A Conceptual Overview of iSCSI

Executive Summary

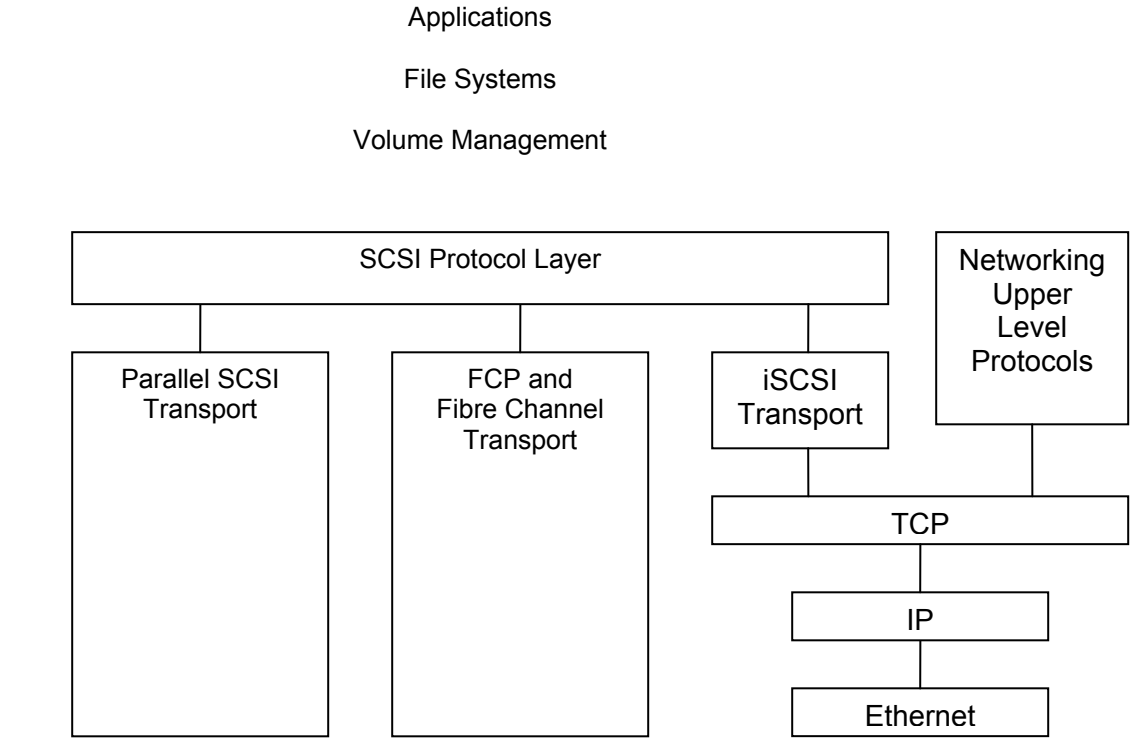
This White Paper provides a conceptual overview of iSCSI emphasizing the relationship of the iSCSI protocol to other computer subsystems. A basic understanding of mass storage and networking technologies is assumed. Concepts presented in this document include:

- iSCSI relationship with SCSI and TCP/IP
- iSCSI co-existence with current network traffic
- iSCSI utilization of mass storage and networking concepts
- iSCSI definition in terms of current network model
- iSCSI definition in terms of current mass storage model

1. Introduction

The Internet Small Computer Systems Interface (iSCSI) is a SCSI mass storage transport that operates between the Transport Control Protocol (TCP) and the SCSI Protocol Layers. The iSCSI protocol is defined in RFC 3720 [iSCSI], which was finalized by the Internet Engineering Task Force (IETF) in April, 2004. A mandate of the iSCSI protocol design required no modifications to the existing SCSI or TCP/IP protocols.

Figure 1. SCSI Transport Protocols



The Small Computer Systems Interface (SCSI) Architectural Model [SAM] is a family of protocols defining a communication protocol with SCSI I/O devices. SCSI is a client-server architecture. The phrase “SCSI Client” is synonymous with “SCSI initiator” and “SCSI server” is synonymous with

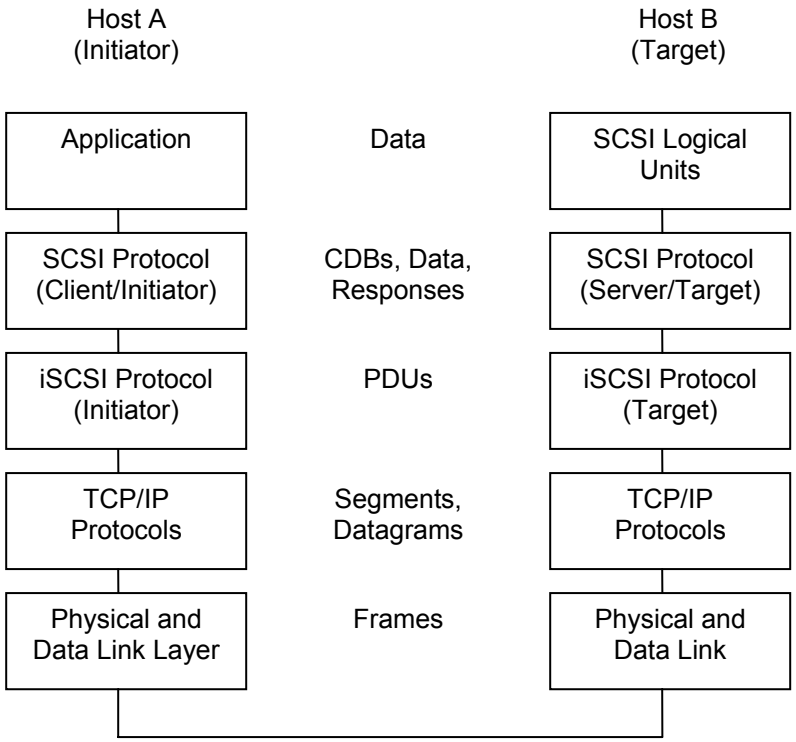
"SCSI target". iSCSI uses the phrases "iSCSI initiator" and "iSCSI target" as well. The term *initiator* or *target* is used without the protocol identifier when the protocol is not an issue.

I/O operations from user or kernel applications are translated to the SCSI command set within the SCSI protocol layer. A SCSI mass storage transport maps the SCSI client-server protocol to a specific physical transport. For example, iSCSI is a SCSI mass storage transport that maps the SCSI protocol to TCP/IP (see Figure 1). Other examples of SCSI mass storage transports are Fibre Channel and Parallel SCSI. The functionality of the traditional network architecture is not modified by iSCSI.

2. Protocol Layering

SCSI commands and data are sent from the initiator to the target, and SCSI data and responses are sent from the target to the initiator. The SCSI data, commands, and responses are moved between the SCSI initiator and the SCSI target by the mass storage transport protocols. Figure 2 depicts the protocol layering when iSCSI is used as a transport.

Figure 2. Protocol Layering



Networking is defined between hosts. The networking components of the TCP, IP, and the physical/data link layers define connectivity between two hosts on a network. It is not possible to determine which host is an iSCSI initiator and which host is an iSCSI target based solely on standard-conformant network stack implementations. Communication between the layers of the network stack is not changed when iSCSI is implemented.

In Figure 2, Host A uses the SCSI Protocol as a SCSI initiator and the iSCSI Protocol as an iSCSI initiator, and Host B uses the SCSI Protocol as a SCSI target and the iSCSI Protocol as an iSCSI target. In other words, Host A is a mass storage initiator, and Host B is a mass storage target.

At this point, the terminology may be getting a bit confusing. The confusion results from the use of 2 previously separate technologies (mass storage and networking) by a single subsystem (iSCSI). From a mass storage perspective, transport drivers such as Parallel SCSI, Fibre Channel, and iSCSI transmit SCSI operations. But for networking, the Transport Layer (TCP) is a layer of the network stack, not the mass storage transport driver. A mass storage host refers to an initiator; however, a network host is any connectivity endpoint. Servers are typically host systems, but SCSI servers are iSCSI targets. For these reasons, in any discussion about iSCSI, you must be aware of the context in which terms are used.

In Figure 2, from an application on the initiator side (Host A), mass storage I/Os are performed through a variety of system calls. When these I/Os reach the SCSI Protocol layer, device specific SCSI Command Descriptor Blocks (CDBs) are constructed to perform the requested operation. A host memory buffer descriptor for transmission or receipt of data (if necessary) is constructed. The SCSI info is then passed to the iSCSI Initiator Protocol layer where iSCSI constructs a Protocol Data Unit (PDU) containing the CDB and SCSI data (if data is to be transmitted/written). The PDU is then passed to the TCP/IP layer, where packetization for TCP segments and IP datagrams is performed. Next, the IP datagram is passed to the link layer, where Ethernet frame packetization is performed. Finally, the Ethernet frames are placed on the network.

When the target side (Host B) receives Ethernet frames, it will remove the frame encapsulation and pass the results up to the TCP/IP Protocol Layer. The IP protocol will remove the IP datagram encapsulation, and the TCP Protocol will remove the TCP segment encapsulation, leaving a PDU to be passed up to the iSCSI Target Protocol Layer. The iSCSI Target Protocol Layer will remove the SCSI CDB and data (if present) from the PDU and pass them to the SCSI Target Layer for interpretation. Finally, the requested mass storage operation will be performed at the SCSI Target Layer.

The operations described in the example above have been greatly simplified. The important thing to note is that each layer of the stack, on either the initiator or the target side, is not dependent on any layer above it in the stack. Functionality defined at the lower levels of the stack will continue to work as defined, regardless of the protocol used in the upper layers of the stack. In effect, existing conformant network functionality is not impacted by the addition of iSCSI.

3. Hardware/Software Separation

At any stack layer below the application layer on Host A in Figure 2, the protocol may be implemented in hardware. Hardware is meant to include tangible components as well as the firmware that runs on those components. Similarly, at any stack layer below the actual SCSI logical units on Host B, the protocol implementation may be in hardware. Layers below the first hardware implementation in either stack must also exist in hardware.

Hardware currently available for offloading protocol processing include

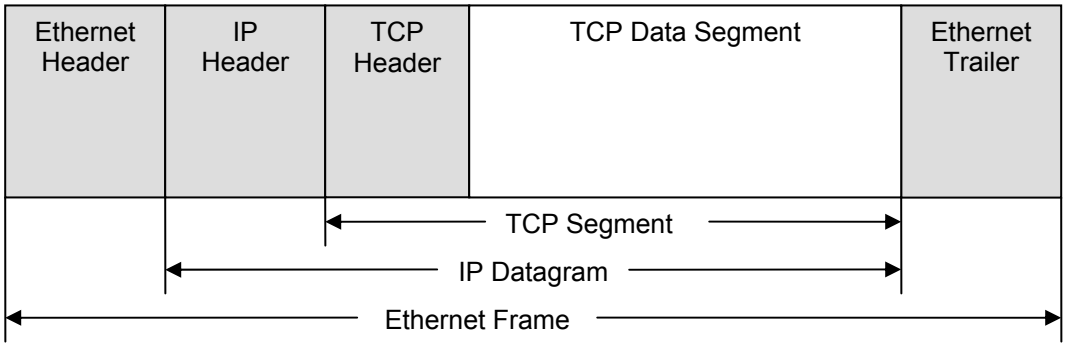
- Network Interface Cards (NICs) for offloading the Physical and Data Link Layer
- TCP Offload Engines (TOE cards) for offloading TCP, IP, and the Physical and Data Link Layers
- iSCSI Host Bus Adapters (iSCSI HBAs) for offloading the iSCSI, TCP, IP, and the Physical and Data Link Layers

NICs, TOE cards, and iSCSI HBAs may be used on the initiator or target side of the connection. There is no iSCSI requirement that corresponding layers of the stack on the initiator or target side both be implemented in hardware/firmware or software. An example would be a target that uses a NIC and an initiator that uses an iSCSI HBA.

4. Frame Construction

In section 2, “Protocol Layering”, an example of layering is provided that references the various levels of encapsulation. An Ethernet frame is depicted in Figure 3.

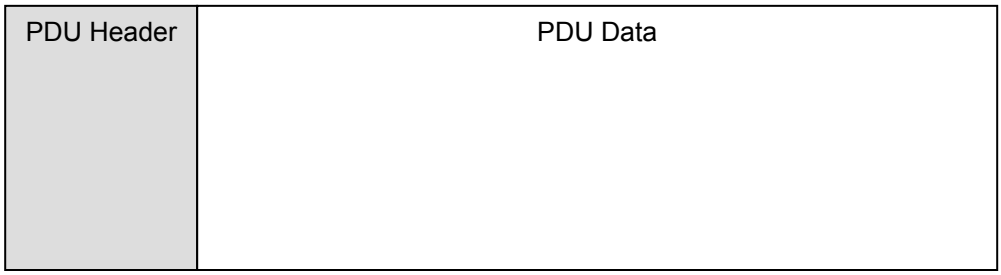
Figure 3. Layout of an Ethernet Frame



Each layer of the network stack, TCP, IP, and the Physical and Data Link layer, provides its own encapsulation. A layer of encapsulation is added or removed as each network layer is traversed.

A TCP Data Segment will contain data obtained from the TCP Stream. iSCSI data in the TCP Stream is formatted into Protocol Data Units (PDUs). Figure 4 depicts a PDU.

Figure 4. Protocol Data Unit (PDU)



There is nothing in the TCP Segment Header to indicate that the TCP Data Segment contains data of a specific protocol. The TCP/IP definition does not prevent iSCSI PDUs and other network data from being transmitted on the same network. Similarly, there is nothing that requires that they be mixed, so a network administrator can determine whether an isolated subnet for iSCSI is necessary or not. The decision is not dictated by the iSCSI protocol.

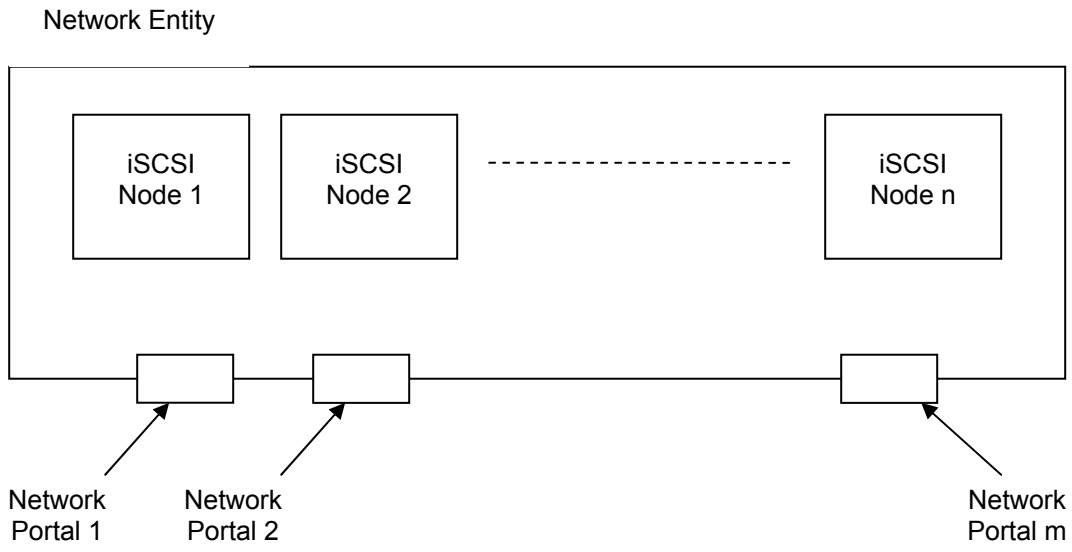
Because iSCSI PDUs are taken from the TCP Stream, they are treated like any other Stream data. A single TCP Data Segment may contain one or more PDUs. A PDU is not required to begin or end on a TCP Data Segment boundary and there is no requirement that a single PDU be wholly contained in a single TCP Data Segment.

There is a note of interest here for users of network protocol analyzers. Because there is no requirement that PDUs begin on a TCP Data Segment boundary, it is possible that simply grabbing an Ethernet frame with a protocol analyzer may result in an incorrect interpretation. To determine the start of a PDU, the end of the previous PDU must be known. While analyzers do use various heuristics to determine the start of a PDU within the first few TCP Data Segments seen, the heuristics are not foolproof, and there may be "false positives" possibly interpreting real data as the start of a PDU for a specific protocol. The potential existence of "false positives" should always be considered when debugging network problems.

5. Network Entities and iSCSI nodes

A network entity is an abstraction for any black-box network host. iSCSI uses the concept of network entity to define a context for initiators and targets. A network entity can be viewed as a container with one or more network portals for connectivity (see Figure 5). A network entity is not a concept specific to iSCSI and may be used to describe network hosts in general.

Figure 5. iSCSI Network Entity



Within the network entity, iSCSI defines one or more iSCSI nodes. An iSCSI initiator or target is typically defined by a single iSCSI node within a network entity. Multiple iSCSI nodes may be defined within a network entity. A network entity may contain both initiator and target iSCSI nodes, as well as unrelated network protocols such as NFS.

Each iSCSI node is identified by a worldwide unique iSCSI name (an ASCII string). iSCSI uses target node names to uniquely identify available SCSI storage on a TCP/IP based network. Because of the vastness of the internet, and the possibility that a small isolated LAN may someday be attached to larger LANs, the uniqueness of all iSCSI node names must be maintained. iSCSI vendors typically pre-define iSCSI node names to maintain their uniqueness, but this is not required.

Accessibility of an iSCSI node from all network portals of a network entity is not mandatory. The number of network portals actually providing access to an iSCSI node is a function of the network

entity. The iSCSI node is independent of the network portals providing access. The iSCSI name uniquely identifies the iSCSI node.

A single iSCSI node within a network entity may be accessed through multiple network portals on that network entity. An iSCSI node is identified by its worldwide unique name, and an iSCSI node is defined independent of the network portals on the network entity.

A network portal for an iSCSI initiator node is identified by its IP address. Because the iSCSI initiator will initiate I/O operations, the TCP port used by the initiator is chosen dynamically. The initiator TCP port number may change on successive application opens. There is no dedicated iSCSI initiator TCP port.

A network portal for an iSCSI target node is identified by its IP address and TCP port number. For an initiator to connect to a target, the IP address and the TCP port number must be known. TCP port 3260 is the default assigned TCP port for iSCSI targets; however, other ports may be used if the target provides this flexibility.

Be careful not to associate a network portal with a physical network port. For a single physical port MAC address, several different IP address/TCP port number pairs can be used to identify several distinct network portals.

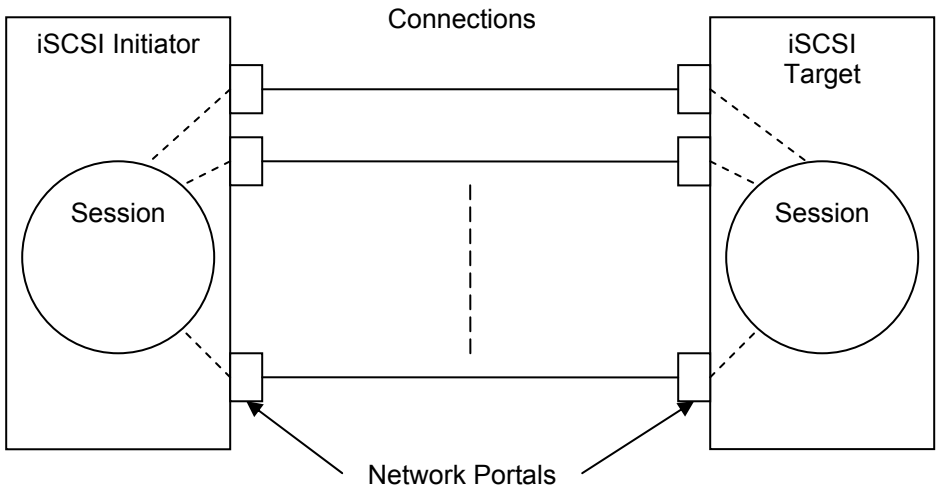
6. Sessions and Connectivity

At the SCSI level, for an initiator and target to communicate, a SCSI I-T Nexus must be established. At the iSCSI level, a concept equivalent to the SCSI I-T Nexus, known as an iSCSI session, exists between an iSCSI target and an iSCSI initiator. iSCSI session components on the initiator and target must be connected to completely establish an iSCSI session. The iSCSI session provides the mass storage transport for a SCSI I-T Nexus.

A TCP/IP connection ties the iSCSI initiator and target session components together. Network portals identified by their IP address and TCP port numbers define the endpoints of a connection. The network entity containing the iSCSI initiator node initiates the establishment of the TCP/IP connection. The initiator IP address is known, and a TCP port is allocated dynamically, thus defining the network portal for the initiator. A network portal of the network entity capable of accessing the iSCSI target node is identified by its IP address and TCP port number. Through TCP/IP, the connection is established. TCP/IP network connection establishment has not changed for iSCSI. TCP/IP protocol permits connection establishment between 2 network entities. For iSCSI, connection establishment is initiated by the network entity on behalf of an iSCSI initiator node only. Network entities will never initiate establishment of a network connection on behalf of an iSCSI target node.

An iSCSI session may have one or more connections between the session components on the initiator and the target (Figure 6). The number of connections present is a function of how iSCSI is implemented, and is not dictated by the iSCSI specification, except that at least one connection must be established for a session. For multi-connection sessions, both session endpoints must support the multi-connection functionality.

Figure 6. Representation of an iSCSI Multi-Connection Session



The individual connections of a session may go through a variety of network infrastructure components (network switches, routers, hubs, etc.). iSCSI is agnostic to the network infrastructure components. The connection endpoints are the IP address/TCP port number pairs that identify the network portals.

7. Portal Groups

A LUN view is a subset of the total set of SCSI LUNs accessible behind an iSCSI target node. To provide a SCSI I-T Nexus with a constant LUN view, each connection of an iSCSI session must have access to exactly the same LUN view. If, for some reason, any connection in a multi-connection session were to terminate, the remaining connections of that session would still have access to all of the LUNs in the LUN view.

Network portals accessing an iSCSI node and having the same SCSI LUN view can be grouped together to define an iSCSI portal group. The portal group is identified by a portal group tag uniquely defined for the iSCSI node.

To access a SCSI LUN View behind an iSCSI target node, an iSCSI address must include the following information:

- the iSCSI node name
- the network portal IP address
- the network portal TCP port number
- the iSCSI target portal group tag

For access to a LUN view through a portal group, there is a single iSCSI node name and a single iSCSI target portal group tag. Multiple network portals (IP address and TCP port pairs) may be defined for a portal group to provide multiple points of access for a network entity through an iSCSI target node to a particular LUN view.

The portal group tag is not necessary in iSCSI initiator node addresses.

8. Bibliography

iSCSI Internet Small Computer Systems Interface, Internet Engineering Task Force, http://www.ietf.org/rfc/rfc3720.txt?number=3720

SAM SCSI Architectural Model, Technical Committee T10, http://www.t10.org

9. Acronyms

The following are acronyms used throughout this document.

CDB	Command Descriptor Block
FCP	Fibre Channel Protocol
HBA	Host Bus Adapter
IETF	Internet Engineering Task Force
I/O	Input/Output
IP	Internet Protocol
iSCSI	internet Small Computer Systems Interface
I-T	Initiator-Target
LAN	Local Area Network
LUN	Logical Unit Number
NIC	Network Interface Card
PDU	Protocol Data Unit
SAM	SCSI Architectural Model
TCP	Transport Control Protocol
TOE	TCP Offload Engine

HP Manufacturing Part Number 5991-1186

03/2005