Features of the iSCSI Protocol

Kalman Z. Meth and Julian Satran, IBM Haifa Research Lab

ABSTRACT

The iSCSI protocol specifies how to access SCSI storage devices over a TCP network. In this article we give a brief introduction to the iSCSI protocol and a brief comparison with alternative technologies. We then discuss the basic features of the iSCSI protocol: sessions, naming, discovery, security, data placement, and recovery.

INTRODUCTION

The SCSI protocol [1] is widely used to access storage devices. The iSCSI protocol [2] is a transport for SCSI over TCP/IP [3, 4]. SAM-2 [1] defines an architecture model for SCSI transports, and iSCSI defines such a transport on top of TCP/IP. Other SCSI transports include SCSI Serial [5] and Fibre Channel Protocol (FCP) [6, 7]. Until recently standard networking hardware (Ethernet) and IP-based [3] protocols could not provide the necessary high bandwidth and low latency needed for storage access. With the recent advances in Ethernet technology, it is now practical from a performance perspective to access storage devices over an IP network. 1 Gb Ethernet is now widely available and is competitive with current 1 and 2 Gb Fibre Channel technology. 10 Gb Ethernet will soon also be widely available. Similar to FCP, iSCSI allows storage to be accessed over a storage area network (SAN), allowing shared access to storage. A major advantage of iSCSI over FCP is that iSCSI can run over standard off-the-shelf network components, such as Ethernet. Furthermore, iSCSI can exploit existing IP-based protocols such as IPSec [8] for security and Service Location Protocol (SLP) [9] for discovery. A network that incorporates iSCSI SANs need use only a single kind of network infrastructure (Ethernet) for both data and storage traffic, whereas use of FCP requires a separate kind of infrastructure (Fibre Channel) for storage (Fig. 1). IP-based SANs using iSCSI can be managed using existing and familiar IP-based tools such as Simple Network Management Protocol (SNMP) [10], whereas FCP SANs require specialized management infrastructure. Furthermore, iSCSI-based SANs can extend over arbitrary distances, just like TCP, and are not subject to distance limitations that currently limit FCP.

In addition to iSCSI, several other protocols have been defined to transport storage over an IP network. FCIP [11] is used to connect separate islands of Fibre Channel SANs over an IP network to form a single unified SAN. iFCP [12] is a gateway-to-gateway protocol for the implementation of Fibre Channel fabric functionality on a network in which TCP/IP switching and routing elements replace Fibre Channel components. Whereas FCIP and iFCP are used to allow the connection of existing Fibre Channel infrastructures to each other and to IP networks, iSCSI enables the creation of SANs completely independent of Fibre Channel.

The remainder of this article describes the main features of the iSCSI protocol.

iSCSI FEATURES

TCP

TCP was chosen as the transport for iSCSI. TCP has many features that are utilized by iSCSI:

- TCP provides reliable in-order delivery of data.
- TCP provides automatic retransmission of data that was not acknowledged.
- TCP is a friendly network citizen in that it provides the necessary flow control and congestion control to avoid overloading a congested network.
- TCP works over a wide variety of physical media and interconnect topologies.

While other protocols such as SCTP [13] also provide many of these features, TCP has the advantage of having been deployed for decades, and is therefore better understood and more widely available.

SESSIONS

SCSI commands are typically issued by a storage initiator (client) to a storage target (server). The relationship between SCSI entities is referred to as a *nexus*. The iSCSI entity corresponding to an I_T_NEXUS (initiator-target nexus) is an iSCSI session. An iSCSI session is a collection of TCP connections between an iSCSI initiator and an iSCSI target used to pass SCSI commands and data between the initiator and the target. The TCP connections of an iSCSI session may go over the same or different physical media.

Even though a single TCP connection is sufficient to establish communication between an initiator and a target, it is often advantageous to use multiple connections:

0163-6804/03/$17.00 © 2003 IEEE

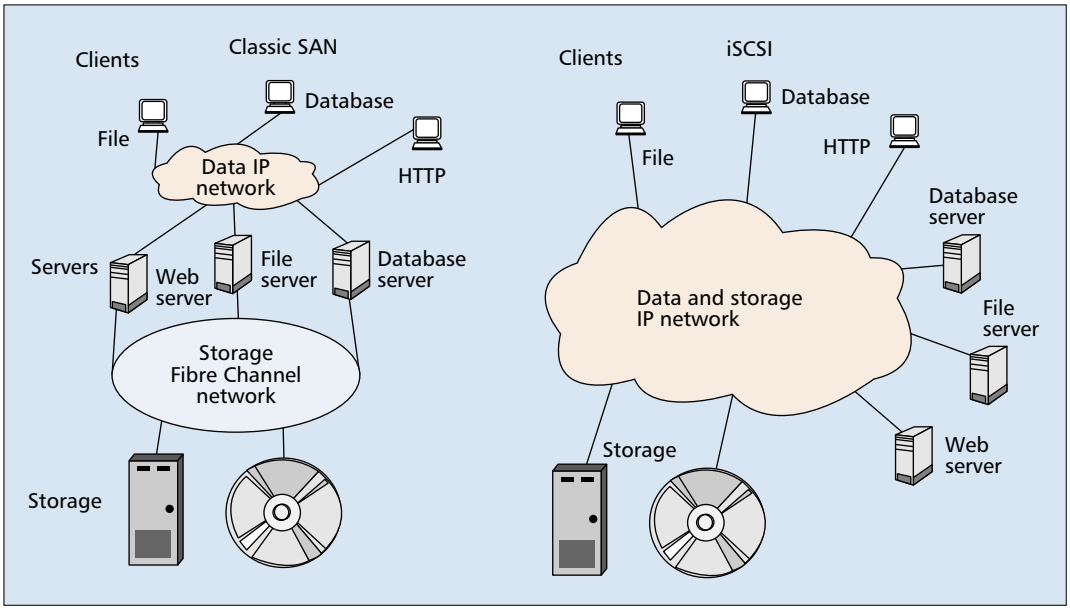

Figure 1. *Classic SAN vs. iSCSI.*

The initiator sends a Login Request PDU to the target. The initiator and target may authenticate each other and may negotiate operational parameters. A default authentication method (CHAP) must be supported by all compliant iSCSI implementations.

- It is often not possible to achieve the maximum bandwidth of the underlying physical interconnect using only a single TCP connection.
- When working on a multiprocessor machine, it may be advantageous to allow separate threads running on the different processors to simultaneously utilize different TCP connections.
- If there are more than one physical interconnect between the initiator and the target, their bandwidth can be aggregated by spreading multiple TCP connections over all the possible physical interconnects.

iSCSI Protocol Data Units

iSCSI defines its own packets, referred to as iSCSI protocol data units (PDUs). iSCSI PDUs consist of a header and possible data, where the data length is specified within the iSCSI PDU header. An iSCSI PDU is sent as the contents of one or more TCP packets.

The most commonly used iSCSI PDU types are:

- SCSI Command/Response
- Data In/Out
- Ready to Transfer (R2T)
- Login Request/Response

The SCSI Command PDU is used to transfer a SCSI command from the initiator to the target. If the SCSI command requests to read data from the target, the target will send the data to the initiator in one or more Data In PDUs. If the SCSI command requests to write data to the target, the initiator will send the data to the target in one or more Data Out PDUs. The target may specify to the initiator which part of the data to send by sending to the initiator an R2T PDU. Upon completion of the entire data transfer, the target sends a SCSI Response PDU to the initiator indicating either successful completion of the command or any error condition detected. For each SCSI Command PDU there is a corresponding single SCSI Response PDU, but possibly multiple (or no) Data PDUs. SCSI Data and Response PDUs must be sent over the same TCP connection on which their corresponding SCSI Command PDU was issued.

Login

Immediately upon establishing a TCP connection between an iSCSI initiator and iSCSI target, a login procedure must be performed. The initiator sends a Login Request PDU to the target. The initiator and target may authenticate each other and may negotiate operational parameters. A default authentication method, Challenge-Handshake Authentication Protocol (CHAP), must be supported by all compliant iSCSI implementations. Some of the operational parameters that may be negotiated are the maximum size of data PDUs, the maximum number of connections to be used in the session, the amount of unsolicited data that may be sent by the initiator (without the use of R2T), the level of error recovery supported, and whether or not digests will be used for error detection. After both sides are satisfied with the authentication and the setting of the operational settings, the target sends a Login Response PDU with an indication that the login procedure has completed. Only then may the connection be used to pass SCSI commands and data.

Naming

Borrowing from other Internet protocols, iSCSI uses a URL-like scheme to name targets. iSCSI names are meant to be global, similar to World Wide Names used by Fibre Channel. An iSCSI entity might have its IP address changed while retaining its name. An iSCSI entity is therefore identified by its name and not its address(es). This allows for easier handling of iSCSI names by proxies, gateways, network address translation boxes, firewalls, and so on. iSCSI names should be unique worldwide. Typical iSCSI names might look like this:

iqn.2001-04.com.acme:storage.disk2.sys1.xyz

TCP has a checksum facility to help detect such errors that occur during transmission. While the probability of the TCP checksum failing to detect an error is quite small, it is not good enough for some storage environments.

The prefix *iqn* stands for *iSCSI qualified name*. The above named device was produced by a company that owned the domain name *acme.com* during *2001-04*. This may be followed by a character string, deemed appropriate by the domain name owner, to further qualify the name of the particular device and make it unique.

Discovery

When using storage devices over a network, one has to deal with the ability of an initiator to discover the devices it may use. One approach is for an administrator to statically configure the initiator, providing the initiator with a list of the names and addresses of the iSCSI devices to which the initiator may connect. If additional iSCSI devices are later added to the network, the statically configured initiator would not be able to access the new devices without being reconfigured. An alternative more dynamic method is to use SLP [9], which already exists in the IP family of protocols. iSCSI targets can register themselves using SLP, and initiators can query SLP agents to obtain information about registered targets. In this way, iSCSI targets can be added to the network, and the topology can change over time, but initiators can easily find new targets without having to be reconfigured. A similar mechanism is provided by the recently defined iSNS protocol [14].

An additional discovery mechanism, SendTargets, is provided in the iSCSI protocol itself, especially useful for gateway devices. In this method, an initiator is statically configured to connect to specific iSCSI gateway devices. The initiator establishes a discovery session with the iSCSI gateway device, and then issues the SendTargets request to the iSCSI gateway device. The iSCSI gateway device then responds with a list of attached iSCSI targets that are available to the initiator. The initiator may then proceed to connect to the specified iSCSI target devices.

Data Integrity

TCP has a checksum facility to help detect errors that occur during transmission. While the probability of the TCP checksum failing to detect an error is quite small, it is not good enough for some storage environments. The TCP checksum also does not provide protection for corruptions that occur while a message is in the memory of some router (when header information might be recalculated, and the data is no longer protected by a checksum). iSCSI therefore defines its own cyclic redundancy check (CRC) checksum to ensure end-to-end integrity of its packet headers and data. Initiators and targets may negotiate whether or not to use this CRC checksum.

Security

When storage devices were directly attached to host machines, the data on the storage devices was considered secure by its being inaccessible to the outside world. With iSCSI attached storage devices, this is no longer the case. A serious security problem may arise if sensitive storage data is accessed over a general data network. One possible solution is to use a physically separate network for the storage data, similar to what is done with Fibre Channel (Fig. 1). This solution requires a second physical IP network, which is still cheaper than having a second physical Fibre Channel network. Alternatively, a single physical IP network can be used together with encryption of the storage data. Encryption of data on an IP network can be provided by IPSec [8]. iSCSI simply uses the existing IP-family security protocol to protect sensitive storage data from possible security attacks such as sniffing and spoofing.

Anticipated Use of iSCSI

Some of the design decisions of iSCSI were strongly influenced by the perception of how iSCSI would eventually be used. iSCSI was designed to allow efficient hardware and software implementations to access I/O devices attached over any IP network. iSCSI was also designed for a wide variety of environments and applications including local and remote storage access, local and remote mirroring, local and remote backup/restore. It was assumed that TCP/IP acceleration adapters and even iSCSI host bus adapters (HBAs) would become prevalent, and it would be strongly desirable to define the protocol to allow high-performance adapter implementations. Mechanisms were therefore included to overcome various anticipated problems, such as maintaining high bandwidth despite frequently dropped packets. Care was taken to not limit the application of iSCSI to disks; mechanisms were provided for various types of SCSI devices, especially tapes, for which it is inconvenient and perhaps prohibitive to cancel and restart commands.

Direct Data Placement

In typical TCP implementations, data that arrives on a TCP connection is first copied into temporary buffers. The TCP driver then examines the connection identification information (source and destination addresses and port numbers) to determine the intended receiver of the data. The data is then copied into the receiver's buffers. For SCSI data, there might be many pending SCSI commands at any given instant, and the received data typically must be copied into the specific buffer provided by the SCSI layer for the particular command. This entire procedure might require the receiving host to copy the data a number of times before the data ends up in its final destination buffer. Such copies require a significant amount of CPU and memory bus usage that would adversely affect the system performance. It is therefore most desirable to be able to place the data in its final destination with a minimum number of copies.

iSCSI Data PDU headers contain sufficient information to allow an iSCSI adapter (HBA) to perform direct data placement. The information provided in an iSCSI Data PDU header includes a transfer tag to identify the SCSI command and its corresponding buffer, a byte offset relative to the beginning of the corresponding buffer, and a data length parameter indicating the number of bytes being transferred in the current data packet. This information is sufficient to enable direct placement of the arriving data into preregistered SCSI-provided buffers. An iSCSI adapter that performs both TCP and iSCSI processing on the

adapter will have sufficient information in the TCP and iSCSI headers to place arriving iSCSI data directly into the appropriate SCSI buffers without having to copy the data into additional temporary buffers on the host machine.

Recovery

The iSCSI protocol defines several levels of recovery to provide resilience in the face of a wide range of possible errors and failures. iSCSI error handling and recovery is expected to be a rare occurrence, and may involve a significant amount of overhead. It is anticipated that most computing environments will not need all the levels of recovery defined in the iSCSI specification.

The most basic recovery class is session failure recovery. All iSCSI specification compliant implementations must implement session failure recovery. Session recovery involves the closing of all of the session's TCP connections, aborting all outstanding SCSI commands on that session, terminating all such aborted SCSI commands with an appropriate SCSI service response at the initiator, and restarting a new set of TCP connections for the particular session. Implementations may perform session failure recovery for any iSCSI error detected.

A less drastic kind of recovery implementations may perform is digest failure recovery. If a CRC checksum error is detected on iSCSI data, the data packet must be discarded. Instead of performing session recovery, implementations may use the digest failure recovery mechanism to ask the connecting peer to resend only the missing data. Similarly, if a sequence reception timeout occurs, a similar mechanism can be used to ask the connecting peer to resend missing commands, responses, or other numbered packets that are expected.

If a CRC checksum error is detected on an iSCSI packet header, the packet must be discarded since it was corrupted. As a result, synchronization between the initiator and target may be lost. The iSCSI protocol allows for a new TCP connection to be established within the session, and defines mechanisms for the initiator and target to synchronize with one another to continue to smoothly interact. A new TCP connection may be designated to take over from an old TCP connection that seems to have become defective. This level of recovery is called connection recovery. Processing of commands that were started on the defective TCP connection can be continued on the new TCP connection.

Conclusion

The iSCSI protocol enables access to storage devices over an IP network. One of the main objectives in defining the iSCSI protocol was to make use of existing IP infrastructure whenever possible. Thus, TCP is used as the underlying transport, IPSec is exploited to provide network security, SLP can be used to provide discovery, and so on. Since iSCSI runs over standard off-the-shelf network components, the cost of setting up an iSCSI SAN is significantly lower than that of a Fibre Channel SAN. This will make iSCSI more affordable and manageable than Fibre Channel, and enable it to become the protocol of choice for SANs.

Acknowledgments

Numerous people contributed to the discussions and working group that resulted in the definition of the iSCSI protocol. We thank them all. The main contributors are listed in the iSCSI protocol specification document [2].

References

[1] Nat'l. Committee for Info. Tech. Stds. (NCITS), "SAM2, SCSI Architecture Model 2," T10, Project 1157-D, Rev. 23, Mar. 16, 2002.
[2] J. Satran *et al.*, "iSCSI (Internet SCSI)," draft-ietf-ips-iscsi-20.txt,Jan. 2003; ietf.org/html.charters/ips-charter.html or www.haifa.il.ibm.com/satran/ips
[3] "Internet Protocol (IP)," RFC 791, DARPA, Sept. 1981; http://ietf.org/rfc.html
[4] "Transmission Control Protocol (TCP)," DARPA, RFC 793, Sept. 1981; http://ietf.org/rfc.html
[5] "SBP-2, Serial Bus Protocol 2," ANSI NCITS.325-1999.
[6] FCP, SCSI-3 Fibre Channel Protocol, ANSI X3.269-1996.
[7] A. Benner, *Fibre Channel: Gigabit Communications and I/O for Computer Networks*, McGraw-Hill, 1996.
[8] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol (IPSec)," RFC 2401, Nov. 1998; http://ietf.org/rfc.html
[9] "Service Location Protocol (SLP)," June 1997, RFC 2165, http://ietf.org/rfc.html
[10] J. Case *et al.*, "Simple Network Management Protocol (SNMP)," RFC 1157, 1990; http:// ietf.org/rfc.html
[11] M Rajagopal *et al.*, "Fibre Channel over TCP/IP (FCIP)," draft-ietf-ips-fcovertcpip-12.txt, Aug. 2002; ietf.org/html.charters/ips-charter.html or www.haifa.il.ibm.com/satran/ips
[12] C. Monia *et al.*, "iFCP — A Protocol for Internet Fibre Channel Storage Networking," draft- ietf-ips-ifcp-13.txt, Augm 2002; ietf.org/html.charters/ips-charter.html or www.haifa.il.ibm.com/satran/ips
[13] R. Stewart *et al.*, "Stream Control Transmission Protocol (SCTP)," RFC 2960, Oct. 2000; http://ietf.org/rfc.html
[14] J. Tseng *et al.*, "Internet Storage Name Service (iSNS)," draft-ietf-ips-isns-14.txt, Oct. 2002; ietf.org/html.charters/ips-charter.html or www.haifa.il.ibm.com/satran/ips

Biographies

KALMAN Z. METH (meth@il.ibm.com) received his B.A. in mathematics and computer science from Temple University in 1982, and his MS and Ph.D. in mathematics from the Courant Institute, New York University, in 1985 and 1988, respectively. He was a lecturer of mathematics at Temple University from 1988 to 1990. He has been a technical staff member at IBM's Haifa Research Laboratory since 1990, working in the areas of operating systems, distributed and parallel computing, real-time systems, file systems, and multimedia. He currently manages the Networked Storage Technologies group at IBM's Haifa Research Laboratory, and is one of the authors of the iSCSI protocol specification.

JULIAN SATRAN (Julian_Satran@il.ibm.com, satran@il.ibm.com) graduated (M.Sc.E.E.) in 1962 from the Polytechnic Institute Bucharest, Romania. After graduation he held senior positions in industry and academia in Romania. He immigrated to Israel in 1979 and has held senior positions in industrial R&D in Israel. Since 1987 he has been with IBM's Haifa Research Laboratory. Currently he is a distinguished engineer; his areas of interest span system and subsystem architecture, networking, development, and operating environments. He has led several pioneering research projects at the laboratory in clustering, file system structure (and object storage), I/O, and networking convergence (iSCSI), and has driven an industry-wide effort to standardize iSCSI. He also frequently teaches both graduate and undergraduate courses (advanced OS, OS, advanced storage) at Haifa University and the Technion.

Since iSCSI runs over standard off-the-shelf network components, the cost of setting up an iSCSI SAN is significantly less than the cost of a Fiber Channel SAN. This will make iSCSI more affordable and manageable than Fiber Channel, and will enable it to become the protocol of choice for SANs.