DECLARATION OF SANDY GINOZA FOR IETF
RFC 1531: (DYNAMIC HOST CONFIGURATION PROTOCOL)

I, Sandy Ginoza, hereby declare that all statements made herein are of my own knowledge and are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code:

1. I am an employee of Association Management Solutions, LLC (AMS), which acts under contract to the IETF Administration LLC (IETF) as the operator of the RFC Production Center. The RFC Production Center is part of the "RFC Editor" function, which prepares documents for publication and places files in an online repository for the authoritative Request for Comments (RFC) series of documents (RFC Series), and preserves records relating to these documents. The RFC Series includes, among other things, the series of Internet standards developed by the IETF. I hold the position of Director of the RFC Production Center. I began employment with AMS in this capacity on 6 January 2010.

2. Among my responsibilities as Director of the RFC Production Center, I act as the custodian of records relating to the RFC Series, and I am familiar with the record keeping practices relating to the RFC Series, including the creation and maintenance of such records.

3. From June 1999 to 5 January 2010, I was an employee of the Information Sciences Institute at University of Southern California (ISI). I held various position titles with the RFC Editor project at ISI, ending with Senior Editor.

4. The RFC Editor function was conducted by ISI under contract to the United States government prior to 1998. In 1998, ISOC, in furtherance of its IETF activity, entered into

the first in a series of contracts with ISI providing for ISI's performance of the RFC Editor function. Beginning in 2010, certain aspects of the RFC Editor function were assumed by the RFC Production Center operation of AMS under contract to ISOC (acting through its IETF function and, in particular, the IETF Administrative Oversight Committee (now the IETF Administration LLC (IETF)). The business records of the RFC Editor function as it was conducted by ISI are currently housed on the computer systems of AMS, as contractor to the IETF.

5. I make this declaration based on my personal knowledge and information contained in the business records of the RFC Editor as they are currently housed at AMS, or confirmation with other responsible RFC Editor personnel with such knowledge.

6. Prior to 1998, the RFC Editor's regular practice was to publish RFCs, making them available from a repository via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication.

7. Since 1998, the RFC Editor's regular practice was to publish RFCs, making them available on the RFC Editor website or via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication. The announcement would go out to all subscribers and a contemporaneous electronic record of the announcement is kept in the IETF mail archive that is available online.

8. Beginning in 1998, any RFC published on the RFC Editor website or via FTP was reasonably accessible to the public and was disseminated or otherwise available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable

diligence could have located it. In particular, the RFCs were indexed and placed in a public repository.

9. The RFCs are kept in an online repository in the course of the RFC Editor's regularly conducted activity and ordinary course of business. The records are made pursuant to established procedures and are relied upon by the RFC Editor in the performance of its functions.

10. It is the regular practice of the RFC Editor to make and keep the RFC records.

11. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 1531 was no later than October 1993, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as an exhibit.

Pursuant to Section 1746 of Title 28 of United States Code, I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct and that the foregoing is based upon personal knowledge and information and is believed to be true.

Date: 1 June 2020 By: ______________________

Sandy Ginoza

4843-1829-9069

Network Working Group
Request for Comments: 1531
Category: Standards Track
R. Droms
Bucknell University
October 1993

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

List of Figures

List of Tables

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:

- "MUST"

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

- "MUST NOT"

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

- "SHOULD"

  This word or the adjective "RECOMMENDED" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

- "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.

- "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:

- "DHCP client"

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

- "DHCP server"

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

- "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.

- "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.

- 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 hand configuration 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].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     op (1)    |   htype (1)   |   hlen (1)    |   hops (1)    |
 +---------------+---------------+---------------+---------------+
 |                            xid (4)                            |
 +-------------------------------+-------------------------------+
 |           secs (2)            |           flags (2)           |
 +-------------------------------+-------------------------------+
 |                          ciaddr  (4)                          |
 +---------------------------------------------------------------+
 |                          yiaddr  (4)                          |
 +---------------------------------------------------------------+
 |                          siaddr  (4)                          |
 +---------------------------------------------------------------+
 |                          giaddr  (4)                          |
 +---------------------------------------------------------------+
 |                                                               |
 |                          chaddr  (16)                         |
 |                                                               |
 |                                                               |
 +---------------------------------------------------------------+
 |                                                               |
 |                          sname   (64)                         |
 +---------------------------------------------------------------+
 |                                                               |
 |                          file    (128)                        |
 +---------------------------------------------------------------+
 |                                                               |
 |                          options (312)                        |
 +---------------------------------------------------------------+
```

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 diferent 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.

```
                     1 1 1 1 1 1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
 -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 B|             MBZ              |
 -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 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 an 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 messages 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	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; 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

```
                Server          Client          Server
            (not selected)                    (selected)

                  v               v               v
                  |               |               |
                  |     Begins initialization     |
                  |               |               |
                  | _____________/|\_____________ |
                  |/ DHCPDISCOVER | DHCPDISCOVER \|
                  |               |               |
              Determines          |          Determines
             configuration        |         configuration
                  |               |               |
                  |\              |  ____________/|
                  | \_________    | /DHCPOFFER    |
                  |  DHCPOFFER\   |/              |
                  |            \  |               |
                  |       Collects replies        |
                  |              \|               |
                  |     Selects configuration     |
                  |               |               |
                  | _____________/|\_____________ |
                  |/ DHCPREQUEST  |  DHCPREQUEST \|
                  |               |               |
                  |               |     Commits configuration
                  |               |               |
                  |               | _____________/|
                  |               |/ DHCPACK      |
                  |               |               |
                  |    Initialization complete    |
                  |               |               |
                  .               .               .
                  .               .               .
                  |               |               |
                  |      Graceful shutdown        |
                  |               |               |
                  |               |\_____________ |
                  |               |  DHCPRELEASE \|
                  |               |               |
                  |               |        Discards lease
                  |               |               |
                  v               v               v
```

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

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 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 prameters. 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.

```
                Server          Client          Server

                  v               v               v
                  |               |               |
                  |             Begins            |
                  |         initialization        |
                  |               |               |
                  |              /|\              |
                  |  ___________/ | \___________  |
                  | /DHCPREQUEST  |  DHCPREQUEST\ |
                  |/              |              \|
                  |               |               |
               Locates            |            Locates
            configuration         |         configuration
                  |               |               |
                  |\              |              /|
                  | \             |  ___________/ |
                  |  \            | /  DHCPACK    |
                  |   \_______    |/              |
                  |    DHCPACK\   |               |
                  |         Initialization        |
                  |            complete           |
                  |              \|               |
                  |               |               |
                  |          (Subsequent          |
                  |            DHCPACKS           |
                  |            ignored)           |
                  |               |               |
                  |               |               |
                  v               v               v
```

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 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 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 MSUT 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	DHCPOFFER	DHCPACK	DHCPNAK
'op'	BOOTREPLY	BOOTREPLY	BOOTREPLY
'htype'	(From "Assigned Numbers" RFC)		
'hlen'	(Hardware address length in octets)		
'hops'	0	0	0
'xid'	'xid' from client DHCPDISCOVER message	'xid' from client DHCPREQUEST message	'xid' from client DHCPREQUEST message
'secs'	0	0	0
'ciaddr'	0	'ciaddr' from DHCPREQUEST or 0	'ciaddr' from DHCPREQUEST or 0
'yiaddr'	IP address offered to client	IP address assigned to client	0
'siaddr'	IP address of next bootstrap server	IP address of next bootstrap server	0
'flags'	if 'giaddr' is not 0 then 'flags' from client message else 0		
'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 options	Server host name or options	(unused)
'file'	Client boot file name or options	Client boot file name or options	(unused)
'options'	options	options	

Option	DHCPOFFER	DHCPACK	DHCPNAK
Requested IP address	MUST NOT	MUST NOT	MUST NOT
IP address lease time	MUST	MUST	MUST NOT
Use 'file'/'sname' 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.

```
 --------                               -------
|        | +-------------------------->|       |<-------------------+
| INIT/  | |     +-------------------->| INIT  |                    |
| REBOOT |DHCPNAK/         +---------->|       |<---+               |
|        |Restart|         |            -------     |               |
 --------  |  DHCPNAK/     |               |                        |
    |      Discard offer   |      -/Send DHCPDISCOVER               |
-/Send DHCPREQUEST         |               |                        |
    |      |     |      DHCPACK            v        |               |
 -----------     |   (not accept.)/   -----------   |               |
|           |    |  Send DHCPDECLINE |           |  |               |
| REBOOTING |    |         |         | SELECTING |  |               |
|           |    |        /          |           |  |               |
 -----------     |       /            -----------   |               |
    |            |      /                  |        |               |
DHCPACK/         |     /  +----------------+        |               |
Record lease,    |    |   v                         |               |
set timers      ------------                        |               |
    |   +----->|            |             DHCPNAK, Lease expired/   |
    |   |      | REQUESTING |                  Halt network         |
    DHCPOFFER/ |            |                       |               |
    Discard     ------------                        |               |
    |   |        |        |                   -----------           |
    |   +--------+     DHCPACK/              |           |          |
    |              Record lease, set    -----| REBINDING |          |
    |                timers T1, T2     /     |           |          |
    |                     |        DHCPACK/   -----------           |
    |                     v     Record lease, set   ^               |
    +----------------> -------      /Timers T1,T2   |               |
               +----->|       |<---+                |               |
               |      | BOUND |<---+                |               |
  DHCPOFFER, DHCPACK, |       |    |            T2 expires/   DHCPNAK/
   DHCPNAK/Discard     -------     |             Broadcast  Halt network
               |       | |         |            DHCPREQUEST         |
               +-------+ |        DHCPACK/          |               |
                    T1 expires/   Record lease, set |               |
                 Send DHCPREQUEST timers T1, T2     |               |
                 to leasing server |                |               |
                         |   ----------             |               |
                         |  |          |------------+               |
                         +->| RENEWING |                            |
                            |          |----------------------------+
                             ----------
```

Figure 5: State-transition diagram for DHCP clients

Field	DHCPDISCOVER	DHCPREQUEST	DHCPDECLINE, 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' flag if client requires broadcast reply 0	Set 'BROADCAST' flag if client requires broadcast reply	
'ciaddr'	0	previously allocated newtork address	ciaddr
'yiaddr'	0	0	0
'siaddr'	0	0	0
'giaddr'	0	0	0
'chaddr'	client's hardware address	client's hardware address	client's hardware address
'sname'	options, if indicated in 'sname/file' option; otherwise unused	options, if indicated in 'sname/file' option; otherwise unused	(unused)
'file'	options, if indicated in 'sname/file' option; otherwise 'generic' name or null	options, if indicated in 'sname/file' option; otherwise 'generic' name or null	(unused)
'options'	options	options	(unused)

Option	DHCPDISCOVER	DHCPREQUEST	DHCPDECLINE, 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/ DHCPRELEASE
Client identifier	MAY	MAY	MAY
Class identifier	SHOULD	SHOULD	MUST NOT
Server identifier	MUST NOT	MUST (after DHCPDISCOVER), MUST NOT (when 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 incldue 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 * duration_of_lease). T2 defaults to (0.875 * 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 an 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

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)

DECLARATION OF SANDY GINOZA FOR IETF
RFC 1533: (DHCP OPTIONS AND BOOTP VENDOR EXTENSIONS)

I, Sandy Ginoza, hereby declare that all statements made herein are of my own knowledge and are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code:

1. I am an employee of Association Management Solutions, LLC (AMS), which acts under contract to the IETF Administration LLC (IETF) as the operator of the RFC Production Center. The RFC Production Center is part of the "RFC Editor" function, which prepares documents for publication and places files in an online repository for the authoritative Request for Comments (RFC) series of documents (RFC Series), and preserves records relating to these documents. The RFC Series includes, among other things, the series of Internet standards developed by the IETF. I hold the position of Director of the RFC Production Center. I began employment with AMS in this capacity on 6 January 2010.

2. Among my responsibilities as Director of the RFC Production Center, I act as the custodian of records relating to the RFC Series, and I am familiar with the record keeping practices relating to the RFC Series, including the creation and maintenance of such records.

3. From June 1999 to 5 January 2010, I was an employee of the Information Sciences Institute at University of Southern California (ISI). I held various position titles with the RFC Editor project at ISI, ending with Senior Editor.

4. The RFC Editor function was conducted by ISI under contract to the United States government prior to 1998. In 1998, ISOC, in furtherance of its IETF activity, entered into

the first in a series of contracts with ISI providing for ISI's performance of the RFC Editor function. Beginning in 2010, certain aspects of the RFC Editor function were assumed by the RFC Production Center operation of AMS under contract to ISOC (acting through its IETF function and, in particular, the IETF Administrative Oversight Committee (now the IETF Administration LLC (IETF)). The business records of the RFC Editor function as it was conducted by ISI are currently housed on the computer systems of AMS, as contractor to the IETF.

5. I make this declaration based on my personal knowledge and information contained in the business records of the RFC Editor as they are currently housed at AMS, or confirmation with other responsible RFC Editor personnel with such knowledge.

6. Prior to 1998, the RFC Editor's regular practice was to publish RFCs, making them available from a repository via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication.

7. Since 1998, the RFC Editor's regular practice was to publish RFCs, making them available on the RFC Editor website or via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication. The announcement would go out to all subscribers and a contemporaneous electronic record of the announcement is kept in the IETF mail archive that is available online.

8. Beginning in 1998, any RFC published on the RFC Editor website or via FTP was reasonably accessible to the public and was disseminated or otherwise available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable

diligence could have located it. In particular, the RFCs were indexed and placed in a public repository.

9. The RFCs are kept in an online repository in the course of the RFC Editor's regularly conducted activity and ordinary course of business. The records are made pursuant to established procedures and are relied upon by the RFC Editor in the performance of its functions.

10. It is the regular practice of the RFC Editor to make and keep the RFC records.

11. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 1533 was no later than October 1993, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as an exhibit.

Pursuant to Section 1746 of Title 28 of United States Code, I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct and that the foregoing is based upon personal knowledge and information and is believed to be true.

Date: 1 June 2020 By: ______________________
Sandy Ginoza

4829-9316-1149

Network Working Group
Request for Comments: 1533
Obsoletes: 1497, 1395, 1084, 1048
Category: Standards Track

S. Alexander
Lachman Technology, Inc.
R. Droms
Bucknell University

October 1993

DHCP Options and BOOTP Vendor Extensions

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) [1] provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the "options" field of the DHCP message. The data items themselves are also called "options."

This document specifies the current set of DHCP options. This document will be periodically updated as new options are defined. Each superseding document will include the entire current list of valid options.

All of the vendor information extensions defined in RFC 1497 [2] may be used as DHCP options. The definitions given in RFC 1497 are included in this document, which supersedes RFC 1497. All of the DHCP options defined in this document, except for those specific to DHCP as defined in section 9, may be used as BOOTP vendor information extensions.

Table of Contents

1. Introduction

This document specifies options for use with both the Dynamic Host Configuration Protocol and the Bootstrap Protocol.

The full description of DHCP packet formats may be found in the DHCP specification document [1], and the full description of BOOTP packet formats may be found in the BOOTP specification document [3]. This document defines the format of information in the last field of DHCP packets ('options') and of BOOTP packets ('vend'). The remainder of this section defines a generalized use of this area for giving information useful to a wide class of machines, operating systems and configurations. Sites with a single DHCP or BOOTP server that is shared among heterogeneous clients may choose to define other, site-specific formats for the use of the 'options' field.

Section 2 of this memo describes the formats of DHCP options and BOOTP vendor extensions. Section 3 describes options defined in previous documents for use with BOOTP (all may also be used with DHCP). Sections 4-8 define new options intended for use with both DHCP and BOOTP. Section 9 defines options used only in DHCP.

References further describing most of the options defined in sections 2-6 can be found in section 12. The use of the options defined in section 9 is described in the DHCP specification [1].

Information on registering new options is contained in section 10.

2. BOOTP Extension/DHCP Option Field Format

DHCP options have the same format as the BOOTP "vendor extensions" defined in RFC 1497 [2]. Options may be fixed length or variable length. All options begin with a tag octet, which uniquely identifies the option. Fixed-length options without data consist of only a tag octet. Only options 0 and 255 are fixed length. All other options are variable-length with a length octet following the tag octet. The value of the length octet does not include the two octets specifying the tag and length. The length octet is followed by "length" octets of data. In the case of some variable-length options the length field is a constant but must still be specified.

Any options defined subsequent to this document should contain a length octet even if the length is fixed or zero.

All multi-octet quantities are in network byte-order.

When used with BOOTP, the first four octets of the vendor information field have been assigned to the "magic cookie" (as suggested in RFC 951). This field identifies the mode in which the succeeding data is to be interpreted. The value of the magic cookie is the 4 octet dotted decimal 99.130.83.99 (or hexadecimal number 63.82.53.63) in network byte order.

All of the "vendor extensions" defined in RFC 1497 are also DHCP options.

Option codes 128 to 254 (decimal) are reserved for site-specific options.

Except for the options in section 9, all options may be used with either DHCP or BOOTP.

Many of these options have their default values specified in other documents. In particular, RFC 1122 [4] specifies default values for most IP and TCP configuration parameters.

3. RFC 1497 Vendor Extensions

This section lists the vendor extensions as defined in RFC 1497. They are defined here for completeness.

3.1. Pad Option

The pad option can be used to cause subsequent fields to align on word boundaries.

The code for the pad option is 0, and its length is 1 octet.

```
 Code
+-----+
|  0  |
+-----+
```

3.2. End Option

The end option marks the end of valid information in the vendor field. Subsequent octets should be filled with pad options.

The code for the end option is 255, and its length is 1 octet.

```
 Code
+-----+
| 255 |
+-----+
```

3.3. Subnet Mask

The subnet mask option specifies the client's subnet mask as per RFC 950 [5].

If both the subnet mask and the router option are specified in a DHCP reply, the subnet mask option MUST be first.

The code for the subnet mask option is 1, and its length is 4 octets.

```
 Code   Len        Subnet Mask
+-----+-----+-----+-----+-----+-----+
|  1  |  4  |  m1 |  m2 |  m3 |  m4 |
+-----+-----+-----+-----+-----+-----+
```

3.4. Time Offset

The time offset field specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC). The offset is expressed as a signed 32-bit integer.

The code for the time offset option is 2, and its length is 4 octets.

```
 Code   Len        Time Offset
+-----+-----+-----+-----+-----+-----+
|  2  |  4  |  n1 |  n2 |  n3 |  n4 |
+-----+-----+-----+-----+-----+-----+
```

3.5. Router Option

The router option specifies a list of IP addresses for routers on the client's subnet. Routers SHOULD be listed in order of preference.

The code for the router option is 3. The minimum length for the router option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  3  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.6. Time Server Option

The time server option specifies a list of RFC 868 [6] time servers available to the client. Servers SHOULD be listed in order of preference.

The code for the time server option is 4. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  4  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.7. Name Server Option

The name server option specifies a list of IEN 116 [7] name servers available to the client. Servers SHOULD be listed in order of preference.

The code for the name server option is 5. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  5  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.8. Domain Name Server Option

The domain name server option specifies a list of Domain Name System (STD 13, RFC 1035 [8]) name servers available to the client. Servers SHOULD be listed in order of preference.

The code for the domain name server option is 6. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  6  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.9. Log Server Option

The log server option specifies a list of MIT-LCS UDP log servers available to the client. Servers SHOULD be listed in order of preference.

The code for the log server option is 7. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  7  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.10. Cookie Server Option

The cookie server option specifies a list of RFC 865 [9] cookie servers available to the client. Servers SHOULD be listed in order of preference.

The code for the log server option is 8. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  8  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.11. LPR Server Option

The LPR server option specifies a list of RFC 1179 [10] line printer servers available to the client. Servers SHOULD be listed in order of preference.

The code for the LPR server option is 9. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  9  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.12. Impress Server Option

The Impress server option specifies a list of Imagen Impress servers available to the client. Servers SHOULD be listed in order of preference.

The code for the Impress server option is 10. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  10 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.13. Resource Location Server Option

This option specifies a list of RFC 887 [11] Resource Location servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 11. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  11 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.14. Host Name Option

This option specifies the name of the client. The name may or may not be qualified with the local domain name (see section 3.17 for the preferred way to retrieve the domain name). See RFC 1035 for character set restrictions.

The code for this option is 12, and its minimum length is 1.

```
 Code   Len                 Host Name
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  12 |  n  |  h1 |  h2 |  h3 |  h4 |  h5 |  h6 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.15. Boot File Size Option

This option specifies the length in 512-octet blocks of the default boot image for the client. The file length is specified as an unsigned 16-bit integer.

The code for this option is 13, and its length is 2.

```
 Code   Len   File Size
+-----+-----+-----+-----+
|  13 |  2  |  l1 |  l2 |
+-----+-----+-----+-----+
```

3.16. Merit Dump File

This option specifies the path-name of a file to which the client's core image should be dumped in the event the client crashes. The path is formatted as a character string consisting of characters from the NVT ASCII character set.

The code for this option is 14. Its minimum length is 1.

```
 Code   Len      Dump File Pathname
+-----+-----+-----+-----+-----+-----+---
|  14 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

3.17. Domain Name

This option specifies the domain name that client should use when resolving hostnames via the Domain Name System.

The code for this option is 15. Its minimum length is 1.

```
 Code   Len        Domain Name
+-----+-----+-----+-----+-----+-----+--
|  15 |  n  |  d1 |  d2 |  d3 |  d4 |  ...
+-----+-----+-----+-----+-----+-----+--
```

3.18. Swap Server

This specifies the IP address of the client's swap server.

The code for this option is 16 and its length is 4.

```
 Code   Len    Swap Server Address
+-----+-----+-----+-----+-----+-----+
|  16 |  n  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

3.19. Root Path

This option specifies the path-name that contains the client's root disk. The path is formatted as a character string consisting of characters from the NVT ASCII character set.

The code for this option is 17. Its minimum length is 1.

```
 Code   Len      Root Disk Pathname
+-----+-----+-----+-----+-----+-----+---
|  17 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

3.20. Extensions Path

A string to specify a file, retrievable via TFTP, which contains information which can be interpreted in the same way as the 64-octet vendor-extension field within the BOOTP response, with the following exceptions:

- the length of the file is unconstrained;
- all references to Tag 18 (i.e., instances of the BOOTP Extensions Path field) within the file are ignored.

The code for this option is 18. Its minimum length is 1.

```
 Code   Len      Extensions Pathname
+-----+-----+-----+-----+-----+-----+---
|  18 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

4. IP Layer Parameters per Host

This section details the options that affect the operation of the IP layer on a per-host basis.

4.1. IP Forwarding Enable/Disable Option

This option specifies whether the client should configure its IP layer for packet forwarding. A value of 0 means disable IP forwarding, and a value of 1 means enable IP forwarding.

The code for this option is 19, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  19 |  1  | 0/1 |
+-----+-----+-----+
```

4.2. Non-Local Source Routing Enable/Disable Option

This option specifies whether the client should configure its IP layer to allow forwarding of datagrams with non-local source routes (see Section 3.3.5 of [4] for a discussion of this topic). A value of 0 means disallow forwarding of such datagrams, and a value of 1 means allow forwarding.

The code for this option is 20, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  20 |  1  | 0/1 |
+-----+-----+-----+
```

4.3. Policy Filter Option

This option specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks which specify destination/mask pairs with which to filter incoming source routes.

Any source routed datagram whose next-hop address does not match one of the filters should be discarded by the client.

See [4] for further information.

The code for this option is 21. The minimum length of this option is 8, and the length MUST be a multiple of 8.

```
 Code   Len         Address 1                  Mask 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  21 |  n  |  a1 |  a2 |  a3 |  a4 |  m1 |  m2 |  m3 |  m4 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
        Address 2                Mask 2
+-----+-----+-----+-----+-----+-----+-----+-----+---
|  a1 |  a2 |  a3 |  a4 |  m1 |  m2 |  m3 |  m4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```

4.4. Maximum Datagram Reassembly Size

This option specifies the maximum size datagram that the client should be prepared to reassemble. The size is specified as a 16-bit unsigned integer. The minimum value legal value is 576.

The code for this option is 22, and its length is 2.

```
 Code   Len      Size
+-----+-----+-----+-----+
|  22 |  2  |  s1 |  s2 |
+-----+-----+-----+-----+
```

4.5. Default IP Time-to-live

This option specifies the default time-to-live that the client should use on outgoing datagrams. The TTL is specified as an octet with a value between 1 and 255.

The code for this option is 23, and its length is 1.

```
 Code   Len   TTL
+-----+-----+-----+
|  23 |  1  | ttl |
+-----+-----+-----+
```

4.6. Path MTU Aging Timeout Option

This option specifies the timeout (in seconds) to use when aging Path MTU values discovered by the mechanism defined in RFC 1191 [12]. The timeout is specified as a 32-bit unsigned integer.

The code for this option is 24, and its length is 4.

```
 Code   Len           Timeout
+-----+-----+-----+-----+-----+-----+
|  24 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

4.7. Path MTU Plateau Table Option

This option specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value cannot be smaller than 68.

The code for this option is 25. Its minimum length is 2, and the length MUST be a multiple of 2.

```
 Code   Len     Size 1      Size 2
+-----+-----+-----+-----+-----+-----+---
|  25 |  n  |  s1 |  s2 |  s1 |  s2 | ...
+-----+-----+-----+-----+-----+-----+---
```

5. IP Layer Parameters per Interface

This section details the options that affect the operation of the IP layer on a per-interface basis. It is expected that a client can issue multiple requests, one per interface, in order to configure interfaces with their specific parameters.

5.1. Interface MTU Option

This option specifies the MTU to use on this interface. The MTU is specified as a 16-bit unsigned integer. The minimum legal value for the MTU is 68.

The code for this option is 26, and its length is 2.

```
 Code   Len      MTU
+-----+-----+-----+-----+
|  26 |  2  |  m1 |  m2 |
+-----+-----+-----+-----+
```

5.2. All Subnets are Local Option

This option specifies whether or not the client may assume that all subnets of the IP network to which the client is connected use the same MTU as the subnet of that network to which the client is directly connected. A value of 1 indicates that all subnets share the same MTU. A value of 0 means that the client should assume that some subnets of the directly connected network may have smaller MTUs.

The code for this option is 27, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  27 |  1  | 0/1 |
+-----+-----+-----+
```

5.3. Broadcast Address Option

This option specifies the broadcast address in use on the client's subnet. Legal values for broadcast addresses are specified in section 3.2.1.3 of [4].

The code for this option is 28, and its length is 4.

```
 Code   Len     Broadcast Address
+-----+-----+-----+-----+-----+-----+
|  28 |  4  |  b1 |  b2 |  b3 |  b4 |
+-----+-----+-----+-----+-----+-----+
```

5.4. Perform Mask Discovery Option

This option specifies whether or not the client should perform subnet mask discovery using ICMP. A value of 0 indicates that the client should not perform mask discovery. A value of 1 means that the client should perform mask discovery.

The code for this option is 29, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  29 |  1  | 0/1 |
+-----+-----+-----+
```

5.5. Mask Supplier Option

This option specifies whether or not the client should respond to subnet mask requests using ICMP. A value of 0 indicates that the client should not respond. A value of 1 means that the client should respond.

The code for this option is 30, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  30 |  1  | 0/1 |
+-----+-----+-----+
```

5.6. Perform Router Discovery Option

This option specifies whether or not the client should solicit routers using the Router Discovery mechanism defined in RFC 1256 [13]. A value of 0 indicates that the client should not perform router discovery. A value of 1 means that the client should perform router discovery.

The code for this option is 31, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  31 |  1  | 0/1 |
+-----+-----+-----+
```

5.7. Router Solicitation Address Option

This option specifies the address to which the client should transmit router solicitation requests.

The code for this option is 32, and its length is 4.

```
 Code   Len            Address
+-----+-----+-----+-----+-----+-----+
|  32 |  4  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

5.8. Static Route Option

This option specifies a list of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority.

The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination.

The default route (0.0.0.0) is an illegal destination for a static route. See section 3.5 for information about the router option.

The code for this option is 33. The minimum length of this option is 8, and the length MUST be a multiple of 8.

```
 Code   Len         Destination 1           Router 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  33 |  n  |  d1 |  d2 |  d3 |  d4 |  r1 |  r2 |  r3 |  r4 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
        Destination 2           Router 2
+-----+-----+-----+-----+-----+-----+-----+-----+---
|  d1 |  d2 |  d3 |  d4 |  r1 |  r2 |  r3 |  r4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```

6. Link Layer Parameters per Interface

This section lists the options that affect the operation of the data link layer on a per-interface basis.

6.1. Trailer Encapsulation Option

This option specifies whether or not the client should negotiate the use of trailers (RFC 893 [14]) when using the ARP protocol. A value of 0 indicates that the client should not attempt to use trailers. A value of 1 means that the client should attempt to use trailers.

The code for this option is 34, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  34 |  1  | 0/1 |
+-----+-----+-----+
```

6.2. ARP Cache Timeout Option

This option specifies the timeout in seconds for ARP cache entries. The time is specified as a 32-bit unsigned integer.

The code for this option is 35, and its length is 4.

```
 Code   Len           Time
+-----+-----+-----+-----+-----+-----+
|  35 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

6.3. Ethernet Encapsulation Option

This option specifies whether or not the client should use Ethernet Version 2 (RFC 894 [15]) or IEEE 802.3 (RFC 1042 [16]) encapsulation if the interface is an Ethernet. A value of 0 indicates that the client should use RFC 894 encapsulation. A value of 1 means that the client should use RFC 1042 encapsulation.

The code for this option is 36, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  36 |  1  | 0/1 |
+-----+-----+-----+
```

7. TCP Parameters

This section lists the options that affect the operation of the TCP layer on a per-interface basis.

7.1. TCP Default TTL Option

This option specifies the default TTL that the client should use when sending TCP segments. The value is represented as an 8-bit unsigned integer. The minimum value is 1.

The code for this option is 37, and its length is 1.

```
 Code   Len   TTL
+-----+-----+-----+
|  37 |  1  |  n  |
+-----+-----+-----+
```

7.2. TCP Keepalive Interval Option

This option specifies the interval (in seconds) that the client TCP should wait before sending a keepalive message on a TCP connection. The time is specified as a 32-bit unsigned integer. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested by an application.

The code for this option is 38, and its length is 4.

```
 Code   Len           Time
+-----+-----+-----+-----+-----+-----+
|  38 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

7.3. TCP Keepalive Garbage Option

This option specifies the whether or not the client should send TCP keepalive messages with a octet of garbage for compatibility with older implementations. A value of 0 indicates that a garbage octet should not be sent. A value of 1 indicates that a garbage octet should be sent.

The code for this option is 39, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  39 |  1  | 0/1 |
+-----+-----+-----+
```

8. Application and Service Parameters

This section details some miscellaneous options used to configure miscellaneous applications and services.

8.1. Network Information Service Domain Option

This option specifies the name of the client's NIS [17] domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set.

The code for this option is 40. Its minimum length is 1.

```
 Code   Len      NIS Domain Name
+-----+-----+-----+-----+-----+-----+---
|  40 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

8.2. Network Information Servers Option

This option specifies a list of IP addresses indicating NIS servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 41. Its minimum length is 4, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  41 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.3. Network Time Protocol Servers Option

This option specifies a list of IP addresses indicating NTP [18] servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 42. Its minimum length is 4, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  42 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.4. Vendor Specific Information

This option is used by clients and servers to exchange vendor-specific information. The information is an opaque object of n octets, presumably interpreted by vendor-specific code on the clients and servers. The definition of this information is vendor specific. The vendor is indicated in the class-identifier option. Servers not equipped to interpret the vendor-specific information sent by a client MUST ignore it (although it may be reported). Clients which do not receive desired vendor-specific information SHOULD make an attempt to operate without it, although they may do so (and announce they are doing so) in a degraded mode.

If a vendor potentially encodes more than one item of information in this option, then the vendor SHOULD encode the option using "Encapsulated vendor-specific options" as described below:

The Encapsulated vendor-specific options field SHOULD be encoded as a sequence of code/length/value fields of identical syntax to the DHCP options field with the following exceptions:

1) There SHOULD NOT be a "magic cookie" field in the encapsulated vendor-specific extensions field.

2) Codes other than 0 or 255 MAY be redefined by the vendor within the encapsulated vendor-specific extensions field, but SHOULD conform to the tag-length-value syntax defined in section 2.

3) Code 255 (END), if present, signifies the end of the encapsulated vendor extensions, not the end of the vendor extensions field. If no code 255 is present, then the end of the enclosing vendor-specific information field is taken as the end of the encapsulated vendor-specific extensions field.

The code for this option is 43 and its minimum length is 1.

```
Code   Len   Vendor-specific information
+-----+-----+-----+-----+---
|  43 |  n  |  i1 |  i2 | ...
+-----+-----+-----+-----+---
```

When encapsulated vendor-specific extensions are used, the information bytes 1-n have the following format:

```
 Code   Len   Data item        Code   Len   Data item       Code
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  T1 |  n  |  d1 |  d2 | ... |  T2 |  n  |  D1 |  D2 | ... | ... |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

8.5. NetBIOS over TCP/IP Name Server Option

The NetBIOS name server (NBNS) option specifies a list of RFC 1001/1002 [19] [20] NBNS name servers listed in order of preference.

The code for this option is 44. The minimum length of the option is 4 octets, and the length must always be a multiple of 4.

```
 Code   Len           Address 1              Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
|  44 |  n  |  a1 |  a2 |  a3 |  a4 |  b1 |  b2 |  b3 |  b4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
```

8.6. NetBIOS over TCP/IP Datagram Distribution Server Option

The NetBIOS datagram distribution server (NBDD) option specifies a list of RFC 1001/1002 NBDD servers listed in order of preference. The code for this option is 45. The minimum length of the option is 4 octets, and the length must always be a multiple of 4.

```
 Code   Len           Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
|  45 |  n  |  a1 |  a2 |  a3 |  a4 |  b1 |  b2 |  b3 |  b4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
```

8.7. NetBIOS over TCP/IP Node Type Option

The NetBIOS node type option allows NetBIOS over TCP/IP clients which are configurable to be configured as described in RFC 1001/1002. The value is specified as a single octet which identifies the client type as follows:

Value	Node Type
0x1	B-node
0x2	P-node
0x4	M-node
0x8	H-node

In the above chart, the notation '0x' indicates a number in base-16 (hexadecimal).

The code for this option is 46. The length of this option is always 1.

```
 Code   Len  Node Type
+-----+-----+-----------+
|  46 |  1  | see above |
+-----+-----+-----------+
```

8.8. NetBIOS over TCP/IP Scope Option

The NetBIOS scope option specifies the NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002. See [19], [20], and [8] for character-set restrictions.

The code for this option is 47. The minimum length of this option is 1.

```
 Code   Len       NetBIOS Scope
+-----+-----+-----+-----+-----+-----+----
|  47 |  n  |  s1 |  s2 |  s3 |  s4 | ...
+-----+-----+-----+-----+-----+-----+----
```

8.9. X Window System Font Server Option

This option specifies a list of X Window System [21] Font servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 48. The minimum length of this option is 4 octets, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+---
|  48 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |   ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```

8.10. X Window System Display Manager Option

This option specifies a list of IP addresses of systems that are running the X Window System Display Manager and are available to the client.

Addresses SHOULD be listed in order of preference.

The code for the this option is 49. The minimum length of this option is 4, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2

+-----+-----+-----+-----+-----+-----+-----+-----+---
|  49 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |   ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```

9. DHCP Extensions

This section details the options that are specific to DHCP.

9.1. Requested IP Address

This option is used in a client request (DHCPDISCOVER) to allow the client to request that a particular IP address be assigned.

The code for this option is 50, and its length is 4.

```
 Code   Len          Address
+-----+-----+-----+-----+-----+-----+
|  50 |  4  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

9.2. IP Address Lease Time

This option is used in a client request (DHCPDISCOVER or DHCPREQUEST) to allow the client to request a lease time for the IP address. In a server reply (DHCPOFFER), a DHCP server uses this option to specify the lease time it is willing to offer.

The time is in units of seconds, and is specified as a 32-bit unsigned integer.

The code for this option is 51, and its length is 4.

```
 Code   Len         Lease Time
+-----+-----+-----+-----+-----+-----+
|  51 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

9.3. Option Overload

This option is used to indicate that the DHCP "sname" or "file" fields are being overloaded by using them to carry DHCP options. A DHCP server inserts this option if the returned parameters will exceed the usual space allotted for options.

If this option is present, the client interprets the specified additional fields after it concludes interpretation of the standard option fields.

The code for this option is 52, and its length is 1. Legal values for this option are:

Value	Meaning
1	the "file" field is used to hold options
2	the "sname" field is used to hold options
3	both fields are used to hold options

```
 Code   Len  Value
+-----+-----+-----+
|  52 |  1  |1/2/3|
+-----+-----+-----+
```

9.4. DHCP Message Type

This option is used to convey the type of the DHCP message. The code for this option is 53, and its length is 1. Legal values for this option are:

Value	Message Type
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNAK
7	DHCPRELEASE

```
 Code   Len  Type
+-----+-----+-----+
|  53 |  1  | 1-7 |
+-----+-----+-----+
```

9.5. Server Identifier

This option is used in DHCPOFFER and DHCPREQUEST messages, and may optionally be included in the DHCPACK and DHCPNAK messages. DHCP servers include this option in the DHCPOFFER in order to allow the client to distinguish between lease offers. DHCP clients indicate which of several lease offers is being accepted by including this option in a DHCPREQUEST message.

The identifier is the IP address of the selected server.

The code for this option is 54, and its length is 4.

```
 Code   Len          Address
+-----+-----+-----+-----+-----+-----+
|  54 |  4  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

9.6. Parameter Request List

This option is used by a DHCP client to request values for specified configuration parameters. The list of requested parameters is specified as n octets, where each octet is a valid DHCP option code as defined in this document.

The client MAY list the options in order of preference. The DHCP server is not required to return the options in the requested order, but MUST try to insert the requested options in the order requested by the client.

The code for this option is 55. Its minimum length is 1.

```
 Code   Len   Option Codes
+-----+-----+-----+-----+---
|  55 |  n  |  c1 |  c2 | ...
+-----+-----+-----+-----+---
```

9.7. Message

This option is used by a DHCP server to provide an error message to a DHCP client in a DHCPNAK message in the event of a failure. A client may use this option in a DHCPDECLINE message to indicate the why the client declined the offered parameters. The message consists of n octets of NVT ASCII text, which the client may display on an available output device.

The code for this option is 56 and its minimum length is 1.

```
 Code   Len     Text
+-----+-----+-----+-----+---
|  56 |  n  |  c1 |  c2 | ...
+-----+-----+-----+-----+---
```

9.8. Maximum DHCP Message Size

This option specifies the maximum length DHCP message that it is willing to accept. The length is specified as an unsigned 16-bit integer. A client may use the maximum DHCP message size option in DHCPDISCOVER or DHCPREQUEST messages, but should not use the option in DHCPDECLINE messages.

The code for this option is 57, and its length is 2. The minimum legal value is 576 octets.

```
 Code   Len     Length
+-----+-----+-----+-----+
|  57 |  2  |  l1 |  l2 |
+-----+-----+-----+-----+
```

9.9. Renewal (T1) Time Value

This option specifies the time interval from address assignment until the client transitions to the RENEWING state.

The value is in units of seconds, and is specified as a 32-bit unsigned integer.

The code for this option is 58, and its length is 4.

```
 Code   Len         T1 Interval
+-----+-----+-----+-----+-----+-----+
|  58 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

9.10. Rebinding (T2) Time Value

This option specifies the time interval from address assignment until the client transitions to the REBINDING state.

The value is in units of seconds, and is specified as a 32-bit unsigned integer.

The code for this option is 59, and its length is 4.

```
 Code   Len         T2 Interval
+-----+-----+-----+-----+-----+-----+
|  59 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

9.11. Class-identifier

This option is used by DHCP clients to optionally identify the type and configuration of a DHCP client. The information is a string of n octets, interpreted by servers. Vendors and sites may choose to define specific class identifiers to convey particular configuration or other identification information about a client. For example, the identifier may encode the client's hardware configuration. Servers not equipped to interpret the class-specific information sent by a client MUST ignore it (although it may be reported).

The code for this option is 60, and its minimum length is 1.

```
Code   Len   Class-Identifier
+-----+-----+-----+-----+---
|  60 |  n  |  i1 |  i2 | ...
+-----+-----+-----+-----+---
```

9.12. Client-identifier

This option is used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain.

Identifiers consist of a type-value pair, similar to the

It is expected that this field will typically contain a hardware type and hardware address, but this is not required. Current legal values for hardware types are defined in [22].

The code for this option is 61, and its minimum length is 2.

```
Code   Len   Type  Client-Identifier
+-----+-----+-----+-----+-----+---
|  61 |  n  |  t1 |  i1 |  i2 | ...
+-----+-----+-----+-----+-----+---
```

10. Extensions

Additional generic data fields may be registered by contacting:

Internet Assigned Numbers Authority (IANA)
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, California 90292-6695

or by email as: iana@isi.edu

Implementation specific use of undefined generic types (those in the range 61-127) may conflict with other implementations, and registration is required.

11. Acknowledgements

The authors would like to thank Philip Almquist for his feedback on this document. The comments of the DHCP Working Group are also gratefully acknowledged. In particular, Mike Carney and Jon Dreyer from SunSelect suggested the current format of the Vendor-specific Information option.

RFC 1497 is based on earlier work by Philip Prindeville, with help from Drew Perkins, Bill Croft, and Steve Deering.

12. References

[1] Droms, R., "Dynamic Host Configuration Protocol", RFC 1531, Bucknell University, October 1993.

[2] Reynolds, J., "BOOTP Vendor Information Extensions", RFC 1497, USC/Information Sciences Institute, August 1993.

[3] Croft, W., and J. Gilmore, "Bootstrap Protocol", RFC 951, Stanford University and Sun Microsystems, September 1985.

[4] Braden, R., Editor, "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, USC/Information Sciences Institute, October 1989.

[5] Mogul, J., and J. Postel, "Internet Standard Subnetting Procedure", STD 5, RFC 950, USC/Information Sciences Institute, August 1985.

[6] Postel, J., and K. Harrenstien, "Time Protocol", STD 26, RFC 868, USC/Information Sciences Institute, SRI, May 1983.

[7] Postel, J., "Name Server", IEN 116, USC/Information Sciences Institute, August 1979.

[8] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, USC/Information Sciences Institute, November 1987.

[9] Postel, J., "Quote of the Day Protocol", STD 23, RFC 865, USC/Information Sciences Institute, May 1983.

[10] McLaughlin, L., "Line Printer Daemon Protocol", RFC 1179, The Wollongong Group, August 1990.

[11] Accetta, M., "Resource Location Protocol", RFC 887, CMU, December 1983.

[12] Mogul, J. and S. Deering, "Path MTU Discovery", RFC 1191, DECWRL, Stanford University, November 1990.

[13] Deering, S., "ICMP Router Discovery Messages", RFC 1256, Xerox PARC, September 1991.

[14] Leffler, S. and M. Karels, "Trailer Encapsulations", RFC 893, U. C. Berkeley, April 1984.

[15] Hornig, C., "Standard for the Transmission of IP Datagrams over Ethernet Networks", RFC 894, Symbolics, April 1984.

[16] Postel, J. and J. Reynolds, "Standard for the Transmission of IP Datagrams Over IEEE 802 Networks", RFC 1042, USC/Information Sciences Institute, February 1988.

[17] Sun Microsystems, "System and Network Administration", March 1990.

[18] Mills, D., "Internet Time Synchronization: The Network Time Protocol", RFC 1305, UDEL, March 1992.

[19] NetBIOS Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP transport: Concepts and Methods", STD 19, RFC 1001, March 1987.

[20] NetBIOS Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP transport: Detailed Specifications", STD 19, RFC 1002, March 1987.

[21] Scheifler, R., "FYI On the X Window System", FYI 6, RFC 1198, MIT Laboratory for Computer Science, January 1991.

[22] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1340, USC/Information Sciences Institute, July 1992.

13. Security Considerations

Security issues are not discussed in this memo.

14. Authors' Addresses

Steve Alexander
Lachman Technology, Inc.
1901 North Naper Boulevard
Naperville, IL 60563-8895

Phone: (708) 505-9555 x256
EMail: stevea@lachman.com

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

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

DECLARATION OF SANDY GINOZA FOR IETF
RFC 1541: (DYNAMIC HOST CONFIGURATION PROTOCOL)

I, Sandy Ginoza, hereby declare that all statements made herein are of my own knowledge and are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code:

1. I am an employee of Association Management Solutions, LLC (AMS), which acts under contract to the IETF Administration LLC (IETF) as the operator of the RFC Production Center. The RFC Production Center is part of the "RFC Editor" function, which prepares documents for publication and places files in an online repository for the authoritative Request for Comments (RFC) series of documents (RFC Series), and preserves records relating to these documents. The RFC Series includes, among other things, the series of Internet standards developed by the IETF. I hold the position of Director of the RFC Production Center. I began employment with AMS in this capacity on 6 January 2010.

2. Among my responsibilities as Director of the RFC Production Center, I act as the custodian of records relating to the RFC Series, and I am familiar with the record keeping practices relating to the RFC Series, including the creation and maintenance of such records.

3. From June 1999 to 5 January 2010, I was an employee of the Information Sciences Institute at University of Southern California (ISI). I held various position titles with the RFC Editor project at ISI, ending with Senior Editor.

4. The RFC Editor function was conducted by ISI under contract to the United States government prior to 1998. In 1998, ISOC, in furtherance of its IETF activity, entered into

the first in a series of contracts with ISI providing for ISI's performance of the RFC Editor function. Beginning in 2010, certain aspects of the RFC Editor function were assumed by the RFC Production Center operation of AMS under contract to ISOC (acting through its IETF function and, in particular, the IETF Administrative Oversight Committee (now the IETF Administration LLC (IETF)). The business records of the RFC Editor function as it was conducted by ISI are currently housed on the computer systems of AMS, as contractor to the IETF.

5. I make this declaration based on my personal knowledge and information contained in the business records of the RFC Editor as they are currently housed at AMS, or confirmation with other responsible RFC Editor personnel with such knowledge.

6. Prior to 1998, the RFC Editor's regular practice was to publish RFCs, making them available from a repository via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication.

7. Since 1998, the RFC Editor's regular practice was to publish RFCs, making them available on the RFC Editor website or via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication. The announcement would go out to all subscribers and a contemporaneous electronic record of the announcement is kept in the IETF mail archive that is available online.

8. Beginning in 1998, any RFC published on the RFC Editor website or via FTP was reasonably accessible to the public and was disseminated or otherwise available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable

diligence could have located it. In particular, the RFCs were indexed and placed in a public repository.

9. The RFCs are kept in an online repository in the course of the RFC Editor's regularly conducted activity and ordinary course of business. The records are made pursuant to established procedures and are relied upon by the RFC Editor in the performance of its functions.

10. It is the regular practice of the RFC Editor to make and keep the RFC records.

11. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 1541 was no later than October 1993, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as an exhibit.

Pursuant to Section 1746 of Title 28 of United States Code, I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct and that the foregoing is based upon personal knowledge and information and is believed to be true.

Date: 1 June 2020 By: ______________________
Sandy Ginoza

4832-2836-9853

Network Working Group
Request for Comments: 1541
Obsoletes: 1531
Category: Standards Track

R. Droms
Bucknell University

October 1993

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]. Due to some errors introduced into RFC 1531 in the editorial process, this memo is reissued as RFC 1541.

Table of Contents

List of Figures

List of Tables

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.

- "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:

- "DHCP client"

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

- "DHCP server"

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

- "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.

- "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.

- 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.

- 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 [21].

- 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:

- 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.

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 'chaddr' is used both as a hardware address for transmission of BOOTP 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].

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     op (1)    |   htype (1)   |   hlen (1)    |   hops (1)    |
   +---------------+---------------+---------------+---------------+
   |                            xid (4)                            |
   +-------------------------------+-------------------------------+
   |           secs (2)            |           flags (2)           |
   +-------------------------------+-------------------------------+
   |                          ciaddr  (4)                          |
   +---------------------------------------------------------------+
   |                          yiaddr  (4)                          |
   +---------------------------------------------------------------+
   |                          siaddr  (4)                          |
   +---------------------------------------------------------------+
   |                          giaddr  (4)                          |
   +---------------------------------------------------------------+
   |                                                               |
   |                          chaddr  (16)                         |
   |                                                               |
   |                                                               |
   +---------------------------------------------------------------+
   |                                                               |
   |                          sname   (64)                         |
   +---------------------------------------------------------------+
   |                                                               |
   |                          file    (128)                        |
   +---------------------------------------------------------------+
   |                                                               |
   |                          options (312)                        |
   +---------------------------------------------------------------+
```

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 diferent 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.

```
                    1 1 1 1 1 1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
B|            MBZ              |
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 'flags' field.

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 1497). The remainder of the 'options' field consists a list of tagged parameters that are called "options". All of the "vendor extensions" listed in RFC 1497 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 an 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 messages 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	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; 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

```
                Server          Client          Server
            (not selected)                    (selected)

                  v               v               v
                  |               |               |
                  |     Begins initialization     |
                  |               |               |
                  | _____________/|\_____________ |
                  |/ DHCPDISCOVER | DHCPDISCOVER \|
                  |               |               |
              Determines          |          Determines
             configuration        |         configuration
                  |               |               |
                  |\              |  ____________/|
                  | \_________    | /DHCPOFFER    |
                  |  DHCPOFFER\   |/              |
                  |            \  |               |
                  |       Collects replies        |
                  |              \|               |
                  |     Selects configuration     |
                  |               |               |
                  | _____________/|\_____________ |
                  |/ DHCPREQUEST  |  DHCPREQUEST \|
                  |               |               |
                  |               |     Commits configuration
                  |               |               |
                  |               | _____________/|
                  |               |/ DHCPACK      |
                  |               |               |
                  |    Initialization complete    |
                  |               |               |
                  .               .               .
                  .               .               .
                  |               |               |
                  |      Graceful shutdown        |
                  |               |               |
                  |               |\_____________ |
                  |               |  DHCPRELEASE \|
                  |               |               |
                  |               |        Discards lease
                  |               |               |
                  v               v               v
```

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

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 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 prameters. 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.

```
                Server          Client          Server

                  v               v               v
                  |               |               |
                  |             Begins            |
                  |         initialization        |
                  |               |               |
                  |              /|\              |
                  |  ___________/ | \___________  |
                  | /DHCPREQUEST  |  DHCPREQUEST\ |
                  |/              |              \|
                  |               |               |
               Locates            |            Locates
            configuration         |         configuration
                  |               |               |
                  |\              |              /|
                  | \             |  ___________/ |
                  |  \            | /  DHCPACK    |
                  |   \_______    |/              |
                  |    DHCPACK\   |               |
                  |         Initialization        |
                  |            complete           |
                  |              \|               |
                  |               |               |
                  |          (Subsequent          |
                  |            DHCPACKS           |
                  |            ignored)           |
                  |               |               |
                  |               |               |
                  v               v               v
```

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 'requested IP address' option to suggest that a particular IP address 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 'end' option, and MAY contain one or more 'pad' options to fill the 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 'option overload' option indicates that the 'file' field contains DHCP 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 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 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	DHCPOFFER	DHCPACK	DHCPNAK
'op'	BOOTREPLY	BOOTREPLY	BOOTREPLY
'htype'	(From "Assigned Numbers" RFC)		
'hlen'	(Hardware address length in octets)		
'hops'	0	0	0
'xid'	'xid' from client DHCPDISCOVER message	'xid' from client DHCPREQUEST message	'xid' from client DHCPREQUEST message
'secs'	0	0	0
'ciaddr'	0	'ciaddr' from DHCPREQUEST or 0	'ciaddr' from DHCPREQUEST or 0
'yiaddr'	IP address offered to client	IP address assigned to client	0
'siaddr'	IP address of next bootstrap server	IP address of next bootstrap server	0
'flags'	if 'giaddr' is not 0 then 'flags' from client message else 0		
'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 options	Server host name or options	(unused)
'file'	Client boot file name or options	Client boot file name or options	(unused)
'options'	options	options	

Option	DHCPOFFER	DHCPACK	DHCPNAK
Requested IP address	MUST NOT	MUST NOT	MUST NOT
IP address lease time	MUST	MUST	MUST NOT
Use 'file'/'sname' 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 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 'client identifier' option, the server will use the contents of the 'chaddr' field to identify the client's lease.

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 0xffffffff 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.

```
 --------                               -------
|        | +-------------------------->|       |<-------------------+
| INIT/  | |     +-------------------->| INIT  |                    |
| REBOOT |DHCPNAK/         +---------->|       |<---+               |
|        |Restart|         |            -------     |               |
 --------  |  DHCPNAK/     |               |                        |
    |      Discard offer   |      -/Send DHCPDISCOVER               |
-/Send DHCPREQUEST         |               |                        |
    |      |     |      DHCPACK            v        |               |
 -----------     |   (not accept.)/   -----------   |               |
|           |    |  Send DHCPDECLINE |           |  |               |
| REBOOTING |    |         |         | SELECTING |  |               |
|           |    |        /          |           |  |               |
 -----------     |       /            -----------   |               |
    |            |      /                  |        |               |
DHCPACK/         |     /  +----------------+        |               |
Record lease,    |    |   v                         |               |
set timers      ------------                        |               |
    |   +----->|            |             DHCPNAK, Lease expired/   |
    |   |      | REQUESTING |                  Halt network         |
    DHCPOFFER/ |            |                       |               |
    Discard     ------------                        |               |
    |   |        |        |                   -----------           |
    |   +--------+     DHCPACK/              |           |          |
    |              Record lease, set    -----| REBINDING |          |
    |                timers T1, T2     /     |           |          |
    |                     |        DHCPACK/   -----------           |
    |                     v     Record lease, set   ^               |
    +----------------> -------      /Timers T1,T2   |               |
               +----->|       |<---+                |               |
               |      | BOUND |<---+                |               |
  DHCPOFFER, DHCPACK, |       |    |            T2 expires/   DHCPNAK/
   DHCPNAK/Discard     -------     |             Broadcast  Halt network
               |       | |         |            DHCPREQUEST         |
               +-------+ |        DHCPACK/          |               |
                    T1 expires/   Record lease, set |               |
                 Send DHCPREQUEST timers T1, T2     |               |
                 to leasing server |                |               |
                         |   ----------             |               |
                         |  |          |------------+               |
                         +->| RENEWING |                            |
                            |          |----------------------------+
                             ----------
```

Figure 5: State-transition diagram for DHCP clients

Field	DHCPDISCOVER	DHCPREQUEST	DHCPDECLINE, 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' flag if client requires broadcast reply 0	Set 'BROADCAST' flag if client requires broadcast reply	
'ciaddr'	0	previously allocated newtork address	ciaddr
'yiaddr'	0	0	0
'siaddr'	0	0	0
'giaddr'	0	0	0
'chaddr'	client's hardware address	client's hardware address	client's hardware address
'sname'	options, if indicated in 'sname/file' option; otherwise unused	options, if indicated in 'sname/file' option; otherwise unused	(unused)
'file'	options, if indicated in 'sname/file' option; otherwise 'generic' name or null	options, if indicated in 'sname/file' option; otherwise 'generic' name or null	(unused)
'options'	options	options	(unused)

Option	DHCPDISCOVER	DHCPREQUEST	DHCPDECLINE, 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/ DHCPRELEASE
Client identifier	MAY	MAY	MAY
Class identifier	SHOULD	SHOULD	MUST NOT
Server identifier	MUST NOT	MUST (after DHCPDISCOVER), MUST NOT (when 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 'parameter request list' option. 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 MUST NOT incldue 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 '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 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 * duration_of_lease). T2 defaults to (0.875 * 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 an 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 1542, 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

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)

DECLARATION OF SANDY GINOZA FOR IETF
RFC 2131: (DYNAMIC HOST CONFIGURATION PROTOCOL)

I, Sandy Ginoza, hereby declare that all statements made herein are of my own knowledge and are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code:

1. I am an employee of Association Management Solutions, LLC (AMS), which acts under contract to the IETF Administration LLC (IETF) as the operator of the RFC Production Center. The RFC Production Center is part of the "RFC Editor" function, which prepares documents for publication and places files in an online repository for the authoritative Request for Comments (RFC) series of documents (RFC Series), and preserves records relating to these documents. The RFC Series includes, among other things, the series of Internet standards developed by the IETF. I hold the position of Director of the RFC Production Center. I began employment with AMS in this capacity on 6 January 2010.

2. Among my responsibilities as Director of the RFC Production Center, I act as the custodian of records relating to the RFC Series, and I am familiar with the record keeping practices relating to the RFC Series, including the creation and maintenance of such records.

3. From June 1999 to 5 January 2010, I was an employee of the Information Sciences Institute at University of Southern California (ISI). I held various position titles with the RFC Editor project at ISI, ending with Senior Editor.

4. The RFC Editor function was conducted by ISI under contract to the United States government prior to 1998. In 1998, ISOC, in furtherance of its IETF activity, entered into

the first in a series of contracts with ISI providing for ISI's performance of the RFC Editor function. Beginning in 2010, certain aspects of the RFC Editor function were assumed by the RFC Production Center operation of AMS under contract to ISOC (acting through its IETF function and, in particular, the IETF Administrative Oversight Committee (now the IETF Administration LLC (IETF)). The business records of the RFC Editor function as it was conducted by ISI are currently housed on the computer systems of AMS, as contractor to the IETF.

5. I make this declaration based on my personal knowledge and information contained in the business records of the RFC Editor as they are currently housed at AMS, or confirmation with other responsible RFC Editor personnel with such knowledge.

6. Prior to 1998, the RFC Editor's regular practice was to publish RFCs, making them available from a repository via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication.

7. Since 1998, the RFC Editor's regular practice was to publish RFCs, making them available on the RFC Editor website or via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication. The announcement would go out to all subscribers and a contemporaneous electronic record of the announcement is kept in the IETF mail archive that is available online.

8. Beginning in 1998, any RFC published on the RFC Editor website or via FTP was reasonably accessible to the public and was disseminated or otherwise available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable

diligence could have located it. In particular, the RFCs were indexed and placed in a public repository.

9. The RFCs are kept in an online repository in the course of the RFC Editor's regularly conducted activity and ordinary course of business. The records are made pursuant to established procedures and are relied upon by the RFC Editor in the performance of its functions.

10. It is the regular practice of the RFC Editor to make and keep the RFC records.

11. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 2131 was no later than April 1997, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as an exhibit.

Pursuant to Section 1746 of Title 28 of United States Code, I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct and that the foregoing is based upon personal knowledge and information and is believed to be true.

Date: 1 June 2020

By: ______________________
Sandy Ginoza

4842-3683-7821

Network Working Group   R. Droms
Request for Comments: 2131   Bucknell University
Obsoletes: 1541   March 1997
Category: Standards Track

Dynamic Host Configuration Protocol

Status of this memo

Abstract

The Dynamic Host Configuration Protocol (DHCP) provides a framework for passing configuration information to hosts on a TCPIP 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, 21], and DHCP participants can interoperate with BOOTP participants [9].

Table of Contents

List of Figures

List of Tables

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 client. In "dynamic allocation", DHCP assigns an IP address to a client for a limited period of time (or until the client explicitly relinquishes the address). In "manual allocation", a client's IP address is assigned by the network administrator, and DHCP is used simply to convey the assigned address to the client. 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 client to which it was assigned. Thus, dynamic allocation is particularly useful for assigning an address to a client that will be connected to the network only temporarily or for sharing a limited pool of IP addresses among a group of clients that do not need permanent IP addresses. Dynamic allocation may also be a good choice for assigning an IP address to a new client being permanently connected to a network where IP addresses are sufficiently scarce that it is important to reclaim them when old clients 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, 21] and to allow interoperability of existing BOOTP clients with DHCP servers. Using BOOTP relay agents eliminates the necessity of having a DHCP server on each physical network segment.

1.1 Changes to RFC 1541

This document updates the DHCP protocol specification that appears in RFC1541. A new DHCP message type, DHCPINFORM, has been added; see section 3.4, 4.3 and 4.4 for details. The classing mechanism for identifying DHCP clients to DHCP servers has been extended to include "vendor" classes as defined in sections 4.2 and 4.3. The minimum lease time restriction has been removed. Finally, many editorial changes have been made to clarify the text as a result of experience gained in DHCP interoperability tests.

1.2 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]. The Address Resolution Protocol (ARP) has been proposed 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.

1.3 Problem definition and issues

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

Not all of these parameters are required for a newly initialized client. 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 client parameters not directly related to the IP protocol. DHCP also does not address registration of newly configured clients with the Domain Name System (DNS) [12, 13].

DHCP is not intended for use in configuring routers.

1.4 Requirements

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

- "MUST"

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

- "MUST NOT"

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

- "SHOULD"

  This word or the adjective "RECOMMENDED" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

- "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.

- "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.

1.5 Terminology

This document uses the following terms:

- "DHCP client"

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

- "DHCP server"

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

- "BOOTP relay agent"

  A BOOTP relay agent or 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.

- "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.

1.6 Design goals

The following list gives general design goals for DHCP.

- 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 Clients should require no manual configuration. Each client 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 manual configuration for individual clients. Under normal circumstances, the network manager should not have to enter any per-client 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 relay agents.

- o A DHCP client 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 RFC 1542 [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 DHCP client at a time,

- o Retain DHCP client configuration across DHCP client reboot. A DHCP client should, whenever possible, be assigned the same configuration parameters (e.g., network address) in response to each request,

- o Retain DHCP client configuration across server reboots, and, whenever possible, a DHCP client should be assigned the same configuration parameters despite restarts of the DHCP mechanism,

- o Allow automated assignment of configuration parameters to new clients to avoid hand configuration for new clients,

- o Support fixed or permanent allocation of configuration parameters to specific clients.

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. RFC 1542 [2] 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 finite 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."

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     op (1)    |   htype (1)   |   hlen (1)    |   hops (1)    |
   +---------------+---------------+---------------+---------------+
   |                            xid (4)                            |
   +-------------------------------+-------------------------------+
   |           secs (2)            |           flags (2)           |
   +-------------------------------+-------------------------------+
   |                          ciaddr  (4)                          |
   +---------------------------------------------------------------+
   |                          yiaddr  (4)                          |
   +---------------------------------------------------------------+
   |                          siaddr  (4)                          |
   +---------------------------------------------------------------+
   |                          giaddr  (4)                          |
   +---------------------------------------------------------------+
   |                                                               |
   |                          chaddr  (16)                         |
   |                                                               |
   |                                                               |
   +---------------------------------------------------------------+
   |                                                               |
   |                          sname   (64)                         |
   +---------------------------------------------------------------+
   |                                                               |
   |                          file    (128)                        |
   +---------------------------------------------------------------+
   |                                                               |
   |                          options (variable)                   |
   +---------------------------------------------------------------+
```

Figure 1: Format of a DHCP message

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 'chaddr' is used both as a hardware address for transmission of BOOTP reply messages and as a client identifier. The 'client identifier' is an opaque key, not to be interpreted by the server; for example, the 'client identifier' 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. The 'client identifier' chosen by a DHCP client MUST be unique to that client within the subnet to which the client is attached. If the client uses a 'client identifier' in one message, it MUST use that same identifier in all subsequent messages, to ensure that all servers correctly identify the client.

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.

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; 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 began address acquisition or renewal process.
flags	2	Flags (see figure 2).
ciaddr	4	Client IP address; only filled in if client is in BOUND, RENEW or REBINDING state and can respond to ARP requests.
yiaddr	4	'your' (client) IP address.
siaddr	4	IP address of next server to use in bootstrap; returned in DHCPOFFER, DHCPACK 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	var	Optional parameters field. See the options documents for a list of defined options.

Table 1: Description of fields in a DHCP message

The 'options' field is now variable length. A DHCP client must be prepared to receive DHCP messages with an 'options' field of at least length 312 octets. This requirement implies that a DHCP client must be prepared to receive a message of up to 576 octets, the minimum IP datagram size an IP 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.

In the case of a client using DHCP for initial configuration (before the client's TCP/IP software has been completely configured), DHCP requires creative use of the client's TCP/IP software and liberal interpretation of RFC 1122. The TCP/IP software SHOULD accept and forward to the IP layer any IP packets delivered to the client's hardware address before the IP address is configured; DHCP servers and BOOTP relay agents may not be able to deliver DHCP messages to clients that cannot accept hardware unicast datagrams before the TCP/IP software is configured.

To work around some clients that cannot accept IP unicast datagrams before the TCP/IP software is configured as discussed in the previous paragraph, 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 'flags' field.

```
                    1 1 1 1 1 1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|B|             MBZ             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

B: BROADCAST flag

MBZ: MUST BE ZERO (reserved for future use)

Figure 2: Format of the 'flags' field

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) (note that the "hardware-address" should be typed by the type of hardware to accommodate possible duplication of hardware addresses resulting from bit-ordering problems in a mixed-media, bridged network) 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 DHCP client that has been moved to a different subnet or has changed hardware addresses (perhaps because the network interface failed and was replaced). The protocol defines that the key will be (IP-subnet-number, hardware-address) unless the client explicitly supplies an identifier using the 'client identifier' option. 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 clients. 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 client 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 allocating server SHOULD probe the reused address before allocating the address, e.g., with an ICMP echo request, and the client SHOULD 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 1497 [17]). The remainder of the 'options' field consists of a list of tagged parameters that are called "options". All of the "vendor extensions" listed in RFC 1497 are also DHCP options. RFC 1533 gives the complete set of options defined for use with DHCP.

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. When allocating a new address, servers SHOULD check that the offered network address is not

already in use; e.g., the server may probe the offered address with an ICMP Echo Request. Servers SHOULD be implemented so that network administrators MAY choose to disable probes of newly allocated addresses. The server transmits the DHCPOFFER message to the client, using the BOOTP relay agent if necessary.

Message	Use
DHCPDISCOVER -	Client broadcast to locate available servers.
DHCPOFFER -	Server to client in response to DHCPDISCOVER with offer of configuration parameters.
DHCPREQUEST -	Client message to servers either (a) requesting offered parameters from one server and implicitly declining offers from all others, (b) confirming correctness of previously allocated address after, e.g., system reboot, or (c) extending the lease on a particular network address.
DHCPACK -	Server to client with configuration parameters, including committed network address.
DHCPNAK -	Server to client indicating client's notion of network address is incorrect (e.g., client has moved to new subnet) or client's lease as expired
DHCPDECLINE -	Client to server indicating network address is already in use.
DHCPRELEASE -	Client to server relinquishing network address and cancelling remaining lease.
DHCPINFORM -	Client to server, asking only for local configuration parameters; client already has externally configured network address.

Table 2: DHCP messages

```
                Server          Client          Server
            (not selected)                    (selected)

                  v               v               v
                  |               |               |
                  |     Begins initialization     |
                  |               |               |
                  | _____________/|\____________  |
                  |/DHCPDISCOVER | DHCPDISCOVER  \|
                  |               |               |
              Determines          |          Determines
             configuration        |         configuration
                  |               |               |
                  |\             |  ____________/ |
                  | \________    | /DHCPOFFER     |
                  | DHCPOFFER\   |/               |
                  |           \  |                |
                  |       Collects replies        |
                  |             \|                |
                  |     Selects configuration     |
                  |               |               |
                  | _____________/|\____________  |
                  |/ DHCPREQUEST  |  DHCPREQUEST\ |
                  |               |               |
                  |               |     Commits configuration
                  |               |               |
                  |               | _____________/|
                  |               |/ DHCPACK      |
                  |               |               |
                  |    Initialization complete    |
                  |               |               |
                  .               .               .
                  .               .               .
                  |               |               |
                  |      Graceful shutdown        |
                  |               |               |
                  |               |\ ____________ |
                  |               | DHCPRELEASE  \|
                  |               |               |
                  |               |        Discards lease
                  |               |               |
                  v               v               v
```

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

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 that MAY include other options specifying desired configuration values. The 'requested IP address' option MUST be set to the value of 'yiaddr' in the DHCPOFFER message from the server. This DHCPREQUEST message is broadcast and relayed through DHCP/BOOTP relay agents. To help ensure that any 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 'client identifier' or '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. Any configuration parameters in the DHCPACK message SHOULD NOT conflict with those in the earlier DHCPOFFER message to which the client is responding. The server SHOULD NOT check the offered network address at this point. The 'yiaddr' field in the DHCPACK messages 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.

5. The client receives the DHCPACK message with configuration parameters. The client SHOULD perform a final check on the parameters (e.g., ARP for allocated network address), and notes the duration of the lease specified in the DHCPACK message. At this

point, the client is configured. If the client detects that the address is already in use (e.g., through the use of ARP), the client MUST send 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. The client should choose to retransmit the DHCPREQUEST enough times to give adequate probability of contacting the server without causing the client (and the user of that client) to wait overly long before giving up; e.g., a client retransmitting as described in section 4.1 might retransmit the DHCPREQUEST message four times, for a total delay of 60 seconds, before restarting the initialization procedure. If the client receives neither a DHCPACK or a DHCPNAK message after employing the retransmission algorithm, 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 with its 'client identifier', or 'chaddr' and network address in the DHCPRELEASE message. If the client used a 'client identifier' when it obtained the lease, it MUST use the same 'client identifier' in the DHCPRELEASE message.

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

If a client remembers and wishes to reuse a previously allocated network address, 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 message includes the client's network address in the 'requested IP address' option. As the client has not received its network address, it MUST NOT fill in the 'ciaddr' field. BOOTP relay agents pass the message on to DHCP servers not on the same subnet. If the client used a 'client identifier' to obtain its address, the client MUST use the same 'client identifier' in the DHCPREQUEST message.

2. Servers with knowledge of the client's configuration parameters respond with a DHCPACK message to the client. Servers SHOULD NOT check that the client's network address is already in use; the client may respond to ICMP Echo Request messages at this point.

```
                Server          Client          Server

                  v               v               v
                  |               |               |
                  |             Begins            |
                  |         initialization        |
                  |               |               |
                  |              /|\              |
                  |   _________ __/ | \__________ |
                  | /DHCPREQU EST |  DHCPREQUEST\ |
                  |/              |              \|
                  |               |               |
               Locates            |            Locates
            configuration         |         configuration
                  |               |               |
                  |\              |              /|
                  | \             |  ___________/ |
                  |  \            | /  DHCPACK    |
                  |   \ _______   |/              |
                  |    DHCPACK\   |               |
                  |          Initialization       |
                  |             complete          |
                  |              \|               |
                  |               |               |
                  |          (Subsequent          |
                  |            DHCPACKS           |
                  |            ignored)           |
                  |               |               |
                  |               |               |
                  v               v               v
```

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

If the client's request is invalid (e.g., the client has moved to a new subnet), servers SHOULD respond with a DHCPNAK message to the client. Servers SHOULD NOT respond if their information is not guaranteed to be accurate. For example, a server that identifies a request for an expired binding that is owned by another server SHOULD NOT respond with a DHCPNAK unless the servers are using an explicit mechanism to maintain coherency among the servers.

If 'giaddr' is 0x0 in the DHCPREQUEST message, the client is on the same subnet as the server. The server MUST broadcast the DHCPNAK message to the 0xffffffff broadcast address because the client may not have a correct network address or subnet mask, and the client may not be answering ARP requests. Otherwise, the server MUST send the DHCPNAK message to the IP address of the BOOTP relay agent, as recorded in 'giaddr'. The relay agent will, in turn, forward the message directly to the client's hardware address, so that the DHCPNAK can be delivered even if the client has moved to a new network.

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 specified in the DHCPACK message. The specific lease is implicitly identified by the 'client identifier' or 'chaddr' and the network address. At this point, the client is configured.

   If the client detects that the IP address in the DHCPACK message is already in use, the client MUST send 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.

   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 client retransmits the DHCPREQUEST according to the retransmission algorithm in section 4.1. The client should choose to retransmit the DHCPREQUEST enough times to give adequate probability of contacting the server without causing the client (and the user of that client) to wait overly long before giving up; e.g., a client retransmitting as described in section 4.1 might retransmit the

DHCPREQUEST message four times, for a total delay of 60 seconds, before restarting the initialization procedure. If the client receives neither a DHCPACK or a DHCPNAK message after employing the retransmission algorithm, 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 its 'client identifier', or 'chaddr' and network address in the DHCPRELEASE message.

   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".

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 Obtaining parameters with externally configured network address

If a client has obtained a network address through some other means (e.g., manual configuration), it may use a DHCPINFORM request message to obtain other local configuration parameters. Servers receiving a DHCPINFORM message construct a DHCPACK message with any local configuration parameters appropriate for the client without: allocating a new address, checking for an existing binding, filling in 'yiaddr' or including lease time parameters. The servers SHOULD unicast the DHCPACK reply to the address given in the 'ciaddr' field of the DHCPINFORM message.

The server SHOULD check the network address in a DHCPINFORM message for consistency, but MUST NOT check for an existing lease. The server forms a DHCPACK message containing the configuration parameters for the requesting client and sends the DHCPACK message directly to the client.

3.5 Client 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. If the client includes a list of parameters in a DHCPDISCOVER message, it MUST include that list in any subsequent DHCPREQUEST messages.

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 'requested IP address' option to suggest that a particular IP address be assigned, and may include the 'IP address lease time' option to suggest the lease time it would like. Other options representing "hints" at configuration parameters are allowed in a DHCPDISCOVER or DHCPREQUEST message. However, additional options may be ignored by servers, and multiple servers may, therefore, not return identical values for some options. The 'requested IP address' option is to be filled in only in a DHCPREQUEST message when the client is verifying network parameters obtained previously. The client fills in the 'ciaddr' field only when correctly configured with an IP address in BOUND, RENEWING or REBINDING state.

If a server receives a DHCPREQUEST message with an invalid 'requested IP address', 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.

3.6 Use of DHCP in clients with multiple interfaces

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

3.7 When clients should use DHCP

A client 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 client's or user's knowledge.

If a client has knowledge of a previous network address and is unable to contact a local DHCP server, the client may continue to use the previous network address until the lease for that address expires. If the lease expires before the client can contact a DHCP server, the client 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). A server with multiple network address (e.g., a multi-homed host) MAY use any of its network addresses in outgoing DHCP messages.

The 'server identifier' field is used both to identify a DHCP server in a DHCP message and as a destination address from clients to servers. A server with multiple network addresses MUST be prepared to to accept any of its network addresses as identifying that server in a DHCP message. To accommodate potentially incomplete network connectivity, a server MUST choose an address as a 'server identifier' that, to the best of the server's knowledge, is reachable from the client. For example, if the DHCP server and the DHCP client are connected to the same subnet (i.e., the 'giaddr' field in the message from the client is zero), the server SHOULD select the IP address the server is using for communication on that subnet as the 'server identifier'. If the server is using multiple IP addresses on that subnet, any such address may be used. If the server has received a message through a DHCP relay agent, the server SHOULD choose an address from the interface on which the message was recieved as the 'server identifier' (unless the server has other, better information on which to make its choice). DHCP clients MUST use the IP address provided in the 'server identifier' option for any unicast requests to the DHCP server.

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 BOOTP relay agent whose address appears in 'giaddr'. If the 'giaddr' field is zero and the 'ciaddr' field is nonzero, then the server unicasts DHCPOFFER and DHCPACK messages to the address in 'ciaddr'. If 'giaddr' is zero and 'ciaddr' is zero, and the broadcast bit is set, then the server broadcasts DHCPOFFER and DHCPACK messages to 0xffffffff. If the broadcast bit is not set and 'giaddr' is zero and 'ciaddr' is zero, then the server unicasts DHCPOFFER and DHCPACK messages to the client's hardware address and 'yiaddr' address. In all cases, when 'giaddr' is zero, the server broadcasts any DHCPNAK messages to 0xffffffff.

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 RFC 1533. If the 'option overload' option is present in the 'options' field, the options in the 'options' field MUST be terminated by an 'end' option, and MAY contain one or more 'pad' options to fill the 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 'option overload' option indicates that the 'file' field contains DHCP options), followed by the 'sname' field.

The values to be passed in an 'option' tag may be too long to fit in the 255 octets available to a single option (e.g., a list of routers in a 'router' option [21]). Options may appear only once, unless otherwise specified in the options document. The client concatenates the values of multiple instances of the same option into a single parameter list for configuration.

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 between retransmissions SHOULD be chosen to allow sufficient time for replies from the server to be delivered based on the characteristics of the internetwork between the client and the server. For example, in a 10Mb/sec Ethernet internetwork, the delay before the first retransmission SHOULD 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 SHOULD be 8 seconds randomized by the value of a uniform number chosen from the range -1 to +1. The retransmission delay SHOULD 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 'xid' field is used by the client to match incoming DHCP messages with pending requests. A DHCP client MUST choose 'xid's in such a way as to minimize the chance of using an 'xid' identical to one used by another client. For example, a client may choose a different, random initial 'xid' each time the client is rebooted, and subsequently use sequential 'xid's until the next reboot. Selecting a new 'xid' for each retransmission is an implementation decision. A client may choose to reuse the same 'xid' or select a new 'xid' for each retransmitted message.

Normally, DHCP servers and BOOTP relay agents attempt to deliver DHCPOFFER, DHCPACK and DHCPNAK messages directly to the client using uicast 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 0xffffffff) 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 0xffffffff) 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 clients attached to the network, may choose to configure DHCP servers to respond only to clients 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 vendor class options included in DHCPDISCOVER or DHCPREQUEST messages when determining the correct parameters for a particular client.

A DHCP server needs to 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 supplies a 'client identifier', the client MUST use the same 'client identifier' in all subsequent messages, and the server MUST use that identifier to identify the client. If the client does not provide a 'client identifier' option, the server MUST use the contents of the 'chaddr' field to identify the client. It is crucial for a DHCP client to use an identifier unique within the subnet to which the client is attached in the 'client identifier' option. Use of 'chaddr' as the client's unique identifier may cause unexpected results, as that identifier may be associated with a hardware interface that could be moved to a new client. Some sites may choose to use a manufacturer's serial number as the 'client identifier', to avoid unexpected changes in a clients network address due to transfer of hardware interfaces among computers. Sites may also choose to use a DNS name as the 'client identifier', causing address leases to be associated with the DNS name rather than a specific hardware box.

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 'vendor class identifier' values.

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
- o DHCPINFORM

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. If an address is available, the new address SHOULD be chosen as follows:

- o The client's current address as recorded in the client's current binding, ELSE

- o The client's previous address as recorded in the client's (now expired or released) 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; the address is selected based on the subnet from which the message was received (if 'giaddr' is 0) or on the address of the relay agent that forwarded the message ('giaddr' when not 0).

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.

Note that, in some network architectures (e.g., internets with more than one IP subnet assigned to a physical network segment), it may be the case that the DHCP client should be assigned an address from a different subnet than the address recorded in 'giaddr'. Thus, DHCP does not require that the client be assigned as address from the subnet in 'giaddr'. A server is free to choose some other subnet, and it is beyond the scope of the DHCP specification to describe ways in which the assigned IP address might be chosen.

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.

Field	DHCPOFFER	DHCPACK	DHCPNAK
'op'	BOOTREPLY	BOOTREPLY	BOOTREPLY
'htype'	(From "Assigned Numbers" RFC)		
'hlen'	(Hardware address length in octets)		
'hops'	0	0	0
'xid'	'xid' from client DHCPDISCOVER message	'xid' from client DHCPREQUEST message	'xid' from client DHCPREQUEST message
'secs'	0	0	0
'ciaddr'	0	'ciaddr' from DHCPREQUEST or 0	0
'yiaddr'	IP address offered to client	IP address assigned to client	0
'siaddr'	IP address of next bootstrap server	IP address of next bootstrap server	0
'flags'	'flags' from client DHCPDISCOVER message	'flags' from client DHCPREQUEST message	'flags' from client DHCPREQUEST message
'giaddr'	'giaddr' from client DHCPDISCOVER message	'giaddr' from client DHCPREQUEST message	'giaddr' from client DHCPREQUEST message
'chaddr'	'chaddr' from client DHCPDISCOVER message	'chaddr' from client DHCPREQUEST message	'chaddr' from client DHCPREQUEST message
'sname'	Server host name or options	Server host name or options	(unused)
'file'	Client boot file name or options	Client boot file name or options	(unused)
'options'	options	options	

Option	DHCPOFFER	DHCPACK	DHCPNAK
Requested IP address	MUST NOT	MUST NOT	MUST NOT
IP address lease time	MUST	MUST (DHCPREQUEST) MUST NOT (DHCPINFORM)	MUST NOT
Use 'file'/'sname' 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	MAY
Vendor class identifier	MAY	MAY	MAY
Server identifier	MUST	MUST	MUST
Maximum message size	MUST NOT	MUST NOT	MUST NOT
All others	MAY	MAY	MUST NOT

Table 3: Fields and options used by DHCP servers

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 client behavior regardless of 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,

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, The server MUST supply as many of the requested parameters as possible and MUST omit any parameters it cannot provide. The server MUST include each requested parameter only once unless explicitly allowed in the DHCP Options and BOOTP Vendor Extensions document.

- o Any parameters from the existing binding that differ from the Host Requirements Document defaults,

- o Any parameters specific to this client (as identified by the contents of 'chaddr' or 'client identifier' 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 'vendor 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 vendor class identifiers and the client's classes identified in the server,

- o Parameters with non-default values on the client's subnet.

The server MAY choose to return the 'vendor class identifier' used to determine the parameters in the DHCPOFFER message to assist the client in selecting which DHCPOFFER to accept. 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, from a client verifying a previously allocated IP address or from a client extending the lease on a network 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 verify or extend an existing lease. If the client uses a 'client identifier' in a DHCPREQUEST message, it MUST use that same 'client identifier' in all subsequent messages. If the client included a list of requested parameters in a DHCPDISCOVER message, it MUST include that list in all subsequent messages.

Any configuration parameters in the DHCPACK message SHOULD NOT conflict with those in the earlier DHCPOFFER message to which the client is responding. The client SHOULD use the parameters in the DHCPACK message for configuration.

Clients send DHCPREQUEST messages as follows:

- o DHCPREQUEST generated during SELECTING state:

  Client inserts the address of the selected server in 'server identifier', 'ciaddr' MUST be zero, 'requested IP address' MUST be filled in with the yiaddr value from the chosen DHCPOFFER.

  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.

- o DHCPREQUEST generated during INIT-REBOOT state:

  'server identifier' MUST NOT be filled in, 'requested IP address' option MUST be filled in with client's notion of its previously assigned address. 'ciaddr' MUST be zero. The client is seeking to verify a previously allocated, cached configuration. Server SHOULD send a DHCPNAK message to the client if the 'requested IP address' is incorrect, or is on the wrong network.

  Determining whether a client in the INIT-REBOOT state is on the correct network is done by examining the contents of 'giaddr', the 'requested IP address' option, and a database lookup. If the DHCP server detects that the client is on the wrong net (i.e., the result of applying the local subnet mask or remote subnet mask (if 'giaddr' is not zero) to 'requested IP address' option value doesn't match reality), then the server SHOULD send a DHCPNAK message to the client.

If the network is correct, then the DHCP server should check if the client's notion of its IP address is correct. If not, then the server SHOULD send a DHCPNAK message to the client. If the DHCP server has no record of this client, then it MUST remain silent, and MAY output a warning to the network administrator. This behavior is necessary for peaceful coexistence of non-communicating DHCP servers on the same wire.

If 'giaddr' is 0x0 in the DHCPREQUEST message, the client is on the same subnet as the server. The server MUST broadcast the DHCPNAK message to the 0xffffffff broadcast address because the client may not have a correct network address or subnet mask, and the client may not be answering ARP requests.

If 'giaddr' is set in the DHCPREQUEST message, the client is on a different subnet. The server MUST set the broadcast bit in the DHCPNAK, so that the relay agent will broadcast the DHCPNAK to the client, because the client may not have a correct network address or subnet mask, and the client may not be answering ARP requests.

o DHCPREQUEST generated during RENEWING state:

'server identifier' MUST NOT be filled in, 'requested IP address' option MUST NOT be filled in, 'ciaddr' MUST be filled in with client's IP address. In this situation, the client is completely configured, and is trying to extend its lease. This message will be unicast, so no relay agents will be involved in its transmission. Because 'giaddr' is therefore not filled in, the DHCP server will trust the value in 'ciaddr', and use it when replying to the client.

A client MAY choose to renew or extend its lease prior to T1. The server may choose not to extend the lease (as a policy decision by the network administrator), but should return a DHCPACK message regardless.

o DHCPREQUEST generated during REBINDING state:

'server identifier' MUST NOT be filled in, 'requested IP address' option MUST NOT be filled in, 'ciaddr' MUST be filled in with client's IP address. In this situation, the client is completely configured, and is trying to extend its lease. This message MUST be broadcast to the 0xffffffff IP broadcast address. The DHCP server SHOULD check 'ciaddr' for correctness before replying to the DHCPREQUEST.

The DHCPREQUEST from a REBINDING client is intended to accommodate sites that have multiple DHCP servers and a mechanism for maintaining consistency among leases managed by multiple servers. A DHCP server MAY extend a client's lease only if it has local administrative authority to do so.

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 available 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.

4.3.5 DHCPINFORM message

The server responds to a DHCPINFORM message by sending a DHCPACK message directly to the address given in the 'ciaddr' field of the DHCPINFORM message. The server MUST NOT send a lease expiration time to the client and SHOULD NOT fill in 'yiaddr'. The server includes other parameters in the DHCPACK message as defined in section 4.3.1.

4.3.6 Client messages

Table 4 details the differences between messages from clients in various states.

	INIT-REBOOT	SELECTING	RENEWING	REBINDING
broad/unicast	broadcast	broadcast	unicast	broadcast
server-ip	MUST NOT	MUST	MUST NOT	MUST NOT
requested-ip	MUST	MUST	MUST NOT	MUST NOT
ciaddr	zero	zero	IP address	IP address

Table 4: Client messages from different states

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

The DHCPINFORM message is not shown in figure 5. A client simply sends the DHCPINFORM and waits for DHCPACK messages. Once the client has selected its parameters, it has completed the configuration process.

Table 5 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.

```
 --------                               -------
|        | +-------------------------->|       |<-------------------+
| INIT-  | |     +-------------------->| INIT  |                    |
| REBOOT |DHCPNAK/         +---------->|       |<---+               |
|        |Restart|         |            -------     |               |
 --------  |  DHCPNAK/     |               |                        |
    |      Discard offer   |      -/Send DHCPDISCOVER               |
-/Send DHCPREQUEST         |               |                        |
    |      |     |      DHCPACK            v        |               |
 -----------     |   (not accept.)/   -----------   |               |
|           |    |  Send DHCPDECLINE |           |                  |
| REBOOTING |    |         |         | SELECTING |<----+            |
|           |    |        /          |           |     |DHCPOFFER/  |
 -----------     |       /            -----------   |  |Collect     |
    |            |      /                  |   |       |  replies   |
DHCPACK/         |     /  +----------------+   +-------+            |
Record lease, set|    |   v   Select offer/                         |
timers T1, T2   ------------  send DHCPREQUEST      |               |
    |   +----->|            |             DHCPNAK, Lease expired/   |
    |   |      | REQUESTING |                  Halt network         |
    DHCPOFFER/ |            |                       |               |
    Discard     ------------                        |               |
    |   |        |        |                   -----------           |
    |   +--------+     DHCPACK/              |           |          |
    |              Record lease, set    -----| REBINDING |          |
    |                timers T1, T2     /     |           |          |
    |                     |        DHCPACK/   -----------           |
    |                     v     Record lease, set   ^               |
    +----------------> -------      /timers T1,T2   |               |
               +----->|       |<---+                |               |
               |      | BOUND |<---+                |               |
  DHCPOFFER, DHCPACK, |       |    |            T2 expires/   DHCPNAK/
   DHCPNAK/Discard     -------     |             Broadcast  Halt network
               |       | |         |            DHCPREQUEST         |
               +-------+ |        DHCPACK/          |               |
                    T1 expires/   Record lease, set |               |
                 Send DHCPREQUEST timers T1, T2     |               |
                 to leasing server |                |               |
                         |   ----------             |               |
                         |  |          |------------+               |
                         +->| RENEWING |                            |
                            |          |----------------------------+
                             ----------
```

Figure 5: State-transition diagram for DHCP clients

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 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, if necessary for delivery of DHCP reply messages. The client MAY include a different unique identifier in the 'client identifier' option, as discussed in section 4.2. If the client included a list of requested parameters in a DHCPDISCOVER message, it MUST include that list in all subsequent messages.

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 0xffffffff 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.

Field	DHCPDISCOVER DHCPINFORM	DHCPREQUEST	DHCPDECLINE, DHCPRELEASE
'op'	BOOTREQUEST	BOOTREQUEST	BOOTREQUEST
'htype'	(From "Assigned Numbers" RFC)		
'hlen'	(Hardware address length in octets)		
'hops'	0	0	0
'xid'	selected by client	'xid' from server DHCPOFFER message	selected by client
'secs'	0 or seconds since DHCP process started	0 or seconds since DHCP process started	0
'flags'	Set 'BROADCAST' flag if client requires broadcast reply	Set 'BROADCAST' flag if client requires broadcast reply	0
'ciaddr'	0 (DHCPDISCOVER) client's network address (DHCPINFORM)	0 or client's network address (BOUND/RENEW/REBIND)	0 (DHCPDECLINE) client's network address (DHCPRELEASE)
'yiaddr'	0	0	0
'siaddr'	0	0	0
'giaddr'	0	0	0
'chaddr'	client's hardware address	client's hardware address	client's hardware address
'sname'	options, if indicated in 'sname/file' option; otherwise unused	options, if indicated in 'sname/file' option; otherwise unused	(unused)
'file'	options, if indicated in 'sname/file' option; otherwise unused	options, if indicated in 'sname/file' option; otherwise unused	(unused)
'options'	options	options	(unused)

Option	DHCPDISCOVER DHCPINFORM	DHCPREQUEST	DHCPDECLINE, DHCPRELEASE
Requested IP address	MAY (DISCOVER) MUST NOT (INFORM)	MUST (in SELECTING or INIT-REBOOT) MUST NOT (in BOUND or RENEWING)	MUST (DHCPDECLINE), MUST NOT (DHCPRELEASE)
IP address lease time	MAY (DISCOVER) MUST NOT (INFORM)	MAY	MUST NOT
Use 'file'/'sname' fields	MAY	MAY	MAY
DHCP message type	DHCPDISCOVER/ DHCPINFORM	DHCPREQUEST	DHCPDECLINE/ DHCPRELEASE
Client identifier	MAY	MAY	MAY
Vendor class identifier	MAY	MAY	MUST NOT
Server identifier	MUST NOT	MUST (after SELECTING) MUST NOT (after INIT-REBOOT, BOUND, RENEWING or REBINDING)	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	MAY	MAY	MUST NOT

Table 5: 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 DHCPACK message. The client SHOULD 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 MUST send a DHCPDECLINE message to the server. 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. The client MUST insert its known network address as a 'requested IP address' option in the DHCPREQUEST message. The client may request specific configuration parameters by including the 'parameter request list' option. 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 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 an externally assigned network address

The client sends a DHCPINFORM message. The client may request specific configuration parameters by including the 'parameter request list' option. The client generates and records a random transaction identifier and inserts that identifier into the 'xid' field. The client places its own network address in the 'ciaddr' field. The client SHOULD NOT request lease time parameters.

The client then unicasts the DHCPINFORM to the DHCP server if it knows the server's address, otherwise it broadcasts the message to the limited (all 1s) broadcast address. DHCPINFORM messages MUST be directed to the 'DHCP server' UDP port.

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

If the client does not receive a DHCPACK within a reasonable period of time (60 seconds or 4 tries if using timeout suggested in section 4.1), then it SHOULD display a message informing the user of the problem, and then SHOULD begin network processing using suitable defaults as per Appendix A.

4.4.4 Use of broadcast and unicast

The DHCP client broadcasts DHCPDISCOVER, DHCPREQUEST and DHCPINFORM messages, unless the client knows the address of a DHCP server. The client unicasts DHCPRELEASE messages to the server. Because the client is declining the use of the IP address supplied by the server, the client broadcasts DHCPDECLINE messages.

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. The client may also use unicast to send DHCPINFORM messages to a known DHCP server. 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.

4.4.5 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. T1 MUST be earlier than T2, which, in turn, MUST be earlier than the time at which the client's lease will expire.

To avoid the need for synchronized clocks, T1 and T2 are expressed in options as relative times [2].

At time T1 the client moves to RENEWING state and sends (via unicast) a DHCPREQUEST message to the server to extend its lease. The client sets the 'ciaddr' field in the DHCPREQUEST to its current network address. 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 '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 T2, 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 * duration_of_lease). T2 defaults to (0.875 * duration_of_lease). Times T1 and T2 SHOULD be chosen with some random "fuzz" around a fixed value, to avoid synchronization of client reacquisition.

A client MAY choose to renew or extend its lease prior to T1. The server MAY choose to extend the client's lease according to policy set by the network administrator. The server SHOULD return T1 and T2, and their values SHOULD be adjusted from their original values to take account of the time remaining on the lease.

In both RENEWING and REBINDING states, if the client receives no response to its DHCPREQUEST message, the client SHOULD wait one-half of the remaining time until T2 (in RENEWING state) and one-half of the remaining lease time (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.

4.4.6 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

The author thanks the many (and too numerous to mention!) members of the DHC WG for their tireless and ongoing efforts in the development of DHCP and this document.

The efforts of J Allard, Mike Carney, Dave Lapp, Fred Lien and John Mendonca in organizing DHCP interoperability testing sessions are gratefully acknowledged.

The development of this document was supported in part by grants from the Corporation for National Research Initiatives (CNRI), Bucknell University and Sun Microsystems.

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 1700, USC/Information Sciences Institute, October 1994.

[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 1542, 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

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)

DECLARATION OF SANDY GINOZA FOR IETF
RFC 2132: (DHCP OPTIONS AND BOOTP VENDOR EXTENSIONS)

I, Sandy Ginoza, hereby declare that all statements made herein are of my own knowledge and are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code:

1. I am an employee of Association Management Solutions, LLC (AMS), which acts under contract to the IETF Administration LLC (IETF) as the operator of the RFC Production Center. The RFC Production Center is part of the "RFC Editor" function, which prepares documents for publication and places files in an online repository for the authoritative Request for Comments (RFC) series of documents (RFC Series), and preserves records relating to these documents. The RFC Series includes, among other things, the series of Internet standards developed by the IETF. I hold the position of Director of the RFC Production Center. I began employment with AMS in this capacity on 6 January 2010.

2. Among my responsibilities as Director of the RFC Production Center, I act as the custodian of records relating to the RFC Series, and I am familiar with the record keeping practices relating to the RFC Series, including the creation and maintenance of such records.

3. From June 1999 to 5 January 2010, I was an employee of the Information Sciences Institute at University of Southern California (ISI). I held various position titles with the RFC Editor project at ISI, ending with Senior Editor.

4. The RFC Editor function was conducted by ISI under contract to the United States government prior to 1998. In 1998, ISOC, in furtherance of its IETF activity, entered into

the first in a series of contracts with ISI providing for ISI's performance of the RFC Editor function. Beginning in 2010, certain aspects of the RFC Editor function were assumed by the RFC Production Center operation of AMS under contract to ISOC (acting through its IETF function and, in particular, the IETF Administrative Oversight Committee (now the IETF Administration LLC (IETF)). The business records of the RFC Editor function as it was conducted by ISI are currently housed on the computer systems of AMS, as contractor to the IETF.

5. I make this declaration based on my personal knowledge and information contained in the business records of the RFC Editor as they are currently housed at AMS, or confirmation with other responsible RFC Editor personnel with such knowledge.

6. Prior to 1998, the RFC Editor's regular practice was to publish RFCs, making them available from a repository via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication.

7. Since 1998, the RFC Editor's regular practice was to publish RFCs, making them available on the RFC Editor website or via FTP. When a new RFC was published, an announcement of its publication, with information on how to access the RFC, would be typically sent out within 24 hours of the publication. The announcement would go out to all subscribers and a contemporaneous electronic record of the announcement is kept in the IETF mail archive that is available online.

8. Beginning in 1998, any RFC published on the RFC Editor website or via FTP was reasonably accessible to the public and was disseminated or otherwise available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable

diligence could have located it. In particular, the RFCs were indexed and placed in a public repository.

9. The RFCs are kept in an online repository in the course of the RFC Editor's regularly conducted activity and ordinary course of business. The records are made pursuant to established procedures and are relied upon by the RFC Editor in the performance of its functions.

10. It is the regular practice of the RFC Editor to make and keep the RFC records.

11. Based on the business records for the RFC Editor and the RFC Editor's course of conduct in publishing RFCs, I have determined that the publication date of RFC 2132 was no later than March 1997, at which time it was reasonably accessible to the public either on the RFC Editor website or via FTP from a repository. An announcement of its publication also would have been sent out to subscribers within 24 hours of its publication. A copy of that RFC is attached to this declaration as an exhibit.

Pursuant to Section 1746 of Title 28 of United States Code, I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct and that the foregoing is based upon personal knowledge and information and is believed to be true.

Date: 1 June 2020

By: ______________________

Sandy Ginoza

4811-9865-4397

Network Working Group
Request for Comments: 2132
Obsoletes: 1533
Category: Standards Track

S. Alexander
Silicon Graphics, Inc.
R. Droms
Bucknell University

March 1997

DHCP Options and BOOTP Vendor Extensions

Status of this memo

This document 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" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

The Dynamic Host Configuration Protocol (DHCP) [1] provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the 'options' field of the DHCP message. The data items themselves are also called "options."

This document specifies the current set of DHCP options. Future options will be specified in separate RFCs. The current list of valid options is also available in ftp://ftp.isi.edu/in-notes/iana/assignments [22].

All of the vendor information extensions defined in RFC 1497 [2] may be used as DHCP options. The definitions given in RFC 1497 are included in this document, which supersedes RFC 1497. All of the DHCP options defined in this document, except for those specific to DHCP as defined in section 9, may be used as BOOTP vendor information extensions.

Table of Contents

1. Introduction

This document specifies options for use with both the Dynamic Host Configuration Protocol and the Bootstrap Protocol.

The full description of DHCP packet formats may be found in the DHCP specification document [1], and the full description of BOOTP packet formats may be found in the BOOTP specification document [3]. This document defines the format of information in the last field of DHCP packets ('options') and of BOOTP packets ('vend'). The remainder of this section defines a generalized use of this area for giving information useful to a wide class of machines, operating systems and configurations. Sites with a single DHCP or BOOTP server that is shared among heterogeneous clients may choose to define other, site-specific formats for the use of the 'options' field.

Section 2 of this memo describes the formats of DHCP options and BOOTP vendor extensions. Section 3 describes options defined in previous documents for use with BOOTP (all may also be used with DHCP). Sections 4-8 define new options intended for use with both DHCP and BOOTP. Section 9 defines options used only in DHCP.

References further describing most of the options defined in sections 2-6 can be found in section 12. The use of the options defined in section 9 is described in the DHCP specification [1].

Information on registering new options is contained in section 10.

This document updates the definition of DHCP/BOOTP options that appears in RFC1533. The classing mechanism has been extended to include vendor classes as described in section 8.4 and 9.13. The new procedure for defining new DHCP/BOOTP options in described in section 10. Several new options, including NIS+ domain and servers, Mobile IP home agent, SMTP server, TFTP server and Bootfile server, have been added. Text giving definitions used throughout the document has been added in section 1.1. Text emphasizing the need for uniqueness of client-identifiers has been added to section 9.14.

1.1 Requirements

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

- "MUST"

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

- "MUST NOT"

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

- "SHOULD"

  This word or the adjective "RECOMMENDED" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

- "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.

- "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.

1.2 Terminology

This document uses the following terms:

- "DHCP client"

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

- o "DHCP server"

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

- 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.

2. BOOTP Extension/DHCP Option Field Format

DHCP options have the same format as the BOOTP 'vendor extensions' defined in RFC 1497 [2]. Options may be fixed length or variable length. All options begin with a tag octet, which uniquely identifies the option. Fixed-length options without data consist of only a tag octet. Only options 0 and 255 are fixed length. All other options are variable-length with a length octet following the tag octet. The value of the length octet does not include the two octets specifying the tag and length. The length octet is followed by "length" octets of data. Options containing NVT ASCII data SHOULD NOT include a trailing NULL; however, the receiver of such options MUST be prepared to delete trailing nulls if they exist. The receiver MUST NOT require that a trailing null be included in the data. In the case of some variable-length options the length field is a constant but must still be specified.

Any options defined subsequent to this document MUST contain a length octet even if the length is fixed or zero.

All multi-octet quantities are in network byte-order.

When used with BOOTP, the first four octets of the vendor information field have been assigned to the "magic cookie" (as suggested in RFC 951). This field identifies the mode in which the succeeding data is to be interpreted. The value of the magic cookie is the 4 octet dotted decimal 99.130.83.99 (or hexadecimal number 63.82.53.63) in network byte order.

All of the "vendor extensions" defined in RFC 1497 are also DHCP options.

Option codes 128 to 254 (decimal) are reserved for site-specific options.

Except for the options in section 9, all options may be used with either DHCP or BOOTP.

Many of these options have their default values specified in other documents. In particular, RFC 1122 [4] specifies default values for most IP and TCP configuration parameters.

Many options supply one or more 32-bit IP address. Use of IP addresses rather than fully-qualified Domain Names (FQDNs) may make future renumbering of IP hosts more difficult. Use of these addresses is discouraged at sites that may require renumbering.

3. RFC 1497 Vendor Extensions

This section lists the vendor extensions as defined in RFC 1497. They are defined here for completeness.

3.1. Pad Option

The pad option can be used to cause subsequent fields to align on word boundaries.

The code for the pad option is 0, and its length is 1 octet.

```
 Code
+-----+
|  0  |
+-----+
```

3.2. End Option

The end option marks the end of valid information in the vendor field. Subsequent octets should be filled with pad options.

The code for the end option is 255, and its length is 1 octet.

```
 Code
+-----+
| 255 |
+-----+
```

3.3. Subnet Mask

The subnet mask option specifies the client's subnet mask as per RFC 950 [5].

If both the subnet mask and the router option are specified in a DHCP reply, the subnet mask option MUST be first.

The code for the subnet mask option is 1, and its length is 4 octets.

```
 Code   Len        Subnet Mask
+-----+-----+-----+-----+-----+-----+
|  1  |  4  |  m1 |  m2 |  m3 |  m4 |
+-----+-----+-----+-----+-----+-----+
```

3.4. Time Offset

The time offset field specifies the offset of the client's subnet in seconds from Coordinated Universal Time (UTC). The offset is expressed as a two's complement 32-bit integer. A positive offset indicates a location east of the zero meridian and a negative offset indicates a location west of the zero meridian.

The code for the time offset option is 2, and its length is 4 octets.

```
 Code   Len        Time Offset
+-----+-----+-----+-----+-----+-----+
|  2  |  4  |  n1 |  n2 |  n3 |  n4 |
+-----+-----+-----+-----+-----+-----+
```

3.5. Router Option

The router option specifies a list of IP addresses for routers on the client's subnet. Routers SHOULD be listed in order of preference.

The code for the router option is 3. The minimum length for the router option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  3  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.6. Time Server Option

The time server option specifies a list of RFC 868 [6] time servers available to the client. Servers SHOULD be listed in order of preference.

The code for the time server option is 4. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  4  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.7. Name Server Option

The name server option specifies a list of IEN 116 [7] name servers available to the client. Servers SHOULD be listed in order of preference.

The code for the name server option is 5. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  5  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.8. Domain Name Server Option

The domain name server option specifies a list of Domain Name System (STD 13, RFC 1035 [8]) name servers available to the client. Servers SHOULD be listed in order of preference.

The code for the domain name server option is 6. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  6  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.9. Log Server Option

The log server option specifies a list of MIT-LCS UDP log servers available to the client. Servers SHOULD be listed in order of preference.

The code for the log server option is 7. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  7  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.10. Cookie Server Option

The cookie server option specifies a list of RFC 865 [9] cookie servers available to the client. Servers SHOULD be listed in order of preference.

The code for the log server option is 8. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  8  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.11. LPR Server Option

The LPR server option specifies a list of RFC 1179 [10] line printer servers available to the client. Servers SHOULD be listed in order of preference.

The code for the LPR server option is 9. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  9  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.12. Impress Server Option

The Impress server option specifies a list of Imagen Impress servers available to the client. Servers SHOULD be listed in order of preference.

The code for the Impress server option is 10. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  10 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.13. Resource Location Server Option

This option specifies a list of RFC 887 [11] Resource Location servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 11. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  11 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.14. Host Name Option

This option specifies the name of the client. The name may or may not be qualified with the local domain name (see section 3.17 for the preferred way to retrieve the domain name). See RFC 1035 for character set restrictions.

The code for this option is 12, and its minimum length is 1.

```
 Code   Len                 Host Name
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  12 |  n  |  h1 |  h2 |  h3 |  h4 |  h5 |  h6 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

3.15. Boot File Size Option

This option specifies the length in 512-octet blocks of the default boot image for the client. The file length is specified as an unsigned 16-bit integer.

The code for this option is 13, and its length is 2.

```
 Code   Len   File Size
+-----+-----+-----+-----+
|  13 |  2  |  l1 |  l2 |
+-----+-----+-----+-----+
```

3.16. Merit Dump File

This option specifies the path-name of a file to which the client's core image should be dumped in the event the client crashes. The path is formatted as a character string consisting of characters from the NVT ASCII character set.

The code for this option is 14. Its minimum length is 1.

```
 Code   Len      Dump File Pathname
+-----+-----+-----+-----+-----+-----+---
|  14 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

3.17. Domain Name

This option specifies the domain name that client should use when resolving hostnames via the Domain Name System.

The code for this option is 15. Its minimum length is 1.

```
 Code   Len        Domain Name
+-----+-----+-----+-----+-----+-----+--
|  15 |  n  |  d1 |  d2 |  d3 |  d4 |  ...
+-----+-----+-----+-----+-----+-----+--
```

3.18. Swap Server

This specifies the IP address of the client's swap server.

The code for this option is 16 and its length is 4.

```
 Code   Len    Swap Server Address
+-----+-----+-----+-----+-----+-----+
|  16 |  n  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

3.19. Root Path

This option specifies the path-name that contains the client's root disk. The path is formatted as a character string consisting of characters from the NVT ASCII character set.

The code for this option is 17. Its minimum length is 1.

```
 Code   Len      Root Disk Pathname
+-----+-----+-----+-----+-----+-----+---
|  17 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

3.20. Extensions Path

A string to specify a file, retrievable via TFTP, which contains information which can be interpreted in the same way as the 64-octet vendor-extension field within the BOOTP response, with the following exceptions:

- the length of the file is unconstrained;
- all references to Tag 18 (i.e., instances of the BOOTP Extensions Path field) within the file are ignored.

The code for this option is 18. Its minimum length is 1.

```
 Code   Len      Extensions Pathname
+-----+-----+-----+-----+-----+-----+---
|  18 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

4. IP Layer Parameters per Host

This section details the options that affect the operation of the IP layer on a per-host basis.

4.1. IP Forwarding Enable/Disable Option

This option specifies whether the client should configure its IP layer for packet forwarding. A value of 0 means disable IP forwarding, and a value of 1 means enable IP forwarding.

The code for this option is 19, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  19 |  1  | 0/1 |
+-----+-----+-----+
```

4.2. Non-Local Source Routing Enable/Disable Option

This option specifies whether the client should configure its IP layer to allow forwarding of datagrams with non-local source routes (see Section 3.3.5 of [4] for a discussion of this topic). A value of 0 means disallow forwarding of such datagrams, and a value of 1 means allow forwarding.

The code for this option is 20, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  20 |  1  | 0/1 |
+-----+-----+-----+
```

4.3. Policy Filter Option

This option specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks which specify destination/mask pairs with which to filter incoming source routes.

Any source routed datagram whose next-hop address does not match one of the filters should be discarded by the client.

See [4] for further information.

The code for this option is 21. The minimum length of this option is 8, and the length MUST be a multiple of 8.

```
 Code   Len         Address 1                  Mask 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  21 |  n  |  a1 |  a2 |  a3 |  a4 |  m1 |  m2 |  m3 |  m4 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
        Address 2                  Mask 2
+-----+-----+-----+-----+-----+-----+-----+-----+---
|  a1 |  a2 |  a3 |  a4 |  m1 |  m2 |  m3 |  m4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```

4.4. Maximum Datagram Reassembly Size

This option specifies the maximum size datagram that the client should be prepared to reassemble. The size is specified as a 16-bit unsigned integer. The minimum value legal value is 576.

The code for this option is 22, and its length is 2.

```
 Code   Len      Size
+-----+-----+-----+-----+
|  22 |  2  |  s1 |  s2 |
+-----+-----+-----+-----+
```

4.5. Default IP Time-to-live

This option specifies the default time-to-live that the client should use on outgoing datagrams. The TTL is specified as an octet with a value between 1 and 255.

The code for this option is 23, and its length is 1.

```
 Code   Len   TTL
+-----+-----+-----+
|  23 |  1  | ttl |
+-----+-----+-----+
```

4.6. Path MTU Aging Timeout Option

This option specifies the timeout (in seconds) to use when aging Path MTU values discovered by the mechanism defined in RFC 1191 [12]. The timeout is specified as a 32-bit unsigned integer.

The code for this option is 24, and its length is 4.

```
 Code   Len           Timeout
+-----+-----+-----+-----+-----+-----+
|  24 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

4.7. Path MTU Plateau Table Option

This option specifies a table of MTU sizes to use when performing Path MTU Discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value cannot be smaller than 68.

The code for this option is 25. Its minimum length is 2, and the length MUST be a multiple of 2.

```
 Code   Len     Size 1      Size 2
+-----+-----+-----+-----+-----+-----+---
|  25 |  n  |  s1 |  s2 |  s1 |  s2 | ...
+-----+-----+-----+-----+-----+-----+---
```

5. IP Layer Parameters per Interface

This section details the options that affect the operation of the IP layer on a per-interface basis. It is expected that a client can issue multiple requests, one per interface, in order to configure interfaces with their specific parameters.

5.1. Interface MTU Option

This option specifies the MTU to use on this interface. The MTU is specified as a 16-bit unsigned integer. The minimum legal value for the MTU is 68.

The code for this option is 26, and its length is 2.

```
 Code   Len      MTU
+-----+-----+-----+-----+
|  26 |  2  |  m1 |  m2 |
+-----+-----+-----+-----+
```

5.2. All Subnets are Local Option

This option specifies whether or not the client may assume that all subnets of the IP network to which the client is connected use the same MTU as the subnet of that network to which the client is directly connected. A value of 1 indicates that all subnets share the same MTU. A value of 0 means that the client should assume that some subnets of the directly connected network may have smaller MTUs.

The code for this option is 27, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  27 |  1  | 0/1 |
+-----+-----+-----+
```

5.3. Broadcast Address Option

This option specifies the broadcast address in use on the client's subnet. Legal values for broadcast addresses are specified in section 3.2.1.3 of [4].

The code for this option is 28, and its length is 4.

```
 Code   Len     Broadcast Address
+-----+-----+-----+-----+-----+-----+
|  28 |  4  | b1  | b2  | b3  | b4  |
+-----+-----+-----+-----+-----+-----+
```

5.4. Perform Mask Discovery Option

This option specifies whether or not the client should perform subnet mask discovery using ICMP. A value of 0 indicates that the client should not perform mask discovery. A value of 1 means that the client should perform mask discovery.

The code for this option is 29, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  29 |  1  | 0/1 |
+-----+-----+-----+
```

5.5. Mask Supplier Option

This option specifies whether or not the client should respond to subnet mask requests using ICMP. A value of 0 indicates that the client should not respond. A value of 1 means that the client should respond.

The code for this option is 30, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  30 |  1  | 0/1 |
+-----+-----+-----+
```

5.6. Perform Router Discovery Option

This option specifies whether or not the client should solicit routers using the Router Discovery mechanism defined in RFC 1256 [13]. A value of 0 indicates that the client should not perform router discovery. A value of 1 means that the client should perform router discovery.

The code for this option is 31, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  31 |  1  | 0/1 |
+-----+-----+-----+
```

5.7. Router Solicitation Address Option

This option specifies the address to which the client should transmit router solicitation requests.

The code for this option is 32, and its length is 4.

```
 Code   Len            Address
+-----+-----+-----+-----+-----+-----+
|  32 |  4  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

5.8. Static Route Option

This option specifies a list of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority.

The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination.

The default route (0.0.0.0) is an illegal destination for a static route. See section 3.5 for information about the router option.

The code for this option is 33. The minimum length of this option is 8, and the length MUST be a multiple of 8.

```
 Code   Len         Destination 1           Router 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  33 |  n  |  d1 |  d2 |  d3 |  d4 |  r1 |  r2 |  r3 |  r4 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
        Destination 2           Router 2
+-----+-----+-----+-----+-----+-----+-----+-----+---
|  d1 |  d2 |  d3 |  d4 |  r1 |  r2 |  r3 |  r4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```

6. Link Layer Parameters per Interface

This section lists the options that affect the operation of the data link layer on a per-interface basis.

6.1. Trailer Encapsulation Option

This option specifies whether or not the client should negotiate the use of trailers (RFC 893 [14]) when using the ARP protocol. A value of 0 indicates that the client should not attempt to use trailers. A value of 1 means that the client should attempt to use trailers.

The code for this option is 34, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  34 |  1  | 0/1 |
+-----+-----+-----+
```

6.2. ARP Cache Timeout Option

This option specifies the timeout in seconds for ARP cache entries. The time is specified as a 32-bit unsigned integer.

The code for this option is 35, and its length is 4.

```
 Code   Len           Time
+-----+-----+-----+-----+-----+-----+
|  35 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

6.3. Ethernet Encapsulation Option

This option specifies whether or not the client should use Ethernet Version 2 (RFC 894 [15]) or IEEE 802.3 (RFC 1042 [16]) encapsulation if the interface is an Ethernet. A value of 0 indicates that the client should use RFC 894 encapsulation. A value of 1 means that the client should use RFC 1042 encapsulation.

The code for this option is 36, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  36 |  1  | 0/1 |
+-----+-----+-----+
```

7. TCP Parameters

This section lists the options that affect the operation of the TCP layer on a per-interface basis.

7.1. TCP Default TTL Option

This option specifies the default TTL that the client should use when sending TCP segments. The value is represented as an 8-bit unsigned integer. The minimum value is 1.

The code for this option is 37, and its length is 1.

```
 Code   Len   TTL
+-----+-----+-----+
|  37 |  1  |  n  |
+-----+-----+-----+
```

7.2. TCP Keepalive Interval Option

This option specifies the interval (in seconds) that the client TCP should wait before sending a keepalive message on a TCP connection. The time is specified as a 32-bit unsigned integer. A value of zero indicates that the client should not generate keepalive messages on connections unless specifically requested by an application.

The code for this option is 38, and its length is 4.

```
 Code   Len           Time
+-----+-----+-----+-----+-----+-----+
|  38 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

7.3. TCP Keepalive Garbage Option

This option specifies the whether or not the client should send TCP keepalive messages with a octet of garbage for compatibility with older implementations. A value of 0 indicates that a garbage octet should not be sent. A value of 1 indicates that a garbage octet should be sent.

The code for this option is 39, and its length is 1.

```
 Code   Len  Value
+-----+-----+-----+
|  39 |  1  | 0/1 |
+-----+-----+-----+
```

8. Application and Service Parameters

This section details some miscellaneous options used to configure miscellaneous applications and services.

8.1. Network Information Service Domain Option

This option specifies the name of the client's NIS [17] domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set.

The code for this option is 40. Its minimum length is 1.

```
 Code   Len      NIS Domain Name
+-----+-----+-----+-----+-----+-----+---
|  40 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

8.2. Network Information Servers Option

This option specifies a list of IP addresses indicating NIS servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 41. Its minimum length is 4, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  41 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.3. Network Time Protocol Servers Option

This option specifies a list of IP addresses indicating NTP [18] servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 42. Its minimum length is 4, and the length MUST be a multiple of 4.

```
     Code   Len         Address 1               Address 2
    +-----+-----+-----+-----+-----+-----+-----+-----+--
    |  42 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
    +-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.4. Vendor Specific Information

This option is used by clients and servers to exchange vendor-specific information. The information is an opaque object of n octets, presumably interpreted by vendor-specific code on the clients and servers. The definition of this information is vendor specific. The vendor is indicated in the vendor class identifier option. Servers not equipped to interpret the vendor-specific information sent by a client MUST ignore it (although it may be reported). Clients which do not receive desired vendor-specific information SHOULD make an attempt to operate without it, although they may do so (and announce they are doing so) in a degraded mode.

If a vendor potentially encodes more than one item of information in this option, then the vendor SHOULD encode the option using "Encapsulated vendor-specific options" as described below:

The Encapsulated vendor-specific options field SHOULD be encoded as a sequence of code/length/value fields of identical syntax to the DHCP options field with the following exceptions:

1) There SHOULD NOT be a "magic cookie" field in the encapsulated vendor-specific extensions field.

2) Codes other than 0 or 255 MAY be redefined by the vendor within the encapsulated vendor-specific extensions field, but SHOULD conform to the tag-length-value syntax defined in section 2.

3) Code 255 (END), if present, signifies the end of the encapsulated vendor extensions, not the end of the vendor extensions field. If no code 255 is present, then the end of the enclosing vendor-specific information field is taken as the end of the encapsulated vendor-specific extensions field.

The code for this option is 43 and its minimum length is 1.

```
    Code   Len   Vendor-specific information
    +-----+-----+-----+-----+---
    |  43 |  n  |  i1 |  i2 | ...
    +-----+-----+-----+-----+---
```

When encapsulated vendor-specific extensions are used, the information bytes 1-n have the following format:

```
 Code   Len   Data item        Code   Len   Data item       Code
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  T1 |  n  |  d1 |  d2 | ... |  T2 |  n  |  D1 |  D2 | ... | ... |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

8.5. NetBIOS over TCP/IP Name Server Option

The NetBIOS name server (NBNS) option specifies a list of RFC 1001/1002 [19] [20] NBNS name servers listed in order of preference.

The code for this option is 44. The minimum length of the option is 4 octets, and the length must always be a multiple of 4.

```
 Code   Len           Address 1              Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
|  44 |  n  |  a1 |  a2 |  a3 |  a4 |  b1 |  b2 |  b3 |  b4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
```

8.6. NetBIOS over TCP/IP Datagram Distribution Server Option

The NetBIOS datagram distribution server (NBDD) option specifies a list of RFC 1001/1002 NBDD servers listed in order of preference. The code for this option is 45. The minimum length of the option is 4 octets, and the length must always be a multiple of 4.

```
 Code   Len           Address 1              Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
|  45 |  n  |  a1 |  a2 |  a3 |  a4 |  b1 |  b2 |  b3 |  b4 | ...
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
```

8.7. NetBIOS over TCP/IP Node Type Option

The NetBIOS node type option allows NetBIOS over TCP/IP clients which are configurable to be configured as described in RFC 1001/1002. The value is specified as a single octet which identifies the client type as follows:

Value	Node Type
0x1	B-node
0x2	P-node
0x4	M-node
0x8	H-node

In the above chart, the notation '0x' indicates a number in base-16 (hexadecimal).

The code for this option is 46. The length of this option is always 1.

```
 Code   Len  Node Type
+-----+-----+-----------+
|  46 |  1  | see above |
+-----+-----+-----------+
```

8.8. NetBIOS over TCP/IP Scope Option

The NetBIOS scope option specifies the NetBIOS over TCP/IP scope parameter for the client as specified in RFC 1001/1002. See [19], [20], and [8] for character-set restrictions.

The code for this option is 47. The minimum length of this option is 1.

```
 Code   Len       NetBIOS Scope
+-----+-----+-----+-----+-----+-----+----
|  47 |  n  |  s1 |  s2 |  s3 |  s4 | ...
+-----+-----+-----+-----+-----+-----+----
```

8.9. X Window System Font Server Option

This option specifies a list of X Window System [21] Font servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 48. The minimum length of this option is 4 octets, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+---
|  48 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |   ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```

8.10. X Window System Display Manager Option

This option specifies a list of IP addresses of systems that are running the X Window System Display Manager and are available to the client.

Addresses SHOULD be listed in order of preference.

The code for the this option is 49. The minimum length of this option is 4, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2

+-----+-----+-----+-----+-----+-----+-----+-----+---
|  49 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |   ...
+-----+-----+-----+-----+-----+-----+-----+-----+---
```

8.11. Network Information Service+ Domain Option

This option specifies the name of the client's NIS+ [17] domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set.

The code for this option is 64. Its minimum length is 1.

```
 Code   Len      NIS Client Domain Name
+-----+-----+-----+-----+-----+-----+---
|  64 |  n  |  n1 |  n2 |  n3 |  n4 | ...
+-----+-----+-----+-----+-----+-----+---
```

8.12. Network Information Service+ Servers Option

This option specifies a list of IP addresses indicating NIS+ servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is 65. Its minimum length is 4, and the length MUST be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
|  65 |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.13. Mobile IP Home Agent option

This option specifies a list of IP addresses indicating mobile IP home agents available to the client. Agents SHOULD be listed in order of preference.

The code for this option is 68. Its minimum length is 0 (indicating no home agents are available) and the length MUST be a multiple of 4. It is expected that the usual length will be four octets, containing a single home agent's address.

```
 Code Len    Home Agent Addresses (zero or more)
+-----+-----+-----+-----+-----+-----+--
| 68  |  n  | a1  | a2  | a3  | a4  | ...
+-----+-----+-----+-----+-----+-----+--
```

8.14. Simple Mail Transport Protocol (SMTP) Server Option

The SMTP server option specifies a list of SMTP servers available to the client. Servers SHOULD be listed in order of preference.

The code for the SMTP server option is 69. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
| 69  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.15. Post Office Protocol (POP3) Server Option

The POP3 server option specifies a list of POP3 available to the client. Servers SHOULD be listed in order of preference.

The code for the POP3 server option is 70. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
| 70  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.16. Network News Transport Protocol (NNTP) Server Option

The NNTP server option specifies a list of NNTP available to the client. Servers SHOULD be listed in order of preference.

The code for the NNTP server option is 71. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
| 71  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.17. Default World Wide Web (WWW) Server Option

The WWW server option specifies a list of WWW available to the client. Servers SHOULD be listed in order of preference.

The code for the WWW server option is 72. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
| 72  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.18. Default Finger Server Option

The Finger server option specifies a list of Finger available to the client. Servers SHOULD be listed in order of preference.

The code for the Finger server option is 73. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
| 73  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.19. Default Internet Relay Chat (IRC) Server Option

The IRC server option specifies a list of IRC available to the client. Servers SHOULD be listed in order of preference.

The code for the IRC server option is 74. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
| 74  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.20. StreetTalk Server Option

The StreetTalk server option specifies a list of StreetTalk servers available to the client. Servers SHOULD be listed in order of preference.

The code for the StreetTalk server option is 75. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
| 75  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

8.21. StreetTalk Directory Assistance (STDA) Server Option

The StreetTalk Directory Assistance (STDA) server option specifies a list of STDA servers available to the client. Servers SHOULD be listed in order of preference.

The code for the StreetTalk Directory Assistance server option is 76. The minimum length for this option is 4 octets, and the length MUST always be a multiple of 4.

```
 Code   Len         Address 1               Address 2
+-----+-----+-----+-----+-----+-----+-----+-----+--
| 76  |  n  |  a1 |  a2 |  a3 |  a4 |  a1 |  a2 |  ...
+-----+-----+-----+-----+-----+-----+-----+-----+--
```

9. DHCP Extensions

This section details the options that are specific to DHCP.

9.1. Requested IP Address

This option is used in a client request (DHCPDISCOVER) to allow the client to request that a particular IP address be assigned.

The code for this option is 50, and its length is 4.

```
 Code   Len          Address
+-----+-----+-----+-----+-----+-----+
|  50 |  4  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

9.2. IP Address Lease Time

This option is used in a client request (DHCPDISCOVER or DHCPREQUEST) to allow the client to request a lease time for the IP address. In a server reply (DHCPOFFER), a DHCP server uses this option to specify the lease time it is willing to offer.

The time is in units of seconds, and is specified as a 32-bit unsigned integer.

The code for this option is 51, and its length is 4.

```
 Code   Len         Lease Time
+-----+-----+-----+-----+-----+-----+
|  51 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

9.3. Option Overload

This option is used to indicate that the DHCP 'sname' or 'file' fields are being overloaded by using them to carry DHCP options. A DHCP server inserts this option if the returned parameters will exceed the usual space allotted for options.

If this option is present, the client interprets the specified additional fields after it concludes interpretation of the standard option fields.

The code for this option is 52, and its length is 1. Legal values for this option are:

Value	Meaning
1	the 'file' field is used to hold options
2	the 'sname' field is used to hold options
3	both fields are used to hold options

```
 Code   Len  Value
+-----+-----+-----+
|  52 |  1  |1/2/3|
+-----+-----+-----+
```

9.4 TFTP server name

This option is used to identify a TFTP server when the 'sname' field in the DHCP header has been used for DHCP options.

The code for this option is 66, and its minimum length is 1.

```
    Code  Len   TFTP server
   +-----+-----+-----+-----+-----+---
   | 66  |  n  |  c1 |  c2 |  c3 | ...
   +-----+-----+-----+-----+-----+---
```

9.5 Bootfile name

This option is used to identify a bootfile when the 'file' field in the DHCP header has been used for DHCP options.

The code for this option is 67, and its minimum length is 1.

```
 Code  Len   Bootfile name
+-----+-----+-----+-----+-----+---
| 67  |  n  |  c1 |  c2 |  c3 | ...
+-----+-----+-----+-----+-----+---
```

9.6. DHCP Message Type

This option is used to convey the type of the DHCP message. The code for this option is 53, and its length is 1. Legal values for this option are:

Value	Message Type
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNAK
7	DHCPRELEASE
8	DHCPINFORM

```
 Code   Len  Type
+-----+-----+-----+
|  53 |  1  | 1-9 |
+-----+-----+-----+
```

9.7. Server Identifier

This option is used in DHCPOFFER and DHCPREQUEST messages, and may optionally be included in the DHCPACK and DHCPNAK messages. DHCP servers include this option in the DHCPOFFER in order to allow the client to distinguish between lease offers. DHCP clients use the contents of the 'server identifier' field as the destination address for any DHCP messages unicast to the DHCP server. DHCP clients also indicate which of several lease offers is being accepted by including this option in a DHCPREQUEST message.

The identifier is the IP address of the selected server.

The code for this option is 54, and its length is 4.

```
 Code   Len          Address
+-----+-----+-----+-----+-----+-----+
|  54 |  4  |  a1 |  a2 |  a3 |  a4 |
+-----+-----+-----+-----+-----+-----+
```

9.8. Parameter Request List

This option is used by a DHCP client to request values for specified configuration parameters. The list of requested parameters is specified as n octets, where each octet is a valid DHCP option code as defined in this document.

The client MAY list the options in order of preference. The DHCP server is not required to return the options in the requested order, but MUST try to insert the requested options in the order requested by the client.

The code for this option is 55. Its minimum length is 1.

```
 Code   Len   Option Codes
+-----+-----+-----+-----+---
|  55 |  n  |  c1 |  c2 | ...
+-----+-----+-----+-----+---
```

9.9. Message

This option is used by a DHCP server to provide an error message to a DHCP client in a DHCPNAK message in the event of a failure. A client may use this option in a DHCPDECLINE message to indicate the why the client declined the offered parameters. The message consists of n octets of NVT ASCII text, which the client may display on an available output device.

The code for this option is 56 and its minimum length is 1.

```
 Code   Len     Text
+-----+-----+-----+-----+---
|  56 |  n  |  c1 |  c2 | ...
+-----+-----+-----+-----+---
```

9.10. Maximum DHCP Message Size

This option specifies the maximum length DHCP message that it is willing to accept. The length is specified as an unsigned 16-bit integer. A client may use the maximum DHCP message size option in DHCPDISCOVER or DHCPREQUEST messages, but should not use the option in DHCPDECLINE messages.

The code for this option is 57, and its length is 2. The minimum legal value is 576 octets.

```
 Code   Len     Length
+-----+-----+-----+-----+
|  57 |  2  |  l1 |  l2 |
+-----+-----+-----+-----+
```

9.11. Renewal (T1) Time Value

This option specifies the time interval from address assignment until the client transitions to the RENEWING state.

The value is in units of seconds, and is specified as a 32-bit unsigned integer.

The code for this option is 58, and its length is 4.

```
 Code   Len         T1 Interval
+-----+-----+-----+-----+-----+-----+
|  58 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

9.12. Rebinding (T2) Time Value

This option specifies the time interval from address assignment until the client transitions to the REBINDING state.

The value is in units of seconds, and is specified as a 32-bit unsigned integer.

The code for this option is 59, and its length is 4.

```
 Code   Len         T2 Interval
+-----+-----+-----+-----+-----+-----+
|  59 |  4  |  t1 |  t2 |  t3 |  t4 |
+-----+-----+-----+-----+-----+-----+
```

9.13. Vendor class identifier

This option is used by DHCP clients to optionally identify the vendor type and configuration of a DHCP client. The information is a string of n octets, interpreted by servers. Vendors may choose to define specific vendor class identifiers to convey particular configuration or other identification information about a client. For example, the identifier may encode the client's hardware configuration. Servers not equipped to interpret the class-specific information sent by a client MUST ignore it (although it may be reported). Servers that respond SHOULD only use option 43 to return the vendor-specific information to the client.

The code for this option is 60, and its minimum length is 1.

```
Code   Len   Vendor class Identifier
+-----+-----+-----+-----+---
|  60 |  n  |  i1 |  i2 | ...
+-----+-----+-----+-----+---
```

9.14. Client-identifier

This option is used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain.

Identifiers SHOULD be treated as opaque objects by DHCP servers.

The client identifier MAY consist of type-value pairs similar to the 'htype'/'chaddr' fields defined in [3]. For instance, it MAY consist of a hardware type and hardware address. In this case the type field SHOULD be one of the ARP hardware types defined in STD2 [22]. A hardware type of 0 (zero) should be used when the value field contains an identifier other than a hardware address (e.g. a fully qualified domain name).

For correct identification of clients, each client's client-identifier MUST be unique among the client-identifiers used on the subnet to which the client is attached. Vendors and system administrators are responsible for choosing client-identifiers that meet this requirement for uniqueness.

The code for this option is 61, and its minimum length is 2.

```
Code   Len   Type  Client-Identifier
+-----+-----+-----+-----+-----+---
|  61 |  n  |  t1 |  i1 |  i2 | ...
+-----+-----+-----+-----+-----+---
```

10. Defining new extensions

The author of a new DHCP option will follow these steps to obtain acceptance of the option as a part of the DHCP Internet Standard:

1. The author devises the new option.
2. The author requests a number for the new option from IANA by contacting:
   Internet Assigned Numbers Authority (IANA)
   USC/Information Sciences Institute
   4676 Admiralty Way
   Marina del Rey, California 90292-6695

   or by email as: iana@iana.org

3. The author documents the new option, using the newly obtained option number, as an Internet Draft.
4. The author submits the Internet Draft for review through the IETF standards process as defined in "Internet Official Protocol Standards" (STD 1). The new option will be submitted for eventual acceptance as an Internet Standard.
5. The new option progresses through the IETF standards process; the new option will be reviewed by the Dynamic Host Configuration Working Group (if that group still exists), or as an Internet Draft not submitted by an IETF working group.
6. If the new option fails to gain acceptance as an Internet Standard, the assigned option number will be returned to IANA for reassignment.

   This procedure for defining new extensions will ensure that:

   * allocation of new option numbers is coordinated from a single authority,
   * new options are reviewed for technical correctness and appropriateness, and
   * documentation for new options is complete and published.

11. Acknowledgements

The author thanks the many (and too numerous to mention!) members of the DHC WG for their tireless and ongoing efforts in the development of DHCP and this document.

The efforts of J Allard, Mike Carney, Dave Lapp, Fred Lien and John Mendonca in organizing DHCP interoperability testing sessions are gratefully acknowledged.

The development of this document was supported in part by grants from the Corporation for National Research Initiatives (CNRI), Bucknell University and Sun Microsystems.

12. References

[1] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, Bucknell University, March 1997.

[2] Reynolds, J., "BOOTP Vendor Information Extensions", RFC 1497, USC/Information Sciences Institute, August 1993.

[3] Croft, W., and J. Gilmore, "Bootstrap Protocol", RFC 951, Stanford University and Sun Microsystems, September 1985.

[4] Braden, R., Editor, "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, USC/Information Sciences Institute, October 1989.

[5] Mogul, J., and J. Postel, "Internet Standard Subnetting Procedure", STD 5, RFC 950, USC/Information Sciences Institute, August 1985.

[6] Postel, J., and K. Harrenstien, "Time Protocol", STD 26, RFC 868, USC/Information Sciences Institute, SRI, May 1983.

[7] Postel, J., "Name Server", IEN 116, USC/Information Sciences Institute, August 1979.

[8] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, USC/Information Sciences Institute, November 1987.

[9] Postel, J., "Quote of the Day Protocol", STD 23, RFC 865, USC/Information Sciences Institute, May 1983.

[10] McLaughlin, L., "Line Printer Daemon Protocol", RFC 1179, The Wollongong Group, August 1990.

[11] Accetta, M., "Resource Location Protocol", RFC 887, CMU, December 1983.

[12] Mogul, J. and S. Deering, "Path MTU Discovery", RFC 1191, DECWRL, Stanford University, November 1990.

[13] Deering, S., "ICMP Router Discovery Messages", RFC 1256, Xerox PARC, September 1991.

[14] Leffler, S. and M. Karels, "Trailer Encapsulations", RFC 893, U. C. Berkeley, April 1984.

[15] Hornig, C., "Standard for the Transmission of IP Datagrams over Ethernet Networks", RFC 894, Symbolics, April 1984.

[16] Postel, J. and J. Reynolds, "Standard for the Transmission of IP Datagrams Over IEEE 802 Networks", RFC 1042, USC/Information Sciences Institute, February 1988.

[17] Sun Microsystems, "System and Network Administration", March 1990.

[18] Mills, D., "Internet Time Synchronization: The Network Time Protocol", RFC 1305, UDEL, March 1992.

[19] NetBIOS Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP transport: Concepts and Methods", STD 19, RFC 1001, March 1987.

[20] NetBIOS Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP transport: Detailed Specifications", STD 19, RFC 1002, March 1987.

[21] Scheifler, R., "FYI On the X Window System", FYI 6, RFC 1198, MIT Laboratory for Computer Science, January 1991.

[22] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, USC/Information Sciences Institute, July 1992.

13. Security Considerations

Security issues are not discussed in this memo.

14. Authors' Addresses

Steve Alexander
Silicon Graphics, Inc.
2011 N. Shoreline Boulevard
Mailstop 510
Mountain View, CA 94043-1389

Phone: (415) 933-6172
EMail: sca@engr.sgi.com

Ralph Droms
Bucknell University
Lewisburg, PA 17837

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