

## Dynamic Host Configuration Protocol

### Status of this memo

This RFC specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Abstract

The Dynamic Host Configuration Protocol (DHCP) provides a framework for passing configuration information to hosts on a TCP/IP network. DHCP is based on the Bootstrap Protocol (BOOTP) [7], adding the capability of automatic allocation of reusable network addresses and additional configuration options [19]. DHCP captures the behavior of BOOTP relay agents [7, 23], and DHCP participants can interoperate with BOOTP participants [9].

### Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .   | 2  |
| 1.1 Related Work . . . . .  | 4  |
| 1.2 Problem definition and issues . . . . .   | 4  |
| 1.3 Requirements . . . . .  | 5  |
| 1.4 Terminology . . . . .   | 6  |
| 1.5 Design goals . . . . .  | 6  |
| 2. Protocol Summary . . . . .   | 8  |
| 2.1 Configuration parameters repository . . . . .   | 10 |
| 2.2 Dynamic allocation of network addresses . . . . .                                       | 11 |
| 3. The Client-Server Protocol . . . . .   | 11 |
| 3.1 Client-server interaction - allocating a network address . . . . .                      | 12 |
| 3.2 Client-server interaction - reusing a previously allocated<br>network address . . . . . | 17 |
| 3.3 Interpretation and representation of time values . . . . .                              | 19 |
| 3.4 Host parameters in DHCP . . . . .   | 19 |
| 3.5 Use of DHCP in clients with multiple interfaces . . . . .                               | 20 |
| 3.6 When clients should use DHCP . . . . .  | 20 |
| 4. Specification of the DHCP client-server protocol . . . . .                               | 21 |
| 4.1 Constructing and sending DHCP messages . . . . .  | 21 |
| 4.2 DHCP server administrative controls . . . . .   | 23 |
| 4.3 DHCP server behavior . . . . .  | 24 |

|  |    |
|--|----|
| 4.3.1 DHCPDISCOVER message . . . . .                             | 24 |
| 4.3.2 DHCPREQUEST message . . . . .                              | 27 |
| 4.3.3 DHCPDECLINE message . . . . .                              | 29 |
| 4.3.4 DHCPRELEASE message . . . . .                              | 29 |
| 4.4 DHCP client behavior . . . . .                               | 29 |
| 4.4.1 Initialization and allocation of network address . . . . . | 29 |
| 4.4.2 Initialization with known network address . . . . .        | 33 |
| 4.4.3 Initialization with a known DHCP server address . . . . .  | 34 |
| 4.4.4 Reacquisition and expiration . . . . .                     | 34 |
| 4.4.5 DHCPRELEASE . . . . .                                      | 35 |
| 5. Acknowledgments . . . . .                                     | 35 |
| 6. References . . . . .  | 36 |
| 7. Security Considerations . . . . .                             | 37 |
| 8. Author's Address . . . . .                                    | 38 |
| A. Host Configuration Parameters . . . . .                       | 39 |

## List of Figures

|   |    |
|---|----|
| 1. Format of a DHCP message . . . . .   | 9  |
| 2. Format of the 'flags' field . . . . .  | 10 |
| 3. Timeline diagram of messages exchanged between DHCP client and servers when allocating a new network address . . . . .               | 15 |
| 4. Timeline diagram of messages exchanged between DHCP client and servers when reusing a previously allocated network address . . . . . | 18 |
| 5. State-transition diagram for DHCP clients . . . . .  | 31 |

## List of Tables

|  |    |
|--|----|
| 1. Description of fields in a DHCP message . . . . . | 14 |
| 2. DHCP messages . . . . .                           | 16 |
| 3. Fields and options used by DHCP servers . . . . . | 25 |
| 4. Fields and options used by DHCP clients . . . . . | 32 |

## 1. Introduction

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

DHCP is built on a client-server model, where designated DHCP server hosts allocate network addresses and 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.

A host should not act as a DHCP server unless explicitly configured to do so by a system administrator. The diversity of hardware and protocol implementations in the Internet would preclude reliable operation if random hosts were allowed to respond to DHCP requests. For example, IP requires the setting of many parameters within the protocol implementation software. Because IP can be used on many dissimilar kinds of network hardware, values for those parameters cannot be guessed or assumed to have correct defaults. Also, distributed address allocation schemes depend on a polling/defense mechanism for discovery of addresses that are already in use. IP hosts may not always be able to defend their network addresses, so that such a distributed address allocation scheme cannot be guaranteed to avoid allocation of duplicate network addresses.

DHCP supports three mechanisms for IP address allocation. In "automatic allocation", DHCP assigns a permanent IP address to a host. In "dynamic allocation", DHCP assigns an IP address to a host for a limited period of time (or until the host explicitly relinquishes the address). In "manual allocation", a host's IP address is assigned by the network administrator, and DHCP is used simply to convey the assigned address to the host. A particular network will use one or more of these mechanisms, depending on the policies of the network administrator.

Dynamic allocation is the only one of the three mechanisms that allows automatic reuse of an address that is no longer needed by the host to which it was assigned. Thus, dynamic allocation is particularly useful for assigning an address to a host that will be connected to the network only temporarily or for sharing a limited pool of IP addresses among a group of hosts that do not need permanent IP addresses. Dynamic allocation may also be a good choice for assigning an IP address to a new host being permanently connected to a network where IP addresses are sufficiently scarce that it is important to reclaim them when old hosts are retired. Manual allocation allows DHCP to be used to eliminate the error-prone process of manually configuring hosts with IP addresses in environments where (for whatever reasons) it is desirable to manage IP address assignment outside of the DHCP mechanisms.

The format of DHCP messages is based on the format of BOOTP messages, to capture the BOOTP relay agent behavior described as part of the BOOTP specification [7, 23] and to allow interoperability of existing BOOTP clients with DHCP servers. Using BOOTP relaying agents eliminates the necessity of having a DHCP server on each physical network segment.

## 1.1 Related Work

There are several Internet protocols and related mechanisms that address some parts of the dynamic host configuration problem. The Reverse Address Resolution Protocol (RARP) [10] (through the extensions defined in the Dynamic RARP (DRARP) [5]) explicitly addresses the problem of network address discovery, and includes an automatic IP address assignment mechanism. The Trivial File Transfer Protocol (TFTP) [20] provides for transport of a boot image from a boot server. The Internet Control Message Protocol (ICMP) [16] provides for informing hosts of additional routers via "ICMP redirect" messages. ICMP also can provide subnet mask information through the "ICMP mask request" message and other information through the (obsolete) "ICMP information request" message. Hosts can locate routers through the ICMP router discovery mechanism [8].

BOOTP is a transport mechanism for a collection of configuration information. BOOTP is also extensible, and official extensions [17] have been defined for several configuration parameters. Morgan has proposed extensions to BOOTP for dynamic IP address assignment [15]. The Network Information Protocol (NIP), used by the Athena project at MIT, is a distributed mechanism for dynamic IP address assignment [19]. The Resource Location Protocol RLP [1] provides for location of higher level services. Sun Microsystems diskless workstations use a boot procedure that employs RARP, TFTP and an RPC mechanism called "bootparams" to deliver configuration information and operating system code to diskless hosts. (Sun Microsystems, Sun Workstation and SunOS are trademarks of Sun Microsystems, Inc.) Some Sun networks also use DRARP and an auto-installation mechanism to automate the configuration of new hosts in an existing network.

In other related work, the path minimum transmission unit (MTU) discovery algorithm can determine the MTU of an arbitrary internet path [14]. Comer and Droms have proposed the use of the Address Resolution Protocol (ARP) as a transport protocol for resource location and selection [6]. Finally, the Host Requirements RFCs [3, 4] mention specific requirements for host reconfiguration and suggest a scenario for initial configuration of diskless hosts.

## 1.2 Problem definition and issues

DHCP is designed to supply hosts with the configuration parameters defined in the Host Requirements RFCs. After obtaining parameters via DHCP, a host should be able to exchange packets with any other host in the Internet. The parameters supplied by DHCP are listed in Appendix A.

Not all of these parameters are required for a newly initialized host. A client and server may negotiate for the transmission of only those parameters required by the client or specific to a particular subnet.

DHCP allows but does not require the configuration of host parameters not directly related to the IP protocol. DHCP also does not address registration of newly configured hosts with the Domain Name System (DNS) [12, 13].

DHCP is not intended for use in configuring routers.

### 1.3 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

- o "MUST"

This word or the adjective "REQUIRED" means that the item is an absolute requirement of this specification.

- o "MUST NOT"

This phrase means that the item is an absolute prohibition of this specification.

- o "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 should be understood and the case carefully weighed before choosing a different course.

- o "SHOULD NOT"

This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

- o "MAY"

This word or the adjective "OPTIONAL" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

#### 1.4 Terminology

This document uses the following terms:

- o "DHCP client"

A DHCP client is an Internet host using DHCP to obtain configuration parameters such as a network address.

- o "DHCP server"

A DHCP server is an Internet host that returns configuration parameters to DHCP clients.

- o "BOOTP relay agent"

A BOOTP relay agent is an Internet host or router that passes DHCP messages between DHCP clients and DHCP servers. DHCP is designed to use the same relay agent behavior as specified in the BOOTP protocol specification.

- o "binding"

A binding is a collection of configuration parameters, including at least an IP address, associated with or "bound to" a DHCP client. Bindings are managed by DHCP servers.

#### 1.5 Design goals

The following list gives general design goals for DHCP.

- o 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.

- o Hosts should require no manual configuration. Each host should be able to discover appropriate local configuration parameters without user intervention and incorporate those parameters into its own configuration.
- o Networks should require no hand configuration for individual hosts. Under normal circumstances, the network manager should not have to enter any per-host configuration parameters.
- o DHCP should not require a server on each subnet. To allow for scale and economy, DHCP must work across routers or through the intervention of BOOTP/DHCP relay agents.
- o A DHCP host must be prepared to receive multiple responses to a request for configuration parameters. Some installations may include multiple, overlapping DHCP servers to enhance reliability and increase performance.
- o DHCP must coexist with statically configured, non-participating hosts and with existing network protocol implementations.
- o DHCP must interoperate with the BOOTP relay agent behavior as described by RFC 951 and by Wimer [21].
- o DHCP must provide service to existing BOOTP clients.

The following list gives design goals specific to the transmission of the network layer parameters. DHCP must:

- o Guarantee that any specific network address will not be in use by more than one host at a time,
- o Retain host configuration across host reboot. A host should, whenever possible, be assigned the same configuration parameters (e.g., network address) in response to each request,
- o Retain host configuration across server reboots, and, whenever possible, a host should be assigned the same configuration parameters despite restarts of the DHCP mechanism,
- o Allow automatic assignment of configuration parameters to new hosts to avoid handconfiguration for new hosts,
- o Support fixed or permanent allocation of configuration parameters to specific hosts.

## 2. Protocol Summary

From the client's point of view, DHCP is an extension of the BOOTP mechanism. This behavior allows existing BOOTP clients to interoperate with DHCP servers without requiring any change to the clients' initialization software. A separate document details the interactions between BOOTP and DHCP clients and servers [9]. There are some new, optional transactions that optimize the interaction between DHCP clients and servers that are described in sections 3 and 4.

Figure 1 gives the format of a DHCP message and table 1 describes each of the fields in the DHCP message. The numbers in parentheses indicate the size of each field in octets. The names for the fields given in the figure will be used throughout this document to refer to the fields in DHCP messages.

There are two primary differences between DHCP and BOOTP. First, DHCP defines mechanisms through which clients can be assigned a network address for a fixed lease, allowing for serial reassignment of network addresses to different clients. Second, DHCP provides the mechanism for a client to acquire all of the IP configuration parameters that it needs in order to operate.

DHCP introduces a small change in terminology intended to clarify the meaning of one of the fields. What was the "vendor extensions" field in BOOTP has been re-named the "options" field in DHCP. Similarly, the tagged data items that were used inside the BOOTP "vendor extensions" field, which were formerly referred to as "vendor extensions," are now termed simply "options."

DHCP defines a new 'client identifier' option that is used to pass an explicit client identifier to a DHCP server. This change eliminates the overloading of the 'chaddr' field in BOOTP messages, where reply messages and as a client identifier. The 'client identifier' option may contain a hardware address, identical to the contents of the 'chaddr' field, or it may contain another type of identifier, such as a DNS name. Other client identifier types may be defined as needed for use with DHCP. New client identifier types will be registered with the IANA [18] and will be included in new revisions of the Assigned Numbers document, as well as described in detail in future revisions of the DHCP Options [2].



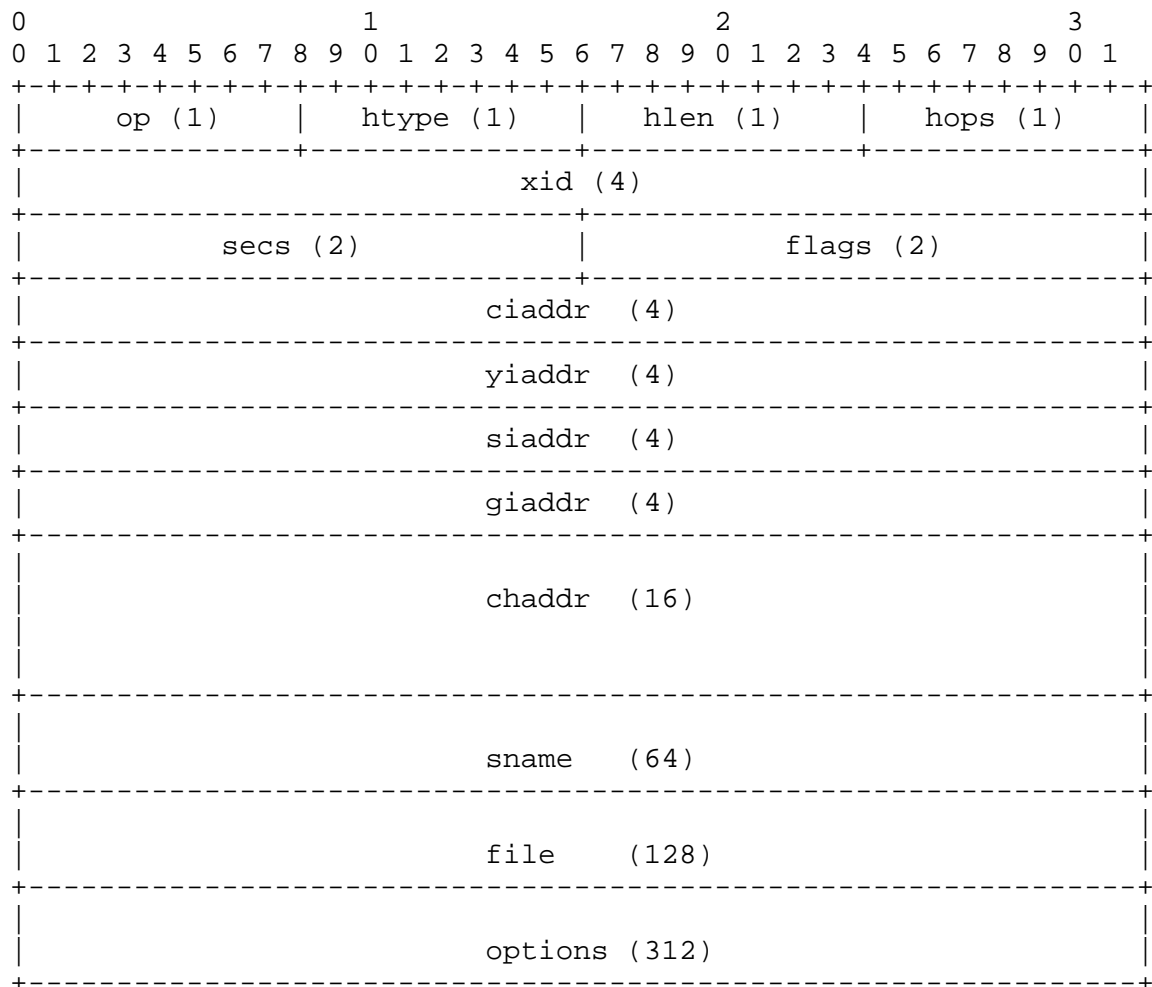


Figure 1: Format of a DHCP message

DHCP clarifies the interpretation of the 'siaddr' field as the address of the server to use in the next step of the client's bootstrap process. A DHCP server may return its own address in the 'siaddr' field, if the server is prepared to supply the next bootstrap service (e.g., delivery of an operating system executable image). A DHCP server always returns its own address in the 'server identifier' option.

The options field is now variable length, with the minimum extended to 312 octets. This brings the minimum size of a DHCP message up to 576 octets, the minimum IP datagram size a host must be prepared to accept [3]. DHCP clients may negotiate the use of larger DHCP messages through the 'Maximum DHCP message size' option. The options field may be further extended into the 'file' and 'sname' fields.

A new option, called 'vendor specific information', has been added to allow for expansion of the number of options that can be supported [2]. Options encapsulated as 'vendor specific information' must be carefully defined and documented so as to allow for interoperability between clients and servers from different vendors. In particular, vendors defining 'vendor specific information' MUST document those options in the form of the DHCP Options document, MUST choose to represent those options either in data types already defined for DHCP options or in other well-defined data types, and MUST choose options that can be readily encoded in configuration files for exchange with servers provided by other vendors. Options included as 'vendor specific options' MUST be readily supportable by all servers.



B: BROADCAST flag

MBZ: MUST BE ZERO (reserved for future use)

Figure 2: Format of the 'flags' field

DHCP uses the 'flags' field [21]. The leftmost bit is defined as the BROADCAST (B) flag. The semantics of this flag are discussed in section 4.1 of this document. The remaining bits of the flags field are reserved for future use. They MUST be set to zero by clients and ignored by servers and relay agents. Figure 2 gives the format of the

## 2.1 Configuration parameters repository

The first service provided by DHCP is to provide persistent storage of network parameters for network clients. The model of DHCP persistent storage is that the DHCP service stores a key-value entry for each client, where the key is some unique identifier (for example, an IP subnet number and a unique identifier within the subnet) and the value contains the configuration parameters for the client.

For example, the key might be the pair (IP-subnet-number, hardware-address), allowing for serial or concurrent reuse of a hardware address on different subnets, and for hardware addresses that may not be globally unique. Alternately, the key might be the pair (IP-subnet-number, hostname), allowing the server to assign parameters intelligently to a host that has been moved to a different subnet or

has changed hardware addresses (perhaps because the network interface failed and was replaced).

A client can query the DHCP service to retrieve its configuration parameters. The client interface to the configuration parameters repository consists of protocol messages to request configuration parameters and responses from the server carrying the configuration parameters.

## 2.2 Dynamic allocation of network addresses

The second service provided by DHCP is the allocation of temporary or permanent network (IP) addresses to hosts. The basic mechanism for the dynamic allocation of network addresses is simple: a client requests the use of an address for some period of time. The allocation mechanism (the collection of DHCP servers) guarantees not to reallocate that address within the requested time and attempts to return the same network address each time the client requests an address. In this document, the period over which a network address is allocated to a client is referred to as a "lease" [11]. The client may extend its lease with subsequent requests. The client may issue a message to release the address back to the server when the client no longer needs the address. The client may ask for a permanent assignment by asking for an infinite lease. Even when assigning "permanent" addresses, a server may choose to give out lengthy but non-infinite leases to allow detection of the fact that the host has been retired.

In some environments it will be necessary to reassign network addresses due to exhaustion of available addresses. In such environments, the allocation mechanism will reuse addresses whose lease has expired. The server should use whatever information is available in the configuration information repository to choose an address to reuse. For example, the server may choose the least recently assigned address. As a consistency check, the allocation mechanism may probe the reused address, e.g., with an ICMP echo request, before allocating the address, and the client will probe the newly received address, e.g., with ARP.

## 3. The Client-Server Protocol

DHCP uses the BOOTP message format defined in RFC 951 and given in table 1 and figure 1. The 'op' field of each DHCP message sent from a client to a server contains BOOTREQUEST. BOOTREPLY is used in the 'op' field of each DHCP message sent from a server to a client.

The first four octets of the 'options' field of the DHCP message contain the (decimal) values 99, 130, 83 and 99, respectively (this

is the same magic cookie as is defined in RFC 1395). The remainder of the 'options' field consists a list of tagged parameters that are called "options". All of the "vendor extensions" listed in RFC 1395 are also DHCP options. A separate document gives the complete set of options defined for use with DHCP [2].

Several options have been defined so far. One particular option - the "DHCP message type" option - must be included in every DHCP message. This option defines the "type" of the DHCP message. Additional options may be allowed, required, or not allowed, depending on the DHCP message type.

Throughout this document, DHCP messages that include a 'DHCP message type' option will be referred to by the type of the message; e.g., a DHCP message with 'DHCP message type' option type 1 will be referred to as a "DHCPDISCOVER" message.

### 3.1 Client-server interaction - allocating a network address

The following summary of the protocol exchanges between clients and servers refers to the DHCP messages described in table 2. The timeline diagram in figure 3 shows the timing relationships in a typical client-server interaction. If the client already knows its address, some steps may be omitted; this abbreviated interaction is described in section 3.2.

1. The client broadcasts a DHCPDISCOVER message on its local physical subnet. The DHCPDISCOVER message may include options that suggest values for the network address and lease duration. BOOTP relay agents may pass the message on to DHCP servers not on the same physical subnet.
2. Each server may respond with a DHCPOFFER message that includes an available network address in the 'yiaddr' field (and other configuration parameters in DHCP options). Servers need not reserve the offered network address, although the protocol will work more efficiently if the server avoids allocating the offered network address to another client. The server unicasts the DHCPOFFER message to the client (using the DHCP/BOOTP relay agent if necessary) if possible, or may broadcast the message to a broadcast address (preferably 255.255.255.255) on the client's subnet.
3. The client receives one or more DHCPOFFER messages from one or more servers. The client may choose to wait for multiple responses. The client chooses one server from which to request configuration parameters, based on the configuration parameters offered in the DHCPOFFER messages. The client broadcasts a

DHCPREQUEST message that MUST include the 'server identifier' option to indicate which server it has selected, and may include other options specifying desired configuration values. This DHCPREQUEST message is broadcast and relayed through DHCP/BOOTP relay agents. To help ensure that any DHCP/BOOTP relay agents forward the DHCPREQUEST message to the same set of DHCP servers that received the original DHCPDISCOVER message, the DHCPREQUEST message must use the same value in the DHCP message header's 'secs' field and be sent to the same IP broadcast address as the original DHCPDISCOVER message. The client times out and retransmits the DHCPDISCOVER message if the client receives no DHCPOFFER messages.

4. The servers receive the DHCPREQUEST broadcast from the client. Those servers not selected by the DHCPREQUEST message use the message as notification that the client has declined that server's offer. The server selected in the DHCPREQUEST message commits the binding for the client to persistent storage and responds with a DHCPACK message containing the configuration parameters for the requesting client. The combination of 'chaddr' and assigned network address constitute a unique identifier for the client's lease and are used by both the client and server to identify a lease referred to in any DHCP messages. The 'yiaddr' field in the DHCPACK message is filled in with the selected network address.

If the selected server is unable to satisfy the DHCPREQUEST message (e.g., the requested network address has been allocated), the server SHOULD respond with a DHCPNAK message.

A server may choose to mark addresses offered to clients in DHCPOFFER messages as unavailable. The server should mark an address offered to a client in a DHCPOFFER message as available if the server receives no DHCPREQUEST message from that client.

| FIELD<br>----- | OCTETS<br>----- | DESCRIPTION<br>-----   |
|----------------|-----------------|--|
| op             | 1               | Message op code / message type.<br>1 = BOOTREQUEST, 2 = BOOTREPLY  |
| htype          | 1               | Hardware address type, see ARP section in "Assigned Numbers" RFC; e.g., '1' = 10mb ethernet.   |
| hlen           | 1               | Hardware address length (e.g. '6' for 10mb ethernet).  |
| hops           | 1               | Client sets to zero, optionally used by relay-agents when booting via a relay-agent.   |
| xid            | 4               | Transaction ID, a random number chosen by the client, used by the client and server to associate messages and responses between a client and a server. |
| secs           | 2               | Filled in by client, seconds elapsed since client started trying to boot.  |
| flags          | 2               | Flags (see figure 2).  |
| ciaddr         | 4               | Client IP address; filled in by client in DHCPREQUEST if verifying previously allocated configuration parameters.                                      |
| yiaddr         | 4               | 'your' (client) IP address.  |
| siaddr         | 4               | IP address of next server to use in bootstrap; returned in DHCPOFFER, DHCPACK and DHCPNAK by server.   |
| giaddr         | 4               | Relay agent IP address, used in booting via a relay-agent.   |
| chaddr         | 16              | Client hardware address.   |
| sname          | 64              | Optional server host name, null terminated string.   |
| file           | 128             | Boot file name, null terminated string; "generic" name or null in DHCPDISCOVER, fully qualified directory-path name in DHCPOFFER.                      |
| options        | 312             | Optional parameters field. See the options documents for a list of defined options.  |

Table 1: Description of fields in a DHCP message



Figure 3: Timeline diagram of messages exchanged between DHCP client and servers when allocating a new network address

| Message<br>----- | Use<br>---   |
|------------------|--|
| DHCPDISCOVER     | - Client broadcast to locate available servers.  |
| DHCPOFFER        | - Server to client in response to DHCPDISCOVER with offer of configuration parameters.                                       |
| DHCPREQUEST      | - Client broadcast to servers requesting offered parameters from one server and implicitly declining offers from all others. |
| DHCPACK          | - Server to client with configuration parameters, including committed network address.                                       |
| DHCPNAK          | - Server to client refusing request for configuration parameters (e.g., requested network address already allocated).        |
| DHCPDECLINE      | - Client to server indicating configuration parameters (e.g., network address) invalid.                                      |
| DHCPRELEASE      | - Client to server relinquishing network address and cancelling remaining lease.   |

Table 2: DHCP messages

5. The client receives the DHCPACK message with configuration parameters. The client performs a final check on the parameters (e.g., ARP for allocated network address), and notes the duration of the lease and the lease identification cookie specified in the DHCPACK message. At this point, the client is configured. If the client detects a problem with the parameters in the DHCPACK message, the client sends a DHCPDECLINE message to the server and restarts the configuration process. The client should wait a minimum of ten seconds before restarting the configuration process to avoid excessive network traffic in case of looping.

If the client receives a DHCPNAK message, the client restarts the configuration process.

The client times out and retransmits the DHCPREQUEST message if the client receives neither a DHCPACK or a DHCPNAK message. The client retransmits the DHCPREQUEST according to the retransmission algorithm in section 4.1. If the client receives neither a DHCPACK or a DHCPNAK message after ten retransmissions of the DHCPREQUEST message, the client reverts to INIT state and restarts the initialization process. The client SHOULD notify the user that the



initialization process has failed and is restarting.

6. The client may choose to relinquish its lease on a network address by sending a DHCPRELEASE message to the server. The client identifies the lease to be released by including its network address in the 'ciaddr' field and its hardware address in the 'chaddr' field.

### 3.2 Client-server interaction - reusing a previously allocated network address

If a client remembers and wishes to reuse a previously allocated network address (allocated either by DHCP or some means outside the protocol), a client may choose to omit some of the steps described in the previous section. The timeline diagram in figure 4 shows the timing relationships in a typical client-server interaction for a client reusing a previously allocated network address.

1. The client broadcasts a DHCPREQUEST message on its local subnet. The DHCPREQUEST message includes the client's network address in the 'ciaddr' field. DHCP/BOOTP relay agents pass the message on to DHCP servers not on the same subnet.
2. Servers with knowledge of the client's configuration parameters respond with a DHCPACK message to the client.

If the client's request is invalid (e.g., the client has moved to a new subnet), servers may respond with a DHCPNAK message to the client.

3. The client receives the DHCPACK message with configuration parameters. The client performs a final check on the parameters (as in section 3.1), and notes the duration of the lease and the lease identification cookie specified in the DHCPACK message. At this point, the client is configured.

If the client detects a problem with the parameters in the DHCPACK message, the client sends a DHCPDECLINE message to the server and restarts the configuration process by requesting a new network address. This action corresponds to the client moving to the INIT state in the DHCP state diagram, which is described in section 4.4.



Figure 4: Timeline diagram of messages exchanged between DHCP client and servers when reusing a previously allocated network address

If the client receives a `DHCPNAK` message, it cannot reuse its remembered network address. It must instead request a new address by restarting the configuration process, this time using the (non-abbreviated) procedure described in section 3.1. This action also corresponds to the client moving to the `INIT` state in the DHCP state diagram.

The client times out and retransmits the `DHCPREQUEST` message if the client receives neither a `DHCPACK` nor a `DHCPNAK` message. The time between retransmission **MUST** be chosen according to the algorithm given in section 4.1. If the client receives no answer after transmitting 4 `DHCPREQUEST` messages, the client **MAY** choose to use the previously allocated network address and

configuration parameters for the remainder of the unexpired lease. This corresponds to moving to BOUND state in the client state transition diagram shown in figure 5.

4. The client may choose to relinquish its lease on a network address by sending a DHCPRELEASE message to the server. The client identifies the lease to be released with the lease identification cookie.

Note that in this case, where the client retains its network address locally, the client will not normally relinquish its lease during a graceful shutdown. Only in the case where the client explicitly needs to relinquish its lease, e.g., the client is about to be moved to a different subnet, will the client send a DHCPRELEASE message.

### 3.3 Interpretation and representation of time values

A client acquires a lease for a network address for a fixed period of time (which may be infinite). Throughout the protocol, times are to be represented in units of seconds. The time value of 0xffffffff is reserved to represent "infinity". The minimum lease duration is one hour.

As clients and servers may not have synchronized clocks, times are represented in DHCP messages as relative times, to be interpreted with respect to the client's local clock. Representing relative times in units of seconds in an unsigned 32 bit word gives a range of relative times from 0 to approximately 100 years, which is sufficient for the relative times to be measured using DHCP.

The algorithm for lease duration interpretation given in the previous paragraph assumes that client and server clocks are stable relative to each other. If there is drift between the two clocks, the server may consider the lease expired before the client does. To compensate, the server may return a shorter lease duration to the client than the server commits to its local database of client information.

### 3.4 Host parameters in DHCP

Not all clients require initialization of all parameters listed in Appendix A. Two techniques are used to reduce the number of parameters transmitted from the server to the client. First, most of the parameters have defaults defined in the Host Requirements RFCs; if the client receives no parameters from the server that override the defaults, a client uses those default values. Second, in its initial DHCPDISCOVER or DHCPREQUEST message, a client may provide the

server with a list of specific parameters the client is interested in.

The client SHOULD include the 'maximum DHCP message size' option to let the server know how large the server may make its DHCP messages. The parameters returned to a client may still exceed the space allocated to options in a DHCP message. In this case, two additional options flags (which must appear in the 'options' field of the message) indicate that the 'file' and 'sname' fields are to be used for options.

The client can inform the server which configuration parameters the client is interested in by including the 'parameter request list' option. The data portion of this option explicitly lists the options requested by tag number.

In addition, the client may suggest values for the network address and lease time in the DHCPDISCOVER message. The client may include the be assigned, and may include the 'IP address lease time' option to suggest the lease time it would like. No other options representing "hints" at configuration parameters are allowed in a DHCPDISCOVER or DHCPREQUEST message. The 'ciaddr' field is to be filled in only in a DHCPREQUEST message when the client is requesting use of a previously allocated IP address.

If a server receives a DHCPREQUEST message with an invalid 'ciaddr', the server SHOULD respond to the client with a DHCPNAK message and may choose to report the problem to the system administrator. The server may include an error message in the 'message' option.

### 3.5 Use of DHCP in clients with multiple interfaces

A host with multiple network interfaces must use DHCP through each interface independently to obtain configuration information parameters for those separate interfaces.

### 3.6 When clients should use DHCP

A host should use DHCP to reacquire or verify its IP address and network parameters whenever the local network parameters may have changed; e.g., at system boot time or after a disconnection from the local network, as the local network configuration may change without the host's or user's knowledge.

If a host has knowledge of a previous network address and is unable to contact a local DHCP server, the host may continue to use the previous network address until the lease for that address expires. If the lease expires before the host can contact a DHCP server, the

host must immediately discontinue use of the previous network address and may inform local users of the problem.

#### 4. Specification of the DHCP client-server protocol

In this section, we assume that a DHCP server has a block of network addresses from which it can satisfy requests for new addresses. Each server also maintains a database of allocated addresses and leases in local permanent storage.

##### 4.1 Constructing and sending DHCP messages

DHCP clients and servers both construct DHCP messages by filling in fields in the fixed format section of the message and appending tagged data items in the variable length option area. The options area includes first a four-octet 'magic cookie' (which was described in section 3), followed by the options. The last option must always be the 'end' option.

DHCP uses UDP as its transport protocol. DHCP messages from a client to a server are sent to the 'DHCP server' port (67), and DHCP messages from a server to a client are sent to the 'DHCP client' port (68).

DHCP messages broadcast by a client prior to that client obtaining its IP address must have the source address field in the IP header set to 0.

If the 'giaddr' field in a DHCP message from a client is non-zero, the server sends any return messages to the 'DHCP server' port on the DHCP relaying agent whose address appears in 'giaddr'. If the 'giaddr' field is zero, the client is on the same subnet, and the server sends any return messages to either the client's network address, if that address was supplied in the 'ciaddr' field, or to the client's hardware address or to the local subnet broadcast address.

If the options in a DHCP message extend into the 'sname' and 'file' fields, the 'option overload' option MUST appear in the 'options' field, with value 1, 2 or 3, as specified in the DHCP options document [2]. If the 'option overload' option is present in the 'options' field, the options in the 'options' field MUST be terminated by an options field. The options in the 'sname' and 'file' fields (if in use as indicated by the 'options overload' option) MUST begin with the first octet of the field, MUST be terminated by an 'end' option, and MUST be followed by 'pad' options to fill the remainder of the field. Any individual option in the 'options', 'sname' and 'file' fields MUST be entirely contained in

that field. The options in the 'options' field MUST be interpreted first, so that any 'option overload' options may be interpreted. The 'file' field MUST be interpreted next (if the options), followed by the 'sname' field.

DHCP clients are responsible for all message retransmission. The client MUST adopt a retransmission strategy that incorporates a randomized exponential backoff algorithm to determine the delay between retransmissions. The delay before the first retransmission MUST be 4 seconds randomized by the value of a uniform random number chosen from the range -1 to +1. Clients with clocks that provide resolution granularity of less than one second may choose a non-integer randomization value. The delay before the next retransmission MUST be 8 seconds randomized by the value of a uniform random number chosen from the range -1 to +1. The retransmission delay MUST be doubled with subsequent retransmissions up to a maximum of 64 seconds. The client MAY provide an indication of retransmission attempts to the user as an indication of the progress of the configuration process. The protocol specification in the remainder of this section will describe, for each DHCP message, when it is appropriate for the client to retransmit that message forever, and when it is appropriate for a client to abandon that message and attempt to use a different DHCP message.

Normally, DHCP servers and BOOTP relay agents attempt to deliver DHCPDISCOVER, DHCPOFFER, DHCPACK and DHCPNAK messages directly to the client using unicast delivery. The IP destination address (in the IP header) is set to the DHCP 'yiaddr' address and the link-layer destination address is set to the DHCP 'chaddr' address. Unfortunately, some client implementations are unable to receive such unicast IP datagrams until the implementation has been configured with a valid IP address (leading to a deadlock in which the client's IP address cannot be delivered until the client has been configured with an IP address).

A client that cannot receive unicast IP datagrams until its protocol software has been configured with an IP address SHOULD set the BROADCAST bit in the 'flags' field to 1 in any DHCPDISCOVER or DHCPREQUEST messages that client sends. The BROADCAST bit will provide a hint to the DHCP server and BOOTP relay agent to broadcast any messages to the client on the client's subnet. A client that can receive unicast IP datagrams before its protocol software has been configured SHOULD clear the BROADCAST bit to 0. The BOOTP clarifications document discusses the ramifications of the use of the BROADCAST bit [21].

A server or relay agent sending or relaying a DHCP message directly to a DHCP client (i.e., not to a relay agent specified in the

'giaddr' field) SHOULD examine the BROADCAST bit in the 'flags' field. If this bit is set to 1, the DHCP message SHOULD be sent as an IP broadcast using an IP broadcast address (preferably 255.255.255.255) as the IP destination address and the link-layer broadcast address as the link-layer destination address. If the BROADCAST bit is cleared to 0, the message SHOULD be sent as an IP unicast to the IP address specified in the 'yiaddr' field and the link-layer address specified in the 'chaddr' field. If unicasting is not possible, the message MAY be sent as an IP broadcast using an IP broadcast address (preferably 255.255.255.255) as the IP destination address and the link-layer broadcast address as the link-layer destination address.

#### 4.2 DHCP server administrative controls

DHCP servers are not required to respond to every DHCPDISCOVER and DHCPREQUEST message they receive. For example, a network administrator, to retain stringent control over the hosts attached to the network, may choose to configure DHCP servers to respond only to hosts that have been previously registered through some external mechanism. The DHCP specification describes only the interactions between clients and servers when the clients and servers choose to interact; it is beyond the scope of the DHCP specification to describe all of the administrative controls that system administrators might want to use. Specific DHCP server implementations may incorporate any controls or policies desired by a network administrator.

In some environments, a DHCP server will have to consider the values of the 'chaddr' field and/or the 'class-identifier' option included in the DHCPDISCOVER or DHCPREQUEST messages when determining the correct parameters for a particular client. For example, an organization might have a separate bootstrap server for each type of client it uses, requiring the DHCP server to examine the 'class-identifier' to determine which bootstrap server address to return in the 'siaddr' field of a DHCPOFFER or DHCPACK message.

A DHCP server must use some unique identifier to associate a client with its lease. The client may choose to explicitly provide the identifier through the 'client identifier' option. If the client does not provide a 'client identifier' option, the server MUST use the contents of the 'chaddr' field to identify the client.

DHCP clients are free to use any strategy in selecting a DHCP server among those from which the client receives a DHCPOFFER message. The client implementation of DHCP should provide a mechanism for the user to select directly the 'class-identifier' value.

### 4.3 DHCP server behavior

A DHCP server processes incoming DHCP messages from a client based on the current state of the binding for that client. A DHCP server can receive the following messages from a client:

- o DHCPDISCOVER
- o DHCPREQUEST
- o DHCPDECLINE
- o DHCPRELEASE

Table 3 gives the use of the fields and options in a DHCP message by a server. The remainder of this section describes the action of the DHCP server for each possible incoming message.

#### 4.3.1 DHCPDISCOVER message

When a server receives a DHCPDISCOVER message from a client, the server chooses a network address for the requesting client. If no address is available, the server may choose to report the problem to the system administrator and may choose to reply to the client with a DHCPNAK message. If the server chooses to respond to the client, it may include an error message in the 'message' option. If an address is available, the new address should be chosen as follows:

- o The client's previous address as recorded in the client's binding, if that address is in the server's pool of available addresses and not already allocated, else
- o The address requested in the 'Requested IP Address' option, if that address is valid and not already allocated, else
- o A new address allocated from the server's pool of available addresses.



| Field<br>-----               | DHCPOFFER<br>-----                                 | DHCPACK<br>-----                               | DHCPNAK<br>-----                               |
|------------------------------|--|--|--|
| 'op'                         | BOOTREPLY  | BOOTREPLY                                      | BOOTREPLY                                      |
| 'htype'                      | (From "Assigned Numbers" RFC)                      |  |  |
| 'hlen'                       | (Hardware address length in octets)                |  |  |
| 'hops'                       | 0  | 0  | 0  |
| 'xid'                        | 'xid' from client                                  | 'xid' from client                              | 'xid' from client                              |
|                              | DHCPDISCOVER<br>message                            | DHCPREQUEST<br>message                         | DHCPREQUEST<br>message                         |
| 'secs'                       | 0  | 0  | 0  |
| 'ciaddr'                     | 0  | 'ciaddr' from<br>DHCPREQUEST or 0              | 'ciaddr' from<br>DHCPREQUEST or 0              |
| 'yiaddr'                     | IP address offered<br>to client                    | IP address<br>assigned to client               | 0  |
| 'siaddr'                     | IP address of next<br>bootstrap server             | IP address of next<br>bootstrap server         | 0  |
| 'flags'                      | if 'giaddr' is not 0                               | then 'flags' from client message               | else 0   |
| 'giaddr'                     | 0  | 0  | 0  |
| 'chaddr'                     | 'chaddr' from<br>client<br>DHCPDISCOVER<br>message | 'chaddr' from<br>client DHCPREQUEST<br>message | 'chaddr' from<br>client DHCPREQUEST<br>message |
| 'sname'                      | Server host name<br>or options                     | Server host name<br>or options                 | (unused)                                       |
| 'file'                       | Client boot file<br>name or options                | Client boot file<br>name or options            | (unused)                                       |
| 'options'                    | options  | options  |  |
| Option<br>-----              | DHCPOFFER<br>-----                                 | DHCPACK<br>-----                               | DHCPNAK<br>-----                               |
| Requested IP address         | MUST NOT   | MUST NOT                                       | MUST NOT                                       |
| IP address lease time        | MUST   | MUST   | MUST NOT                                       |
| Use 'file'/'sname'<br>fields | MAY  | MAY  | MUST NOT                                       |
| DHCP message type            | DHCPOFFER  | DHCPACK  | DHCPNAK  |
| Parameter request list       | MUST NOT   | MUST NOT                                       | MUST NOT                                       |
| Message                      | SHOULD   | SHOULD   | SHOULD   |
| Client identifier            | MUST NOT   | MUST NOT                                       | MUST NOT                                       |
| Class identifier             | MUST NOT   | MUST NOT                                       | MUST NOT                                       |
| Server identifier            | MUST   | MAY  | MAY  |
| Maximum message size         | MUST NOT   | MUST NOT                                       | MUST NOT                                       |
| All others                   | MAY  | MAY  | MUST NOT                                       |

Table 3: Fields and options used by DHCP servers

As described in section 4.2, a server MAY, for administrative reasons, assign an address other than the one requested, or may refuse to allocate an address to a particular client even though free addresses are available.

While not required for correct operation of DHCP, the server should not reuse the selected network address before the client responds to the server's DHCPOFFER message. The server may choose to record the address as offered to the client.

The server must also choose an expiration time for the lease, as follows:

- o IF the client has not requested a specific lease in the DHCPDISCOVER message and the client already has an assigned network address, the server returns the lease expiration time previously assigned to that address (note that the client must explicitly request a specific lease to extend the expiration time on a previously assigned address), ELSE
- o IF the client has not requested a specific lease in the DHCPDISCOVER message and the client does not have an assigned network address, the server assigns a locally configured default lease time, ELSE
- o IF the client has requested a specific lease in the DHCPDISCOVER message (regardless of whether the client has an assigned network address), the server may choose either to return the requested lease (if the lease is acceptable to local policy) or select another lease.

Once the network address and lease have been determined, the server constructs a DHCPOFFER message with the offered configuration parameters. It is important for all DHCP servers to return the same parameters (with the possible exception of a newly allocated network address) to ensure predictable host behavior regardless of the which server the client selects. The configuration parameters MUST be selected by applying the following rules in the order given below. The network administrator is responsible for configuring multiple DHCP servers to ensure uniform responses from those servers. The server MUST return to the client:

- o The client's network address, as determined by the rules given earlier in this section, and the subnet mask for the network to which the client is connected,
- o The expiration time for the client's lease, as determined by the rules given earlier in this section,
- o Parameters requested by the client, according to the following rules:
  - IF the server has been explicitly configured with a default value for the parameter, the server MUST include that value in an appropriate option in the 'option' field, ELSE
  - IF the server recognizes the parameter as a parameter defined in the Host Requirements Document, the server MUST include the default value for that parameter as given in the Host Requirements Document in an appropriate option in the 'option' field, ELSE
  - The server MUST NOT return a value for that parameter,
- o Any parameters from the existing binding that differ from the Host Requirements documents defaults,
- o Any parameters specific to this client (as identified by the contents of 'chaddr' in the DHCPDISCOVER or DHCPREQUEST message), e.g., as configured by the network administrator,
- o Any parameters specific to this client's class (as identified by the contents of the 'class identifier' option in the DHCPDISCOVER or DHCPREQUEST message), e.g., as configured by the network administrator; the parameters MUST be identified by an exact match between the client's 'client class' and the client class identified in the server,
- o Parameters with non-default values on the client's subnet.

The server inserts the 'xid' field from the DHCPDISCOVER message into the 'xid' field of the DHCPOFFER message and sends the DHCPOFFER message to the requesting client.

#### 4.3.2 DHCPREQUEST message

A DHCPREQUEST message may come from a client responding to a DHCPOFFER message from a server, or from a client verifying a previously allocated IP address. If the DHCPREQUEST message contains a 'server identifier' option, the message is in response to a

DHCPOFFER message. Otherwise, the message is a request to renew or extend an existing lease.

Consider first the case of a DHCPREQUEST message in response to a DHCPOFFER message. If the server is identified in the 'server identifier' option in the DHCPREQUEST message, the server checks to confirm that the requested parameters are acceptable. Usually, the requested parameters will match those returned to the client in the DHCPOFFER message; however, the client may choose to request a different lease duration. Also, there is no requirement that the server cache the parameters from the DHCPOFFER message. The server must simply check that the parameters requested in the DHCPREQUEST are acceptable. If the parameters are acceptable, the server records the new client binding and returns a DHCPACK message to the client.

If the requested parameters are unacceptable, e.g., the requested lease time is unacceptable to local policy, the server sends a DHCPNAK message to the client. The server may choose to return an error message in the 'message' option.

If a different server is identified in the 'server identifier' field, the client has selected a different server from which to obtain configuration parameters. The server may discard any information it may have cached about the client's request, and may free the network address that it had offered to the client.

Note that the client may choose to collect several DHCPOFFER messages and select the "best" offer. The client indicates its selection by identifying the offering server in the DHCPREQUEST message. If the client receives no acceptable offers, the client may choose to try another DHCPDISCOVER message. Therefore, the servers may not receive a specific DHCPREQUEST from which they can decide whether or not the client has accepted the offer. Because the servers have not committed any network address assignments on the basis of a DHCPOFFER, servers are free to reuse offered network addresses in response to subsequent requests. As an implementation detail, servers should not reuse offered addresses and may use an implementation-specific timeout mechanism to decide when to reuse an offered address.

In the second case, when there is no 'server identifier' option, the client is renewing or extending a previously allocated IP address. The server checks to confirm that the requested parameters are acceptable. If the parameters specified in the DHCPREQUEST message match the previous parameters, or if the request for an extension of the lease (indicated by an extended 'IP address lease time' option) is acceptable, the server returns a DHCPACK message to the requesting client. Otherwise, the server returns a DHCPNAK message to the

client. In particular, if the previously allocated network address in the 'ciaddr' field from the client does not match the network address recorded by the server for that client, the server sends a DHCPNAK to the client.

A DHCP server chooses the parameters to return in a DHCPACK message according to the same rules as used in constructing a DHCPOFFER message, as given in section 4.3.1.

#### 4.3.3 DHCPDECLINE message

If the server receives a DHCPDECLINE message, the client has discovered through some other means that the suggested network address is already in use. The server **MUST** mark the network address as not allocated and **SHOULD** notify the local system administrator of a possible configuration problem.

#### 4.3.4 DHCPRELEASE message

Upon receipt of a DHCPRELEASE message, the server marks the network address as not allocated. The server should retain a record of the client's initialization parameters for possible reuse in response to subsequent requests from the client.

### 4.4 DHCP client behavior

Figure 5 gives a state-transition diagram for a DHCP client. A client can receive the following messages from a server:

- o DHCPOFFER
- o DHCPACK
- o DHCPNAK

Table 4 gives the use of the fields and options in a DHCP message by a client. The remainder of this section describes the action of the DHCP client for each possible incoming message. The description in the following section corresponds to the full configuration procedure previously described in section 3.1, and the text in the subsequent section corresponds to the abbreviated configuration procedure described in section 3.2.

#### 4.4.1 Initialization and allocation of network address

The client begins in INIT state and forms a DHCPDISCOVER message. The client should wait a random time between one and ten seconds to desynchronize the use of DHCP at startup. The client sets 'ciaddr'

to all 0x00000000. The client MAY request specific parameters by including the 'parameter request list' option. The client MAY suggest a network address and/or lease time by including the 'requested IP address' and 'IP address lease time' options. The client MUST include its hardware address in the 'chaddr' field for use in delivery of DHCP reply messages. The client MAY include a different unique identifier in the 'client identifier' option. If the client does not include the

The client generates and records a random transaction identifier and inserts that identifier into the 'xid' field. The client records its own local time for later use in computing the lease expiration. The client then broadcasts the DHCPDISCOVER on the local hardware broadcast address to the all-ones IP broadcast address and 'DHCP server' UDP port.

If the 'xid' of an arriving DHCPOFFER message does not match the 'xid' of the most recent DHCPDISCOVER message, the DHCPOFFER message must be silently discarded. Any arriving DHCPACK messages must be silently discarded.

The client collects DHCPOFFER messages over a period of time, selects one DHCPOFFER message from the (possibly many) incoming DHCPOFFER messages (e.g., the first DHCPOFFER message or the DHCPOFFER message from the previously used server) and extracts the server address from the 'server identifier' option in the DHCPOFFER message. The time over which the client collects messages and the mechanism used to select one DHCPOFFER are implementation dependent. The client may perform a check on the suggested address to ensure that the address is not already in use. For example, if the client is on a network that supports ARP, the client may issue an ARP request for the suggested request. When broadcasting an ARP request for the suggested address, the client must fill in its own hardware address as the sender's hardware address, and 0 as the sender's IP address, to avoid confusing ARP caches in other hosts on the same subnet. If the network address appears to be in use, the client sends a DHCPDECLINE message to the server and waits for another DHCPOFFER. As the client does not have a valid network address, the client must broadcast the DHCPDECLINE message.

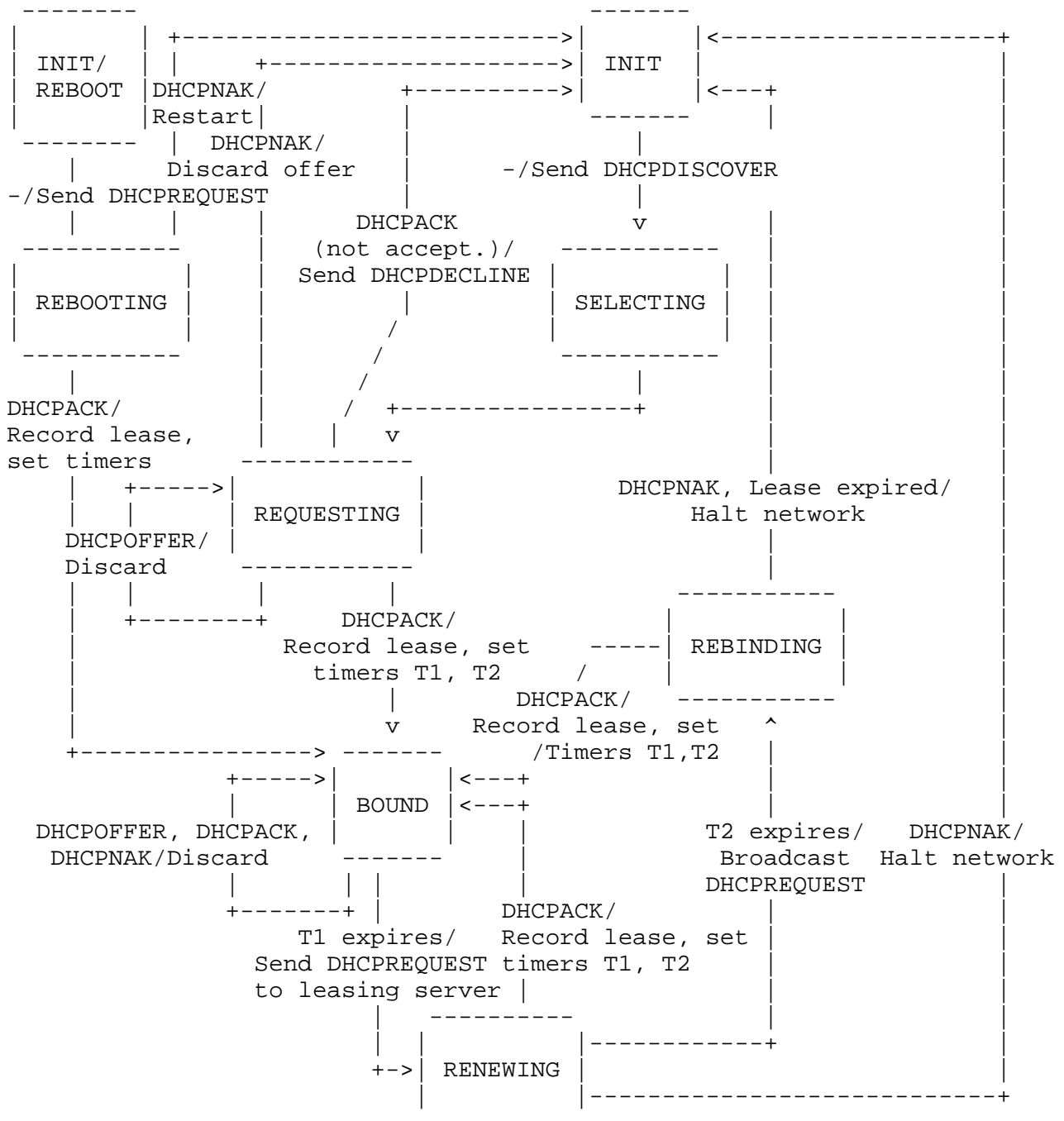


Figure 5: State-transition diagram for DHCP clients

| Field     | DHCPDISCOVER  | DHCPREQUEST   | DHCPDECLINE,<br>DHCPRELEASE     |
|-----------|---|---|---------------------------------|
| -----     | -----   | -----   | -----                           |
| 'op'      | BOOTREQUEST   | BOOTREQUEST   | BOOTREQUEST                     |
| 'htype'   | (From "Assigned Numbers" RFC)   |   |                                 |
| 'hlen'    | (Hardware address length in octets)   |   |                                 |
| 'hops'    | 0   | 0   | 0                               |
| 'xid'     | selected by client  | selected by client  | selected by client              |
| 'secs'    | (opt.)  | (opt.)  | 0                               |
| 'flags'   | Set 'BROADCAST'<br>flag if client<br>requires broadcast<br>reply<br>0                         | Set 'BROADCAST'<br>flag if client<br>requires broadcast<br>reply<br>0                         |                                 |
| 'ciaddr'  | 0   | previously<br>allocated newtork<br>address  | ciaddr                          |
| 'yiaddr'  | 0   | 0   | 0                               |
| 'siaddr'  | 0   | 0   | 0                               |
| 'giaddr'  | 0   | 0   | 0                               |
| 'chaddr'  | client's hardware<br>address  | client's hardware<br>address  | client's<br>hardware<br>address |
| 'sname'   | options, if<br>indicated in<br>'sname/file'<br>option; otherwise<br>unused                    | options, if<br>indicated in<br>'sname/file'<br>option; otherwise<br>unused                    | (unused)                        |
| 'file'    | options, if<br>indicated in<br>'sname/file'<br>option; otherwise<br>'generic' name or<br>null | options, if<br>indicated in<br>'sname/file'<br>option; otherwise<br>'generic' name or<br>null | (unused)                        |
| 'options' | options   | options   | (unused)                        |



| Option                    | DHCPDISCOVER | DHCPREQUEST  | DHCPDECLINE,<br>DHCPRELEASE |
|---------------------------|--------------|--|-----------------------------|
| -----                     | -----        | -----  | -----                       |
| Requested IP address      | MAY          | MUST NOT   | MUST NOT                    |
| IP address lease time     | MAY          | MAY  | MUST NOT                    |
| Use 'file'/'sname' fields | MAY          | MAY  | MAY                         |
| DHCP message type         | DHCPDISCOVER | DHCPREQUEST  | DHCPDECLINE/<br>DHCPRELEASE |
| Client identifier         | MAY          | MAY  | MAY                         |
| Class identifier          | SHOULD       | SHOULD   | MUST NOT                    |
| Server identifier         | MUST NOT     | MUST (after<br>DHCPDISCOVER),<br>MUST NOT (when<br>renewing) | MUST                        |
| Parameter request list    | MAY          | MAY  | MUST NOT                    |
| Maximum message size      | MAY          | MAY  | MUST NOT                    |
| Message                   | SHOULD NOT   | SHOULD NOT   | SHOULD                      |
| Site-specific             | MAY          | MAY  | MUST NOT                    |
| All others                | MUST NOT     | MUST NOT   | MUST NOT                    |

Table 4: Fields and options used by DHCP clients

If the parameters are acceptable, the client records the address of the server that supplied the parameters from the 'server identifier' field and sends that address in the 'server identifier' field of a DHCPREQUEST broadcast message. Once the DHCPACK message from the server arrives, the client is initialized and moves to BOUND state. The DHCPREQUEST message contains the same 'xid' as the DHCPOFFER message. The client records the lease expiration time as the sum of the time at which the original request was sent and the duration of the lease from the DHCPOFFER message. The client SHOULD broadcast an ARP reply to announce the client's new IP address and clear any outdated ARP cache entries in hosts on the client's subnet.

#### 4.4.2 Initialization with known network address

The client begins in INIT-REBOOT state and sends a DHCPREQUEST message with the 'ciaddr' field set to the client's network address. The client may request specific configuration parameters by including the random transaction identifier and inserts that identifier into the computing the lease expiration. The client MUST NOT include a 'server identifier' in the DHCPREQUEST message. The client then broadcasts the DHCPREQUEST on the local hardware broadcast address to the 'DHCP server' UDP port.

Once a DHCPACK message with an 'xid' field matching that in the client's DHCPREQUEST message arrives from any server, the client is

initialized and moves to BOUND state. The client records the lease expiration time as the sum of the time at which the DHCPREQUEST message was sent and the duration of the lease from the DHCPACK message.

#### 4.4.3 Initialization with a known DHCP server address

When the DHCP client knows the address of a DHCP server, in either INIT or REBOOTING state, the client may use that address in the DHCPDISCOVER or DHCPREQUEST rather than the IP broadcast address. If the client receives no response to DHCP messages sent to the IP address of a known DHCP server, the DHCP client reverts to using the IP broadcast address.

#### 4.4.4 Reacquisition and expiration

The client maintains two times, T1 and T2, that specify the times at which the client tries to extend its lease on its network address. T1 is the time at which the client enters the RENEWING state and attempts to contact the server that originally issued the client's network address. T2 is the time at which the client enters the REBINDING state and attempts to contact any server.

At time T1 after the client accepts the lease on its network address, the client moves to RENEWING state and sends (via unicast) a DHCPREQUEST message to the server to extend its lease. The client generates a random transaction identifier and inserts that identifier into the 'xid' field in the DHCPREQUEST. The client records the local time at which the DHCPREQUEST message is sent for computation of the lease expiration time. The client MUST NOT include a 'server identifier' in the DHCPREQUEST message.

Any DHCPACK messages that arrive with an 'xid' that does not match the When the client receives a DHCPACK from the server, the client computes the lease expiration time as the sum of the time at which the client sent the DHCPREQUEST message and the duration of the lease in the DHCPACK message. The client has successfully reacquired its network address, returns to BOUND state and may continue network processing.

If no DHCPACK arrives before time T2 ( $T2 > T1$ ) before the expiration of the client's lease on its network address, the client moves to REBINDING state and sends (via broadcast) a DHCPREQUEST message to extend its lease. The client sets the 'ciaddr' field in the DHCPREQUEST to its current network address. The client MUST NOT include a 'server identifier' in the DHCPREQUEST message.

Times T1 and T2 are configurable by the server through options. T1

defaults to  $(0.5 * \text{duration\_of\_lease})$ . T2 defaults to  $(0.875 * \text{duration\_of\_lease})$ . Times T1 and T2 should be chosen with some random "fuzz" around a fixed value, to avoid synchronization of client reacquisition.

In both RENEWING and REBINDING state, if the client receives no response to its DHCPREQUEST message, the client should wait one-half the remaining time until the expiration of T1 (in RENEWING state) and T2 (in REBINDING state) down to a minimum of 60 seconds, before retransmitting the DHCPREQUEST message.

If the lease expires before the client receives a DHCPACK, the client moves to INIT state, MUST immediately stop any other network processing and requests network initialization parameters as if the client were uninitialized. If the client then receives a DHCPACK allocating that client its previous network address, the client SHOULD continue network processing. If the client is given a new network address, it MUST NOT continue using the previous network address and SHOULD notify the local users of the problem.

#### 4.4.5 DHCPRELEASE

If the client no longer requires use of its assigned network address (e.g., the client is gracefully shut down), the client sends a DHCPRELEASE message to the server. Note that the correct operation of DHCP does not depend on the transmission of DHCPRELEASE messages.

#### 5. Acknowledgments

Greg Minshall, Leo McLaughlin and John Veizades have patiently contributed to the the design of DHCP through innumerable discussions, meetings and mail conversations. Jeff Mogul first proposed the client-server based model for DHCP. Steve Deering searched the various IP RFCs to put together the list of network parameters supplied by DHCP. Walt Wimer contributed a wealth of practical experience with BOOTP and wrote a document clarifying the behavior of BOOTP/DHCP relay agents. Jesse Walker analyzed DHCP in detail, pointing out several inconsistencies in earlier specifications of the protocol. Steve Alexander reviewed Walker's analysis and the fixes to the protocol based on Walker's work. And, of course, all the members of the Dynamic Host Configuration Working Group of the IETF have contributed to the design of the protocol through discussion and review of the protocol design.

## 6. References

- [1] Acetta, M., "Resource Location Protocol", RFC 887, CMU, December 1983.
- [2] Alexander, S., and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 1533, Lachman Technology, Inc., Bucknell University, October 1993.
- [3] Braden, R., Editor, "Requirements for Internet Hosts -- Communication Layers", STD 3, RFC 1122, USC/Information Sciences Institute, October 1989.
- [4] Braden, R., Editor, "Requirements for Internet Hosts -- Application and Support", STD 3, RFC 1123, USC/Information Sciences Institute, October 1989.
- [5] Brownell, D, "Dynamic Reverse Address Resolution Protocol (DRARP)", Work in Progress.
- [6] Comer, D., and R. Droms, "Uniform Access to Internet Directory Services", Proc. of ACM SIGCOMM '90 (Special issue of Computer Communications Review), 20(4):50--59, 1990.
- [7] Croft, B., and J. Gilmore, "Bootstrap Protocol (BOOTP)", RFC 951, Stanford and SUN Microsystems, September 1985.
- [8] Deering, S., "ICMP Router Discovery Messages", RFC 1256, Xerox PARC, September 1991.
- [9] Droms, D., "Interoperation between DHCP and BOOTP" RFC 1534, Bucknell University, October 1993.
- [10] Finlayson, R., Mann, T., Mogul, J., and M. Theimer, "A Reverse Address Resolution Protocol", RFC 903, Stanford, June 1984.
- [11] Gray C., and D. Cheriton, "Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency", In Proc. of the Twelfth ACM Symposium on Operating Systems Design, 1989.
- [12] Mockapetris, P., "Domain Names -- Concepts and Facilities", STD 13, RFC 1034, USC/Information Sciences Institute, November 1987.
- [13] Mockapetris, P., "Domain Names -- Implementation and Specification", STD 13, RFC 1035, USC/Information Sciences Institute, November 1987.

- [14] Mogul J., and S. Deering, "Path MTU Discovery", RFC 1191, November 1990.
- [15] Morgan, R., "Dynamic IP Address Assignment for Ethernet Attached Hosts", Work in Progress.
- [16] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, USC/Information Sciences Institute, September 1981.
- [17] Reynolds, J., "BOOTP Vendor Information Extensions", RFC 1497, USC/Information Sciences Institute, August 1993.
- [18] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1340, USC/Information Sciences Institute, July 1992.
- [19] Jeffrey Schiller and Mark Rosenstein. A Protocol for the Dynamic Assignment of IP Addresses for use on an Ethernet. (Available from the Athena Project, MIT), 1989.
- [20] Sollins, K., "The TFTP Protocol (Revision 2)", RFC 783, NIC, June 1981.
- [21] Wimer, W., "Clarifications and Extensions for the Bootstrap Protocol", RFC 1532, Carnegie Mellon University, October 1993.

## 7. Security Considerations

DHCP is built directly on UDP and IP which are as yet inherently insecure. Furthermore, DHCP is generally intended to make maintenance of remote and/or diskless hosts easier. While perhaps not impossible, configuring such hosts with passwords or keys may be difficult and inconvenient. Therefore, DHCP in its current form is quite insecure.

Unauthorized DHCP servers may be easily set up. Such servers can then send false and potentially disruptive information to clients such as incorrect or duplicate IP addresses, incorrect routing information (including spoof routers, etc.), incorrect domain nameserver addresses (such as spoof nameservers), and so on. Clearly, once this seed information is in place, an attacker can further compromise affected systems.

Malicious DHCP clients could masquerade as legitimate clients and retrieve information intended for those legitimate clients. Where dynamic allocation of resources is used, a malicious client could claim all resources for itself, thereby denying resources to legitimate clients.

## 8. Author's Address

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

Phone: (717) 524-1145  
EMail: [droms@bucknell.edu](mailto:droms@bucknell.edu)

## A. Host Configuration Parameters

## IP-layer\_parameters,\_per\_host:\_

|   |         |             |
|---|---------|-------------|
| Be a router                                 | on/off  | HRC 3.1     |
| Non-local source routing                    | on/off  | HRC 3.3.5   |
| Policy filters for non-local source routing | (list)  | HRC 3.3.5   |
| Maximum reassembly size                     | integer | HRC 3.3.2   |
| Default TTL                                 | integer | HRC 3.2.1.7 |
| PMTU aging timeout                          | integer | MTU 6.6     |
| MTU plateau table                           | (list)  | MTU 7       |

## IP-layer\_parameters,\_per\_interface:\_

|                             |                       |             |
|-----------------------------|-----------------------|-------------|
| IP address                  | (address)             | HRC 3.3.1.6 |
| Subnet mask                 | (address mask)        | HRC 3.3.1.6 |
| MTU                         | integer               | HRC 3.3.3   |
| All-subnets-MTU             | on/off                | HRC 3.3.3   |
| Broadcast address flavor    | 0x00000000/0xffffffff | HRC 3.3.6   |
| Perform mask discovery      | on/off                | HRC 3.2.2.9 |
| Be a mask supplier          | on/off                | HRC 3.2.2.9 |
| Perform router discovery    | on/off                | RD 5.1      |
| Router solicitation address | (address)             | RD 5.1      |
| Default routers, list of:   |                       |             |
| router address              | (address)             | HRC 3.3.1.6 |
| preference level            | integer               | HRC 3.3.1.6 |
| Static routes, list of:     |                       |             |
| destination                 | (host/subnet/net)     | HRC 3.3.1.2 |
| destination mask            | (address mask)        | HRC 3.3.1.2 |
| type-of-service             | integer               | HRC 3.3.1.2 |
| first-hop router            | (address)             | HRC 3.3.1.2 |
| ignore redirects            | on/off                | HRC 3.3.1.2 |
| PMTU                        | integer               | MTU 6.6     |
| perform PMTU discovery      | on/off                | MTU 6.6     |

## Link-layer\_parameters,\_per\_interface:\_

|                        |                    |             |
|------------------------|--------------------|-------------|
| Trailers               | on/off             | HRC 2.3.1   |
| ARP cache timeout      | integer            | HRC 2.3.2.1 |
| Ethernet encapsulation | (RFC 894/RFC 1042) | HRC 2.3.3   |

## TCP\_parameters,\_per\_host:\_

|                      |         |              |
|----------------------|---------|--------------|
| TTL                  | integer | HRC 4.2.2.19 |
| Keep-alive interval  | integer | HRC 4.2.3.6  |
| Keep-alive data size | 0/1     | HRC 4.2.3.6  |

## Key:

MTU = Path MTU Discovery (RFC 1191, Proposed Standard)  
 RD = Router Discovery (RFC 1256, Proposed Standard)