IPS
Internet Draft
Document: draft-ietf-ips-iscsi-boot-01.txt
Category: Standards Track

Prasenjit Sarkar
IBM
Duncan Missimer
HP
Constantin Sapuntzakis
Cisco
12 January 2001

A Standard for BootStrapping Clients using the iSCSI Protocol

Status of this Memo

Abstract

The Small Computer Systems Interface (SCSI) is a popular family of protocols for communicating with I/O devices, especially storage devices. iSCSI is a proposed transport protocol for SCSI that operates on top of TCP[12]. This memo describes a standard mechanism to enable clients to bootstrap themselves using the iSCSI protocol. The goal of this standard is to enable clients to obtain the information to open an iSCSI session with the iSCSI bootstrpping server, assuming this information is not available.

1. Requirements

1. There must be no restriction of network topology between the iSCSI boot client and the boot server. Consequently, it is possible for an iSCSI boot client to boot from an iSCSI boot server behind gateways/firewalls/etc as long as it is possible to establish an iSCSI session between the client and the server.

2. The following represents the minimum information required for an

iSCSI boot client to contact an iSCSI boot server: (a) the client's IP address (IPv6 or IPv4); (b) the server's iSCSI Service Delivery Port Name; and (c) mandatory iSCSI initiator capability.

The above assumes that the default LUN for the boot process is 0 and the default port for the iSCSI boot server is the well-known iSCSI port. However, both may be overridden at the time of configuration.

Additional information may be required at each stage of the boot process.

3. It is possible for the iSCSI boot client to have none of the above information or capability on starting.

4. The client should be able to complete boot without user intervention (for boots that occur during an unattended power-up). However, there should be a mechanism for the user to input values so as to bypass stages of the boot protocol.

5. Additional protocol software (for example, DHCP) may be necessary if the minimum information required for an iSCSI session is not provided.

2. Related Work

The Reverse Address Resolution Protocol (RARP)[7](through the extensions defined in the Dynamic RARP (DRARP))[4] explicitly addresses the problem of network address discovery, and includes an automatic IP address assignment mechanism. The Trivial File Transfer Protocol (TFTP)[9] provides for transport of a boot image from a boot server. BOOTP[5,8,10] is a transport mechanism for a collection of configuration information. BOOTP is also extensible, and official extensions have been defined for several configuration parameters. DHCPv4[3,6] and DHCPv6[13] are standards for hosts to be dynamically configured in an IP network. The Service Location Protocol RLP provides for location of higher level services[1,15].

3. Software stage

Some iSCSI boot clients may lack the resources to boot up with the mandatory iSCSI initiator capability. Such boot clients may choose to obtain iSCSI initiator software from a boot server. Currently, there are many established protocols that allow such a service to enable clients to load software images. For example, BOOTP and DHCP servers have the capability to provide software images on requests from boot clients. A particular implementation of this approach is the PXE protocol[17], which uses DHCP extensions and MTFTP to allow boot clients to load software images.

It is to be noted that this document does not recommend any of the above protocols, and the final decision of which boot protocol is to be used to load iSCSI initiator software is left to the discretion of the implementor.

4. DHCP stage

In order to use an iSCSI boot server, the following pieces of information are required.

- The IP address of the iSCSI boot client (IPv4 or IPv6)

- The IP transport endpoint for the iSCSI service delivery port for the iSCSI boot server. If the transport is TCP, for example, this has to resolve to an IP address and a TCP port number.

- The eight-byte LUN structure identifying the device within the iSCSI boot server.

At boot time, all or none of this information may be stored in the firmware of the iSCSI boot client. This section describes techniques for obtaining the required information.

An iSCSI boot client which does not know its IP address at power-on may acquire its IP address via DHCP. An iSCSI boot client which is capable of using both DHCPv6 and DHCPv4 should first attempt to use DHCPv6 to obtain its IP address, falling back on DHCPv4 in the event of failure.

Unless otherwise specified here, DHCP fields such as the client ID and gateway information are used identically with applications other than iSCSI.

A DHCP server (v4 or v6) may instruct an iSCSI client how to reach its boot device. This is done using a variable length DHCP option field known as the ISCSI Boot Service option. The option identifier is to be allocated by the IESG during the approval process[19].

The field consists of an UTF-8[20] string and has the following composition:

<servername> ":" <port> ":" <LUN> ":" <targetname>

The fields "port", "LUN" and "targetname" are optional and should be left blank if there are no values corresponding to the fields.

The "servername" is the name of iSCSI server and contains either a

valid domain name, a literal IPv4 address, or a bracketed literal IPv6 address. If the servername field contains a literal IPv4 address, the IPv4 address is in standard dotted decimal notation. If the servername field contains an IPv6 address, the address is represented in bracketed literal IPv6 address format.

If the "servername" is a domain name, then the reply from the host configuration server may contain the Domain Name Server Option[2].

The "port" is the decimal representation of the port on which the iSCSI boot server is listening. If not specified, the port defaults to the well-known iSCSI port.

The "LUN" field is a 16 byte hexadecimal representation of the 8-byte LU number in hex. Digits above 9 may be either lower or upper case, and all 16 nibbles must be present. If the LUN field is blank, then LUN 0 is assumed.

Note that SCSI targets are allowed to present different LU numberings for different SCSI initiators, so that to our knowledge nothing precludes a SCSI target from exporting several different devices to several different SCSI initiators as their respective LU 0s.

The "targetname" field is a string containing the name of the iSCSI target, the details of which are specified by the iSCSI standard[12]. If the targetname is provided, the iSCSI boot client may use the targetname as mandated by the iSCSI standard.

The above assumes that the default connection method uses TCP as stated in the iSCSI standard. Should SCTP[18] be also approved as a transport mechanism for iSCSI, then the draft will be amended to provide for alternate transport protocols.

5. Discovery Service stage:

This stage is required if the DHCP server (v4 or v6) is unaware of the identity of the iSCSI boot server.

The iSCSI boot client then may start the discovery process according to the specifications stated in the iSCSI Naming and Discovery document[14]. The discovery service provides the boot client with a list of SCSI targets the client is allowed to access, along with the access permissions for each of the target. The nature and implemention of the discovery service is outside the scope of this document.

The iSCSI boot client goes through the list of SCSI targets and must select the first SCSI target with the bootable attribute as the iSCSI

boot server. If such an attribute does not exist in any of the SCSI targets, the boot client must select the first SCSI target in the list of SCSI targets as the iSCSI boot server.

If the list of SCSI targets is empty, subsequent actions are left to the discretion of the implementor.

The packets and software requirements are stated in the iSCSI Naming and Discovery document[14].

6. Boot Stage

Once the iSCSI boot client has obtained the minimum information to open an iSCSI session with the iSCSI boot server, the actual booting process can start.

The actual sequence of iSCSI commands needed to complete the boot process is left to the implementor. This was done because of varying requirements from different vendors and equipments, making it difficult to specify a common subset of the iSCSI standard that would be acceptable to everybody.

The iSCSI session established for boot may be taken over the booted software in the boostrapping client - this is left to the discretion of the implementor.

7. Security

Securing the host configuration protocol is beyond the scope of this document. Authentication of DHCP messages is described in [16].

The iSCSI standard support various methods of authenticated login and encrypted and authenticated connections for security. How to configure the security parameters of an iSCSI boot client is beyond the scope of this document.

The security discussions in the iSCSI standard[12] are applicable to this document.

Acknowledgments

We wish to thank John Hufferd for taking the initiative to form the iSCSI boot team. We also wish to thank Doug Otis and David Robinson for helpful suggestions and pointers regarding the draft document.

References

[1] Guttman, E., Perkins, C., Verizades, J., Day, M., "Service

Location Protocol v2", RFC 2608, June 1999.

[2] Alexander, S., and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, Lachman Technology, Inc., Bucknell University, October 1993.

[3] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, Bucknell University, March 1997.

[4] Brownell, D, "Dynamic Reverse Address Resolution Protocol (DRARP)", Work in Progress.

[5] Croft, B., and J. Gilmore, "Bootstrap Protocol (BOOTP)", RFC 951, Stanford and SUN Microsystems, September 1985.

[6] Droms, D., "Interoperation between DHCP and BOOTP" RFC 1534, Bucknell University, October 1993.

[7] Finlayson, R., Mann, T., Mogul, J., and M. Theimer, "A Reverse Address Resolution Protocol", RFC 903, Stanford, June 1984.

[8] Reynolds, J., "BOOTP Vendor Information Extensions", RFC 1497, USC/Information Sciences Institute, August 1993.

[9] Sollins, K., "The TFTP Protocol (Revision 2)", RFC 783, NIC, June 1981.

[10] Wimer, W., "Clarifications and Extensions for the Bootstrap Protocol", RFC 1532, Carnegie Mellon University, October 1993.

[11] Bradner, S., "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

[12] Satran, J., "iSCSI", Internet-Draft, November 2000.

[13] Bound, J., Canney, M., and Perkins, C., "Dynamic Host Configuration Protocol for IPv6", Internet-Draft, November 2000.

[14] Voruganti, K. et al., "iSCSI Naming and Discovery", Internet-Draft, November 2000.

[15] Veizades, J., Guttman, E., Perkins, C., Kaplan, S., "Service Location Protocol", RFC 2165, June 1997.

[16] Droms, R., Arbaugh, W., "Authentication for DHCP Messages", Internet-Draft, November 2000.

[17] http://developer.intel.com/ial/WfM/wfm20/design/pxedt/index.htm

[18] Stewart, R., et al. "Stream Control Transmission Protocol", RFC 2960, October 2000.

[19] Droms, R., "Procedures and IANA Guidelines for Approval of New DHCP Options and Message Types", RFC 2939, September 2000.

[20] Yergeau, F., "UTF-8: A Transformation Format for ISO-10646", RFC 2279, January 1998.

Author's Addresses

Prasenjit Sarkar
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120, USA
Phone: +1 408 927 1417
Email: psarkar@almaden.ibm.com

Duncan Missimer
Hewlett-Packard Company
19420 Homestead Road, M/S 43lo
Cupertino, CA 95014, USA
Phone: +1 408 447 5390
Email: duncan_missimer@hp.com

Constantine Sapuntzakis
Cisco Systems, Inc.
170 W. Tasman Drive
San Jose, CA 95134, USA
Phone: +1 650 520 0205
Email: csapuntz@cisco.com

Full Copyright Statement