                    DHCPv6 option for network boot
                  draft-ietf-dhc-dhcpv6-opt-netboot-01

Status of this Memo

   By submitting this Internet-Draft, each author represents that any
   applicable patent or other IPR claims of which he or she is aware
   have been or will be disclosed, and any of which he or she becomes
   aware will be disclosed, in accordance with Section 6 of BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as Internet-
   Drafts.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   The list of current Internet-Drafts can be accessed at
   http://www.ietf.org/ietf/1id-abstracts.txt.

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html.

   This Internet-Draft will expire on April 17, 2009.

Abstract

   The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) provides a
   framework for passing configuration information to nodes on a
   network. This document describes a new option for DHCPv6 to convey
   information, required for network booting, to the nodes.

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 download such a file, the
firmware (BIOS) running on the client node must be provided with
information such as: the server on which the boot files can be found,
the protocol to be used for the download (for example TFTP [RFC1350])
and the name of the boot file. Since some kernels or boot loaders
need to be provided with additional parameters, there should also be
the possibility to pass additional parameters along with the server
address, the protocol and the file name.

DHCPv6 allows client nodes to ask a DHCPv6 server for configuration
parameters. Contrary to its IPv4 predecessor, DHCPv6 does not 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 a new DHCPv6 option which is 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. Netboot option format

The netboot option is used as an encapsulation for suboptions which
carry the actual information needed to boot a client. This option
will be used by clients to request boot information from a server.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          OPT_NET_BOOT         |             option-len        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       suboption-code 1        |           suboption-len       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   subopt-data 1 (variable length)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                       <multiple suboptions>                  .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       suboption-code n        |           suboption-len       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   subopt-data n (variable length)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
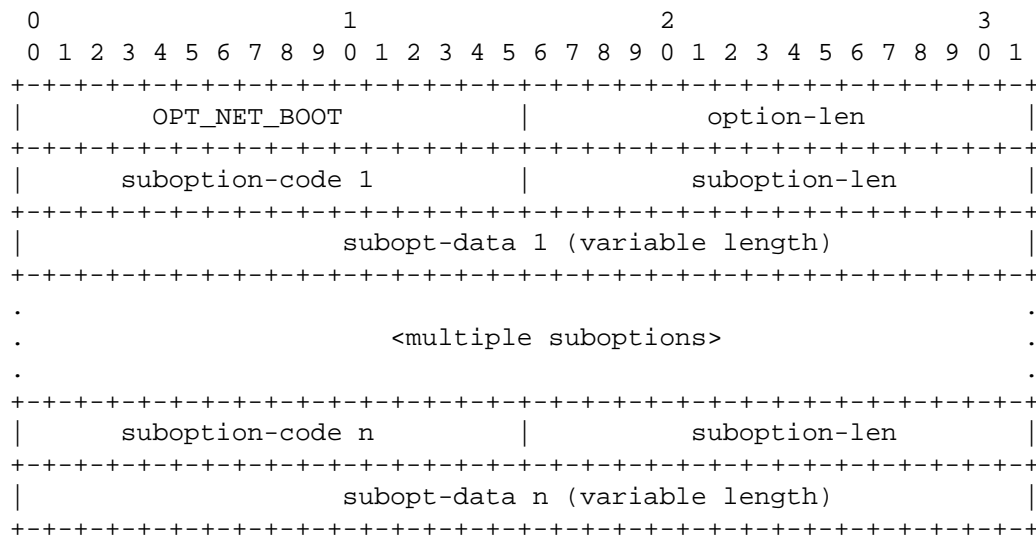
    option-code   OPT_NET_BOOT (tbd).

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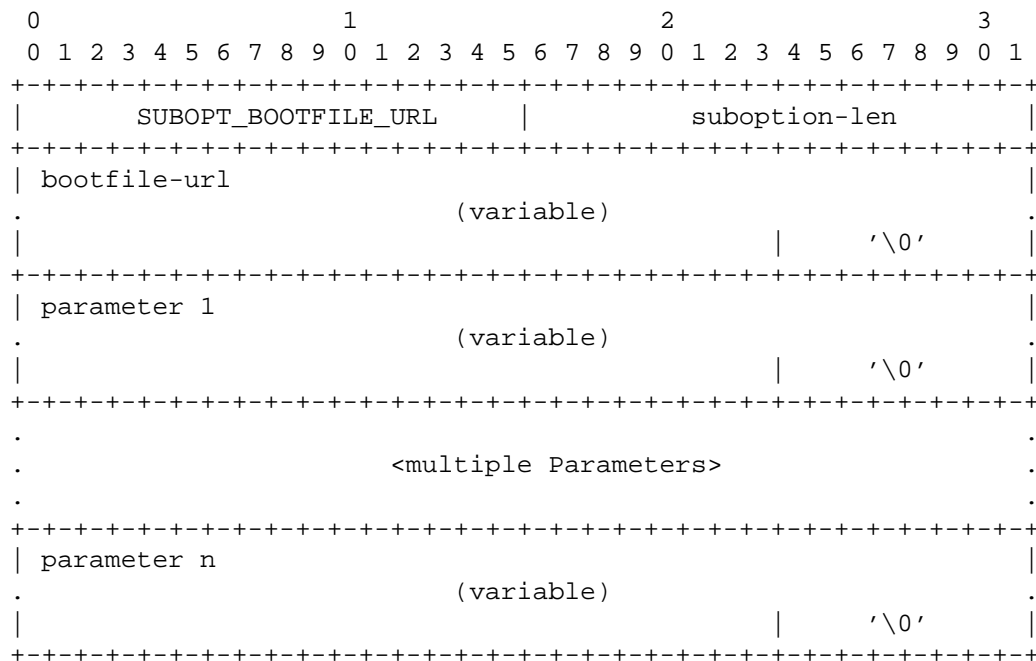   the size of the option-code and option-len fields).

   Multiple occurences of each suboption-type can occur within a netboot
   option (for example when more than one boot server is available).
   Clients MUST process the suboptions in the order in which they appear
   in the message sent by the server.

   So far, only the suboptions in the following chapters have been
   defined. Other suboptions might be defined in future RFCs.

4. Suboption: Boot file Uniform Resource Locator (URL)

   This suboption consists of multiple null-terminated strings. It is
   used to convey an URL to a boot file together with additional
   parameters for the boot file (e.g. parameters for the kernel or boot
   loader program).

   Since multiple occurrences of SUBOPT_BOOTFILE_URL can be present in a
   single OPT_NETBOOT message, clients MUST process them in the order in
   which they appear within the message. For example in the case of a
   boot file URL the first file should be downloaded and executed. In
   case of a failure the process should continue with the second one and
   so on.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |        SUBOPT_BOOTFILE_URL       |          suboption-len      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | bootfile-url                                                  |
    .                          (variable)                          .
    |                                              |     '\0'       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | parameter 1                                                  |
    .                          (variable)                          .
    |                                              |     '\0'       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                                                              .
    .                      <multiple Parameters>                   .
    .                                                              .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | parameter n                                                  |
    .                          (variable)                          .
    |                                              |     '\0'       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Format description:

suboption-code   SUBOPT_BOOTFILE_URL (tbd).

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

bootfile-url   This NULL-terminated ASCII string is the URL
                  (conforming to [RFC2396]) to 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
                  (see also the note below), the path, file name and
                  query parts of the URL follow.

parameters 1...n These NULL-terminated ASCII strings are parameters
                  needed for booting, e.g. kernel parameters. In
                  cases where no parameters are needed, everything
                  but the boot file URL can be omitted. Parameters
                  following the boot file name should be directly
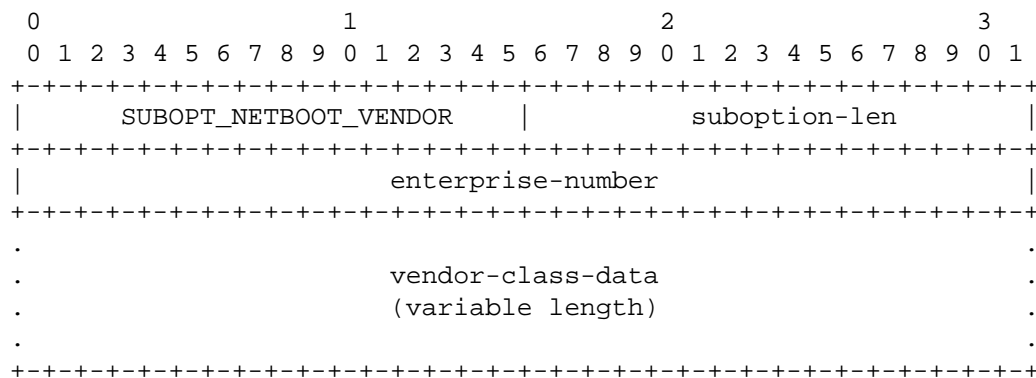   related to the boot file (kernel) itself.

   Note about the bootfile-url: This string can either contain a
   hostname or an IPv6 address to specify the server where the boot file
   should be downloaded from. All clients which implement support for

   the SUBOPT_BOOTFILE_URL suboption MUST be able to handle IPv6
   addresses here. The IPv6 address in the URL then MUST be enclosed in
   "[" and "]" characters, conforming to [RFC2732]. 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.

5. Suboption: Vendor class extension

   With this suboption, vendors can define their own netboot suboptions:
   It can be used by clients and servers to exchange vendor-specific
   information which is related to network booting.

   This suboption can occur multiple times within a OPT_NET_BOOT option
   (also with different enterprise-numbers in case a server and client
   implementation supports different vendor extensions). Clients MUST
   process them in the order in which they appear within the message.
   Unsupported vendor extensions MUST be ignored.


```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      SUBOPT_NETBOOT_VENDOR     |          suboption-len        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        enterprise-number                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   .                                                               .
   .                       vendor-class-data                       .
   .                       (variable length)                       .
   .                                                               .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Format description:

   suboption-code   SUBOPT_NETBOOT_VENDOR (tbd).

   suboption-len   Length of the vendor class suboption in octets
                       (not including the size of the suboption-code and
                       suboption-len fields).

   enterprise-number   The enterprise number of the vendor as registered
                       with IANA (see [VENDORIDS]).

   vendor-class-data Vendor-specific information. The meaning is
                      defined by the vendor identified by the
                      enterprise-number.

6. Appearance of these options

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

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

   The suboptions MUST appear only in the netboot option.

7. Boot protocol considerations

   RFC 906 [RFC906] suggests to use TFTP for bootstrap loading. Because
   it is easy to implement this protocol in firmware (where one has to
   deal with size and complexity constraints), this is still the
   recommended protocol for network booting. Every firmware
   implementation SHOULD at least support this protocol. The boot file
   URLs then must be specified according to RFC 3617 [RFC3617].

   An alternative approach to TFTP network booting is to bootstrap the
   system with iSCSI. In this case, the URL in the SUBOPT_BOOTFILE_URL
   suboption 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 SUBOPT_BOOTFILE_URL it is also not necessary
   anymore. So for IPv6 iSCSI booting, the "iscsi:" string MUST be
   specified as URL in the SUBOPT_BOOTFILE_URL suboption instead.

   In some different scenarios, it might also be useful to use other
   protocols like FTP or HTTP for network booting, so a firmware
   implementation can support these protocols, too. Then it is up to
   the network administrator to choose the appropriate boot protocol for
   the network, and to specify the right boot file URLs in the DHCPv6
   server configuration file.

8. IANA considerations

   The following option needs 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_NET_BOOT |  tbd  |  Section 3   |
+--------------+-------+--------------+
```

The netboot suboptions numbers form a new name space to be defined by
the IANA:

```
+-----------------------+-------+--------------+
|    Suboption name     | Value | Specified in |
+-----------------------+-------+--------------+
|  SUBOPT_BOOTFILE_URL   |  tbd  |  Section 4   |
| SUBOPT_NETBOOT_VENDOR  |  tbd  |  Section 5   |
+-----------------------+-------+--------------+
```

9. Security considerations

   The new DHCPv6 option 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 RFC
   3315 [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.

10. Acknowledgements

   The authors would like to thank Ketan P. Pancholi for corrections and
   suggestions.

   Vijayabhaskar Kalusivalingam and Senthil Balasubramanian published a
   similar draft for IPv6 network booting some years ago (available at
   http://tools.ietf.org/html/draft-ietf-dhc-dhcpv6-opt-rboot-00), which

however was abandoned for unknown reasons.

11. References

11.1. Normative References

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

   [RFC2396] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
              Resource Identifiers (URI): Generic Syntax", RFC 2396,
              August 1998.

   [RFC2732] Hinden, R., Carpenter, B., and L. Masinter, "Format for
              Literal IPv6 Addresses in URL's", RFC 2732, December 1999.

   [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.

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

   [VENDORIDS]
              IANA, "Private Enterprise Numbers",
              <http://www.iana.org/assignments/enterprise-numbers>.

11.2. Informative References

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

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

Authors' Addresses

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

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

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

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

Full Copyright Statement

Intellectual Property