Internet Engineering Task Force
INTERNET DRAFT
DHC Working Group
Obsoletes: draft-ietf-dhc-dhcpv6-18.txt

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

Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
draft-ietf-dhc-dhcpv6-18.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" [13], and can be used separately or concurrently with the latter to obtain configuration parameters.

Contents

Status of This Memo i

Abstract i

1. Introduction 1

2. Requirements 1

3. Background 1

4. Design Goals 3

5. Non-Goals 3

6. Terminology 4
6.1. IPv6 Terminology 4
6.2. DHCP Terminology 5

7. DHCP Constants 6
7.1. Multicast Addresses 7
7.2. UDP ports 7
7.3. DHCP message types 7
7.4. Error Values 9
7.4.1. Generic Error Values 9
7.4.2. Server-specific Error Values 9
7.5. Configuration Variables 10

8. Overview 10
8.1. How does a node know to use DHCP? 10
8.2. What if the client and server(s) are on different links? 10
8.3. How does a client request configuration parameters from servers? 11
8.4. How do clients and servers identify and manage addresses? 12
8.5. Can a client release its assigned addresses before the lease expires? 12
8.6. What if the client determines one or more of its assigned addresses are already being used by another client? . 12
8.7. How are clients notified of server configuration changes? 12

9. Message Formats 13
9.1. DHCP Solicit Message Format 13
9.2. DHCP Advertise Message Format 14
9.3. DHCP Request Message Format 14
9.4. DHCP Confirm Message Format 15

9.5. DHCP Renew Message Format 15
9.6. DHCP Rebind Message Format 15
9.7. DHCP Reply Message Format 16
9.8. DHCP Release Message Format 16

9.9. DHCP Decline Message Format 17
9.10. DHCP Reconfigure-init Message Format 17

10. Relay messages 17
10.1. Relay-forward message 18
10.2. Relay-reply message 18

11. Identity association 19

12. DHCP Server Solicitation 19
12.1. Solicit Message Validation 19
12.2. Advertise Message Validation 19
12.3. Client Behavior 19
12.3.1. Creation and sending of the Solicit message . . . 20
12.3.2. Time out and retransmission of Solicit Messages . 20
12.3.3. Receipt of Advertise messages 20
12.4. Server Behavior 21
12.4.1. Receipt of Solicit messages 21
12.4.2. Creation and sending of Advertise messages . . . 21

13. DHCP Client-Initiated Configuration Exchange 22
13.1. Client Message Validation 23
13.2. Server Message Validation 23
13.3. Client Behavior 24
13.3.1. Creation and sending of Request messages 24
13.3.2. Creation and sending of Confirm messages 25
13.3.3. Creation and sending of Renew messages 26
13.3.4. Creation and sending of Rebind messages 27
13.3.5. Receipt of Reply message in response to a Reply, Confirm, Renew or Rebind message 28
13.3.6. Creation and sending of Release messages 30
13.3.7. Time out and retransmission of Release Messages . 30
13.3.8. Creation and sending of Decline messages 30
13.3.9. Time out and retransmission of Decline Messages . 31
13.3.10. Receipt of Reply message in response to a Release message 31
13.4. Server Behavior 31
13.4.1. Receipt of Request messages 32
13.4.2. Receipt of Confirm messages 32
13.4.3. Receipt of Renew messages 33
13.4.4. Receipt of Rebind messages 34
13.4.5. Receipt of Release messages 35
13.4.6. Sending of Reply messages 35

14. DHCP Server-Initiated Configuration Exchange 36
14.1. Reconfigure-init Message Validation 36
14.2. Server Behavior 36
14.2.1. Creation and sending of Reconfigure-init messages 36

14.2.2. Time out and retransmission of unicast
Reconfigure-init messages 37
14.2.3. Time out and retransmission of multicast
Reconfigure-init messages 38
14.2.4. Receipt of Request messages 38

14.3. Client Behavior 38
14.3.1. Receipt of Reconfigure-init messages 38
14.3.2. Creation and sending of Request messages 38
14.3.3. Time out and retransmission of Request messages . 39
14.3.4. Receipt of Reply messages 39

15. Relay Behavior 39
15.1. Relaying of client messages 39
15.2. Relaying of server messages 40

16. DHCP options 40
16.1. Format of DHCP options 40
16.2. Identity association option 41
16.3. Option request option 43
16.4. Client message option 43
16.5. Server message option 44
16.6. Retransmission parameter option 44
16.7. Reconfigure-delay option 44
16.8. DSTM Global IPv4 Address Option 45
16.9. Authentication option 46

17. DHCP Client Implementor Notes 46
17.1. Primary Interface 46
17.2. Advertise Message and Configuration Parameter Caching . . 46
17.3. Time out and retransmission variables 47
17.4. Server Preference 47

18. DHCP Server Implementor Notes 47
18.1. Client Bindings 47
18.2. Reconfigure-init Considerations 47
18.2.1. Reliable transmission of multicast Reconfigure-init messages 48
18.3. Server Preference 48
18.4. Request Message Transaction-ID Cache 48

19. DHCP Relay Implementor Notes 48

20. Open Issues for Working Group Discussion 49
20.1. Authentication 49
20.2. Identification of IAs by servers 49
20.3. DHCP-DNS interaction 49
20.4. Temporary addresses 49
20.5. Use of term "agent" 49
20.6. Client behavior when response to Rebind is not received . 49
20.7. Additional options 50
20.8. Operational parameters 50

21. Security 50

22. Year 2000 considerations 50

23. IANA Considerations 50

24. Acknowledgments 51

A. Comparison between DHCPv4 and DHCPv6 51

B. Full Copyright Statement 53

C. Changes in this draft 53
C.1. New messages for confirming addresses and extending the lease on an IA 54
C.2. New message formats 54
C.3. Renamed Server-forward message 54
C.4. Clarified relay forwarding of messages 54
C.5. Addresses and options in Advertise messages 54
C.6. Clarification of IA option format 54
C.7. Specification of transaction ID in Solicit message . . . 54
C.8. Edits to definitions 55
C.9. Relay agent messages 55
C.10. Relay agent behavior 55
C.11. Transmission of all client messages through relays . . . 55
C.12. Reconfigure-init messages 55
C.13. Ordering of sections 55
C.14. DSTM option 55
C.15. Message and option numbering 55
C.16. Inclusion of IAs in Solicit message by client 56
C.17. Clarification of destination of client messages 56
C.18. Clarification of client use of Confirm messages 56

Chair's Address 58

Author's Address 58

1. Introduction

This document describes DHCP for IPv6 (DHCP), a UDP [12] 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 Autoconfiguration" [13]. 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 [6] will find DHCP for IPv6 provides a superset of features, and benefits from the additional features of IPv6 and freedom from BOOTP [4]-backward compatibility constraints. 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 [2].

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

3. Background

Related work in IPv6 that would best serve an implementor to study is the IPv6 Specification [5], the IPv6 Addressing Architecture [7],

IPv6 Stateless Address Autoconfiguration [13], IPv6 Neighbor Discovery Processing [10], and Dynamic Updates to DNS [15]. These specifications enable DHCP to build upon the IPv6 work to provide

both robust stateful autoconfiguration and autoregistration of DNS Host Names.

The IPv6 Specification provides the base architecture and design of IPv6. A key point for DHCP implementors to understand is that IPv6 requires that every link in the Internet have an MTU of 1280 octets or greater (in IPv4 the requirement is 68 octets). This means that a UDP packet of 536 octets will always pass through an internetwork (less 40 octets for the IPv6 header), as long as there are no IP options prior to the UDP header in the packet. But, IPv6 does not support fragmentation at routers, so that fragmentation takes place end-to-end between hosts. If a DHCP implementation needs to send a packet greater than 1500 octets it can either fragment the UDP packet into fragments of 1500 octets or less, or use Path MTU Discovery [8] to determine the size of the packet that will traverse a network path.

DHCP clients use Path MTU discovery when they have an address of sufficient scope to reach the DHCP server. If a DHCP client does not have such an address, that client MUST fragment its packets if the resultant message size is greater than the minimum 1280 octets.

Path MTU Discovery for IPv6 is supported for both UDP and TCP and can cause end-to-end fragmentation when the PMTU changes for a destination.

The IPv6 Addressing Architecture specification [7] 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 [13] (Addrconf) specifies procedures by which a node may autoconfigure addresses based on router advertisements [10], 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 addrconf is a design requirement of DHCP (see Section 4).

IPv6 Neighbor Discovery [10] is the node discovery protocol in IPv6 which replaces and enhances functions of ARP [11]. To understand IPv6 and Addrconf it is strongly recommended that implementors

understand IPv6 Neighbor Discovery.

Dynamic Updates to DNS [15] 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 autoconf [13].

- 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 coexists with statically configured, non-participating nodes and with existing network protocol implementations.

- 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 [3].

- 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

6.1. IPv6 Terminology

IPv6 terminology relevant to this specification from the IPv6 Protocol [5], IPv6 Addressing Architecture [7], and IPv6 Stateless Address Autoconfiguration [13] 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 IP 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.

abort status	A status value returned to the application that has invoked a DHCP client operation, indicating anything other than success.
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 server's information about the addresses in an IA and any other configuration information assigned to the client. A binding is indexed by the tuple <prefix, DUID>, where the 'prefix' is a prefix assigned to the link to which the client is attached and 'DUID' is the DUID from the IA in the binding. DISCUSSION: The indexing of an IA by <prefix, DUID> is still under discussion.
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. DISCUSSION: Rules for choosing a DUID are TBD.
Identity association (IA)	A collection of addresses assigned to a client. Each IA has an associated DUID. An IA may have 0 or more addresses associated with it.
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 those agents' link-local address(es). 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.

DISCUSSION:

Is there a requirement for a site-scoped "All DHCP Clients" multicast address, to be used as the default in sending Reconfigure messages.

7.2. UDP ports

DHCP uses the following destination UDP [12] 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 9. Message types 0 and TBD--255 are reserved and MUST be silently ignored. The message code

for each message type is shown with the message name.

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

ADVERTISE (2) The DHCP Advertise (or Advertise) message is used by servers responding to Solicits.

REQUEST (3) The DHCP Request (or Request) message is used by clients to request configuration parameters from servers.

CONFIRM (4) The DHCP Confirm (or 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 DHCP Renew (or 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 DHCP Rebind (or 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 DHCP Reply (or 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 DHCP Release (or Release) message is used by clients to return one or more IP addresses to servers.

DECLINE (9) The DHCP Decline (or 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 (10) The DHCP Reconfigure-init (or 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 DHCP Relay-forward (or 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 DHCP Relay-reply (or 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. Error Values

This section describes error values exchanged between DHCP implementations.

7.4.1. Generic Error Values

The following symbolic names are used between client and server implementations to convey error conditions. The following table contains the actual numeric values for each name. Note that the numeric values do not start at 1, nor are they consecutive. The errors are organized in logical groups.

Error_Name	Error_ID	Description
Success	00	Success
UnspecFail	16	Failure, reason unspecified
AuthFailed	17	Authentication failed or nonexistent
PoorlyFormed	18	Poorly formed message
Unavail	19	Addresses unavailable

7.4.2. Server-specific Error Values

The following symbolic names are used by server implementations to convey error conditions to clients. The following table contains the actual numeric values for each name.

Error_Name	Error_ID	Description
NoBinding	20	Client record (binding) unavailable
ConfNoMatch	21	Client record Confirm not match IA
RenwNoMatch	22	Client record Renew not match IA
RebdNoMatch	23	Client record Rebind not match IA
InvalidSource	24	Invalid Client IP address
NoServer	25	Relay cannot find Server Address

|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. The client-specific variables MAY be configured on the server and MAY be delivered to the client through the "DHCP Retransmission Parameter Option" in a Reply message.

Parameter	Default	Description
MIN_SOL_DELAY	1	MIN (secs) to delay 1st mesg
MAX_SOL_DELAY	5	MAX (secs) to delay 1st mesg
ADV_MSG_TIMEOUT	500	SOL Retrans timer (msecs)
ADV_MSG_MAX	30	MAX timer value (secs)
SOL_MAX_ATTEMPTS	-1	MAX attempts (-1 = infinite)
REP_MSG_TIMEOUT	250	Retrans timer (msecs) for Reply
QRY_MSG_ATTEMPTS	10	MAX Request/Confirm/Renew/Rebind attempts
REL_MSG_ATTEMPTS	5	MAX Release/Decline attempts
RECREP_MSG_TIMEOUT	2000	Retrans timer (msecs)
REC_MSG_ATTEMPTS	10	Reconfigure attempts
REC_REP_MIN	5	Minimum pause interval (secs)
REC_REP_MAX	7200	Maximum pause interval (secs)
REC_THRESHOLD	100	% of required clients
SRVR_PREF_WAIT	2	Advertise Collect timer (secs)

8. Overview

This section provides a general overview of the interaction between the functional entities of DHCP. The overview is organized as a series of questions and answers. Details of DHCP such as message formats and retransmissions can be found in later sections of this document.

8.1. How does a node know to use DHCP?

An unconfigured node determines that it is to use DHCP for configuration of an interface by detecting the presence (or absence) of routers on the link. If router(s) are present, the node examines router advertisements to determine if DHCP should be used to configure the interface. If there are no routers present, then the node MUST use DHCP to configure the interface. Detail on this process can be found in neighbor discovery [10] and stateless autoconfiguration [13].

8.2. What if the client and server(s) are on different links?

Use of DHCP in such environments requires one or more DHCP relays be set up on the client's link, because a client may only have a link-local address. Relays receive messages from the client and forward them to some set of servers within the DHCP domain. The

client message is forwarded verbatim as an option in the message from the relay to the server. A relay will include one of its own addresses (of sufficient scope) from the interface on the same link as the client, as well as the prefix length of that address, in its message to the server. Servers receiving the forwarded traffic use this information to aid in selecting configuration parameters appropriate to the client's link.

Servers use relays to forward messages to clients. The message intended for the client is carried as an option in the message to the relay. The relay extracts the message from the optin and forwards it to the client. Servers use the relay's address as the destination to forward client-destined messages for final delivery by the relay.

Relays forward client messages to servers using some combination of the All DHCP Servers site-local multicast address, some other (perhaps a combination) of site-local multicast addresses set up within the DHCP domain to include the servers in that domain, or a list of unicast addresses for servers. The network administrator makes relay configuration decisions based upon the topological requirements (scope) of the DHCP domain they are managing. Note that if the DHCP domain spans more than the site-local scope, then the relays MUST be configured with global addresses for the client's link so as to be reachable by servers outside the relays' site-local environment.

8.3. How does a client request configuration parameters from servers?

To request configuration parameters, the client forms a Request message, and sends it to the server either directly (the server is on the same link as the client) or indirectly (through the on-link relay). The client MAY include a Option Request Option 16.3 (ORO) along with other options to request specific information from the server. Note that the client MAY form multiple Request messages and send each of them to different servers to request potentially different information (perhaps based upon what was advertised) in order to satisfy its needs. As a client's needs may change over time (perhaps based upon an application's requirements), the client may form additional Request messages to request additional information as it is needed.

The server(s) respond with Reply messages containing the requested configuration parameters, which can include status information regarding the information requested by the client. The Reply MAY also include additional information, such as a reconfiguration event multicast group for the client to join to monitor reconfiguration events, as described in section 8.7.

8.4. How do clients and servers identify and manage addresses?

Servers and clients manage addresses in groups called "identity associations." Each identity associations is identified using a unique identifier. An identity association may contain one or more IPv6 addresses. DHCP servers assign addresses to identity associations. DHCP clients use the addresses in an identity association to configure interfaces. There is always at least one identity association per interface that a client wishes to configure. Each address in an IA has its own preferred and valid lifetime. Over time, the server may change the characteristics of the addresses in an IA; for example, by changing the preferred or valid lifetime for an address in the IA. The server may also add or delete addresses from an IA; for example, deleting old addresses and adding new addresses to renumber a client. A client can request the current list of addresses assigned to an IA from a server through an exchange of protocol messages.

8.5. Can a client release its assigned addresses before the lease expires?

A client forms a Release message, including options identifying the IA to be released. The client sends the Release to the server which assigned the addresses to the client initially. If that server cannot be reached after a certain number of attempts (see section 7.5), the client can abandon the Release attempt. In this case, the address(es) in the IA will be reclaimed by the server(s) when the lifetimes on the addresses expire.

8.6. What if the client determines one or more of its assigned addresses are already being used by another client?

If the client determines through a mechanism like Duplicate Address Detection [13] that the address it was assigned by the server is already in use by another client, the client will form a Decline message, including the option carrying the in-use address. The option's status field MUST be set to the value reflecting the "in use" status of the address.

8.7. How are clients notified of server configuration changes?

There are two possibilities. Either the clients discover the new information when they revisit the server(s) to request additional configuration information/extend the lifetime on an address. or through a server-initiated event known as a reconfigure event.

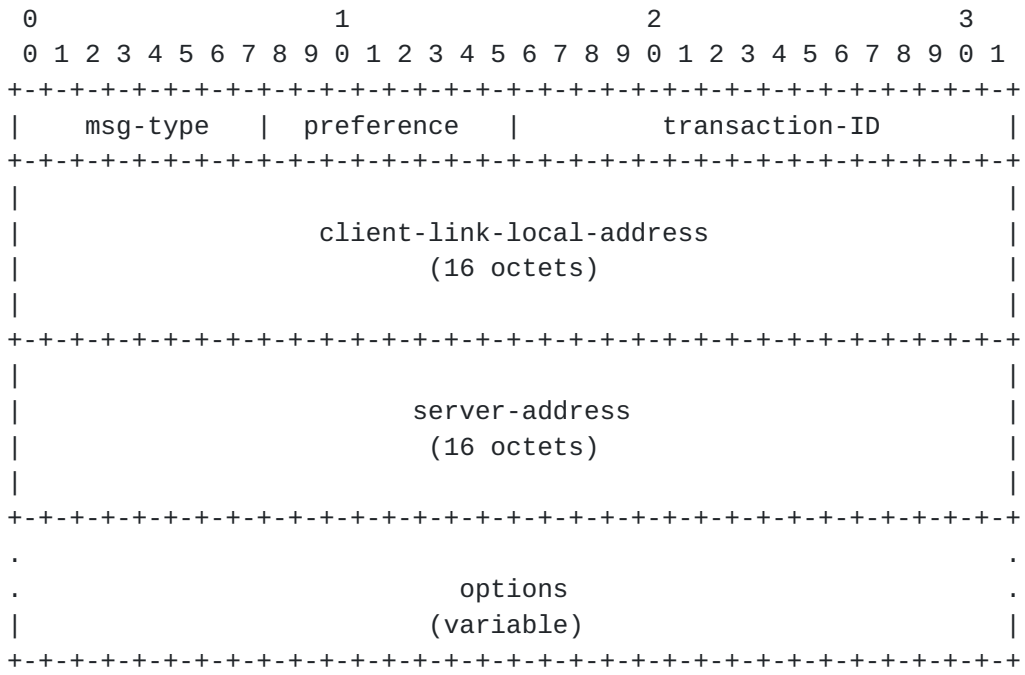
The reconfiguration feature of DHCP offers network administrators the opportunity to update configuration information on DHCP clients whenever necessary. To signal the need for client reconfiguration, the server will unicast a Reconfigure-init message to each

client individually. The server may use multicast to signal the reconfiguration to multiple clients simultaneously. (Note that there is no mechanism defined in the protocol to guarantee that every client actually performs a reconfiguration in response to a multicast reconfigure-init message.) A Reconfigure-init is a trigger which will cause the client(s) to initiate a standard Request/Reply exchange with the server in order to acquire the new or updated addresses.

9. Message Formats

Each DHCP message has an identical fixed format header; some messages also allow a variable format area for options. Not all fields in the header are used in every message. In this section, every field is described for every message and 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.

The DHCP message header:



9.1. DHCP Solicit Message Format

msg-type	SOLICIT
preference	(unused) MUST be 0

transaction-ID	An unsigned integer generated by the client used to identify this Solicit message.

client-link-local-address	The link-local address of the interface for which the client is using DHCP.
server-address	(unused) MUST be 0
options	See section 16.

9.2. DHCP Advertise Message Format

msg-type	ADVERTISE
preference	An unsigned integer indicating a server's willingness to provide service to the client.
transaction-ID	An unsigned integer used to identify this Advertise message. Copied from the client's Solicit message.
client-link-local-address	The IP link-local address of the client interface from which the client issued the Solicit message.
server-address	The IP address of the server that generated this message. If the DHCP domain crosses site boundaries, then this address MUST be globally-scoped.
options	See section 16.

9.3. DHCP Request Message Format

msg-type	REQUEST
preference	(unused) MUST be 0
transaction-ID	An unsigned integer generated by the client used to identify this Request message.
client-link-local-address	The link-local address of the client interface from which the client will issue the 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 16.

9.4. DHCP Confirm Message Format

msg-type	CONFIRM
preference	(unused) MUST be 0
transaction-ID	An unsigned integer generated by the client used to identify this Confirm message.
client-link-local-address	The link-local address of the client interface from which the client will issue the Confirm message.
server-address	MUST be zero.
options	See section 16.

9.5. DHCP Renew Message Format

msg-type	RENEW
preference	(unused) MUST be 0
transaction-ID	An unsigned integer generated by the client used to identify this Renew message.
client-link-local-address	The link-local address of the client interface from which the client will issue the 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 16.

9.6. DHCP Rebind Message Format

msg-type	REBIND
preference	(unused) MUST be 0
transaction-ID	An unsigned integer generated by the client used to identify this Rebind

message.

client-link-local-address	The link-local address of the client interface from which the client will issue the Rebind message.
server-address	MUST be zero.
options	See section 16.

9.7. DHCP Reply Message Format

msg-type	REPLY
preference	An unsigned integer indicating a server's willingness to provide service to the client.
transaction-ID	An unsigned integer used to identify this Reply message. Copied from the client's Request, Confirm, Renew or Rebind message.
client-link-local-address	The link-local address of the interface for which the client is using DHCP.
server-address	The IP address of the server. If the DHCP domain crosses site boundaries, then this address MUST be globally-scoped.
options	See section 16.

9.8. DHCP Release Message Format

msg-type	RELEASE
preference	(unused) MUST be 0
transaction-ID	An unsigned integer generated by the client used to identify this Release message.
client-link-local-address	The client's link-local address for the interface from which the client will send the Release message.
server-address	The IP address of the server that

assigned the IA.

options	See section 16.

9.9. DHCP Decline Message Format

msg-type	DECLINE
preference	(unused) MUST be 0
transaction-ID	An unsigned integer generated by the client used to identify this Decline message.
client-link-local-address	The client's link-local address for the interface from which the client will send the Decline message.
server-address	The IP address of the server that assigned the addresses.
options	See section 16.

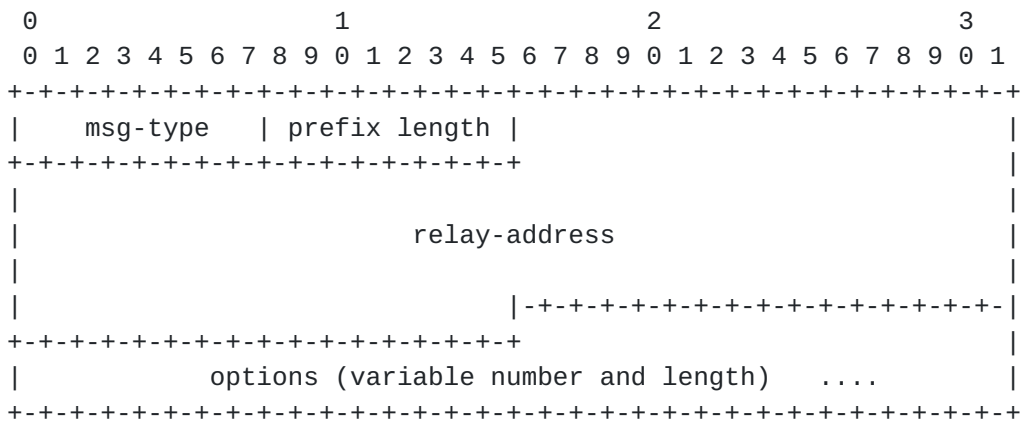
9.10. DHCP Reconfigure-init Message Format

preference	(unused) MUST be 0
transaction-ID	An unsigned integer generated by the server to identify this Reconfigure-init message
client-link-local-address	(unused) MUST be 0
server-address	The IP address of the DHCP server issuing the Reconfigure-init message. MUST be of sufficient scope to be reachable by all clients.
options	See section 16.

10. Relay messages

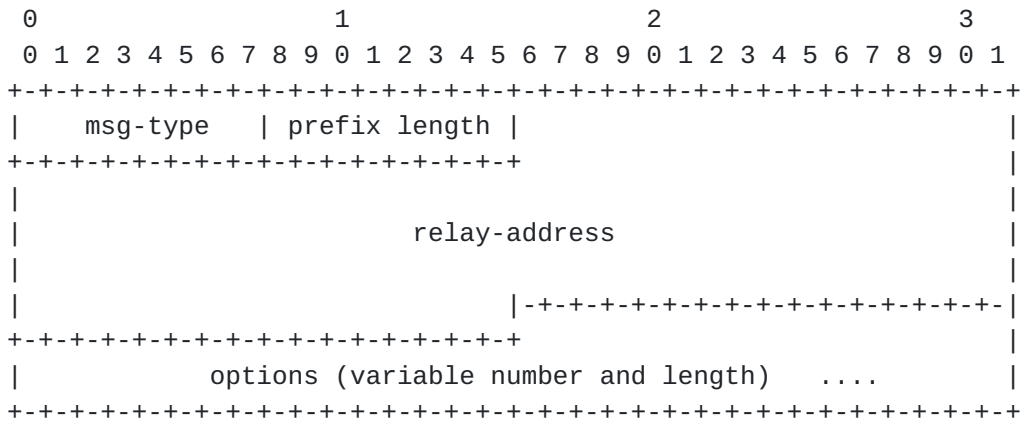
Relay agents exchange messages with servers to forward messages between clients and servers that are not connected to the same link.

10.1. Relay-forward message



msg-type	RELAY-FORW
prefix-length	The length of the prefix in the address in the "relay-address" field.
relay-address	An address assigned to the interface through which the message from the client was received.
options	MUST include a "Client message option"; see section 16.4.

10.2. Relay-reply message



msg-type	RELAY-REPL
prefix-length	The length of the prefix in the address in the "relay-address" field.

relay-address	An address identifying the interface through which the message from the server should be forwarded; copied from the "client-forward" message.

options	MUST include a "Server message option"; see section 16.5.

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 a DUID 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 16.2 for the representation of an IA in a DHCP message.

12. DHCP Server Solicitation

This section describes how a client locates servers. The behavior of client and server implementations is discussed, along with the messages they use.

12.1. Solicit Message Validation

Clients MUST silently discard any received Solicit messages.

Agents MUST silently discard any received Solicit messages if the "client-link-local-address" field does not contain a valid link-local address.

12.2. Advertise Message Validation

Servers MUST discard any received Advertise messages.

Clients MUST discard any Advertise messages that meet any of the following criteria:

- The "Transaction-ID" field value does not match the value the client used in its Solicit message.
- The "client-link-local-address" field value does not match the link-local address of the interface upon which the client sent the Solicit message.

12.3. Client Behavior

Clients use the Solicit message to discover DHCP servers configured

to serve addresses on the link to which the client is attached.

12.3.1. Creation and sending of the Solicit message

The client sets the "msg-type" field to SOLICIT, and places the link-local address of the interface it wishes to configure in the "client-link-local-address" field.

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

The client MUST include options for any IAs to which the client is expecting to have the server assign addresses. Because the client does not have any IAs with addresses when sending a Solicit message, all of the IAs MUST be empty. 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.

The client sends the Solicit message to the All DHCP Agents multicast address, destination port 547. The source port selection can be arbitrary, although it SHOULD be possible using a client configuration facility to set a specific source port value.

12.3.2. Time out and retransmission of Solicit Messages

The client's first Solicit message on the interface MUST be delayed by a random amount of time between the interval of 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 waits ADV_MSG_TIMEOUT, collecting Advertise messages. If no Advertise messages are received, the client retransmits the Solicit, and doubles the ADV_MSG_TIMEOUT value. This process continues until either one or more Advertise messages are received or ADV_MSG_TIMEOUT reaches the ADV_MSG_MAX value. Thereafter, Solicits are retransmitted every ADV_MSG_MAX until SOL_MAX_ATTEMPTS have been made, at which time the client stops trying to DHCP configure the interface. An event external to DHCP is required to restart the DHCP configuration process.

Default and initial values for MIN_SOL_DELAY, MAX_SOL_DELAY, ADV_MSG_TIMEOUT, AND ADV_MSG_MAX are documented in section 7.5.

12.3.3. Receipt of Advertise messages

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

- Those Advertise messages with the highest server preference value (see section 17.4) 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, one server may be advertising the availability of IP addresses which have an address scope of interest to the client.

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 client's invoking user, 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 server with the next highest preference value.

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.

12.4. Server Behavior

For this discussion, the server is assumed to have been configured in an implementation specific manner. This configuration is assumed to contain all network topology information for the DHCP domain, as well as any necessary authentication information.

12.4.1. Receipt of Solicit messages

If the server receives a Solicit message, the client must be on the same link as the server. If the server receives a Relay-forward message containing a Solicit message, the client must be on the link to which the prefix identified by the "relay-address" and "prefix-length" fields in the Relay-forward message is assigned. The server records the "relay-address" field from the Relay-forward message and extracts the solicit message from the "client-message" option.

If administrative policy permits the server to respond to a client on that link, the server will generate and send an Advertise message to the client.

12.4.2. Creation and sending of Advertise messages

The server sets the "msg-type" field to ADVERTISE and copies the values of the following fields from the client's Solicit to the Advertise message:

- transaction-ID
- client-link-local-address

The server places one of its IP addresses (determined through administrator setting) in the "server-address" field of the Advertise message. The server sets the "preference" field according to its configuration information. See section 18.3 for a description of server preference.

The server MUST include options to the Advertise message containing any addresses that would be assigned to IAs contained in the Solicit message from the client. 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.

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 to the address in the "relay-address" field from the Relay-forward message.

If the Solicit message was received directly by the server, the server unicasts the Advertise message directly to the client using the "client-link-local-address" field value as the destination address. The Advertise message MUST be unicast through the interface on which the Solicit message was received.

13. 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.

The client uses the following messages to initiate a configuration event with a server or servers:

Request — Obtain initial configuration information (from a server identified in a previously received Advertise message) when the client has no assigned addresses

Confirm — Confirm the validity of assigned addresses and other configuration changes through the server from which the configuration information was obtained when the client's

assigned addresses may not be valid; for example, when
the client reboots or loses its connection to a link

Renew	Extend the lease on an IA through the server that originally assigned the IA
Rebind	Extend the lease on an IA through any server willing to extend the lease

A client uses the Release/Reply message exchange to indicate to the DHCP server that the client will no longer be using the addresses in the released IA.

A client uses the Decline/Reply message exchange to indicate to the DHCP server that the client has detected that one or more addresses assigned by the server is already in use on the client's link.

13.1. Client Message Validation

Clients MUST silently discard any received client messages (Request, Confirm, Renew, Rebind, Release or Decline messages).

Agents MUST discard any received client messages in which the "client-link-local-address" field does not contain a valid link-local address.

Servers MUST discard any received client messages in which the "options" field contains an authentication option, and the server cannot successfully authenticate the client.

Servers MUST discard any received Request, Renew, Release or Decline message in which the "server-address" field value does not match any of the server's addresses.

13.2. Server Message Validation

Servers MUST silently discard any received server messages (Reply or Reconfigure-init messages).

Clients MUST discard any server messages that meet any of the following criteria:

- The "transaction-ID" field value in the server message does not match the value the client used in its Request or Release message.
- The "client-link-local-address" field value in the server message does not match the link-local address of the interface from which the client sent in its Request, Confirm, Renew, Rebind, Release or Decline message.

- The server message contains an authentication option, and the client's attempt to authenticate the message fails.

Relays MUST discard any Relay-reply message in which the "client-link-local-address" in the encapsulated Reply message does not contain a valid link-local address.

13.3. Client Behavior

A client will use Request, Confirm, Renew and Rebind messages to acquire and confirm the validity of configuration information. A client may initiate such an exchange automatically in order to acquire the necessary network parameters to communicate with nodes off-link. 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.

A client uses the Release message in the management of IAs when the client has been instructed to release the IA prior to the IA expiration time since it is no longer needed.

A client uses the Decline message when the client has determined through DAD or some other method that one or more of the addresses assigned by the server in the IA is already in use by a different client.

13.3.1. Creation and sending of Request messages

If a client has no valid IPv6 addresses of sufficient scope to communicate with a DHCP server, it may send a Request message to obtain new addresses. 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 sets the "msg-type" field to REQUEST, and places the link-local address of the interface it wishes to acquire configuration information for in the "client-link-local-address" field.

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

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

The client adds any 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.

The client sends the Request message to the All DHCP Agents multicast address, destination port 547. The source port selection

can be arbitrary, although it SHOULD be possible using a client configuration facility to set a specific source port value.

The server will respond to the Request message with a Reply message. If no Reply message is received within REP_MSG_TIMEOUT milliseconds, the client retransmits the Request with the same transaction-ID, and doubles the REP_MSG_TIMEOUT value, and waits again. The client continues this process until a Reply is received or REQUEST_MSG_ATTEMPTS unsuccessful attempts have been made, at which time the client MUST abort the configuration attempt. The client SHOULD report the abort status to the application layer.

Default and initial values for REP_MSG_TIMEOUT and REQ_MSG_ATTEMPTS are documented in section 7.5.

13.3.2. Creation and sending of Confirm messages

Whenever a client may have moved to a new link, its IPv6 addresses may no longer be valid. Examples of times when a client may have moved to a new link include:

- The client reboots
- The client is physically disconnected from a wired connection
- The client returns from sleep mode
- 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. The server will indicate the acceptability of the addresses with the status in the IA it returns to the client.

DISCUSSION:

> This section used to allow servers to change the addresses in an IA. Without some additional mechanism, servers responding to Confirm messages can't change safely change the addresses in IAs (although they can change the lifetimes), because servers may send back different addresses.

The client sets the "msg-type" field to CONFIRM, and places the link-local address of the interface it wishes to acquire configuration information for in the "client-link-local-address" field.

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 adds any appropriate options, including one or more IA options (if the client is requesting that the server confirm the validity of some network 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, destination port 547. The source port selection can be arbitrary, although it SHOULD be possible using a client configuration facility to set a specific source port value.

Servers will respond to the Confirm message with a Reply message. If no Confirm message is received within REP_MSG_TIMEOUT milliseconds, the client retransmits the Confirm with the same transaction-ID, and doubles the REP_MSG_TIMEOUT value, and waits again. The client continues this process until a Reply is received or QRY_MSG_ATTEMPTS unsuccessful attempts have been made, at which time the client MUST abort the configuration attempt. The client SHOULD report the abort status to the application layer.

Default and initial values for REP_MSG_TIMEOUT and QRY_MSG_ATTEMPTS are documented in section 7.5.

If the client receives no response to its Confirm message, it MAY restart the configuration process by locating a DHCP server with an Advertise message and sending a Request to that server, as described in section 13.3.1.

13.3.3. Creation and sending 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 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 Request 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 server's administrative configuration. The server may also add new addresses to the IA. The server 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 Request/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 Request message. The client multicasts this Request message to the All DHCP Agents multicast address.

The client sets the "msg-type" field to RENEW, and places the link-local address of the interface it wishes to acquire configuration information for in the "client-link-local-address" field.

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

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

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.

The client sends the Renew message to the All DHCP Agents multicast address, destination port 547. The source port selection can be arbitrary, although it SHOULD be possible using a client configuration facility to set a specific source port value.

The server will respond to the Renew message with a Reply message. If no Reply message is received within REP_MSG_TIMEOUT milliseconds, the client retransmits the Renew with the same transaction-ID, and doubles the REP_MSG_TIMEOUT value, and waits again. The client continues this process until a Reply is received or until time T2 is reached (see section 13.3.4).

Default and initial values for REP_MSG_TIMEOUT are documented in section 7.5.

13.3.4. Creation and sending 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 multicasts this message to the All DHCP Agents multicast address.

The client sets the "msg-type" field to REBIND, and places the link-local address of the interface it wishes to acquire configuration information for in the "client-link-local-address" field.

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 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, destination port 547. The source port selection can be arbitrary, although it SHOULD be possible using a client configuration facility to set a specific source port value.

The server will respond to the Rebind message with a Reply message. If no Reply message is received within REP_MSG_TIMEOUT milliseconds, the client retransmits the Rebind with the same transaction-ID, and doubles the REP_MSG_TIMEOUT value, and waits again. The client continues this process until a Reply is received.

Default and initial values for REP_MSG_TIMEOUT are documented in section 7.5.

DISCUSSION:

The client has several alternatives to choose from if it receives no response to its Rebind message.

- 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

13.3.5. Receipt of Reply message in response to a Reply, 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. If the "status"

field contains a non-zero value, the client reports the error status to the application layer.

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 16.

When the client receives an Unavail error 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 error 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 Conf_NoMatch error 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 error 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 Renw_NoMatch error 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 Unavail error 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 error 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 Rebd_NoMatch error 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 Unavail error 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.

13.3.6. Creation and sending of Release messages

The client sets the "msg-type" field to RELEASE, and places the link-local address of the interface associated with the configuration information it wishes to release in the "client-link-local-address" field.

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 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.

If the client is configured to use authentication, the client generates the appropriate authentication option, and adds this option to the "options" field. Note that the authentication option MUST be the last option in the "options" field. See section 16.9 for more details about the authentication option.

The client send the Release message to the All DHCP Agents multicast address.

13.3.7. Time out and retransmission of Release Messages

If no Reply message is received within REP_MSG_TIMEOUT milliseconds, the client retransmits the Release, doubles the REP_MSG_TIMEOUT value, and waits again. The client continues this process until a Reply is received or REL_MSG_ATTEMPTS unsuccessful attempts have been made, at which time the client SHOULD abort the release attempt. The client SHOULD return the abort status to the application, if an application initiated the release.

Default and initial values for REP_MSG_TIMEOUT and REL_MSG_ATTEMPTS are documented in section 7.5.

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.

13.3.8. Creation and sending of Decline messages

The client sets the "msg-type" field to DECLINE, and places the

link-local address of the interface associated with the configuration information it wishes to decline in the "client-link-local-address" field.

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 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.

If the client is configured to use authentication, the client generates the appropriate authentication option, and adds this option to the "options" field. Note that the authentication option MUST be the last option in the "options" field. See section 16.9 for more details about the authentication option.

The client send the Decline message to the All DHCP Agents multicast address.

13.3.9. Time out and retransmission of Decline Messages

If no Reply message is received within REP_MSG_TIMEOUT milliseconds, the client retransmits the Decline, doubles the REP_MSG_TIMEOUT value, and waits again. The client continues this process until a Reply is received or REL_MSG_ATTEMPTS unsuccessful attempts have been made, at which time the client SHOULD abort the attempt to decline the address. The client SHOULD return the abort status to the application, if an application initiated the release.

Default and initial values for REP_MSG_TIMEOUT and REL_MSG_ATTEMPTS are documented in section 7.5.

13.3.10. 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.

13.4. Server Behavior

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

13.4.1. Receipt of Request messages

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
preference	Enter the servers preference to provide services to the client.
transaction-ID	Enter the transaction-ID from the Request message.
client-link-local address	Enter the client-link-local address from the Request message.
server address	Enter the IP address of the server.

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 clients IA and associate a binding for that client in an implementation-specific manner within the servers configuration parameter database for DHCP clients.

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 Unavail.

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.

13.4.2. 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
preference	Enter the servers preference to provide services to the client.
transaction-ID	Enter the transaction-ID from the Confirm message.
client-link-local address	Enter the client-link-local address from the Confirm message.
server address	Enter the server's address.

When the server receives a Confirm and an IA option is included the client is requesting confirmation that the addresses in the IA are valid. The server 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 information for the client does not match what is in the servers records for that client the server should send back an empty IA with status set to Conf_NoMatch.

If the server finds a match to the Confirm then the server should send back the IA to the client with status set to success.

13.4.3. Receipt of Renew messages

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

preference	Enter the servers preference to provide services to the client.

transaction-ID	Enter the transaction-ID from the Confirm message.
client-link-local address	Enter the client-link-local address from the Confirm message.
server address	Enter the server's address.

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 Renw_NoMatch.

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 Unavail.

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.

13.4.4. 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
preference	Enter the servers preference to provide services to the client.
transaction-ID	Enter the transaction-ID from the Confirm message.

client-link-local address	Enter the client-link-local address from the Confirm message.
server address	Enter the server's address.

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 Rebd_NoMatch.

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 Unavail.

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.

13.4.5. Receipt of Release messages

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 IA, 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 sets the "status" field to "Success". If any of the IAs were invalid or if any of the addresses were not successfully released, the server releases none of the addresses in the message and sets the "status" field to "NoBinding"(section 7.4).

DISCUSSION:

What is the behavior of the server relative to a "partially released" IA; i.e., an IA for which some but not all addresses are released?

Can a client send an empty IA to release all addresses in the IA?

If the IA becomes empty - all addresses are released - can the server discard any record of the IA?

13.4.6. Sending of Reply messages

If the Request, Confirm, Renew, Rebind or Release message from the client was originally received by the server, the server

unicasts the Reply message to the link-local address in the "client-link-local-address" field.

If the message was originally received in a Forward-request or Forward-release message from a relay, the server places the Reply message in the options field of a Response-reply message and unicasts the message to the relay's address from the original message.

14. DHCP Server-Initiated Configuration Exchange

A server initiates a configuration exchange to force 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).

14.1. Reconfigure-init Message Validation

Agents MUST silently discard any received Reconfigure-init messages.

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

Clients MUST discard any Reconfigure-init messages that contain a transaction-ID that matches the transaction-ID in a Reconfigure-init message previously received from the same DHCP server.

14.2. Server Behavior

A server sends a Reconfigure-init message to trigger a client to initiate immediately a Request/Reply message exchange with the server. A server may unicast a Reconfigure-init message directly to a single client or use multicast to deliver a Reconfigure-init message to multiple clients.

14.2.1. Creation and sending of Reconfigure-init messages

The server sets the "msg-type" field to RECONFIG. 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.

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 MAY include a Reconfigure-delay option in a Reconfigure-init message to be unicast to a client, and MUST include a Reconfigure-delay option in a Reconfigure-init message to be multicast to a group of clients.

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

The server may either unicast the Reconfigure-init message to one client or multicast the message to one or more Reconfigure Multicast Addresses previously sent as options to the clients. The server may unicast Reconfigure-init messages to more than one client concurrently; for example, to reliably reconfigure all clients, the server will unicast a Reconfigure-init message to each client.

If the server unicasts to one or more clients, it waits for a Request message from those clients confirming that it has received the Reconfigure-init and are thus initiating a Request/Reply transaction with the server. The server can determine that a Request message is in response to a Reconfigure-init because the transaction-ID in the Request will be the same value as was used in the Reconfigure-init message.

If the server multicasts the Reconfigure-init message, it must use some TBD authentication mechanism that can authenticate the server to multiple clients. There is no reliability mechanism for multicast Reconfigure-init messages. A server might use multicast in the case where it does not have a list of its clients; for example, a server that distributes configuration information to clients using stateless autoconfiguration might not keep a list of clients it has communicated with.

DISCUSSION:

> Authentication of multicast reconfigure-init is still an open issue.

See section 18.2 for recommendations on the use of multicast and unicast Reconfigure-init messages for reliable client reconfiguration.

14.2.2. Time out and retransmission of unicast 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.

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

14.2.3. Time out and retransmission of multicast Reconfigure-init messages

After the server transmits the initial Reconfigure-init message, the server waits RECREP_MSG_TIMEOUT milliseconds. The server then retransmits the Reconfigure-init message, doubles the RECREP_MSG_TIMEOUT value and waits again. The server repeats this process until a total of REC_MSG_ATTEMPTS Reconfigure-init messages have been transmitted.

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

14.2.4. Receipt of Request messages

The server generates and sends Reply message(s) to the client as described in section 13.4.6, including in the "option" field new values for configuration parameters.

14.3. 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.

14.3.1. Receipt of Reconfigure-init messages

Upon receipt of a valid Reconfigure-init message, the client initiates a Request/Reply transaction with the server.

14.3.2. 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 13.3.1 with the following differences:

transaction-ID	The client copies the transaction-ID from the

	Reconfigure-init message into the Request message.
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.
Pause before sending Request	The client pauses before sending the Request for a random value within the range REC_REP_MIN and REC_REP_MAX seconds. This delay helps reduce the load on the server generated by processing large numbers of triggered Request messages from a multicast Reconfigure-init message.

14.3.3. 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 13.3.1 for details.

14.3.4. Receipt of Reply messages

Upon the receipt of a valid Reply message, the client extracts the contents of the "option" 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.

15. 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 will use the All DHCP Servers multicast address as the default.

15.1. Relaying of client messages

When a Relay receives a valid client message, it constructs a Relay-forward message. The relay places an address from

the interface on which the client message was received in the "relay-address" field and the prefix length for that address in the "prefix-length" field. This address will be used by the server to identify the link to which the client is connected and will be used by the relay to forward the Advertise message from the server back to the client.

The relay constructs a "client-message" option 16.4 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.

15.2. Relaying of server messages

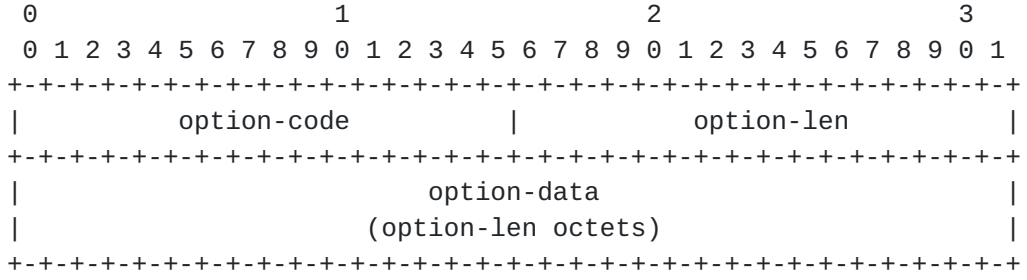
When the relay receives a Relay-reply message, it extracts the server message from the "server-message" option and forwards the message to the address in the client-link-local-address field in the server message. The relay forwards the server message through the interface identified in the "relay-address" field in the Relay-reply message.

16. 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 16.1.

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.

16.1. Format of DHCP options



option-code An unsigned integer identifying the specific option

type carried in this option.

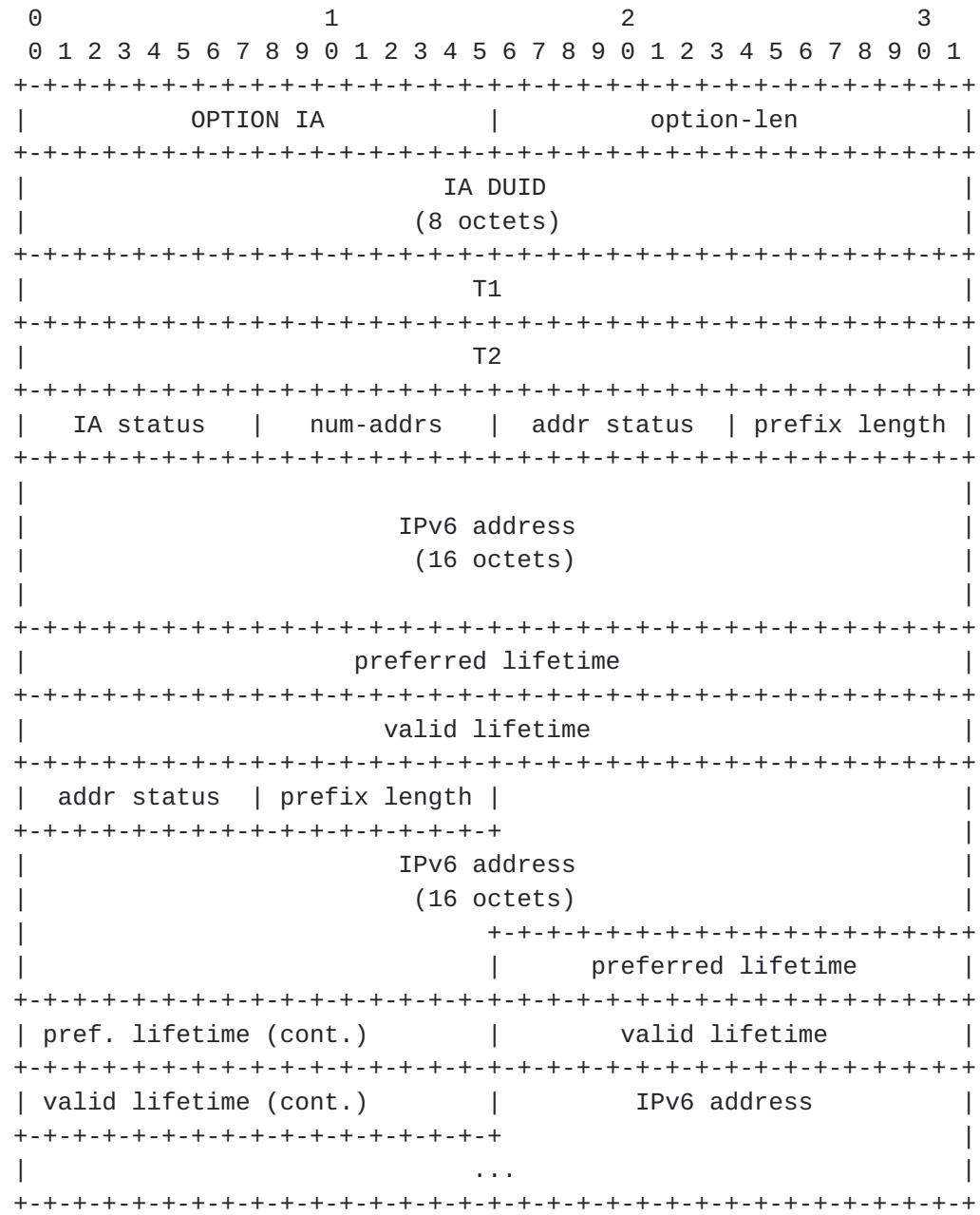
option-len An unsigned integer giving the length of the data in this option in octets.

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

16.2. 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 (1)
```

option-len Variable; equal to 18 + num-addrs*25

IA DUID 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.

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 this address.

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.

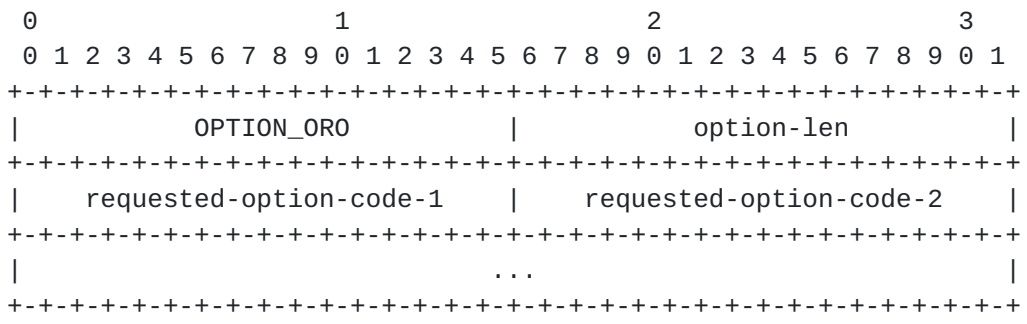
The "IPv6 address", "preferred lifetime" and "valid lifetime" fields are repeated for each address in the IA option (as determined by the "num-addrs" field).

DISCUSSION:

The details of the format and the selection of an IA's DUID are TBD.

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.

16.3. Option request option

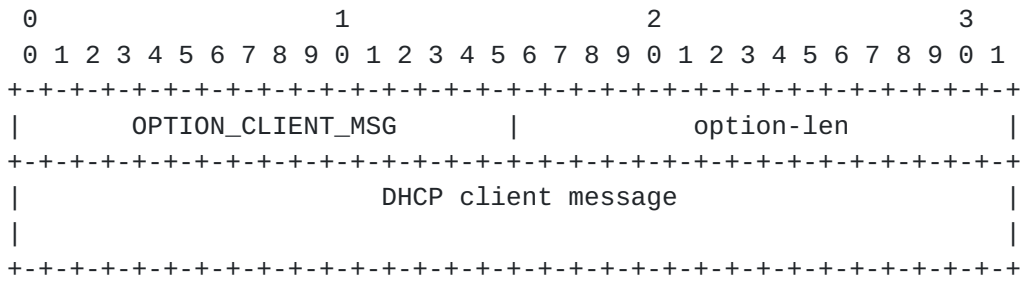


option-code OPTION_ORO (2)

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.

16.4. Client message option

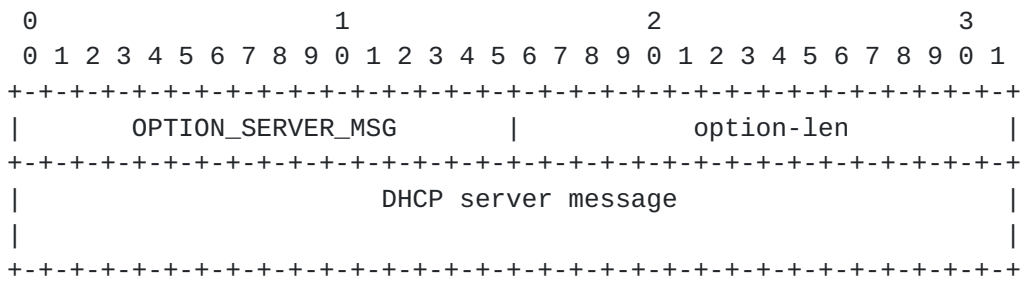


option-code OPTION_CLIENT_MSG (3)

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.

16.5. Server message option

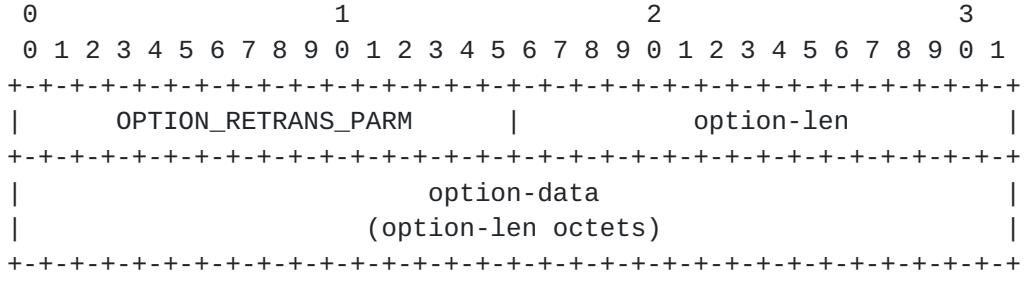


option-code OPTION_SERVER_MSG (4)

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.

16.6. Retransmission parameter option



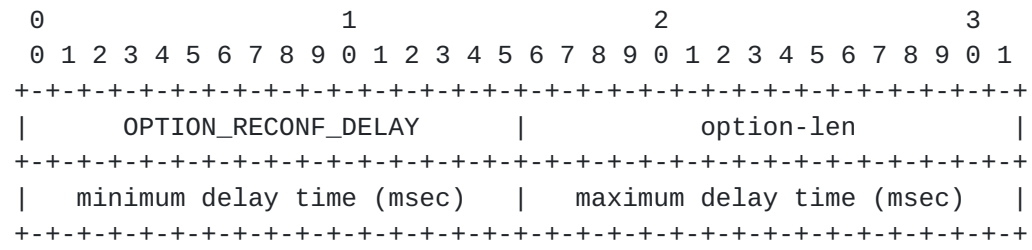
option-code OPTION_RETRANS_PARM (5)

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

option-data TBD - The details of the operational parameters to be set in the client

16.7. Reconfigure-delay option

The Reconfigure-delay option specifies the amount of time a client should delay before sending a Request message in response to a Reconfigure-init message.



option-code OPTION_RECONF_DELAY (6)

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

minimum delay time An unsigned integer giving the minimum delay time in milliseconds

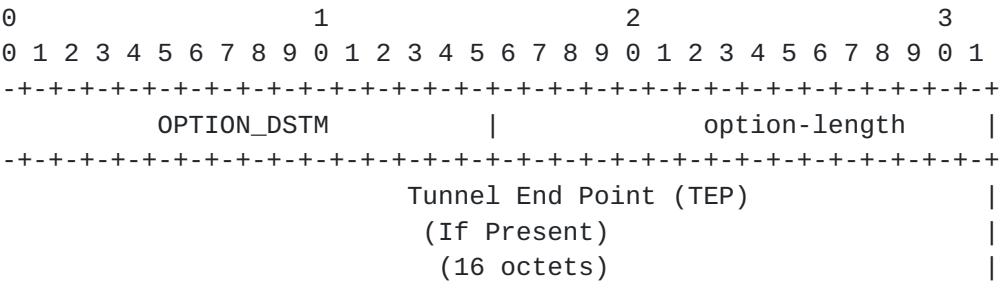
maximum delay time An unsigned integer giving the maximum delay time in milliseconds

The client chooses a random number between the minimum delay time and the maximum delay time and delays that number of milliseconds before sending its Request message.

16.8. 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 [7] 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 an IPv6 address to be used as the Tunnel Endpoint (TEP) to encapsulate an IPv4 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 (7)

option length Variable: 0 or 16

tunnel end point IPv6 Address if Present

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

16.9. Authentication option

The authentication option is TBD.

17. DHCP Client Implementor Notes

This section provides helpful information for the client implementor regarding their implementations. The text described here is not part of the protocol, but rather a discussion of implementation features we feel the implementor should consider during implementation.

17.1. Primary Interface

Since configuration parameters acquired through DHCP can be interface-specific or more general, the client implementor SHOULD provide a mechanism by which the client implementation can be configured to specify which interface is the primary interface. The client SHOULD always query the DHCP data associated with the primary interface for non-interface specific configuration parameters. An implementation MAY implement a list of interfaces which would be scanned in order to satisfy the general request. In either case, the first interface scanned is considered the primary interface.

By allowing the specification of a primary interface, the client implementor identifies which interface is authoritative for non-interface specific parameters, which prevents configuration information ambiguity within the client implementation.

17.2. Advertise Message and Configuration Parameter Caching

If the hardware the client is running on permits it, the implementor SHOULD provide a cache for Advertise messages and a cache of configuration parameters received through DHCP. Providing these caches prevents unnecessary DHCP traffic and the subsequent load this generates on the servers. The implementor SHOULD provide a configuration knob for setting the amount of time the cache(s) are

valid.

17.3. Time out and retransmission variables

Note that the client time out and retransmission variables outlined in section 7.5 can be configured on the server and sent to the client through the use of the "DHCP Retransmission Parameter Option", which is documented in section 16.6. A client implementation SHOULD be able to reset these variables using the values from this option.

17.4. Server Preference

A client MUST wait for SRVR_PREF_WAIT seconds after sending a DHCP Solicit message to collect Advertise messages and compare their preferences (see section 18.3), unless it receives an Advertise message with a preference of 255. If the client receives an Advertise message with a preference of 255, then the client MAY act immediately on that Advertise without waiting for any more additional Advertise messages.

18. DHCP Server Implementor Notes

This section provides helpful information for the server implementor.

18.1. Client Bindings

A server implementation MUST use the IA's DUID and the prefix specification from which the client sent its Request message(s) as an index for finding configuration parameters assigned to the client. While it isn't critical to keep track of the other parameters assigned to a client, the server MUST keep track of the addresses it has assigned to an IA.

The server should periodically scan its bindings for addresses whose leases have expired. When the server finds expired addresses, it MUST delete the assignment of those addresses, thereby making these addresses available to other clients.

The client bindings MUST be stored in non-volatile storage.

The server implementation should provide policy knobs to control whether or not the lifetimes on assigned addresses are renewable, and by how long.

18.2. Reconfigure-init Considerations

A server implementation MUST provide an interface to the

administrator for initiating reconfigure-init events.

A server implementation may provide a mechanism for allowing the specification of how many clients comprise a reconfigure multicast

group. This enables the administrator to control the processing load impact of the multicast of a Reconfigure-init message.

18.2.1. Reliable transmission of multicast Reconfigure-init messages

Because clients will ignore Reconfigure-init messages with the same transaction-ID, a server can retransmit a Reconfigure-init message (using the same transaction-ID) without causing any client to reply more than once. A server SHOULD retransmit a multicast Reconfigure-init message several times to maximize the probability that all clients in the multicast group have received the Reconfigure-init message.

If a server does not receive a Reply message from some clients in a multicast group, the server MAY choose to unicast a Reconfigure-init message to those clients. Because the clients may have received the multicast Reconfigure-init messages while the server did not receive the clients' Reply messages, the server SHOULD use a different transaction-ID in the unicast Reconfigure-init messages to trigger the client to reconfigure.

18.3. Server Preference

The server implementation SHOULD allow the setting of a server preference value by the administrator. The server preference variable is an unsigned single octet value (0--255), with the lowest preference being 0 and the highest 255. Clients will choose higher preference servers over those with lower preference values. If you don't choose to implement this feature in your server, you MUST set the server preference field to 0 in the Advertise messages generated by your server.

18.4. Request Message Transaction-ID Cache

In order to improve performance, a server implementation MAY include an in memory transaction-ID cache. This cache is indexed by client binding and transaction-ID, and enables the server to quickly determine whether a Request is a retransmission or a new Request without the cost of a database lookup. If an implementor chooses to implement this cache, then they SHOULD provide a configuration knob to tune the lifetime of the cache entries.

19. DHCP Relay Implementor Notes

A relay implementation SHOULD allow the specification of a list of

destination addresses for forwarded messages. This list MAY contain any mixture of unicast addresses and multicast addresses.

If a relay receives an ICMP message in response to a DHCP message it has forwarded, it SHOULD log this event.

20. Open Issues for Working Group Discussion

This section contains some items for discussion by the working group.

20.1. Authentication

Authentication is not discussed in this document. Authentication will be modeled on DHCPv4 authentication. Authentication of multicast Reconfigure-init messages is a special problem.

20.2. Identification of IAs by servers

Do servers identify an IA just by its DUID or by <prefix, DUID>? If just by DUID, are DUIDs guaranteed unique (within the DHCP universe)? If so, how is that guarantee implemented?

20.3. DHCP-DNS interaction

Interaction among DHCP servers, clients and DNS servers is not discussed in this document.

20.4. Temporary addresses

How does DHCPv6 interact with temporary addresses? If the server assigns temporary addresses (e.g., addresses with short lifetimes), how can a client application choose an temporary address as a source address in preference to a non-temporary address?

20.5. Use of term "agent"

The term "agent", taken to mean "relay agent or server", may be confusing. "relay agent or server" might be clearer.

20.6. Client behavior when response to Rebind is not received

Section 13.3.4 describes several plausible ways in which a client might respond when it does not receive a Reply to a Rebind message. The acceptable client behaviors need to be defined and described in 13.3.4.

20.7. Additional options

Which additional options should be included in this base spec document?

20.8. Operational parameters

Should servers have an option to set operational parameters - retransmission timeouts, number of retries - in clients?

21. Security

This document references an "authentication option" which is TBD.

DISCUSSION:

> Based on the discussion of security issues at the 8/31/00 design team teleconference and subsequent DHC WG mailing list discussion, DHCPv6 will use the security model from DHCPv4, as described in draft-ietf-dhc-authentication-16.txt (which is soon to be an RFC).

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 message types TBD to be received by UDP at port numbers 546 and 547. Additional message types may be defined in the future.

Section 7.1 lists several multicast addresses used by DHCP.

This document also defines several status codes that are to be returned with the Reply message (see section 9.7). The non-zero values for these status codes which are currently specified are shown in the table in section 7.4.

There is a DHCPv6 option described in section 16.6, which allows clients and servers to exchange values for some of the timing and retransmission parameters defined in section 7.5. Adding new parameters in the future would require extending the values by which the parameters are indicated in the DHCP option. Since there needs

to be a list kept, the default values for each parameter should also be stored as part of the list.

All of these protocol elements may be specified to assume new values at some point in the future. New values should be approved by the process of IETF Consensus [9].

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, Jack McCann, Yakov Rekhter, Matt Thomas, Sue Thomson, and Phil Wells.

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

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 [6, 1] 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 autoconfiguration supporting Duplicate Address

Detection and the two IPv6 lifetimes, to facilitate the dynamic renumbering of addresses and the management of those addresses.

- o Multiple addresses per interface are inherently supported in IPv6.

- o Some DHCPv4 options are unnecessary now because the configuration parameters are either obtained through IPv6 Neighbor Discovery or the Service Location protocol [14].

DHCPv6 Architecture/Model Changes:

- o The message type is the first octet in the packet.

- o IPv6 Address allocations are now handled in a message option as opposed to the message header.

- o Client/Server bindings are now mandatory and take advantage of the client's link-local address to always permit communications either directly from an on-link server, or from a off-link server through an on-link relay.

- o Servers are discovered by a client Solicit, followed by a server Advertise message

- o The client will know if the server is on-link or off-link.

- o The on-link relay may locate off-link server addresses from system configuration or by the use of a site-wide multicast packet.

- o ACKs and NAKs are not used.

- o 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.

- o Clients can issue multiple, unrelated Request messages to the same or different servers.

- o 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.

- o Clients MUST listen to their UDP port for the new Reconfigure message from servers.

- o New options have been defined.

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

- Configuration of Dynamic Updates to DNS

- o Address deprecation, for dynamic renumbering.
- o Relays can be preconfigured with server addresses, or use of multicast.
- o Authentication
- o Clients can ask for multiple IP addresses.
- o Addresses can be reclaimed using the Reconfigure-init message.
- o Integration between stateless and stateful address autoconfiguration.
- o Enabling relays to locate off-link servers.

B. Full Copyright Statement

C. Changes in this draft

This section describes the changes between this version of the DHCPv6

specification and draft-ietf-dhc-dhcpv6-16.txt.

C.1. New messages for confirming addresses and extending the lease on an IA

Four new messages, DHCP Confirm, DHCP Renew, DHCP Rebind and DHCP Decline, have been added and are described in section 13. Client behavior - when and how to send these new messages - and server behavior - how to respond to each - has been defined. The message type codes for these messages have been added to section 7.3.

C.2. New message formats

Section 9 has been restructured to include only one copy of the DHCP message header, because now all the messages have the same header format. Descriptions of the use of header fields in the Confirm, Renew, Rebind and Decline messages have been added to 9.

C.3. Renamed Server-forward message

Section 10.2 has been renamed "relay-reply" for consistency with the rest of the document

C.4. Clarified relay forwarding of messages

Added text to sections on relay behavior to clarify encapsulation and decapsulation of client messages in Relay-forward and Relay-reply messages.

C.5. Addresses and options in Advertise messages

Modified section 12.4.2 so that servers include addresses to be assigned and other options in Advertise messages. Also added text to section 12.3.1 to disallow option values (except as noted in option definitions) in Solicit messages.

C.6. Clarification of IA option format

Changed the label of the prefix length field in an IA option to "prefix length" in the option format diagram, and moved the prefix before the address for consistency with relay messages and other IPv6 protocols.

C.7. Specification of transaction ID in Solicit message

Add text (which was missing) to specify the insertion of a

transaction ID in Solicit messages.

C.8. Edits to definitions

Some of the definitions in section 6 have been edited for clarity.

C.9. Relay agent messages

The formats of relay agent messages are now described in a separate section, 10.

C.10. Relay agent behavior

The behavior of relay agents for all client and server messages is now described in a single section, 15.

C.11. Transmission of all client messages through relays

All client messages are now multicast to the All Agents multicast address and forwarded by relays as appropriate.

C.12. Reconfigure-init messages

Client behavior in response to a Reconfigure-init messages has been extended to accommodate receipt of multiple copies of a Reconfigure-init message due to duplicate messages or retransmission.

Server use of multicast Reconfigure-init has been specified.

Hints about use of multicast and unicast for reliable reconfiguration have been added to server implementor's hints.

C.13. Ordering of sections

Several sections have been re-ordered for clarity.

C.14. DSTM option

The DSTM option has been added (section 16.8).

C.15. Message and option numbering

(In rev -18) Replaced TBD for message and option code numbering with names and temporary values.

C.16. Inclusion of IAs in Solicit message by client

Added text to section 12.3.1 explaining that clients include empty IA(s) in a Solicit message and to section 12.3.3 explaining that server advertises addresses in client IAs.

C.17. Clarification of destination of client messages

Added text to section 13 clarifying the destination (specific server or any available server) of client messages

C.18. Clarification of client use of Confirm messages

Changed text in section 13.3.2 to correctly describe behavior of clients in response to Replay messages from servers.

References

[1] S. Alexander and R. Droms. DHCP Options and BOOTP Vendor Extensions. Request for Comments (Draft Standard) 2132, Internet Engineering Task Force, March 1997.

[2] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. Request for Comments (Best Current Practice) 2119, Internet Engineering Task Force, March 1997.

[3] S. Bradner and A. Mankin. The Recommendation for the IP Next Generation Protocol. Request for Comments (Proposed Standard) 1752, Internet Engineering Task Force, January 1995.

[4] W. J. Croft and J. Gilmore. Bootstrap Protocol. Request for Comments 951, Internet Engineering Task Force, September 1985.

[5] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. Request for Comments (Draft Standard) 2460, Internet Engineering Task Force, December 1998.

[6] R. Droms. Dynamic Host Configuration Protocol. Request for Comments (Draft Standard) 2131, Internet Engineering Task Force, March 1997.

[7] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. Request for Comments (Proposed Standard) 2373, Internet Engineering Task Force, July 1998.

[8] J. McCann, S. Deering, and J. Mogul. Path MTU Discovery for

IP version 6. Request for Comments (Proposed Standard) 1981,
Internet Engineering Task Force, August 1996.

[9] T. Narten and H. Alvestrand. Guidelines for Writing an IANA Considerations Section in RFCs. Request for Comments (Best Current Practice) 2434, Internet Engineering Task Force, October 1998.

[10] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). Request for Comments (Draft Standard) 2461, Internet Engineering Task Force, December 1998.

[11] D. C. Plummer. Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware. Request for Comments (Standard) 826, Internet Engineering Task Force, November 1982.

[12] J. Postel. User Datagram Protocol. Request for Comments (Standard) 768, Internet Engineering Task Force, August 1980.

[13] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. Request for Comments (Draft Standard) 2462, Internet Engineering Task Force, December 1998.

[14] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan. Service Location Protocol. Request for Comments (Proposed Standard) 2165, Internet Engineering Task Force, June 1997.

[15] P. Vixie, Ed., S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS UPDATE). Request for Comments (Proposed Standard) 2136, Internet Engineering Task Force, April 1997.

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

Author's Address

Questions about this memo can be directed to:

Jim Bound
Nokia Networks
5 Wayside Road
Burlington, MA 01803
USA
Phone: +1-781-492-6010
Email: jim.bound@nokia.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