Network Working Group   R. Hibbs
INTERNET-DRAFT   Richard Barr Hibbs, P.E.
Category: Informational   R. Stevens
Expires: December 17, 2006   (no affiliation)
                                                    June 15, 2006

Implementation Issues with RFC 2131, "Dynamic Host Configuration Protocol (DHCPv4)"

<draft-ietf-dhc-implementation-02.txt>
Saved: Tuesday, June 15, 2006, 13:27:17

Intellectual Property Rights

Status of this Memo

Copyright Notice

Abstract

This memo identifies implementation issues with RFC 2131, "Dynamic Host Configuration Protocol," reported by a number of implementers, assesses the severity of the problem, then proposes changes to RFC 2131 intended to overcome the issues. This is intended for use as

the basis for discussion of RFC 2131 before it is proposed for Internet Standard status.

Table of Contents

1 Introduction

This memo was produced by the DHC Working Group and attempts to identify all known implementation issues with RFC 2131 as a basis for discussion of RFC 2131 before it is published as an Internet Standard.

This memo grew from a discussion item during the DHC Working Group meeting at IETF-55 in Atlanta during November 2002.

The editors have solicited input through a general call for participation and by direct request to all implementers that they could identify.

There are four possible outcomes of this work:

1. The RFC Editor could publish the agreed clarifications as a note ("Errata") linked to RFC 2131 in the RFC Editor's document database. This almost insures that only the most conscientious reviewer would ever read and apply the changes.

2. The proposed changes could be reworded and reformatted into a new standards-track document, "RFC 2131 Errata." This has the advantage of not disturbing RFC 2131 as it advances to Internet Standard, but leaves the reader with multiple documents to read for a full understanding of DHCP.

3. Proposed changes could be applied to RFC 2131 without assigning a new document number, in effect becoming "RFC 2131-bis." This course may not be open as the changes would still require IESG approval, and it is unlikely that any changes other than editorial or clarification would be permitted.

4. A new standards-track document would be created, obsoleting RFC 2131 (and possibly other documents as well.) This would probably not require any more effort than outcome (2), but conceivably take years to advance the document to full Standard.

The editors have not specifically addressed RFC 2132, although we believe that it ought to be updated in conjunction with any updates to RFC 2131. We propose that an update of the DHCP Options Document should be separately considered in a second memo.

2 Terminology

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3 Applicability

The intent of the work item that resulted in this memo was to identify and clarify issues with RFC 2131 that made implementation difficult, behavior ambiguous, or conflicted with other RFCs. The authors imagined that RFC 2132 could also be updated as a result, without expecting that additional RFCs might be affected.

As our investigation proceeded, it became evident that clarifications might very well extend to other RFCs. The exact scope of this effort will require additional discussion by the DHC Working Group.

4 Issues with RFC 2131

This list may not include every implementation issue for RFC 2131 as it is based on reported problems and those known to the editors.

4.1 Outdated RFC Boilerplate

RECOMMENDATIONS:

1. "Status of This Memo" should be replaced with standardized language for RFCs as described in "Guidelines to Authors of Internet-Drafts," dated March 25, 2005.

2. Section 1.4, "Requirements," should be replaced with standardized language referring to RFC 2119 regarding the definition and interpretation of specific key words.

3. References should be separated into normative and non-normative sections.

4.2 Organization and Typography

4.2.1 Outdated References

RECOMMENDATIONS:

- References to the "Assigned Numbers" RFC [STD 2, RFC 1700] should be changed to the "Assigned Numbers" database maintained by IANA. References are found in Tables 3 and 5, and in the "References" section.

- References to the "Interaction between DHCP and BOOTP" RFC 1534 should be integrated with the text of the memo, and RFC 1534 deprecated.

4.2.2 Typographical Errors

RECOMMENDATIONS:

- Page 23, third paragraph of section 4.1 -- "received" should be "received."

- Page 23, sixth paragraph of section 4.1 -- refers to RFC 1533, not RFC 2132.

- Page 15, Figure 3. Table is misformatted.

- Page 18, Figure 4. Table is misformatted.

- Page 25, ninth paragraph of section 4.1 -- "uicast" should be "unicast."

- Page 38, first paragraph after Table 5. Orphaned sentence: "The DHCPREQUEST message contains the same 'xid' as the DHCPOFFER message." No, it does not. Not only that, but this sentence makes no sense in its current location. It should be removed.

- Page 39, Last paragraph of 4.4.3 should be moved up as the last paragraph of 4.4.2. When the text for DHCPINFORM was added, the text describing what a client should do if no DHCPACK is received was mistakenly pushed below it.

- Apostrophes (') are used as single quotation marks, but outside of an enclosing quotation (") throughout the document.

- Page 30, section 4.3.1, second from last bullet: "...client s vendor class identifier and client's classes identified in the server". This text makes no sense and should be deleted.

- Inconsistent style regarding placement of periods (.), commas (,) and semi-colons (;) with respect to quotation marks throughout the document.

- Quotation marks (single and double) are overused through the document.

4.2.3 Omissions

In several places there is missing or incomplete information, including:

- Table 3, pages 27 and 28. The "options" entry for "DHCPNAK" in the "fields" portion of the table is missing. All entries in this line should refer to the subsequent "options" table. Suggested replacements for Table 3 are shown in sections 5.2.3 and 5.2.4.

- Page 33, Table 4, and Page 35, Figure 5, do not include the "DHCPINFORM" message. Suggested replacement for Table 4 is shown in section 5.2.5.

- Table 5, pages 37 and 38. The "options" entry for "DHCPDECLINE, DHCPRELEASE" is missing. Suggested replacement for Table 5 is shown in sections 5.2.6 and 5.2.7.

4.2.4 Tables

RECOMMENDATIONS:

- Table 3 (pages 28 and 29) and Table 5 (pages 37 and 38) should be separated into two tables each for readability. Suggested

replacements for Table 3 are shown in sections 5.2.3 and 5.2.4, and for Table 5 are show in sections 5.2.6 and 5.2.7.

- Table 4 should be reorganized to show all messages (except DHCPRELEASE) that are sent from each client state. Suggested replacement for Table 4 is shown in section 5.2.5.

- O The "Host Configuration Parameters" table now in an appendix should be omitted and described in a revised "DHCP Options" document (RFC 2132). Suggested replacement for that table is shown in sections 5.2.8-5.2.10.

4.2.5 Inconsistencies

RECOMMENDATIONS:

- O Page 1, Abstract, "TCPIP" should be "TCP/IP" as it is in the rest of the document.

- O Page 10, Table 3, description of 'htype' and 'hlen' fields does not capitalize "Ethernet."

- O Lack of consistency when describing "IP broadcast." Sometimes it is "0xffffffff IP broadcast," elsewhere "limited broadcast," or "broadcast." Suggest using the "255.255.255.255 IP broadcast address" form, as that is the most specific. References include:

  a. Page 19, third paragraph of section 3.2, List item #2.

  b. Page 23, fifth paragraph of section 4.1 (twice).

  c. Page 25, thirteenth paragraph of section 4.1 (twice).

  d. Page 32, section 4.3.2, third bullet item.

  e. Page 32, section 4.3.2, fifth bullet item.

  f. Page 36, second paragraph of section 4.4.1.

  g. Page 39, last paragraph of section 4.4.1.

  h. Page 39, second paragraph of section 4.4.3.

4. Lack of consistency when referring to the BROADCAST (B) flag: it is also referred to as the "broadcast bit."

5. Table 3, "Fields and options used by DHCP servers," is problematic. It indicates that we MUST fill in both the "Server Identifier" (and siaddr) in our DHCPACK (and DHCPNAK) response. That is a change from RFC 1541 (which specifies a "MAY" and which is consistent with RFC 2132 section 9.7 and identical to RFC 1533 section 9.5 wording).

4.3 Policy Issues

There has in general been a certain amount of overlap in DHCP

between protocol and policy. The matters include lease times, whether servers are willing to extend leases, timeouts, and re-transmission.

We SHOULD clarify what is dictated by the protocol and what is a policy decision at a given site.

The DHC Working Group philosophy ought to be to constrain client behavior more closely than server behavior. DHCP interactions are initiated (and continued) by clients: clients outnumber servers by many tens of thousands to one; client implementers cannot be quite certain of all the environments in which their client may ultimately appear, whereas server implementations may be designed for very specific environments. Policy is likely to be a matter of centralized control, whereas clients are not likely to enjoy a sufficient status to impose policy on servers.

The previous paragraph implies that the WORKING GROUP should tighten the protocol with respect to such issues as retries and backoffs, whereas servers should not be constrained on issues such as how to uniquely identify clients, whether to offer or extend leases etc.

4.4 The Client Hardware Address, "chaddr"

The value of "chaddr" MUST NOT change from DHCPDISCOVER to DHCPREQUEST, although the wording in Table 3 makes this point unclear.

Further, the length of "chaddr" SHOULD be exactly specified by "hlen," which SHOULD match the address length for "htype."

RECOMMENDATIONS:

O Update Table 3.

4.5 The DHCP Client Identifier

4.5.1 Uniqueness

DHCP servers must uniquely identify DHCP clients requesting services in order to configure the client correctly. DHCP does not require global uniqueness for client identifiers, only uniqueness within the scope of [sub-] networks reachable by DHCP packets in any installation. This is sometimes called "subnet uniqueness."

RFC 2131 provides two specific methods for identifying a client: (1) the client identifier (DHCP Option 61) [RFC2132], and (2) the "chaddr" field of the BOOTPREQUEST packet.

Confusion arises from the language of RFC 2131 Section 4.2. A DHCP client "...MAY choose to explicitly provide the identifier through the 'client identifier' option. If the client supplies a 'client identifier,' the client MUST use that same 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."

The text of Section 4.2 goes on to state that subnet uniqueness is a requirement for an identifier, but points out that "chaddr" may not satisfy that requirement. Two alternatives for a unique identifier were given: an unspecified manufacturer's serial number or a DNS name.

RFC 2132 adds to the confusion by stating that the client identifier "...is expected to be unique for all clients in an administrative domain" without specifying what an "administrative domain" is.

RFC 2132 continues by suggesting use of "...type-value pairs similar to the 'htype'/'chaddr' fields defined in" [RFC951], and that 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)."

This suggestion of using type-value pairs has been widely adopted by DHCP client implementers, but the suggestion fails to heed the warning about uniqueness issues with "chaddr."

RECOMMENDATIONS:

1. RFC 2131 SHOULD have made required the "client identifier" either to be globally unique or, to be unique within an "administrative domain," and, in the latter case, defined "administrative domain."

2. RFC 2131 SHOULD NOT have suggested the use of DNS names for the "client identifier" without also suggesting some mechanism for maintaining a consistent name-to-address mapping.

3. RFC 2132 SHOULD NOT have suggested using the "htype" and "chaddr" fields as a type-value pair because of the warning in RFC 2131 Section 4.2 about potential problems using "chaddr" for the purpose.

4. RFC 2132 SHOULD NOT have used the word "SHOULD" when suggesting the use of type-value pairs for "client identifier" with a type of 0 (zero) when the value is anything other than a hardware address.

4.5.2 Prohibition in DHCPOFFER and DHCPACK

Table 3, in the "options" section, specifies that the server MUST NOT send the "client identifier" in the DHCPOFFER or DHCPACK messages, but MAY send it in a DHCPNAK message. There is no good reason why DHCPNAK should be treated differently, and there is considerable utility in returning the client identifier, as it allows clients further corroboration, beyond that implied by matching "xid s" (see 2.23), that they are the intended recipient.

RECOMMENDATION:

Change the text in Table 3, for all three message-types, to read, "MAY -- if included, MUST BE the client identifier sent by the client." Suggested replacement for Table 3 are shown in sections

5.2.3 and 5.2.4.

4.6 Duplicate Address Detection

RFC 2131 Page 7, section 1.6, second set of bullet items, first bullet says that DHCP must: "Guarantee that any specific network address will not be in use by more than one client at a time," but the protocol as later described does not fulfill this requirement.

Two mechanisms are presented: an ARP request generated by the client, and an ICMP ECHO request generated by the server:

O Page 12, second paragraph of section 2.2, last sentence.

O Page 15, list item 2, section 3.1.

O Page 38, first paragraph after Table 5, section 4.4.1.

4.6.1 Client-side ARP

To meet the requirement of RFC 2131 page 7, a DHCP client MUST send an ARP request for the IP address contained in a DHCPACK before using it. This is presently a SHOULD:

> Page 12, second paragraph of section 2.2: "... and the client SHOULD probe the newly received address, e.g., with ARP."

There has been confusion on this topic because many clients are sending an ARP reply (after the DHCPACK). This often has nothing to do with DHCP, and is triggered in many systems whenever an interface IP address changes. (Without access to kernel code, there is nothing to be done about it.)

RECOMMENDATION:

The Working Group should consider whether to make client ARP a MUST.

4.6.2 Server side PING

ICMP is inherently unreliable. Furthermore since success is "no response" it is an imprecise matter to decide how long to wait before one is certain that no response will ever occur. A possible suggestion is a back off and retry for ping.

In cable modem environments, PING is not helpful because it is the cable modem termination system (CMTS) that replies from its cache: a cache which may not be perfectly reliable and which in many cases has been constructed by listening to the DHCP traffic in the first place!

It is known that some network administrators try to block propagation of ICMP ECHO messages through internal routers, which removes one of the two address conflict detection mechanisms.

RECOMMENDATION:

Use of ICMP on the server should be a "MAY", not a "SHOULD".

4.6.3 Other Mechanisms

Both the ICMP ECHO (Ping) and Address Resolution Protocol (ARP) mechanisms are very lightweight by design, depending on clients with conflicting addresses to "defend" their address by responding to queries to show that an address is in use. Is there a better alternative to ICMP ECHO and ARP that is backward compatible with these two protocols?

4.7 DHCP Relay Agents

4.7.1 Relay Agent Source Addresses

There should be some text that specifies what the relay agent should use for the IP source address of relayed packets. Because relay agents change the payload ("giaddr" and relay agent option 82), their operation does not amount to IP forwarding. The IP source address they use should be their own. [Aside: for security purposes it might have been better than they retain the source IP address of the original packet, but it is too late to change all that.]

RECOMMENDATION:

4.7.2 Relay Agent Port Usage

Relay agents should use port 67 as the source port number. Relay agents always listen on port 67, but port 68 has sometimes been used as the source port number probably because it was copied from the source port of the incoming packet.

Cable modem vendors would like to install filters blocking outgoing packets with source port 67.

RECOMMENDATIONS:

O Relay agents MUST use 67 as their source port number.

O Relay agents MUST NOT forward packets with non-zero giaddr unless the source port number on the packet is 67.

4.8 Host Name, Domain Name, and FQDNs

A fully qualified domain name (FQDN) consists of two conceptual parts: the host name portion and the domain name portion. Host names consist of one or more non-null parts separated by the ISO period (.) character ("separator") while domain names consist of two or more non-null parts delimited by the separator, one of which must be a valid top-level domain (TLD) name. DHCP options exist for hostname (option 12) and domain name (option 15), and are proposed for FQDN (draft-ietf-dhc-fqdn-option-05.txt) but the FQDN option is not required to be a concatenation of hostname and domain name.

Should RFC 2131 explicitly state that the client FQDN MUST be the host name (option 12) concatenated with the domain name (option 15)?

4.9 Overloading of DHCPREQUEST

The client sends a DHCPREQUEST message from several different states: INIT, INIT-REBOOT, REBINDING, and RENEWING. Differentiation among the states is done according to the context of other message fields and option values. At this point, there probably can be no change in this usage, but the content of other message fields and option values should be carefully reviewed to ensure consistency.

4.10 DHCPINFORM

The intent of DHCPINFORM messages is to allow clients to query servers for configuration information WHETHER OR NOT their IP address has been assigned by DHCP.

Section 3.4 Para 2 Page 21 states: "The server SHOULD check the network address in a DHCPINFORM message for consistency." What the server should be checking is a mystery. Possibly the intent was that servers should verify that the source IP address in the packet is identical to the "ciaddr." Since the server should reply to "ciaddr," this affords some measure of security, preventing third parties from discovering configuration information pertaining to other clients. Whether that is desirable, or whether instead DHCPINFORM should be available to third parties, such as proxies, has never been resolved.

Section 4.4.4 Para 1, "Use of broadcast and unicast," hints that clients may be able to broadcast DHCPINFORM messages to servers: "The DHCP client broadcasts DHCPDISCOVER, DHCPREQUEST and DHCPINFORM messages, unless the client knows the address of a DHCP server."

This text suggests that a DHCP client may choose to broadcast a DHCPINFORM request for whatever reason, and points out the need for clarification of all text concerning multiple server responses and consistency of returned options.

RECOMMENDATIONS:

- DHCPINFORM messages should be included in Table 4 to summarize the fields and options usage with this message type. Suggested replacement for Table 4 is shown in section 5.2.5.

- The Working Group should consider the ramifications of permitting third party DHCPINFORMs, that is, DHCPINFORM messages NOT sent by the DHCP client, but by other processes having access to the ports.

- Section 4.3.1, "DHCPDISCOVER Message," and section 4.3.2, "DHCPREQUEST Message" briefly mention consistency and uniform responses from multiple servers: this text SHOULD be clarified to state what consistency is expected or required of the server, and what a client should do if a server supplies inconsistent data.

4.11 Unicast of DHCPDISCOVER

Section 4.4.4 Paragraph 1, "Use of broadcast and unicast," hints that clients may be able to unicast DHCPDISCOVER messages to servers: "The DHCP client broadcasts DHCPDISCOVER, DHCPREQUEST and

DHCPINFORM messages, unless the client knows the address of a DHCP server."

This would be pointless unless "ciaddr" were non-zero, because the server would not know how to respond. Neither does table 4 admit the possibility.

We believe it is common practice for BOOTP Relay Agents to only fill-in "giaddr" for broadcast packets. This requires investigation: such behavior would restrict the use of unicast DHCPDISCOVER messages to the same subnet on which the server resides -- a very restricted condition.

One circumstance in which this might make sense is for proxies gathering IP addresses on behalf of other clients. In that case, the proxy could put its own IP address in "ciaddr" and perhaps use multiple different client identifiers in multiple transmissions. Table 5, however, asserts that ciaddr must be zero.

RECOMMENDATIONS:

- The Working Group should consider whether to allow this kind of proxy usage, and what changes that might imply to RFC 2131.

- Tables 4 and 5 SHOULD be updated to reflect the possibility of unicast DHCPDISCOVER messages. Suggested replacements for Tables 4 and 5 are shown in sections 5.2.5-5.2.7.

- Figure 5 SHOULD be updated to reflect the uses of unicast and broadcast packets. Suggested replacements for Figure 5 are shown in sections 5.2.6 and 5.2.7.

4.12 DHCPRELEASE

There are several MUST NOT entries in the "options" portion of RFC 2131 table 5 specifying the inclusion of options in the DHCPRELEASE. Some customers complained that a particular vendor included the "hostname" option and that this seemed innocuous. The vendor said that their reading of the RFC allows such an option to be included.

In the "fields" portion of table 5 there is the word "unused" for the "sname" and "file" fields of a DHCPRELEASE message, while "options" for the DHCPRELEASE was left blank.

A DHCPRELEASE message SHOULD be subject to some verification criteria to reduce the chance of a bogus release. Two possible changes to these tables are:

- In the "fields" portion of table 5, change the "xid" from "selected by client" to "xid from server DHCPACK message."

- In the "options" portion of table 5, change the entry for "client identifier" from "MAY" to "client identifier used in the DHCPDISCOVER message."

4.13 Client State Diagram

Section 4.3.1 and Figure 5 do not accurately describe DHCP client behavior: DHCP clients send messages to servers from the INIT, INIT-REBOOT, SELECTING, REQUESTING, and BOUND states, not from RENEWING or REBINDING.

RECOMMENDATION:

Change the text of section 4.3.1 and its schematic representation in Figure 5 to correctly represent the states, transitions, triggering events, and messages sent. Suggested replacements for Figure 5 are shown in sections 5.2.6 and 5.2.7.

4.14 Options

The language in RFC 2131 concerning whether and which options to return to the client is convoluted and apparently contradictory.

4.14.1 Which Options to Return?

There are two opposing philosophies regarding which options servers should return to clients: to return every option with values within the client s scope, or to return only those options specifically requested by a client and within scope. The following arguments have been cited:

O Supporting the return of every option:

a. Consistency. A network administrator wants all of the configured options to show up on each client on the network, regardless of client vendor.

b. A DHCP client is likely only to request the options it supports. However, many application layer options are not used by the DHCP client but are useful to applications.

c. A DHCP client would either need to be configured or updated to request new options. The whole idea of DHCP is to keep configuration on the server, not on the client, which is pointed out in: Page 7, second and third bullets of section 1.6.

5. Supporting the return only of requested options:

a. Some DHCP clients may reject packets containing options that they did not request especially if they are ignorant of their semantics; therefore a DHCP server should only return the options requested.

b. The DHCP packet size is limited. Options are often configured on a per-network rather than a per-client basis, and to return unwanted options risks exhausting the space available while options remain which the client needs.

RFC 2131 does little to resolve the matter. Two different sections

are relevant: section 3.5 describes mechanisms to limit the number of options sent, while section 4.3.1 subsequently presents an apparently conflicting description of how to select values for options requested by the client.

6. RFC 2131, Section 3.5:

> "First, most 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."

The list of parameters with a cross-reference to the defining RFC is given in Appendix A of RFC 2131.

Several sources contend that virtually none of the parameters in the list have a meaningful default value, which raises the issue of viability of the technique described in this section for reducing total server response message size.

Even if the option has a default value defined in [RFC1122], RFC 2131 is silent on the question of whether or not the server MUST, SHOULD, or MAY choose not to send that option when its value is the same as the default.

7. RFC 2131, Section 4.3:

> "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 word "default" in the first statement seems misplaced. The second statement seems contrary to the intent of minimizing the amount of data sent by the server: if the scope of the Host Requirements RFCs applies to all Internet-connected hosts, then a DHCP server SHOULD NOT have to supply these values -- they should already be assumed by the client as the default for the requested option.

There is no mention of a minimum set of parameters to be sent to a requesting client, nor any mention of which parameters to send if the client does not request any not any guidance for what to do when there is more data than will fit in a response packet. Can the options be somehow prioritized? Could additional options be obtained using the DHCPINFORM mechanism? Should an additional bit in the "flags" field be defined as a "packet overflow" bit?

RECOMMENDATIONS:

- Clients MUST include the same parameter request list on all messages.

- Clients MUST be prepared to receive responses containing options

they did not request and/or whose semantics are unknown. They MAY choose silently to ignore such options.

- Language implying that parameters in "Requirements for Internet Hosts" have defaults should be removed.

4.14.2 Multiple Instances of Options

Page 24, seventh paragraph, section 4.1: "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."

RECOMMENDATION:

The first sentence SHOULD begin "Options MUST appear only once, unless...." The second sentence belongs in the options memo [RFC2132] for options where there can be multiple instances. Together, these two sentences are confusing.

4.14.3 Option Ordering

A number of clients require that the DHCP message type be the first option (after the magic cookie).

RECOMMENDATION:

With the exception of option 82, which must be last (save option 255 which acts as a terminator), the clients MUST NOT make any assumption about the ordering of options.

4.14.4 Options 66 and 67

Options 66 (TFTP server name) and 67 (bootfile name) were introduced as an alternative to the fixed fields "sname" and "file" in BOOTP. As discussed elsewhere, space is at a premium in DHCP, and reserving 64 octets ("sname") and 128 octets ("file") to contain values are that potentially, and commonly, much shorter is wasteful. Furthermore, the existence of these options allows the client to either request those values from the server or not, according to need.

At present, servers are at liberty to return values for these options in either the fixed fields, or encapsulated like any other DHCP option. Clients have sometimes assumed only the former. RFC 2131 should address this issue.

RECOMMENDATIONS:

1. Using "sname" and "file" for these options SHOULD be deprecated.

2. Clients MUST be prepared, at least for the time being, for either method of delivery.

4.15 Vendor Classes

Page 3, section 1.1, first paragraph - includes the following sentence: "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." Vendor classing has been there since RFC 1541, thus there is nothing new about it. Should this section be referring to User classing?

4.15.1 Character Set

Some new clients have spaces in their identifier, which broke some implementations with configuration file records delimited by whitespace.

4.15.2 Form of the Name Space

An early suggestion (RFC 1541 time-frame), expressed symbolically, was the form "Stock symbol/Organization...." e.g., "SUNW.class-1.class-2" or "CMU.edu.class-1.class-2". This would have had the advantage of preventing collisions between vendors. This was not adopted, and it is probably too late to resurrect it.

4.15.3 Relationship to Vendor Options

Text is needed describing how each unique vendor class identifier implies a 254 unique encapsulated option name space. There are 254, because even within the vendor space options 0 and 255 retain their meaning as the pad value and terminator, respectively. An occasional misconception is that there is only a single unique encapsulated 254 option name space shared by all vendors, with the effect that the same values being returned to *any* client regardless of vendor class identifier. Obviously, we should include text to clarify the relationship between Vendor Class identifier and the encapsulated Vendor option.

4.15.4 Multiplicity

How many vendor class identifiers can a client have? Only one class identifier, because the client is unique to a specific vendor. If the client were to send more than one vendor class option it would be impossible for the server to decide which set of encapsulated vendor options to select.

Here is some more text regarding vendor options from a note by Mike Carney regarding the use of vendor class / encapsulated options:

> Vendor class support requires the ability to configure a DHCP server to support a new vendor class by associating that vendor class identifier with 254 options whose types can then be defined by following the DHCP client's documentation. Each group of 254 options has the "scope" of that vendor. For example, suppose we have the following two clients:
>
> Vendor Class "SunBeam.Toaster.2slots"
> Options for this class:
> Code Len Data
> 1 2 Darkness setting (a 2 octet integer)

```
Vendor Class "Voxpopulis.Answering.Machine"
Options for this class:
Code  Len  Data
  1    4   Outgoing message index (pointer to messages)
```

Both clients are on the same network ("Kitchen"), and are clients of the same DHCP server. Note that both use encapsulated option code

1. Looks like a conflict, but it is not: in the syntax of the DHCP server's configuration table, one configures two new options, each which has the "scope" of the vendor class.

What this means is that when the toaster boots, the DHCP server only returns vendor class options associated with the "SunBeam.Toaster.2slots" class. When the answering machine boots, it only sees vendor class options associated with the "GE.Answering.Machine" class. Clients of vendor classes not currently configured on the server do not see any encapsulated vendor options.

4.16 Client/Server Retransmission

Because DHCP servers are the passive participants and DHCP clients are the active participants, the DHCP protocol is susceptible to poorly behaved clients (retransmitting too fast, for example). However, there is no text describing this susceptibility. Furthermore, the use of the power-of-2 retransmission algorithm is a SHOULD/MAY. This probably should be MUST. If we need different retransmission algorithms for different media, then we should develop/document them in table form. The specification as it stands is too loose and does cause inter-operability problems:

- Page 16, section 3.1, last sentence of list item 3.1.
- Page 17, third paragraph of list item 5, section 3.1
- Page 24, eighth paragraph of section 4.1
- Page 36, first paragraph of section 4.4.1

4.17 Transmission of DHCPNAKs

DHCPNAKs MUST be broadcast or unicast to at the link level because the client has no valid IP address. The same comment applies to DHCPOFFERs but with one significant difference: a DHCPOFFER has a valid "yiaddr" which a relay agent can use as the destination IP address. It is not clear that whatever mechanism relay agents are using to transmit offers will work when "yiaddr" is 0.0.0.0. Therefore, for safety s sake, the servers MUST set the broadcast bit in DHCPNAK packets. The text describing a server's behavior when the client is accessible through a BOOTP relay agent does not do this:

- Page 19, last paragraph of list item 2, section 3.2.
- Page 23, fifth paragraph of section 4.1.
- Page 32, Last paragraph of "DHCPREQUEST generated during INIT-

REBOOT state," bullet, section 4.3.2.

This last describes the behavior that is required -- a server MUST set the broadcast bit in order for the relay agent to properly broadcast the DHCPNAK.

RECOMMENDATION:

Items (1) and (2) above should either duplicate the text of (3), or should reference section 4.3.2.

4.18 Use of ciaddr

According to RFC 951 and RFC 1542, clients use "ciaddr" when they have received an IP address from a source outside of BOOTP/DHCP, and can respond to ARPs.

The text in RFC 2131 is mostly supportive of this point with the following exception:

> Page 32, "DHCPREQUEST generated during REBINDING state:" section 4.3.2: "The DHCP server SHOULD check 'ciaddr' for correctness before replying to the DHCPREQUEST."

RECOMMENDATION:

This line should be struck from the document. Servers trust "ciaddr," period.

4.19 Size of a BOOTP/DHCP Frame

The description in RFC 2131 relating to the size constraints of DHCP packets (Page 10, first paragraph after Table 1, section 2) is inadequate.

4.19.1 Minimum Packet Size

RFC 951 states that a minimum BOOTP frame is 300 octets in length. Some BOOTP relay agents have been known to drop frames of less than 300 octets. RFC 951 is explicit on this point, but RFC 2131 just refers to RFC 951. Since DHCP is intended to be backward compatible with BOOTP, the protocol should continue to observe this lower bound.

RECOMMENDATION:

Text should be added stating explicitly that the minimum size of a DHCP frame is 300 octets.

4.19.2 Maximum Size, MTU, and Message Size Option

It has been thought necessary to avoid fragmentation of the IP packets in DHCP/BOOTP due to concerns that some clients would be unable to reassemble fragments before the IP stack is properly configured. RFC 951 states, "For simplicity it is assumed that the BOOTP packet is never fragmented." Regardless of theoretical

limitations in IP stack implementations, it is certain that there are several DHCP/BOOTP implementations, at both ends of the protocol, which will not reassemble.

Various comments in the WORKING GROUP imply that fragmentation could be avoided were the client consistently to include the MTU of the link layer interface. However, clients cannot be expected to be omniscient about other media over which packets travel en route to

servers. Servers must be endowed with this knowledge, which they MUST use to avoid packet fragmentation.

Once the IP stack is configured, and the IP stack is fully configured, the aforementioned limitation ceases to exist, and later stages of the protocol could allow larger packets (up to the UDP limit). DHCPINFORMs, especially, could benefit from this relaxation. There probably should be explicit text to allow larger packets (presumably up to the maximum PDU size) for later stages of the protocol.

A number of clients send small packets with the assumption that servers will not return a packet that is any larger than the one received from the client. Clients MUST NOT assume this. If the client cannot process a response larger than a certain size, the client MUST use the message size option to inform servers of this size. Note that this is NOT the same option as the MTU.

RECOMMENDATIONS:

- Servers and relay agents MUST ensure that IP datagram fragmentation does not occur at any stage in the protocol before the client IP stack is fully configured.

- Clients SHOULD communicate their link-layer frame size to the DHCP server via the DHCP MTU option.

- Clients MUST NOT assume that servers will return a packet no larger than the one they send. If the client has a limit on the size of the packet that it can process it MUST convey that limit to the server in the "maximum message size" option (57).

- Page 21, second paragraph, section 3.5, the first sentence SHOULD be changed to "The client SHOULD include the 'maximum DHCP message size' option to let the server know how large the server may make its DHCP messages, and the value of this option SHOULD be the MTU of the [client] network interface being configured."

- The WORKING GROUP SHOULD consider whether to allow fragmentation of packets after the client is fully configured, and how servers can divine this fact (e.g. a non-zero "ciaddr.")

4.20 Use of giaddr

Page 23, fifth paragraph, section 4.1: "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.'" True for DHCPACK, false for DHCPOFFER (a DHCPDISCOVER will never have anything but 0 as "ciaddr.")

4.21 Address Selection

Page 27, third paragraph, section 4.3.1: "Note that, in some network architectures (e.g., internets with more than on 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 recording in 'giaddr.'"

There are two differing view of this sentence:

> There is considerable detail in the rest of RFC 2131 trying to get the use of "giaddr" clear as it relates to BOOTP relay agents (RFC 951 and RFC 1542), then this sentence "undoes" this work. Serving multiple IP networks on the same wire should be either described in detail in its own section (with caveats) or as a separate informational RFC. Otherwise, the use of "giaddr" is unclear.

Alternatively:

> Additional supporting text should be added to RFC 2131 to the effect that servers having knowledge of network topology MAY choose to offer an address inconsistent with "giaddr" but consistent with that topology. Furthermore, the address offered may differ depending upon the contents of the vendor class, user class, and even the client identifier. All of these things are policy matters for the server.

4.22 Use of "secs" Field

The "secs" field has not been discussed much: many clients simply leave its value as zero, and few if any servers have used its value to modify their behavior. These practices seem acceptable. The value of "secs" SHOULD be the elapsed time (in seconds) since the client began trying to acquire or extend a lease on an IP address. A sixteen bit field, its maximum value is 65536. It is conceivable that due to server or network failure that a client may have been waiting longer than this.

RECOMMENDATION:

A client MAY choose to leave to ignore the secs field. If so, its value MUST be set to zero. If the client chooses to insert a value, the value SHOULD be time elapsed since the client began negotiating for an IP address. If the client has been waiting longer than 65536 seconds its value SHOULD be 65536. The value SHOULD NOT wrap around to zero.

4.23 Use of "htype" and "hlen" Fields

At least one vendor has used chaddr as a place holder for a value that was not in fact a link-layer (hardware) address, while at the same type using an htype of 1 (meant to be Ethernet) but an hlen of 16 (instead of 6). Many servers will reject a packet with this kind of inconsistency between the htype and hlen fields.

Values of htype not equal to zero MUST correspond to the link-level medium to which the DHCP client is attached according to IANA s

Assigned Numbers database.

RECOMMENDATIONS:

1. The value of hlen MUST be consistent with the length of a link-level address implied by htype.

2. An htype of zero SHOULD be used to mean that chaddr is an identifier unrelated to a specific link-level medium.

4.24 Use of "xid" Field

This field exists to allow clients to match replies to requests. In two places RFC 2131 erroneously states that the client should use the "xid" in the server s DHCPOFFER as the value in its follow up request.

1. Table 5, DHCPREQUEST column.

2. Section 4.4.1, Paragraph 5.

In principle, the 32 bits of "xid" should be sufficient to make the chance of collisions almost nil, provided it is randomly generated as 2131 suggests in section 4.4.1 paragraph 3. However, some vendors have admitted to generating "xid" which may not be sufficiently uniformly distributed.

The randomness requirement on "xid" is not as stringent as would be required, say, in selecting a cryptographic key. It is quite permissible that the initial key be predictable given sufficient knowledge of the client, but clients MUST ensure that these identifiers are generated in such a way that the chance of collision with other clients in the DHCP administrative domain is what one should expect from a truly random number.

Permitting the use of the same "xid" on a re-transmission might marginally improve the efficiency of the protocol. Server responses to the first transmission, which arrived after the timeout and retransmission would be accepted, and might avoid yet another client timeout.

4.25 Options in DHCPOFFER and DHCPACK

RFC 2131 says that the options delivered in these two cases should not be in conflict. It does not say what "conflict" means in that case. This SHOULD be clarified.

RECOMMENDATIONS:

1. Servers MAY deliver a full configuration in a DHCPOFFER, but are NOT required to do so. The DHCPOFFER MUST contain an IP address and a lease time, and MAY contain other information. As a client will presumably choose among multiple offers based on some criteria, perhaps completeness of response, the server SHOULD be permitted to return the 'parameter request list' including the option code for each option it is prepared to deliver in a DHCPACK message.

2. In a DHCPACK message, the lease time offered MUST be at least as long as that in the DHCPOFFER. The IP address MUST be the same.

3. The options delivered in a DHCPACK message MAY differ from those in a DHCPOFFER. Some DHCP servers attempt to balance the load for other services by permuting lists. For instance, a server configured with three DNS server addresses may rotate that list

each time a client is serviced. It is a problem for some servers to deliver an identical OFFER and ACK (it implies keeping state.)

4. If a particularly long option list must be delivered to the client, it might not be possible to fit all options in the DHCP payload of a UDP packet. RFC 2131 appears to permit a long list of options to be sent partly in the DHCPOFFER message and partly in the DHCPACK message.

4.26 Lease Times

RFC 2131 has some language (section 4.3.1) that might be interpreted as constraining the duration of a lease that can be offered based on history or what the client wants. This SHOULD be rewritten to make clear that in a DHCPOFFER a server can offer whatever lease time that local policy finds acceptable, without regard to what the client requests, or what was offered last time around. If a server offers a longer lease than the client requested, the client can simply enter the RENEWING or REBINDING states, or send a DHCPRELEASE message according to its desired [earlier] times.

For RENEWALS, the text should be made clear that servers are not obligated to extend leases merely because the client wishes an extension [see also general comment below about policy.]

There has sometimes been an issue with T1 and T2 times as follows. Let us say that a new lease is offered with a certain T0 (T0 is lease duration) and T1 = T0/2. Then, when T1 expired, the client attempts a renewal. The server in question, for whatever reason, does not want to extend the lease, but is willing to confirm the residual time (T0/2). If it also returns T1 in the options, it should ensure that T1 is adjusted from the original value else the new ACK will have T0 and T1 identical.

4.27 Miscellaneous

There are many SHOULDs and SHOULD NOTs that should perhaps be converted into MUSTs or MUST NOTs. Here is a summary:

1. Page 16, item 4, section 3.1 "The server SHOULD NOT check the offered network address at this point." (MUST NOT)

2. Page 16, item 5, section 3.1 "The client SHOULD perform a final check on the parameters..." (MUST)

3. Page 17, item 5, section 3.1 "The client SHOULD wait a minimum of ten seconds..." (MUST)

4. Page 18, item 2, section 3.2 "Servers SHOULD NOT check that the

client's network address is already in use..." (MUST NOT)

5. Page 19, item 2, second paragraph, section 3.2 "...servers SHOULD respond with a DHCPNAK message to the client" (MUST). The following sentences are rather dubious in this paragraph as well.

6. Page 21, first paragraph, section 3.4 "The servers SHOULD unicast the DHCPACK replay to the address given in the 'ciaddr' field of the DHCPINFORM message" (MUST)

7. Page 22, last paragraph, section 3.5 "If a server receives a DHCP request message with an invalid 'requested IP address', the server SHOULD respond to the client with a DHCPNAK message...." (MUST)

RECOMMENDATION:

The WORKING GROUP should review the use of emphasis words (e.g., MAY) in RFC 2131. Those SHOULDs that remain should list the valid exceptions (some do; most don't).

5 Proposed Replacements for RFC 2131 Figures and Tables

5.1 Figures

5.1.1 Figure 1: Format of a DHCP message

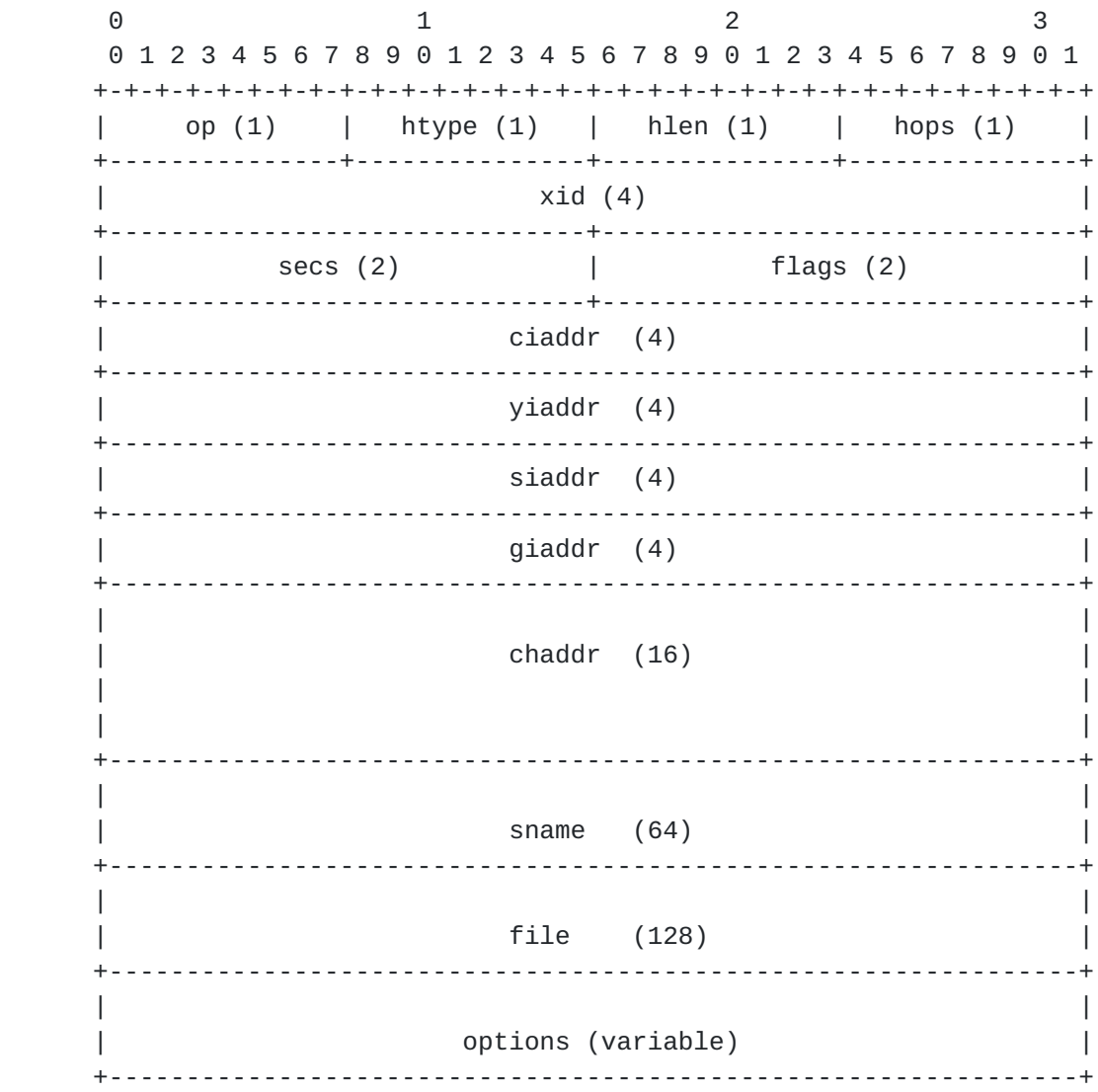


Figure 1: Format of a DHCP Message

5.1.2 Figure 2: Format of the 'flags' Field

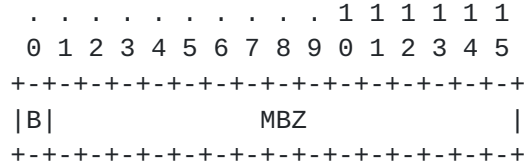


Figure 2: Format of the 'flags' Field

5.1.3 Figure 3: Timeline Diagram--Allocating a New Address

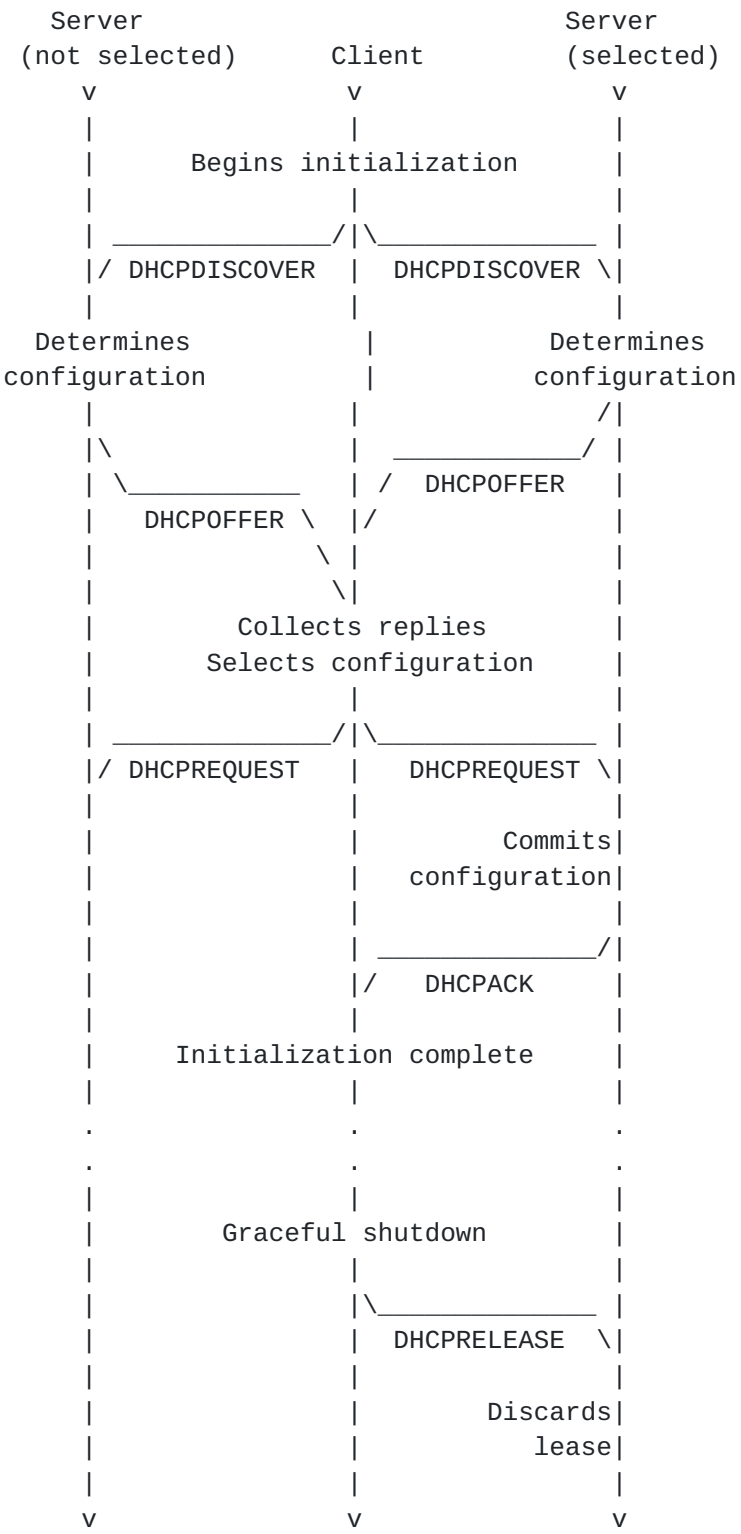


Figure 3: Timeline Diagram of Messages Exchanged between DHCP Client and Servers When Allocating a New Network Address

5.1.4 Figure 4: Timeline Diagram--Reusing an Address

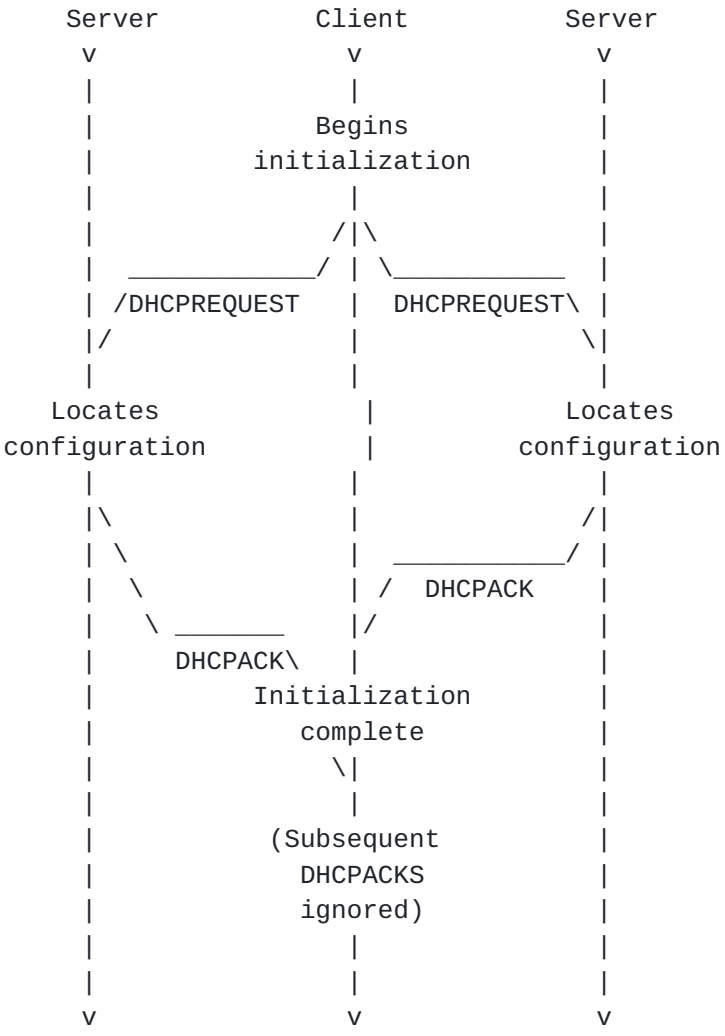
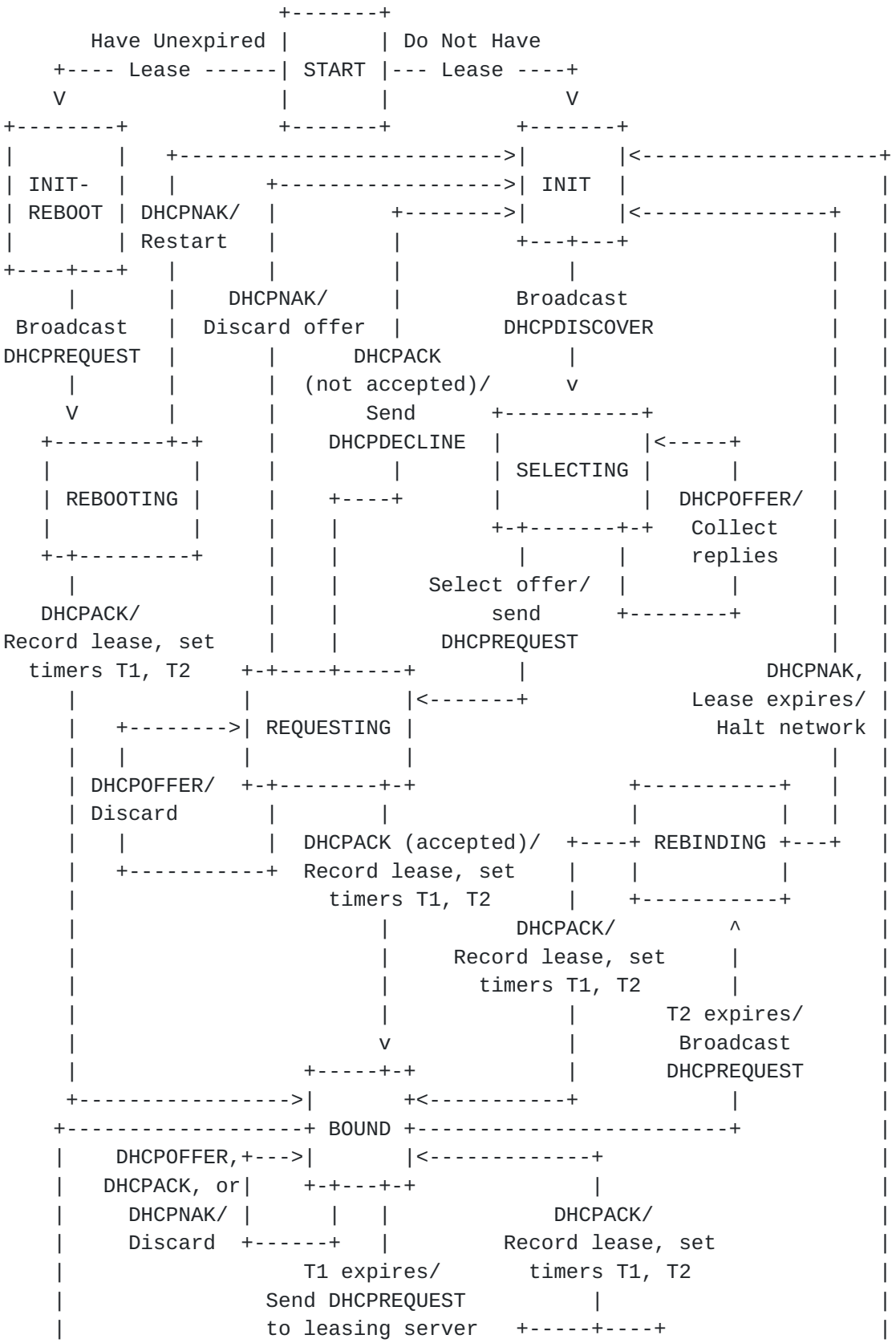


Figure 4: Timeline Diagram of Messages Exchanged between DHCP Client and Servers When Reusing a Previously Allocated Address

5.1.4 Figure 5: State-Transition Diagram for DHCP Clients



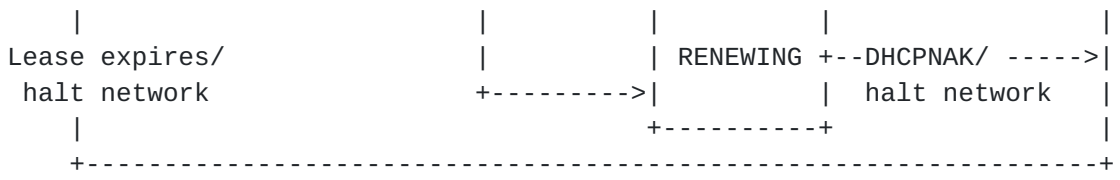


Figure 5: State-transition diagram for DHCP clients

5.2 Tables

5.2.1 Table 1: Description of fields in a DHCP Message

FIELD	OCTETS	DESCRIPTION
'op'	1	Message op code / message type. 1 = BOOTREQUEST, 2 = BOOTREPLY
'htype'	1	Hardware address type, see ARP section in "Assigned Numbers" RFC; 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'	vari-able	Optional parameters field. See the options documents for a list of defined options.

Table 1: Description of Fields in a DHCP message

5.2.2 Table 2: DHCP Messages

Message	Usage
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 canceling remaining lease.
DHCPINFORM	Client to server, asking only for local configuration parameters; client already has externally configured network address.

Table 2: DHCP Messages

5.2.3 Table 3: Fields Used by DHCP Servers

Field	DHCPOFFER	DHCPACK	DHCPNAK
'op'	2	2	2
'htype'	(From "Assigned Numbers" Database)		
'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'	(see Table 4)	(see Table 4)	(see Table 4)

Table 3: Fields Used by DHCP Servers

5.2.4 Table 4: Options Used by DHCP servers

Option	DHCPOFFER	DHCPACK	DHCPNAK
Requested IP address	MUST NOT	MUST NOT	MUST NOT
IP address lease time	MUST	MUST (DHCPREQUEST)	MUST NOT
		MUST NOT (DHCPINFORM)	
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 4: Options Used by DHCP servers

5.2.5 Table 5: Client Messages from Different States

	INIT-REBOOT	SELECTING	RENEWING	REBINDING
broad/uni-cast	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 5: Client Messages from Different States

5.2.6 Table 6: Fields Used by DHCP Clients

Field	DHCPDISCOVER, DHCPINFORM	DHCPREQUEST	DHCPDECLINE, DHCPRELEASE
'op'	1	1	1
'htype'	(From "Assigned Numbers" Database)		
'hlen'	(Hardware address length in octets)		
'hops'	0	0	0
'xid'	selected by client	'xid' from server DHCPOFFER message	selected by client
'secs'	seconds since DHCP process started	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'	(see Table 7)	(see Table 7)	(see Table 7)

Table 6: Fields Used by DHCP Clients

5.2.7 Table 7: Options Used by DHCP Clients

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 7: Options Used by DHCP Clients

5.2.8 Table 8: Host Configuration Parameters--IP Layer

IP-layer parameters, per host:		
Parameter	**Permissible Values**	**Reference**
Be a router	on/off	HR 3.1
Non-local source routing	on/off	HR 3.3.5
Policy filters for non-local source routing	(list)	HR 3.3.5
Maximum reassembly size	integer	HR 3.3.2
Default TTL	integer	HR 3.2.1.7
PMTU aging timeout	integer	MTU 6.6
MTU plateau table	(list)	MTU 7
IP-layer parameters, per interface:		
Parameter	**Permissible Values**	**Reference**
IP address	(address)	HR 3.3.1.6
Subnet mask	(address mask)	HR 3.3.1.6
MTU	integer	HR 3.3.3
All-subnets-MTU	on/off	HR 3.3.3
Broadcast address flavor	0.0.0.0 or 255.255.255.255	HR 3.3.6
Perform mask discovery	on/off	HR 3.2.2.9
Be a mask supplier	on/off	HR 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)	HR 3.3.1.6
preference level	integer	HR 3.3.1.6
Static routes, list of:		
destination	(host/subnet/net)	HR 3.3.1.2
destination mask	(address mask)	HR 3.3.1.2
type-of-service	integer	HR 3.3.1.2
first-hop router	(address)	HR 3.3.1.2
ignore redirects	on/off	HR 3.3.1.2
PMTU	integer	MTU 6.6
perform PMTU discovery	on/off	MTU 6.6

Key:

HR -- Host Requirements Communications Layers (RFC 1122, Internet Standard)

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

RD -- Router Discovery (RFC 1256, Proposed Standard)

Table 8: Host Configuration Parameters--IP Layer

5.2.9 Table 9: Host Configuration Parameters--Link Layer

Link-layer parameters, per interface:		
Parameter	Permissible Values	Reference
Trailers	on/off	HR 2.3.1
ARP cache timeout	integer	HR 2.3.2.1
Ethernet encapsulation	(RFC 894/RFC 1042)	HR 2.3.3
Key:		
HR -- Host Requirements Communications Layers (RFC 1122, Internet Standard)		

Table 9: Host Configuration Parameters--Link Layer

5.2.10 Table 10: Host Configuration Parameters--TCP

TCP parameters, per host:		
Parameter	Permissible Values	Reference
TTL	integer	HR 4.2.2.19
Keep-alive interval	integer	HR 4.2.3.6
Keep-alive data size	0/1	HR 4.2.3.6
Key:		
HR -- Host Requirements Communications Layers (RFC 1122, Internet Standard)		

Table 10: Host Configuration Parameters--TCP

6 Contributors

This document is the result of work undertaken the by DHCP working group. The editors would like to include a number of contributors to this effort including Mike Carney of Sun Microsystems, Steve Tulloh of Shadow Support, Bernie Volz, Ted Lemon of Nominum, Simon Vogl, Edward Mascarenhas of SGI, Andre Kostur of Incognito, Bud Millwood of Weird Solutions, Patrick Guélat of ImproWare Network Services, and Swamy Narasimha of Nokia.

7 IANA Considerations

This memo contains no values requiring IANA attention.

8 Security Considerations

(To be defined when suggested text changes for RFC 2131 are completed.)

A separate Internet-Draft is being created to provide a threat analysis of RFCs 2131 and 3118.

9 References

9.1 Normative References

[RFC951] Croft, W., and Gilmore, J., "Bootstrap Protocol," RFC 951, September 1985.

[RFC1123] R. Braden, "Requirements for Internet Hosts -- Application and Support," October 1989.

[RFC1542] W. Wimer, "Clarifications and Extensions for the Bootstrap Protocol" RFC 1542, October 1993

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol," March 1997.

[RFC2132] Alexander, S. and Droms, R., "DHCP Options and BOOTP Vendor Extensions," March 1997.

9.2 Informative References

[RFC3203] T'Joens, Y., Hublet, C., and De Schrijver, P., "The DHCP Reconfigure Extension," July 2001.

[RFC4388] Woundy, R., and Kinnear, K., "Dynamic Host Configuration Protocol (DHCP) Leasequery," February 2006.

<draft-ietf-dhc-fqdn-option-13.txt>, Stapp, M., and Rekhter, Y.,

"The DHCP Client FQDN Option," March 2006.

Authors' Addresses

Barr Hibbs
Richard Barr Hibbs, P.E.
952 Sanchez Street
San Francisco, California 94114-3362
USA

Phone: +1-(415)-648-3920
E-mail: rbhibbs@pacbell.net

Rob Stevens
308 Arthur Avenue
Aptos, California 95003-5202
USA

Phone: +1-(831)-688-9722
E-mail: robs@cruzio.com

Full Copyright Statement

Intellectual Property

specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

APPENDIX: NOTES

This appendix will be removed when this memo goes to Working Group Last Call.

Issues List

Open, unresolved issues about RFC 2131 include:

1. What is the correct use of client identifier in DHCPOFFER and DHCPACK messages?

2. Are there any effective alternatives to ICMP ECHO and ARP for address-in-use detection?

3. What is the definition of a Fully Qualified Domain Name?

4. Should DHCPINFORM messages be allowed from client proxies?

5. Should client proxies, in general, be allowed, and how does a client proxy know the IP address of a DHCP server?

6. Should a DHCP server send only options requested by a client, or should a server send all options for which it is configured with a value?

7. Should required usage of "xid" and "client identifier" be changed to support server verification of DHCPRELEASE messages?

8. What is the correct statement about selecting an IP address to offer a client when the offered address is on a different subnet than the client's "giaddr?"

9. Should a new flags bit to signify "more options data available" be added?

10. Do we need a new "Maximum Relay MTU Size" option to ensure that all reply packets sent by a server will pass without fragmenting or dropping packets?

11. Would it help to set a sort of "more to come" option, indicating that more options will follow in a consecutive DHCPACK, where the subsequent DHCPACKs would have a "additional information" option indicating that the message contains only new options data similar to a DHCPACK in response to a DHCPINFORM message?

12. Are unicast DHCPDISCOVER messages permitted? What are the requirements for specific message fields and options in this case?

13. What level of consistency is required among responses from multiple servers?

14. Can the REBINDING state be entered from the BOUND state on expiration of T2, or only if there is a timeout in RENEWING state?

15. Should the text of RFC 4361, Node-Specific Client Identifiers, be folded into a revised RFC 2131 and RFC 2132?

16. Should the text of RFC 4388, DHCP Leasequery, be folded into a revised RFC 2131 and RFC 2132?

Changes from Prior Drafts

"-00" Draft

The "-00" revision was the initial version of this memo, submitted to the Internet-Drafts editor on 23 February 2003.

"-01" Draft

The "-01" revision contains substantial changes following a detailed review of DHC Working Group mailing list discussions on RFC 2131 clarification issues, consideration of several directed questions, and comments received by the authors. Changes include:

- Reorganization of the document to group all typographical errors together, separate from protocol or policy issues.
- Elimination of "Interaction with DNS" and "Client and Server Administration" sections because the authors saw no clear resolution to the topics.
- Creation of an issues list in section 4.1.

"-02" Draft

The "-02" revision consists of numerous minor typographical, spelling, and grammatical updates, plus:

- Replaced all Internet-Draft Boilerplate with current versions.
- Removed all of Section 5, "Notes" to this appendix.
- Replaced Section 5 with the proposed figures and tables for use with revised text of RFC 2131-bis.
- Removed Appendix A.
- Separation of Section 9, "References," into Normative and Informative subsections.
- Updated Authors' Addresses.
- Added text to Section 4.5.1 covering RFC 4361, "Node-Specific Client Identifiers for DHCPv4."

- Recent mailing list discussion about the correct use of the "secs" field was incorporated into Section 4.22.