DHC
Internet-Draft
Intended status: Standards Track
Expires: October 16, 2009

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

April 14, 2009

DHCPv6 option for network boot
draft-ietf-dhc-dhcpv6-opt-netboot-04

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

Network booting means that a node which should be booted fetches the files required for booting via its network device from a server. Network booting is, for example, very useful in environments where the administrators have to maintain a large number of nodes. Since all boot and configuration files are stored on a central server, the maintenance of all nodes can be kept simple this way.

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 are required for downloading and executing the boot file such as: 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. As second step, download the boot file from the file server and execute it.

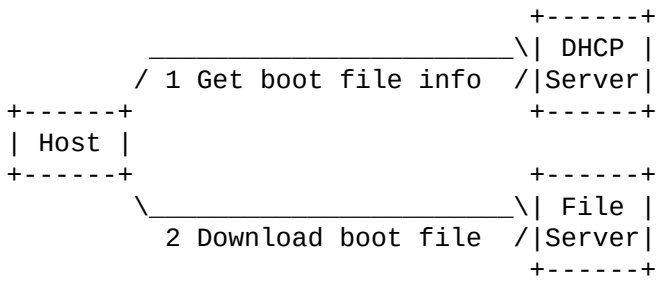


Figure 1: Network Boot Sequence

DHCPv6 allows client nodes to ask a DHCPv6 server for configuration parameters. Contrary to its IPv4 predecessor, DHCPv6 does not yet define a way to query network boot options such as the IPv6 address of a boot file server and boot file names. Therefore this document defines new DHCPv6 options which are required for network booting clients.

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].

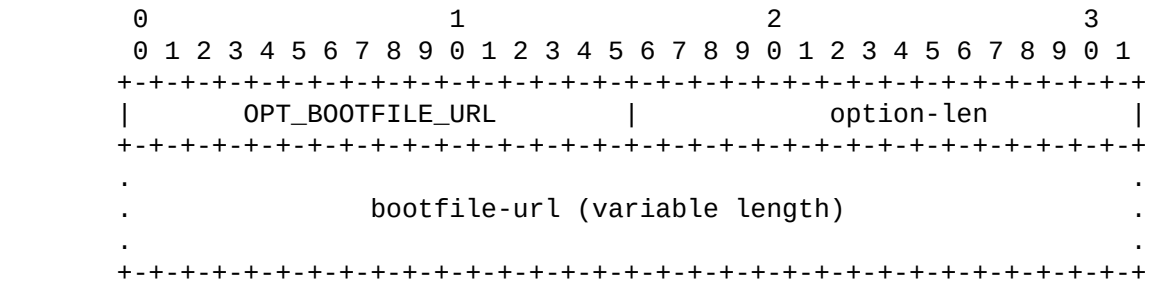
Terminology specific to IPv6 and DHCPv6 are used in the same way as defined in the "Terminology" sections of RFC 3315 [RFC3315].

3. Options

As specified in the DHCPv6 RFC [RFC3315], all values in the options are in network byte order. Options are byte-aligned but are not aligned in any other way such as on 2 or 4 byte boundaries. There is no padding between the options.

3.1. Boot File Uniform Resource Locator (URL) Option

This option consists of an 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).

bootfile-url This ASCII string is the URL (conforming to [RFC3986]) for a boot file. This string starts with the protocol which is used for downloading. Separated by "://", the hostname or IPv6 address of the server hosting the boot file follows, and then the path, file name and query parts of the URL. The string is not null-terminated.

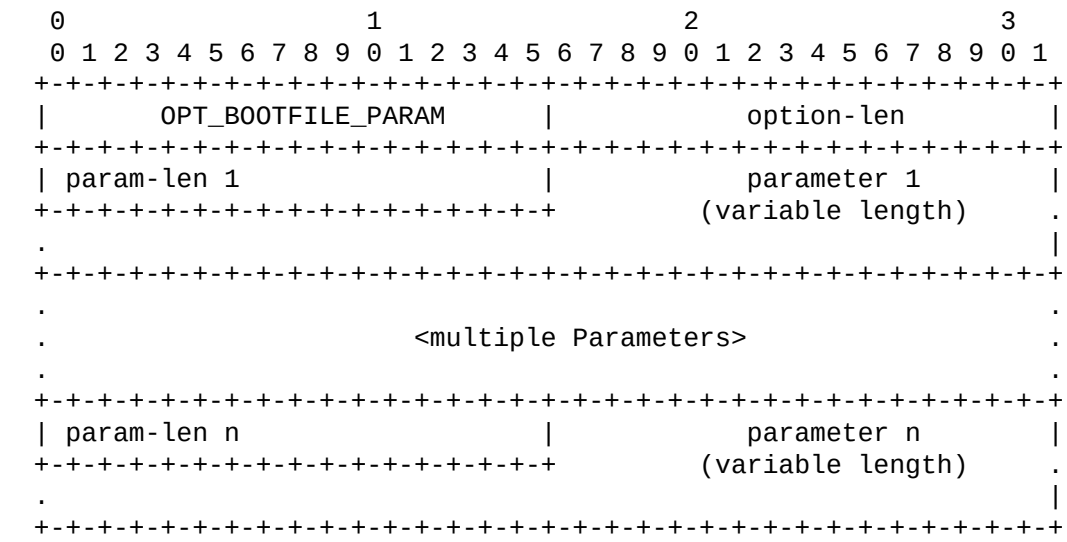
Note about the bootfile-url: This string can either contain a hostname or a literal IPv6 address to specify the server where the boot file should be downloaded from. All clients which implement the OPT_BOOTFILE_URL option MUST be able to handle IPv6 addresses here and SHOULD also be able to handle a hostname in the URL. The IPv6

address in the URL then MUST be enclosed in "[" and "]" characters, conforming to [RFC3986]. Clients SHOULD also be able to handle hostnames in the URLs. However, in this case the firmware implementation on the client machine must support DNS, too. Due to size limitations, this might not be possible in all firmware implementations, so support for hostnames in the URLs is only optional.

Multiple occurrences of OPT_BOOTFILE_URL can be present in a single DHCP message. Clients MUST process them in the order in which they appear within the message. The client starts with the first file that should be downloaded and executed. In case of a failure the process should continue with the second one and so on.

3.2. Boot File Parameters Option

This option consists of multiple 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).

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).

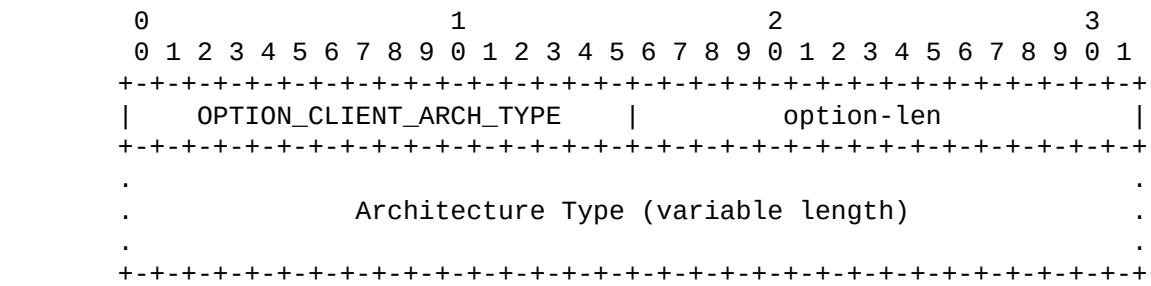
parameters 1...n These ASCII strings are parameters needed for booting, e.g. kernel parameters. The strings are not null-terminated.

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.

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.

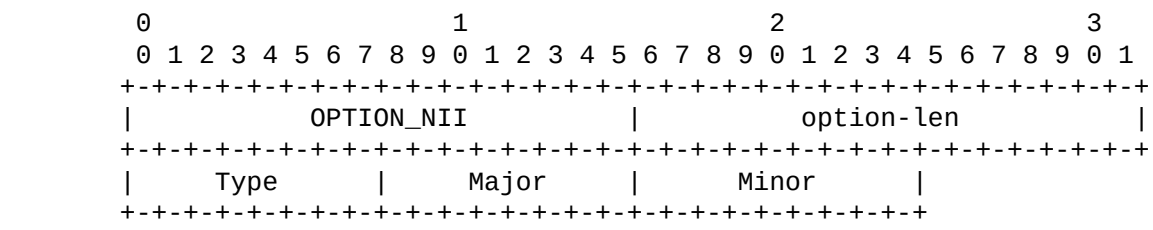
3.4. Client Network Interface Identifier Option

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

Depending on the network infrastructure, various special requirements could be imposed on the download protocol, so this document does not force one protocol for all scenarios. However, in case there there are no special requirements, the HTTP protocol SHOULD be used as download protocol.

RFC 906 [RFC906] suggested to use TFTP for bootstrap loading. Since TFTP is based on UDP, it has the advantage that it can also be used in firmware implementations which have to deal with size and complexity constraints and thus can not include a full-blown TCP/IP stack. It can also be used in multicast mode (see [RFC2090]) which is useful when a lot of nodes boot the same boot file at the same

time. So if TFTP should be used as download protocol, the boot file URLs then must be specified according to RFC 3617 [RFC3617].

However, TFTP also has some severe limitations, for example performance limitations due to acknowledging each packet and size limitations due to using only 16-bit packet counters. So this specification suggests to use now the well-known and established hypertext transfer protocol (HTTP, see [RFC2616]) as default for network booting instead. If a secure download is required, it is also possible to use HTTP with TLS (HTTPS, see [RFC2818]).

An alternative approach to network booting is to bootstrap the system with iSCSI. In this case, the URL in the OPT_BOOTFILE_URL option MUST be specified according to the "iscsi:" string definition in chapter 5 of [RFC4173]. Note that [RFC4173] also suggests that the "iscsi:" string should be specified in the so-called "Root Path" option. However, this option does not exist for DHCPv6 yet, and with the OPT_BOOTFILE_URL it is also not necessary anymore. So for IPv6 iSCSI booting, the "iscsi:" string MUST be specified as URL in the OPT_BOOTFILE_URL option instead.

If multiple interfaces are available for booting, it might be a good strategy to send out requests on each interface in parallel to speed up the discovery. However how to handle multiple replies, i.e. replies from more than one DHCP server is not a problem that can be easily solved on the protocol level. It is up to the implementors to provide users with a possibility to either choose a network interface to boot from, or to assign a preference to interfaces or even known DHCP servers.

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

The new DHCPv6 options described in this document could be sent in untrusted networks by malicious people with a fake DHCPv6 server to confuse the booting clients. The clients could 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 SHOULD 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.

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 and Alfred Hoenes 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.

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

[RFC2090] Emberson, A., "TFTP Multicast Option", RFC 2090, February 1997.

[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.

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

[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