. Vendor-assigned unique ID based on Enterprise Number (VUID-EN)

The vendor-assigned unique ID based on Enterprise Number consists of

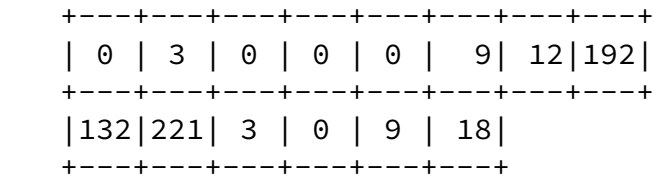
the vendor's registered Private Enterprise Number as maintained by IANA [7] followed by the value of the identifier.

The following diagram summarizes the structure of a VUID-EN:



The source of the identifier is left up to the vendor defining it, but each identifier part of each VUID-EN MUST be unique to the device that is using it, and MUST be assigned to the device at the time of manufacture and stored in some form of non-volatile storage. The VUID SHOULD be recorded in non-erasable storage. The enterprise-number is the vendor's registered Private Enterprise Number as maintained by IANA [7]. The enterprise-number is stored as an unsigned 32 bit number.

An example DUID of this type might look like this:

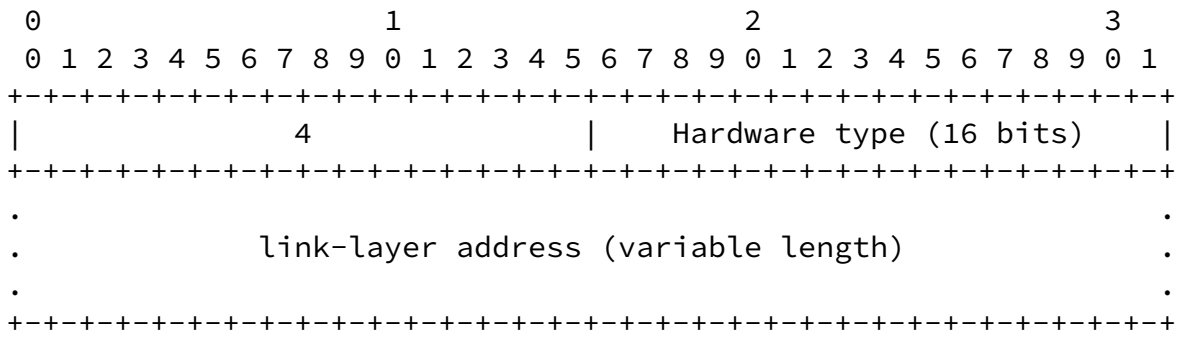


This example includes the two-octet type of 3, the Enterprise Number (9), followed by eight octets of identifier data.

9.4. Link-layer address

This type of DUID consists of two octets containing the DUID type 4, 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, this DUID could be used by a host that has a network interface implemented in a chip that is unlikely to be removed and used elsewhere. The hardware type MUST be a valid hardware type assigned by the IANA as described in the section on ARP in RFC 826. The hardware type is stored in network byte order.

The following diagram illustrates the format of a DUID based on link-layer address:



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 is being generated. The same DUID 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.

This type of DUID is recommended for devices that have a permanently-connected network interface with a link-layer address and do not have nonvolatile, writable stable storage. This type of DUID 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 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 and uses that IA 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 and other parameters. The parameters are specified as DHCP options within the IA, and are associated with the addresses in the IA and the interface to which the IA belongs. Other parameters that are not associated with a particular interface may be specified in the options section of a DHCP message, outside the scope of any IA.

Each address in an IA has a preferred lifetime and a valid lifetime, as defined in RFC2462 [18]. 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 RFC2462.

See section 22.4 for the representation of an IA in a DHCP message.

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 unicast 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.

A server MUST NOT assign an address that is otherwise reserved for some other purpose. These reserved addresses may be specified as 128-bit IPv6 addresses or as interface identifiers that are reserved for all subnets. For example, a server MUST NOT assign reserved anycast addresses, as defined in RFC2526, from any subnet.

12. Management of temporary addresses

A client may be assigned temporary addresses (temporary addresses are defined in RFC 3041 [13]). Clients and servers simply identify addresses as "temporary". 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.

Unless otherwise stated, an IA_TA option is used in the same way in as an IA option. In the protocol specification, unless otherwise stated, a reference to an IA should be read as either an IA or an IA_TA.

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

The server MAY update the DNS for a temporary address as described in section 4 of RFC3041.

13. Transmission of messages by a client

Unless otherwise specified, a client sends DHCP messages to the All_DHCP_Relay_Agents_and_Servers.

If the client is attached to a link that supports multicast transmission, the client sends DHCP messages to the All_DHCP_Relay_Agents_and_Servers address.

A client may send some messages directly to a server using unicast, as described in section 22.13.

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. 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 Information-request message must not include an IA option. Clients and server MAY choose to extract information from such a message if the information is of use to the recipient.

Message validation based on DHCP authentication is discussed in section 21.5.3.

15.1. Use of Transaction-ID field

The "transaction-ID" field holds a value used by clients and servers to synchronize server responses to client messages. A client SHOULD choose a different transaction-ID for each new message it sends. 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 identifier
- the message does not include a Client Identifier option

15.5. Confirm message

Clients MUST discard any received Confirm messages.

Servers MUST discard any Confirm messages received 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 do 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 do 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 do 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 messages that meet 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
- the message does not include a Client Identifier option and the

original message from the client contained a Client Identifier option

- the message includes a Client Identifier option and the contents of the Client Identifier option does not match the DUID of the client or the client did not include a Client Identifier option in the original message

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 fails any of the following conditions:

- the message MUST include a Server Identifier option
- the message MUST include one of the available security mechanisms:
  - the server sends a Reconfigure Nonce option whose value matches the current server nonce value known to the client
  - the server uses DHCP authentication: beginitemize
  - the message MUST contain an authentication option
  - the message MUST 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

includes a Server Identifier option and the DUID in the option does not match the server's DUID.

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 interface are attached to the same link. In addition, the client MUST use the IPv6 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 configuration information, including 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 configuration information to the IA respond to the client with an Advertise message. The client then initiates a configuration exchange as described in section 18.

Whenever a client initiates server solicitation with a Solicit message, it discards any reconfigure nonce values it may have previously recorded.

17.1. Client Behavior

A client uses the Solicit message to discover DHCP servers configured to serve addresses 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 MUST include one or more 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 options to request the assignment of non-temporary addresses and uses IA_TA options to request the assignment of

temporary addresses. Either IA or IA_TA options, or a combination of both can be included in DHCP messages.

The client MAY request specific options from the server by including an Option Request option (see section 22.7) as a hint about the

options the client is interested in receiving. If the client requires a consistent prioritization of the options it receives, it includes an Option Request option indicating the options it needs (see section 17.2.2). The client MAY include options with data values as hints to the server about parameter values the client would like to have returned.

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 22.15) in the Solicit message.

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 MIN_SOL_DELAY and MAX_SOL_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 RFC2462). 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 and is waiting for a Reply message, the client terminates the retransmission process as soon as a Reply message is received. 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 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 Solicit that does not include a Preference option is considered to have a preference value of 0. If the client receives an Advertise message with a preference value of 255, then the client SHOULD act immediately on that Advertise message without waiting for any additional Advertise messages.

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 choose to 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 AddrUnavail, 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. If the client receives a valid Reply message that includes a Rapid Commit option, it processes the message as described in section 18.1.6.

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.

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. If the server has been configured to respond to the client but has not been configured to respond with committed address assignments and other resources, the server responds with an Advertise message.

Otherwise, the server generates and sends an Advertise message to the client.

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 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 server will not assign any addresses to 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 the status code set to AddrUnavail and a status message for the user.

The server includes other options the server will return to the client in a subsequent Reply message. The server SHOULD limit the options returned to the client so that the DHCP message header and options do not cause fragmentation. The information in these options will 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 uses that option as a hint about the options the client has a preference for receiving from the server.

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 through the interface on 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 "server-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-Advertise message exchange, a server need not commit the assignment of configuration information to the client or otherwise keep state about the client before the server sends the Advertise message to the client. The client will choose one of the responding servers and send a Request message to obtain configuration information. The other servers can make any addresses they might have offered to the client available for assignment to other clients.

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 and the addresses from the other servers will be committed to the client but not 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.

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 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 (Rebind and Renew messages).

18.1. Client Behavior

A client will use Request, Confirm, Renew, Rebind and Information-request messages to acquire and confirm the validity of configuration information. The client uses the server identifier information and information about IAs from previous Advertise messages for use in constructing these messages.

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, with addresses and information about the IAs that were obtained from the server in a previous Advertise 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 MAY request specific options from the server by including an Option Request option (see section 22.7) as a hint about the options the client is interested in receiving. If the client requires a consistent prioritization of the options it receives, it includes an Option Request option indicating the options it needs (see section 18.2.1). The client MAY include options with data values as hints to the server about parameter values the client would like to have returned.

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 22.13) from the server, the client SHOULD unicast the Request message to the server.

DISCUSSION:

Use of multicast on a link and relay agents 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.

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 MAY choose one of the following actions:

- 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, its IPv6 addresses and other configuration information may no longer be valid. Examples of times when a client may have moved to a new link include:

- o The client reboots
- o The client is physically disconnected from a wired connection

- The client returns from sleep mode

- 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, along with the addresses associated with those IAs, in its Confirm message. Any responding servers will indicate the acceptability of the addresses 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 adds any appropriate options, including one or more IA options. The client MUST include the addresses the

client currently has associated with those IAs. The client fills in the T1 and T2 fields in the IA options and the preferred-lifetime and valid-lifetime fields in the IA Address options with preferred values or 0 if the client has no preference about those values.

The client MAY request specific options from the server by including an Option Request option (see section 22.7) as a hint about the options the client is interested in receiving. If the client requires a consistent prioritization of the options it receives, it includes an Option Request option indicating the options it needs (see section 18.2.2). The client MAY include options with data values as hints to the server about parameter values the client would like to have returned.

When the client sends the Confirm message, it MUST use an IPv6 address that the client has confirmed to be valid on the link to which it is currently attached and that is assigned to the interface for which the client is interested in obtaining configuration information as the source address in the IP header of the datagram carrying the Confirm message.

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 as described in section 14 terminates, 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 associated with addresses, the client sends a Renew message to the server containing an IA option for the IA and its associated addresses. 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.

If T1 or T2 is set to 0 by the server, the client does not send a Renew or Rebind message, respectively, for the 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.

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 MAY request specific options from the server by including an Option Request option (see section 22.7) as a hint about the options the client is interested in receiving. If the client requires a consistent prioritization of the options it receives, it includes an Option Request option indicating the options it needs (see section 18.2.3). The client MAY include options with data values as hints to the server about parameter values the client would like to have returned.

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 (see section 22.13) from the server, the client SHOULD unicast the Renew message to the server.

DISCUSSION:

> Use of multicast on a link and relay agents enables the inclusion of relay agent options in all messages sent by the client. The server MUST NOT enable the use of unicast for a client when relay agent options are required for that client.

The client transmits the message according to section 14, using the following parameters:

IRT REN_TIMEOUT

MRT REP_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 sends the Rebind message to the All_DHCP_Relay_Agents_and_Servers multicast address. 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 MAY request specific options from the server by including an Option Request option (see section 22.7) as a hint about the options the client is interested in receiving. If the client requires a consistent prioritization of the options it receives, it includes an Option Request option indicating the options it needs (see section 18.2.4). 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 mechanism in section 14 is modified as follows for use in the transmission of Rebind messages. The message exchange is terminated when the valid lifetimes of all of the addresses assigned to the IA expire (see section 10), at which time the client has several

alternative actions to choose from:

- 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 MAY request specific options from the server by including an Option Request option (see section 22.7) as a hint about the options the client is interested in receiving. If the client requires a consistent prioritization of the options it receives, it includes an Option Request option indicating the options it needs (see section 18.2.5). 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 INF_TIMEOUT

MRT INF_MAX_RT

MRC 0

MRD 0

18.1.6. Receipt of Reply message in response to a Request, Confirm, Renew, Rebind or Information-request message

Upon the receipt of a valid Reply message in response to a 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 [18] 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 18.1.9.

The client records the T1 and T2 times for each IA in the Reply message. The client records any addresses included with IAs in the Reply message. The client updates the preferred and valid lifetimes for the addresses in the IA from the lifetime information in the IA option. The client leaves any addresses that the client has associated with the IA that are not included in the IA option unchanged.

If the Reply was received in response to a Request, Renew or Rebind message, the client must update the information in any IA option contained in the Reply message. The client adds any new addresses from the IA option to the IA, updates lifetimes for existing addresses in the IA from the IA option and discards any addresses with a lifetime of zero in the IA option.

Management of the specific configuration information is detailed in the definition of each option, in section 22.

When the client receives a NotOnLink status in an IA from the server in response to a Confirm message, the client can assume it needs to

send a Request to the server to obtain appropriate addresses for the IA. 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 do indicate a NotOnLink status.

When the client receives a NoBinding status in an IA from the server for a Renew message the client can assume it needs to send a Request to reestablish an IA with the server.

When the client receives a NoBinding status in an IA from the server for a Rebind message the client can assume it needs to send a Request to reestablish an IA with the server or try another server.

When the client receives an AddrUnavail status in an IA from the server for a Rebind message the client can assume it needs to send a Request to reestablish an IA with the server or try another server.

18.1.7. 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 should not be in added to the IAs.

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

If the client has a source address of sufficient scope that can be used by the server as a return address and the client has received a Server Unicast option (section 22.13) from the server, the client SHOULD unicast the Release message to the server.

DISCUSSION:

> Use of multicast on a link and relay agents enables the inclusion of relay agent options in all messages sent by the client. The server MUST NOT enable the use of unicast for a client when relay agent options are required for that client.

The client SHOULD choose to guarantee the delivery of the Release message using the retransmission strategy in section 14. An example of a situation in which a client would not guarantee delivery would be when the client is powering down or restarting because of some error condition.

The client transmits the message according to section 14, using the following parameters:

IRT REL_TIMEOUT

MRT 0

MRC REL_MAX_MRC

MRD 0

The client MUST abandon the attempt to release addresses if the Release message exchange fails.

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, the addresses assigned to the IA will be reclaimed by the server when the lifetime of the address expires.

18.1.8. Receipt of Reply message in response to a Release message

Upon receipt of a valid Reply message, the client can consider the Release event successful.

18.1.9. Creation and transmission of Decline messages

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 in added to the IAs.

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

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 22.13) from the server, the client SHOULD unicast the Decline message to the server.

DISCUSSION:

> Use of multicast on a link and relay agents enables the inclusion of relay agent options in all messages sent by the client. The server MUST NOT enable the use of unicast for a client when relay agent options are required for that client.

The client transmits the message according to section 14, using the following parameters:

IRT DEC_TIMEOUT

MRT DEC_MAX_RT

MRC DEC_MAX_RC

MRD 0

The client MUST abandon the attempt to decline addresses if the Decline message exchange fails.

If addresses are released 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.

18.1.10. Receipt of Reply message in response to a Decline message

Upon receipt of a valid Reply message, the client can consider the Decline event successful.

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.

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 value UseMulticast and no other options.

When the server receives a Request the client is requesting the configuration of IAs by the server. The server creates the bindings for that client according to the server's policy and configuration information and records the IAs and other information about the client.

The server constructs a Reply message by setting the "msg-type" field to REPLY, 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 a valid prefix for the link to which the client is connected, the server MUST return the IA to the client with the status field set to NotOnLink.

If the server cannot assign any addresses to any of the IAs in the message from the client, the server MUST include the IAs in the Reply message with the status field set to AddrUnavail and no addresses in the IA.

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.

If the server will use a reconfigure nonce value for security of Reconfigure messages, the server generates a new nonce value for the client, records the value and includes it in a Reconfigure Nonce option (see section 22.21) in the Reply message.

The server includes other options containing configuration information to be returned to the client. The server SHOULD limit the options returned to the client so that the DHCP message header and options do not cause fragmentation. If the client has included an Option Request option in the Solicit message, the server uses that option as a hint about the options the client has a preference for receiving from the server.

18.2.2. Receipt of Confirm messages

When the server receives a Confirm message, the client is requesting confirmation that the IP addresses it will use is valid and requesting the most current configuration information for the client. The server compares the addresses in the IA Address options in the

Confirm message from the client with the addresses in the binding for the client.

If the server finds that the addresses in the Confirm message do not match what is in the binding for that client or the addresses in the Confirm message are not appropriate for the link from which the client sent the Confirm message, the server sends a Reply message containing a Status Code option with the value ConfNoMatch.

If the server finds that the addresses in the Confirm message match the addresses in the binding for that client, and the configuration information is still valid, the server sends a Reply message containing a Status Code option with the value Success.

If the server cannot determine if the information in the Confirm message is valid or invalid, the server MUST NOT send a reply to the client. For example, if the server does not have a binding for the client, but the configuration information in the Confirm message appears valid, the server does not reply.

The server constructs a Reply message by setting the "msg-type" field to REPLY, 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 IA options for each of the IA options in the Confirm message. The server chooses values for T1, T2 and lifetimes for each of the addresses in the IAs according to the server's configured policies. The values for T1, T2 and the lifetimes sent by the client are the client's preferences for those values. The server also includes options for any other configuration information to be sent to the client.

The server includes other options containing configuration information to be returned to the client. The server SHOULD limit the options returned to the client so that the DHCP message header and options do not cause fragmentation. If the client has included an Option Request option in the Renew message, the server uses that

option as a hint about the options the client has a preference for receiving from the server.

The Reply message from the server MUST include a Status Code option.

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 value UseMulticast and no other options.

When the server receives a Renew and 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 status set to NoBinding in the Renew message.

If the server finds that any of the addresses are no longer valid for the client, 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, and includes a Status Code option with value Success. 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, 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.

18.2.4. Receipt of Rebind messages

When the server receives a Rebind and 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 status set to NoBinding in the Rebind message.

If the server finds that the any of the addresses are no longer valid for the client, 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, 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. The server SHOULD limit the options returned to the client so that the DHCP message header and options do not cause fragmentation. If the client has included an Option Request option in the Rebind message, the server uses that option as a hint about the options the client has a preference for receiving from the server.

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, 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. The server SHOULD limit the options returned to the client so that the DHCP message header and options do not cause fragmentation. If the client has included an Option Request option in the Information-request message, the server uses that option as a hint about the options the client has a preference for receiving from the server.

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

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 value UseMulticast 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 IA, the server deletes the addresses from the IAs. 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. The server ignores addresses not assigned to the IA (though it may choose to log an error if it finds such an address).

After all the address 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.

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 "server-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.

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 places its identifier in a Server Identifier option.

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.

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 one of the following two security mechanisms:

- The server includes a Reconfigure Nonce option containing the reconfiugre nonce value currently assigned to the client

- The server includes an authentication option in the Reconfigure message

The server MUST include a Reconfigure Message option (defined in section 22.20) 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. The server may obtain the address of the client through the information that the server has about clients that have been in contact with the server, or the server may be configured with the address of the client 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 RECREP_TIMEOUT value and waits again. The server continues this process until REC_MAX_R 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_RT are documented in section 5.5.

19.1.3. Receipt of Renew messages

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

The server MAY choose to send a Reply with the IAs and other parameters to be reconfigured, even if those IAs and parameters were not requested in the Renew message from the client.

19.2. Receipt of Information-request messages

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

The server MAY choose to send a Reply with the other parameters to be reconfigured, even if those parameters were not specified in the Information-request message from the client.

19.3. Client Behavior

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

19.3.1. Receipt of Reconfigure messages

Upon receipt of a valid Reconfigure message, the client initiates a transaction with the server by sending a Reply or Information-request message. While the transaction is in progress, the client silently discards any Reconfigure messages it receives.

The client responds with either a Renew message or an Information-request message as indicated by the Reconfigure Message option (as defined in section 22.20).

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 (regardless of the transaction ID in the Reconfigure message) until the exchange is complete. Subsequent Reconfigure messages (again independent of the transaction ID) 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.

19.3.2. 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: if the server included on Option Request option specifying the IA option, the client MUST include IA options containing the addresses the client currently has assigned to ALL IAs for the interface through which the Reconfigure message was received.

19.3.3. 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.

19.3.4. 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.

19.3.5. Receipt of Reply messages

Upon the receipt of a valid Reply message, the client extracts the contents of the "options" field, 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

For this discussion, the relay agent MAY be configured to use a list of server destination addresses, which MAY include unicast addresses, the All_DHCP_Servers multicast address, or other multicast 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.

20.1. Relaying of client messages

When a relay agent receives a valid client message, it constructs a Relay-forward message. 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.

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 forwarded, the relay agent MUST include an Interface-id option (see section 22.19) 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.

The relay agent copies the source address from the IP datagram in which the message was received from the client into the client-address field in the Relay-forward message.

The relay agent constructs a Client Message option (see section 22.10) that contains the entire DHCP message (excluding the IP and UDP headers) from the client in the data field of the option. The relay agent places the Client Message option along with any "relay agent-specific" options in the options field of the Relay-forward message. The relay agent sends the Relay-forward message to all the servers in the list of server destination addresses with which it has been configured or to the All_DHCP_Servers address if it has not been explicitly configured with server destination addresses.

20.2. Relaying of server messages

The relay agent processes any other options included in the Relay-reply message as appropriate to those options. The relay

agents then discards those options.

If the Relay-reply message includes a Interface-id option, the relay agent forwards the message from the server to the client on the link identified by the Interface-id option. Otherwise, the relay agent forwards the message on the link identified by the link-address field.

In either case, the relay agent extracts the server message from the Server Message option (see section 22.11) and forwards the message to the address in the client-address field in the Relay-reply message.

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 [5].

21.1. DHCP threat model

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.

The attack specific to a DHCP client is the possibility of the establishment of a "rogue" server with the intent of providing incorrect configuration information to the client. The motivation

for doing so may be to establish a "man in the middle" attack or it may be for a "denial of service" attack.

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

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

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.

21.2. Security of messages sent between servers and relay agents

Relay agents and servers that choose to exchange messages securely use the IPsec mechanisms for IPv6 [8]. The way in which IPsec is employed by relay agents and servers is not specified in this document.

21.3. Summary of DHCP authentication

Authentication of DHCP messages is accomplished through the use of the Authentication option (see section 22.12). The authentication information carried in the Authentication option can be used to reliably identify the source of a DHCP message and to confirm that the contents of the DHCP message have not been tampered with.

The Authentication option provides a framework for multiple authentication protocols. One such protocol is defined here.

Other protocols defined in the future will be specified in separate documents.

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.

21.4. 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 [10]) can reduce the danger of replay attacks. This method MUST be supported by all protocols.

21.5. Delayed authentication protocol

If the protocol field is 1, 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 [9] using the MD5 hash [17] is defined here.

21.5.1. Management issues in the delayed authentication protocol

The "delayed authentication" protocol does not attempt to address situations where a client may roam from one administrative domain to another, i.e. interdomain roaming. This protocol is focused on solving the intradomain problem where the out-of-band exchange of a shared key is feasible.

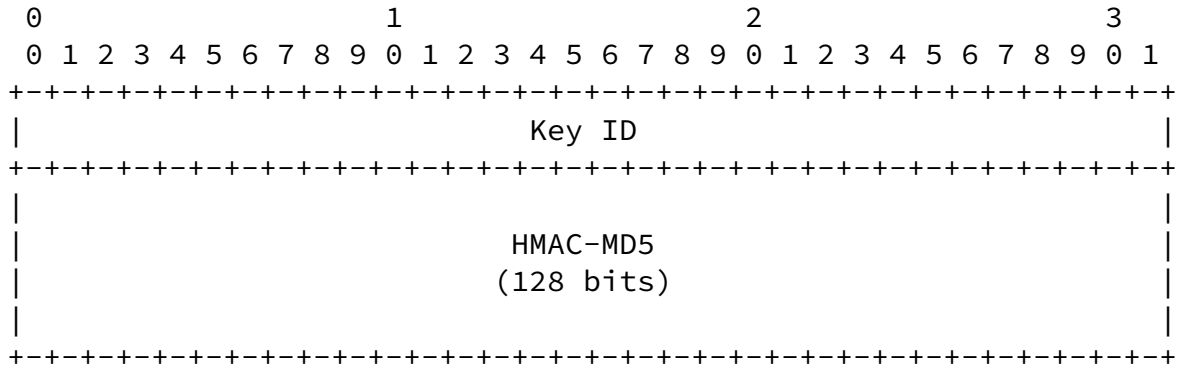
21.5.2. Use of the Authentication option in the delayed authentication protocol

In a Solicit message, the Authentication option carries the Protocol, Algorithm and fields, but no Replay detection or Authentication information.

In an Advertise, Request, Renew, Rebind, Confirm, Decline, Release or Information-request message, the Authentication option carries the Protocol, Algorithm, RDM and Replay detection fields and Authentication information.

A DHCP message MUST NOT contain more than one Authentication option when using the delayed authentication protocol.

The format of the Authentication information is:



The following definitions will be used in the description of the authentication information for delayed authentication, algorithm 1:

Replay Detection - as defined by the RDM field
K - a key (secret value) shared between the source and destination of the message; each key has a unique identifier (key ID)

key ID - the unique identifier for the key value
used to generate the MAC for this message
HMAC-MD5 - the MAC generating function.

The sender computes the MAC using the HMAC generation algorithm [9] and the MD5 hash function [17]. The entire DHCP message (setting the MAC field to zero), including the DHCP message header and the options field, is used as input to the HMAC-MD5 computation function. The 'key ID' field MUST be set to the identifier of the key used to generate the MAC.

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.

Delayed authentication requires a shared secret key for each client on each DHCP server with which that client may wish to use the DHCP protocol. Each key has a unique identifier that can be used by a receiver to determine which key was used to generate the MAC in the DHCP message. Therefore, delayed authentication may not scale well in an architecture in which a DHCP client connects to multiple administrative domains.

21.5.3. Message validation

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 [9]. The entire DHCP message (except the MAC field of the authentication option itself), 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.

21.5.4. Key utilization

Each DHCP client has a key, K. The server uses the client's DUID to identify the client's key. The client uses its key to encode any messages it sends to the server and to authenticate and verify any messages it receives from the server. The client's key is initially distributed to the client through some out-of-band mechanism, and is stored locally on the client for use in all authenticated DHCP messages. Once the client has been given its key, it uses that key for all transactions even if the client's configuration changes; for example, if the client is assigned a new network address.

Each DHCP server knows, or be able to obtain in a secure manner, the keys for all authorized clients. If all clients use the same key, clients can perform both entity and message authentication for all messages received from servers. However, the sharing of keys is strongly discouraged as it allows for unauthorized clients to masquerade as authorized clients by obtaining a copy of the shared key and allows for trivial spoofing of an authenticated DHCP server. To authenticate the identity of individual clients, each client must be configured with a unique key and a key ID for that key.

21.5.5. Client considerations for delayed authentication protocol

21.5.5.1. 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 21.5. The client does not include any replay detection or authentication information in the Authentication option.

21.5.5.2. 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 21.5.3.

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 a 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.

21.5.5.3. 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 21.5. When the client sends a subsequent message, it MUST use the same key used by the server to generate the authentication information.

21.5.5.4. Sending Information-request messages

If the server has selected a key for the client in a previous message exchange (see section 21.5.6.1, the client MUST use the same key to generate the authentication information. If the client has not previously been given a key with the server, the client MUST use a key that has been selected for the client through some external mechanism.

21.5.5.5. 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 validation and MAY log the validation failure. If the Reply fails to pass validation, 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.

21.5.5.6. Receiving Reconfigure messages

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

21.5.6. Server considerations for delayed authentication protocol

21.5.6.1. 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 21.5. 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.

21.5.6.2. 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 21.5.3. If the message fails to pass validation 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 procedure, 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 21.5.

21.5.6.3. Sending Reconfigure messages

The server MUST include an Authentication option in a Reconfigure message, generated as specified in section 21.5 using the key the server initially selected for the client to which the Reconfigure message is to be sent.

If the server has not previously selected a key for the client, the server MUST use a key that has been selected for the client through some external mechanism.

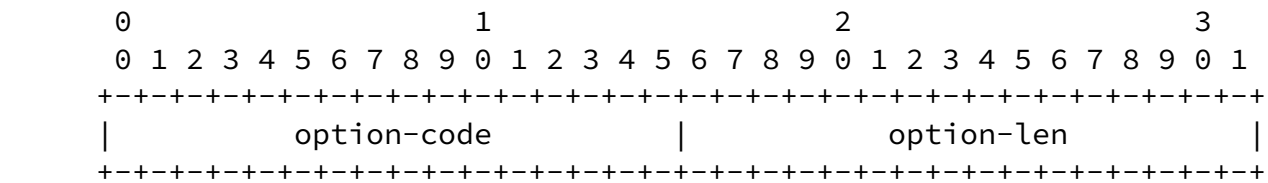
22. DHCP options

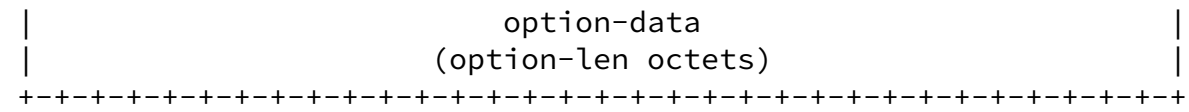
Options are used to carry additional information and parameters in DHCP messages. Every option shares a common base format, as described in section 22.1. All values in options are represented in network byte order.

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





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

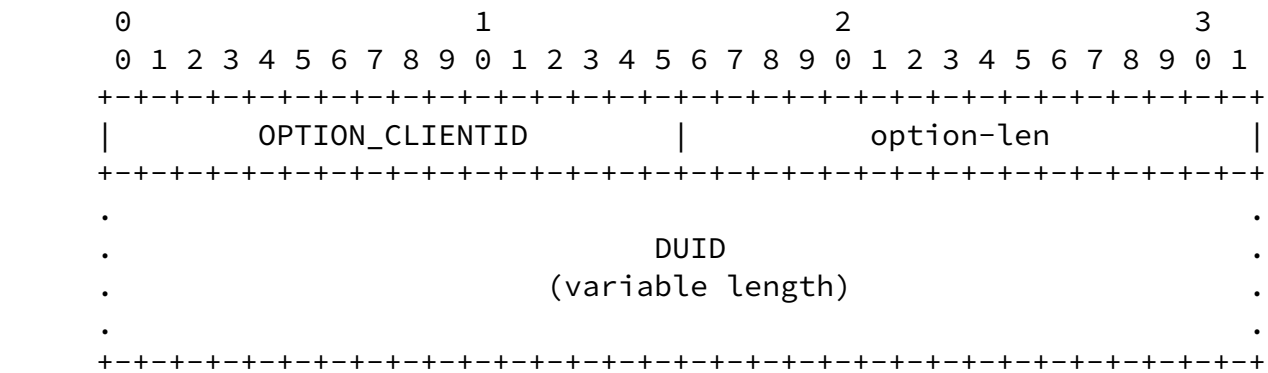
option-len An unsigned integer giving the length of the option-data field in this option in octets.

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



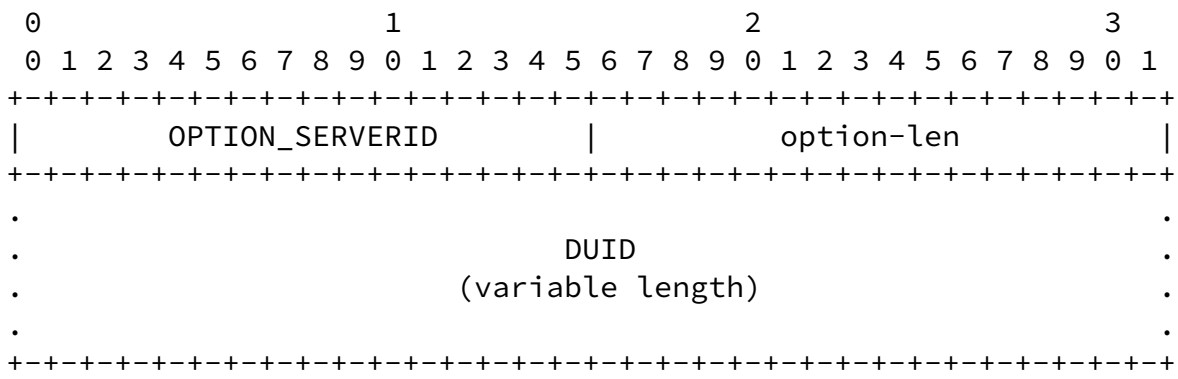
option-code OPTION_CLIENTID (1)

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



option-code OPTION_SERVERID (2)

option-len Length of DUID in octets

DUID The DUID for the server

A server MUST process any message it receives that contains a Server Identifier option with a DUID that matches the server's DUID.

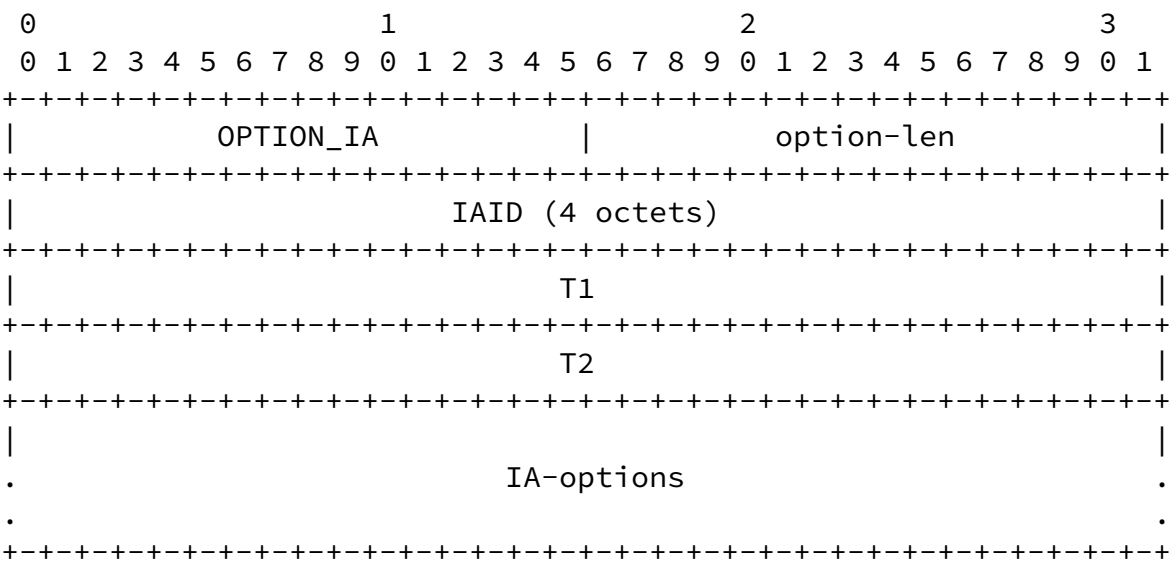
22.4. Identity Association option

The Identity Association option (IA option) is used to carry an identity association, the parameters associated with the IA and the addresses associated with the IA.

Addresses appearing in an IA option are not temporary addresses (see

section 22.5).

The format of the IA option is:



option-code	OPTION_IA (3)
option-len	12 + length of IA-options field
IAID	The unique identifier for this IA; the IAID must be unique among the identifiers for all of this client's IAs. The number space for IA IAIDs is separate from the number space for IA_TA IAIDs.
T1	The time at which the client contacts the server from which the addresses in the IA were obtained to extend the lifetimes of the addresses assigned to the IA; 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; T2 is a time duration relative to the current time expressed in units of seconds
IA-options	Options associated with this IA.

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

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

The status of any operations involving this IA is indicated in a Status Code option in the IA-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 have expired, the IA 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.

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 may send 0 if it has no preference for T1 and T2. 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. 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 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, respectively. If the server does not intend for a client to extend the lifetimes of the addresses in an IA, the server sets T1 and T2 to 0.

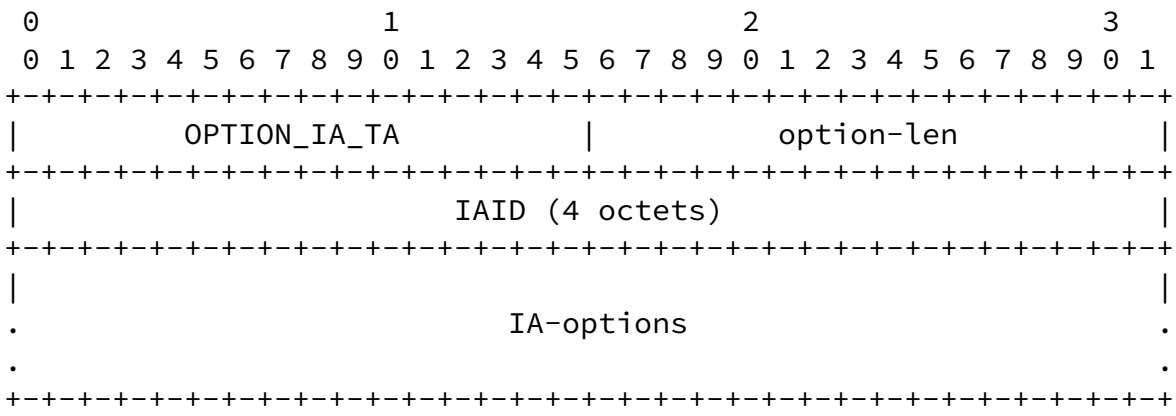
T1 is the time at which the client begins the lifetime extension process by sending a Renew message to the server that originally assigned the addresses to the IA. T2 is the time at which the client starts sending a Rebind message to any server.

T1 and T2 are specified as unsigned integers that specify the time in seconds relative to the time at which the messages containing the option is received.

22.5. Identity Association for Temporary Addresses option

The Identity Association for Temporary Addresses (IA_TA) option is used to carry an IA, the parameters associated with the IA and the addresses associated with the IA. All of the addresses in this option are used by the client as temporary addresses, as defined in RFC 3041.

The format of the IA_TA option is:



option-code OPTION_IA_TA (4)

option-len 4 + length of IA-options field

IAID The unique identifier for this IA; the IAID must be unique among the identifiers for all of this client's IAs. The number space for IA_TA IAIDs is separate from the number space for IA IAIDs.

IA-options Options associated with this IA.

The IA-Options field encapsulates those options that are specific to this IA. For example, all of the IA Address Options carrying the addresses associated with this IA are in the IA-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 is indicated in a Status Code option in the IA-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 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.

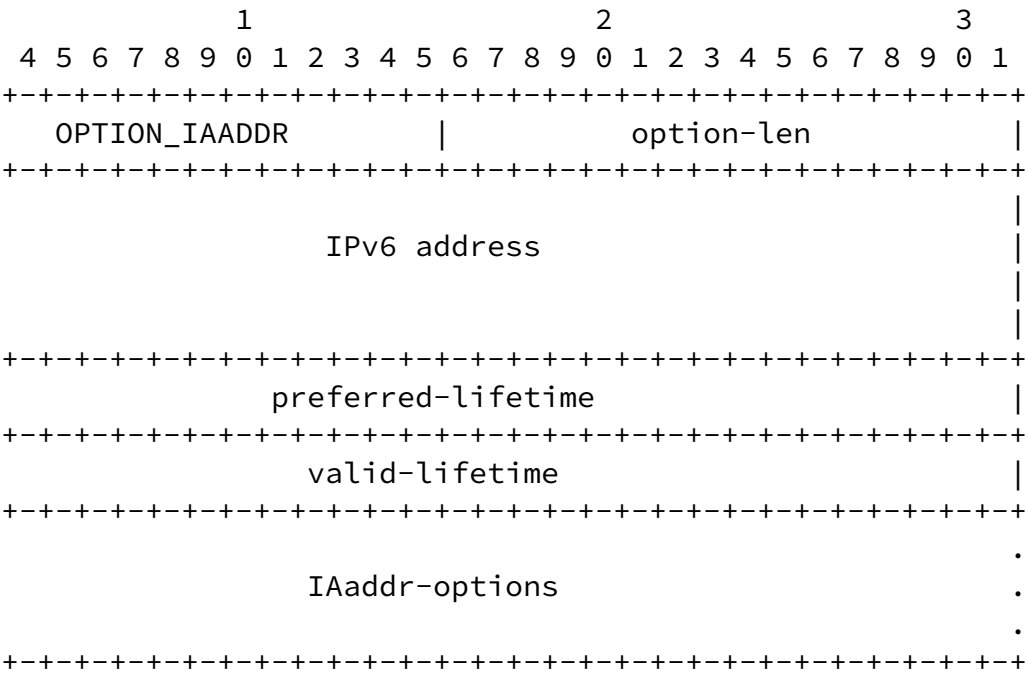
A server MUST return the same set of temporary address for the same IA_TA (as identified by the IAID) as long as those addresses are still valid. After the lifetimes of the addresses in an IA_TA have expired, the IAID may be reused to identify a new IA_TA with new temporary addresses.

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. The IA Address option must be encapsulated in the Options field of an Identity Association option. The Options field encapsulates those options that are specific to this address.

The format of the IA Address option is:



option-code OPTION_IADDR (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.

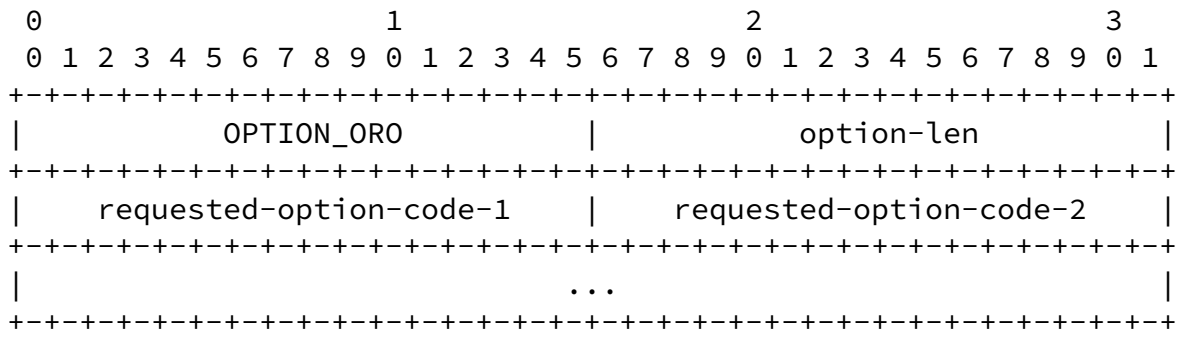
An IA Address option may appear only in an IA option or an IA_TA option. More than one IA Address Options can appear in an IA option or an IA_TA option.

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:



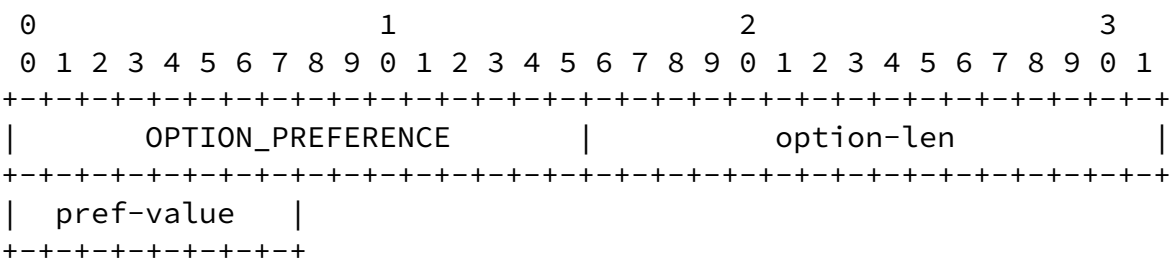
option-code OPTION_ORO (6)

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



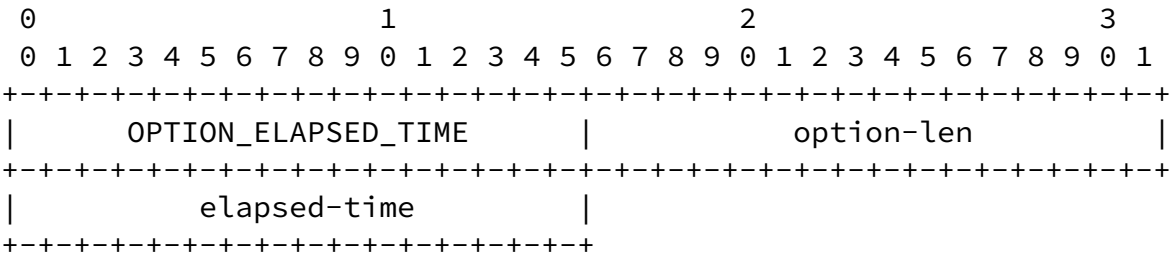
option-code OPTION_PREFERENCE (7)

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



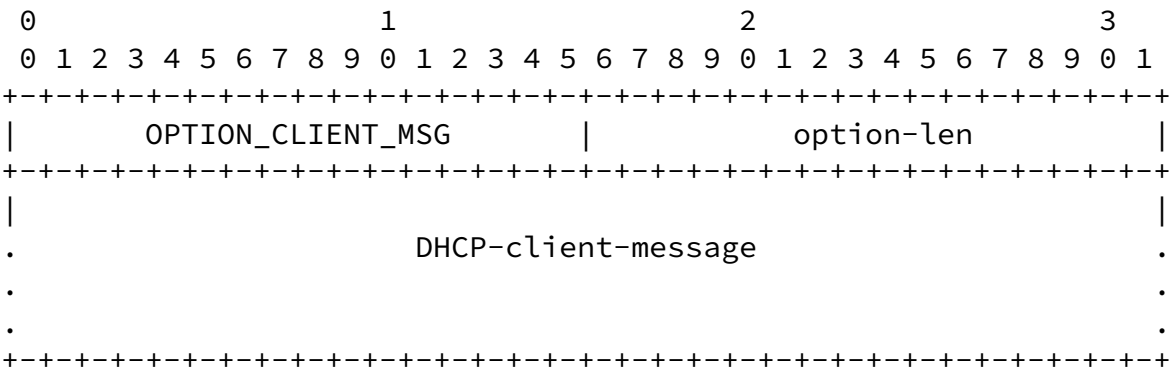
option-code OPTION_ELAPSED_TIME (8)

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 transaction. 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 hasn't answered in a reasonable time.

22.10. Client message option



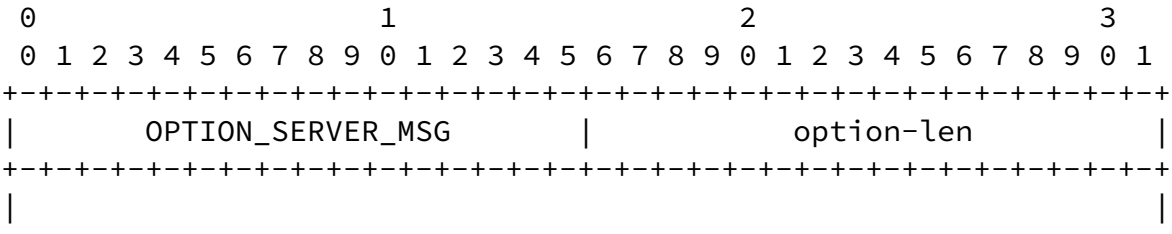
option-code OPTION_CLIENT_MSG (9)

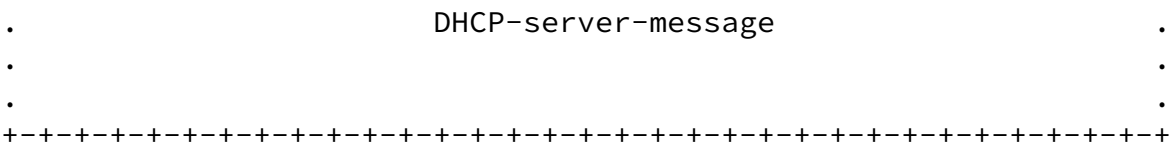
option-len Length of DHCP client message.

DHCP-client-message The message received from the client; forwarded verbatim to the server.

A relay agent forwards a message from a client to a server as the contents of a Client Message option in a Relay-forward message.

22.11. Server message option





option-code OPTION_SERVER_MSG (10)

option-len Length of DHCP server message.

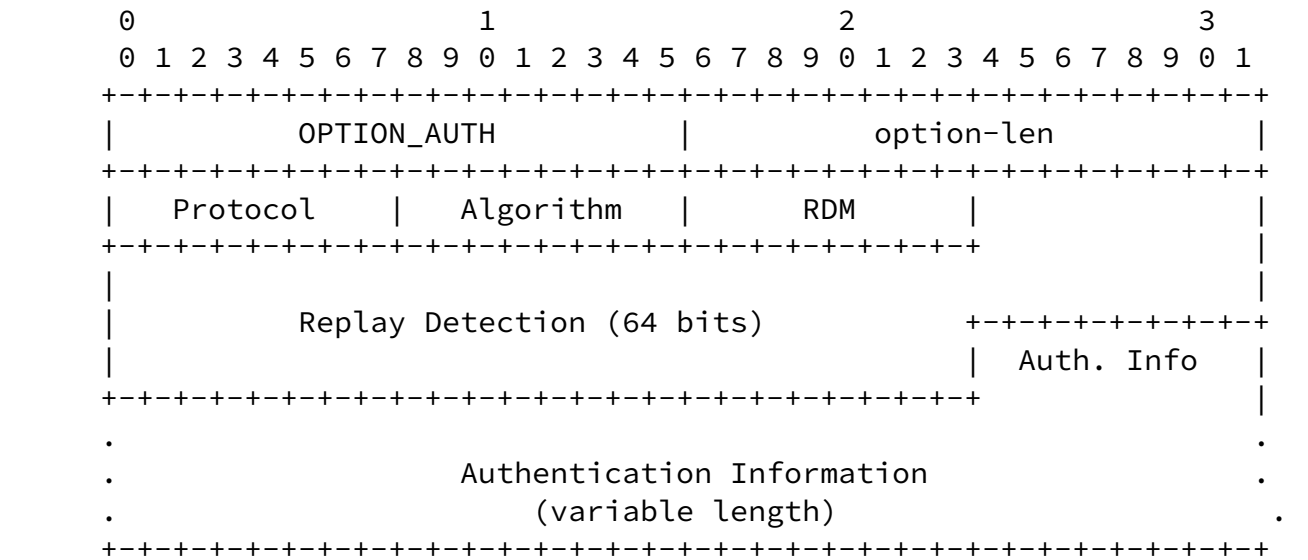
DHCP-server-message The message received from the server; forwarded verbatim to the client.

A server sends a DHCP message to be forwarded to a client by a relay agent as the contents of a Server Message option in a Relay-reply message.

22.12. 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:



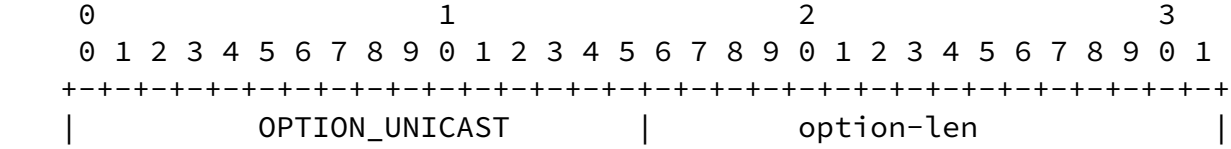
option-code	OPTION_AUTH (11)
option-len	15 + length of Authentication Information field
protocol	The authentication protocol used in this authentication option
algorithm	The algorithm used in the authentication protocol
RDM	The replay detection method used in this authentication option
Replay detection	The replay detection information for the RDM

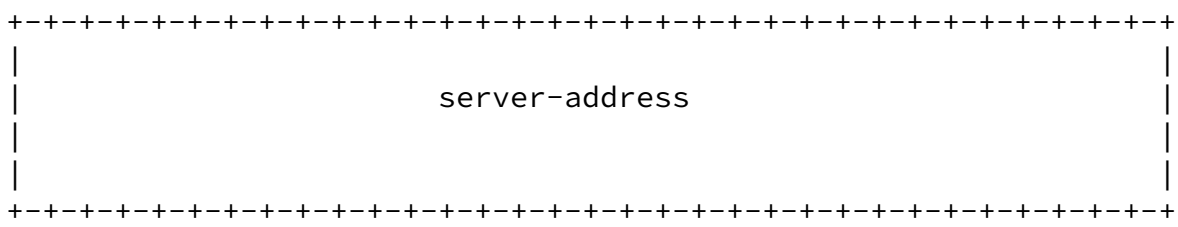
Authentication information	The authentication information, as specified by the protocol and algorithm used in this authentication option

22.13. 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 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.

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





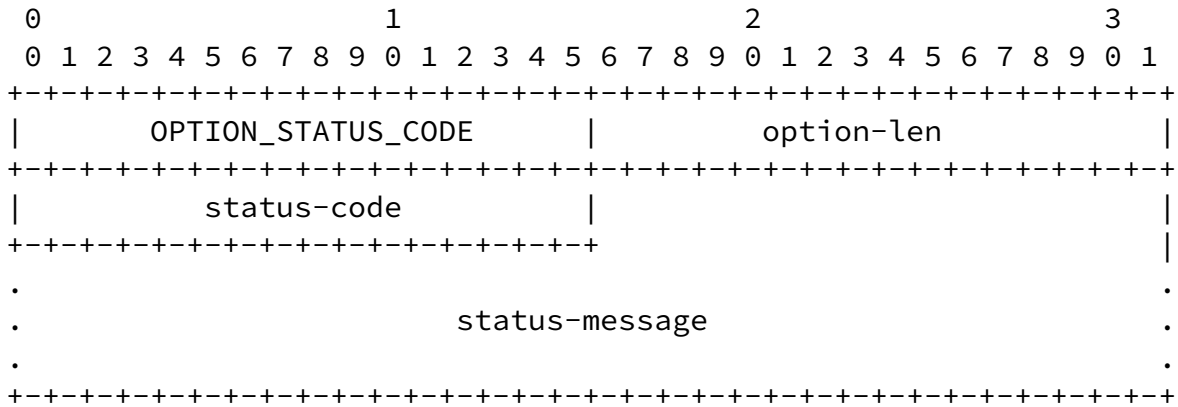
option-code OPTION_UNICAST (12)

option-len 16

server-address The IP address to which the client should send messages delivered using unicast

22.14. Status Code Option

This option returns a status indication related to the DHCP message or option in which it appears.



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.

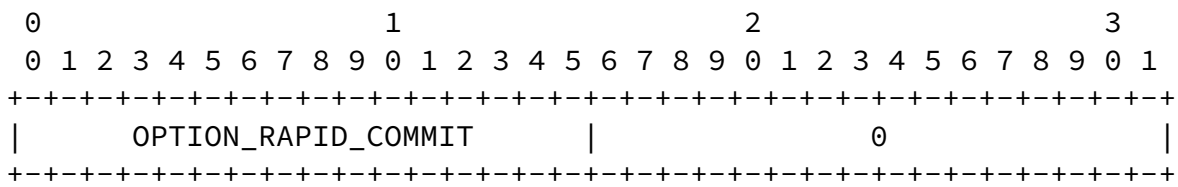
status-message A UTF-8 encoded text string, which MUST NOT be null-terminated.

A Status Code option may appear in a DHCP message option, or in the options area 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.

22.15. Rapid Commit option

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.



option-code OPTION_RAPID_COMMIT (14)

option-len 0

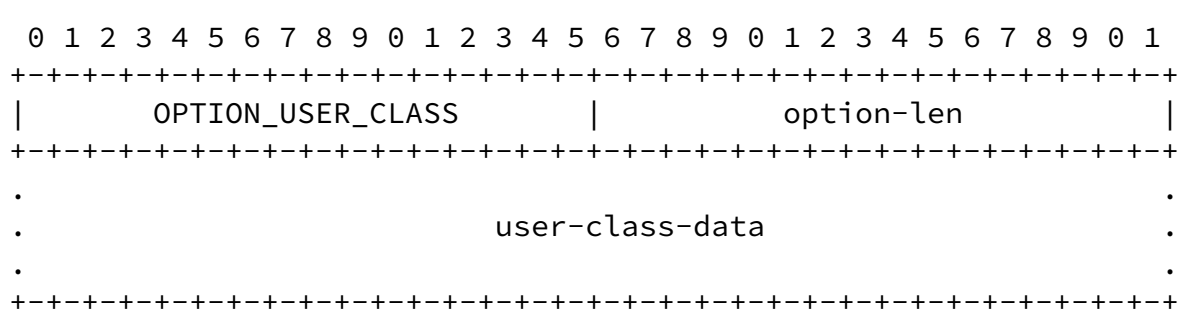
DISCUSSION:

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.

22.16. User Class Option

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

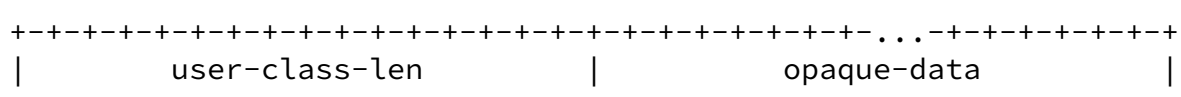


option-code OPTION_USER_CLASS (15)

option-len Length of user class data field

user-class-data The user classes carried by the client.

The 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:

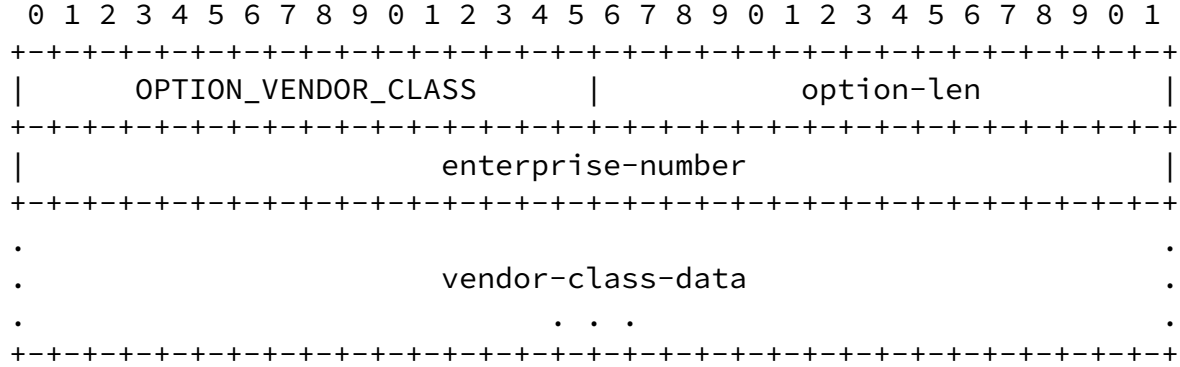


The user-class-len is two octets long and specifies the length of the opaque user class data in network byte order.

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.

22.17. 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.



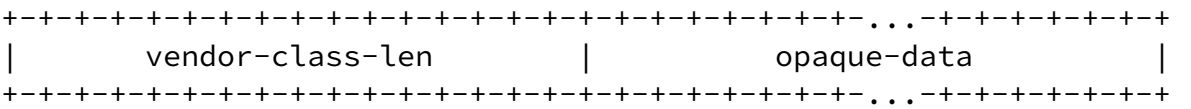
option-code OPTION_VENDOR_CLASS (16)

option-len 4 + length of vendor class data field

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

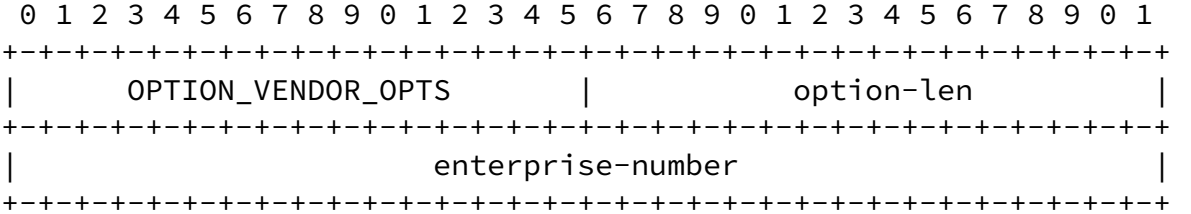
Each instance of the vendor-class-data is formatted as follows:

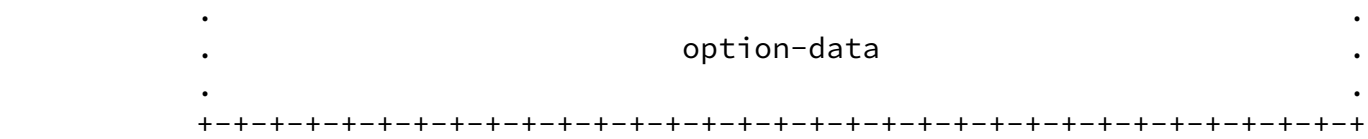


The vendor-class-len is two octets long and specifies the length of the opaque vendor class data in network byte order.

22.18. Vendor-specific Information option

This option is used by clients and servers to exchange vendor-specific information. The definition of this information is vendor specific. The vendor is indicated in the enterprise-number field. Clients that do not receive desired vendor-specific information SHOULD make an attempt to operate without it, although they may do so (and announce they are doing so) in a degraded mode.





option-code OPTION_VENDOR_OPTS (17)

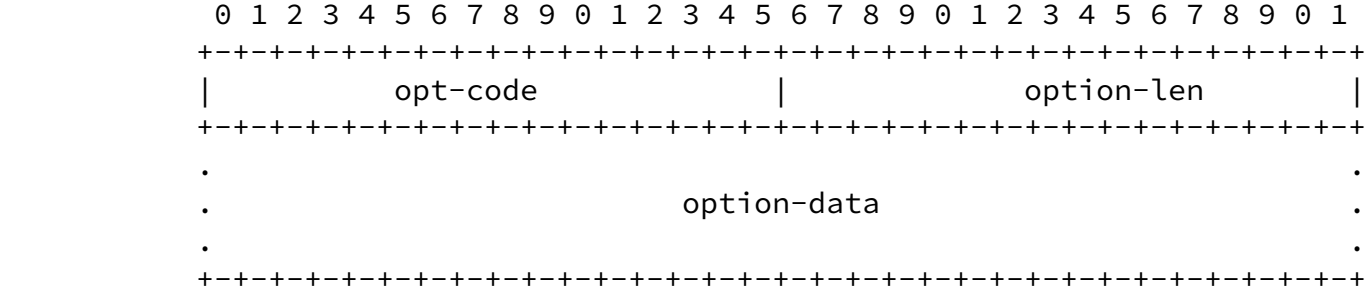
option-len 4 + length of option-data field

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

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

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

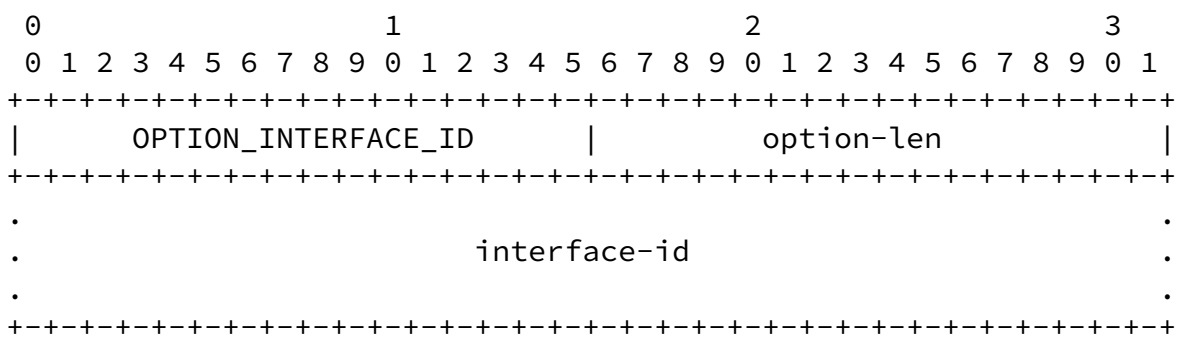
option-len An unsigned integer giving the length of the option-data field in this encapsulated option in octets.

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. A DHCP message MUST NOT contain more than one Vendor-specific Information option with the same Enterprise Number.

22.19. 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 forwards the message to the client through the interface identified by the option.



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

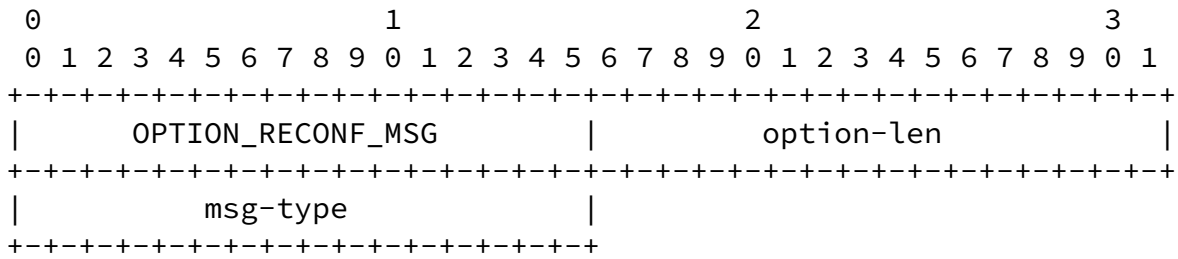
relay agent's interfaces

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 string 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.

22.20. 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.

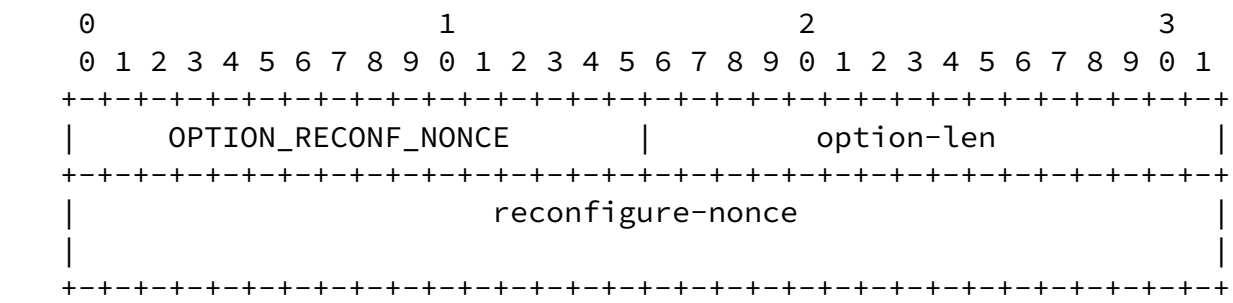


option-code	OPTION_RECONF_MSG (19)
option-len	1
msg-type	1 for Renew message, 2 for Information-request message

22.21. Reconfigure Nonce option

If a server uses a reconfigure nonce to provide security for Reconfigure messages, the server maintains a nonce value for each client. It initially informs the client of the nonce value and then includes the nonce value in any Reconfigure message sent to the client.

The following figure gives the format of the Reconfigure Nonce option:



option-code	OPTION_RECONF_NONCE (20)	
option-len	8	
reconfigure-nonce	reconfigure nonce value sent to the client	

The Reconfigure Nonce option MUST NOT appear in any DHCP message other than Reply or Reconfigure.

23. Security Considerations

Section 21 describes a threat model and an option that provides an authentication framework to defend against that threat model.

24. IANA Considerations

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

- Multicast addresses
- Message types
- Option codes
- Status codes
- DUID

- Authentication
    - * Protocol
    - * Algorithm
    - * RDM

IANA is requested to manage a registry of values for each of these name spaces, which are described in the remainder of this section.

These name spaces are all to be managed separately from the name spaces defined for DHCPv4 [4, 1].

New values in each of these name spaces except for DHCP option codes should be approved by the process of IETF consensus [12]. New DHCP option codes should be approved through expert review by a designated expert [12].

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 is requested to record the following message types (defined in section 5.3). IANA is requested to maintain a 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 is requested to record the following option-codes (as defined in section 22). IANA is requested to maintain a registry of DHCP option codes.

OPTION_CLIENTID	1
OPTION_SERVERID	2
OPTION_IA	3
OPTION_IA_TMP	4
OPTION_IADDR	5
OPTION_ORO	6

OPTION_PREFERENCE	7
OPTION_ELAPSED_TIME	8
OPTION_CLIENT_MSG	9
OPTION_SERVER_MSG	10
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_NONCE	20

24.4. Status codes

IANA is requested to record the status codes defined in the following table. IANA is requested to 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
AuthFailed	2	Authentication failed or nonexistent
AddrUnavail	3	Addresses unavailable
NoBinding	4	Client record (binding) unavailable
ConfNoMatch	5	Client record Confirm doesn't match IA
NotOnLink	6	One or more prefixes of the addresses in the IA is not valid for the link from which the client message was received
UseMulticast	7	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 is requested to record the following DUID types (as defined in section 9.1). IANA is requested to manage definition of additional DUID types in the future.

Link-layer address plus time	1
VUID-EN	2
Link-layer address	3

24.6. Authentication option

Section 21 references three name spaces associated with the Authentication Option (section 22.12), which are defined in the authentication mechanism for DHCPv4 [5].

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.

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) Bill Arbaugh, Thirumalesh Bhat, Steve Bellovin, Vijayabhaskar, Brian Carpenter, Matt Crawford, Francis Dupont, Tony Lindstrom, Josh Littlefield, Gerald Maguire, Jack McCann, Thomas Narten, Erik Nordmark, Yakov Rekhter, Mark Stapp, Matt Thomas, Sue Thomson, 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. Changes in draft-ietf-dhc-dhcpv6-25.txt

- Eliminated definition of VUID-DN.

- Changed the second sentence in section 17.1.2 to:

  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 RFC2462).

- Changed Rapid Commit to allow client to use Advertise messages received while waiting for Reply, rather than restarting with Solicit; if client receives Advertise with preference 255, client immediately sends Request to that server.

- Removed the use of All_DHCP_Servers multicast address as destination address in section 18.1.5.

- Added text improving summary description of Confirm, Renew, Rebind.

- Removed restriction on extension of lifetimes for temporary addresses; added text pointing to RFC 3041 for guidance on extending lifetimes for temporary addresses and when to request additional temporary addresses.

- Clarified text in section 20 to emphasize that the relay agent puts an address in the link-address field regardless of whether it includes an Interface-ID option; added text explaining why the interface-identifier for an interface should remain stable.

- Changed use of T1/T2 and lifetimes in Confirm message from client: client uses those fields for preferred values or sets to 0; server checks only addresses for correctness and returns values chosen by server for T1/T2 and lifetimes. Clarified

that server checks addresses and returns current configuration information for the client.

- Description of User Class option extended with an example and clarified to indicate that use of User Class options is determined by configuration/policies on server.

- Clarified description of the use of Vendor-specific information to indicate that client need not receive all requested vendor-specific information before proceeding with normal operation.

- Clarified use of Status Code option in Release message.

- Edited section 14 to make clear that a client transmits until it receives a response only if both MRC and MRD are zero.

- Removed suggestions about ordering options (for example, for improved performance).

- Edited section 16 to clarify interface selection.

- Removed use of anycast; use of anycast over individual link technologies will be specified in separate documents.

- Removed replay detection information field from Solicit message to avoid potential DOS attack.

- Clarified capabilities and constraints on relay agent forwarding.

- Edited definition of vendor class data to clarify that instances

of vendor-class-data are individual characteristics of the client.

- Added text in section 21.5.4 to specify that client key is identified by client DUID.

- Removed "Year 2000 Considerations" section; hope we don't need a "Year 3000 Consideration" section.

- Authentication mechanism now shares Protocol, Algorithm and RDM name spaces with DHCPv4.

- Added text to specify return of NoBinding if server cannot find binding for IA in Decline; added text allowing client to disregard NoBinding in Reply to Decline.

- Clarify that Solicit, Confirm and Rebind are invalid if Server Identifier option is included.

- Edited text about Option Request option to clarify that the option is a hint from the client to the server about options the

client has a preference to receive; includes recommendation that client send Option Request option if it has options it requires.

- Added nonce value for security of Reconfigure messages.

References

[1] S. Alexander and R. Droms. DHCP Options and BOOTP Vendor Extensions, March 1997. RFC 2132.

[2] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels, March 1997. RFC 2119.

[3] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification, December 1998. RFC 2460.

[4] R. Droms. Dynamic Host Configuration Protocol, March 1997. RFC 2131.

[5] R. Droms, Editor, W. Arbaugh, and Editor. Authentication for DHCP Messages, June 2001. RFC 3118.

[6] R. Hinden and S. Deering. IP Version 6 Addressing Architecture, July 1998. RFC 2373.

[7] IANA. Private Enterprise Numbers. http://www.iana.org/assignments/enterprise-numbers.

[8] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol, November 1998. RFC 2401.

[9] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication, February 1997. RFC 2104.

[10] David L. Mills. Network Time Protocol (Version 3) Specification, Implementation, March 1992. RFC 1305.

[11] P.V. Mockapetris. Domain names - implementation and specification, November 1987. RFC 1035.

[12] T. Narten and H. Alvestrand. Guidelines for Writing an IANA Considerations Section in RFCs, October 1998. RFC 2434.

[13] T. Narten and R. Draves. Privacy Extensions for Stateless Address Autoconfiguration in IPv6, January 2001. RFC 3041.

[14] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6), December 1998. RFC 2461.

[15] D.C. Plummer. Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware, November 1982. RFC 826.

[16] J. Postel. User Datagram Protocol, August 1980. RFC 768.

[17] R. Rivest. The MD5 Message-Digest Algorithm, April 1992. RFC 1321.

[18] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration, December 1998. RFC 2462.

[19] P. Vixie, Ed., S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS UPDATE), April 1997. RFC 2136.

Chair's Address

The working group can be contacted via the current chair:

Ralph Droms
Cisco Systems
300 Apollo Drive
Chelmsford, MA 01824

Phone: (978) 244-4733
E-mail: rdroms@cisco.com

Authors' Addresses

Questions about this memo can be directed to:

Jim Bound
Hewlett Packard Corporation
ZK3-3/W20
110 Spit Brook Road
Nashua, NH 03062-2698
USA
Voice: +1 603 884 0062
E-mail: Jim.Bound@hp.com

Mike Carney
Sun Microsystems, Inc
Mail Stop: UMPK17-202
901 San Antonio Road
Palo Alto, CA 94303-4900
USA
Voice: +1-650-786-4171
E-mail: mwc@eng.sun.com

Charles E. Perkins
Communications Systems Lab
Nokia Research Center
313 Fairchild Drive
Mountain View, California 94043
USA
Voice: +1-650 625-2986
E-mail: charliep@iprg.nokia.com
Fax: +1 650 625-2502

Ted Lemon
Nominum, Inc.
950 Charter Street
Redwood City, CA 94043
E-mail: Ted.Lemon@nominum.com

Bernie Volz
Ericsson
959 Concord St
Framingham, MA 01701
Voice: +1-508-875-3162
Fax: +1-508-875-3018
E-mail: bernie.volz@ericsson.com

Ralph Droms
Cisco Systems
300 Apollo Drive
Chelmsford, MA 01824

USA
Voice: +1 978 479 4733
E-mail: rdroms@cisco.com

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/ IA_TA	Option Request	Pref	Time	Client Msg.	Server Msg.
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 19.3.3).

	Auth	Server Unica.	Status Code	Rap. Comm.	User Class	Vendor Class	Vendor Spec.	Inter. ID	Recon. Msg.
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/ IA_TA	IAADDR	Relay Forw.	Relay Reply
Client ID	*				
Server ID	*				
IA/IA_TA	*				
IAADDR		*			
ORO	*				
Pref	*				
Time	*				
Authentic.	*				
Server Uni.	*				
Status Code	*	*	*	*	*
Rapid Comm.	*				
User Class	*				
Vendor Class	*				
Vendor Info.	*				
Interf. ID				*	*
Reconf. msg.	*				

C. Full Copyright Statement