Internet Engineering Task Force
INTERNET DRAFT
DHC Working Group
Obsoletes: draft-ietf-dhc-dhcpv6-08.txt

J. Bound
Digital Equipment Corp.
C. Perkins
Sun Microsystems
27 February 1997

Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
draft-ietf-dhc-dhcpv6-09.txt

Status of This Memo

Abstract

The Dynamic Host Configuration Protocol (DHCPv6) provides a framework for passing configuration information, via extensions, to IPv6 nodes. 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 (DHCPv6, or in this document usually DHCP) provides configuration parameters to Internet nodes. DHCP consists of a protocol for delivering node-specific configuration parameters from a DHCP server to a client, and a mechanism for allocation of network addresses and other related parameters to IPv6 [3] nodes.

DHCP is built on a client-server model, where designated DHCP servers allocate network addresses and automatically deliver configuration parameters to dynamically configurable clients. Throughout the remainder of this document, the term "server" refers to a node providing initialization parameters by way of the DHCP protocol, and the term "client" refers to a node requesting initialization parameters from a DHCP server. DHCP servers maintain state for their clients, in contrast to IPv6 Stateless Address Autoconfiguration [11], where IPv6 nodes 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. DHCP supports optional configuration parameters and processing for nodes through extensions described in its companion document ``Extensions for the Dynamic Host Configuration Protocol for IPv6'' [7].

The IPv6 Addressing Architecture [4] and IPv6 Stateless Address Autoconfiguration specifications provide new features not available in IP version 4 (IPv4) [10], which are used to simplify and generalize the operation of DHCP clients.

Section 2 provides definitions for terminology used throughout this document. Section 3 provides an overview of the protocol design model that guided 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. Appendix B itemizes changes between different versions of this protocol specification. Appendix C discusses the differences between DHCPv4 and DHCPv6.

2. Terminology and Definitions

Relevant terminology from the IPv6 Protocol [3], IPv6 Addressing Architecture [4], and IPv6 Stateless Address Autoconfiguration [11] 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 IP.
router	A node that forwards IP 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 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 that can be used to reach neighboring nodes attached to the same link. All interfaces have a link-local address.
neighbor	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	A unit of data carried in a datagram, exchanged between DHCP agents and clients.

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.

DHCP client A node that initiates requests on a link to obtain configuration parameters.

DHCP server A server is a node that responds to requests from clients to provide: addresses, prefix lengths, or other configuration parameters.

DHCP relay A node that acts as an intermediary to deliver DHCP messages between clients and servers.

DHCP Agent
Either a DHCP server or a DHCP relay.

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

transaction-ID
The transaction-ID is a monotonically increasing integer identifier specified by the client or server, and used to match a response to a pending message.

binding A binding (or, client binding) in DHCP contains the data which a DHCP server maintains for each of its clients (see Section 6).

2.3. 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.

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 DHCP specification.

- DHCP should be a mechanism rather than a policy. DHCP 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.

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

- DHCP MUST NOT require a server on each link. To allow for scale and economy, DHCP MUST work across DHCP relays.

- A DHCP client MUST be prepared to receive multiple (possibly different) responses to solicitations for DHCP servers. Some installations may include multiple, overlapping DHCP servers to enhance reliability and/or to increase performance.

- DHCP MUST coexist with statically configured, non-participating nodes and with existing network protocol implementations.

- DHCPv6 MUST be compatible with IPv6 Stateless Address Autoconfiguration [11].

- DHCP MUST support the requirements of automated renumbering of IP addresses [1].

- DHCP servers should be able to support Dynamic Updates to DNS [12].

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

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

- It is NOT a design goal of DHCP 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. DHCP Messages

Each DHCP message contains a type, which defines their function within the protocol. Processing details for these DHCP messages are specified in Sections 5, 6, and 7. The message types are as follows:

01 DHCP Solicit

The DHCP Solicit message is a DHCP message sent to one or more DHCP Agents.

02 DHCP Advertise

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

03 DHCP Request

The DHCP Request is an IP unicast message from a client to a server to request configuration parameters on a network.

04 DHCP Reply

The DHCP Reply is an IP unicast message sent by a server to respond to a client's DHCP Request. Extensions [7] 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 DHCP 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 as invalid, or release resources in the server's binding for the client.

06 DHCP Reconfigure

The DHCP Reconfigure message is used by a DHCP server to inform its 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

The request/response processing for DHCPv6 is transaction based and uses a best-effort set of messages to guarantee a completed transaction.

Transactions are usually started by a client with a DHCP Request, which may be issued after the client knows the server's address. The response (DHCP Reply) 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 the corresponding DHCP Reply or Replies, or until it can be reasonably certain that the desired DHCP Server is unavailable. The timeout and retransmission guidelines and configuration variables are discussed in Section 8.

All DHCP Agents (Servers and Relays) MUST join the link-local All-DHCP-Agent multicast group at the well-known multicast address FF02:0:0:0:0:0:1:2. All DHCP Servers MUST, in addition, join the site-local All-DHCP-Servers multicast group at the well-known multicast address FF05:0:0:0:0:0:1:3. All DHCP Relays MUST, on the other hand, join in addition the site-local All-DHCP-Relays multicast group at the well-known multicast address FF05:0:0:0:0:0:1:4.

DHCP uses the UDP [9] protocol to communicate between clients and servers. UDP is not reliable, but DHCP has to 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 Section 8. The All-DHCP-Relays address will be used eventually when DHCP Servers wish to automatically configure all site DHCP Relays.

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

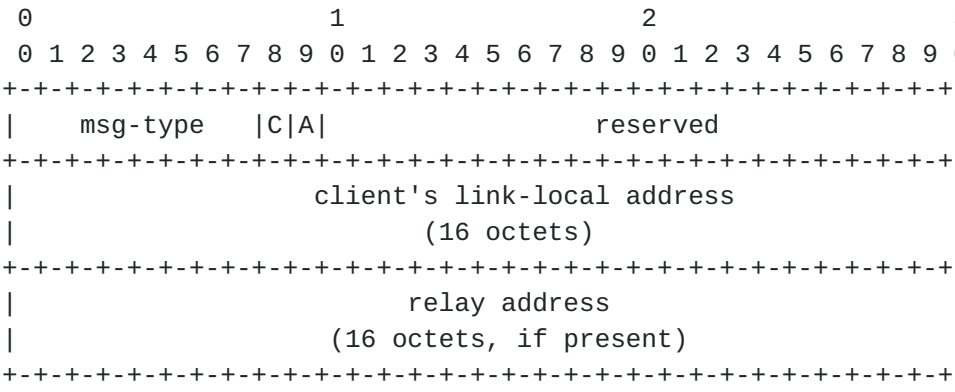
A DHCP Agent MUST transmit all messages to clients over UDP using port 546 as the destination port. A DHCP Agent MUST receive all messages over UDP using port 547. The source port for DHCP messages is arbitrary.

4. DHCP Message Formats and Field Definitions

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

4.1. DHCP Solicit Message Format

A DHCP client (or DHCP relay on behalf of a client) transmits a DHCP Solicit message to obtain one or more DHCP server addresses.



msg-type 1

C If set, the client requests that all servers receiving the message deallocate the resources associated with the client.

A If set, the relay's address is present

reserved 0

client's link-local address
The IP link-local address of the client interface from which the client issued the DHCP Request message

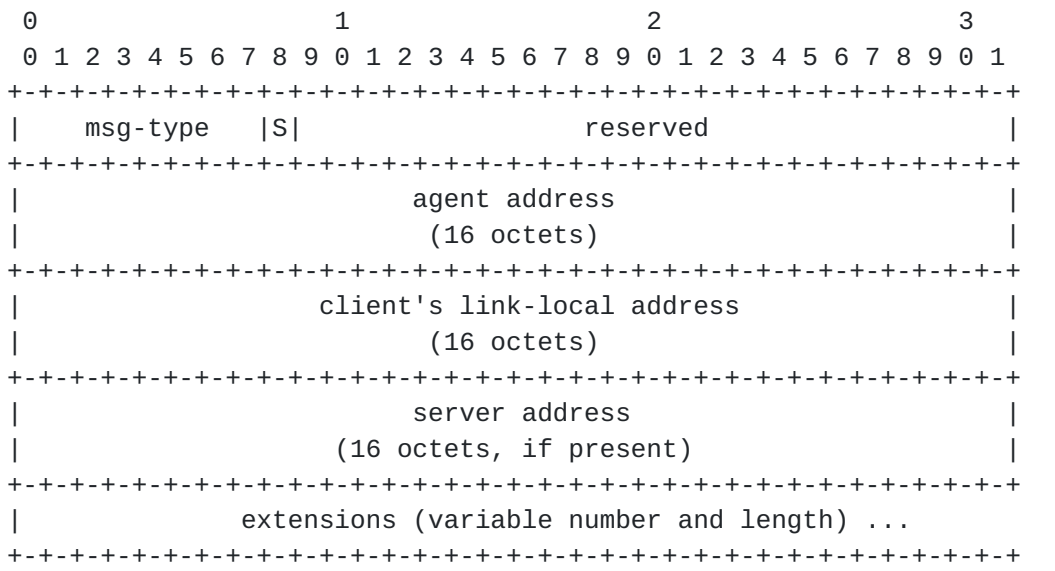
relay address
If present, the IP address of the interface on which the relay received the client's DHCP Solicit message

If a DHCP 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 well-known All DHCP Agents multicast address FF02:0:0:0:0:0:1:2. Any DHCP Relay receiving the solicitation MUST forward it to the All-DHCP-Servers multicast address, to instruct DHCP Servers to send their

advertisements to the prospective client. In that case, the relay MUST copy the client's link-local address into the message, copy the address of its interface from which the client's solicitation was received into the agent's address field, and set the 'A' bit.

4.2. DHCP Advertise Message Format

A DHCP agent sends a DHCP Advertise message to inform a prospective client about the IP address of a DHCP Agent to which a DHCP Request message may be sent. When the client and server are on different links, the server sends the advertisement back through the DHCP Relay whence the solicitation came.



msg-type 2

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

reserved 0

agent address
The IP address of a DHCP Agent interface on the same link as the client.

client's link-local address
The IP link-local address of the client interface from which the client issued the DHCP Request message

server address
The IP address of the DHCP server

extensions See [7].

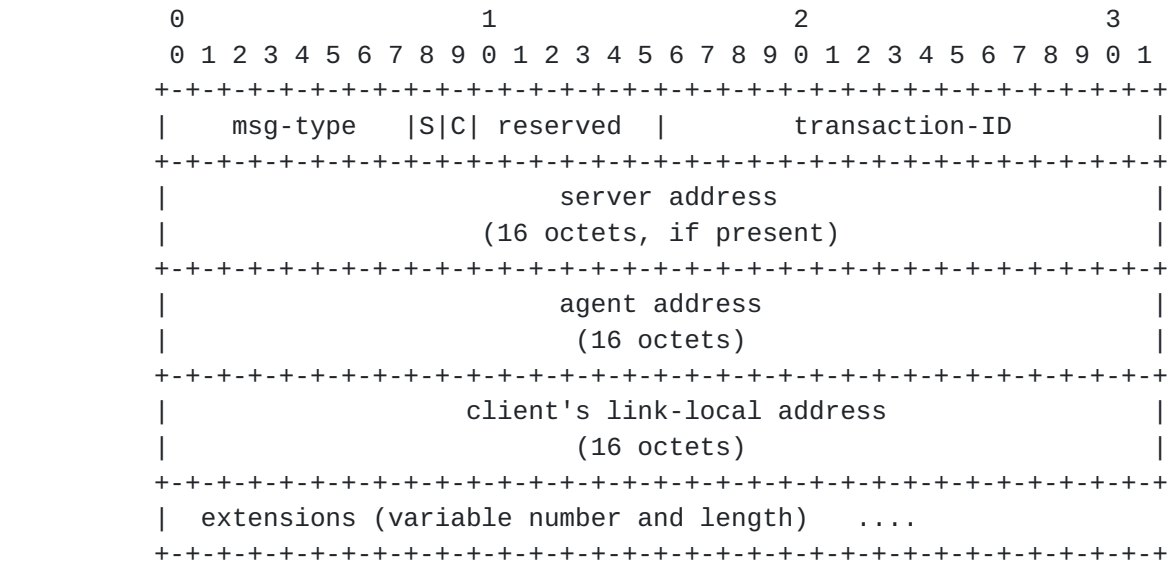
Suppose that a DHCP server on the same link as a client issues the DHCP Advertise in response to a DHCP Solicit message sent to the All-DHCP-Agent multicast address. Then the agent address will be the IP address of one of the server's interfaces, the 'S' bit will be set, the agent address will be an address of the server, and there will be no server address sent in the DHCP Advertise message. It is an error for server-count to be zero if the 'S' bit is not set.

The DHCP Server MUST copy the link-local address into the advertisement which is sent in response to a DHCP Solicit. The source IP address of the IP header of any DHCP Advertise message MUST have sufficient scope to be reachable by the DHCP Client. Moreover, the source address of any DHCP Advertise message sent by a DHCP relay MUST NOT be a link-local address. In situations where there are no routers sending Router Advertisements, then a DHCP Server MUST be configured on the same link as prospective clients.

4.3. DHCP Request Message Format

In order to request parameters from a DHCP server, a client sends a DHCP Request message, and MAY append the appropriate extensions [7]. If the client does not know any DHCP server address, it MUST first obtain a server address by multicasting a DHCP Solicit message (see Section 4.1). If the client does not have a valid IP address of sufficient scope for the DHCP server to communicate with the client, the client MUST use the unicast IP address of a local DHCP relay (which then becomes the agent address in the message header) as the Destination IP address. In this case, the client cannot send the message directly to the DHCP server because the server could not return any response to the client. 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 IP destination address in the IP header.



msg-type 3

S If set, the server address is present

C If set, the client requests the server to clear all other existing resources and bindings (not requested in extensions) currently associated with the client, deallocating as needed.

reserved 0

transaction-ID
A monotonically increasing number used to identify this Request, and copied into the Reply.

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

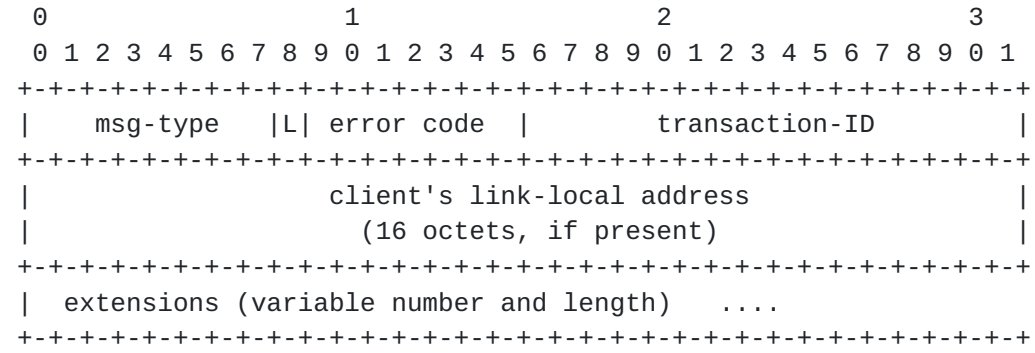
agent address
The IP address of a relay or server interface, copied from a DHCP Advertisement message.

client's link-local address
The IP link-local address of the client interface from which the client issued the DHCP Request message

extensions See [7].

4.4. DHCP Reply Message Format

The server sends one DHCP Reply message in response to every DHCP Request or DHCP Release 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 4

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 or Release
19	Resources unavailable
20	Client record unavailable
21	Invalid client IP address in Release
23	Relay cannot find Server Address
24	Cannot understand selected Character Set
64	Server unreachable (ICMP error)

transaction-ID
A monotonically increasing number used to identify this Reply, and copied from the client's Request.

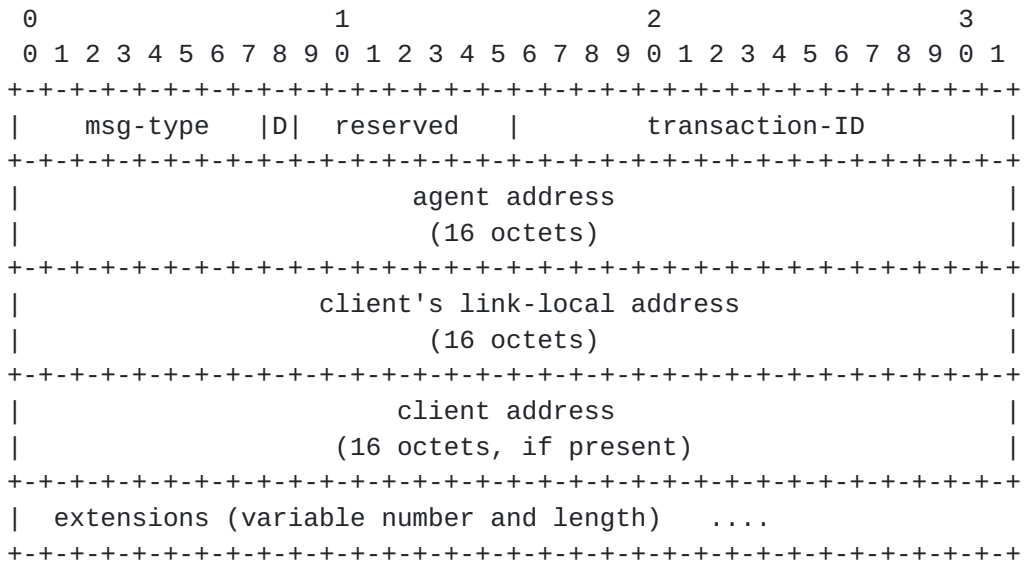
client's link-local address
If present, the IP address of the client interface which issued the corresponding DHCP Request message.

extensions
See [7].

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. If the length in the UDP header preceding the DHCP message does not match that which is expected in the DHCP Request, error code 23 MUST be sent.

4.5. DHCP Release Message Format

The DHCP Release message is sent without the assistance of any DHCP relay. When a client sends a Release message, it is assumed to have a valid IP 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 4.4, 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.

reserved 0

transaction-ID
: A monotonically increasing number used to identify this Release, and copied into the Reply.

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

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

client address
: The IP address of the client interface from which the the client issued the DHCP Request message. The client address field is present whenever the 'D' bit is set, even if it is equal to the link-local address.

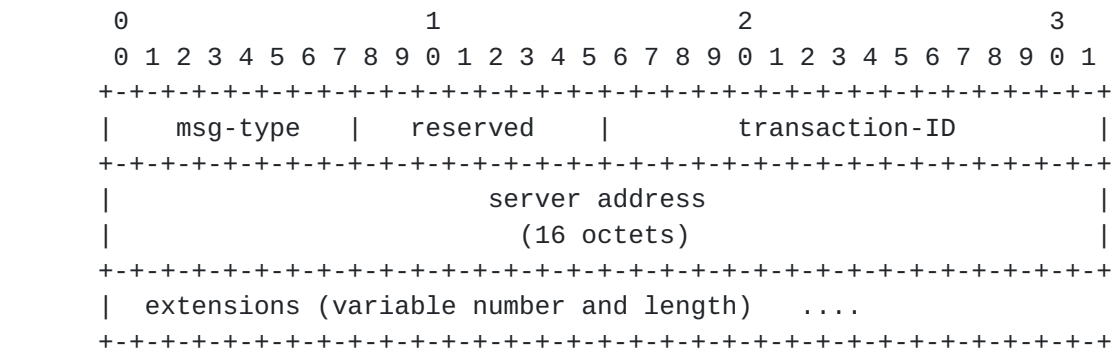
extensions See [7]

Suppose that the client has an IP address that 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 client address field 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.

4.6. DHCP Reconfigure Message Format

The DHCP Reconfigure message is sent without the assistance of any DHCP relay. When a server sends a Reconfigure message, the client to which it is sent is assumed to have a valid IP 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.



msg-type 6

reserved 0

transaction-ID
A monotonically increasing number used to identify this Reconfigure message, and copied into the client's Request.

server address
The IP address of the DHCP server issuing the DHCP Reconfigure message.

extensions See [7]

5. DHCP Client Considerations

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

A DHCP client MAY retain its configured parameters and resources across client system reboots and DHCP client program restarts. However, in these circumstances a DHCP client MUST 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 clean up stale client binding information at the server which may no longer be in use by the client; stale information is that which the client does not include in extensions to such request messages.

If the server does not respond to the DHCP Request message, the client may still use any addresses which have not yet expired. In this case, however, the client MUST begin to search for another

server by multicasting a new DHCP Solicit message, again with the 'C' bit set, containing its IP address in the appropriate extension.

This also handles the case wherein a client restarts on a new network, so that its IP address is no longer valid. When the client multicasts a new DHCP Discover message, servers will respond with the information needed for the client to release its old address, if need be, and request an address reachable on the new network. In this situation, when the client receives a new IP address and the old IP address is no longer reachable, the client MUST release its old IP address by issuing a DHCP Release message with the appropriate extension.

5.1. Sending DHCP Solicit Messages

If a node wishes to become a new DHCP client, it MUST first locate a DHCP Server. The client does this by multicasting a DHCP Solicit message to the All-DHCP-Agents address multicast address FF02:0:0:0:0:0:1:2, setting the Hop Limit == 1. If there are no DHCP servers on the same link as the node, then a DHCP Relay MUST be present for further handling of the solicitation. The prospective client SHOULD wait for ADV_WAIT seconds to get all the DHCP Advertisement messages which may be sent in response to the solicitation.

If a DHCP client reboots and does not have a valid IP address, it MUST set the 'C' bit in the DHCP Solicit message it sends when restarting. By setting the 'C' bit in the solicitation, a DHCP client requests that all the DHCP Servers that receive the solicitation should clean up their stale client records that match its link-local address.

If a client sends a DHCP Solicit message after it reboots, the solicitation SHOULD be delayed after reception of the first Router Advertisement [6] message, by at least some random amount of time between MIN_SOLICIT_DELAY and MAX_SOLICIT_DELAY seconds. This delay is intended to help stagger requests to DHCP Servers (and avoid link-layer collisions) after a power outage causes many nodes to reboot all at once. Each subsequent DHCP Solicit message that is issued before receiving an advertisement MUST be delayed by twice the amount by which the previous DHCP Solicit message was delayed.

5.2. Receiving DHCP Advertise Messages

When a DHCP 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 no server address field 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 DHCP server on the same link as the client. In this case, the client MUST use the agent address as the destination address for any future DHCP message transactions sent to that server.

Advertisements may have extensions; this might allow the DHCP client to select the configuration that best meets its needs from among several prospective servers.

5.3. Sending DHCP Request Messages

A DHCP client obtains configuration information from a DHCP server by sending a DHCP Request message. The client MUST know the server's address before sending the Request message, and client MUST have acquired a (possibly identical) 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 any datagrams were lost due to collisions, etc.

If the client has no valid IP address of sufficient scope, 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 IP destination address of the Request message will be a DHCP relay address.

Otherwise, if the client already has a valid IP address of sufficient scope and knows the IP address of a candidate DHCP server, it SHOULD send the Request message directly to the DHCP server without requiring the services of the local DHCP relay.

If a client wishes to instruct a DHCP server to deallocate all unknown 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.

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 DHCP 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 desired extensions have been applied, the DHCP 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 (which can be the same as the server 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 a DHCP Reply message (Section 5.4) with the same transaction-ID as a pending DHCP Request message within REPLY_MSG_INITIAL_TIMEOUT seconds, or if the received DHCP Reply message contains a DHCP Authentication extension which fails to provide the correct authentication information, the client MUST retransmit the Request with the same transaction-ID and continue to retransmit according to the rules in Section 8.

If the client transmits a DHCP Request in response to a DHCP Reconfigure message (see Section5.6), the client can continue to operate with its existing configuration information and resources until it receives the corresponding DHCP Reply from the server. The same retransmission rules apply as for any other DHCP Request message from the client.

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.

If the Reply message is acceptable, the client processes each Extension [7], extracting the relevant configuration information and parameters for its network operation. The client can determine when all extensions in the Reply have been processed by using the Length field of the Reply. Some extensions in the Reply may have error codes, when the server was unable to honor the request, which

will indicate to the client the reason for failure. If the server is simply unable to honor the request in an extension included by the client, that extension may simply be omitted from the Reply.

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 DHCP Extensions document [7]. These "resource-server" associations are used when sending DHCP Release messages.

5.5. Sending DHCP Release Messages

If a DHCP client determines that some of its network configuration parameters are no longer needed, it SHOULD enable the DHCP server to release allocated resources which are no longer in use by sending a DHCP Release message to the server. The client consults 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 relevant 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.

Suppose a client wishes to release resources which were granted to it on another link. 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 (Error Code 21) to include within the DHCP Release message both the 'D' bit and an IP address extension which has the IP address used as the client IP address field of the DHCP Release message header.

5.6. Receiving DHCP Reconfigure Messages

Each DHCP client MUST 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. In some cases, the IP address at which the client listens will be a multicast address sent to the client by the DHCP server in an extension to an earlier DHCP Reply message. 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 IP address and/or other resources allocated to the client.

See discussion in 6.5. The client MAY receive an update to the prefix for their addresses and then MUST use that prefix for their addresses.

If a DHCP 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 Request message, which it formulates to send to the server specified in the server address field of the message. The client also copies a transaction-ID from the Reconfigure message into the Request message. From then on, processing is the same as specified above in Section 5.3.

Resources held by the client which are not identified by Extensions in the server's Reconfigure message are not affected.

Note that a server may ask its client 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 maintains a collection of client records, called ``bindings''. Each binding is uniquely identifiable by the ordered pair <link-local address, agent address>, since the link-local address is guaranteed to be unique [11] on the link identified by the agent address. An implementation MUST support bindings consisting of at least a client's link-local address, agent address, preferred lifetime and valid lifetime [11] for each client address, and the transaction-ID. A client binding may be used to store any other information, resources, and configuration data which will be associated with the client. A DHCP server MUST retain its clients' bindings across server reboots, and, whenever possible, a DHCP client should be assigned the same configuration parameters despite server system reboots and DHCP server program restarts. A DHCP server MUST

support fixed or permanent allocation of configuration parameters to specific clients.

Servers on the same link as the client MUST use the source address in the IP header from the client as the destination address in DHCP response messages sent by the server to the client.

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. In that case, if the source address is a link-local address, the server MUST silently discard the packet. If the UDP length disagrees with the length determined by the format of the DHCP Solicit message, the server MUST drop the packet and SHOULD log the error. Note that if the client sends a DHCP Solicit message from a link-local address, the multicast destination will be the All-DHCP-Agents address, not the All-DHCP-Servers address.

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 an IP address of the interface on which it received the Solicit message as the source address field of the IP 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.

6.3. DHCP Request and Reply Messages

The DHCP server MUST check to ensure that the client's link-local address field of the Request message contains an address which could be a valid link-local address. 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 IP 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 an error code of 20.

While processing the Request, the server MUST first 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_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, if the transaction-ID has not been recently used, the server identifies and allocates all the relevant information, resources, and configuration data that is associated with the client. Then it sends that information to its DHCP client by constructing a DHCP Reply message and including the client's information in DHCP Extensions to the Reply message. The DHCP Reply message uses the same transaction-ID as found in the received DHCP Request message. Note that the reply message MAY contain information not specifically requested by the client.

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 DHCP 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 DHCP 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. The server sets the Error Code appropriately for those extensions which return error status to the client. The DHCP server sends a single Reply message in response to each DHCP Request, with the same transaction-ID as the Request.

Whenever it is able to, the server includes an extension in the Reply message for every extension sent by the client in the Request message. If the client requests some extensions that cannot be supplied by the server, the server can simply fail to provide them,

not including them in the Reply. Other extensions can be rejected by including them in the Reply with an appropriate error code indicating failure.

When a client DHCP Request is received that has the 'C' bit set, the server should check to find out whether the extensions listed in the Request message match those which it has associated with the client's binding. Any resources which are not indicated by the client are presumed to be unknown by the client, and thus possible candidates for reallocation to satisfy requests from other clients. The DHCP Server MUST deallocate all resources associated with the client upon reception of a DHCP Request with the 'C' bit set, except for those which the server is willing to reallocate response to the client's request. It may be more efficient to avoid deallocating any resources until after the list of extensions to the request have been inspected.

6.4. Receiving DHCP Release Messages

If the server receives a DHCP Release Message, it MUST verify that the link-local address field of the message contains an address which could be a valid link-local address (i.e., one with the prefix FE80::0000/64). If not, the message MUST be silently discarded.

In response to a DHCP Release Message with a valid client's link-local address and agent address, the DHCP 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 an error code of 20.

Otherwise, if the agent address and link-local address indicate a binding known to the server, then the server continues processing 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 DHCP 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. For extensions in the DHCP Release message which cannot be successfully processed by the server, a DHCP Reply message containing extensions with the appropriate error codes MUST be returned by the server.

6.5. Sending DHCP Reconfigure Messages

If a DHCP 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 [7]. The Reconfigure MAY be sent to a multicast address chosen by the server and sent to each of its clients in an extension to a previous DHCP Reply message.

7. DHCP Relay Considerations

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

All relays MUST use the IP address of the interface from which the DHCP request was received as the source address for the IP header of that DHCP message.

The main purpose of the DHCP Relay is to enable clients and servers to carry out DHCP protocol transactions. DHCP Solicit messages are issued by the relay when initiated by prospective DHCP clients. By default, the relay discovers local DHCP Servers by use of multicasting DHCP solicitations to the All-DHCP-Servers multicast address, but relays SHOULD allow this behavior to be configurable. The relay SHOULD NOT send such a multicast solicitation on the interface from which it received the solicitation from the client.

7.1. DHCP Solicit and DHCP Advertise Message Processing

Upon receiving a DHCP Solicit message from a prospective client, a relay, by default, forwards the message to all DHCP Servers at a site according to the following procedure:

- copying the prospective client's solicitation message fields into the appropriate fields of the outgoing solicitation,

- setting the 'A' bit,

- copying the address of its interface from which the solicitation was received from the client into the DHCP Relay address field, and

- finally, sending the resulting message to the All-DHCP-Servers multicast address, FF05:0:0:0:0:0:1:3, over all interfaces except that from which the client's solicitation was received.

When the relay receives a DHCP advertisement with the 'A' bit set, it relays the advertisement to the client at the indicated link-local address by way of the interface indicated in the agent's address field.

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 IP address associated to the interface from which the DHCP Request message was received. If any of these checks fail, the Relay MUST silently discard the Request message.

The relay MUST also check whether the 'S' bit is set in the message header. If not, the datagram is discarded, and and the relay SHOULD return a DHCP Reply message to the source address of the Request message with error code 18.

If the received request message is acceptable, the relay then transmits the DHCP Request message to the address of the DHCP server found in the Server IP 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 IP header will differ from the datagram received from the client, not the payload. If the Relay receives an ICMP error, the Relay SHOULD return a DHCP Reply message to the client address (which can be found in the payload of the ICMP message [2]), with error code 64.

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 IP address that has prefix FE80::0000/64. 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 IP header will differ from the datagram received from the server, not the payload.

8. Retransmission and Configuration Variables

When a DHCP client does not receive a DHCP Reply in response to a pending DHCP Request, the client MUST retransmit the identical DHCP Request, with the same transaction-ID, to the same server again until it can be reasonably sure that the DHCP server is unavailable and an alternative can be chosen. The DHCP Server assumes that the client has received the configuration information included with the extensions to the DHCP Reply message, and it is up to the client to continue to try for a reasonable amount of time to complete the transaction in order to make that assumption hold true. All the actions specified for DHCP Request in this section hold also for DHCP Release messages sent by the DHCP Client.

Similarly, when a client sends a DHCP Request message in response to a Reconfigure message from the server, the client assumes that the DHCP server has received the Request. The server MUST retransmit the identical DHCP Reconfigure to the client for a reasonable amount of time, to try to elicit the Request message from the client, in order to make the best effort for that assumption to hold true. 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 DHCP implementation that MUST be configurable by a client or server:

ADV_WAIT

The amount of time a client waits to hear DHCP Advertisements after issuing a DHCP Solicit to the All-DHCP Agents multicast address.

Default: 5 seconds

REPLY_MSG_INITIAL_TIMEOUT

The time in seconds that a DHCP 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 DHCP client should retransmit, before aborting the request, possibly retrying the Request with another Server, and logging a DHCP 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 DHCP 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 DHCP 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 DHCP System Error.

Default: 10 retransmissions.

RECONF_MSG_RETRANS_INTERVAL

The least 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: 2 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: 10 seconds.

MIN_SOLICIT_DELAY

The maximum amount of time a prospective client is required to wait, after determining from a Router Discovery message that the client should perform stateful address configuration, before sending a DHCP Solicit to a DHCP Server.

Default: 1 second

MAX_SOLICIT_DELAY

The maximum amount of time a prospective client is required to wait, after determining from a Router Discovery message that the client should perform stateful address configuration, before sending a DHCP Solicit to a DHCP Server.

Default: 5 seconds

XID_TIMEOUT

The amount of time a DHCP server has to keep track of client transaction-IDs in order to make sure that client retransmissions using the same transaction-ID are idempotent.

Default: 600 seconds

Note that, if a client receives a DHCP message which fails authentication, it should continue to wait for another message which might be correctly authenticated just as if the failed message had never arrived; however, receiving such failed messages SHOULD be logged.

9. Security Considerations

DHCP clients and servers often have 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 IP address. However, a client often has only a link-local address, and such an address is not sufficient for a DHCP server which is off-link. In those circumstances the DHCP relay is involved, so that the DHCP message MUST have the relay's address in the IP destination address field, even though the client aims to deliver the message to the DHCP server. The DHCP Client-Server Authentication Extension [7] is intended to be used in these circumstances.

10. 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. Thanks to Stuart Cheshire for his excellent minutes.

A. Related Work in IPv6

The related work in IPv6 that would best serve an implementor to study is the IPv6 Specification [3], the IPv6 Addressing Architecture [4], IPv6 Stateless Address Autoconfiguration [11], IPv6 Neighbor Discovery Processing [6], and Dynamic Updates to DNS [12]. 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 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, so that fragmentation takes place end-to-end between hosts. If a DHCP implementation needs to send a datagram greater than 536 octets it can either fragment the UDP datagram in UDP or use Path MTU Discovery [5] to determine the size of the datagram that will traverse a network path. It is implementation dependent how this is accomplished in DHCP.

The IPv6 Addressing Architecture specification [4] 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 multicast addressing is required, and nodes can create link-local addresses during initialization of the nodes environment. This means that a client immediately can configure an IP address at initialization for an interface, before communicating in any manner on the link. The client 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 DHCP server or relay.

IPv6 Stateless Address Autoconfiguration [11] (addrconf) specifies procedures by which a node may autoconfigure addresses based on router advertisements [6], and the use of a validation 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 3.1).

IPv6 Neighbor Discovery [6] is the node discovery protocol in IPv6 (replaces and enhances functions of ARP [8]). To truly understand IPv6 and addrconf it is strongly recommended that implementors understand IPv6 Neighbor Discovery.

Dynamic Updates to DNS [12] 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 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.

Added initial capability for the DHCP Relay to multicast and obtain DHCP Server addresses.

Added capability for Servers to add Extensions to their Advertisements.

Added 'C' bit to DHCP Solicit for deallocating resources after client crash.

Added DHCP Advertisement lifetimes for use by DHCP Relay agents that need to periodically update their list of DHCP servers.

B.3. Changes from June 96 to August 96 Drafts

Since the working group indicated that DHCP solicitation traffic was not considered to be a significant factor affecting network load, it was decided to modify the handling of solicitations so

that DHCP relays, by default, multicast DHCP Solicit message to all DHCP servers at a site. This entailed a number of changes to the protocol, namely:

- Adding fields to the DHCP Solicit and DHCP Advertise messages to contain the DHCP client's link-local addresses.

- Adding the 'L' bit to the DHCP Solicit and DHCP Advertise messages to indicate whether the link-local address is present

- Adding a 'P' bit to the DHCP Solicit message so that the client can allow the Relay to use its non-default behavior, which is to return cached DHCP Server addresses to the client in response to the client's DHCP Solicit message.

- Specified a new multicast address (the All-DHCP-Servers address) for use by DHCP Relays when "relaying" client solicitations.

Added a random backoff after reboot so that clients' solicitations don't immediately swamp DHCP Servers after power outages.

Added new multicast addresses for All DHCP Servers and All DHCP Relays.

B.4. Changes from August 96 to November 96 Drafts

Clarified language regarding treatment by the DHCP server of DHCP Requests with the 'C' bit set.

Specified that the UDP source port for DHCP messages is arbitrary.

Added description for Appendix C.

Changed must to MUST where appropriate.

Changed definitions for client, server, and relay to be definitions for DHCP client, DHCP server, and DHCP relay.

Changed definitions of DHCP multicast addresses to conform to recent IANA allocations.

Corrected references to "leases", to more accurately refer to IPv6 address lifetimes.

B.5. Changes from November 96 to February 97 Drafts

Clients can continue to use valid addresses, after restarts or any request triggered by a DHCP Reconfigure message, at least until it receives the DHCP Request from the server.

All extensions sent in response to a single DHCP Request now must be part of the same DHCP Reply message. If some requested resources and configuration parameters are not available or cannot be allocated, each particular extension will either have the appropriate error code indicating the particular problem, or simply will not be included in the Reply. The extensions are to be modified to have fields for error codes whenever the server might have to indicate to the client a reason why the information requested in its extension was unable to be supplied.

If a client receives a DHCP Reconfigure message which does not list some the client's configuration information, it can continue to assume that configuration information is valid.

If a client reboots, it MUST set the 'C' bit and transmit a DHCP Request. If it doesn't have a valid server address, it MUST set the 'C' bit in its DHCP Solicit message.

Relays are no longer allowed to cache server addresses. The DHC working group decided to ice this plan until there was some determination that it might be useful. This caused the elimination of the 'P' bit, and quite a bit of discussion about the 'P' bit and DHCP server address caching was eliminated. The 'server count' field of the Advertisement and the lifetime field were eliminated, since relays never keep track of server addresses and clients have to solicit again whenever they lose their DHCP server.

The working group decided to make programming as simple as possible, and therefore to include IP addresses in the appropriate DHCP message headers whenever those addresses would otherwise have to be discovered by manipulating the IP header itself. This caused many changes to the message header formats. The 'L' bit in the DHCP Solicit and DHCP Advertise messages is no longer necessary, because the link-local address of the client is always present in the header.

Previously, there was language which required the client to match pending Requests with Reply messages with the same destination agent addresses. Those agent addresses were to be determined by inspecting the IP headers of the DHCP Reply messages. We deleted the requirement, in preference to loading possibly two more agent addresses in every DHCP Advertise message and DHCP Reply message.

The DHCP Reconfigure message now has a transaction ID which the client copies into the corresponding DHCP Request, and then which subsequently the server copies again into the corresponding DHCP Reply message.

Clients now use the DHCP server address found in the appropriate field of the DHCP Reconfigure message header instead of inspecting the IP header of the Reconfigure message.

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. There are three key reasons for the differences:

- IPv6 inherently supports a new model and architecture for communications and autoconfiguration of addresses.
- DHCPv6 in its design was able to take advantage of the inherent benefits of IPv6.
- New features were added to support the evolution and the existence of mature Internet users in the industry.

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 a server or relay agent on the local link.
- 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.
- 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.
- 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.
- Multiple addresses per interface are inherently supported in IPv6.

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

DHCPv6 Architecture/Model Changes:

- The message type is the first byte in the packet.

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

- 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 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.

- 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 additional addresses for server applications can be requested by a client in an implementation.
- Reclaiming addresses allocated with very long lifetimes 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] A. Conta and S. Deering. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6). RFC 1885, December 1995.

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

[4] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. RFC 1884, December 1995.

[5] J. McCann, S. Deering, and J. Mogul. Path MTU Discovery for IP version 6. RFC 1981, August 1996.

[6] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP version 6 (IPv6). RFC 1970, August 1996.

[7] C. Perkins. Extensions to DHCPv6, February 1997. draft-ietf-dhc-dhcpv6ext-05.txt, work in progress.

[8] 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.

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

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

[11] S. Thomson and T. Narten. IPv6 stateless address autoconfiguration. RFC 1971, August 1996.

[12] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS). draft-ietf-dnsind-dynDNS-11.txt, November 1996. (work in progress).

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
Netcentricity Group
Sun Microsystems, Inc.
2550 Garcia Avenue.
Mountain View, CA 94043
+1-415-336-7153
+1-415-336-0673
charles.perkins@corp.sun.com