Internet Engineering Task Force
INTERNET-DRAFT

R. Droms
Bucknell University
June, 1992

Dynamic Host Configuration Protocol

1 Abstract

The Dynamic Host Configuration Protocol (DHCP) passes network configuration information to participants in a TCP/IP protocol network.

2 Status of this memo

This draft document is a product of the IETF Dynamic Host Configuration Working Group; it will be submitted to the RFC editor as a standards document. Distribution of this memo is unlimited. It is available in both ASCII and PostScript formats. The figures are described in PostScript and are not included with the ASCII version of the document. Copies of the figures are available from the author. Please respond with comments to the `host-conf@sol.cs.bucknell.edu` mailing list. This document will expire on January 1, 1993.

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress." Please check the `1id-abstracts.txt` listing contained in the internet-drafts Shadow Directories on `nic.ddn.mil`, `nnsc.nsf.net`, `nic.nordu.net`, `ftp.nisc.sri.com`, or `munnari.oz.au` to learn the current status of any Internet Draft.

3 Introduction

The Dynamic Host Configuration Protocol (DHCP) provides configuration parameters to Internet hosts. DHCP consists of three components: a protocol for delivering host–specific configuration parameters from a DHCP server to a host; a mechanism for allocation of network addresses to hosts; and a protocol through which a collection of DHCP servers can cooperatively allocate network addresses from a shared pool of network addresses.

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.

DHCP uses only designated DHCP servers, rather than allowing any Internet host to act as a DHCP server. 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 provides two forms of IP address allocation: “automatic” and “dynamic.” Automatic allocation assigns an IP address to a host newly attached to an IP subnet. The new address is permanently assigned to the host and cannot be recovered through DHCP. Dynamic allocation temporarily assigns an IP address to a host. The address can be reused when no longer needed by the host.

The format of DHCP messages is based on the format of BOOTP [6] messages, to capture the BOOTP relay agent behavior described as part of the BOOTP specification [6, 19] 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.

3.1 Related Work

There are several Internet protocols and related mechanisms that address some parts of the dynamic host configuration problem. RARP [8] (through the extensions defined in the DRARP draft RFC [4]) explicitly addresses the problem of network address discovery, and includes an automatic IP address assignment mechanism. TFTP [18] provides for transport of a boot image from a boot server. ICMP [14] 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 [7].

BOOTP is a transport mechanism for a collection of configuration information. BOOTP is also extensible, and official extensions [15, 16] have been defined for several configuration parameters. Morgan has proposed extensions to BOOTP for dynamic IP address assignment [13]. NIP, used by the Athena project at MIT, is a distributed mechanism for dynamic IP address assignment [17]. RLP [1] provides for location of higher level services. Sun Microsystems[1] 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. 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 MTU discovery algorithm can determine the MTU of an arbitrary internet path [12]. Comer and Droms have proposed the use of ARP as a transport protocol for resource location and selection [5]. Finally, the Host Requirements RFCs [2, 3] mention specific requirements for host reconfiguration and suggest a scenario for initial configuration of diskless hosts.

3.2 Problem definition and issues

DHCP is designed to supply hosts with configuration parameters as defined in the Host Requirements RFCs. After obtaining parameters from 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. A site may choose to use vendor extensions to return information about higher level protocols to the host. DHCP also does not address registration of newly configured hosts with DNS[10, 11].

DHCP is not intended for use in configuring routers.

[1] Sun Microsystems, Sun Workstation and SunOS are trademarks of Sun Microsystems, Inc.

3.3 Requirements

The following list gives general requirements for DHCP.

- DHCP should be a mechanism not 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.
- 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.
- 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.
- 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.
- 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.
- DHCP must coexist with statically configured, non–participating hosts and with existing network protocol implementations.
- DHCP must interoperate with the BOOTP relay agent behavior as described by RFC 951 and by Wimer's Internet Draft.
- DHCP must interoperate with existing BOOTP clients.

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

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

4 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 (currently an Internet Draft) details the interactions between BOOTP and DHCP clients and servers, as well as interactions between specific implementations of the DHCP specification. There are some new, optional transactions that optimize the interaction between DHCP clients and servers that are described in sections 5 and 6.

There are two primary differences between DHCP and BOOTP. First, DHCP defines many new vendor extensions to transmit all of the network configuration parameters required for client operation. Second, DHCP defines mechanisms through which DHCP servers coordinate the dynamic allocation of network addresses to requesting clients.

DHCP extends the use of the 'htype', 'hlen' and 'chaddr' fields to allow the use of client identifiers that are not hardware addresses. The hardware type values defined in the ARP section of the "Assigned Numbers" RFC are reserved for use when the client identifier is a hardware address. Other client identifier types (e.g., a machine–specific serial number permanently stored with the client) must be defined for use with DHCP.

There is also a minor difference between the format of DHCP messages and BOOTP messages. In DHCP, the vendor extension field is extended to 312 octets to bring the minimum size of a DHCP message to 576 octets, the minimum IP datagram size a host must be prepared to accept [2, Sec. 3.2.2, p. 56]. DHCP clients may negotiate the use of larger DHCP messages through the 'Maximum DHCP message size' vendor extension.

4.1 Components of the Protocol

DHCP provides three distinct services:

- A persistent, dynamic repository of configuration information for clients.
- Dynamic allocation of configuration resources such as network layer addresses.
- Distribution of configuration information among protocol servers.

This document will separately describe the mechanisms through which DHCP provides the first two of these services. A separate document will describe the operation of the DHCP distributed database.

4.2 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 (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.

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.

4.3 Dynamic allocation of network addresses

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" [9]. The client may extend its lease with subsequent requests. The client may ask for a permanent assignment by asking for an infinite lease. The client may issue a message to release the address back to the server when the client no longer needs the address.

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 will 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.

5 The Client–Server Protocol

The DHCP protocol messages given in table 1 all have the same format, given in Appendix B, based on the message format defined for BOOTP messages in RFC 951. In each of these messages, the 'op' field contains BOOTREQUEST in any message from a client to a server, and the 'op' field contains BOOTREPLY in any message from a server to a client. The 'DHCP message type'

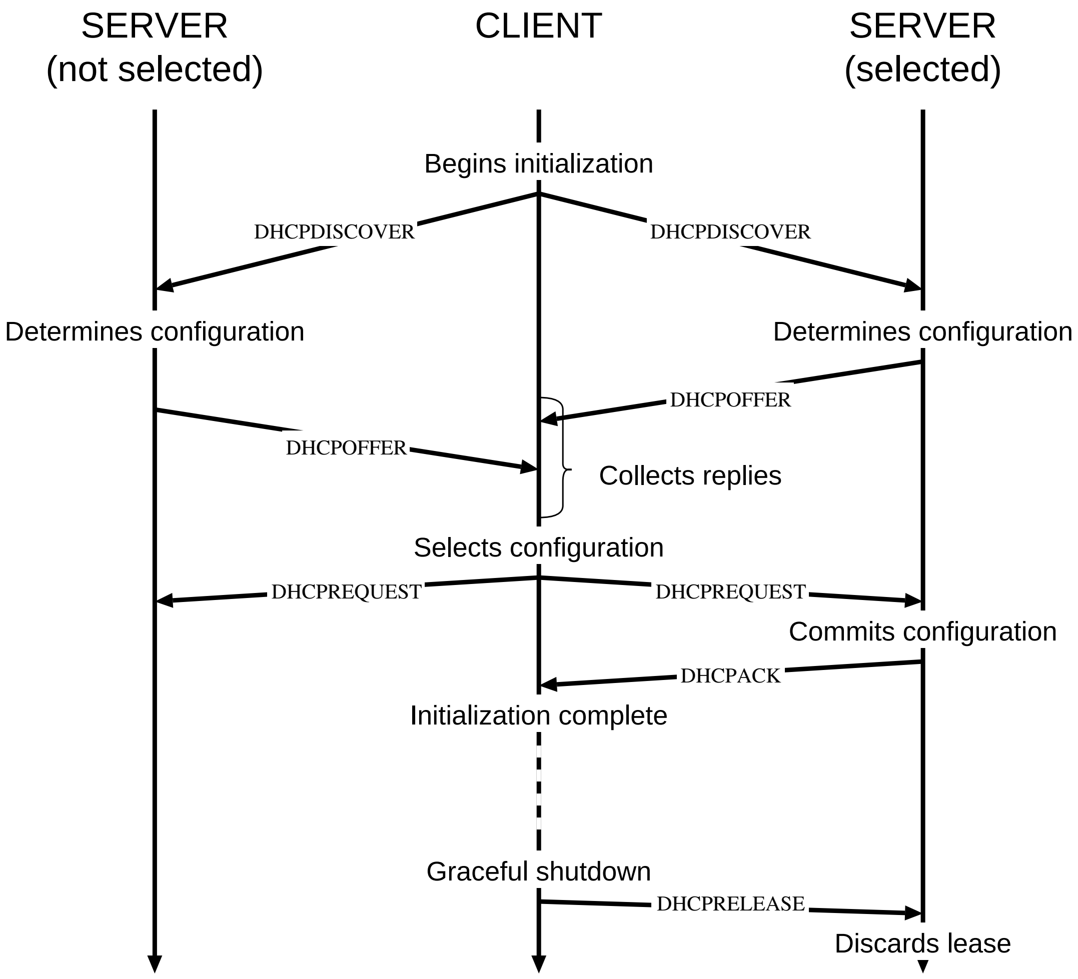

Figure 1: Timeline diagram of messages exchanged between DHCP client and servers when allocating a new network address

vendor extension indicates the type of the DHCP message.

A separate document (currently an Internet Draft) gives the vendor extensions defined by DHCP.

5.1 Client–server interaction – allocating a network address

The following summary of the protocol exchanges between clients and servers refers to the DHCP protocol messages described in table 1. The timeline diagram in figure 1 shows the timing relationships in a typical client–server interaction, for a client requesting dynamic allocation of a new network address.

1. The client broadcasts a DHCPDISCOVER message on its local physical subnet. The DHCPDISCOVER message may include suggestions for specific parameter values, e.g., network address or lease duration, from the client. DHCP/BOOTP relay agents pass the message on to DHCP servers not on the same physical subnet.

2. Each server may respond with a DHCPOFFER message with all configuration parameters including network address. 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 on the client's subnet.

3. The client receives DHCPOFFER message(s) from the server(s). The client may choose to wait for multiple responses. The client chooses one server from which to request configuration parameters, based on offered configuration parameters in the DHCPOFFER messages. The client broadcasts a DHCPREQUEST message, specifying the desired server, desired network address and any other desired configuration values in vendor extension fields. This DHCPREQUEST message is broadcast and relayed through DHCP/BOOTP relay agents. Any DHCP/BOOTP relay agents must ensure that any messages from this client are forwarded to the same set of DHCP servers to ensure that the DHCPREQUEST message reaches the selected DHCP server.

   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. The servers not selected in 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 server inserts a unique lease identification cookie as a vendor extension.

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

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[2].

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

[2]The client should wait a minimum of ten seconds before restarting the configuration process to avoid excessive network traffic in case of looping.

Message		Use
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 1: DHCP messages

The client times out and retransmits the DHCPREQUEST message if the client receives neither a DHCPACK or a DHCPNAK message.

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 with the lease identification cookie.

5.2 Client–server interaction – reusing a previously allocated network address

The timeline diagram in figure 2 shows the timing relationships in a typical client–server interaction for a client reusing a previously allocated network address.

1. Client broadcasts a DHCPREQUEST message on its local subnet. The DHCPREQUEST message includes the clients network address in the 'ciaddr' field and any other suggested configuration values in vendor extensions. DHCP/BOOTP relay agents pass the message on to DHCP servers not on the same subnet.

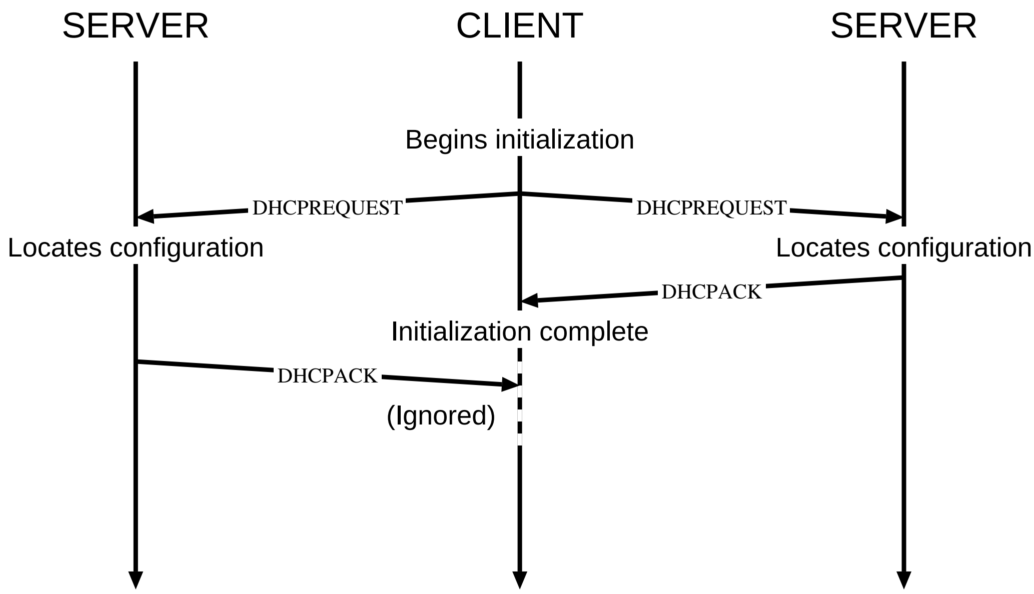

Figure 2: Timeline diagram of messages exchanged between DHCP client and servers when reusing a previously allocated network address

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. Client receives the DHCPACK message with configuration parameters. The client performs a final check on the parameters, 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 in INIT state, requesting a new network address.

   If the client receives a DHCPNAK message, the client restarts the configuration process. The client restarts the configuration process in INIT state, requesting a new network address.

   The client times out and retransmits the DHCPREQUEST message if the client receives neither a DHCPACK or a DHCPNAK message.

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.

5.3 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 all 1s 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 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.

5.4 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; in lieu of parameterization to the contrary, a client uses those default values. Second, a client may explicitly request only certain parameters in its initial DHCPREQUEST message.

The parameters returned to a client may still exceed the space allocated to vendor extensions in a DHCP message. In this case, two additional vendor extension flags (which must appear in the vendor extension area of the message) indicate that the ’file’ and ’sname’ fields are to be used for vendor extensions.

A client can fill in vendor extensions in a DHCPDISCOVER and DHCPREQUEST to supply hints or request specific values from a server. The server fills in vendor extensions in DHCPOFFER and DHCPACK messages to supply specific configuration values to the client.

A client can also request specific configuration parameters without supplying hints through the ’parameter request vector’ and ’parameter request list’ vendor extensions. In the parameter request vector, a one bit in position n in the vector represents an explicit request for the vendor extensions parameter with tag *n*. The parameter request list is a list of vendor extension tags explicitly requested by the client.

The 'ciaddr' field is to be filled in by the client only if the client has explicit knowledge of its network address. The client can supply a hint or a preferred network address through the IP address vendor extension.

If a server receives a DHCPREQUEST message with an invalid 'ciaddr', the server responds 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' vendor extension.

The client should specify the largest acceptable DHCP message with the 'DHCP message size' vendor extension to ensure that the server can transmit all the appropriate parameters in a single DHCP message.

5.5 Use of DHCP in hosts 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.

5.6 Use of DHCP by hosts

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.

6 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, and that the server will independently employ the server–server protocol to obtain more network addresses when necessary. Each server also maintains a database of allocated addresses and leases in local permanent storage.

6.1 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 vendor extension information in the variable format vendor extension area. The vendor extension area includes first a four–octet 'magic cookie', the vendor extensions and an 'end' tag, which indicates the end of the vendor extensions.

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 client' 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 known) or to the client's hardware address or to the local subnet broadcast address.

6.2 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:

- DHCPDISCOVER
- DHCPREQUEST
- DHCPDECLINE
- DHCPRELEASE

Table 2 gives the use of the fields and vendor extensions in a DHCP message by a server. The remainder of this section describes the action of the DHCP server for each possible incoming message.

6.2.1 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

Field	DHCPOFFER	DHCPACK	DHCPNAK
'op'	BOOTREPLY	BOOTREPLY	BOOTREPLY
'htype'	(From "Assigned Numbers" RFC; 0 implies "other type identifier")		
'hlen'	(Hardware adress length in octets)		
'hops'	0	0	0
'xid'	'xid' from client DHCP-DISCOVER message	'xid' from client DHCP-REQUEST message	'xid' from client DHCP-REQUEST message
secs	0	0	0
'ciaddr'	0	0	0
'yiaddr'	IP address offered to client	IP address assigned to client	0
'siaddr'	IP address of server	IP address of server	IP address of server
'giaddr'	0	0	0
'chaddr'	'chaddr' from client DHCPDISCOVER message	'chaddr' from client DHCPREQUEST message	'chaddr' from client DHCPREQUEST message
'sname'	Server host name or vendor extensions	Server host name or vendor extensions	(unused)
'file'	Client boot file name or vendor extensions	Client boot file name or vendor extensions	(unused)
'vend'	Vendor extensions	Vendor extensions	

Vendor extension	DHCPOFFER	DHCPACK	DHCPNAK
Requested IP address	MAY NOT	MAY NOT	MAY NOT
IP address lease time	MUST	MUST	MAY NOT
Use 'file'/'sname' fields	MAY	MAY	MAY NOT
DHCP message type	DHCPOFFER	DHCPACK	DHCPNAK
Lease identifier cookie	MUST	MUST	MAY NOT
Parameter request vector	MAY NOT	MAY NOT	MAY NOT
Parameter request list	MAY NOT	MAY NOT	MAY NOT
NIS (YP) domain	MAY	MAY	MAY NOT
NIS (YP) servers	MAY	MAY	MAY NOT
NTP (RFC 1129) servers	MAY	MAY	MAY NOT
Message	MAY	MAY	MAY
All others	MAY	MAY	MAY NOT

Table 2: Fields and vendor extensions used by DHCP servers

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' vendor extension. If an address is available, the new address should be chosen as follows:

- 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
- The address requested in the 'Requested IP Address' vendor extension, if that address is valid and not already allocated, else
- A new address allocated from the server's pool of available addresses.

While not required for correct operation of DHCP, the server should arrange to avoid reusing the selected network address soon after offering the address to the client. 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:

- 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 existing lease expiration time.
- 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.
- 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 initialization parameters:

- The client's network address and subnet mask.
- The expiration time for the client's lease.
- Parameters requested by the client.
- 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.

6.2.2 DHCPREQUEST message

If the server is identified in the 'server identifier' vendor extension in the DHCPREQUEST message or if there is no 'server identifier' vendor extension, the server checks to confirm that the requested parameters are acceptable and returns a DHCPACK message to the requesting client with the client's initialization parameters. The server records the allocated address and lease duration in the server database. The server should also record the client's initialization parameters for reuse in response to subsequent requests from 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' vendor extension.

If a different server is identified in the 'server identifier' field, the client has selected a different server form which to obtain configuration parameters. The server may discard any hints about 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 try to arrange to avoid reusing offered addresses and may use an implementation–specific timeout mechanism to decide when to reuse an offered address.

6.2.3 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 marks the network address as not allocated and may notify the local system administrator of a possible configuration problem.

6.2.4 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.

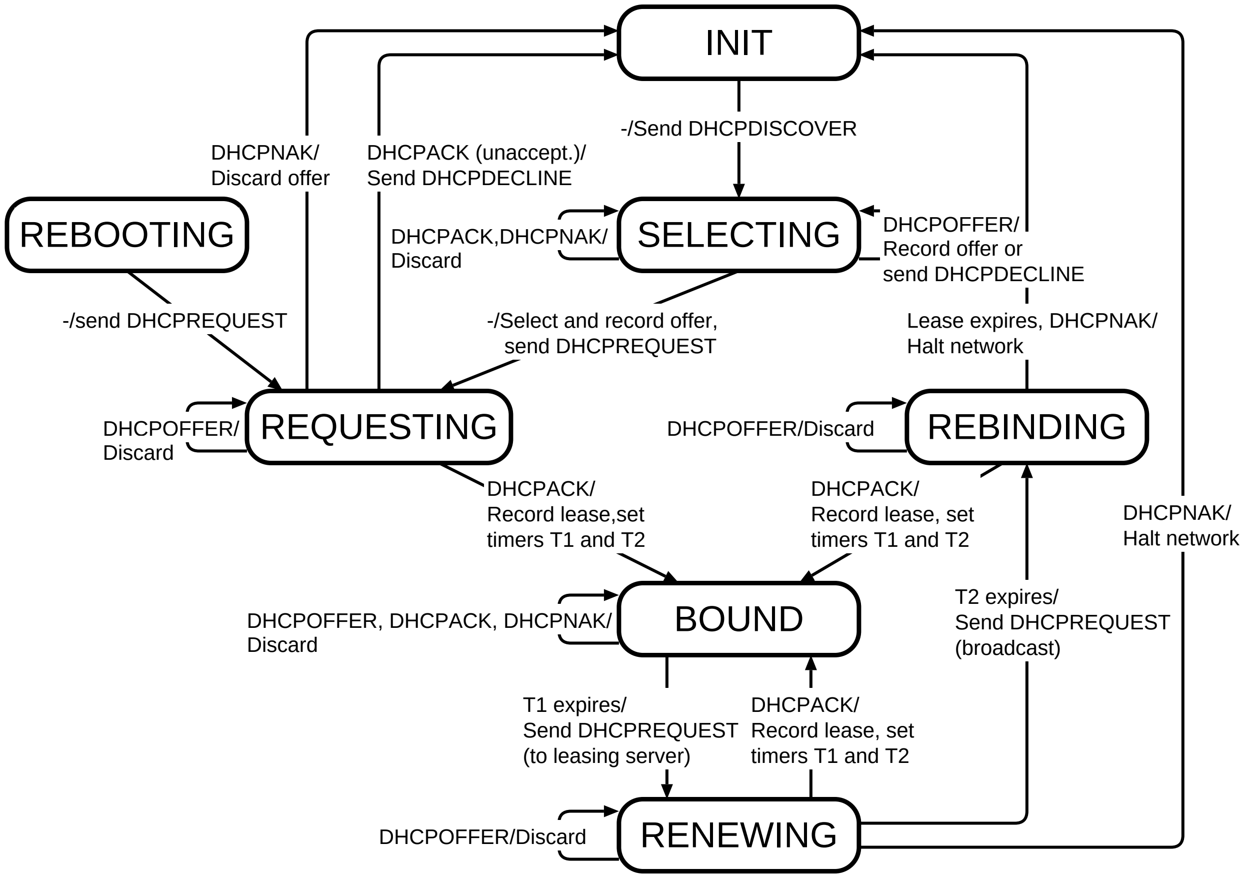

Figure 3: State–transition diagram for DHCP clients

6.3 Client behavior

Figure 3 gives a state–transition diagram for a DHCP client. A client can receive the following messages from a server:

- DHCPOFFER
- DHCPACK
- DHCPNAK

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

Field	DHCPDISCOVER	DHCPREQUEST	DHCPDECLINE, DHCPRELEASE
'op'	BOOTREQUEST	BOOTREQUEST	BOOTREQUEST
'htype'	(From “Assigned Numbers” RFC; 0 implies “other type identifier”)		
'hlen'		(Hardware adress length in octets)	
'hops'	0	0	0
'xid'	selected by client	selected by client	selected by client
'secs'	(opt.)	(opt.)	0
'ciaddr'	0	(if known)	0
'yiaddr'	0	0	0
'siaddr'	0	0	0
'giaddr'	0	0	0
'chaddr'	client's hardware address or identifier	client's hardware address or identifier	client's hardware address or identifier
'sname'	Vendor extensions (opt.)	Vendor extensions (opt.)	(unused)
'file'	Vendor extensions (opt.)	Vendor extensions (opt.)	(unused)
'vend'	Vendor extensions	Vendor extensions	(unused)

Vendor extension	DHCPDISCOVER	DHCPREQUEST	DHCPDECLINE, DHCPRELEASE
Requested IP address	MAY	MAY	MAY NOT
IP address lease time	MAY	MAY	MAY NOT
Use 'file'/'sname' fields	MAY	MAY	MAY
DHCP message type	DHCPDISCOVER	DHCPREQUEST	DHCPDECLINE/ DHCPRELEASE
Lease identifier cookie	MAY NOT	MAY NOT	MAY NOT
Parameter request vector	MAY	MAY	MAY NOT
Parameter request list	MAY	MAY	MAY NOT
NIS (YP) domain	MAY NOT	MAY NOT	MAY NOT
NIS (YP) servers	MAY NOT	MAY NOT	MAY NOT
NTP (RFC 1129) servers	MAY NOT	MAY NOT	MAY NOT
Message	MAY NOT	MAY NOT	MAY NOT
All others	MAY NOT	MAY NOT	MAY NOT

Table 3: Fields and vendor extensions used by DHCP clients

6.3.1 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 0s. The client may include a suggested network address and time for the duration of the lease and requests for any other specific parameters the client might need in the vendor extensions. Any vendor extension fields are interpreted by the server as requests for specific parameters and will be filled in and returned in the server's DHCPOFFER. 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.

> AUTHOR'S NOTE: The client must implement a timeout and retransmission with exponential backoff algorithm for the receipt of the DHCPOFFER message.

If the 'xid' of an arriving DHCPOFFER message does not match the 'xid' of the most recent DHCPDISCOVER message, the DHCPOFFER message is silently discarded. Any arriving DHCPACK messages are 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 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[3]. 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.

If the parameters are acceptable, the client records the address of the server that supplied the parameters from the 'siaddr' 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.

[3]When ARPing 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.

6.3.2 Initialization with known network address

The client begins in REBOOTING state and sends a DHCPREQUEST message with the 'ciaddr' field set to the client's network address. The client may include requests for for any other specific parameters the client might need in the vendor extensions. Any vendor extension fields are interpreted by the server as requests for specific parameters and will be filled in and returned in the server's DHCPOFFER. 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 DHCPREQUEST on the local hardware broadcast address to the 'DHCP server' UDP port.

> AUTHOR'S NOTE: The client must implement a timeout and retransmission with exponential backoff algorithm for the receipt of the DHCPOFFER message.

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.

6.3.3 Reacquisition and Expiration

At time T_1 before the expiration of the client's 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.

Any DHCPACK messages that arrive with an 'xid' that does not match the 'xid' of the client's DHCPREQUEST message are silently discarded. 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 T_2 ($T_2 < T_1$) 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.

> AUTHOR'S NOTE: The client must implement a timeout and retransmission with

exponential backoff algorithm to retry the unicast and broadcast DHCPDISCOVER messages.

Times T_1 and T_2 are configurable by the server through vendor extensions. T_1 defaults to (0.5 * duration_of_lease). T_2 defaults to (0.875 * duration_of_lease). Times T_1 and T_2 should be chosen with some random "fuzz" around a fixed value, to avoid synchronization of client reacquisition.

If the lease expires before the client receives a DHCPACK, the client moves to INIT state, must immediately stop any other network processing and request 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 may continue network processing. If the client is given a new network address, it may not continue using the previous network address and must notify the local users of the problem.

6.3.4 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.

7 Security Considerations

This document does not address security issues.

8 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.

9 Author's Address

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

(717) 524–1145

droms@bucknell.edu

References

[1] M. Acetta. Resource Location Protocol. RFC 887, NIC, December 1983.

[2] R. Braden (Ed.). Requirements for Internet Hosts – Communication Layers. RFC 1122, NIC, October 1989.

[3] R. Braden (Ed.). Requirements for Internet Hosts – Application and Support. RFC 1123, NIC, October 1989.

[4] David Brownell. Dynamic Reverse Address Resolution Protocol (DRARP). RFC DRAFT, NIC, 1989.

[5] D. Comer 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.

[6] B Croft and J. Gilmore. Bootstrap Protocol (BOOTP). RFC 951, NIC, September 1985.

[7] S. Deering. ICMP Router Discovery Messages. RFC 1256, NIC, September 1991.

[8] R. Finlayson, T. Mann, J. Mogul, and M. Theimer. A Reverse Address Resolution Protocol. RFC 903, NIC, June 1984.

[9] C. G. Gray and D. R. Cheriton. Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency. In *Proc. of the Twelfth ACM Symposium on Operating Systems Design*, 1989.

[10] P. Mockapetris. Domain Names – Concepts and Facilities. RFC 1034, NIC, November 1987.

[11] P. Mockapetris. Domain Names – Implementation and Specification. RFC 1035, NIC, November 1987.

[12] J. Mogul and S. Deering. Path MTU Discovery. RFC 1191, NIC, November 1990.

[13] R. L. Morgan. Dynamic IP Address Assignment for Ethernet Attached Hosts. RFC DRAFT, NIC, 1989.

[14] J. Postel. Internet Control Message Protocol. RFC 792, NIC, September 1981.

[15] P. Prindeville. BOOTP Vendor Information Extensions. RFC 1048, NIC, February 1988.

[16] J. Reynolds. BOOTP Vendor Information Extensions. RFC 1084, NIC, December 1988.

[17] 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.

[18] K. Sollins. The TFTP Protocol (Revision 2). RFC 783, NIC, June 1981.

[19] W. Wimer. Clarifications and Extensions for the Bootstrap Protocol. Internet Draft, NIC, 1991.

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	all 0s/all 1s	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:

HRC = Requirements for Internet Hosts – Communication Layers (RFC 1122)

MTU = Path MTU Discovery (RFC 1191, Proposed Standard)

RD = Router Discovery (RFC 1256, Proposed Standard)

B Format of a DHCP message

FIELD	OCTETS	DESCRIPTION
op	1	Message op code / message type. 1 = BOOTREQUEST, 2 = BOOTREPLY
htype	1	Hardware address type, see ARP section in "Assigned Numbers" RFC. '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 in relay–agent booting.
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.
–	2	Unused.
ciaddr	4	Client IP address; filled in by client in DHCPDISCOVER if known.
yiaddr	4	'your' (client) IP address; filled by server if client doesn't know its own address ('ciaddr' was 0).
siaddr	4	Server IP address; returned in DHCPOFFER, DHCPACK and DHCPNAK by server.
giaddr	4	Relay agent IP address, used in optional relay–agent booting.
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.
vend	312	Optional vendor extensions field. See the vendor extension documents for a list of defined vendor extensions.