DHC
Internet-Draft
Intended status: Standards Track
Expires: February 11, 2010

T. Huth
J. Freimann
IBM Germany Research &
Development GmbH
V. Zimmer
Intel
D. Thaler
Microsoft

August 10, 2009

DHCPv6 option for network boot
draft-ietf-dhc-dhcpv6-opt-netboot-05

Status of this Memo

Copyright Notice

Abstract

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) provides a framework for passing configuration information to nodes on a network. This document describes new options for DHCPv6 which are required for booting a node from the network.

Table of Contents

1. Introduction

This draft describes DHCPv6 options that can be used to provide configuration information for a node that must be booted using the network, rather than from local storage.

Network booting is used, for example, in some environments where administrators have to maintain a large number of nodes. By serving all boot and configuration files from central server, the effort required to maintain these nodes is greatly reduced.

A typical boot file would be, for example, an operating system kernel or a boot loader program. To be able to execute such a file, the firmware (BIOS) running on the client node must perform the following two steps (see Figure 1): First get all information which is required for downloading and executing the boot file. Second, download the boot file and execute it.

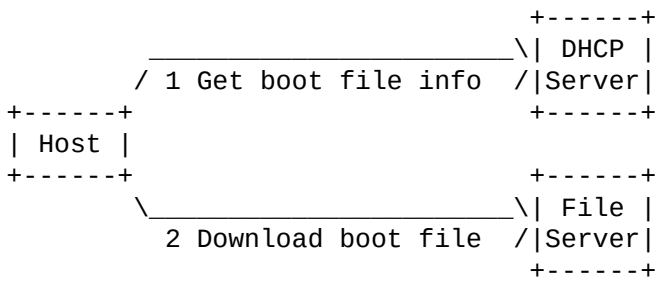


Figure 1: Network Boot Sequence

Information that is required for booting over the network can include information about the server on which the boot files can be found, the protocol to be used for the download (for example HTTP [RFC2616] or TFTP [RFC1350]), the name of the boot file and additional parameters which should be passed to the OS kernel or boot loader program respectively.

DHCPv6 allows client nodes to ask a DHCPv6 server for configuration parameters. This document provides new options which a client can request from the DHCPv6 server to satisfy its requirements for booting.

2. Conventions

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].

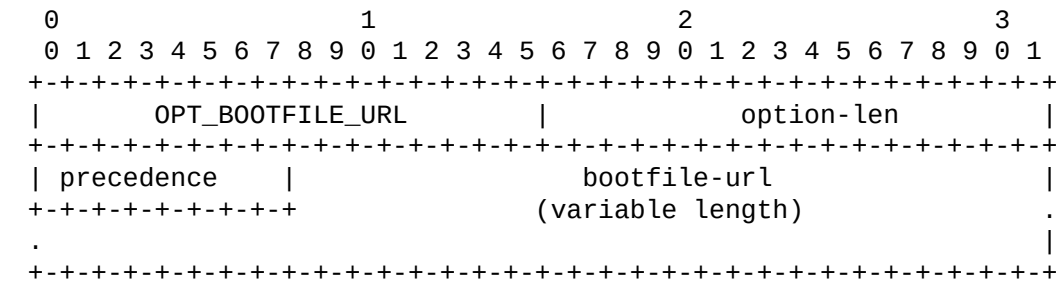
Terminology specific to IPv6 and DHCPv6 are used in the same way as defined in the "Terminology" sections of RFC 3315 [RFC3315].

3. Options

Option formats comply with DHCPv6 options per [RFC3315] (section 6).

3.1. Boot File Uniform Resource Locator (URL) Option

This option consists of an US-ASCII string. It is used to convey an URL to a boot file.



Format description:

option-code OPT_BOOTFILE_URL (TBD1).

option-len Length of the bootfile URL option in octets (not including the size of the option-code and option-len fields).

precedence A single unsigned octet indicating the order in which this URL should be processed, if more than one URL appears in the message.

bootfile-url This US-ASCII string is the URL for the boot file, as defined in [RFC3986]. The string is not NUL-terminated.

The node identifier in the URL must be reachable using IPv6. If the URL is expressed using an IPv6 address rather than a domain name, the address in the URL then MUST be enclosed in "[" and "]" characters, conforming to [RFC3986]. Clients that have DNS implementations should support the use of domain names in the URL.

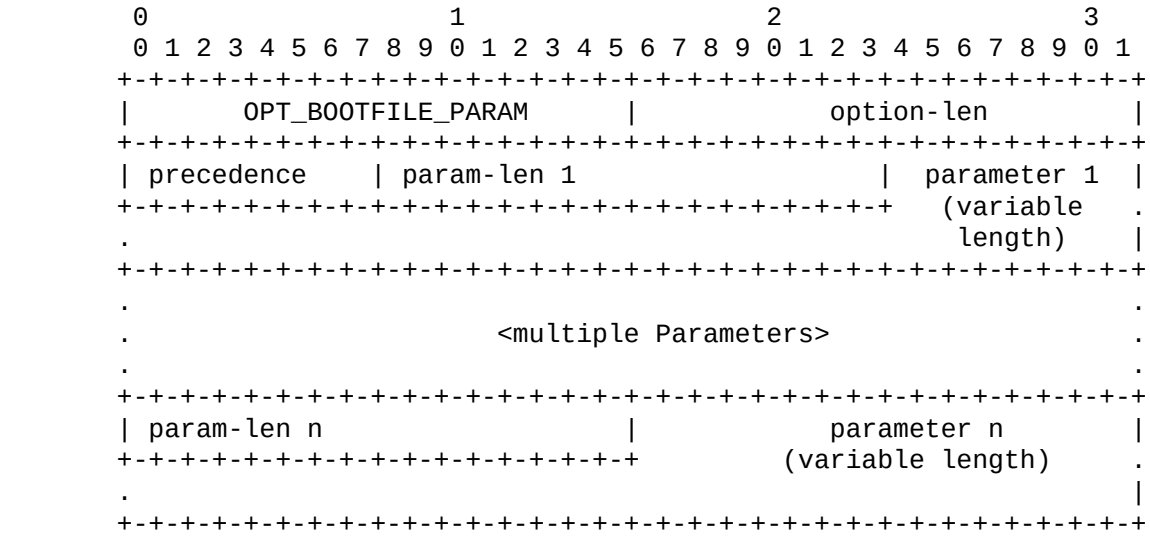
Multiple occurrences of OPT_BOOTFILE_URL MAY be present in a single DHCP message. Clients MUST process them according to the value of the precedence field - the lowest precedence should be processed first. If this fails, then the second-lowest should be used, and so on.

Servers SHOULD NOT send two Bootfile URL options with the same precedence. Clients receiving more than one OPT_BOOTFILE_URL option with the same precedence SHOULD discard any extra such options. The order in which the client processes options is not specified, and therefore server implementations cannot assume that the client will discard a particular such option.

The value of the precedence field MUST NOT be zero.

3.2. Boot File Parameters Option

This option consists of multiple US-ASCII strings. They are used to specify parameters for the boot file (e.g. parameters for the kernel or boot loader program).



Format description:

option-code OPT_BOOTFILE_PARAM (TBD2).

option-len Length of the bootfile parameters option in octets (not including the size of the option-code and option-len fields).

precedence	A one-octet quantity indicating the bootfile-url option to which this set of parameters applies.
param-len 1...n	This is a 16-bit integer which specifies the length of the following parameter in octets (not including the parameter-length field).
parameters 1...n	These US-ASCII strings are parameters needed for booting, e.g. kernel parameters. The strings are not NUL-terminated.

The firmware MUST pass these parameters in the order they appear in the OPT_BOOTFILE_PARAM option to the boot file which has been specified in the OPT_BOOTFILE_URL option.

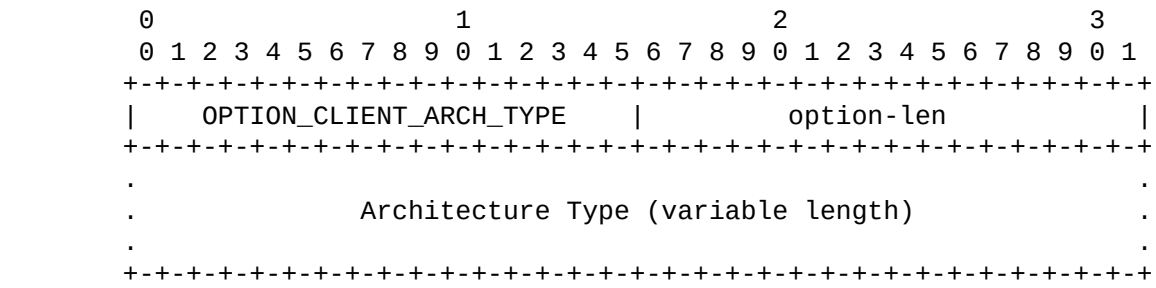
Multiple occurrences of OPT_BOOTFILE_PARAM MAY be present in a single DHCP message. Clients MUST process them according to the value of the precedence field:

- If the precedence field of the Bootfile Parameters option is zero, the client SHOULD provide these parameters when it attempts to execute any Bootfile it has loaded using any of the provided Bootfile URL options.
- If the precedence field of the Bootfile Parameters option is nonzero, the client SHOULD provide these parameters only when it attempts to execute a Bootfile it loaded using a Bootfile URL option with a precedence field that has the same value.
- In the event that the client receives both a Bootfile Parameters option with a precedence field of zero and one with a precedence field that matches a certain Bootfile URL option, the client MUST use the Bootfile Parameters option whose predence matches the precedence of the Bootfile URL option.

3.3. Client System Architecture Type Option

This option provides parity with the Client System Architecture Type Option defined for DHCPv4 in [RFC4578] section 2.1.

The format of the option is:



option-code OPTION_CLIENT_ARCH_TYPE (TBD3).

option-len Length of the "processor architecture type" field in octets (not including the option-code and option-len fields). It MUST be an even number greater than zero. See [RFC4578] section 2.1 for details.

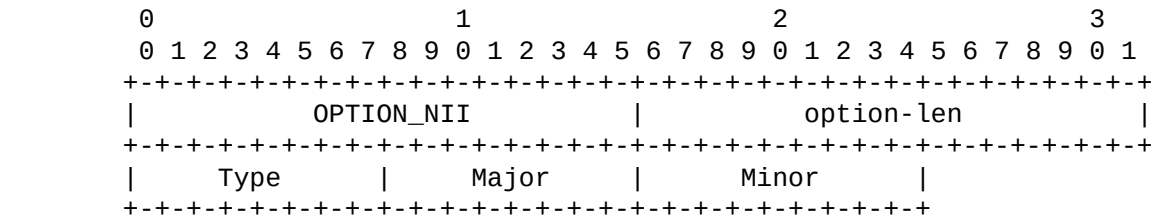
Architecture Type A list of one or more architecture types, as specified in [RFC4578] section 2.1.

3.4. Client Network Interface Identifier Option

The Client Network Interface Identifier option is sent by a DHCP client to a DHCP server to provide information about its level of Universal Network Device Interface (UNDI) support (see also [PXE21] and [UEFI22]).

This option provides parity with the Client Network Interface Identifier Option defined for DHCPv4 in [RFC4578] section 2.2.

The format of the option is:



option-code OPTION_NII (TBD4).

option-len 3

Type As specified in [RFC4578] section 2.2.

Major As specified in [RFC4578] section 2.2.

Minor As specified in [RFC4578] section 2.2.

4. Appearance of the options

These options MUST NOT appear in DHCPv6 messages other than the types Solicit, Advertise, Request, Renew, Rebind, Information-Request and Reply.

The option-codes of these options MAY appear in the Option Request Option in the DHCPv6 message types Solicit, Request, Renew, Rebind, Information-Request and Reconfigure.

5. Download protocol considerations

The Bootfile URL option does not place any constraints on the protocol used for downloading the Bootfile, other than that it must be possible to specify it in a URL. For the sake of administrative simplicity, we strongly recommend that, at a mininum, implementors of network boot loaders implement the well-known and established hypertext transfer protocol (HTTP, see [RFC2616]) for downloading. Please note that for IPv6, this supersedes [RFC906] which recommended to use TFTP for downloading (see [RFC3617] for TFTP URL definition).

When using iSCSI for booting, the "iscsi:"-URI is formed as defined in [RFC4173]. The functionality attributed in RFC4173 to a root path option is provided for IPv6 by the bootfile URL option instead.

6. IANA considerations

The following options need to be assigned by the IANA from the option number space defined in the chapter 22 of the DHCPv6 RFC [RFC3315].

Option name	Value	Specified in
OPT_BOOTFILE_URL	TBD1	Section 3.1
OPT_BOOTFILE_PARAM	TBD2	Section 3.2
OPTION_CLIENT_ARCH_TYPE	TBD3	Section 3.3
OPTION_NII	TBD4	Section 3.4

This document also introduces a new IANA registry for processor architecture types. The name of this registry shall be "Processor Architecture Type". Registry entries consist of a 16-bit integer recorded in decimal format, and a descriptive name. The initial values of this registry can be found in [RFC4578] section 2.1.

The assignment policy for values shall be Expert Review (see [RFC5226]), and any requests for values must supply the descriptive name for the processor architecture type.

7. Security considerations

In untrusted networks, a rogue DHCPv6 server could send the new DHCPv6 options described in this document. The booting clients could then be provided with a wrong URL so that the boot either fails, or even worse, the client boots the wrong operating system which has been provided by a malicious file server. To prevent this kind of attack, clients can use authentication of DHCPv6 messages (see chapter 21. in [RFC3315]).

Note also that DHCPv6 messages are sent unencrypted by default. So the boot file URL options are sent unencrypted over the network, too. This can become a security risk since the URLs can contain sensitive information like user names and passwords (for example a URL like "ftp://username:password@servername/path/file"). At the current point in time, there is no possibility to send encrypted DHCPv6 messages, so it is strongly recommended not to use sensitive information in the URLs in untrusted networks.

Even if the DHCPv6 transaction is secured, this does not protect against attacks on the bootfile download channel. Consequently, we recommend that either a protocol like HTTPS (see [RFC2817] and [RFC2818]) be used to prevent spoofing, or that the boot loader implementation implement a mechanism for signing boot images and a configurable signing key in memory, so that if a malicious image is provided, it can be detected and rejected.

8. Acknowledgements

The authors would like to thank Ruth Li, Dong Wei, Kathryn Hampton, Phil Dorah, Richard Chan, and Fiona Jensen for discussions that led to this document.

The authors would also like to thank Ketan P. Pancholi, Alfred Hoenes, Gabriel Montenegro and Ted Lemon for corrections and suggestions.

9. References

9.1. Normative References

[PXE21] Johnston, M., "Preboot Execution Environment (PXE) Specification", September 1999, <http://www.pix.net/software/pxeboot/archive/pxespec.pdf>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4173] Sarkar, P., Missimer, D., and C. Sapuntzakis, "Bootstrapping Clients using the Internet Small Computer System Interface (iSCSI) Protocol", RFC 4173, September 2005.

[RFC4578] Johnston, M. and S. Venaas, "Dynamic Host Configuration Protocol (DHCP) Options for the Intel Preboot eXecution Environment (PXE)", RFC 4578, November 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[UEFI22] UEFI Forum, "Unified Extensible Firmware Interface Specification, Version 2.2", September 2008, <http://www.uefi.org/>.

9.2. Informative References

[RFC1350] Sollins, K., "The TFTP Protocol (Revision 2)", STD 33, RFC 1350, July 1992.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2817] Khare, R. and S. Lawrence, "Upgrading to TLS Within HTTP/1.1", RFC 2817, May 2000.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3617] Lear, E., "Uniform Resource Identifier (URI) Scheme and Applicability Statement for the Trivial File Transfer Protocol (TFTP)", RFC 3617, October 2003.

[RFC906] Finlayson, R., "Bootstrap Loading using TFTP", RFC 906, June 1984.

Authors' Addresses

Thomas H. Huth
IBM Germany Research & Development GmbH
Schoenaicher Strasse 220
Boeblingen 71032
Germany

Phone: +49-7031-16-2183
Email: thuth@de.ibm.com

Jens T. Freimann
IBM Germany Research & Development GmbH
Schoenaicher Strasse 220
Boeblingen 71032
Germany

Phone: +49-7031-16-1122
Email: jfrei@de.ibm.com

Vincent Zimmer
Intel
2800 Center Drive
DuPont WA 98327
USA

Phone: +1 253 371 5667
Email: vincent.zimmer@intel.com

Dave Thaler
Microsoft
One Microsoft Way
Redmond WA 98052
USA

Phone: +1 425 703-8835
Email: dthaler@microsoft.com