Internet Engineering Task Force
INTERNET DRAFT
DHC Working Group
Obsoletes: draft-ietf-dhc-dhcpv6-19.txt

J. Bound
Compaq
M. Carney
Sun Microsystems, Inc
C. Perkins
Nokia Research Center
R. Droms(ed.)
Cisco Systems
15 Oct 2001

Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
draft-ietf-dhc-dhcpv6-20.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" [20], and can be used separately or concurrently with the latter to obtain configuration parameters.

Contents

1. Introduction

This document describes DHCP for IPv6 (DHCP), a UDP [18] client/server protocol designed to reduce the cost of management of IPv6 nodes in environments where network managers require more control over the allocation of IPv6 addresses and configuration of network stack parameters than that offered by "IPv6 Stateless Address Autoconfiguration" [20]. DHCP is a stateful counterpart to stateless autoconfiguration. Note that both stateful and stateless autoconfiguration can be used concurrently in the same environment, leveraging the strengths of both mechanisms in order to reduce the cost of ownership and management of network nodes.

DHCP reduces the cost of ownership by centralizing the management of network resources such as IP addresses, routing information, OS installation information, directory service information, and other such information on a few DHCP servers, rather than distributing such information in local configuration files among each network node. DHCP is designed to be easily extended to carry new configuration parameters through the addition of new DHCP "options" defined to carry this information.

Those readers familiar with DHCP for IPv4 [7] will find DHCP for IPv6 provides a superset of features, and benefits from the additional features of IPv6 and freedom from the constraints of backward compatibility with BOOTP [5]. For more information about the differences between DHCP for IPv6 and DHCP for IPv4, see Appendix A.

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 [3].

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 is the IPv6 Specification [6], the IPv6 Addressing Architecture [9], IPv6 Stateless Address Autoconfiguration [20], IPv6 Neighbor Discovery Processing [16], and Dynamic Updates to DNS [22]. These specifications enable DHCP to build upon the IPv6 work to

provide both robust stateful autoconfiguration and autoregistration of DNS Host Names.

The IPv6 Addressing Architecture specification [9] 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.

IPv6 Stateless Address Autoconfiguration [20] specifies procedures by which a node may autoconfigure addresses based on router advertisements [16], 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 (see Section 4).

IPv6 Neighbor Discovery [16] is the node discovery protocol in IPv6 which replaces and enhances functions of ARP [17]. To understand IPv6 and stateless address autoconfiguration it is strongly recommended that implementors understand IPv6 Neighbor Discovery.

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

4. Design Goals

- DHCP is a mechanism rather than a policy. Network administrators set their administrative policies through the configuration parameters they place upon the DHCP servers in the DHCP domain they're managing. DHCP is simply used to deliver parameters according to that policy to each of the DHCP clients within the domain.

- DHCP is compatible with IPv6 stateless address autoconfiguration [20], statically configured, non-participating nodes and with existing network protocol implementations.

- DHCP does not require manual configuration of network parameters on DHCP clients, except in cases where such configuration is needed for security reasons. A node configuring itself using DHCP should require no user intervention.

- DHCP does not require a server on each link. To allow for scale and economy, DHCP must work across DHCP relays.

- DHCP clients can operate on a link without IPv6 routers present.

- DHCP will provide the ability to renumber network(s) when required by network administrators [4].

- A DHCP client can make multiple, different requests for configuration parameters when necessary from one or more DHCP servers at any time.

- DHCP will contain the appropriate time out and retransmission mechanisms to efficiently operate in environments with high

latency and low bandwidth characteristics.

5. Non-Goals

This specification explicitly does not cover the following:

- Specification of a DHCP server to server protocol.
- How a DHCP server stores its DHCP data.
- How to manage a DHCP domain or DHCP server.
- How a DHCP relay is configured or what sort of information it may log.

6. Terminology

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

6.1. IPv6 Terminology

IPv6 terminology relevant to this specification from the IPv6 Protocol [6], IPv6 Addressing Architecture [9], and IPv6 Stateless Address Autoconfiguration [20] is included below.

address	An IP layer identifier for an interface or a set of interfaces.
unicast address	An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that 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.
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::0000/64), that can be used to reach neighboring nodes attached to the same link. Every interface has a link-local address.
message	A unit of data carried in a packet, exchanged between DHCP agents and clients.
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 address 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.

6.2. DHCP Terminology

Terminology specific to DHCP can be found below.

agent address	The address of a neighboring DHCP Agent on the same link as the DHCP client.
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, IAID>.
DHCP	Dynamic Host Configuration Protocol for IPv6. The terms DHCPv4 and DHCPv6 are used only in contexts where it is necessary to avoid ambiguity.
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 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 server (or server)	A server is a node that responds to requests from clients, and may or may not be on the same link as the

client(s).

DHCP relay (or relay) A node that acts as an intermediary to deliver DHCP messages between clients and servers, and is on the same link as a client.

DHCP agent (or agent) Either a DHCP server on the same link as a client, or a DHCP relay.

DUID A DHCP Unique IDentifier for a client.

Identity association (IA) A collection of addresses assigned to a client. Each IA has an associated IAID. An IA may have 0 or more addresses associated with it.

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.

transaction-ID An unsigned integer to match responses with replies initiated either by a client or server.

7. DHCP Constants

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

7.1. Multicast Addresses

DHCP makes use of the following multicast addresses:

All_DHCP_Agents address: FF02::1:2 This link-scoped multicast address is used by clients to communicate with the on-link agent(s) when they do not know the link-local

address(es) for those agents. All agents (servers and relays) are members of this multicast group.

All_DHCP_Servers address: FF05::1:3 This site-scoped multicast address is used by clients or relays to communicate with server(s), either because they want to send messages to all servers or because they do not know the server(s) unicast address(es). Note that in order for a client to use this address, it must have an address of sufficient scope to be reachable by the server(s). All servers within the site are members of this multicast group.

7.2. UDP ports

DHCP uses the following destination UDP [18] port numbers. While source ports MAY be arbitrary, client implementations SHOULD permit their specification through a local configuration parameter to facilitate the use of DHCP through firewalls.

546 Client port. Used by servers as the destination port for messages sent to clients and relays. Used by relay

agents as the destination port for messages sent to clients.

547 Agent port. Used as the destination port by clients for messages sent to agents. Used as the destination port by relays for messages sent to servers.

7.3. DHCP message types

DHCP defines the following message types. More detail on these message types can be found in Section 8. Message types 0 and 13-255 are reserved for future use. The message code for each message type is shown with the message name.

SOLICIT (1) The Solicit message is used by clients to locate servers.

ADVERTISE (2)	The Advertise message is used by servers responding to Solicits.
REQUEST (3)	The Request message is used by clients to request configuration parameters from servers.
CONFIRM (4)	The Confirm message is used by clients to confirm that the addresses assigned to an IA and the lifetimes for those addresses, as well as the current configuration parameters assigned by the server to the client are still valid.
RENEW (5)	The Renew message is used by clients to obtain the addresses assigned to an IA and the lifetimes for those addresses, as well as the current configuration parameters assigned by the server to the client. A client sends a Renew message to the server that originally assigned the IA when the lease on an IA is about to expire.
REBIND (6)	The Rebind message is used by clients to obtain the addresses assigned to an IA and the lifetimes for those addresses, as well as the current configuration parameters assigned by the server to the client. A clients sends a Rebind message to all available DHCP servers when the lease on an IA is about to expire.
REPLY (7)	The Reply message is used by servers responding to Request, Confirm, Renew, Rebind, Release and Decline messages. In the

case of responding to a Request, Confirm, Renew or Rebind message, the Reply contains configuration parameters destined for the client.

RELEASE (8)	The Release message is used by clients to return one or more IP addresses to servers.
DECLINE (9)	The Decline message is used by clients to indicate that the client has determined that one or more addresses in an IA are already in use on the link to which the client is connected.
RECONFIG-INIT (10)	The Reconfigure-init message is sent by server(s) to inform client(s) that the server(s) has new or updated configuration parameters, and that the client(s) are to initiate a Request/Reply transaction with the server(s) in order to receive the updated information.
RELAY-FORW (11)	The Relay-forward message is used by relays to forward client messages to servers. The client message is encapsulated in an option in the Relay-forward message.
RELAY-REPL (12)	The Relay-reply message is used by servers to send messages to clients through a relay. The server encapsulates the client message as an option in the Relay-reply message, which the relay extracts and forwards to the client.

7.4. Status Codes

This section describes status codes exchanged between DHCP implementations. These status codes may appear in the Status Code option or in the status field of an IA.

7.4.1. Generic Status Codes

The status codes in this section are used between clients and servers to convey status conditions. The following table contains the status codes, the name for each code (as used in this document) and a brief description. Note that the numeric values do not start at 1, nor are they consecutive. The status codes are organized in logical groups.

Name	Code	Description
Success	0	Success
UnspecFail	16	Failure, reason unspecified
AuthFailed	17	Authentication failed or nonexistent
PoorlyFormed	18	Poorly formed message
AddrUnavail	19	Addresses unavailable
OptionUnavail	20	Requested options unavailable

7.4.2. Server-specific Status Codes

The status codes in this section are used by servers to convey status conditions to clients. The following table contains the status codes, the name for each code (as used in this document) and a brief description. Note that the numeric values do not start at 1, nor are they consecutive. The status codes are organized in logical groups.

Name	Code	Description
NoBinding	32	Client record (binding) unavailable
ConfNoMatch	33	Client record Confirm not match IA
RenwNoMatch	34	Client record Renew not match IA
RebdNoMatch	35	Client record Rebind not match IA
InvalidSource	36	Invalid Client IP address
NoServer	37	Relay cannot find Server Address
NoPrefixMatch	38	One or more prefixes of the addresses in the IA is not valid for the link from which the client message was received
ICMPError	64	Server unreachable (ICMP error)

7.5. Configuration Variables

This section presents a table of client and server configuration variables and the default or initial values for these variables.

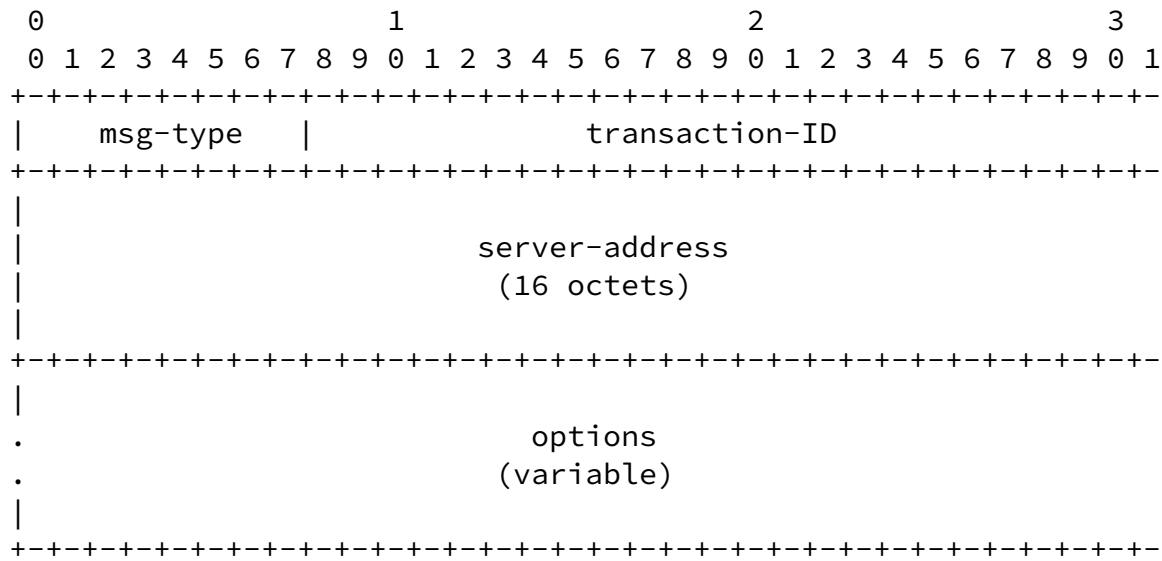
Parameter	Default	Description
MIN_SOL_DELAY	1 sec	Min delay of first Solicit
MAX_SOL_DELAY	5 secs	Max delay of first Solicit
SOL_TIMEOUT	500 msecs	Initial Solicit timeout
SOL_MAX_RT	30 secs	Max Solicit timeout value
REQ_TIMEOUT	250 msecs	Initial Request timeout
REQ_MAX_RT	30 secs	Max Request timeout 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 sec	Initial Rebind timeout
REB_MAX_RT	600 secs	Max Rebind timeout value
REL_TIMEOUT	250 msecs	Initial Release timeout
REL_MAX_RT	1 sec	Max Release timeout
REL_MAX_RC	5	MAX Release/Decline attempts
DEC_TIMEOUT	250 msecs	Initial Release timeout
DEC_MAX_RT	1 sec	Max Release timeout
DEC_MAX_RC	5	MAX Release/Decline attempts

8. Message Formats

All DHCP messages sent between clients and servers share an identical fixed format header and a variable format area for options. Not all fields in the header are used in every message.

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

The following diagram illustrates the DHCP message header:



The following sections describe the use of the fields in the DHCP message header in each of the DHCP messages. In these descriptions, fields that are not used in a message are marked as "unused". All unused fields in a message MUST be transmitted as zeroes and ignored by the receiver of the message.

8.1. DHCP Solicit Message Format

msg-type	SOLICIT
transaction-ID	An unsigned integer generated by the client used to identify this Solicit message.
server-address	(unused) MUST be 0
options	See section 20.

8.2. DHCP Advertise Message Format

msg-type	ADVERTISE
transaction-ID	An unsigned integer used to identify this Advertise message. Copied from the Solicit message received from the client.
server-address	The IP address of the server that generated this message. The address must have sufficient scope to be reachable from the client.
options	See section 20.

8.3. DHCP Request Message Format

msg-type	REQUEST
transaction-ID	An unsigned integer generated by the client used to identify this Request message.

server-address	The IP address of the server to which the this message is directed, copied from an Advertise message.
options	See section 20.

8.4. DHCP Confirm Message Format

msg-type	CONFIRM
transaction-ID	An unsigned integer generated by the client used to identify this Confirm message.
server-address	MUST be zero.
options	See section 20.

8.5. DHCP Renew Message Format

msg-type	RENEW
transaction-ID	An unsigned integer generated by the client used to identify this Renew message.
server-address	The IP address of the server to which this Renew message is directed, which MUST be the address of the server from which the IAs in this message were originally assigned.
options	See section 20.

8.6. DHCP Rebind Message Format

msg-type	REBIND
transaction-ID	An unsigned integer generated by the client used to identify this Rebind message.
server-address	MUST be zero.
options	See section 20.

8.7. DHCP Reply Message Format

msg-type	REPLY
transaction-ID	An unsigned integer used to identify this Reply message. Copied from the client Request, Confirm, Renew or Rebind message received from the client.
server-address	The IP address of the server. The address must have sufficient scope to be reachable from the client.
options	See section 20.

8.8. DHCP Release Message Format

msg-type	RELEASE
transaction-ID	An unsigned integer generated by the client used to identify this Release message.
server-address	The IP address of the server that assigned the addresses.
options	See section 20.

8.9. DHCP Decline Message Format

msg-type	DECLINE
transaction-ID	An unsigned integer generated by the client used to identify this Decline message.
server-address	The IP address of the server that assigned the addresses.
options	See section 20.

8.10. DHCP Reconfigure-init Message Format

msg-type	RECONFIG-INIT
transaction-ID	An unsigned integer generated by the server used

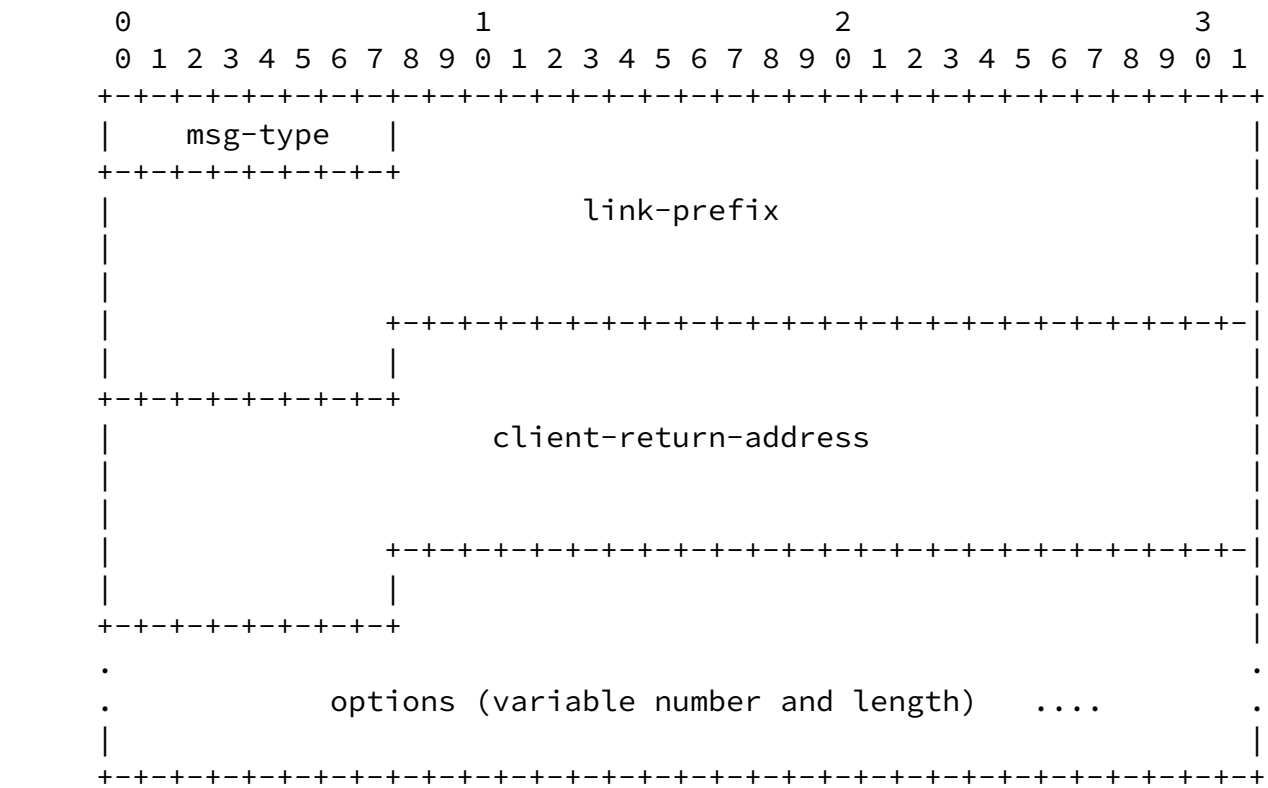
to identify this Reconfigure-init message.

server-address The IP address of the DHCP server issuing the Reconfigure-init message. The address must have sufficient scope to be reachable from the client.

options See section 20.

9. Relay 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 messages, which share the following format:



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

9.1. Relay-forward message

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

msg-type	RELAY-FORW
link-prefix	An address with a prefix that is assigned to the link from which the client should be assigned an address.
client-return-address	The source address from the IP datagram in which the message from the client was received by the relay agent
options	MUST include a "Client message option"; see section 20.7; MAY include other options added by the relay agent

9.2. Relay-reply message

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

msg-type	RELAY-REPL
link-prefix	An address with a prefix that is assigned to the link from which the client should be assigned an address.
client-return-address	The source address from the IP datagram in which the message from the client was received by the relay agent
options	MUST include a "Server message option"; see section 20.8; MAY include other options

10. DHCP unique identifier (DUID)

Each DHCP client has a DUID. DHCP servers use DUIDs to identify

clients for the selection of configuration parameters and in the association of IAs with clients. See section 20.2 for the representation of a DUID in a DHCP message.

Servers MUST treat DUIDs as opaque values and must only compare DUIDs for equality. Servers MUST NOT in any other way interpret DUIDs. 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 options (e.g., messages sent by servers need not include a DUID). The DUID must be unique across all DHCP clients, and it must also be consistent for the same client - that is, the DUID used by a client SHOULD NOT change over time; for example, as a result of network hardware reconfiguration.

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.

10.1. DUID contents

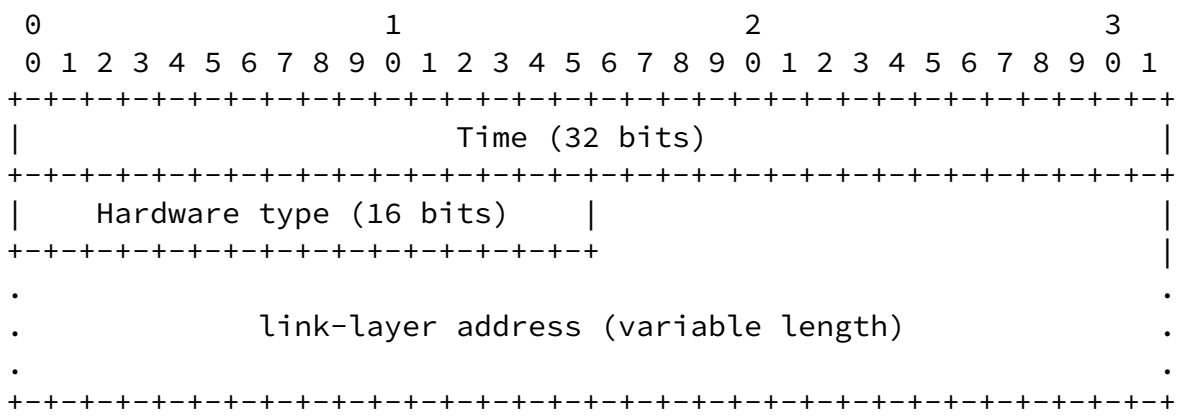
A DUID consists of a sixteen-bit type code represented in network order, followed by a variable number of octets that make up the actual identifier. A DUID can be no more than 256 octets long. The following types are currently defined:

1	Link-layer address plus time
2	Vendor-assigned unique ID
3	Link-layer address

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

10.2. DUID based on link-layer address plus time

This type of DUID consists of four octets containing a time value, followed by a two octet network hardware type code, followed by link-layer address of any one network interface that is connected to the DHCP client 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 order.



The choice of network interface can be completely arbitrary, as long as that interface provides a unique link-layer address, 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.

DHCP clients 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. DHCP clients that do not have any stable storage MUST NOT use this type of DUID.

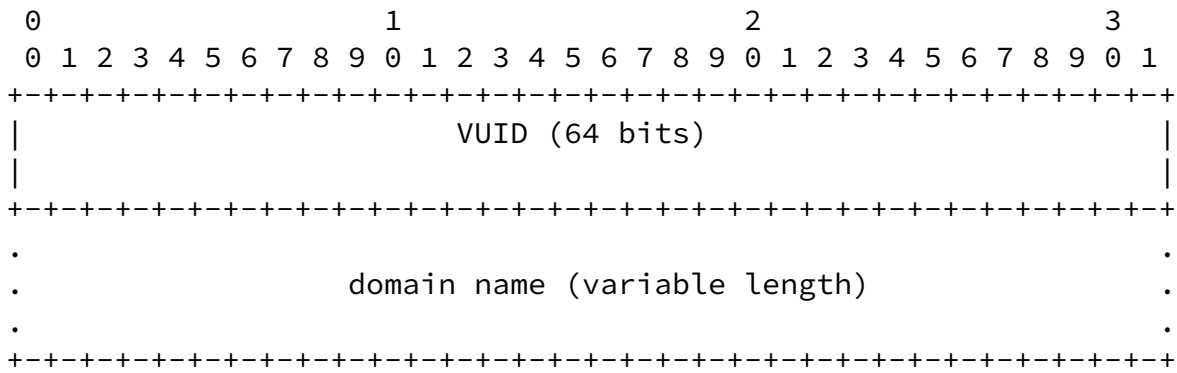
DHCP clients that use this DUID SHOULD attempt to configure the time prior to generating the DUID, if that is possible, and MUST use some sort of time source (e.g., a real-time clock) in generating the DUID, even if that time source is not configured by the user prior to generating the DUID. The use of a time source makes it unlikely that if the network interface is removed from the client and another client then uses the same network interface to generate a DUID, that two identical DUIDs will be generated. A DUID collision is

very unlikely even if the clocks haven't been configured prior to generating the DUID.

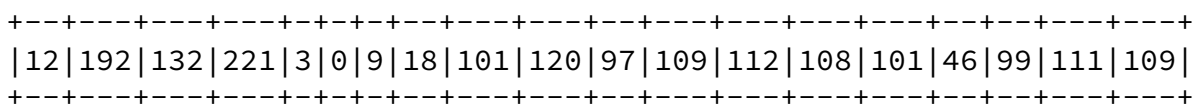
This method of DUID generation is recommended for all general purpose computing devices such as desktop computers and laptop computers, and also for devices such as printers, routers, and so on, that contain some form of writable non-volatile storage.

10.3. Vendor-assigned unique ID.

The vendor-assigned unique ID consists of an eight-octet vendor-unique identifier, followed by the vendor's registered domain name.



The structure of the VUID is left up to the vendor defining it, but each device containing such a VUID MUST be unique to each 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 domain name is simply any domain name that has been legally registered by the vendor in the domain name system, stored in canonical form. An example DUID of this type might look like this:

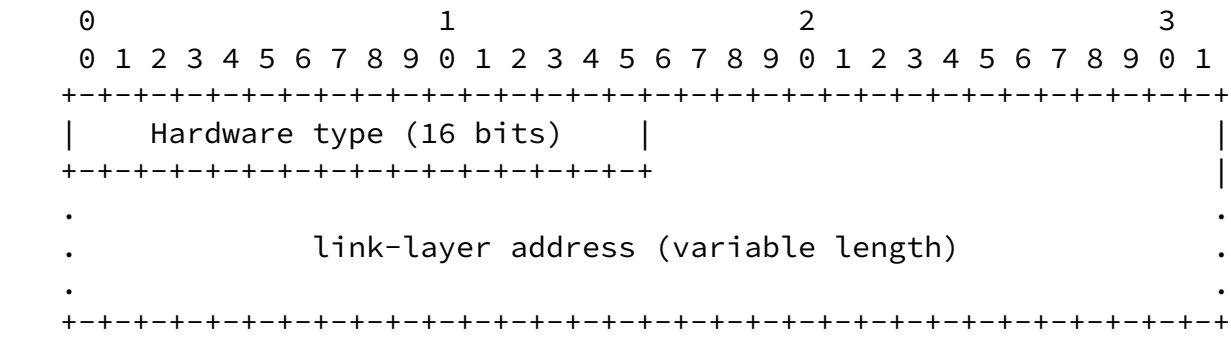


This is eight octets of VUID data, followed by "example.com" represented in ASCII.

10.4. Link-layer address

This type of DUID consists of a two octet network hardware type code, followed by the link-layer address of any one network interface that is permanently connected to the DHCP client device. 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 order.



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 that cannot tell whether or not a network interface is permanently attached to the device on which the DHCP client is running.

11. 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 a list of associated IPv6 addresses (the list may be empty). A client associates an IA with one of its interfaces and uses the IA to obtain IPv6 addresses for that interface from a server.

See section 20.3 for the representation of an IA in a DHCP message.

12. 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 the following sources:

- The link to which the client 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 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 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
    * 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-prefix field in the message from the relay is attached
- The DUID supplied by the client
- Other information in options supplied by the client
- Other information in options supplied by the relay agent

13. Reliability of Client Initiated Message Exchanges

DHCP clients are responsible for reliable delivery of messages in the client-initiated message exchanges described in sections 15 and 16. 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.

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 describe by five 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 numbers from each invocation of the DHCP client.

RT for the first message transmission is based on IRT:

```
RT = 2*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. If MRC has a value of 0, the client MUST continue to retransmit the original message until a response is received. Otherwise, the message exchange fails if the client attempts to transmit the original message more than MRC times.

MRD specifies an upper bound on the length of time a client may retransmit a message. If MRD has a value of 0, the client MUST continue to retransmit the original message until a response is received. Otherwise, the message exchange fails if the client attempts to transmit the original message more than MRD seconds.

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

14. Message validation

Servers MUST discard any received messages that include authentication information and fail the authentication check by the server.

Clients MUST discard any received messages that include authentication information and fail the authentication check by the client, except as noted in section 19.6.5.2.

14.1. 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.

14.2. Solicit message

Clients MUST discard any received Solicit messages.

Relay agents MUST discard any Solicit messages received through port 546.

14.3. Advertise message

Clients MUST discard any received Advertise messages in which 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.

14.4. Request message

Clients MUST discard any received Request messages.

Relay agents MUST discard any Request messages received through port 546.

Servers MUST discard any received Request message in which the value in the ``server-address'' field does not match any of the addresses used by the server.

14.5. Confirm message

Clients MUST discard any received Confirm messages.

Relay agents MUST discard any Confirm messages received through port 546.

14.6. Renew message

Clients MUST discard any received Renew messages.

Relay agents MUST discard any Renew messages received through port 546.

Servers MUST discard any received Renew message in which the value in the ``server-address'' field does not match any of the addresses used by the server.

14.7. Rebind message

Clients MUST discard any received Rebind messages.

Relay agents MUST discard any Rebind messages received through port 546.

14.8. Decline messages

Clients MUST discard any received Decline messages.

Relay agents MUST discard any Decline messages received through port 546.

Servers MUST discard any received Decline message in which the value in the ``server-address'' field does not match any of the addresses used by the server.

14.9. Release message

Clients MUST discard any received Release messages.

Relay agents MUST discard any Release messages received through port 546.

Servers MUST discard any received Release message in which the value in the ``server-address'' field does not match any of the addresses used by the server.

14.10. Reply message

Clients MUST discard any received Reply messages in which the ``transaction-ID'' field in the message does not match the value used

in the original message.

Servers and relay agents MUST discard any received Reply messages.

14.11. Reconfigure-init message

Servers and relay agents MUST discard any received Reconfigure-init messages.

Clients MUST discard any Reconfigure-init messages that do not contain an authentication option or that fail the authentication performed by the client.

14.12. Relay-forward message

Clients MUST discard any received Relay-forward messages.

14.13. Relay-reply message

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

15. DHCP Server Solicitation

This section describes how a client locates servers.

15.1. 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.

15.1.1. 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 DUID option to identify itself to the server. The client MUST include options for any IAs to which it wants the server to assign addresses. The client MAY choose not to include any IAs in the Solicit message if it does not need to request that any addresses be assigned. The client MAY include addresses in the IAs as a hint to the server about addresses for which the client may have a preference. The client MAY include an Option Request Option in the Solicit message. The client MUST NOT include any other options except those specifically allowed as defined by specific options.

15.1.2. Transmission of Solicit Messages

The client sends the Solicit message to the All_DHCP_Agents multicast address. The client MUST use an IPv6 address 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 Solicit message.

The Solicit message MUST be transmitted on the link that the interface for which configuration information is being obtained is attached to. The client SHOULD send the message through that interface. The client MAY send the message through another interface attached to the same link if and only if the client is certain the the two interface are attached to the same link.

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. This random delay desynchronizes clients which start at the same time (e.g., after a power outage).

The client transmits the message according to section 13, using the following parameters:

IRT SOL_TIMEOUT

MRT SOL_MAX_RT

MRC 0

MRD 0

The mechanism in section 13 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 SOL_TIMEOUT has elapsed. Rather, the client collects Advertise messages until SOL_TIMEOUT has elapsed. The first RT MUST be selected to be strictly greater than SOL_TIMEOUT by choosing RAND to be strictly greater than 0.

A client MUST collect Advertise messages for SOL_TIMEOUT seconds, unless it receives an Advertise message with a preference value of 255. The preference value is carried in the Preference option (section 20.5). 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 MAY act immediately on that Advertise message without waiting for any more 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 MAY 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.

15.1.3. 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. Depending on the requirements of the user that invoked the DHCP client, the client MAY initiate a configuration exchange with the server(s) immediately, or MAY defer this exchange until later.

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.

15.2. Server Behavior

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

15.2.1. Receipt of Solicit messages

The server determines the information about the client and its location as described in section 12. If administrative policy permits the server to respond to the client, the server will generate and send an Advertise message to the client.

15.2.2. 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 places one of its IP addresses (determined through administrator setting) in the "server-address" field of 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 Solicit message from the client included no IAs, the server MUST not include any IAs in the Advertise message. If the server will not assign any addresses to IAs in a subsequent Request from the client, the server MAY choose to 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 MAY include other options the server will return to the client in a subsequent Reply message. 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. The server SHOULD include options specifying values for options requested by the client in an Option Request Option included in the Solicit message.

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. 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.

16. 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 or when requested to do so by the application layer.

16.1. Client Behavior

A client will use Request, Confirm, Renew and Rebind messages to acquire and confirm the validity of configuration information. The client uses the server address information from previous Advertise message(s) for use in constructing Request and Renew message(s). Note that a client may request configuration information from one or more servers at any time.

16.1.1. Creation and transmission of Request messages

If the client is using stateful address configuration and needs either an initial set of addresses or additional addresses, it MUST send a Request message to obtain new addresses and other configuration information. The client includes one or more IAs in the Request message, to which the server assigns new addresses. The server then returns IA(s) to the client in a Reply message.

The client generates a transaction ID and inserts this value in the "transaction-ID" field.

The client places the address of the destination server in the "server-address" field.

The client MUST include a DUID 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 list of addresses in each included IA MUST be empty. If the client is not requesting that the server assign it any addresses, the client omits the IA option.

If the client has a source address that can be used by the server as a return address and the client has received a Client Unicast option (section 20.11) from the server, the client SHOULD unicast the Request message to the server. Otherwise, the client MUST send the Request message to the All_DHCP_Agents multicast address. The client MUST use an address 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 Request message.

DISCUSSION:

Use of multicast 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.

If the client multicasts the Request message, the message MUST be transmitted on the link that the interface for which configuration information is being obtained is attached to. The client SHOULD send the message through that interface. The client MAY send the message through another interface attached to the same link if and only if the client is certain the the two interface are attached to the same link.

The client transmits the message according to section 13, 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; e. g., servers that responded with an Advertise message
- Initiate the server discovery process described in section 15
- Terminate the configuration process and report failure

16.1.2. 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
- o The client returns from sleep mode
- o The client using a wireless technology changes cells

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 sets the "server-address" field to 0.

The client MUST include a DUID option to identify itself to the

server. The client adds any appropriate options, including one or more IA options (if the client is requesting that the server confirm the validity of some IPv6 addresses). If the client does include any IA options, it MUST include the list of addresses the client currently has associated with that IA.

The client sends the Confirm message to the All_DHCP_Agents multicast address. The client MUST use an IPv6 address 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 Confirm message MUST be transmitted on the link that the interface for which configuration information is being obtained is attached to. The client SHOULD send the message through that interface. The client MAY send the message through another interface attached to the same link if and only if the client is certain the the two interface are attached to the same link.

The client transmits the message according to section 13, 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 13 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.

16.1.3. Creation and transmission of Renew messages

IPv6 addresses assigned to a client through an IA use the same preferred and valid lifetimes as IPv6 addresses obtained through stateless address autoconfiguration. The server assigns preferred and valid lifetimes to the IPv6 addresses it assigns to an IA. To extend those lifetimes, 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 the server does not assign an explicit value to T1 or T2 for an IA, T1 defaults to 0.5 times the shortest preferred lifetime of any address assigned to the IA and T2 defaults to 0.875 times the shortest preferred lifetime of any address assigned to 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 address of the destination server in the "server-address" field.

The client MUST include a DUID option to identify itself to the server. The client adds any appropriate options, including one or more IA options (if the client is requesting that the server extend the lease on some IAs; note that the client may check the status of other configuration parameters without asking for lease extensions). If the client does include any IA options, it MUST include the list of addresses the client currently has associated with that IA.

If the client has a source address that can be used by the server as a return address and the client has received a Client Unicast option (section 20.11) from the server, the client SHOULD unicast the Renew message to the server. Otherwise, the client sends the Renew message to the All_DHCP_Agents multicast address. The client MUST use an address 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 Renew message.

If the Renew message is multicast, it MUST be transmitted on the link that the interface for which configuration information is being obtained is attached to. The client SHOULD send the message through that interface. The client MAY send the message through another interface attached to the same link if and only if the client is certain the the two interface are attached to the same link.

The client transmits the message according to section 13, using the following parameters:

IRT REN_TIMEOUT

MRT REP_MAX_RT

MRC 0

MRD 0

The mechanism in section 13 is modified as follows for use in the transmission of Renew messages. The message exchange is terminated when time T2 is reached (see section 16.1.4), at which time the client begins a Rebind message exchange.

16.1.4. 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. The client includes an IA option with all addresses currently assigned to the IA in its Rebind message. The client sends this message to the All_DHCP_Agents multicast address.

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

The client sets the "server-address" field to 0.

The client MUST include a DUID option to identify itself to the server. The client adds any appropriate options, including one or more IA options. If the client does include any IA options (if the client is requesting that the server extend the lease on some IAs; note that the client may check the status of other configuration parameters without asking for lease extensions), it MUST include the list of addresses the client currently has associated with that IA.

The client sends the Rebind message to the All_DHCP_Agents

multicast address. The client MUST use an IPv6 address 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 Rebind message.

The Rebind message MUST be transmitted on the link that the interface for which configuration information is being obtained is attached to. The client SHOULD send the message through that interface. The client MAY send the message through another interface attached to the same link if and only if the client is certain the the two interface are attached to the same link.

The client transmits the message according to section 13, using the following parameters:

IRT REB_TIMEOUT

MRT REB_MAX_RT

MRC 0

MRD 0

The mechanism in section 13 is modified as follows for use in the transmission of Rebind messages. The message exchange is terminated when the lease for the IA expires (see section 11), at which time the client has several alternative actions to choose from:

- When the lease on the IA expires, 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

- Some addresses in the IA may have lifetimes that extend beyond the lease of the IA, so the client may choose to continue to use those addresses; once all of the addresses have expired, the client may choose to locate a new DHCP 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

16.1.5. Receipt of Reply message in response to a Request, Confirm, Renew or Rebind message

Upon the receipt of a valid Reply message in response to a Request, Confirm, Renew or Rebind 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 [20] on each of the addresses in any IAs it receives in the Reply message. 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 16.1.8.

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.

Management of the specific configuration information is detailed in the definition of each option, in section 20.

When the client receives a NoPrefixMatch status in an IA from the server 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 NoPrefixMatch status, the

client can use the addresses in the IA and ignore any messages that do indicate a NoPrefixMatch status.

When the client receives an AddrUnavail status in an IA from the server for a Request message the client will have to find a new server to create an IA.

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

When the client receives a ConfNoMatch status in an IA from the server for a Confirm message the client can send a Renew message to the server to extend the lease for the addresses.

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 RenwNoMatch 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 an AddrUnavail 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 a RebdNoMatch 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.

16.1.6. Creation and transmission of Release messages

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 IP address of the server that allocated the address(es) in the "server-address" field.

The client MUST include a DUID option to identify itself to the server. The client includes options containing the IAs it is releasing in the "options" field. The addresses to be released MUST be included in the IAs. The appropriate "status" field in the options MUST be set to indicate the reason for the release.

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

If the client has a source address that can be used by the server as a return address and the client has received a Client Unicast option (section 20.11) from the server, the client SHOULD unicast the Release message to the server. Otherwise, the client MUST send the Release message to the All_DHCP_Agents multicast address. The client MUST use an address for the interface to which the IAs in the Release message are assigned as the source address for the Release message.

DISCUSSION:

> Use of multicast 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.

If the Release message is multicast, it MUST be transmitted on the link that the interface for which configuration information is being obtained is attached to. The client SHOULD send the message through that interface. The client MAY send the message through another interface attached to the same link if and only if the client is certain the the two interface are attached to the same link.

A client MAY choose to wait for a Reply message from the server in response to the Release message. If the client does wait for a Reply, the client MAY choose to retransmit the Release message.

The client transmits the message according to section 13, 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 in the IA(s) being released as soon as the client begins the Release message exchange process. If an IA is 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 IA, the addresses assigned to the IA will be reclaimed by the server when the lease associated with it expires.

16.1.7. 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, and SHOULD return the successful status to the application layer, if an application initiated the release.

16.1.8. 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 IP address of the server that allocated the address(es) in the "server-address" field.

The client MUST include a DUID option to identify itself to the server. The client includes options containing the IAs it is declining in the "options" field. The addresses to be released MUST be included in the IAs. The appropriate "status" field in the options MUST be set to indicate the reason for declining the address.

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

If the client has a source address that can be used by the server as a return address and the client has received a Client Unicast option (section 20.11) from the server, the client SHOULD unicast the Decline message to the server. Otherwise, the client MUST send the Decline message to the All_DHCP_Agents multicast address. The client MUST use an IPv6 address for the interface to which the IAs in the Release message are assigned as the source address for the Decline message.

DISCUSSION:

Use of multicast 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.

If the Decline message is multicast, it MUST be transmitted on the link that the interface for which configuration information is being obtained is attached to. The client SHOULD send the message through that interface. The client MAY send the message through another interface attached to the same link if and only if the client is certain the the two interface are attached to the same link.

The client transmits the message according to section 13, 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.

16.1.9. 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.

16.2. Server Behavior

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

16.2.1. Receipt of Request messages

The server MAY choose to discard Request messages received via unicast from a client to which the server has not sent a unicast option.

Upon the receipt of a valid Request message from a client the server can respond to, (implementation-specific administrative policy satisfied) the server scans the options field.

The server then constructs a Reply message and sends it to the client.

The server SHOULD process each option for the client in an implementation-specific manner. The server MUST construct a Reply message containing the following values:

msg-type	REPLY
transaction-ID	The transaction-ID from the Request message.
server address	One of the IP addresses assigned to the interface through which the server received the message from the client.

When the server receives a Request and IA option is included the client is requesting the configuration of a new IA by the server. The server MUST take the IA from the client and associate a binding for that client in an implementation-specific manner within the configuration parameter database for DHCP clients managed by the server.

If the server finds that the prefix on one or more IP addresses in any IA in the message fro 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 NoPrefixMatch.

If the server cannot provide addresses to the client it SHOULD send back an empty IA to the client with the status field set to

AddrUnavail.

If the server can provide addresses to the client it MUST send back the IA to the client with all fields entered and a status of Success, and add the IA as a new client binding.

The server adds options to the Reply message for any other configuration information to be assigned to the client.

16.2.2. Receipt of Confirm messages

Upon the receipt of a valid Confirm message from a client the server can respond to, (implementation-specific administrative policy satisfied) the server scans the options field.

The server then constructs a Reply message and sends it to the client.

The server SHOULD process each option for the client in an implementation-specific manner. The server MUST construct a Reply message containing the following values:

msg-type	REPLY
transaction-ID	The transaction-ID from the Confirm message.
server address	One of the IP addresses assigned to the interface through which the server received the message from the client.

When the server receives a Confirm message, the client is requesting confirmation that the configuration information it will use is valid. The server SHOULD locate the binding for that client and compare the information in the Confirm message from the client to the information associated with that client.

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.

If the server finds that the information for the client does not match what is in the binding for that client or the configuration information is not valid, the server sends a Reply message containing a Status Code option with the value ConfNoMatch.

If the server finds that the information for the client does match the information 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.

The Reply message from the server MUST contain a Status Code option and MUST NOT include any other options.

16.2.3. Receipt of Renew messages

The server MAY choose to discard Renew messages received via unicast from a client to which the server has not sent a unicast option.

Upon the receipt of a valid Renew message from a client the server can respond to, (implementation-specific administrative policy satisfied) the server scans the options field.

The server then constructs a Reply message and sends it to the client.

The server SHOULD process each option for the client in an implementation-specific manner. The server MUST construct a Reply message containing the following values:

msg-type	REPLY
transaction-ID	The transaction-ID from the Confirm message.
server address	One of the IP addresses assigned to the interface through which the server received the message from the client.

When the server receives a Renew and IA option from a client it SHOULD locate the clients binding and verify the information in the IA from the client matches the information stored for that client.

If the server cannot find a client entry for this IA the server SHOULD return an empty IA with status set to NoBinding.

If the server finds that the addresses in the IA for the client do not match the clients binding the server should return an empty IA with status set to RenwNoMatch.

If the server cannot Renew addresses for the client it SHOULD send back an empty IA to the client with the status field set to AddrUnavail.

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 lease times and T1/T2 times if the default is not being used, and set status to Success.

16.2.4. Receipt of Rebind messages

Upon the receipt of a valid Rebind message from a client the server can respond to, (implementation-specific administrative policy satisfied) the server scans the options field.

The server then constructs a Reply message and sends it to the client.

The server SHOULD process each option for the client in an implementation-specific manner. The server MUST construct a Reply message containing the following values:

msg-type	REPLY
transaction-ID	The transaction-ID from the Confirm message.
server address	One of the IP addresses assigned to the interface through which the server received the message from the client.

When the server receives a Rebind and IA option from a client it SHOULD locate the clients binding and verify the information in the IA from the client matches the information stored for that client.

If the server cannot find a client entry for this IA the server SHOULD return an empty IA with status set to NoBinding.

If the server finds that the addresses in the IA for the client do not match the clients binding the server should return an empty IA with status set to RebdNoMatch.

If the server cannot Rebind addresses for the client it SHOULD send back an empty IA to the client with the status field set to AddrUnavail.

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 lease times and T1/T2 times if the default is not being used, and set status to Success.

There is a significant difference between Renew and Rebind messages: Because the Renew message is processed by a single server, the

responding server can actually change the addresses in the IA. However, because multiple servers may respond to a Rebind, all they can safely do is update T1, T2 (for the IA) and lifetimes (for individual addresses).

16.2.5. Receipt of Release messages

The server MAY choose to discard Release messages received via unicast from a client to which the server has not sent a unicast option.

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 then generates a Reply message. If all of the IAs were valid and the addresses successfully released, the server includes a Status Code option with value Success. If any of the IAs were invalid or if any of the addresses were not successfully released, the server leaves all of the IAs in the message unchanged (the server releases none of the addresses in any of the IAs in the message) and includes a Status Code option with value NoBinding. The server MUST NOT include any other options in the Reply message.

A client can send an option containing an IA with no listed addresses

to release implicitly all of the addresses in the IA.

A server is not required to (but may choose to as an implementation strategy) retain any record of an IA from which all of the addresses have been released.

16.2.6. Receipt of Decline messages

The server MAY choose to discard Decline messages received via unicast from a client to which the server has not sent a unicast option.

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 then generates a Reply message. If all of the IAs were valid and the addresses successfully declined,, the server includes

a Status Code option with value Success. If any of the IAs were invalid or if any of the addresses were not successfully declined, the server leaves all of the IAs in the message unchanged (the server releases none of the addresses in any of the IAs in the message) and includes a Status Code option with value NoBinding. The server MUST NOT include any other options in the Reply message.

A client can send an option containing an IA with no listed addresses to decline implicitly all of the addresses in the IA.

16.2.7. Sending of Reply messages

If the Request, Confirm, Renew, Rebind, Release or Decline message from the client was originally received in a Relay-forward message from a relay, the server places the Reply message in the options field of a Relay-response message and copies the link-prefix and

client-return-address fields from the Relay-forward message into the Relay-response message.

The server then unicasts the Reply or Relay-reply to the source address from the IP datagram in which the original message was received.

17. 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 (system or application).

17.1. Server Behavior

A server sends a Reconfigure-init message to cause a client to initiate immediately a Request/Reply message exchange with the server.

17.1.1. Creation and transmission of Reconfigure-init messages

The server sets the "msg-type" field to RECONFIG-INIT. The server generates a transaction-ID and inserts it in the "transaction-ID" field. The server places its address (of appropriate scope) in the "server-address" field.

The server MAY include an ORO 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 ORO if the server wants the client to obtain new address information.

The server MUST include an authentication option with the appropriate settings and add that option as the last option in the "options"

field of the Reconfigure-init message.

The server MUST NOT include any other options in the Reconfigure-init except as specifically allowed in the definition of individual options.

A server sends each Reconfigure-init 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-init message to additional clients while previous reconfiguration message exchanges are still in progress.

The Reconfigure-init message causes the client to initiate a Request/Reply message exchange with the server. The server interprets the receipt of a Request message from the client as satisfying the Reconfigure-init message request.

17.1.2. Time out and retransmission of Reconfigure-init messages

If the server does not receive a Request message from the client in RECREP_MSG_TIMEOUT milliseconds, the server retransmits the Reconfigure-init message, doubles the RECREP_MSG_TIMEOUT value and waits again. The server continues this process until REC_MSG_ATTEMPTS unsuccessful attempts have been made, at which point the server SHOULD abort the reconfigure process for that client.

Default and initial values for RECREP_MSG_TIMEOUT and REC_MSG_ATTEMPTS are documented in section 7.5.

17.1.3. Receipt of Request messages

The server generates and sends Reply message(s) to the client as described in section 16.2.7, including in the "options" field new values for configuration parameters.

It is possible that the client may send a Request message after the server has sent a Reconfigure-init but before the Reconfigure-init is received by the client. In this case, the Request message from the client may not include all of the IAs and requests for parameters

to be reconfigured by the server. To accommodate this scenario, 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 in the Request message from the client.

17.2. Client Behavior

A client MUST always monitor UDP port 546 for Reconfigure-init messages on interfaces upon which it has acquired DHCP parameters. 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.

17.2.1. Receipt of Reconfigure-init messages

Upon receipt of a valid Reconfigure-init message, the client initiates a Request/Reply transaction with the server. While the Request/Reply transaction is in progress, the client silently discards any Reconfigure-init messages it receives.

DISCUSSION:

> The Reconfigure-init message acts as a trigger that signals the client to complete a successful Request/Reply message exchange. Once the client has received a Reconfigure-init, the client proceeds with the Request/Reply message exchange (retransmitting the Request if necessary); the client ignores any additional Reconfigure-init messages (regardless of the transaction ID in the Reconfigure-init message) until the Request/Reply exchange is complete. Subsequent Reconfigure-init messages (again independent of the transaction ID) cause the client to initiate a new Request/Reply exchange.
>
> How does this mechanism work in the face of duplicated or retransmitted Reconfigure-init messages? Duplicate messages will be ignored because the client will begin the Request/Reply exchange after the receipt of the first Reconfigure-init. Retransmitted messages will either trigger the Request/Reply exchange (if the first

Reconfigure-init was not received by the client) or will be ignored. The server can discontinue retransmission of Reconfigure-init messages to the client once the server receives the Request from the client.

It might be possible for a duplicate or retransmitted Reconfigure-init to be sufficiently delayed (and delivered out of order) to arrive at the client after the Request/Reply exchange (initiated by the original Reconfigure-init) has been completed. In this case, the

client would initiate a redundant Request/Reply 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.

17.2.2. Creation and sending of Request messages

When responding to a Reconfigure-init, the client creates and sends the Request message in exactly the same manner as outlined in section 16.1.1 with the following difference:

IAs The client includes IA options containing the addresses the client currently has assigned to those IAs for the interface through which the Reconfigure-init message was received.

17.2.3. Time out and retransmission of Request messages

The client uses the same variables and retransmission algorithm as it does with Request messages generated as part of a client-initiated configuration exchange. See section 16.1.1 for details.

17.2.4. 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. If the configuration parameters changed were requested by the

application layer, the client notifies the application layer of the changes using an implementation-specific interface.

As discussed in section 17.1.3, the Reply from the server may include IAs and parameters that were not included in the Request message from the client. The client MUST configure itself with all of the IAs and parameters in the Reply from the server.

18. Relay Behavior

For this discussion, the Relay 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 has not been explicitly configured, it MUST use the All_DHCP_Servers multicast address as the default.

18.1. Relaying of client messages

When a Relay receives a valid client message, it constructs a Relay-forward message. The relay places an address with a prefix assigned to the link on which the client should be assigned an address in the link-prefix 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 cannot use the address in the link-prefix field to identify the interface through which the response to the client will be forwarded, the relay MUST include a circuit-id option (see section 20.15)in the Relay-forward message. The server will include the circuit-id option in its Relay-reply message.

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

The relay constructs a "client-message" option 20.7 that contains the entire message from the client in the data field of the option. The relay places the "relay-message" option along with any "relay-specific" options in the options field of the Relay-forward message. The Relay then sends the Relay-forward message to the list of server destination addresses that it has been configured with.

18.2. Relaying of server messages

When the relay receives a Relay-reply message, it extracts the server message from the "server-message" option. If the Relay-reply message includes a circuit-id option, the relay forwards the message from the server to the client on the link identified by the circuit-id option. Otherwise, the relay forwards the message on the link identified by the link-prefix option. In either case, the relay forwards the message to the address in the client-return-address field in the Relay-reply message.

19. 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.

Because of the risk of denial of service attacks against DHCP clients, the use of authentication is mandated in Reconfigure-init

messages. A DHCP server MUST include an authentication option in Reconfigure-init messages sent to clients.

The DHCP authentication mechanism is based on the design of authentication for DHCP for IPv4 [8].

19.1. 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. In current practice, redundancy mitigates DoS attacks the best.

19.2. 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 [10]. The way in which IPsec is employed by relay agents and servers is not specified in this document.

19.3. Summary of DHCP authentication

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

The Authentication option provides a framework for multiple authentication protocols. Two such protocols are defined here. Other protocols defined in the future will be specified in separate documents.

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.

19.4. 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 (e.g., an NTP-format timestamp [12]) can reduce the danger of replay attacks. This method MUST be supported by all protocols.

19.5. Configuration token protocol

If the protocol field is 0, the authentication information field holds a simple configuration token. The configuration token is an opaque, unencoded value known to both the sender and receiver. The sender inserts the configuration token in the DHCP message and the receiver matches the token from the message to the shared token. If the configuration option is present and the token from the message does not match the shared token, the receiver MUST discard the message.

Configuration token may be used to pass a plain-text configuration token and provides only weak entity authentication and no message authentication. This protocol is only useful for rudimentary protection against inadvertently instantiated DHCP servers.

DISCUSSION:

> The intent here is to pass a constant, non-computed token such as a plain-text password. Other types of entity

authentication using computed tokens such as Kerberos tickets or one-time passwords will be defined as separate protocols.

19.6. 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 [11] using the MD5 hash [19] is defined here.

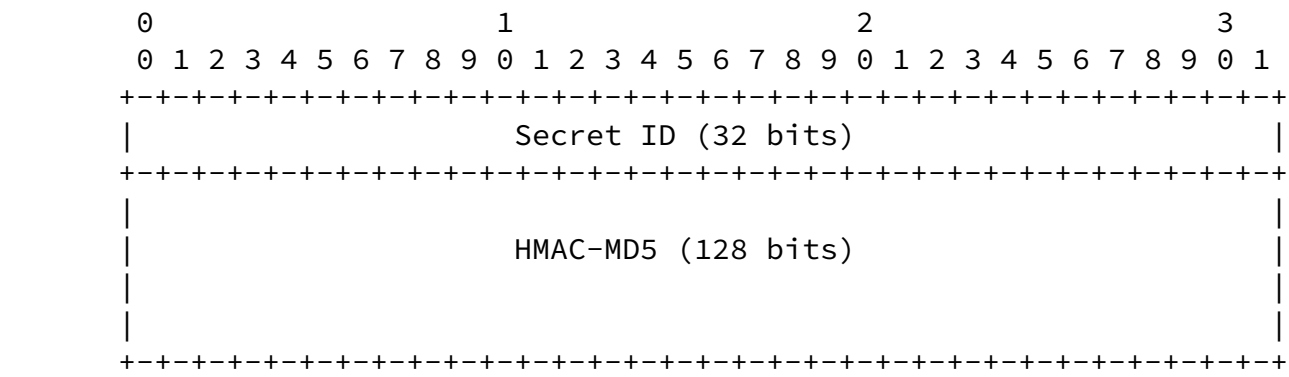
19.6.1. Management issues in the delayed authentication protocol

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 secret is feasible.

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

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

In an Advertise, Request, Renew, Rebind or Confirm message, the Authentication option carries the Protocol, Algorithm, RDM and Replay detection fields and Authentication information. 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 secret value shared between the source and destination of the message; each secret has a unique identifier (secret ID)
secret ID - the unique identifier for the secret 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 [11] and the MD5 hash function [19]. The entire DHCP message (except the MAC field of the authentication option itself), including the DHCP message header and the options field, is used as input to the HMAC-MD5 computation function. The 'secret ID' field MUST be set to the identifier of the secret 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 secret key has a unique identifier that can be used by a receiver to determine which

secret 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.

19.6.3. 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 [11]. The receiver MUST set the 'MAC' field of the authentication option to all 0s for computation of the MAC. If the MAC computed by the receiver does not match the MAC contained in the authentication option, the receiver MUST discard the DHCP message.

19.6.4. Key utilization

Each DHCP client has a key, K. 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 SHOULD be initially distributed to the client through some out-of-band mechanism, and SHOULD be stored locally on the client for use in all authenticated DHCP messages. Once the client has been given its key, it SHOULD use that key for all transactions even if the client's configuration changes; e.g., if the client is assigned a new network address.

Each DHCP server MUST know, 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. To authenticate the identity of individual clients, each client MUST be configured with a unique key.

19.6.5. Client considerations for delayed authentication protocol

19.6.5.1. 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, RDM and replay detection field as described in section 19.6. The client does not include any authentication information in the Authentication option.

19.6.5.2. 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 19.6.3.

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. 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.

19.6.5.3. Sending Request, Confirm, Renew, Rebind 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 19.6. When the client sends a subsequent message, it MUST use the same secret used by the server to generate the authentication information.

19.6.5.4. 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. The client MAY choose to remember which server replied with a Reply message that failed to pass validation and discard subsequent messages from that server.

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.

19.6.6. Server considerations for delayed authentication protocol

19.6.6.1. Receiving Solicit messages and Sending Advertise messages

The server selects a secret for the client and includes authentication information in the Advertise message returned to the client as specified in section 19.6. The server MUST record the identifier of the secret selected for the client and use that same secret for validating subsequent messages with the client.

19.6.6.2. Receiving Request, Confirm, Renew, Rebind or Release messages and Sending Reply messages

The server uses the secret identified in the message and validates the message as specified in section 19.6.3. If the message fails to pass validation or the server does not know the secret identified by the 'secret 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 16.2. The server MUST include authentication information generated using the secret identified in the received message as specified in section 19.6.

19.6.6.3. Sending Reconfigure-Init messages

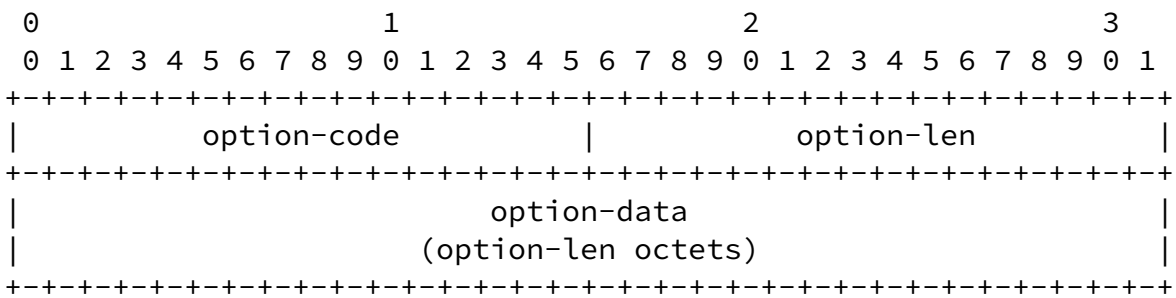
The server MUST include authentication information in a Reconfigure-Init message, generated as specified in section 19.6 using the secret the server initially selected for the client to which the Reconfigure-Init message is to be sent.

20. 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 20.1.

This document describes the DHCP options defined as part of the base DHCP specification. Other options may be defined in the future in a separate document.

20.1. Format of DHCP options



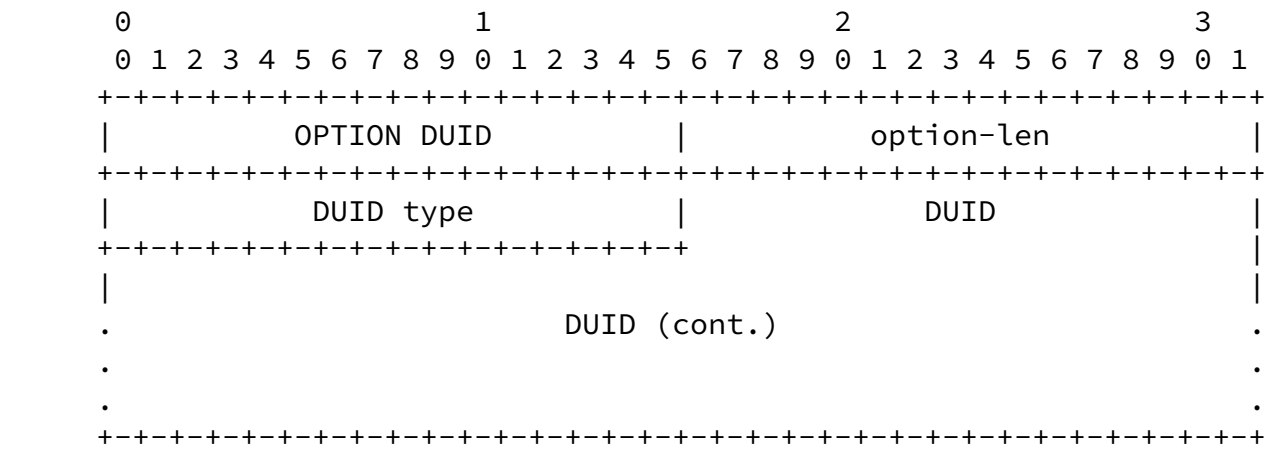
option-code An unsigned integer identifying the specific option type carried in this option.

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

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

20.2. DHCP unique identifier option

The DHCP unique identifier option is used to carry a DUID. The format for the DUID is keyed to mark the type of identifier and is of variable length. The format of the DUID option is:

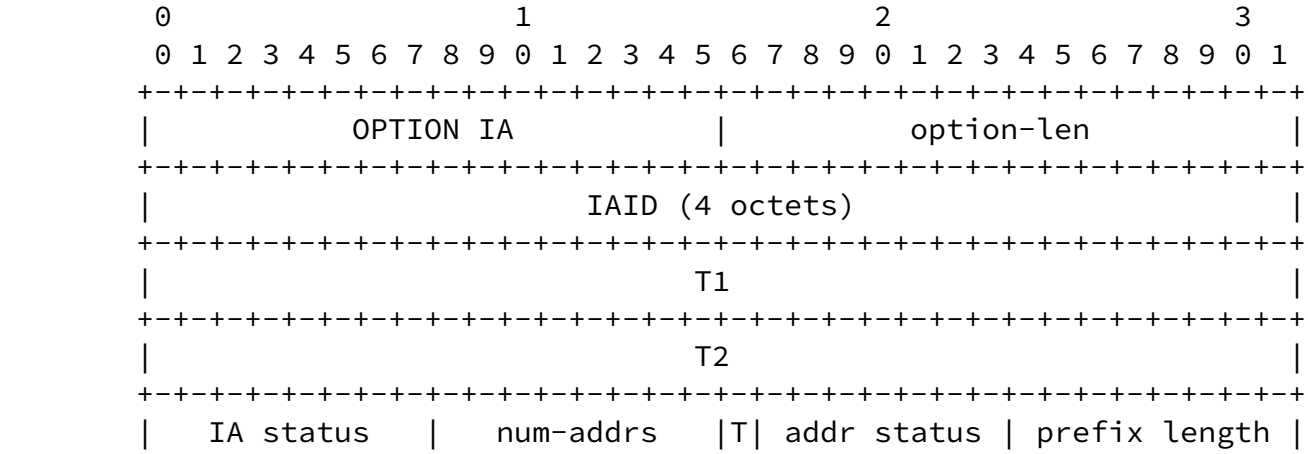


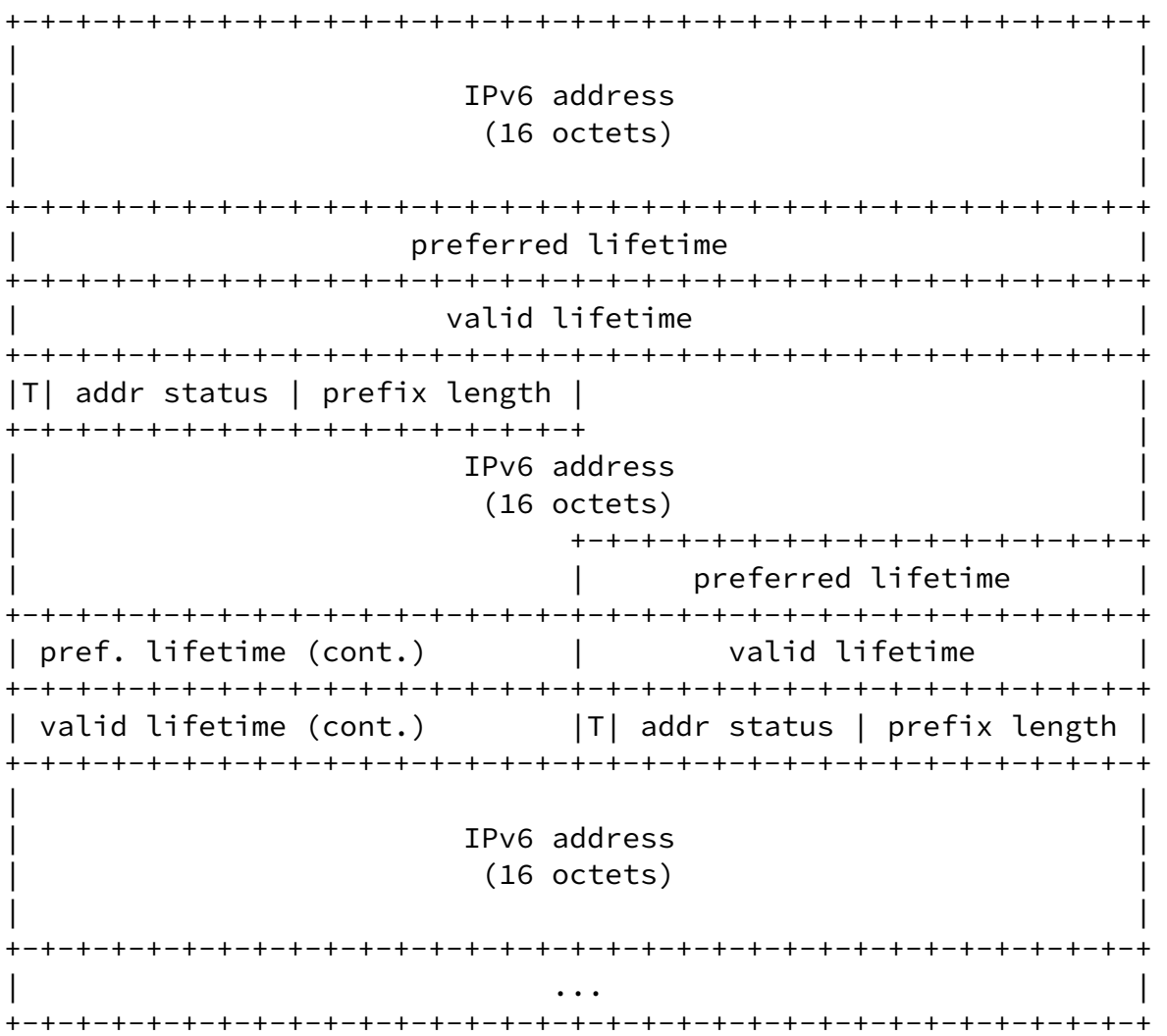
20.3. Identity association option

The identity association option is used to carry an identity association, the parameters associated with the IA and the addresses

assigned to the IA.

The format of the IA option is:





option-code	OPTION_IA (TBD)
option-len	Variable; equal to 24 + num-addrs*26
IA ID	The unique identifier for this IA; chosen by the client

T1	The time at which the client contacts the server from which the addresses in the IA

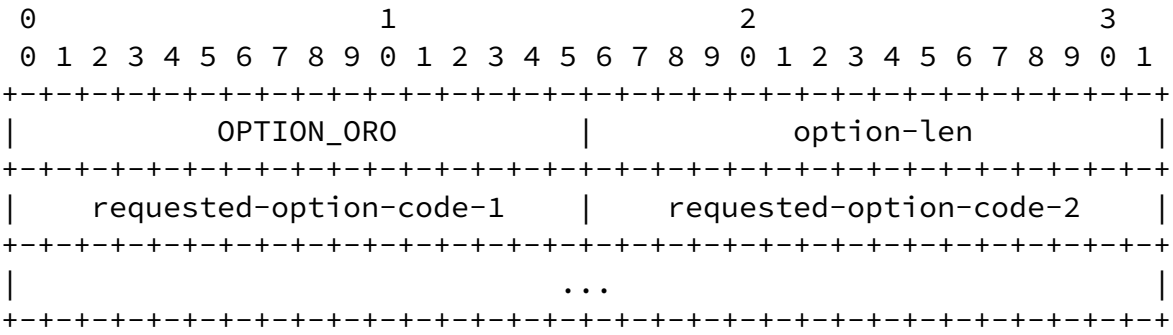
	were obtained to extend the lifetimes of the addresses assigned to the IA.
T2	The time at which the client contacts any available server to extend the lifetimes of the addresses assigned to the IA.
T	When set to 1, indicates that this address is a "temporary address" [15]; when set to 0, the address is not a temporary address.
IA status	Status of the IA in this option.
num-addrs	An unsigned integer giving the number of addresses carried in this IA option (MAY be zero).
addr status	Status of the addresses in this IA.
prefix length	Prefix length for this address.
IPv6 address	An IPv6 address assigned to this IA.
preferred lifetime	The preferred lifetime for the associated IPv6 address.
valid lifetime	The valid lifetime for the associated IPv6 address.

The "IPv6 address", "preferred lifetime" and "valid lifetime" fields are repeated for each address in the IA option (as determined by the "num-addrs" field).

Note that an IA has no explicit "lifetime" or "lease length" of its own. When the 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.

The 'T' bit identifies the associated address as a temporary address. If the server is configured to assign temporary addresses to the client, the server marks those temporary addresses with the 'T' bit. The number of temporary addresses assigned to the client and the lifetimes of those addresses is determined by the administrative configuration of the server. The 'T' bit only identifies an address as a temporary address; identification of an address as "temporary" has no implication on the lifetime of the extensibility of the lifetime of the address.

20.4. Option request option



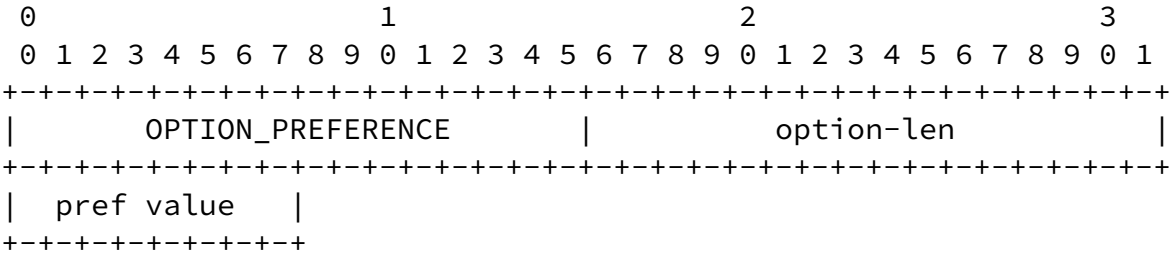
option-code OPTION_ORO (TBD)

option-len Variable; equal to twice the number of option codes carried in this option.

option-data A list of the option codes for the options requested in this option.

A client MAY include an Option Request option in a Solicit, Request, Renew, Rebind or Confirm message to inform the server about options the client wants the server to send to the client.

20.5. Preference option



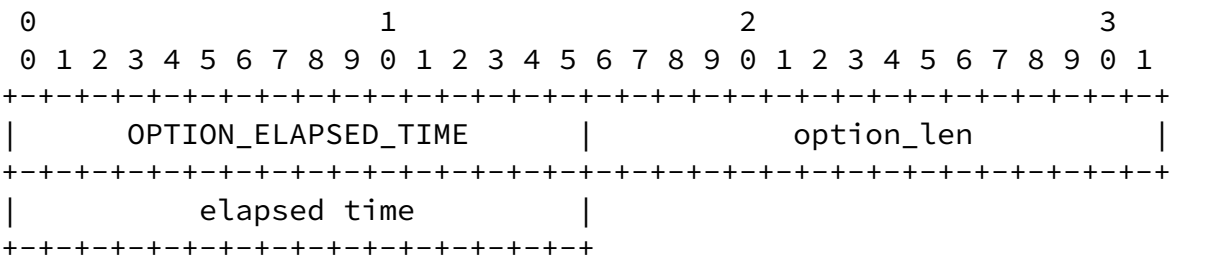
option-code OPTION_PREFERENCE (TBD)

option-len MUST be 1

option-data 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 15.1.3 for the use of the Preference option by the client and the interpretation of Preference option data value.

20.6. Elapsed Time



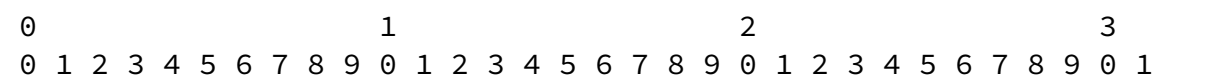
option-code OPTION_ELAPSED_TIME (TBD)

option-len MUST be 2

option-data 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 MAY include an Elapsed Time option in messages to indicate how long the client has been trying to complete a DHCP transaction. Servers MAY use the data value in this option as input to policy controlling how a server responds to a client message.

20.7. Client message option

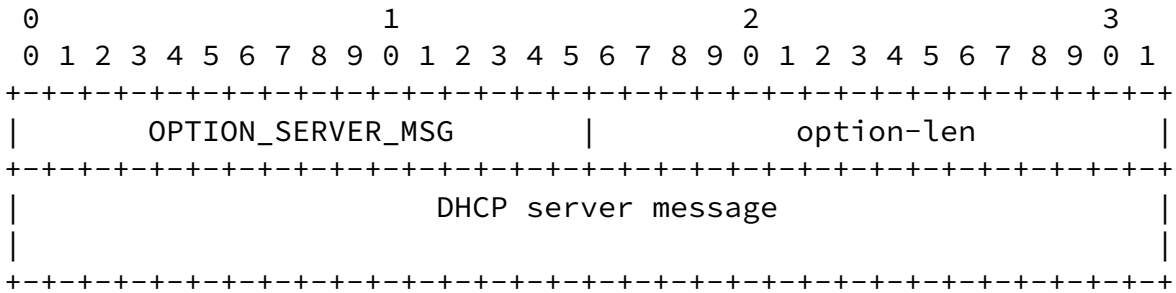


option-code OPTION_CLIENT_MSG (TBD)

option-len Variable; equal to the length of the forwarded DHCP client message.

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

20.8. Server message option



option-code OPTION_SERVER_MSG (TBD)

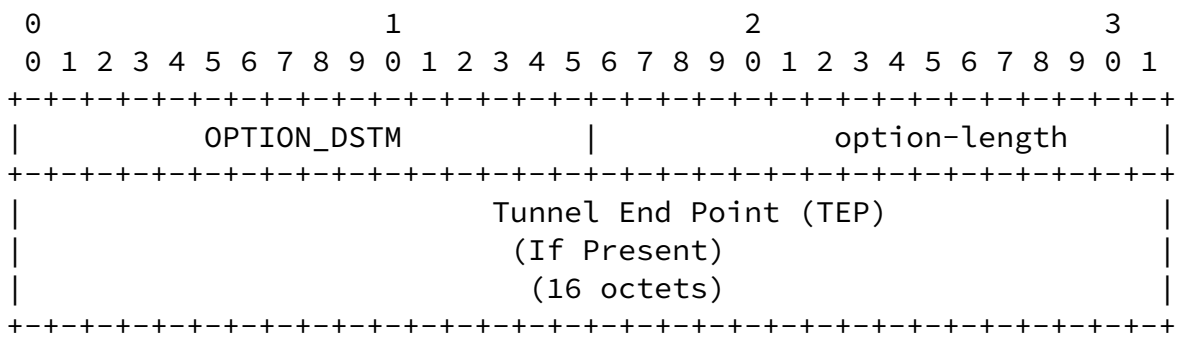
option-len Variable; equal to the length of the forwarded DHCP server message.

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

20.9. DSTM Global IPv4 Address Option

The DSTM Global IPv4 Address Option informs a client or server that the Identity Association Option (IA) following this option will contain an IPv4-Mapped IPv6 Address [9] in the case of a Client receiving the option, or is a Request for an IPv4-Mapped IPv6 Address from a client in the case of a DHCPv6 Server receiving the option. The option can also provide a set of IPv6 addresses to be used as the Tunnel Endpoint (TEP) to encapsulate an IPv6 packet within IPv6.

This option can be used with the Request, Reply, and Reconfigure-Init Messages for cases where a server wants to assign to clients IPv4-Mapped IPv6 Addresses, thru the Option Request Option (ORO).



option code OPTION_DSTM (TBD)

option length Variable: 0 or multiple of 16

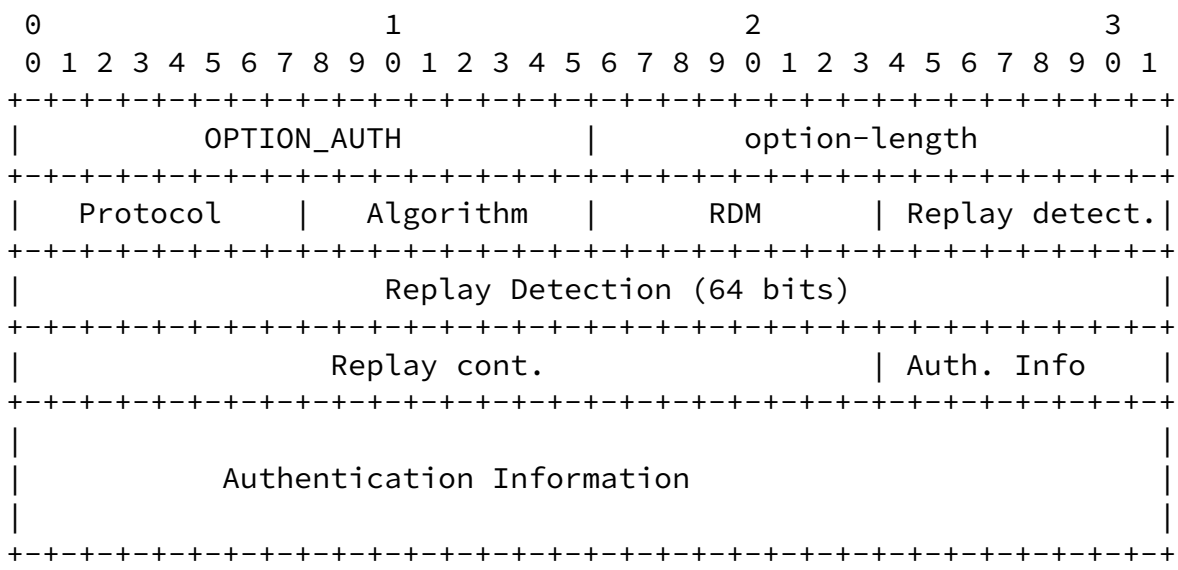
tunnel end point IPv6 Address or addresses if Present

A DSTM IPv4 Global Address Option MUST only apply to the IA following this option.

20.10. 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 19.

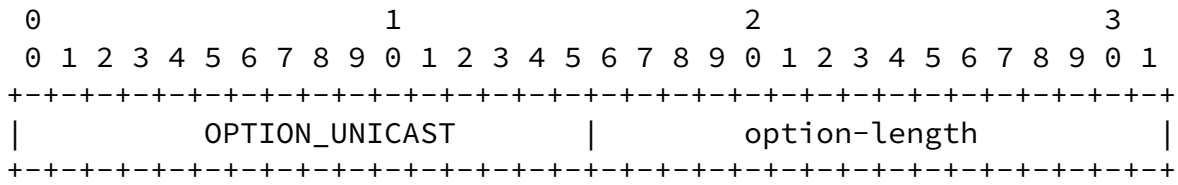
The format of the Authentication option is:



option-code	OPTION_AUTH (TBD)
option-length	Variable
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

20.11. Server unicast option

This option is used by a server to send to a client to inform the client it MAY send a Request, Renew, Release, and Decline by unicasting directly to the server instead of the All_DHCPv6_Agents Multicast address as an optimization, when the client as an address of sufficient scope to reach the server.



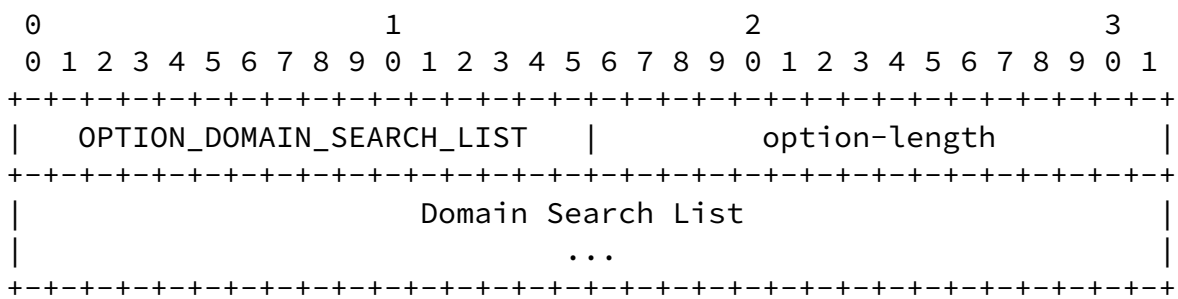
option-code OPTION_UNICAST (TBD)

option-length 0

This option only applies to the server address that sends this to the client.

20.12. Domain Search Option

This option provides a list of domain names a client can use to resolve DNS names.



option-code OPTION_DOMAIN_SEARCH_LIST (TBD)

option-length variable

Domain Search List The DNS domain search list the client

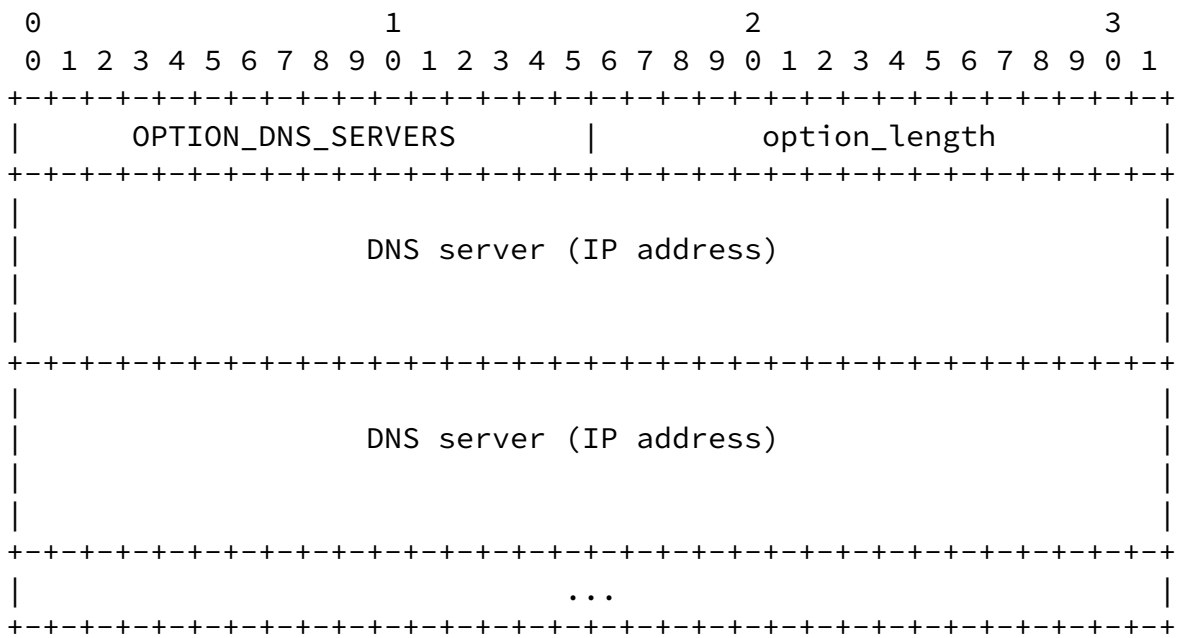
should use to resolve names.

So that the search list may be encoded compactly and uniformly, search strings in the search list are concatenated and encoded using the technique described in section 4.1 of [13].

For use in this specification, the compression pointer (see section 4.1.4 of [13]) refers to the offset within the SearchString portion of the option.

20.13. Domain Name Server Option

This option provides a list of Domain Name System [13] that a client name resolver can use to access DNS services. There must be at least 1 server listed in this option.



option-code OPTION_DNS_SERVERS (11)

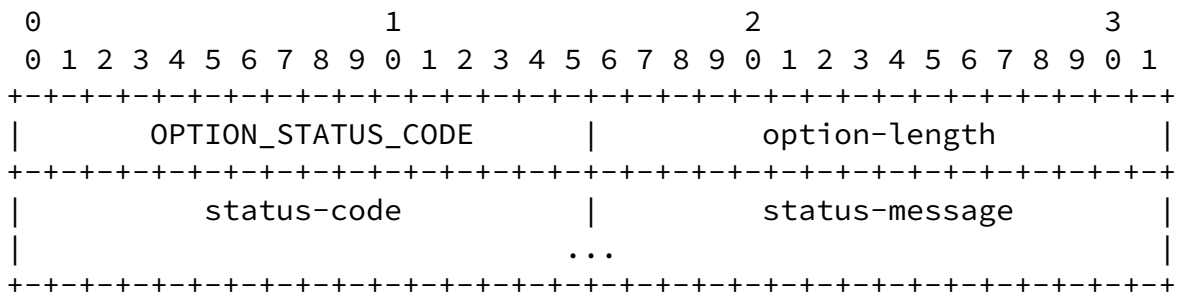
option-length variable

DNS server IPv6 address of a DNS name server for the

client to use. The DNS servers are listed in the order of preference for use by the client resolver.

20.14. Status Code Option

This option returns indications of status not related to a specific option.



option-code OPTION_STATUS_CODE (TBD)

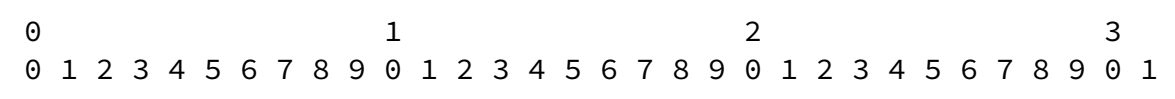
option-length variable

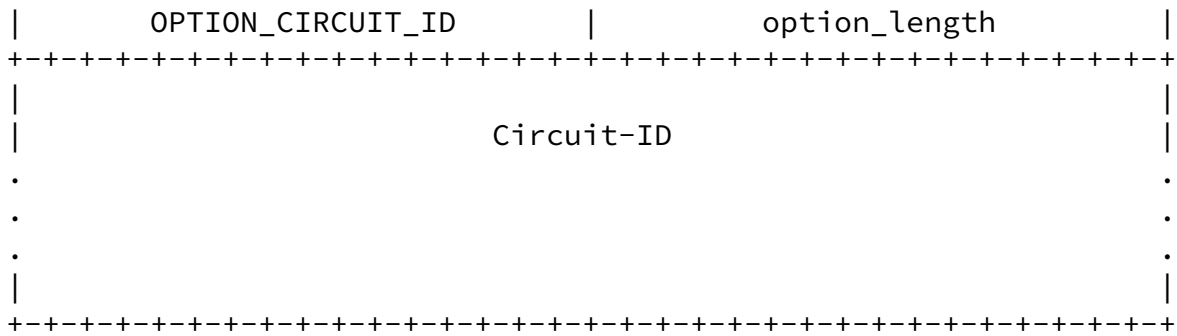
status-code The numeric code for the status encoded in this option. The status codes are defined in section 7.4.

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

20.15. Circuit-ID Option

This option provides a mechanism through which a relay agent can identify the network attachment point through which a message was received from a DHCP client.



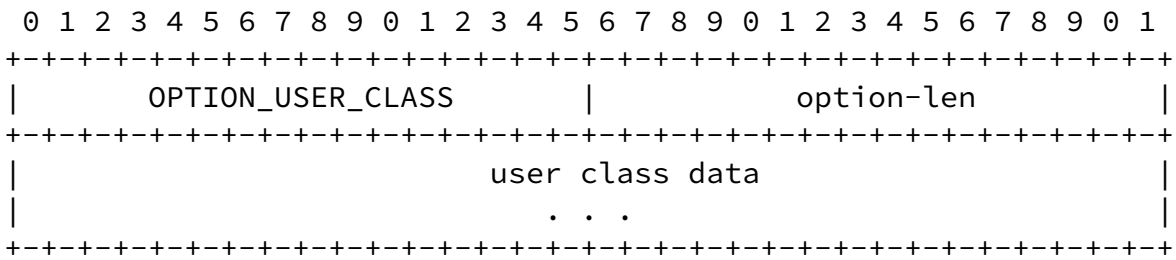


option-code	OPTION_CIRCUIT_ID (TBD)	
option-length	variable	
Circuit-ID	An opaque value of arbitrary length; this value must uniquely identify one of the network attachments used by the relay agent	

20.16. User Class Option

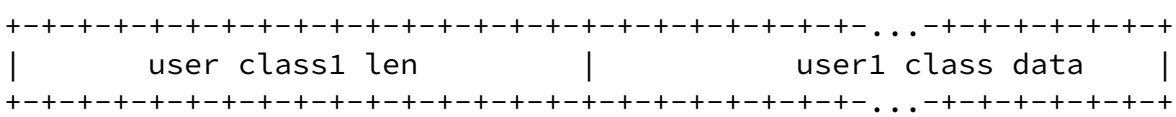
This option is used by a client to identify the type or category of user or applications it represents. The information contained in this option is an opaque field that represents the user class of which the client is a member. Based on this class, a DHCP server selects the appropriate address pool to assign an address to the

client and the appropriate configuration parameters.



option-code	TBD
option-len	Variable; If n user classes are carried by the option, the length of the option option-len = sum of each of the user class lengths + 2*n.
option-data	The user classes carried by the client.

The user class option may contain one or more instances of user class data. Each instance of the user class data is formatted as follows:



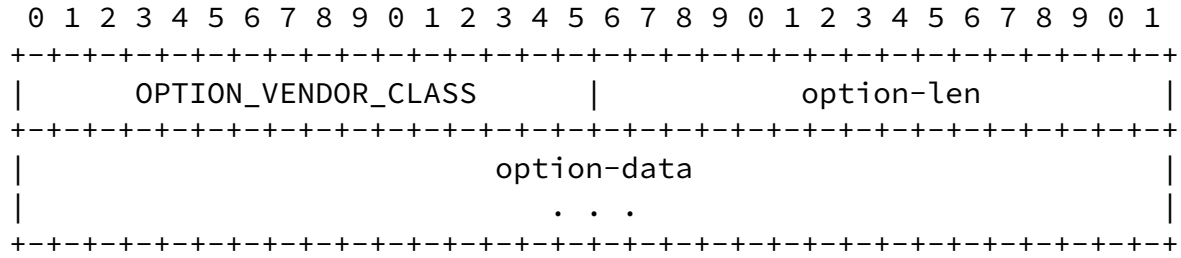
The user class length is two octets long and specifies the length of the opaque user class data in network byte order.

Servers may interpret the meanings of multiple class specifications in an implementation dependent or configuration dependent manner, and so the use of multiple classes by a DHCP client should be based on the specific server implementation and configuration which will be used to process that User class option. Servers not equipped to interpret the user class information sent by a client MUST ignore it (although it may be reported).

20.17. Vendor Class 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 vendor class identifier option. Servers not equipped to interpret the vendor-specific

information sent by a client MUST ignore it (although it may be reported). Clients which 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.

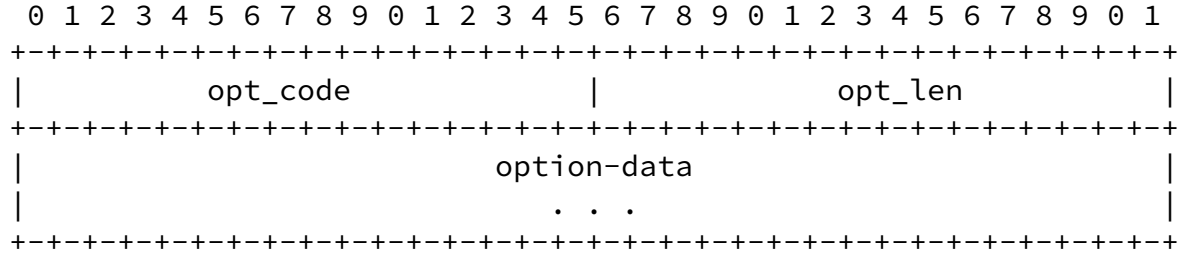


option-code	TBD
option-len	Variable
option-data	The information is an opaque object of option-len octets, presumably interpreted by vendor-specific code on the clients and servers

If a vendor potentially encodes more than one item of information in this option, then the vendor SHOULD encode the option using "Encapsulated vendor-specific options".

The Encapsulated vendor-specific options field SHOULD be encoded as a sequence of code/length/value fields of identical syntax to the DHCP options field.

When encapsulated vendor-specific extensions are used, each of the encapsulated options is formatted as follows.



opt_code	The code for the encapsulated option
opt_len	The length of the encapsulated option

option-data The data area for the encapsulated option

21. Security Considerations

Section 19 describes a threat model and an option that provides an authentication framework to defend against that threat model.

22. Year 2000 considerations

Since all times are relative to the current time of the transaction, there is no problem within the DHCPv6 protocol related to any hardcoded dates or two-digit representation of the current year.

23. IANA Considerations

This document defines several new name spaces associated with DHCPv6 and DHCPv6 options. IANA is requested to manage the allocation of values from 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 [7, 2].

New values in each of these name spaces should be approved by the process of IETF consensus [14].

23.1. Multicast addresses

Section 7.1 defines the following multicast addresses, which have been assigned by IANA for use by DHCPv6:

All_DHCP_Agents address: FF02::1:2

All_DHCP_Servers address: FF05::1:3

IANA is requested to manage definition of additional multicast

addresses in the future.

23.2. DHCPv6 message types

IANA is requested to record the message types defined in section 7.3. IANA is requested to manage definition of additional message types in the future.

23.3. DUID

IANA is requested to record the DUID types defined in section 10.1. IANA is requested to manage definition of additional DUID types in the future.

23.4. DHCPv6 options

IANA is requested to assign option-codes to the options defined in section 20.1. IANA is requested to manage the definition of additional DHCPv6 option-codes in the future.

23.5. Status codes

IANA is requested to record the status codes defined in section 7.4. IANA is requested to manage the definition of additional status codes in the future.

23.6. Authentication option

Section 19 defines three new name spaces associated with the Authentication Option (section 20.10), which are to be created and maintained by IANA: Protocol, Algorithm and RDM.

Initial values assigned from the Protocol name space are 0 (for the configuration token Protocol in section 19.5) and 1 (for the delayed authentication Protocol in section 19.6). Additional protocols may

be defined in the future.

The Algorithm name space is specific to individual Protocols. That is, each Protocol has its own Algorithm name space. The guidelines for assigning Algorithm name space values for a particular protocol should be specified along with the definition of a new Protocol.

For the configuration token Protocol, the Algorithm field MUST be 0, as described in section 19.5. For the delayed authentication Protocol, the Algorithm value 1 is assigned to the HMAC-MD5 generating function as defined in section 19.6. Additional algorithms for the delayed authentication protocol may be defined in the future.

The initial value of 0 from the RDM name space is assigned to the use of a monotonically increasing value as defined in section 19.4. Additional replay detection methods may be defined in the future.

24. Acknowledgments

Thanks to the DHC Working Group for their time and input into the specification. Ralph Droms and Thomas Narten have had a major role in shaping the continued improvement of the protocol by their careful reviews. Many thanks to Matt Crawford, Erik Nordmark, Gerald Maguire, and Mike Carney for their studied review as part of the Last Call process. Thanks also for the consistent input, ideas, and review by (in alphabetical order) Brian Carpenter, Francis DuPont, Ted Lemon, Jack McCann, Yakov Rekhter, Matt Thomas, Sue Thomson, Bernie Volz 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.

Bill Arbaugh reviewed the authentication mechanism described in section 19.

The Domain Search option described in section 20.12 is based on the DHCPv4 domain search option, [1], and was reviewed by Bernard Aboba.

A. Comparison between DHCPv4 and DHCPv6

This appendix is provided for readers who will find it useful to see a model and architecture comparison between DHCPv4 [7, 2] and DHCPv6. There are three key reasons for the differences:

- IPv6 inherently supports a new model and architecture for communications and autoconfiguration of addresses.
- DHCPv6 benefits from the new IPv6 features.
- New features were added to support the expected evolution and the existence of more complicated Internet network service requirements.

IPv6 Architecture/Model Changes:

- The link-local address permits a node to have an address immediately when the node boots, which means all clients have a source IP address at all times to locate an on-link server or relay.
- The need for BOOTP compatibility and the broadcast flag have been removed.
- Multicast and address scoping in IPv6 permit the design of discovery packets that would inherently define their range by the multicast address for the function required.
- Stateful autoconfiguration has to coexist and integrate with stateless address autoconfiguration supporting duplicate address detection [20] and the two IPv6 address lifetimes, to facilitate the dynamic renumbering of addresses and the management of those addresses.
- Multiple addresses per interface are inherently supported in IPv6.
- Some DHCPv4 options are unnecessary now because the configuration parameters are either obtained through IPv6 Neighbor Discovery or the Service Location protocol [21].

DHCPv6 Architecture/Model Changes:

- The message type is the first octet in the packet.
- IPv6 Address allocations are now handled in a message option as opposed to the message header.
- Client/Server bindings are now mandatory and take advantage of the link-local address of the client to always permit communications either directly from an on-link server, or from a off-link server through an on-link relay.
- Servers are discovered by a client Solicit, followed by a server Advertise message
- The client will know if the server is on-link or off-link.
- The on-link relay may locate off-link server addresses from system configuration or by the use of a site-wide multicast packet.
- ACKs and NAKs are not used.
- The server assumes the client receives its responses unless it receives a retransmission of the same client request. This permits recovery in the case where the network has faulted.
- Clients can issue multiple, unrelated Request messages to the same or different servers.
- The function of DHCPINFORM is inherent in the new packet design; a client can request configuration parameters other than IPv6 addresses in the optional option headers.
- Clients MUST listen to their UDP port for the new Reconfigure-init message from servers.
- New options have been defined.

With the changes just enumerated, we can support new user features, including

- Configuration of Dynamic Updates to DNS
- Address deprecation, for dynamic renumbering.
- Relays can be preconfigured with server addresses, or use of multicast.
- Authentication

- Clients can ask for multiple IP addresses.

- Addresses can be reclaimed using the Reconfigure-init message.
- Integration between stateless and stateful address autoconfiguration.
- Enabling relays to locate off-link servers.

B. Full Copyright Statement

References

[1] B. Aboba. DHCP Domain Search Option. Internet Draft, Internet Engineering Task Force, December 2000. Work in progress.

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

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

[4] S. Bradner and A. Mankin. The Recommendation for the IP Next Generation Protocol, January 1995. RFC 1752.

[5] W.J. Croft and J. Gilmore. Bootstrap Protocol, September 1985. RFC 951.

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

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

[8] R. Droms and W. Arbaugh. Authentication for DHCP Messages. Internet Draft, Internet Engineering Task Force, January 2001. Work in progress.

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

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

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

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

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

[14] T. Narten and H. Alvestrand. Guidelines for Writing an IANA

Considerations Section in RFCs, October 1998. RFC 2434.

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

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

[17] 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.

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

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

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

[21] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan. Service Location Protocol, June 1997. RFC 2165.

[22] 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

Compaq Computer Corporation
ZK3-3/W20
110 Spit Brook Road
Nashua, NH 03062-2698
USA
Phone: +1 603 884 0062
Email: Jim.Bound@compaq.com

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

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

Ralph Droms
Cisco Systems
300 Apollo Drive
Chelmsford, MA 01824
USA
Phone: +1 978 244 4733
Email: rdroms@cisco.com