NetBoot 2.0:
Boot Server Discovery Protocol (BSDP)

Author: Dieter Siegmund, dieter@apple.com
Date: December 8, 2003
Version: 1.0.7

1 Introduction

A NetBoot 2.0 client uses the Boot Server Discovery Protocol (BSDP) to dynamically acquire resources that enable it to boot a suitable operating system. The client uses DHCP to acquire its IP address and BSDP to acquire boot image resources. The protocols are initiated by the client at boot time.

This document describes the protocol and how it interoperates with DHCP. It also describes specifics of the packet encodings and the client and server logic.

1.1 Version History

Version	Date	Who	Modifications
0.1	8/12/1999	Dieter Siegmund	Initial version with basic protocol description.
1.0.2	10/25/1999	Dieter Siegmund	Specified packet/option encoding; detailed client logic and behavior.
1.0.2b	11/3/1999	Dieter Siegmund	Revised based on review input.
1.0.3	01/18/2000	Dieter Siegmund	Fixed DHCP OFFER tables.
1.0.4	02/26/2002	Dieter Siegmund	Defined boot_image_id encoding. Removed combined BSDP/DHCP server scenario. Defined a new Boot Image List option to encode list of images directly in the ACK[LIST]. Marked Boot Image List Path option as unused. Detailed non-firmware BSDP client details. Added firmware identification section. Bumped the BSDP version to 0x0101.
1.0.5	04/29/2002	Dieter Siegmund	Added image type attributes. Added NetBoot 1.0 Firmware option. Clarified that the entire (32-bit) boot_image_id should be tested for uniqueness, not just the 16-bit index.
1.0.6	11/17/2003	Dieter Siegmund	Added definitions in support of diagnostics boot.
1.0.7	12/08/2003	Dieter Siegmund	Modified protocol description in support of Image List filtering to make the client logic more compatible with existing NetBoot servers, and simplify server logic.

2 NetBoot 2.0 vs. NetBoot 1.0

NetBoot 2.0 has several new features and improvements over the previous version of NetBoot:

1. The client gets its IP address using DHCP
2. Multiple boot servers can serve the same subnet (or subnets)
3. A single boot server can serve multiple operating system images e.g. Mac OS 9, Mac OS 9 French, Mac OS 9 Japanese, Mac OS X

3 BSDP Overview

BSDP is implemented on top of DHCP. DHCP allows vendor-specific information to be identified and encapsulated in a standard way. BSDP uses vendor-specific information to provide the additional NetBoot functionality not present in standard DHCP. The use of DHCP packets also enables a client to boot from a server on a remote subnet through the standard BOOTP/DHCP relay agent mechanism (RFC 951 and 1542).

The protocol is implemented in client firmware. At boot time, the client obtains an IP address via DHCP (RFC 2131) then discovers boot servers using BSDP. Each BSDP server responds with boot information consisting of:

1. A list of bootable operating system images
2. The default operating system image
3. The client's currently selected operating system image (if defined)

The client chooses an operating system from the list and sends a message to the server indicating its selection. The selected boot server responds supplying the boot file and boot image, and any other information needed to download and execute the selected operating system. The client receives the message then downloads the boot image using TFTP and begins executing it.

Once a client has configured a boot image with a particular BSDP server, that server responds to the client's subsequent DISCOVER requests supplying boot image information in the OFFER. This means that the client can skip BSDP on subsequent reboots.

3.1 Protocol Details

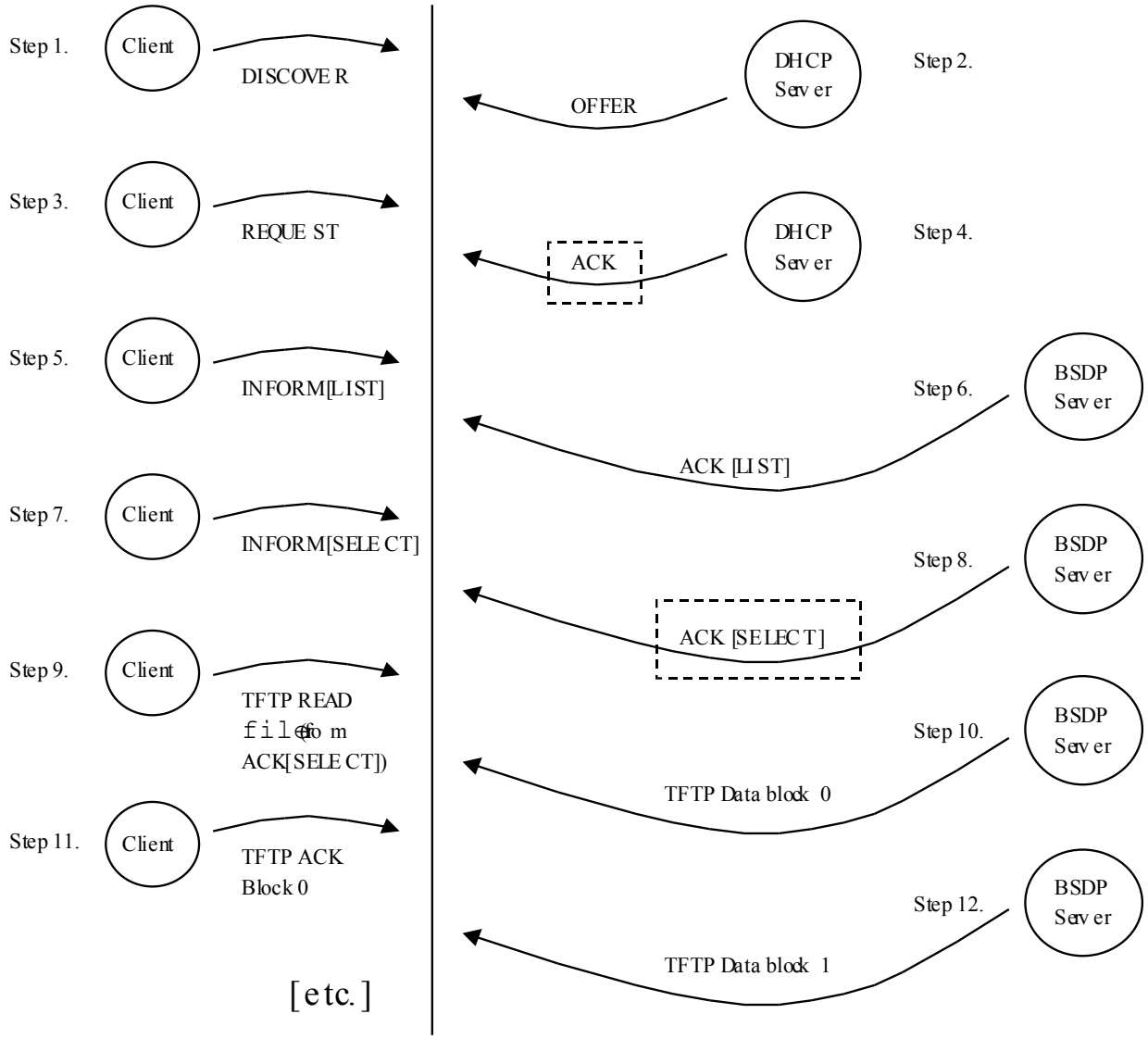
BSDP uses a two-packet exchange mechanism modeled after DHCP. The first packet sent by the client generates a response from a BSDP server. The response contains a list of possible boot images that the client may load. The second packet sent by the client selects a particular server and boot image. The successful response from the selected server confirms the selection.

BSDP uses DHCP INFORM and ACK packets for the communication between the client and server. The Vendor Class Identifier option is set to a value that identifies it as a BSDP packet. The Vendor Specific Information option includes a BSDP Message Type option that specifies one of LIST, SELECT, or FAILED.

The next section gives an overview of the packet exchanges used by a client during its first boot. Subsequent sections describe the packet exchanges used during subsequent boots.

Notational note: In the sections that follow, a DHCP packet that contains a BSDP Message Type option is written as "DHCP message[BSDP message]". For example, a DHCP INFORM packet with the BSDP Message Type option set to LIST is written as INFORM[LIST].

Figure 1: Initial Boot Sequence



3.2 Initial Boot Sequence

The boot sequence for a client that's booting for the first time (see Figure 1) is:

Step 1. Client sends DISCOVER

The client broadcasts a DISCOVER packet to generate an offer of an IP address from DHCP servers.

Step 2. DHCP Server sends OFFER

DHCP server responds with an OFFER packet containing the suggested IP address in the `yiaddr` field (see RFC 2131). The server supplies a Server Identifier option that is used in the next step to identify this particular server.

Step 3. Client sends REQUEST

The client selects an OFFER from the previous step and broadcasts a REQUEST message containing the Server Identifier option from the selected server.

Step 4. DHCP Server sends ACK
The selected DHCP server sends an ACK packet to the client to confirm the IP address binding. **Note:** Client firmware must save this ACK in a location accessible by the loaded operating system.

Step 5. Client sends INFORM[LIST]
The client broadcasts an INFORM[LIST] packet to generate a reply from BSDP servers.

Step 6. BSDP server sends ACK[LIST]
A BSDP Server responds with an ACK[LIST] packet containing the BSDP Boot Image List Path option, the BSDP Default Boot Image option, and the BSDP Server Identifier option.

Step 7. Client sends INFORM[SELECT]
The client selects an ACK[LIST] from the previous step[1], and broadcasts an INFORM[SELECT] packet containing the BSDP Server Identifier option for the selected server and the BSDP Selected Boot Image option to indicate which OS it has selected

Step 8. BSDP server sends ACK[SELECT]
The BSDP server receives the INFORM[SELECT] and replies with an ACK[SELECT] that contains a path to the boot `file` to be downloaded using TFTP plus other options required to boot the selected boot image. **Note:** Client firmware must save this ACK[SELECT] in a location accessible by the loaded operating system.

Steps 9 and onward: Client TFTP's boot `file` from BSDP server
The client sends a TFTP Read request to the BSDP server specifying the path `file`, the server responds with Data block 0, the client replies with an ack, the server sends the next data block, and so on until the entire image is downloaded. The client begins executing the image.

3.3 Subsequent Boot Sequence

On subsequent boots, several steps in the boot process can be skipped because the client has a boot image binding with a particular BSDP server. The BSDP server responds to the client's initial DHCP DISCOVER with an OFFER that contains all of the options required to load and boot the client's currently selected boot image.

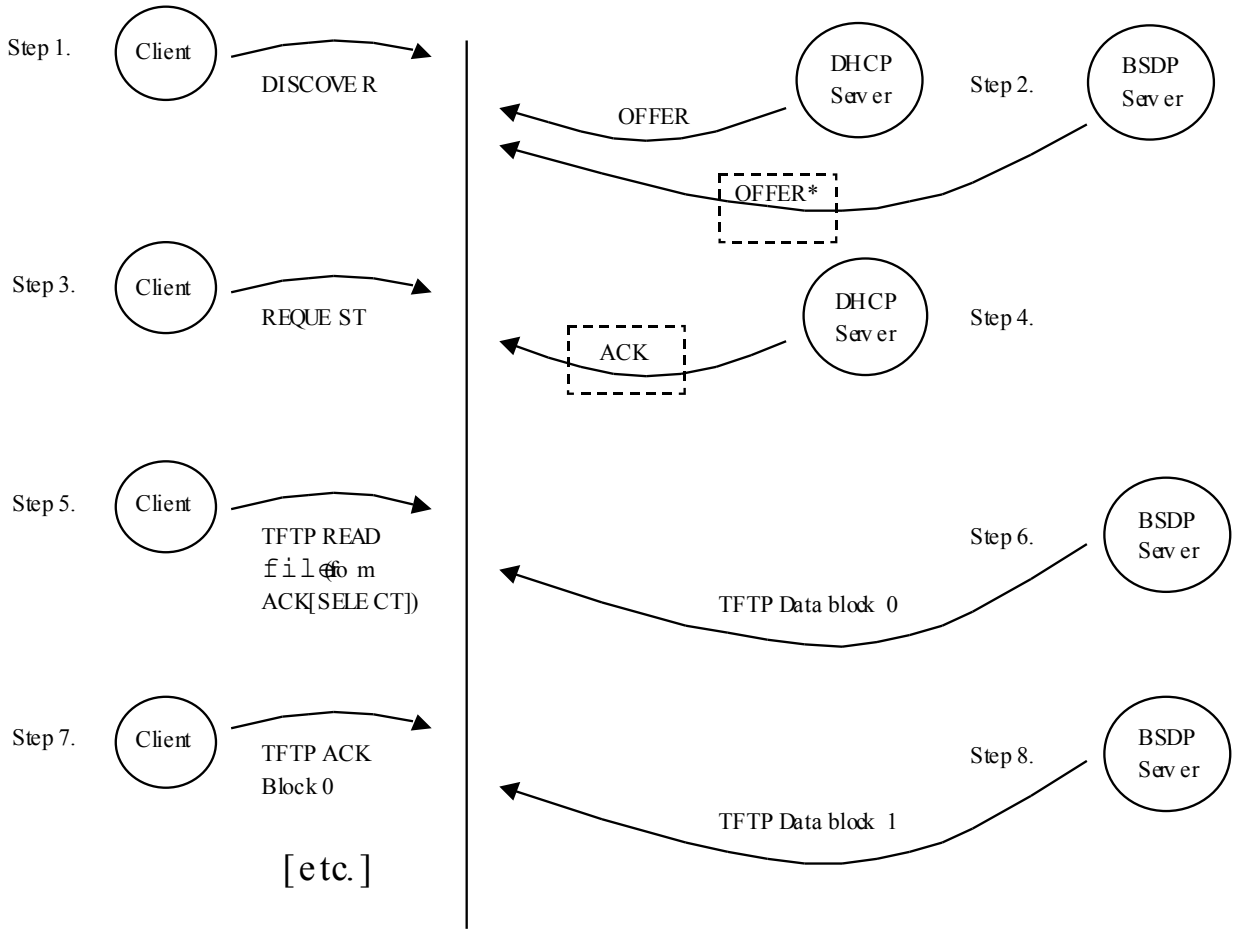
The boot sequence in this case is (see Figure 2):

Step 1. Client sends DISCOVER

[1] Details on client selection criteria are discussed fully in a later section.

The client broadcasts a DISCOVER packet to generate an offer of an IP address from DHCP servers and an offer of a boot image from a BSDP server.

Figure 2: Subsequent Boot Sequence



Step 2. DHCP server sends OFFER, BSDP server sends OFFER*

A DHCP server responds with an OFFER of an IP address in the `yiaddr` field and inserts its Server Identifier option. A BSDP server responds with an OFFER (labeled OFFER* in Figure 2) that contains the boot `file` and boot image options but no IP address i.e. the `yiaddr` field contains 0.0.0.0.

Note: Client firmware must save the OFFER* from the BSDP server in a location accessible by the loaded operating system.

Step 3. Client sends REQUEST

The client selects an OFFER from one of the DHCP servers in the previous step and broadcasts a REQUEST message containing the Server Identifier option from the selected server.

Step 4. DHCP Server sends ACK
The selected DHCP server sends an ACK message to the client to confirm the IP address binding. **Note:** Client firmware must save this ACK in a location accessible by the loaded operating system.

Steps 5 and onward. Client TFTP's boot `file` from BSDP server

3.4 Packet Format and Option Encoding

BSDP uses DHCP packets as defined in RFC 2131 and the option encoding as defined in RFC 2132. BSDP defines its own encoding for the Vendor Class Identifier and Vendor Specific Information options. These options allow the BSDP-specific data to be encapsulated and ignored by DHCP servers that don't understand BSDP.

Options defined in RFC 2132 are denoted as "DHCP option code xxx" in the descriptions that follow. Options appearing inside the Vendor Specific Information option are denoted as "BSDP option code xxx".

3.4.1 Basic Types

The following table defines the types that are used in the descriptions that follow. All multiple byte integer values are in network byte order i.e. big-endian.

Table 1: Basic Types

Type	Encoding/Description
`string`	usual C-language string i.e. null-terminated string of ASCII characters
`ascii_string`	a sequence of ASCII characters, no null-termination, length given by option length
`opaque`	a sequence of bytes, length is given by the option length
`uint8`	unsigned 8-bit integer (1-byte)
`uint16`	unsigned 16-bit integer (2-bytes)
`uint32`	unsigned 32-bit integer (4-bytes)
`ipaddr`	`uint32` containing an IP address
`port`	`uint16` containing an IP port number
`boot_image_id`	`uint32` containing boot image id (see section 3.4.2)

3.4.2 Encoding of `boot_image_id`

The `boot_image_id` is a `uint32`. The following table gives its encoding. Byte 0 appears in memory first, followed by byte 1, 2, and byte 3. The bits of a byte are numbered with the most significant bit numbered 7, and the least significant bit numbered 0. For example, setting bit 7 in a byte gives a value of 0x80, setting bit 0 gives a value of 0x1.

<table>
<tr><th colspan="10">boot_image_id</th></tr>
<tr><th colspan="8">Attributes</th><th colspan="2">Index</th></tr>
<tr><th colspan="8">Byte 0 (MSB)</th><th colspan="8">Byte 1</th><th>Byte 2</th><th>Byte 3 (LSB)</th></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td colspan="2" rowspan="2">uint16</td></tr>
<tr><td>I</td><td colspan="7">Kind</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td colspan="8">I – Install: (Byte 0, bit 7)
I = 0: non-install
I = 1: install

Kind – The kind of image: (Byte 0, bits 0..6)
Values are defined as:
0x0 (0): Mac OS 9
0x1 (1): Mac OS X
0x2 (2): Mac OS X Server
0x3 (3): Hardware Diagnostics
0x4 – 0x7f (4 – 127): reserved

All remaining bits are reserved for future use.</td><td colspan="2">Values are defined as:
Index = 0x0 (0)
Null image, server messages never contain this value.

Index = 0x1 – 0xfff (1 – 4095)
Server-specific image. An image with an Index in this range is specific to the server and must be presented to the user as a unique item.

Index = 0x1000 – 0xffff
(4096 – 65535)
Globally unique. An image with an Index in this range is unique across all servers, and may be presented as a single choice when multiple servers provide an image with the same Index</td></tr>
</table>

The `boot_image_id` is encoded as two 16-bit values: **Attributes** and **Index**. The **Attributes** field contains attributes of the image. The **Index** is the identifier for the image and is encoded as a `uint16`.

The `boot_image_id` value zero (0) is used to denote the null image, and will never appear in messages from the server.

The **Attributes** field contains information about the type of image and includes the `Install` bit and the `Kind` field. When `Install` is set (`I=1`), the image is an installation image. When it is not set (`I=0`), the image is not an installation image. The `Kind` field indicates the kind of image. There are four kinds of images defined in the table: Mac OS 9, Mac OS X, Mac OS X Server, and Hardware Diagnostics (see the table above for the values). All remaining values are reserved for future use. The remaining bits of the **Attributes** field are reserved for future use, and must not be interpreted by the client.

The **Index** field is broken into two value ranges. One range (`0x1..0xfff`) is reserved for server-specific images. The second range (`0x1000..0xffff`) is reserved for images that are globally unique and are served by multiple NetBoot servers. An **Index** value of zero (0) is not valid in messages from the server (see the discussion above describing `boot_image_id`).

The **Index** allows differentiation between multiple occurrences of one `Kind` of image and helps form a unique identifier for the image.

A server-specific image must be presented to the user as a unique choice in a list of images. A globally unique image that is served by multiple servers may be presented to the user as a single choice if more than one server supplies an image with the same **boot_image_id** value.

3.4.3 Vendor Class Identifier option: DHCP option code 60

BSDP uses an encoding for this option that serves two main purposes:

- Identify the packet as a BSDP packet
- Identify the client's architecture and system type and revision so that a BSDP server can provide appropriate boot image information

Every packet that is sent by a BSDP client and server contains this option. It is encoded as an `ascii_string` containing "`AAPLBSDPC`" in messages from the server and "`AAPLBSDPC/<arch>/<system_id>`" in messages from the client.

`<arch>` indicates the architecture of the client. The defined values are "`ppc`" and "`i386`". `<system_id>` contains a value that identifies the client's system type and revision.

On Apple hardware, the `<arch>` contains "`ppc`" and the `<system_id>` contains the `model` property taken from the Open Firmware device tree root node "`device-tree:/`". Two example values are "`PowerMac3,1`" and "`PowerBook2,1`".

The encoding for server messages is:

Code	Length	Vendor Class Identifier								
60	9	A	A	P	L	B	S	D	P	C

The encoding for client messages is:

Code	Length	Vendor Class Identifier																	
60	11+i+j	A	A	P	L	B	S	D	P	C	/	A1	A2	…	Ai	/	S1	S2	…	Sj

where `A1, A2, …, Ai` and `S1, S2, …, Sj` correspond to the characters of the architecture type and system identifier respectively.

3.4.4 Vendor Specific Information option: DHCP option code 43

This option is used to encapsulate the BSDP options using the format specified in RFC 2132 for encapsulated options.

This option may appear more than once in responses from the server.

This option is encoded as:

Code	Length	Vendor Specific Information			
43	n	V1	V2	…	Vn

3.4.5 BSDP Message Type option: BSDP option code 1

The BSDP Message Type option is used to indicate the type of the BSDP message. The length of the option is 1. The possible values are:

Value	Message Type
1	LIST
2	SELECT
3	FAILED

The encoding is:

Code	Length	Message Type
1	1	1-3

3.4.6 BSDP Version option: BSDP option code 2

The client supplies this option in its messages to let the BSDP server know which version of the protocol it is using so it can respond to the client's request appropriately.

This length of this option is two. It is encoded as a `uint16` with value `0x0101` (corresponding to version 1.1):

Code	Length	Version	
2	2	0x01	0x01

Note: previous versions of this document specified a Version of 0x100.

3.4.7 BSDP Server Identifier option: BSDP option code 3

The client uses this option in its INFORM[SELECT] message to identify a particular BSDP server. This option is equivalent to the DHCP Server Identifier option (code 54). The DHCP option could not be used because it may not appear in a DHCP INFORM message.

The length of this option is 4 and the encoding is an `ipaddr`:

Code	Length	Address			
3	4	a1	a2	a3	a4

3.4.8 BSDP Server Priority option: BSDP option code 4

A BSDP server provides this option in an ACK[LIST] message to help the client select from multiple responses. The option indicates the priority of the server. A client favors a server that has the highest priority.

One purpose of this option is to help distribute load amongst several BSDP servers. A BSDP server lowers its priority as it becomes busier, encouraging clients to select a server that's less busy.

This option is a `uint16` with a range from 0, the lowest priority, to 65535, the highest priority. The client favors responses with the highest priority value.

The length of this option is 2 and the encoding is **:**

Code	Length	Priority	
4	2	p1	p2

3.4.9 BSDP Reply Port option: BSDP option code 5

A BSDP client inserts this option in BSDP messages to have the BSDP server direct its reply to a port other than the default, UDP port 68. If this option is not present, the BSDP server sends the reply to the default port. This option can only appear in the INFORM[LIST] or INFORM[SELECT] messages sent by the client.

The purpose of this option is to make it easier to write a BSDP client that runs on a particular operating system. The BOOTP/DHCP client port, UDP port 68, may be in use by a regular DHCP client, so having the server redirect its response allows the client to use a generic UDP port.

Note: this port must be a privileged port i.e. it must have a value less than 1024.

The length of the option is 2, and the encoding is `uint16`:

Code	Length	Reply Port	
5	2	p1	p2

3.4.10 BSDP Boot Image List Path option: BSDP option code 6 (NOT USED)

The purpose of this option is to provide the boot image list information in a format that includes an icon for each image. This format is too large to fit inside the ACK[LIST] packet itself, so must be retrieved using TFTP.

This option may be supplied by a BSDP server in an ACK[LIST] packet. It contains an `ascii_string` path to a file that can be retrieved from the server using TFTP. The format of this file is TBD.

Note: this option is not currently used.

The length of this option is variable. It is encoded as:

Code	Length	Path			
6	n	p1	p2	…	pn

3.4.11 BSDP Default Boot Image ID option: BSDP option code 7

The server supplies this option in an ACK[LIST] message to indicate which boot image is the default selection. The client may choose the value of this option when user input is not available and no response contains the BSDP Selected Boot Image ID option (see below).

This option is encoded as a `boot_image_id`. Its length is 4 and the encoding is:

Code	Length	Default Boot Image ID			
7	4	i1	i2	i3	i4

3.4.12 BSDP Selected Boot Image ID option: BSDP option code 8

Both the server and the client use this option. The server supplies this option in an ACK[LIST] message to indicate which boot image the client has currently selected. This option is only present if the server already has a boot image binding for the client. The client chooses a response that contains this option when user input is not available.

The client supplies this option in an INFORM[SELECT] message to tell the server which boot image it has selected.

This option is encoded as a `boot_image_id`. Its length is 4 and the encoding is:

Code	Length	Selected Boot Image ID			
8	4	i1	i2	i3	i4

3.4.13 BSDP Boot Image List option: BSDP option code 9

A BSDP server supplies this option in the ACK[LIST] packet if the BSDP Version is greater than or equal to `0x0101` (see section 3.4.6) It contains the list of boot images that are available from the server.

This option may appear multiple times, in multiple instances of the Vendor Specific option. Multiple instances are likely because the Vendor Specific option is limited to 255 bytes, and the server may have many images to vend. Regardless of where the instances of this option occur, the client must be able to retrieve all of them, and re-construct a single, concatenated list of images.

This option has the following encoding:

Code	Length	Image Description 1								
		ID 1 (4 bytes)				Count 1 (1 byte)	Name 1 (C1 bytes)			
9	L	i1	i2	i3	i4	C1	N1	N2	…	N_{C1}

Image Description 2								
ID 2 (4)				Count 2 (1)	Name 1 (C2)			
i1	i2	i3	i4	C2	N1	N2	…	N_{C2}

Image Description k								
ID k (4)				Count k (1)	Name 1 (Ck)			
i1	i2	i3	i4	Ck	N1	N2	…	N_{Ck}

After the **Code** byte and **Length** byte come any number of **Image Description**'s. An **Image Description** contains a 4-byte **ID**, followed by a 1-byte **Count**, followed by **Count**-bytes of **Name**. **ID** is encoded as a `boot_image_id` (see 3.4.2). The **Name** is encoded as a UTF-8 string. The overall length `L` of this option is:

```
L = 5k + SUM(Cj, j = 1, j <= k)
```

The client should set the Maximum DHCP Message Size option (code 57) to the maximum value allowed by the physical medium. For Ethernet this should be set to 0x5dc (1500). This ensures that the client will be able to retrieve the maximum number of boot images possible. Failure to supply this option may result in a truncated list.

3.4.14 BSDP NetBoot 1.0 Firmware option: BSDP option code 10

This option is supplied by a non-firmware BSDP client to inform the server that the client's firmware uses NetBoot 1.0, a simpler, BOOTP-based protocol. If this option is present, the server only responds if it has been explicitly configured to handle NetBoot 1.0 clients. This option allows a non-firmware BSDP client running on a machine with older firmware to use BSDP to negotiate which image to NetBoot, yet still honor the booting restrictions inherent in the older protocol.

This option is has no data thus its length is zero:

Code	Length
10	0

3.4.15 BSDP Boot Image Attributes Filter List option: BSDP option code 11

This option allows the client to request that the server filter images and return results that match the given list of image **Attributes**. Each **Attributes** value is 2 bytes long, and matches the definition described in section 3.4.2. Use of this option is optional.

To receive filtered responses, the client inserts this option in the DHCP DISCOVER, DHCP REQUEST, and BSDP INFORM[LIST] packets that it sends. A server, upon receipt of a packet containing this option, eliminates images that do not match the given list of **Attributes**. The filtering applies to each of the following returned options:

- Default Boot Image ID option (section 3.4.11)
- Selected Boot Image ID option (section 3.4.12)
- Boot Image List option (section 3.4.13)

If no images match the given image **Attributes** list, a server that supports this option will not respond. **Note:** existing servers that do not support this option will respond as if no image attributes filter were present. The client needs to verify that any BSDP ACK[LIST] packet it receives matches the desired image **Attributes** list. In particular, if the client supplies a single **Attributes** value, the client verifies that the Default and Selected Boot Image ID options values match the desired **Attributes**. For example, if the client wants to receive only "Hardware Diagnostics" image (Kind = 3), the client supplies the option:

Code	Length	Attributes1	
11	2	3	0

and checks that the Default Boot Image ID and Selected Boot Image ID options match:

Code	Length	Default Boot Image ID			
7	4	3	0	x	x

Code	Length	Selected Boot Image ID			
8	4	3	0	x	x

The data for this option must be a positive multiple of 2 bytes (N >= 1):

Code	Length	Attributes1		Attributes2		…	AttributesN	
11	2 * N	A11	A12	A21	A22	…	AN1	AN2

4 BSDP Firmware Client Details

This section describes the logic used by a BSDP firmware client during system boot. DHCP defines several states that the client transitions between to acquire an IP address. Sections 4.1through 4.2 describe how BSDP is intertwined with these states. The remaining sections describe the additional BSDP states that the client uses to acquire a suitable boot image.

4.1 DHCP INIT state

Soon after power-on, client firmware begins the network boot process in the DHCP INIT state (RFC 2131). The client forms a DHCP DISCOVER packet that includes the BSDP Vendor Class Identifier (see Table 2 below for the exact packet contents). It broadcasts the DISCOVER and sets a timer. If no responses arrive before the timer expires, the client retransmits a DISCOVER. The timeout values are doubled each time and a small amount of random fuzz is added to avoid synchronized broadcasts among several clients. The client retries several times, and if no packets arrive, it gives up and may choose to boot from an alternate boot device i.e. its internal hard disk. It may also choose to sleep for a period of time and begin the process over again. Once responses begin to arrive, the client transitions to the DHCP SELECT state.

Table 2: DHCP DISCOVER packet

DHCP DISCOVER PACKET			
IP and UDP Information			
Field		**Value**	**Description**
IP destination address		255.255.255.255	IP broadcast address
IP source address		0.0.0.0	Client has no address yet
UDP destination port		67	BOOTP/DHCP server port
UDP source port		68	BOOTP/DHCP client port
DHCP FIELDS			
Field	**Length (bytes)**	**Value**	**Description**
op	1	1	1=BOOTREQUEST
htype	1	?	1=10Mb Ethernet
hlen	1	?	6=10Mb Ethernet address length
hops	1	0	
xid	4	?	
secs	2	?	
flags	2	?	
ciaddr	4	0.0.0.0	
yiaddr	4	0.0.0.0	
siaddr	4	0.0.0.0	
giaddr	4	0.0.0.0	
chaddr	16	xx:xx:xx:xx:xx:xx	Client's thernet address
sname	64	0:0:…:0	unused
file	128	0:0:…:0	unused
magic	4	99.130.83.99	RFC 2132 magic number
DHCP OPTIONS			
Tag	**Code**	**Length (bytes)**	**Description**
Message Type	53	1	1=DISCOVER
Vendor Class Identifier	60	variable	see section 3.4.2
Vendor Specific Information	43	variable	see below and section 3.4.4
Parameter Request List	55	>= 4	Contains (at least) the bytes {1, 3, 43, 60}
IP Address Lease Time	51	4	optional; client can suggest a length of lease
Maximum DHCP Message Size	57	2	optional; used if the client can handle a packet larger than 576 bytes (the DHCP minimum)
BSDP OPTIONS (encapsulated in option 43)			
Tag	**Code**	**Length (bytes)**	**Description**
BSDP Version	2	2	see section 3.4.6
BSDP Image Attributes Filter	11	2 (multiple of)	optional, see section 3.4.15

DHCP SELECT state

Once the client receives its first appropriate response, the client cancels the previous timer, sets a new "gathering" timer[2] and continues to wait for responses. The client gathers:

1. DHCP OFFER from a DHCP server with IP address specified, or
2. BOOTP REPLY from a BOOTP server IP address specified
3. BSDP OFFER from a BSDP server with boot file specified (see Table 3 below)

The following pseudo-code details the gathering logic:

```
t = INITIAL_TIMEOUT;
gathering = DHCP_offer = BSDP_offer = BOOTP_reply = 0;
for (try = 0; try < MAX_TRIES; try++) {
    transmit(DISCOVER);
    set_timeout(t); t = t * 2;
    while (read_packet(&packet) != TIMEDOUT) {
        if (gathering == 0) {
            clear_timeout();
            set_timeout(GATHERING_SECS);
            gathering = 1;
        }
        if (packet.type == BOOTP) {
            if (BOOTP_reply == 0) {
                BOOTP_reply = packet;
            }
        }
        else if (packet.msgtype != OFFER)
            ;
        else if (packet.vendor_class_id == BSDP) {
            BSDP_offer = packet;
        }
        else { /* DHCP packet */
            DHCP_offer = packet;
        }
        if (DHCP_offer != 0 && BSDP_offer != 0)
            break; /* out of while */
    } /* while */
    if (DHCP_offer != 0 || BSDP_offer != 0 || BOOTP_reply != 0)
        break; /* out of for */
} /* for */
clear_timeout();
```

After the gathering period ends, the client looks at the responses it saved and decides how to proceed. If it received no IP address information i.e. no DHCP OFFER or BOOTP REPLY, it retries DHCP. If the client receives a DHCP OFFER, it moves to the DHCP REQUEST state to confirm the IP address (discussed in the next section). If the client receives a BOOTP REPLY, it can simply use the specified IP address.

If no BSDP response is received, the client continues on and tries BSDP (discussed in the following sections). The following pseudo-code details the post-gathering logic:

[2] A reasonable length for the gathering time is 2 to 4 seconds.

```
if (BOOTP_reply == 0 && DHCP_offer == 0)
    goto retry_dhcp;

DHCP_ack = 0;
if (DHCP_offer != 0) {
    /* send DHCP REQUEST to DHCP_offer.server_id, wait for ACK */
    if (requesting_state(DHCP_offer,&DHCP_ack) == FAILURE) {
        /* retry or abort */
    }
    my_ip = DHCP_ack.yiaddr;
    options = DHCP_ack.options;
}
else {
    my_ip = BOOTP_reply.yiaddr;
    options = BOOTP_reply.options;
}

configure_IP(my_ip,options.subnet_mask,options.router);

bootfile = 0;
if (BSDP_offer != 0) {
    bootfile = BSDP_offer.file; /* (Figure 2) */
}
else {
    BSDP_ack = 0;
    if (Do_BSDP_Protocol(&BSDP_ack) == FAILURE) {
        /* retry or abort */
    }
    bootfile = BSDP_ack.file;
}
load and execute(bootfile);
```

If the firmware client received a BSDP OFFER (Table 3), it must save it so that it can be accessed by the loaded operating system. The client then loads and executes the bootfile.

If the client selected a DHCP OFFER, it sends a DHCP REQUEST (Table 4) packet to transition to the DHCP REQUEST state. If the client selected a BOOTP REPLY, the client firmware must save the packet in memory so that the booted operating system can retrieve it.

4.2 DHCP REQUEST state

The DHCP REQUEST state logic is fully specified in RFC2131 and is not duplicated here. After the client has confirmed its IP address binding, it uses the logic described in RFC2131 to safely probe whether the IP address is in use using ARP. If the ARP probe test indicates that the IP address is in use, it sends a DHCP DECLINE to the server, and starts DHCP over again.

If the IP address is not in use, the client configures its IP parameters and continues with BSDP or downloads the appropriate boot file (see the previous sections).

The client firmware must save the DHCP ACK so that it can be accessed by the loaded operating system.

Table 3: BSDP OFFER packet

BSDP OFFER PACKET			
UDP Information			
Field		**Value**	**Description**
UDP destination port		68	BOOTP/DHCP client port
UDP source port		67	BOOTP/DHCP server port
DHCP FIELDS			
Field	**Length (bytes)**	**Value**	**Description**
`op`	1	2	2=BOOTREPLY
`htype`	1	?	from DISCOVER
`hlen`	1	?	from DISCOVER
`hops`	1	?	from DISCOVER
`xid`	4	?	from DISCOVER
`secs`	2	?	from DISCOVER
`flags`	2	?	from DISCOVER
`ciaddr`	4	0.0.0.0	
`yiaddr`	4	0.0.0.0	client has no address
`siaddr`	4	x.x.x.x	server's IP address
`giaddr`	4	?	from DISCOVER
`chaddr`	16	?	from DISCOVER
`sname`	64	?	server 's host name, or option data if Option Overload is set to 2 or 3 (see below)
`file`	128	?	boot file name, or option data if Option Overload is set to 1 or 3 (see below)
`magic`	4	99.130.83.99	RFC 2132 defined magic number
DHCP OPTIONS			
Tag	**Code**	**Length (bytes)**	**Description**
Message Type	53	1	2=OFFER
Server Identifier	54	4	IP address of this DHCP server
Vendor Class Identifier	60	9	value is AAPLBSDPC ; see section 3.4.2
Vendor Specific Information	43	variable	see BSDP options below and section 3.4.4
Option Overload	52	1	optional; value 1, 2 or 3: see RFC 2132 section 9.3
TFTP server name	66	variable	optional; server's host name if `sname` is overloaded
Bootfile name	67	variable	optional; boot file name if `file` is overloaded
BSDP OPTIONS (encapsulated in option 43)			
Tag	**Code**	**Length (bytes)**	**Description**
BSDP Selected Boot Image ID	8	4	see section 3.4.12
Other boot image specific options	?	?	boot image specific options are encapsulated here as well

Table 4: DHCP REQUEST packet

DHCP REQUEST PACKET			
IP and UDP Information			
Field		**Value**	**Description**
IP destination address		255.255.255.255	IP broadcast address
IP source address		0.0.0.0	no IP address yet
UDP destination port		67	BOOTP/DHCP server port
UDP source port		68	BOOTP/DHCP client port
DHCP FIELDS			
Field	**Length (bytes)**	**Value**	**Description**
`op`	1	1	1=BOOTREQUEST
`htype`	1	?	1=10Mb Ethernet
`hlen`	1	?	6=10Mb Ethernet address length
`hops`	1	0	
`xid`	4	?	
`secs`	2	?	
`flags`	2	?	
`ciaddr`	4	0.0.0.0	
`yiaddr`	4	0.0.0.0	
`siaddr`	4	0.0.0.0	
`giaddr`	4	0.0.0.0	
`chaddr`	16	xx:xx:xx:xx:xx:xx	Client's thernet address
`sname`	64	0:0:…:0	unused
`file`	128	0:0:…:0	unused
`magic`	4	99.130.83.99	RFC 2132 magic number
DHCP OPTIONS			
Tag	**Code**	**Length (bytes)**	**Description**
Message Type	53	1	3=REQUEST
Requested IP Address	50	4	`yiaddr` from OFFER
Server Identifier	54	4	from Server Identifier in OFFER
Vendor Class Identifier	60	variable	see section 3.4.2
Vendor Specific Information	43	variable	see below and section 3.4.4
Parameter Request List	55	>= 4	Contains (at least) the bytes {1, 3, 43, 60}
IP Address Lease Time	51	4	optional; client can suggest the lease length
Maximum DHCP Message Size	57	2	optional; used if the client can handle a packet larger than 576 bytes (the DHCP minimum)
BSDP OPTIONS (encapsulated in option 43)			
Tag	**Code**	**Length (bytes)**	**Description**
BSDP Version	2	2	see section 3.4.6
BSDP Image Attributes Filter	11	2 (multiple of)	optional, see section 3.4.15

4.3 BSDP Client States

The BSDP protocol uses client state transitions similar to the first four defined in DHCP: INIT, SELECT, REQUEST and BOUND. Unlike DHCP, once in the BOUND state, the client does not need to contact the server again to maintain the binding, it may use the resources as long as it wants. The server may use an aging mechanism to free boot image resources if they are not used for some period of time, but must do so in a way that will not adversely affect existing booted clients.

Figure 4 below illustrates the BSDP client state transitions. The sections that follow describe each of the BSDP client states.

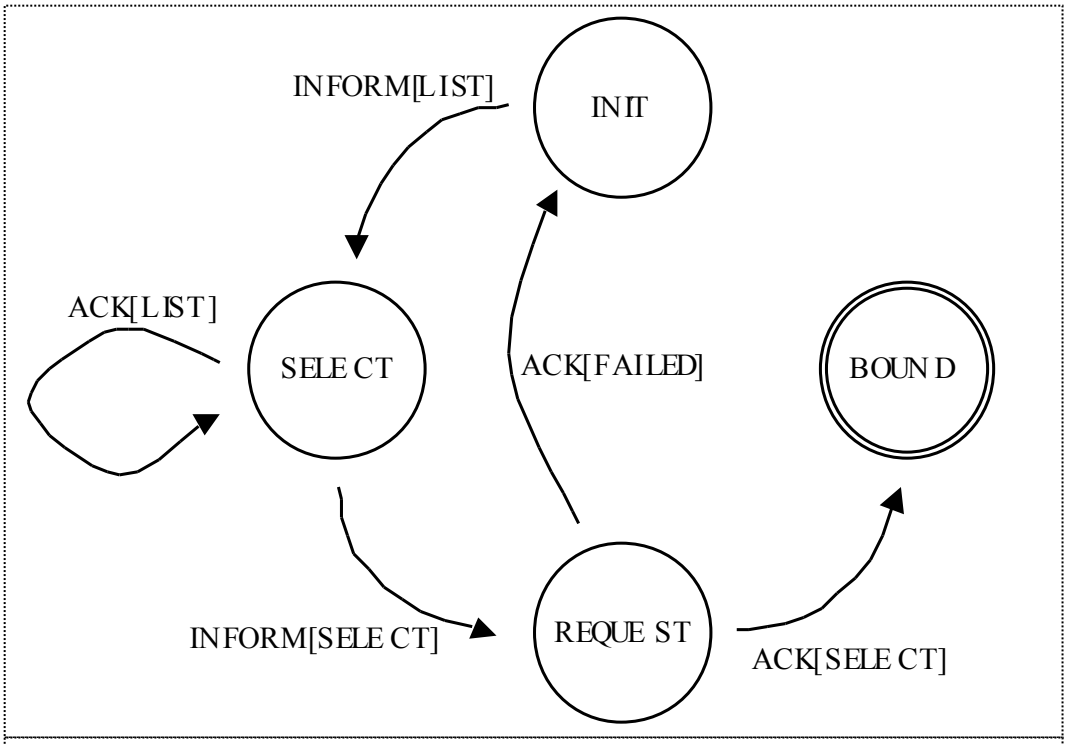


Figure 4: BSDP client state transitions

4.4 BSDP INIT state

After determining its IP address (see the previous sections), the client begins in the BSDP INIT state. It broadcasts an INFORM[LIST] packet (see Table 5 below) and sets a timer. If no responses arrive before the timer expires, the client retransmits the INFORM[LIST] (updating the `xid` and `secs` fields as usual). The timeout values are doubled each time and a small amount of random fuzz is added to avoid synchronized broadcasts among several clients.

The client retries several times. If no packets arrive, the client aborts network boot, and sends a DHCP RELEASE packet to the DHCP server indicating that it no longer requires the IP address (see RFC 2131).

Once the first ACK[LIST] packet arrives (see Table 6), the client transitions to the BSDP SELECT state.

Table 5: BSDP INFORM[LIST] packet

BSDP INFORM[LIST] PACKET			
IP and UDP Information			
Field		**Value**	**Description**
IP destination address		255.255.255.255	IP broadcast address
IP source address		x.x.x.x	client's IP address
UDP destination port		67	BOOTP/DHCP server port
UDP source port		68	BOOTP/DHCP client port
DHCP FIELDS			
Field	**Length (bytes)**	**Value**	**Description**
op	1	1	1=BOOTREQUEST
htype	1	?	1=10Mb Ethernet
hlen	1	?	6=10Mb Ethernet address length
hops	1	0	
xid	4	?	
secs	2	?	
flags	2	?	
ciaddr	4	x.x.x.x	client's IP address
yiaddr	4	0.0.0.0	
siaddr	4	0.0.0.0	
giaddr	4	0.0.0.0	
chaddr	16	xx:xx:xx:xx:xx:xx	client's hardware (thernet) address
sname	64	0:0:…:0	unused
file	128	0:0:…:0	unused
magic	4	99.130.83.99	RFC 2132 magic number
DHCP OPTIONS			
Tag	**Code**	**Length (bytes)**	**Description**
Message Type	53	1	8=INFORM
Vendor Class Identifier	60	variable	see section 3.4.2
Vendor Specific Information	43	variable	see below and section 3.4.4
Parameter Request List	55	>= 2	Contains (at least) the bytes {43, 60}
Maximum DHCP Message Size	57	2	optional; used if the client can handle a packet larger than 576 bytes (the DHCP minimum)
BSDP OPTIONS (encapsulated in option 43)			
Tag	**Code**	**Length (bytes)**	**Description**
BSDP Message Type	1	1	1=LIST; see section 3.4.5
BSDP Version	2	2	see section 3.4.6
Reply Port	5	2	optional; see section 3.4.9
BSDP NetBoot 1.0 Firmware	10	1	Only supplied in some circumstances, see section 3.4.14
BSDP Image Attributes Filter	11	2 (multiple of)	optional, see section 3.4.15

Table 6: BSDP ACK[LIST] packet

BSDP ACK[LIST] PACKET			
UDP Information			
Field		**Value**	**Description**
UDP destination port		?	BSDP Reply Port option from INFORM[LIST] if present; 68 otherwise
UDP source port		67	BOOTP/DHCP server port
DHCP FIELDS			
Field	**Length (bytes)**	**Value**	**Description**
`op`	1	2	2=BOOTREPLY
`htype`	1	?	from INFORM[LIST]
`hlen`	1	?	from INFORM[LIST]
`hops`	1	?	from INFORM[LIST]
`xid`	4	?	from INFORM[LIST]
`secs`	2	?	from INFORM[LIST]
`flags`	2	?	from INFORM[LIST]
`ciaddr`	4	x.x.x.x	from INFORM[LIST]
`yiaddr`	4	0.0.0.0	
`siaddr`	4	x.x.x.x	server's IP address
`giaddr`	4	?	from INFORM[LIST]
`chaddr`	16	?	from INFORM[LIST]
`sname`	64	?	server 's host name, or option data if Option Overload is set to 2 or 3 (see below)
`file`	128	?	empty, or option data if Option Overload is set to 1 or 3 (see below)
`magic`	4	99.130.83.99	RFC 2132 defined magic number
DHCP OPTIONS			
Tag	**Code**	**Length (bytes)**	**Description**
Message Type	53	1	5=ACK
Server Identifier	54	4	IP address of this DHCP server
Vendor Class Identifier	60	9	value `AAPLBSDPC`; see section 3.4.2
Vendor Specific Information	43	variable	see BSDP options below and section 3.4.4
Option Overload	52	1	optional; value 1, 2 or 3: see RFC 2132 section 9.3
BSDP OPTIONS (encapsulated in option 43)			
Tag	**Code**	**Length (bytes)**	**Description**
BSDP Message Type	1	1	1=LIST; see section 3.4.5
BSDP Server Priority	4	2	see section 3.4.8
BSDP Default Boot Image ID	7	4	see section 3.4.11
BSDP Selected Boot Image ID	8	4	optional; see section 3.4.12
BSDP Boot Image List	9	variable	see section 3.4.13

4.5 BSDP SELECT state

Once the client receives its first ACK[LIST] response, the client cancels the previous timer, sets a new "gathering" timer[3] and continues to wait for responses. The action that the client takes next depends on whether it is able to present a menu of choices to the user.

If the client is able to offer a menu selection, it should gather all the responses it received and build a single list of boot images. Once the user selects an image, the client records the `boot_image_id` value for that image, and remembers which ACK[LIST] the image appeared in. If the client is unable to offer a menu selection, it selects from the responses automatically. If an ACK[LIST] contains the BSDP Selected Boot Image ID option, the client selects that response and records the `boot_image_id` value from the option. Otherwise, the client selects the ACK[LIST] with the highest BSDP Server Priority option value and records the `boot_image_id` value from the BSDP Default Boot Image ID option.

Once an image is selected, the client forms an INFORM[SELECT] packet, setting the BSDP Server Identifier option to the value of the DHCP Server Identifier option from the selected ACK[LIST], and setting the BSDP Selected Boot Image ID option to the recorded `boot_image_id` value. The client broadcasts the INFORM[SELECT] and moves to the BSDP REQUEST state.

BSDP INFORM[SELECT] PACKET			
IP and UDP Information			
Field		**Value**	**Description**
IP destination address		255.255.255.255	IP broadcast address
IP source address		x.x.x.x	client's IP address
UDP destination port		67	BOOTP/DHCP server port
UDP source port		68	BOOTP/DHCP client port
DHCP FIELDS			
Field	**Length (bytes)**	**Value**	**Description**
`op`	1	1	1=BOOTREQUEST
`htype`	1	?	1=10Mb Ethernet
`hlen`	1	?	6=10Mb Ethernet address length
`hops`	1	0	
`xid`	4	?	
`secs`	2	?	
`flags`	2	?	
`ciaddr`	4	x.x.x.x	client's IP address
`yiaddr`	4	0.0.0.0	
`siaddr`	4	0.0.0.0	
`giaddr`	4	0.0.0.0	
`chaddr`	16	xx:xx:xx:xx:xx:xx	client's hardware (ethernet) address
`sname`	64	0:0:…:0	unused
`file`	128	0:0:…:0	unused

[3] A reasonable length for the gathering time is 2 to 4 seconds.

magic	4	99.130.83.99	RFC 2132 magic number

DHCP OPTIONS			
Tag	**Code**	**Length (bytes)**	**Description**
Message Type	53	1	8=INFORM
Vendor Class Identifier	60	variable	see section 3.4.2
Vendor Specific Information	43	variable	see below and section 3.4.4
Parameter Request List	55	>= 2	Contains (at least) the bytes {43, 60}
Maximum DHCP Message Size	57	2	optional; used if the client can handle a packet larger than 576 bytes (the DHCP minimum)

BSDP OPTIONS (encapsulated in option 43)			
Tag	**Code**	**Length (bytes)**	**Description**
BSDP Message Type	1	1	2=SELECT; see section 3.4.5
BSDP Version	2	2	see section 3.4.6
BSDP Server Identifier	3	4	from the DHCP Server Identifier option in ACK[LIST]; see section 0
BSDP Reply Port	5	2	optional; see section 3.4.9
BSDP Selected Boot Image ID	8	4	the `boot_image_id` selected by the client; see section 3.4.12
BSDP NetBoot 1.0 Firmware	10	1	Only supplied in some circumstances, see section 3.4.14

4.6 BSDP REQUEST state

In this state, the client is waiting for an ACK[SELECT] response (see Table 7 below), or if an error occurred, an ACK[FAILED] response (see Table 8 below). If an ACK[FAILED] response appears, the client MUST wait for at least 10 seconds before going back to the BSDP INIT state: the client must avoid looping quickly between the INIT, SELECT, REQUEST states in the case of a misbehaving BSDP server that always returns an ACK[FAILED]. If multiple BSDP server's respond, the client could choose a different server the next time around if one server returns the ACK[FAILED] message more than once. The client may choose to abort the network boot procedure after attempting BSDP a few times.

Once the client receives the ACK[SELECT], it saves it in a location accessible by the loaded operating system, and begins downloading the boot file.

Table 7: BSDP ACK[SELECT] packet

BSDP ACK[SELECT] PACKET			
UDP Information			
Field		**Value**	**Description**
UDP destination port		?	BSDP Reply Port option from INFORM[SELECT] if present; 68 otherwise
UDP source port		67	BOOTP/DHCP server port
DHCP FIELDS			
Field	**Length (bytes)**	**Value**	**Description**
op	1	2	2=BOOTREPLY
htype	1	?	from INFORM[SELECT]
hlen	1	?	from INFORM[SELECT]

hops	1	?	from INFORM[SELECT]
xid	4	?	from INFORM[SELECT]
secs	2	?	from INFORM[SELECT]
flags	2	?	from INFORM[SELECT]
ciaddr	4	x.x.x.x	from INFORM[SELECT]
yiaddr	4	0.0.0.0	
siaddr	4	x.x.x.x	server's IP address
giaddr	4	?	from INFORM[SELECT]
chaddr	16	?	from INFORM[SELECT]
sname	64	?	server 's host name, or option data if Option Overload is set to 2 or 3 (see below)
file	128	?	boot file name, or option data if Option Overload is set to 1 or 3 (see below)
magic	4	99.130.83.99	RFC 2132 defined magic number

DHCP OPTIONS			
Tag	**Code**	**Length (bytes)**	**Description**
Message Type	53	1	5=ACK
Server Identifier	54	4	IP address of this DHCP server
Vendor Class Identifier	60	9	value AAPLBSDPC; see section 3.4.2
Vendor Specific Information	43	variable	see BSDP options below and section 3.4.4
Option Overload	52	1	optional; value 1, 2 or 3: see RFC 2132 section 9.3
TFTP server name	66	variable	optional; server's host name if sname is overloaded
Bootfile name	67	variable	optional; boot file name if file is overloaded

BSDP OPTIONS (encapsulated in option 43)			
Tag	**Code**	**Length (bytes)**	**Description**
BSDP Message Type	1	1	2=SELECT; see section 3.4.5
BSDP Selected Boot Image ID	8	4	see section 3.4.12
Other boot image specific options	?	?	boot image specific options are encapsulated here as well

Table 8: BSDP ACK[FAILED] packet

BSDP ACK[FAILED] PACKET		
UDP Information		
Field	**Value**	**Description**
UDP destination port	?	BSDP Reply Port option from INFORM[SELECT] if present; 68 otherwise
UDP source port	67	BOOTP/DHCP server port

DHCP FIELDS			
Field	**Length (bytes)**	**Value**	**Description**
op	1	2	2=BOOTREPLY
htype	1	?	from INFORM[SELECT]
hlen	1	?	from INFORM[SELECT]
hops	1	?	from INFORM[SELECT]
xid	4	?	from INFORM[SELECT]
secs	2	?	from INFORM[SELECT]
flags	2	?	from INFORM[SELECT]
ciaddr	4	x.x.x.x	from INFORM[SELECT]
yiaddr	4	0.0.0.0	

siaddr	4	x.x.x.x	server's IP address
giaddr	4	?	from INFORM[SELECT]
chaddr	16	?	from INFORM[SELECT]
sname	64	?	server 's host name
file	128	?	empty
magic	4	99.130.83.99	RFC 2132 defined magic number

DHCP OPTIONS			
Tag	**Code**	**Length (bytes)**	**Description**
Message Type	53	1	5=ACK
Server Identifier	54	4	IP address of this DHCP server
Vendor Class Identifier	60	9	value AAPLBSDPC; see section 3.4.2
Vendor Specific Information	43	variable	see BSDP options below and section 3.4.4

BSDP OPTIONS (encapsulated in option 43)			
Tag	Code	Length (bytes)	Description
BSDP Message Type	1	1	3=FAILED; see section 3.4.5

5 BSDP Non-firmware Client

A non-firmware client such as the Startup Disk panel in Mac OS 9 and X can use BSDP to select an image that the system will use the next time the system boots over the network. This is possible for two reasons:

1. The NetBoot server remembers the binding that a client negotiates.
2. Image allocation is a separate operation from IP allocation and uses the client's hardware address as the identifier.

The client may choose to specify an alternate port to use in the reply from the server (see 3.4.9) to avoid conflicts with the DHCP client. The protocol proceeds as described starting in section 4.3.

5.1 Pre-BSDP firmware limitations

The discussion up to this point details BSDP with the assumption that the firmware is BSDP capable. BSDP is a relatively recent development, so systems that pre-date it have either no NetBoot capability, or limited capability.

The reason this is important only to the non-firmware BSDP client is that unlike the actual firmware, it can run on any system, regardless of its firmware version. It must adjust its behavior to match the capabilities of the underlying firmware to avoid surprising the user the next time the system reboots.

Apple systems developed before the original iMac are not able to NetBoot. The non-firmware BSDP client must prevent the user from negotiating BSDP on such systems. Failure to do so means the user is left with a system that will fail to boot.

Apple systems that pre-date BSDP, such as the original iMac and blue and white PowerMac G3 (Yosemite) have an Open Firmware version predating BSDP. These systems use an enhanced version of BOOTP now referred to as NetBoot 1.0. Such

systems cannot fully benefit from the features of BSDP, and are limited to being served by a single NetBoot server.

On NetBoot 1.0 systems, the non-firmware BSDP client must identify itself as a NetBoot 1.0 client in its BSDP messages by supplying the BSDP NetBoot 1.0 Firmware option (see section 3.4.14 for more information).

5.2 How to identify non-NetBoot, NetBoot 1.0, and NetBoot 2.0 firmware

The Open Firmware version is identified using the device tree node "`device-tree:/rom/boot-rom`".

On non-NetBoot capable machines, this node will not exist.

On NetBoot-capable machines, read the third long-word (4-byte) quantity. If the value is less than 0x33000, the client is NetBoot 1.0. Otherwise it is NetBoot 2.0.

The following code fragment describes the logic:

```
int
NetBootVersion()
{
    void *         info;
    DeviceTreeNode node;
    PropertyList   plist;
    int            client_type;

    info = 0
    node = DeviceTreeNodeLookup("device-tree:/rom/boot-rom");
    if (node != NULL) {
        plist = DeviceTreeNodeGetProperties(node, plist);
        info = PropertyListGetValue(plist, "info");
    }
    if (info == NULL) {
        client_type = NETBOOT_VERSION_NONE; /* no NetBoot support */
    else {
        uint32_t third_longword;

        third_longword = *(uint32_t *)(info + 8);
        if (ntohl(third_longword) < 0x33000)
            client_type = NETBOOT_VERSION_1; /* NetBoot 1.0 */
        else
            client_type = NETBOOT_VERSION_2; /* NetBoot 2.0 */
    }
    return (client_type);
}
```