NISHAN SYSTEMS

iSCSI Technical White Paper

Nishan Systems
3850 North First Street
San Jose, CA 95134
Tel 408-519-3700
Fax 408-519-3705
www.NishanSystems.com

Introduction

In this paper we will examine the main technical attributes of the iSCSI protocol and what is required to transport block data in a native IP end-to-end storage solution. The iSCSI protocol initiative has to contend with the inherent contradiction posed by the requirements of SCSI for stable, responsive communications and the potentially unstable environment typical of IP networks. The iSCSI protocol specification therefore devotes ample space for validation of data and command exchanges as well as for recovery from data loss that may be engendered by the lower layer transport.

iSCSI is based on the Small Computer Systems Interface (SCSI) which enables host computer systems to perform block data input/output (I/O) operations with a variety of peripheral devices. Target devices may include disk and tape devices, optical storage devices, as well as printers and scanners. The traditional SCSI connection between a host system and peripheral devices is based on parallel cabling, which has inherent distance and device support limitations. For storage applications, these limitations have fostered the development of new technologies based on networking architectures such as Fibre Channel and Gigabit Ethernet.

IP Storage networks based on serial gigabit transport layers overcome the distance, performance, scalability, and availability restrictions of parallel SCSI implementations. By leveraging SCSI protocols over networked infrastructures, storage networking enables flexible high-speed block data transfers for a variety of applications, including tape backup, server clustering, storage consolidation, and disaster recovery. The Internet SCSI (iSCSI) protocol defines a means to enable block storage applications over TCP/IP networks.

The SCSI architecture is based on a client/server model, and iSCSI takes this into account to deliver storage functionality over TCP/IP networks. The client is typically a host system such as file server that issues requests to read or write data. The server is a resource such as a disk array that responds to client requests. In storage parlance, the client is an initiator and plays the active role in issuing commands. The server is a target and has a passive role in fulfilling client requests, having one or more logical units that process initiator commands. Logical units are assigned identifying numbers, or logical unit numbers (LUNs).

The commands processed by a logical unit are contained in a Command Descriptor Block (CDB) issued by the host system. A CDB sent to a specific logical unit, for example, might be a command to read a specified number of data blocks. The target's logical unit would begin the transfer of the requested blocks to the initiator, terminating with a status to indicate completion of the request. The central mission of iSCSI is to encapsulate and reliably deliver CDB transactions between initiators and targets over TCP/IP networks.

iSCSI Network Architecture

As shown in Figure 1, an iSCSI IP Storage network may be composed of native iSCSI initiators, such as file servers, and iSCSI targets, such as disk arrays and tape subsystems.

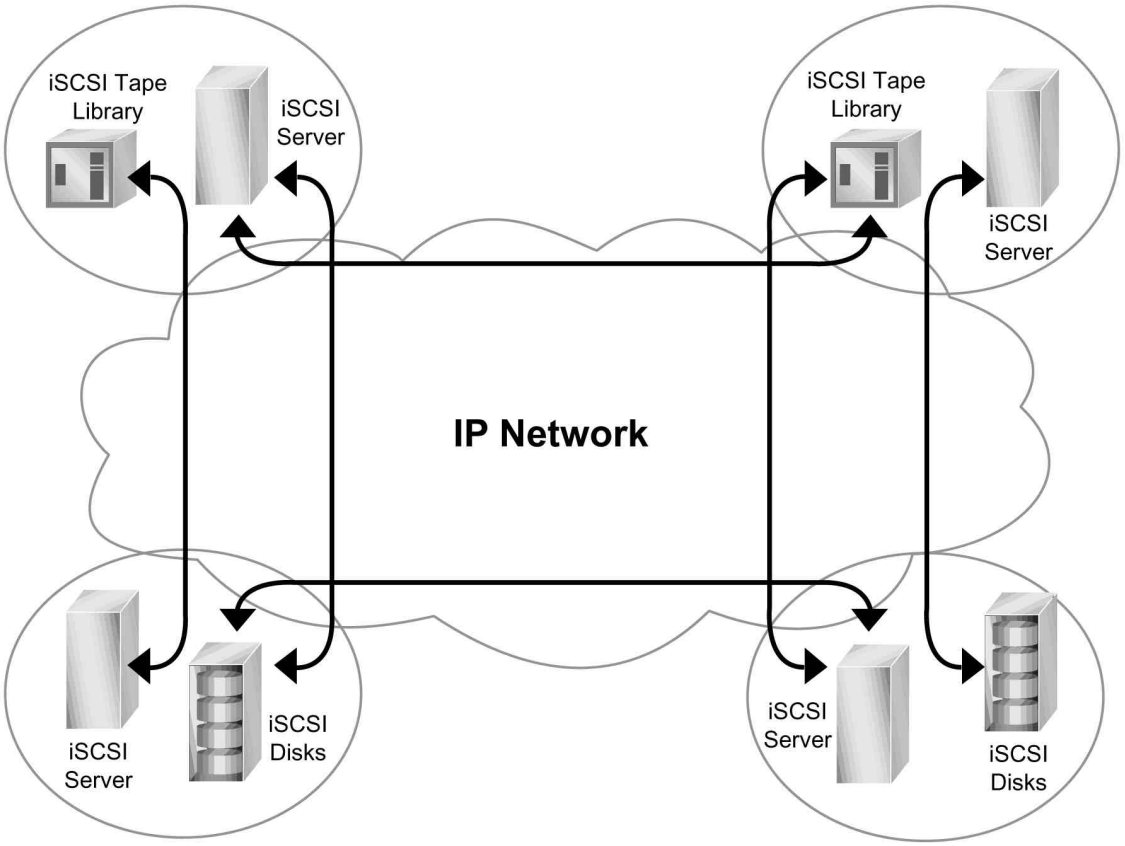

Figure 1: *A native iSCSI storage network of initiators (servers) and targets (storage resources)*

In this example, each host and storage resource supports a Gigabit Ethernet interface and iSCSI protocol stack. This enables storage devices to be plugged directly into Gigabit Ethernet switches and/or IP routers and appear simply as any other IP entities in the network. As with normal IP implementations, direct connection to the IP network pushes responsibility for device discovery and connection establishment back onto the end device. In order for an initiator to discover storage resources, for example, it would require a list of IP addresses of its intended targets. This list could be provided by a lookup table or by a DNS-type service in the network. The Internet Storage Name Service (iSNS) protocol facilitates device discovery for iSCSI initiators. In the diagram above, an iSCSI initiator would first query an iSNS server to learn the IP addresses of potential target resources, and then establish TCP/IP connections to them.

iSCSI Protocol Model

iSCSI uses TCP/IP for reliable data transmission over potentially unreliable networks. As shown in Figure 2, the iSCSI layer interfaces to the operating systems's standard SCSI set. The iSCSI layer includes encapsulated SCSI commands, with data and status reporting capability. When, for example, the operating system or application requires a data write operation, the SCSI CDB must be encapsulated for transport over a serial gigabit link and delivered to the target.

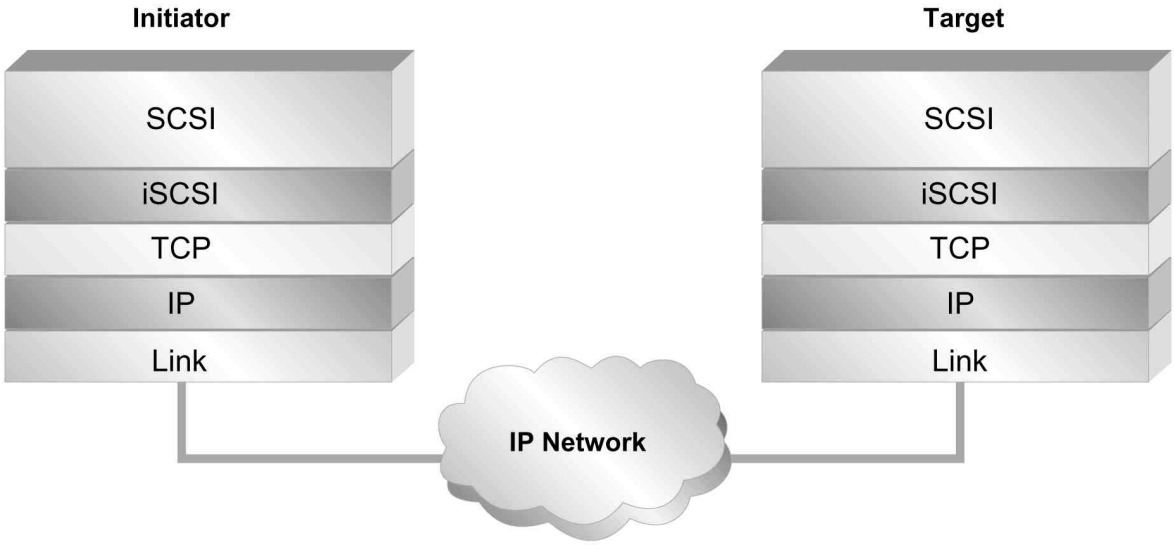

Figure 2: *iSCSI protocol layering model*

The iSCSI protocol monitors the block data transfer and validates completion of the I/O operation. This occurs over one or more TCP connections between initiator and target. In practical applications, an initiator may have multiple target resources over an IP network, and consequently have multiple concurrent active TCP connections.

iSCSI Address and Naming Conventions

Following the SCSI Architectural Model (SAM-2), iSCSI implements a client-server model between disk targets and initiating hosts. Since iSCSI targets and initiators are participants on an IP network, the clients and servers have a Network Entity identity which is equivalent to the IP addresses they are assigned. As shown in Figure 3, the Network Entity may contain one or more iSCSI Nodes.

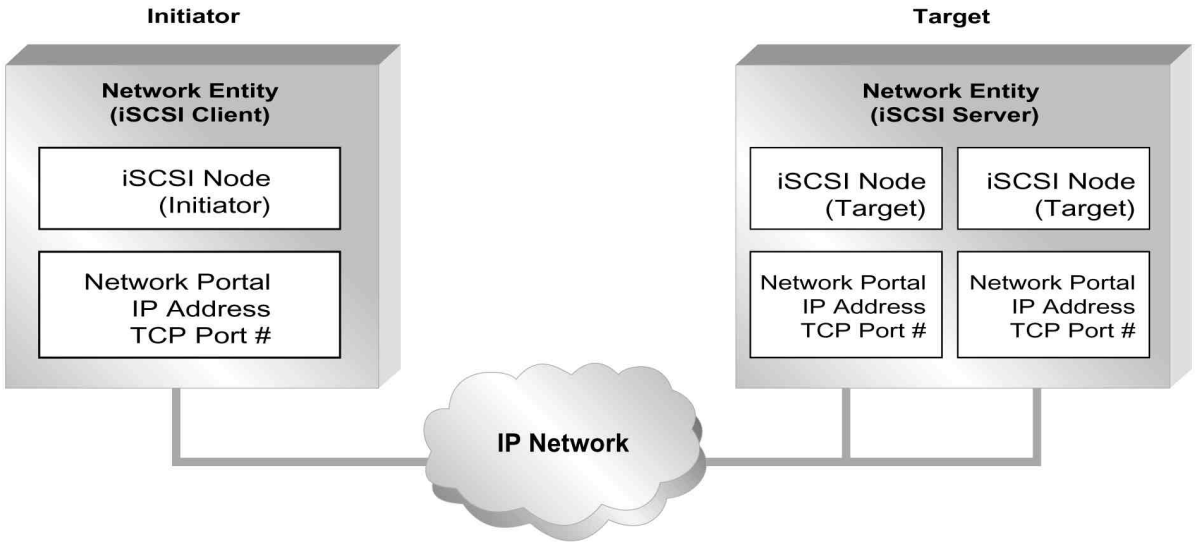

Figure 3: *Highest level SCSI name objects*

The iSCSI Node object identifies a SCSI device within a Network Entity that is accessible through the network. The Network Portal is the combination of the Node's assigned IP address and the TCP port number. Since a Network Entity may represent a gateway fronting multiple initiators or targets, the Network Entity object allows for multiple iSCSI Nodes. Each iSCSI Node is identified by a unique iSCSI name that may be up to 255 bytes long. While this may seem to be a fairly long identifier, the iSCSI protocol attempts to follow Internet conventions with human-readable names that can be parsed by a Domain Name Server (DNS) or other resource locator implementations. The 255 bytes insures globally unique names that can be formatted for the convenience of the storage administrator.

The combination of IP address and TCP port generates a unique network address for an iSCSI device. The 255 byte iSCSI name provides a unique human-readable identity. The separation of iSCSI names and iSCSI addresses insures that a storage device will have a unique identity in the network regardless of its location in the network. While the IP address plus TCP port number will necessarily change if a device is moved onto a different network segment, the iSCSI name will travel with the device, allowing it to be rediscovered. A further benefit of iSCSI naming is that it is soft-assigned and remains independent of supporting hardware. This allows, for example, a device driver on a host platform to be assigned a single iSCSI name even if multiple storage NICs are used to attach the host to the network. Likewise, a target device could have multiple connections to the network for redundant pathing and yet be consistently identified as a single entity via the iSCSI name.

The iSCSI naming convention is meant to assist the discovery process and validate a device's identity during the iSCSI login between initiator and target. The potentially very long 255 byte iSCSI name is therefore not used for routing, as it would place an unreasonable burden on network parsing engines. Instead, once the IP address and TCP port number are established for a specific iSCSI Node, only the IP address/TCP port combination is required for storage transactions.

iSCSI naming rules create a naming scheme that provides unique identity and is scalable and human-readable. The default iSCSI name is "iSCSI" and is offered as a convenience for probing real iSCSI names if the IP address/TCP port number is known.

The standard iSCSI name is composed of three parts: a type designator, the naming authority (e.g., the company administering iSCSI names), and a unique identifier assigned by the naming authority. For example, a fully qualified name would have a type extension of "iqn", "iSCSI qualified name". The naming authority might be a corporation maintaining a DNS server, "somecom.com". The unique device name might be "bigarray.engineering.105". When assembled as a fully qualified name, the authority field is reversed, yielding "fqn.com.somecom.bigarray.engineering.105" as the full iSCSI name for that device.

In Fibre Channel, the globally unique identity of a device is provided by a 64-bit World Wide Name (WWN), while the functional network address is the 24-bit Fibre Channel address. The WWN is also accommodated by iSCSI naming as an IEEE EUI format or "eui". This allows for a more streamlined, if less user-friendly name string, since the resulting iSCSI name is simply "eui" followed by the hexadecimal WWN, e.g., "eui.0300732A32598D26". Typically, the WWN reflects a range of unique numbers granted to the manufacturer.

In either format, the usual URL-type rules apply. No special characters (other than ASCII dots and dashes) and no white spaces are allowed. The fully qualified name format, in particular, enables storage administrators to assign meaningful names to storage devices and thus manage devices more easily. The unique identifier component can be a combination of department, application, manufacturer name, serial number, asset number, or any tag useful for recognizing and managing a storage resource.

In addition to assigned iSCSI names, the iSCSI protocol provides a supplementary alias name option. The alias name is provided as a convenience for quickly identifiying user resources, particularly if iSCSI names have been assigned by a manufacturer or third party and have little relevance to the customer's network naming conventions.

Alias names may be exchanged during iSCSI login and can also be up to 255 bytes long. Management software is normally required to make these names visible to the administrator via a command line interface or management GUI.

Discovery using iSCSI names can be performed using the Internet Storage Name Service (iSNS) or other resource locator. As implied by the structure of iSCSI names, either a distributed or centralized DNS-type lookup facilitates mapping of iSCSI names required for iSCSI login to actual iSCSI network addresses.

iSCSI Session Management

An iSCSI session between initiator and target must be enabled through an iSCSI login process. The iSCSI Login Phase is analogous to the Fibre Channel Port Login (PLOGI) process. The iSCSI Login Phase is used to negotiate any variable parameters between the two iSCSI entities and may invoke a security routine to authenticate allowable connectivity. If the iSCSI Login Phase is successful, the target will issue a login accept to the initiator; otherwise, the login is rejected, and the TCP connection is broken.

The iSCSI Login negotiations are processed via text fields with proferred values for a series of parameters. Parameter keys and the corresponding range of valid values supported by each iSCSI entity are exchanged. When differences in values occur between two devices (e.g., the number of concurrent TCP connections that may be supported), the lower of the two values is used as the common session parameter. Text fields are also used to exchange names and aliases, as well as negotiable parameters such as the type of security protocol to be used, the maximum data payload size supported, support for unsolicited data, and timeout values.

An iSCSI Network Entity may have one or more Network Portals (IP address plus TCP Port number) for attachment to the network. The Network Entity may also contain one or more iSCSI Nodes, represented by unique iSCSI names. As initiators establish iSCSI sessions with targets, session IDs are generated to uniquely identify individual conversations between specific iSCSI Nodes within the corresponding Network Entities. An initiator logging onto a target, for example, would include its iSCSI name and an initiator session ID (ISID), the combination of which would be unique within its host Network Entity. A target responding to the login request, would generate a unique target session ID (TSID), which likewise, in combination with its iSCSI name, gives that session a unique identity within the Network Entity in which it resides. A single ISID/TSID session pair may have multiple TCP connections between them, per the results of login negotiation.

Once login is completed, the iSCSI session leaves the Login Phase and enters the Full Feature Phase for normal SCSI transactions. If multiple TCP connections for that session have been established, individual command/response pairs must flow over the same TCP connection. This is known as *connection allegiance* and insures that specific read or write commands are fulfilled without the additional overhead of monitoring multiple connections to see if a particular request is completed. A SCSI write, for example, would be performed over a single TCP connection until all data was transmitted. Unrelated transactions, however, could simultaneously be issued on their own TCP connections during the same session.

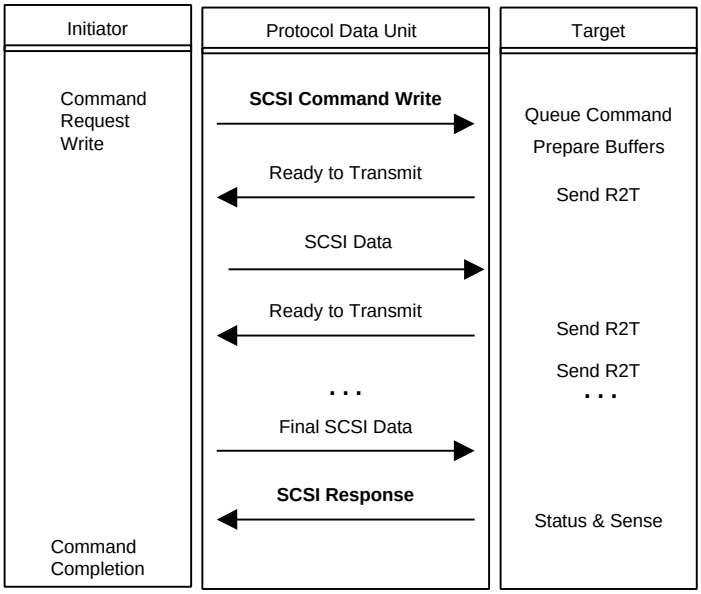

Figure 4: *iSCSI write example*

As shown in Figure 4, SCSI operations for iSCSI involve an exchange of command and responses between initiator and target. iSCSI Protocol Data Units (PDUs) are used to send CDB commands, status, and data. In the SCSI write example, the Ready to Transmit (R2T) PDUs fulfill the role of upper layer SCSI flow control between target and initiator. The R2T PDUs are issued by the target device as buffers become available to receive more data. At the completion of the write, the target issues status and sense, indicating a successful transaction.

The status of SCSI data transport during reads and writes is monitored through status and data sequence numbers as well as the buffer offset/transfer length fields in the iSCSI PDU. The target paces the flow of data by indicating the amount of data it is able to receive via a transfer length field. The target can request data blocks in any order by referencing a buffer offset established when the transaction began.

If an initator has outstanding requests to a target and receives no response, it can send the equivalent of an IP “ping” to the target to verify its status. The iSCSI NOP-Out command with the P (ping) bit set may also include test data that should be returned intact by the target. If the target does not respond, or responds with corrupted or incomplete test data, the initiator may close the connection and establish a new one for recovery.

As in other SCSI implementations, iSCSI sessions and their associated connections normally remain open, awaiting additional SCSI commands from the upper layer applications. A file server, for example, typically has assigned disk resources in the Storage Area Network (SAN) and rarely breaks connection with them unless rebooted. In an iSCSI environment, however, initiator activity may require more TCP connections for some transactions than others, and so allowance is made for selectively logging out of previously established connections. In some

instances, too, it may be desirable to logout of a session and all connections completely, for example, taking a storage resource off-line for maintenance. The iSCSI logout command supplies reason codes for terminating sessions or connections within a session, and in the latter case using the connection ID (CID) to specify which connection should be taken down. In the event of a connection error, the logout command can be issued on an alternate connection (or a newly established one) to clear a problematic TCP connection.

iSCSI Error Handling

The traditional SCSI architecture assumes a relatively error-free environment. Direct-attached SCSI devices share a dedicated parallel bus, isolated from any network disruptions. An IP Storage network may be deployed on separate, low bit error networks based on Gigabit Ethernet, but unlike traditional SCSI, enable use of standard IP networks and WAN links as well. The iSCSI specification therefore attempts to accommodate a wide variety of possible error conditions that may occur from introducing storage data into inherently unreliable network infrastructures, including the Internet.

For iSCSI error handling and recovery to function correctly, both initiators and targets must have the ability to buffer commands and responses until they are acknowledged. In a SCSI write, for example, the initiator should keep data it has just transmitted in its buffer until it has received another R2T from the target, indicating that the previous data had been received and the target is ready for more. The iSCSI end devices must be able to selectively rebuild the missing or corrupted PDU for retransmission.

The hierachy of iSCSI error detection and recovery includes, at the lowest level, detection and recovery within a SCSI task, such as retransmission of a missing or corrupt PDU. At the next layer, the TCP connection that carries a task may experience an error or failure, in which case connection recovery is attempted via a command restart. Multiple TCP connections may occur within an iSCSI session, and so errors within session may require reconstruction of individual connections.

An individual PDU may have missing or inconsistent fields within the frame. This is known as a format error and engenders a Reject iSCSI PDU in response. The Reject PDU contains an offset indicator for the first bad byte detected in the PDU header.

Another class of iSCSI errors covers corruption of data in either the data payload or header content. These are known as digest (content) errors, and include Header Digest Error and Data Digest Error conditions. These errors also trigger a Reject PDU which in turn initiates recovery of the failed PDU. In the case of a data digest error, a target can request retransmission by manipulating the offset field in the R2T PDU.

The iSCSI command for requesting retransmission of missing PDUs is the sequence number acknowledgement or SNACK PDU. The SNACK will indicate the number of missing PDUs, calculated on the last valid PDU received. In some instances, a gap may appear in a sequence of frames, with valid frames received on either side of the missing PDUs. For efficiency, only the missing PDUs are requested via the SNACK command, with the remaining valid PDUs steered

into application memory (if the synchronization and steering layer is available). SNACKs may be issued against both command and data PDUs.

And finally, the session itself may fail. Session termination and recovery is normally not required unless all other levels of recovery have failed. It requires closing all still existing TCP connections, aborting all queued tasks and outstanding SCSI commands, and restarting the session through login.

iSCSI Security

Because iSCSI must accommodate untrusted IP environments, the iSCSI specification allows for multiple security methods to be implemented. Encryption solutions that reside below iSCSI such as IPSec require no special negotiation between iSCSI end devices and are transparent to the upper layers. For other authentication implementations, such as Kerberos or Public/Private Key exchanges, the iSCSI Login Phase provides text fields for negotiating the type of security supported by both end devices. If the negotiation is successful, the PDUs exchanged between iSCSI devices will be formatted for appropriate security validation required by the agreed upon security routine. The iSNS server may also assist this process by, for example, serving as a repository for public keys.

iSCSI Issues

The SCSI protocol demands stability, data integrity, and, in current implementations, expects high bandwidth on demand. IP networks, by contrast, are inherently unstable, may drop packets under congested conditions, and have highly variable bandwidth. The TCP layer is meant to deal with the instability and packet loss that may accompany IP transport, while higher speed wide area connections can alleviate bandwidth issues for block storage data. In addition, the internal mechanisms of the iSCSI protocol provide additional monitoring of TCP connections and for recovering from lost or corrupted command and data PDUs.

For high performance storage networking applications, the iSCSI protocol is dependent on several other technologies to make it a viable alternative to Fibre Channel SANs. Imposing TCP overhead on servers, for example, is unacceptable for storage applications where server CPU cycles are at a premium. For optimum performance, iSCSI adapters require TCP/IP off-load engines (TOEs) to minimize processing overhead. TOEs will greatly assist iSCSI's ability to provide enterprise-class solutions that run at or near wire speeds.

Storage applications using iSCSI will also benefit greatly from the introduction of 10 Gigabit Ethernet. Ten gigabit and faster Ethernet enables scalable IP Storage networks that support higher populations of servers and storage devices and a variety of storage applications that can be run concurrently over the same network infrastructure. With TCP off-load engines on servers and large data pipes in the network, iSCSI solutions can achieve an enterprise-ready status for IP Storage networks.

Summary

As a standards draft proposal for IP Storage networks, iSCSI presents a end-to-end IP Storage networks solution. The iSCSI draft specification provides functionality for encapsulating SCSI CDBs and incorporates additional features optimized for TCP/IP networks. These features enable enterprise networks to implement homogeneous IP solutions for storage as well as mainstream data communications. Combined with Gigabit and 10 Gigabit Ethernet transports, IP security, and quality of service protocols, iSCSI opens new opportunities for highly scalable and secure shared IP Storage networks.

Sections of this white paper are reprinted from *IP SANs, A Guide to iSCSI, iFCP and FCIP Protocols for Storage Area Networks*, Tom Clark, Addison-Wesley, anticipated November 2001.

Printed in the U.S.A. NSWP-08, August 2001.