Internet Engineering Task Force
INTERNET DRAFT
DHC Working Group
Obsoletes: draft-ietf-dhc-dhcpv6-04.txt

J. Bound
Digital Equipment Corp.
C. Perkins
IBM Research
12 June 1996

# Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
# draft-ietf-dhc-dhcpv6-05.txt

## Status of This Memo



## Abstract

The Dynamic Host Configuration Protocol (DHCP) provides a framework for passing configuration information, via extensions, to IPv6 hosts. It offers the capability of automatic allocation of reusable network addresses and additional configuration flexibility. This protocol should be considered a stateful counterpart to the IPv6 Stateless Address Autoconfiguration protocol specification.

Contents

## 1. Introduction

The Dynamic Host Configuration Protocol (DHCP) provides configuration parameters to Internet hosts. DHCP consists of a protocol for delivering host-specific configuration parameters from a DHCP server to a host, and a mechanism for allocation of network addresses and other related parameters to IPv6 hosts.

DHCP is built on a client-server model, where designated DHCP server hosts allocate network addresses and automatically deliver configuration parameters to dynamically configured hosts. Throughout the remainder of this document, the term "server" refers to a host providing initialization parameters through DHCP, and the term "client" refers to a host requesting initialization parameters from a DHCP server. DHCPv6 servers maintain state for their clients, in contrast to IPv6 Stateless Address Autoconfiguration [9], where IPv6 hosts should get the same results if they repeat the autoconfiguration procedure multiple times.

DHCPv6 uses Request and Reply messages to support a client/server processing model whereby both client and server are assured that requested configuration parameters have been received and accepted by the client. DHCPv6 supports optional configuration parameters and processing for hosts through its companion document Extensions for the Dynamic Host Configuration Protocol for IPv6 [5].

The IPv6 Addressing Architecture [3] and IPv6 Stateless Address Autoconfiguration specifications provide new features not available in IP version 4 (IPv4) [8], which are used to simplify and generalize the operation of DHCPv6 clients.

Section 2 provides definitions for terminology used throughout this document. Section 3 provides a overview of the protocol design model that guides the design choices in the specification; section 3.2 briefly describes the protocol messages and their semantics. Section 4 provides the message formats and field definitions used for each message. Sections 5, 6, and 7 specify how clients, servers, and relays interact. Appendix A summarizes related work in IPv6 that will provide helpful context; it is not part of this specification, but included for informational purposes.

### 1.1. Specification Language

In this document, several words are used to signify the requirements of the specification. These words are often capitalized.

MUST — This word, or the adjective "required", means that the definition is an absolute requirement of the specification.

MUST NOT — This phrase means that the definition is an absolute prohibition of the specification.

SHOULD — This word, or the adjective "recommended", means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications must be understood and carefully weighed before choosing a different course. Unexpected results may result otherwise.

MAY — This word, or the adjective "optional", means that this item is one of an allowed set of alternatives. An implementation which does not include this option MUST be prepared to interoperate with another implementation which does include the option.

silently discard — The implementation discards the datagram without further processing, and without indicating an error to the sender. The implementation SHOULD provide the capability of logging the error, including the contents of the discarded datagram, and SHOULD record the event in a statistics counter.

## 2. Terminology and Definitions

Relevant terminology from the IPv6 Protocol [2], IPv6 Addressing Architecture, and IPv6 Stateless Address Autoconfiguration will be provided, and then the DHCPv6 terminology.

### 2.1. IPv6 Terminology

IP — Internet Protocol Version 6 (IPv6). The terms IPv4 and IPv6 are used only in contexts where necessary to avoid ambiguity.

node — A device that implements IPv6.

router — A node that forwards IPv6 datagrams not explicitly addressed to itself.

host — Any node that is not a router.

link A communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IPv6. Examples are Ethernet (simple or bridged); 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 links, and E.164 addresses for ISDN links.

link-local address
An address having link-only scope that can be used to reach neighboring nodes attached to the same link. All interfaces have a link-local address.

neighbors
Nodes attached to the same link.

interface
A node's attachment to the link.

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

message The data exchanged between DHCP agents and clients; in this specification, messages are delivered via IPv6 and UDP.

datagram An IP header plus payload.

unicast address
An identifier for a single interface. A datagram 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 datagram sent to a multicast address is delivered to all interfaces identified by that address.

## 2.2. DHCPv6 Terminology

configuration parameter
Any parameter that can be used by a node to configure

its network environment and enable communication on a link or internetwork.

client — A host that initiates requests on a link to obtain configuration parameters.

server — A server is a node that responds to requests from clients on a link to provide: addresses, dynamic updates to DNS, or other configuration parameters.

relay — A node that may advertise DHCP server addresses, or may act as an intermediary to deliver DHCP messages between clients and servers.

DHCP Agent
Either a DHCPv6 server or a DHCPv6 relay.

agent address
The address of a neighboring DHCP relay or DHCP server on the same link as the DHCP client.

msg-type — The msg-type defines the DHCPv6 protocol type for a message.

transaction-ID
The transaction-ID is a monotonically increasing integer identifier specified by the client and is used by the client to match a DHCP Reply to a pending DHCP Request.

server address
The server address specifies the address for the server responding to a client.

binding — A binding in DHCPv6 contains the data which a DHCPv6 server MUST maintain for each of its clients. An implementation MUST support bindings consisting of at least a client's link-local address, agent address, preferred lifetime and valid lifetime [9] for each client address, and the transaction-ID.

## 3. Protocol Design Model

This section is provided for implementors to understand the DHCPv6 protocol design model from an architectural perspective. The goals, conceptual models and implementation examples presented in this section do not specify requirements of the DHCPv6 protocol.

### 3.1. Design Goals

The following list gives general design goals for this DHCPv6 specification.

- DHCPv6 should be a mechanism rather than a policy. DHCPv6 must allow local system administrators control over configuration parameters where desired; e.g., local system administrators should be able to enforce local policies concerning allocation and access to local resources where desired.

- DHCPv6 MUST NOT introduce any requirement for manual configuration of DHCPv6 client hosts, except possibly for manually configured keys. Each host should be able to discover appropriate local configuration parameters without user intervention, and incorporate those parameters into its own configuration.

- DHCPv6 MUST NOT require a server on each link. To allow for scale and economy, DHCPv6 must work across relay agents.

- A DHCPv6 client must be prepared to receive multiple responses to solicitations for DHCP servers. Some installations may include multiple, overlapping DHCPv6 servers to enhance reliability and/or to increase performance.

- DHCPv6 must coexist with statically configured, non-participating hosts and with existing network protocol implementations.

- DHCPv6 MUST be compatible with IPv6 Stateless Address Autoconfiguration.

- DHCPv6 must support the requirements of automated renumbering of IPv6 addresses [1].

- DHCPv6 servers should be able to support Dynamic Updates to DNS [10].

- DHCPv6 servers MUST be able to handle multiple IPv6 addresses for each client.

- A DHCPv6 server to server protocol is NOT part of this DHCPv6 specification.

- It is NOT a design goal of DHCPv6 to specify how a server configuration parameter database is maintained or determined. Methods for configuring DHCP servers are outside the scope of this document.

### 3.2. DHCPv6 Messages

Each DHCPv6 message contains a type, which defines their function within the protocol. The message types are as follows:

01 DHCP Solicit

The DHCP Solicit message is a DHCPv6 multicast (or in special circumstances unicast) message from a client to one or more neighboring DHCPv6 Agents.

02 DHCP Advertise

The DHCP Advertise is an IPv6 unicast message from a DHCP Agent in response to a client DHCP Solicit message.

03 DHCP Request

The DHCP Request is an IPv6 unicast message from a client to a server, when the client knows the IPv6 unicast address of a server, to request configuration parameters on a network.

04 DHCP Reply

The DHCP Reply is an IPv6 unicast message sent by a server to respond to a client's DHCP Request. Extensions [5] to the DHCP Reply describe the resources that the DHCP Server has committed and allocated to the client, and may contain other information for use by the client.

05 DHCP Release

The DHCP Release message is used by a DHCPv6 client to inform the server that the client is releasing a particular address, or set of addresses or resources, so that the server may subsequently mark the addresses or resources as invalid in the server's binding for the client.

06 DHCP Reconfigure

The DHCP Reconfigure message is used by a DHCPv6 server to inform the client that the server has new configuration information of importance to the client. The client is expected to initiate a new Request/Reply transaction.

### 3.3. Request/Response Processing Model

Processing details for the DHCP messages listed above are specified in Sections 5, 6, and 7.

The request/response processing for DHCPv6 is transaction based and uses a best-effort set of messages to guarantee a completed transaction. The timeout and retransmission guidelines and configuration variables are discussed in Section 7.4.

The request/response set is always started by a client with a DHCP Request, which may be issued after the client knows the server's address. The response message or messages (the DHCP Reply) are is from the server (possibly via a DHCP Relay). At this point in the flow all data has been transmitted and, hopefully, received. To provide a method of recovery if either the client or server do not receive the messages to complete the transaction, the client is required to retransmit any DHCP Request message until it elicits a DHCP Reply, or until it can be reasonably certain that the desired DHCP Server is unavailable.

## 4. DHCPv6 Message Formats and Field Definitions

All fields in DHCPv6 messages MUST be initialized to binary zeroes by both the client and server unless otherwise noted. DHCPv6 message types not defined here (msg-types 0 and 7-255) are reserved.

All DHCP Agents MUST join the DHCPv6 Server/Relay-Agent multicast group at the well-known multicast address FF02:0:0:0:0:0:1:0. All DHCP Servers MUST join the DHCPv6 Server multicast group at the well-known multicast address FF02:0:0:0:0:0:tbd.

Servers on the same link as the client MUST use the source address in the IPv6 header from the client as the destination address in the server's response message.

### 4.1. UDP Ports used for DHCPv6 messages

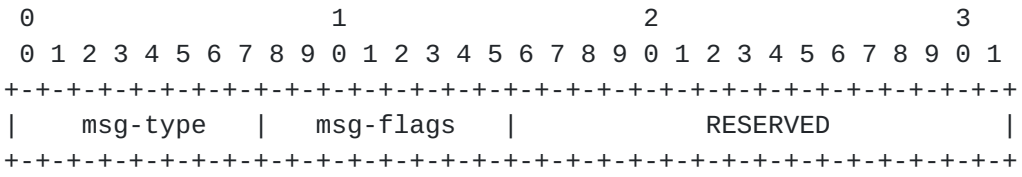
DHCPv6 uses the UDP [7] protocol to communicate between clients and servers. UDP is not reliable, but DHCPv6 must provide some reliability between clients and servers. If a response is not received after transmission of a DHCP message, the message MUST be retransmitted according to the rules specified in 7.4.

A Client MUST transmit all messages over UDP using UDP port 547 as the destination port. A client MUST receive all messages from UDP port 546.

A DHCP Agent MUST transmit all messages over UDP using UDP port 546 as the destination port. A DHCP Agent MUST receive all messages over UDP using UDP port 547.

## 4.2. DHCP Solicit Message Format

A DHCPv6 client transmits a DHCP Solicit message to obtain the address of a neighboring DHCP Agent, and to obtain one or more addresses for DHCP servers which the DHCP Agent is configured to advertise. If a DHCPv6 client does not know any DHCP Agent address, or wants to locate a new server to receive configuration parameters, the client SHOULD use, as the destination IP address, the DHCPv6 Server/Relay-Agent multicast address FF02:0:0:0:0:0:1:0.




msg-type 1

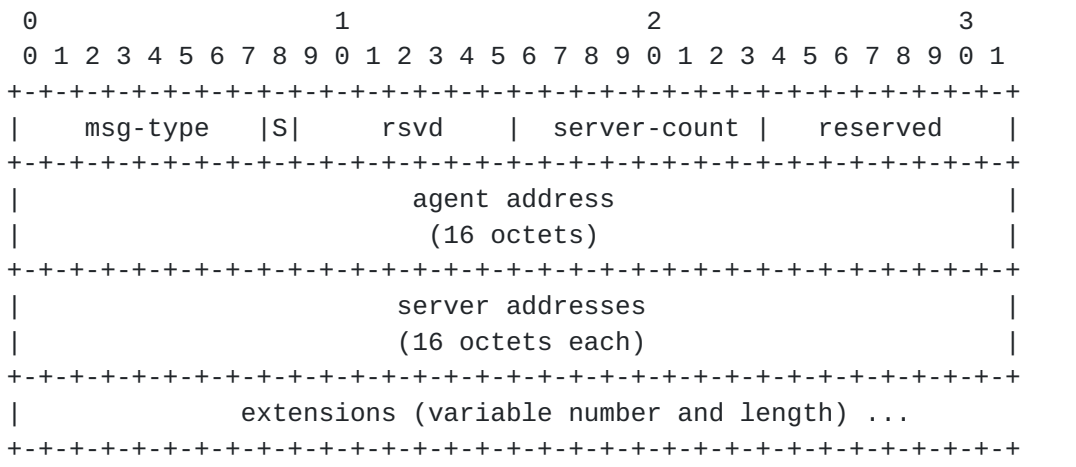
msg-flags 0

RESERVED 0

## 4.3. DHCP Advertise Message Format

A DHCPv6 agent sends a DHCP Advertise message to inform a prospective client about the IPv6 address of a DHCP Agent to which a DHCP Request message may be sent.

A DHCPv6 agent MAY periodically transmit DHCP Advertise messages to the All-DHCPv6 Clients multicast address, no more often than once per second, and with TTL == 1.




msg-type 2

S If set, the agent address is also a server address.

T If set, the advertisement contains a time interval which is the lifetime of the advertisement.

rsvd 0

server-count
The number of addresses listed in the server addresses field.

reserved 0

agent address
The IPv6 address of a neighboring DHCP Agent interface

server addresses
The IPv6 address(es) of the DHCPv6 server(s) which the DHCP Agent has been configured to advertise.

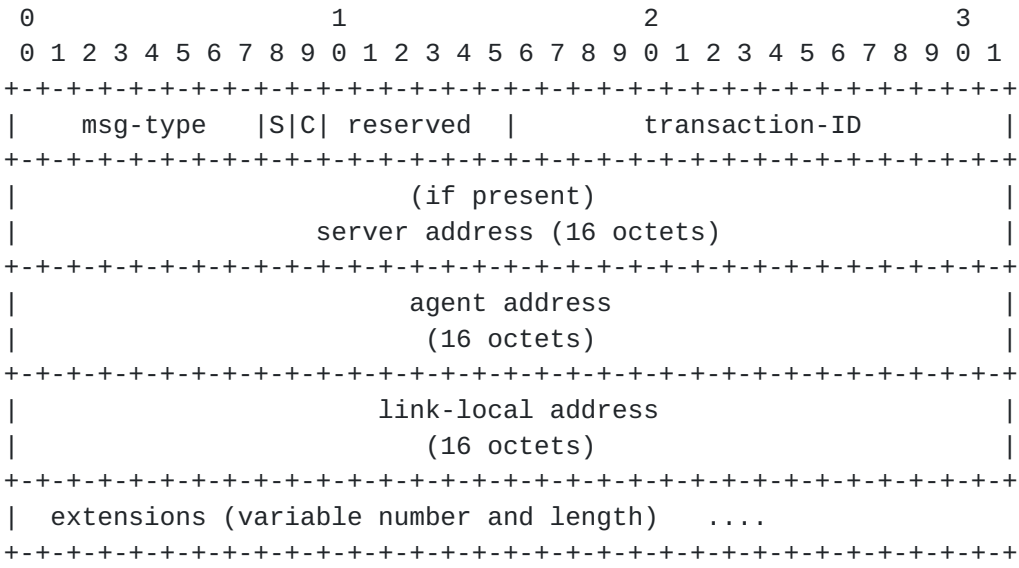
extensions See [5].

Note that if a neighboring DHCPv6 server issues the DHCP Advertise, then the agent address will be the IPv6 address of one of the server's interfaces, the 'S' bit will be set, the agent address will be an address of the server, and there may be zero server addresses

sent in the DHCP Advertise message. It is an error for server-count to be zero if the 'S' bit is not set.

### 4.4. DHCP Request Message Format

In order to request parameters from a DHCP server, a client sends a DHCP Request message and appends the appropriate extensions [5]. If the client does not know any DHCPv6 server address, it must first obtain a server address by multicasting a DHCP Solicit message (see Section 4.2). If the client does not have a valid IPv6 address which is reachable by the DHCPv6 server, the client MUST use the unicast IP address of a local DHCPv6 relay as the destination IP address. Otherwise, the client MAY omit the server address in the DHCP Request message; in this case, the client MUST send the DHCP Request message directly to the server, using the server address as the IPv6 destination address in the IPv6 header.




msg-type 3

S If set, the server address is present

C If set, the client requests the server to clear all existing resources and bindings currently associated with the client, deallocating as needed.

reserved 0

transaction-ID
A monotonically increasing number which the client asks

the server to copy into its DHCP Reply, so that the client can match Replies with pending Requests.

server address
: If present, the IPv6 address of the DHCPv6 server which should receive the client's DHCP Request message.

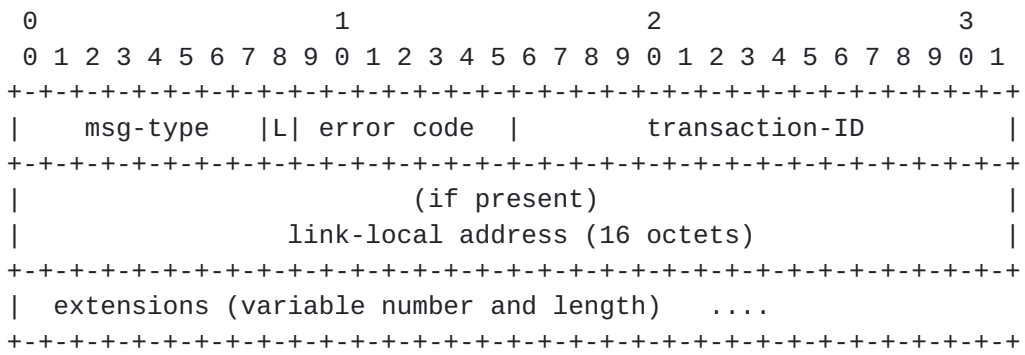
agent address
: The IPv6 address of the relay or server interface from which the client received the DHCP Advertise message

link-local address
: The IPv6 link-local address of the client interface from which the the client issued the DHCP Request message

extensions See [5].

## 4.5. DHCP Reply Message Format

The server sends at least one DHCP Reply message in response to every DHCP Request received. If the request comes with the 'S' bit set, the client could not directly send the Request to the server and had to use a neighboring relay agent. In that case, the server sends back the DHCP Reply with the 'L' bit set, and the DHCP Reply is addressed to the agent address found in the DHCP Request message. If the 'L' bit is set, then the client's link-local address will also be present.




msg-type 3

L If set, the link-local address is present

error code
One of the following values:

| | |
|---|---|
| 0 | Success |
| 16 | Failure, reason unspecified |
| 17 | Authentication failed or nonexistent |
| 18 | Poorly formed request |
| 19 | Resources unavailable |
| 20 | Client record unavailable |
| 21 | Invalid source address in Release |
| 22 | Unable to honor mandatory request |
| 24 | Bad Hair Day |
| 32 | Insufficient Funds |
| 64 | Server unreachable (ICMP error) |

transaction-ID
Copied from the transaction-ID which the DHCPv6 server received in the DHCP Request. to help the client match this reply with an outstanding Request.

link-local address
If present, the IPv6 address of the client interface which issued the corresponding DHCP Request message.

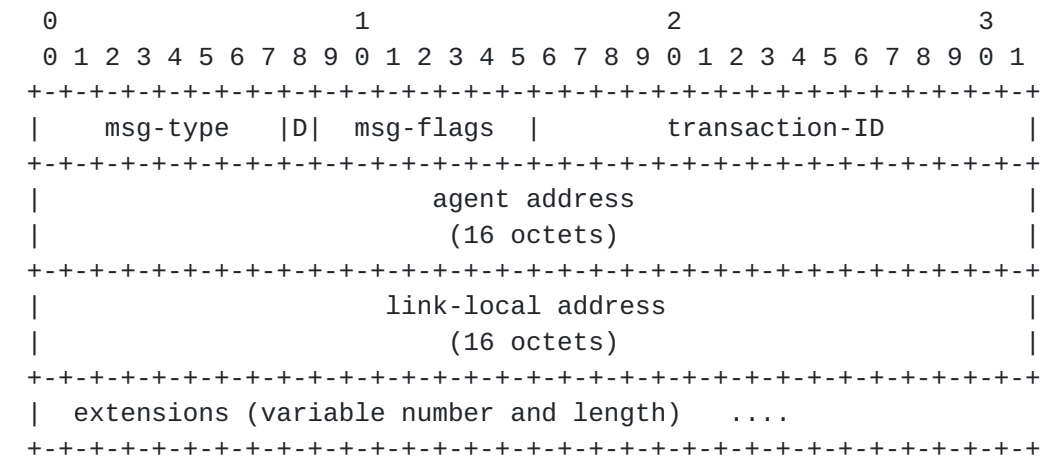
extensions
See [5].

If the 'L' bit is set, and thus the link-local address is present in the Reply message, the Reply is sent by the server to the relay's address which was specified as the agent address in the DHCP Request message, and the relay uses the link-local address to deliver the Reply message to the client. Error Code 22 MUST be sent only in the case that the Server could otherwise honor the requested resource, if the client had not made the parameter values (included in the relevant Extension requesting the resource) mandatory for the server to obey.

## 4.6. DHCP Release Message Format

The DHCP Release message is sent without the assistance of any DHCPv6 relay. When a client sends a Release message, it is assumed to have a valid IPv6 address with sufficient scope to allow access to the target server. Only the parameters which are specified in the

extensions are released. The DHCP server acknowledges the Release message by sending a DHCP Reply (Section 6.3).




msg-type 5

D If the 'D' ("Direct") flag is set, the client instructs the server to send the DHCP Reply directly back to the client, instead of using the given agent address and link-local address to relay the Reply message.

msg-flags 0

transaction-ID
A monotonically increasing number which the client asks the server to use in its DHCP Reply, to help the client match Replies with outstanding Releases.

agent address
The IPv6 address of the agent interface to which the client issued the DHCP Request message

link-local address
The IPv6 link-local address of the client interface from which the the client issued the DHCP Request message

extensions See [5]

Suppose that the client knows that the address it uses as the source IP address in its IPv6 header will still be valid after the server performs the operations requested in the extensions to the DHCP Release message. In that case, and only then, the client SHOULD then specify the 'D' flag. When the 'D' flag is set, the server MUST send the DHCP Reply back to the client's address as shown in the source

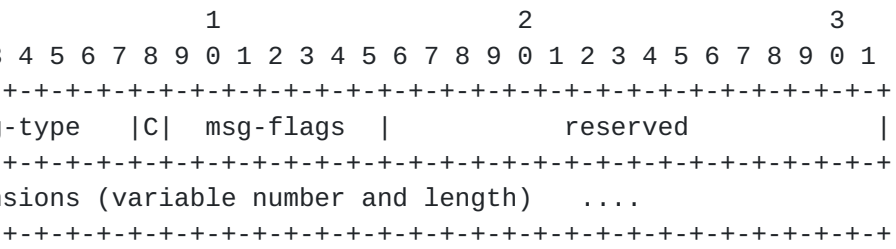
address of the IPv6 header of the Release message. Otherwise, when the 'D' bit is not set, the server MUST use the agent address and link-local address in its DHCP Reply message to forward the Reply message back to the releasing client.

## 4.7. DHCP Reconfigure Message Format

The DHCP Reconfigure message is sent without the assistance of any DHCPv6 relay. When a server sends a Reconfigure message, the client to which it is sent is assumed to have a valid IPv6 address with sufficient scope to be accessible by the server. Only the parameters which are specified in the extensions to the Reconfigure message need be requested again by the client.

The client SHOULD listen at UDP port 546 to receive possible DHCP Reconfigure messages, except in cases where the client knows that no Reconfigure message will ever be issued. If the client does not listen for DHCP Reconfigure messages, it is possible that the client will not receive notification that its DHCP server has deallocated the client's IPv6 address and/or other resources allocated to the client.

See discussion in 6.5.




| | |
|---|---|
| msg-type | 6 |
| msg-type | If set, the DHCPv6 client requests that all servers receiving the message deallocate the resources associated with the client. |
| msg-flags | 0 |
| reserved | 0 |
| extensions | See [5] |

## 5. DHCP Client Considerations

A DHCPv6 client MUST silently discard any DHCP Solicit, DHCP Request, or DHCP Release message it receives.

A DHCPv6 client should retain its configured parameters and resources across client system reboots and DHCPv6 client program restarts. However, in these circumstances a DHCPv6 client SHOULD also formulate a DHCP Request message to verify that its configured parameters and resources are still valid. This Request message MUST have the 'C' bit set, to clear client binding information at the server, of which the client may no longer have any record.

### 5.1. Sending DHCP Solicit Messages

If a node wishes to become a new DHCPv6 client, it must first locate a neighboring DHCP Agent. The client does this by multcasting a DHCP Solicit message to the well-known multicast address FF02:0:0:0:0:0:1:0 (All DHCP Agents Address), setting the TTL == 1.

By setting the 'C' bit in the Solicitation, a DHCPv6 client requests that all the DHCP Servers that receive the solicitation should deallocate their client records that match its link-local address.

### 5.2. Receiving DHCP Advertise Messages

When a DHCPv6 client receives a DHCP Advertise message, it may formulate a DHCP Request message to receive configuration information and resources from the DHCP servers listed in the advertisement. If the Advertise message has zero server addresses and does not have the 'S' bit set, the client MUST silently discard the message. If the server's address is shown as a Multicast address, the advertisement MUST be silently discarded.

If the 'S' bit is set, the DHCP Advertise message was transmitted by a DHCPv6 server on the same link as the client. In this case, the client MUST use the agent address as the address of its server for future DHCPv6 message transactions. Also in this case, the Advertise message may have Extensions; this might allow the DHCPv6 client to select the configuration that best meets its needs from among several prospective servers.

### 5.3. Sending DHCP Request Messages

A DHCPv6 client obtains configuration information from a DHCPv6 server by sending a DHCP Request message. The client must know the server's address before sending the Request message. In addition, the client must have acquired a valid DHCP agent address. If the client and server are on the same link, the agent address used by the client MUST be the same as the DHCP server's address. A DHCP Request message MUST NOT be sent to any multicast address, since otherwise multiple DHCP agents would possibly allocate resources to the client in response to the same Request, and the client would have no way to know which servers had made the allocations.

If the client has no valid IPv6 address and the DHCP server is off-link, then the client MUST include the server address in the appropriate field of the DHCP Request message and set the 'S' bit. In this case, the IPv6 destination address of the Request message MUST be the agent address.

Otherwise, if the client already has a valid IPv6 address and knows the IPv6 address of a candidate IPv6 server, it MUST send the Request message directly to the DHCPv6 server without requiring the services of the local DHCPv6 relay.

If a client wishes to instruct a DHCP server to deallocate all previous resources, configuration information, and bindings associated with its agent address and link-local address, it sets the 'C' bit in the DHCP Request. A client MAY send in such a Request even when it is no longer attached to the link on which the relay address is attached.

A client MAY maintain information about which relay address and server address it has been using for use after a reboot. Even so, after a reboot the client MUST issue its next DHCP Request with the 'C' bit set.

In any case, after choosing a transaction-ID which is numerically greater than its previous transaction-ID, and filling in the appropriate fields of the DHCP Request message, the client MAY append various DHCPv6 Extensions to the message. These Extensions denote specific requests by the client; for example, a client may request a particular IP address, or request that the server send an update containing the client's new IP address to a Domain Name Server. When all Extensions have been applied, the DHCPv6 client unicasts the DHCP Request to the appropriate DHCP Agent.

For each pending DHCP Request message, a client MUST maintain the following information:

- The transaction-ID of the Request message,
- The server address,
- The agent address,
- The time at which the next retransmission will be attempted, and
- All Extensions appended to the request message.

If a client does not receive all the relevant DHCP Reply messages with the same transaction-ID as a pending DHCP Request message within REPLY_MSG_INITIAL_TIMEOUT seconds, it MUST retransmit the Request with the same transaction-ID and continue to retransmit according to the rules in Section 7.4.

## 5.4. Receiving DHCP Reply Messages

When a client receives a DHCP Reply message, it MUST check whether the transaction-ID in the Reply message matches the transaction-ID of a pending DHCP Request message. If no match is found, the Reply message MUST be silently discarded, and an error SHOULD be logged. If the transaction-ID matches that of a pending Request, and the 'L' bit is set, but the source address in the IPv6 header does not match the pending agent address, the client MUST discard the message, and SHOULD log the event. Likewise, if the transaction-ID matches that of a pending Request, and the 'L' bit is not set, but the source address in the IPv6 header does not match the pending server address, the client MUST discard the message, and SHOULD log the event.

If the Reply message is acceptable, the client processes each Extension [5], extracting the relevant configuration information and parameters for its network operation. The Error Code found in the Reply message applies to all extensions found in the Reply. If all expected extensions are not found in the same Reply message, then they are likely to be located in another Reply, possibly with a different Error Code, but with the same Transaction ID. The DHCPv6 Client MUST continue processing DHCP Reply messages until all requested extensions are accounted for. If some requested extensions are not accounted for within DHCP Reply messages sent by the server, the client MUST reissue the entire DHCPv6 Request again, with all extensions, and the same Transaction ID.

Some configuration information extracted from the Extensions to the DHCP Reply message must remain associated with the DHCP server that sent the message. The particular extensions that require this extra measure of association with the server are indicated in the DHCPv6

Extensions document [5]. These associations may be useful with DHCP Release messages.

### 5.5. Sending DHCP Release Messages

If a DHCPv6 client determines that some of its network configuration parameters are no longer needed, it may enable the DHCPv6 server to release allocated resources which are no longer in use by sending a DHCP Release message to the server. The client must consult its list of resource-server associations in order to determine which server should receive the desired Release message. If a client wishes to ask the server to release all information and resources relevent to the client, the client specifies no Extensions; this is preferable to sending a DHCP Request message with the 'C' bit set and no extensions.

Suppoase client wishes to release resources which were granted to it at another link-local address. In that case, the client must instruct the server to send the DHCP Reply directly back to the client, instead of performing the default processing of sending the DHCP Reply back through the agent-address included in the DHCP Release. This is done by setting the 'D' bit in the DHCP Release message. Note that it is an error to include within the DHCP Release message an IPv6 address extension which has the IPv6 address used as the source address of the datagram containing DHCP Release message.

### 5.6. Receiving DHCP Reconfigure Messages

If a DHCPv6 client receives a DHCP Reconfigure message, it is a request for the client to initiate a new DHCP Request/Reply transaction with the server which sent the Reconfigure message. The server sending the Reconfigure message MAY be different than the server which sent a DHCP Reply message containing the original configuration information.

For each Extension which is present in the Reconfigure message, the client appends a matching Extension to its DHCP Request message which it formulates to send to the DHCPv6 server which is found in the IP source address of the message. The client also selects a transaction-ID numerically greater than its last choice and inserts it into the Request message. From then on, processing is the same as specified above in Section 5.3.

Note that a client may be requested by its server to join a multicast group for the purpose of receiving DHCP Reconfigure messages. When a Reconfigure message is delivered to the client by way of the selected

multicast address, the client must delay its further response for a random amount of time uniformly distributed within the interval between RECONF_MSG_MIN_RESP and RECONF_MSG_MAX_RESP seconds. This will minimize the likelihood that the server will be bombarded with DHCP Request messages all at the same time.

## 6. DHCP Server Considerations

A server MUST ignore any DHCP Advertise, DHCP Reply, or DHCP Reconfigure message it receives.

A server uses the combination <link-local address, agent address> to index into its client records. A client record (called a "client binding", or sometimes just a "binding") is used to store all the relevant information, resources, and configuration data which will be associated with the client. Each client binding is uniquely identifiable by the ordered pair <link-local address, agent address>, since the link-local address is guaranteed to be unique [9] on the link identified by the agent address. A DHCPv6 server should retain its clients' bindings across server reboots, and, whenever possible, a DHCPv6 client should be assigned the same configuration parameters despite server host system reboots and DHCPv6 server program restarts. A DHCPv6 server MUST support fixed or permanent allocation of configuration parameters to specific clients.

### 6.1. Receiving DHCP Solicit Messages

If the DHCP Solicit message was received at the All-DHCP-Servers multicast address, the DHCP Server MUST check to make sure that the source address is not a link-local address. If the source address is a link-local address, the server MUST silently discard the packet.

### 6.2. Sending DHCP Advertise Messages

Upon receiving and verifying the correctness of a DHCP Solicit message, a server constructs a DHCP Advertise message and transmits it on the same link as the solicitation was received from. The destination address of the advertisement MUST be the source address of the solicitation. The DHCP server must use a IPv6 address of the interface on which it received the Solicit message as the source address field of the IPv6 header of the message.

The DHCP server MAY append extensions to the Advertisement, in order to offer the soliciting node the best possible information about the services and resources which the server may be able to make

available, should the solicitor choose to subsequently send a DHCP Request to the server.

### 6.3. DHCP Request and Reply Messages

The DHCPv6 server MUST check to ensure that a valid link-local address is present in the client's link-local address field of the Request message. If not, the message MUST be silently discarded. Otherwise, it checks for the presence of the 'S' bit. If the 'S' bit is set, the server MUST check that the server address matches the destination IPv6 address at which the Request message was received by the server. If the server address does not match, the Request message MUST be silently discarded.

If the received agent address and link-local address do not correspond to any binding known to the server, then the server MAY create a new binding for the previously unknown client; otherwise, it SHOULD return a DHCP Reply with a error code of 5.

Before processing the Request, the server must determine whether or not the Request is a retransmission of an earlier DHCP Request from the same client. This is done by comparing the transaction-ID to all those transaction-IDs received from the same client during the previous XID_ID_TIMEOUT seconds. If the transaction-ID is the same as one received during that time, the server MUST take the same action (e.g., retransmit the same DHCP Reply to the client) as it did after processing the previous DHCP Request with the same transaction-ID.

Otherwise (the transaction-ID has not been recently used), when the server has identified and allocated all the relevant information, resources, and configuration data that is associated with the client, it sends that information to its DHCPv6 client by constructing a DHCP Reply message and including the client's information in DHCPv6 Extensions to the Reply message. The DHCP Reply message uses the same transaction-ID as found in the received DHCP Request message.

If the DHCP Request message has the 'S' bit set in the message header, then the Request was sent to the server by a DHCP Relay. In this case, the DHCPv6 server MUST send the corresponding DHCP Reply message to the agent address found in the Request (see section 7.2).

The DHCP Request may contain Extensions, which are interpreted (by default) as advisory information from the client about its configuration preferences. For instance, if the IP Address Extension is present, the DHCPv6 server SHOULD attempt to allocate or extend

the lifetime of the address indicated by the Extension. Some Extensions may be marked by the client as Required.

the DHCP server may accept some extensions for successful processing and allocation, while still rejecting others, or the server may reject various extensions for different reasons (and therefore different Error Codes). The Error Code found in the Reply message applies to all extensions found in the Reply. The DHCP server can send multiple Reply messages in response to the same DHCP Request, each possibly with a different Error Code, but all with the same Transaction ID. The DHCPv6 server MUST send enough DHCP Reply messages to account for all requested extensions. The DHCPv6 server SHOULD attempt to put all the Extensions that were processed with the same Error Code into the same DHCP Reply, in the order in which they were received.

### 6.4. Receiving DHCP Release Messages

If the server receives a DHCP Release Message, it MUST verify that a valid link-local address is present in the link-local address field of the message. If not, the message MUST be silently discarded.

In response to a DHCP Release Message with a valid link-local address, the DHCPv6 server formulates a DHCP Reply message that will be sent back to the releasing client by way of the client's link-local address. A DHCP Reply message sent in response to a DHCP Release message MUST be sent to the client's link-local address via the agent address in the Release message and set the 'L' bit in the Reply, (unless the 'D' bit is set in the Release message).

If the received agent address and link-local address do not correspond to any binding known to the server, then the server SHOULD return a DHCP Reply with a error code of 5.

Otherwise, if the agent address and link-local address indicate a binding known to the server, then the server continues processing for the Release message. If there are any Extensions, the server releases the particular configuration items specified in the extensions. Otherwise, if there are no extensions, the server releases all configuration information in the client's binding.

After performing the operations indicated in the DHCP Release message and its Extensions, the DHCPv6 server formulates a DHCP Reply message, copying the transaction-ID, from the DHCP Release message. For each Extension in the DHCP Release message successfully processed by the server, a matching Extension is appended to the DHCP Reply message. Extensions in the DHCP Release message which cannot

be successfully processed by the server MUST NOT correspond to any Extension appended to the Reply by the server.

### 6.5. Sending DHCP Reconfigure Messages

If a DHCPv6 server needs to change the configuration associated to any of its clients, it constructs a DHCP Reconfigure message and sends it to each such client. The Reconfigure message MUST contain the particular Extensions which inform the client about which configuration information needs to be changed. The Reconfigure MAY be sent to a multicast address chosen by the server and sent to each of its clients.

## 7. DHCP Relay Considerations

The DHCPv6 protocol is constructed so that a relay does not have to maintain any state in order to facilitate DHCPv6 client/server interactions.

All relays MUST use the IPv6 address of the interface from which the DHCPv6 message is transmitted as the source address for the IP header of that DHCPv6 message.

The main purpose of the DHCP Relay is to assist clients and servers to carry out DHCPv6 protocol transactions. This, naturally, requires that the relay be able to discover the addresses of those DHCP Servers whose services can be advertised to propective clients. In addition, this discovery has to be able to happen without specific configuration within the DHCP Relay. By default, the relay discovers local DHCP Servers by use of multicasting DHCP solicitations to the All-DHCP-Servers multicast address, but this behavior is fully configurable. This solicitation occurs every RELAY_DISCOVERY_PERIOD seconds, and the relay updates its list of available servers after each solicitation, after waiting MAX_ADV_WAIT seconds to receive the updated advertisements from the DHCP Servers.

The DHCP Relay MUST be able to be configured with additional DHCP Server address information for its subsequent advertisements to link-local DHCP solicitations. Such configured Server addresses are NOT subject to updates by way of the abovementioned multicast solicitions.

DISCUSSION: TODO: Make the DHCP Server able to include a "advertisement lifetime" in the DHCP Advertisement returned to the DHCP Relay multicast. TODO: Make the DHCP Server able to specify that each Client

Solicitation is "passed through" the relay so that the DHCP Server can append specific Extensions to the Advertisement which is then returned to that client by way of the link-local DHCP Relay. This will be done soon, pending the outcome of discussion within the DHC Working Group.

## 7.1. DHCP Solicit and DHCP Advertise Message Processing

Upon receiving a DHCP Solicit message from a client, a relay constructs a DHCP Advertise message and transmits it to the soliciting client on the same link as the solicitation was received from. The destination address of the advertisement MUST be the source address of the solicitation.

DISCUSSION: If the Solicit is delivered to a server and the server is allowed to send the corresponding Advertise back to a client, the server could then include some prospective information to "entice" a client to use its services. For instance, a server could include proposed lifetime information, and a client could pick the server with the best "terms". Presumably, this forwarding behavior should be a matter of relay configuration instead of client request. I'll assume that for now and try to make the protocol reflect the ability of DHCP Advertise messages to contain Extensions and come from DHCP servers off-link. That may take a little more doing which isn't in the protocol right now, be patient.

DISCUSSION: What about the 'C' bit in Advertisements?

When transmitting a DHCP Advertise message, a relay indicates how many server addresses are included in the advertisement, and includes each address in the DHCP Advertise message. The DHCP Advertise message must use a routeable IPv6 address in the source address of the IPv6 header of the message. In particular, the source address of any DHCP Advertise message sent by a DHCPv6 relay MUST NOT be a link-local address.

## 7.2. DHCP Request Message Processing

When a relay receives a DHCP Request message, it MUST check that the message is received from a link-local address, that the link-local address matches the link-local address field in the Request message

header, and that the agent address field of the message matches an IPv6 address associated to the interface from which the DHCP Request message was received. The relay MUST also check whether the 'S' bit is set in the message header. If any of these checks fail, the message is not acceptable and MUST be silently discarded.

If the received request message is acceptable, the relay then transmits the DHCP Request message to the DHCPv6 server found in the Server Address field of the received DHCP Request message. All of the fields of DHCP Request message header transmitted by the relay are copied over unchanged from the DHCP Request received from the client. Only the fields in the IPv6 header will differ from the datagram received from the client, not the payload.

DISCUSSION: Would an error return be better?

DISCUSSION: How about ICMP Unreachable when the Request fails?

### 7.3. DHCP Reply Message Processing

When the relay receives a DHCP Reply, it MUST check whether the message has the 'L' bit set. It must check whether the link-local address field contains an IPv6 address that has prefix FE80::00 . If all the checks are satisfied, the relay MUST send a DHCP Reply message to the link-local address listed in the received Reply message. Only the fields in the IPv6 header will differ from the datagram received from the server, not the payload.

### 7.4. Retransmission and Configuration Variables

When a DHCPv6 client does not receive a DHCP Reply in response to a pending DHCP Request, the client MUST retransmit the identical DHCP Request to the same server again until it can be reasonably sure that the DHCPv6 server is unavailable and an alternative can be chosen. It is important for the DHCP Server to be sure that its client has received the configuration information included with the Extensions to the DHCP Reply message.

Likewise, but less commonly, when a DHCP server does not receive a DHCP Request message in response to its DHCP Reconfigure message to the client, the server MUST retransmit the identical DHCP Reconfigure to the client until it is reasonably certain that the client is not available for reconfiguration. If no corresponding DHCP Request is ever received by the server, the server MAY erase or deallocate information as needed from the client's binding.

These retransmissions occur using the following configuration variables for a DHCPv6 implementation that MUST be configurable by a client or server are as follows:

REPLY_MSG_INITIAL_TIMEOUT

The time in seconds that a DHCPv6 client waits to receive a server's DHCP Reply before retransmitting a DHCP Request.

Default: 2 seconds.

REPLY_MSG_MIN_RETRANS

The minimum number of DHCP Request transmissions that a DHCPv6 client should retransmit, before aborting the request, possibly retrying the Request with another Server, and logging a DHCPv6 System Error.

Default: 10 retransmissions.

REPLY_MSG_RETRANS_INTERVAL

The time between successive retransmissions of DHCP Request messages.

Default: 2 seconds.

RECONF_MSG_INITIAL_TIMEOUT

The time in seconds that a DHCPv6 server waits to receive a client's DHCP Request before retransmitting its DHCP Reconfigure.

Default: 2 seconds.

RECONF_MSG_MIN_RETRANS

The minimum number of DHCP Reconfigure messages that a DHCPv6 server should retransmit, before assuming the the client is unavailable and that the server can proceed with the needed reconfiguration of that client's resources, and logging a DHCPv6 System Error.

Default: 10 retransmissions.

RECONF_MSG_RETRANS_INTERVAL

The time between successive retransmissions of DHCP Reconfigure messages.

Default: 2 seconds.

RECONF_MSG_MIN_RESP

The minimum amount of time before a client can respond to a DHCP Reconfigure message sent to a multicast address.

Default: 1 second.

RECONF_MSG_MAX_RESP

The maximum amount of time before a client MUST respond to a DHCP Reconfigure message sent to a multicast address.

Default: 8 second.

RELAY_DISCOVERY_PERIOD

The period fo time between successive attempts by the DHCP Relay to discovery available DHCP Servers.

Default: 3600 seconds (1 hour).

MAX_ADV_WAIT

The amount of time the relay waits to hear DHCP Advertisements from DHCP Servers after the relay issues its periodic solicitation to the All-DHCP Servers multicast address.

The following parameter specifies how long a DHCPv6 server has to keep track of client transaction-IDs in order to make sure that client retransmissions using the same transaction-ID are idempotent.

XID_IT_TIMEOUT

Default: 10800 seconds

## 8. Security Considerations

DHCP clients and servers must often be able to authenticate the messages they exchange. For instance, a DHCP server may wish to be certain that a DHCP Request originated from the client identified by

the <link-local address, agent address> fields included within the Request message header. Conversely, it is often essential for a DHCP client to be certain that the configuration parameters and addresses it has received were sent to it by an authoritative DHCP server. Similarly, a DHCP server should only accept a DHCP Release message which seems to be from one of its clients, if it has some assurance that the client actually did transmit the Release message. At the time of this writing, there is no generally accepted mechanism useful with DHCPv4 that can be extended for use with DHCPv6.

The IPv6 Authentication Header can provide security for DHCPv6 messages when both endpoints have a suitable IPv6 address. However, a client often has only a link-local address, and such an address is not sufficient for a DHCPv6 server which is off-link. In those circumstances the DHCP relay must be involved, so that the DHCP message MUST have the relay's address in the IPv6 destination address field, even though the client aims to deliver the message to the DHCPv5 server. The DHCPv6 Client-Server Authentication Extension is intended to be used in these circumstances.

## 9. Acknowledgements

Thanks to the DHC Working Group for their time and input into the specification. A special thanks for the consistent input, ideas, and review by (in alphabetical order) Brian Carpenter, Ralph Droms, Thomas Narten, 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.

The authors MUST also thank their employers for the opportunity and funding to work on DHCPv6 and IPv6 in general as an individual in the IETF.

## A. Related Work in IPv6

The related work in IPv6 that would best serve an implementor to study is the IPv6 Specification [2], the IPv6 Addressing Architecture [3], IPv6 Stateless Address Autoconfiguration [9], IPv6 Neighbor Discovery Processing [4], and Dynamic Updates to DNS [10]. These specifications afford DHCPv6 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 DHCPv6 implementors to understand is that IPv6 requires that every link in the internet have an MTU of 576 octets or greater (in IPv4 the requirement is 68 octets). This means that a UDP datagram of 536 octets will always pass through an internet (less 40 octets for the IPv6 header), as long as there are no IP options prior to the UDP header in the datagram. But, IPv6 does not support fragmentation at routers and fragmentation must take place end-to-end between hosts. If a DHCPv6 implementation needs to send a datagram greater than 536 octets it can either fragment the UDP datagram in UDP or use Path MTU Discovery [2] to determine the size of the datagram that will traverse a network path. It is implementation defined how this is accomplished in DHCPv6.

The IPv6 Addressing Architecture Specification provides 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 multicast addressing is well defined, and nodes can create link-local addresses during initialization of the nodes environment. This means that a host immediately can configure an IPv6 address at initialization for an interface, before communicating in any manner on the link. The host can then use a well-known multicast address to begin communications to discover neighbors on the link, or to send a DHCP Solicit and locate a DHCPv6 server or relay.

IPv6 Stateless Address Autoconfiguration [9] specifies procedures by which a host may autoconfigure addresses based on neighbor discovery router advertisements, and the use of a validation lifetime to support renumbering of addresses on the Internet. In addition the protocol interaction by which a host begins stateless or stateful autoconfiguration is specified. DHCPv6 is one vehicle to perform stateful autoconfiguration. Compatibility with addrconf is a design goal of DHCPv6.

IPv6 Neighbor Discovery [4] is the node discovery protocol in IPv6 (replaces and enhances functions of IPv4 ARP Model [6]). To truly understand IPv6 and addrconf it is strongly recommended that implementors understand IPv6 Neighbor Discovery.

Dynamic Updates to DNS [10] is a specification that supports the dynamic update of DNS records for both IPv4 and IPv6. DHCPv6 can use the dynamic updates to DNS to now integrate addresses and name space to not only support autoconfiguration, but also autoregistration in IPv6.

## B. Change History

### B.1. Changes from November 95 to February 96 Drafts

Substituted use of client's link-local address for previous uses of client's interface token.

Reorganized DHCP messages into Solicit/Advertise, Request/Reply, Release, and Reconfigure.

Made message-specific formats instead of using the same DHCP header for each message.

Eliminated retransmission message types.

Server commits after receiving DHCP Request, and optimistically depends on client retransmissions as negative acknowledgement.

Eliminated total-addrs.

Eliminated all definitions and most fields related to allocating IPv6 addresses (moved to the Extensions specification).

Renamed "gateway address" to be "agent address".

Added "Considerations" sections.

### B.2. Changes from February 96 to June 96 Drafts

Added language referring to DHCP Client-Server Authentication extension.

Moved the 'L' bit in the DHCP Reply Message format to save 32 bits.

Added language for multicast Reconfigure message handling.

In the process of adding capability for the DHCP Relay to multicast and obtain DHCP Server addresses.

## C. 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 and DHCPv6. The differences are because of three key reasons:

- o IPv6 inherently supports a new model and architecture for communications and autoconfiguration of addresses.

- o DHCPv6 in its design was able to take advantage of the inherent benefits of IPv6.

- o New features were added to support the evolution and the existence of mature Internet users in the industry.

IPv6 Architecuture/Model Changes:

- o 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 a server or relay agent on the local link.

- o The need for bootp compatibility and broadcast flags are removed, which permitted a great deal of freedom in designing the new packet formats for the client and server interaction.

- o Multicast and the scoping methods in IPv6 permitted the design of discovery packets that would inherently define their range by the multicast address for the function required.

- o Stateful autoconfiguration must coexist and integrate with stateless autoconfiguration supporting Duplicate Address Detection and two lease times to facilitate the dynamic renumbering of addresses and the management of those addresses.

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

- o Most DHCPv4 options are unnecessary now because the configuration parameters are either obtained thru IPv6 Neighbor Discovery or the Service Location protocol.

DHCPv6 Architecture/Model Changes:

- o Message types are the first bytes in the packet.

- o IPv6 Address allocations are now handled in a message extension as opposed to the main 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 remote server through an on-link relay-agent.

- Servers are now discovered by a client solicit and server or relay-agent advertisement model.

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

- The client after a solicit will be returned the addresses of available servers either from an on-link server or from an on-link relay-agent as agents providing the advertisements.

- The on-link relay-agent will obtain the location of remote server addresses from system configuration or by the use of a site wide DHCPv6 Multicast packet.

- The protocol is optimized and removes the use of ACKs and NAKs once the client and server set-up is complete.

- The server uses client retransmits to indicate that the client may have become invalid, which permits recovery in the case where the network has faulted.

- DHCPINFORM is inherent in the new packet design; a client can request configuration parameters other than IPv6 addresses in the optional extension headers.

- Clients must listen to their UDP port for the new Reconfigure message type from servers, unless they join the appropriate multicast group as specified by the DHCP server.

- Dynamic Updates to DNS are supported in the IPv6 Address extension.

- New extensions have been defined.

New Internet User Features:

- Configuration of Dynamic Updates to DNS to support multiple implementation policy requirements.

- Configuration of what policy is enforced when addresses are deprecated for dynamic renumbering can be implemented.

- Configuration of how relay-agents locate remote servers for a link can be implemented.

- An Authentication extension has been added.

- Configuration of IPv6 Mobile Binding Updates and Anycast Addresses can be facilitated and implemented to support Multihomed nodes.

- Configuration of additional addresses for server applications can be requested by a client in an implementation.

- Configuration of reclaiming infinite leases can be implemented using the Reconfigure message type.

- Configuration of tightly coupled integration between stateless and stateful address autoconfiguration can be implemented.

References

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

[2] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 1883, December 1995.

[3] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. RFC 1883, December 1995.

[4] T. Narten, E. Nordmark, and W. Simpson. IPv6 Neighbor Discovery. draft-ietf-ipngwg-discovery-03.txt -- work in progress, November 1995.

[5] C. Perkins. Extensions to DHCPv6. draft-ietf-dhc-dhcpv6ext-02.txt -- work in progress, June 1996.

[6] David C. Plummer. An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Addresses for Transmission on Ethernet Hardware. RFC 826, November 1982.

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

[8] J. B. Postel, Editor. Internet Protocol. RFC 791, September 1981.

[9] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. draft-ietf-addrconf-ipv6-auto-06.txt - work in progress, November 1995.

[10] S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS). draft-ietf-dnsind-dynDNS-06.txt -- work in progress, February 1996.

## Chair's Address

The working group can be contacted via the current chair:

Ralph Droms
Computer Science Department
323 Dana Engineering
Bucknell University
Lewisburg, PA 17837

Phone: (717) 524-1145
E-mail: droms@bucknell.edu

## Author's Address

Questions about this memo can be directed to:

Jim Bound
Digital Equipment Corporation
110 Spitbrook Road, ZKO3-3/U14
Nashua, NH 03062

Phone: +1-603-881-0400
Fax:
E-mail: bound@zk3.dec.com

Charles Perkins
T. J. Watson Research Center
IBM Corporation
30 Saw Mill River Rd., Rm H3-D34
Hawthorne, NY 10532
+1-914-784-7350
+1-914-784-6205
perk@watson.ibm.com