        Small Computer Systems Interface protocol over the Internet (iSCSI)
                    Requirements and Design Considerations

Status of this Memo

Copyright Notice

Abstract

   This document specifies the requirements iSCSI and its related
   infrastructure should satisfy and the design considerations guiding
   the iSCSI protocol development efforts. In the interest of timely
   adoption of the iSCSI protocol, the IPS group has chosen to focus the
   first version of the protocol to work with the existing SCSI
   architecture and commands, and the existing TCP/IP transport layer.
   Both these protocols are widely-deployed and well-understood. The
   thought is that using these mature protocols will entail a minimum of
   new invention, the most rapid possible adoption, and the greatest
   compatibility with Internet architecture, protocols, and equipment.

Conventions used in this document

   This document describes the requirements for a protocol design, but
   does not define a protocol standard. Nevertheless, 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 [2].

Table of Contents

1. Introduction

   The IP Storage Working group is chartered with developing
   comprehensive technology to transport block storage data over IP
   protocols. This effort includes a protocol to transport the Small
   Computer Systems Interface (SCSI) protocol over the Internet (iSCSI).
   The initial version of the iSCSI protocol will define a mapping of
   SCSI transport protocol over TCP/IP so that SCSI storage controllers
   (principally disk and tape arrays and libraries) can be attached to
   IP networks, notably Gigabit Ethernet (GbE) and 10 Gigabit Ethernet
   (10 GbE).

The iSCSI protocol is a mapping of SCSI to TCP, and constitutes a
"SCSI transport" as defined by the ANSI T10 document SCSI SAM-2
document [SAM2, p. 3, "Transport Protocols"].

2. Summary of Requirements

The iSCSI standard:

From section 3.2 Performance/Cost:

   MUST allow implementations to equal or improve on the current
   state of the art for SCSI interconnects.

   MUST enable cost competitive implementations.

   SHOULD minimize control overhead to enable low delay
   communications.

   MUST provide high bandwidth and bandwidth aggregation.

   MUST have low host CPU utilizations, equal to or better than
   current technology.

   MUST be possible to build I/O adapters that handle the entire SCSI
   task.

   SHOULD permit direct data placement architectures.

   MUST NOT impose complex operations on host software.

   MUST provide for full utilization of available link bandwidth.

   MUST allow an implementation to exploit parallelism (multiple
   connections) at the device interfaces and within the interconnect
   fabric.

From section 3.4 High Bandwidth/Bandwidth Aggregation:

   MUST operate over a single TCP connection.

   SHOULD support 'connection binding', and it MUST be optional to
   implement.

From section 4 Ease of Implementation/Complexity of Protocol:

   SHOULD keep the protocol simple.

   SHOULD minimize optional features.

      MUST specify feature negotiation at session establishment (login).

      MUST operate correctly when no optional features are negotiated as
      well as when individual option negotions are unsuccessful.

   From section 5.1 Detection of Data Corruption:

      MUST support a data integrity check format for use in digest
      generation.

      MAY use separate digest for data and headers.

      iSCSI header format SHOULD be extensible to include other data
      integrity digest calculation methods.

   From section 5.2 Recovery:

      MUST specify mechanisms to recover in a timely fashion from
      failures on the initiator, target, or connecting infrastructure.

      MUST specify recovery methods for non-idempotent requests.

      SHOULD take into account fail-over schemes for mirrored targets or
      highly available storage configurations.

      SHOULD provide a method for sessions to be gracefully terminated
      and restarted that can be initiated by either the initiator or
      target.

   From section 6 Interoperability:

      iSCSI protocol document MUST be clear and unambiguous.

   From section 6.1 Internet Infrastructure:

      MUST:
      -- be compatible with both IPv4 and IPv6
      -- use TCP connections conservatively, keeping in mind there may
         be many other users of TCP on a given machine.

      MUST NOT require changes to existing Internet protocols.

      SHOULD minimize required changes to existing TCP/IP
      implementations.

      MUST be designed to allow future substitution of SCTP (for TCP) as
      an IP transport protocol with minimal changes to iSCSI protocol
      operation, protocol data unit (PDU) structures and formats.

   From section 6.2 SCSI:

      Any feature SAM2 requires in a valid transport mapping MUST be
      specified by iSCSI.

      MUST specify strictly ordered delivery of SCSI commands over an
      iSCSI session between an initiator/target pair.

      The command ordering mechanism SHOULD seek to minimize the amount
      of communication necessary across multiple adapters doing
      transport off-load.

      MUST specify for each feature whether it is OPTIONAL, RECOMMENDED
      or REQUIRED to implement and/or use.

      MUST NOT require changes to the SCSI-3 command sets and SCSI
      client code except except where SCSI specifications point to
      "transport dependent" fields and behavior.

      SHOULD track changes to SCSI and the SCSI Architecture Model.

      MUST be capable of supporting all SCSI-3 command sets and device
      types.

      SHOULD support ACA implementation.

      MUST allow for the construction of gateways to other SCSI
      transports

      MUST reliably transport SCSI commands from the initiator to the
      target.

      MUST correctly deal with iSCSI packet drop, duplication,
      corruption, stale packets, and re-ordering.

   From section 7.1 Extensible Security:

      SHOULD require minimal configuration and overhead in the insecure
      operation.

      MUST provide for strong authentication when increased security is
      required.

      SHOULD allow integration of new security mechanisms without
      breaking backwards compatible operation.

   From section 7.2 Authentication:

      MAY support various levels of authentication security.

      MUST support private authenticated login.

      iSCSI authenticated login MUST be resilient against attacks.

      MUST support data origin authentication of its communications;
      data origin authentication MAY be optional to use.

   From section 7.3 Data Integrity:

      SHOULD NOT preclude use of additional data integrity protection
      protocols (IPSec, TLS).

   From section 7.4 Data Confidentiality:

      MUST provide for the use of a data encryption protocol such as TLS
      or IPsec ESP to provide data confidentiality between iSCSI
      endpoints

   From section 8 Management:

      SHOULD be manageable using standard IP-based management protocols.

      iSCSI protocol document MUST NOT define the management
      architecture for iSCSI, or make explicit references to management
      objects such as MIB variables.

   From section 8.1 Naming:

      MUST support the naming architecture of SAM-2. The means by which
      an iSCSI resource is located MUST use or extend existing Internet
      standard resource location methods.

      MUST provide a means of identifying iSCSI targets by a unique
      identifier that is independent of the path on which it is found.

      The format for the iSCSI names MUST use existing naming
      authorities.

      An iSCSI name SHOULD be a human readable string in an
      international character set encoding.

      Standard Internet lookup services SHOULD be used to resolve iSCSI
      names.

SHOULD deal with the complications of the new SCSI security
architecture.

iSCSI naming architecture MUST address support of SCSI 3rd party
operations such as EXTENDED COPY.

From section 8.2 Discovery:

MUST have no impact on the use of current IP network discovery
techniques.

MUST provide some means of determining whether an iSCSI service is
available through an IP address.

SCSI protocol-dependent techniques SHOULD be used for further
discovery beyond the iSCSI layer.

MUST provide a method of discovering, given an IP end point on its
well-known port, the list of SCSI targets available to the
requestor. The use of this discovery service MUST be optional.

From section 9 Internet Accessability.

SHOULD be scrutinized for denial of service issues and they should
be addressed.

From section 9.2 Firewalls and Proxy Servers

SHOULD allow deployment where functional and optimizing middle-
boxes such as firewalls, proxy servers and NATs are present.

use of IP addresses and TCP ports SHOULD be firewall friendly.

From section 9.3 Congestion Control and Transport Selection

MUST be a good network citizen with TCP-compatible congestion
control (as defined in [RFC2914]).

iSCSI implementations MUST NOT use multiple connections as a means
to avoid transport-layer congestion control.

3. iSCSI Design Considerations

3.1. General Discussion

Traditionally, storage controllers (e.g., disk array controllers,
tape library controllers) have supported the SCSI-3 protocol and have
been attached to computers by SCSI parallel bus or Fibre Channel.

The IP infrastructure offers compelling advantages for volume/
block-oriented storage attachment. It offers the opportunity to take
advantage of the performance/cost benefits provided by competition in
the Internet marketplace. This could reduce the cost of storage
network infrastructure by providing economies arising from the need
to install and operate only a single type of network.

In addition, the IP protocol suite offers the opportunity for a rich
array of management, security and QoS solutions. Organizations may
initially choose to operate storage networks based on iSCSI that are
independent of (isolated from) their current data networks except for
secure routing of storage management traffic. These organizations
anticipated benefits from the high performance/cost of IP equipment
and the opportunity for a unified management architecture. As
security and QoS evolve, it becomes reasonable to build combined
networks with shared infrastructure; nevertheless, it is likely that
sophisticated users will choose to keep their storage sub-networks
isolated to afford the best control of security and QoS to ensure a
high-performance environment tuned to storage traffic.

Mapping SCSI over IP also provides:

    -- Extended distance ranges
    -- Connectivity to "carrier class" services that support IP

The following applications for iSCSI are contemplated:

    -- Local storage access, consolidation, clustering and pooling (as
       in the data center)
    -- Network client access to remote storage (eg. a "storage service
       provider")
    -- Local and remote synchronous and asynchronous mirroring between
       storage controllers
    -- Local and remote backup and recovery

iSCSI will support the following topologies:

    -- Point-to-point direct connections
    -- Dedicated storage LAN, consisting of one or more LAN segments
    -- Shared LAN, carrying a mix of traditional LAN traffic plus
       storage traffic
    -- LAN-to-WAN extension using IP routers or carrier-provided "IP
       Datatone"
    -- Private networks and the public Internet

IP LAN-WAN routers may be used to extend the IP storage network to
the wide area, permitting remote disk access (as for a storage
utility), synchronous and asynchronous remote mirroring, and remote

backup and restore (as for tape vaulting). In the WAN, using TCP
end-to-end avoids the need for specialized equipment for protocol
conversion, ensures data reliability, copes with network congestion,
and provides retransmission strategies adapted to WAN delays.

The iSCSI technology deployment will involve the following elements:

(1) Conclusion of a complete protocol standard and supporting
     implementations;
(2) Development of Ethernet storage NICs and related driver and
     protocol software; [NOTE: high-speed applications of iSCSI are
     expected to require significant portions of the iSCSI/TCP/IP
     implementation in hardware to achieve the necessary throughput.]
(3) Development of compatible storage controllers; and
(4) The likely development of translating gateways to provide
     connectivity between the Ethernet storage network and the Fibre
     Channel and/or parallel-bus SCSI domains.
(5) Development of specifications for iSCSI device management such
     as MIBs, LDAP or XML schemas, etc.
(6) Development of management and directory service applications to
     support a robust SAN infrastructure.

Products could initially be offered for Gigabit Ethernet attachment,
with rapid migration to 10 GbE. For performance competitive with
alternative SCSI transports, it will be necessary to implement the
performance path of the full protocol stack in hardware. These new
storage NICs might perform full-stack processing of a complete SCSI
task, analogous to today's SCSI and Fibre Channel HBAs, and might
also support all host protocols that use TCP (NFS, CIFS, HTTP, etc).

The charter of the IETF IP Storage Working Group (IPSWG) describes
the broad goal of mapping SCSI to IP using a transport that has
proven congestion avoidance behavior and broad implementation on a
variety of platforms. Within that broad charter, several transport
alternatives may be considered. Initial IPS work focuses on TCP, and
this requirements document is restricted to that domain of interest.

3.2. Performance/Cost

In general, iSCSI MUST allow implementations to equal or improve on
the current state of the art for SCSI interconnects. This goal
breaks down into several types of requirement:

Cost competitive with alternative storage network technologies:

In order to be adopted by vendors and the user community, the iSCSI
protocol MUST enable cost competitive implementations when compared
to other SCSI transports (Fibre Channel).

Low delay communication:

Conventional storage access is of a stop-and-wait remote procedure
call type. Applications typically employ very little pipelining of
their storage accesses, and so storage access delay directly impacts
performance. The delay imposed by current storage interconnects,
including protocol processing, is generally in the range of 100
microseconds. The use of caching in storage controllers means that
many storage accesses complete almost instantly, and so the delay of
the interconnect can have a high relative impact on overall
performance. When stop-and-wait IO is used, the delay of the
interconnect will affect performance. The iSCSI protocol SHOULD
minimize control overhead, which adds to delay.

Low host CPU utilization, equal to or better than current technology:

For competitive performance, the iSCSI protocol MUST allow three key
implementation goals to be realized:

(1)   iSCSI MUST make it possible to build I/O adapters that handle an
      entire SCSI task, as alternative SCSI transport implementations
      do.
(2)   The protocol SHOULD permit direct data placement ("zero-copy"
      memory architectures, where the I/O adapter reads or writes host
      memory exactly once per disk transaction.
(3)   The protocol SHOULD NOT impose complex operations on the host
      software, which would increase host instruction path length
      relative to alternatives.

Direct data placement (zero-copy iSCSI):

Direct data placement refers to iSCSI data being placed directly "off
the wire" into the allocated location in memory with no intermediate
copies. Direct data placement significantly reduces the memory bus
and I/O bus loading in the endpoint systems, allowing improved
performance. It reduces the memory required for NICs, possibly
reducing the cost of these solutions.

This is an important implementation goal. In an iSCSI system, each
of the end nodes (for example host computer and storage controller)
should have ample memory, but the intervening nodes (NIC, switches)
typically will not.

High bandwidth, bandwidth aggregation:

The bandwidth (transfer rate, MB/sec) supported by storage
controllers is rapidly increasing, due to several factors:

1. Increase in disk spindle and controller performance;
2. Use of ever-larger caches, and improved caching algorithms;
3. Increased scale of storage controllers (number of supported
    spindles, speed of interconnects).

The iSCSI protocol MUST provide for full utilization of available
link bandwidth. The protocol MUST also allow an implementation to
exploit parallelism (multiple connections) at the device interfaces
and within the interconnect fabric.

The next two sections further discuss the need for direct data
placement and high bandwidth.

3.3. Framing

Framing refers to the addition of information in a header, or the
data stream to allow implementations to locate the boundaries of an
iSCSI protocol data unit (PDU) within the TCP byte stream. There are
two technical requirements driving framing: interfacing needs, and
accelerated processing needs.

A framing solution that addresses the "interfacing needs" of the
iSCSI protocol will facilitate the implementation of a message-based
upper layer protocol (iSCSI) on top of an underlying byte streaming
protocol (TCP). Since TCP is a reliable transport, this can be
accomplished by including a length field in the iSCSI header. Finding
the protocol frame assumes that the receiver will parse from the
beginning of the TCP data stream, and never make a mistake (lose
alignment on packet headers).

The other technical requirement for framing, "accelerated
processing", stems from the need to handle increasingly higher data
rates in the physical media interface. Two needs arise from higher
data rates:

(1) LAN environment - NIC vendors seek ways to provide "zero-copy"
     methods of moving data directly from the wire into application
     buffers.

(2) WAN environment- the emergence of high bandwidth, high latency,
     low bit error rate physical media places huge buffer
     requirements on the physical interface solutions.

First, vendors are producing network processing hardware that
offloads network protocols to hardware solutions to achieve higher
data rates. The concept of "zero-copy" seeks to store blocks of data
in appropriate memory locations (aligned) directly off the wire, even
when data is reordered due to packet loss. This is necessary to
drive actual data rates of 10 Gigabit/sec and beyond.

Secondly, in order for iSCSI to be successful in the WAN arena it
must be possible to operate efficiently in high bandwidth, high delay
networks. The emergence of multi-gigabit IP networks with latencies
in the tens to hundreds of milliseconds presents a challenge. To
fill such large pipes, it is necessary to have tens of megabytes of
outstanding requests from the application. In addition, some
protocols potentially require tens of megabytes at the transport
layer to deal with buffering for reassembly of data when packets are
received out-of-order.

In both cases, the issue is the desire to minimize the amount of
memory and memory bandwidth required for iSCSI hardware solutions.

Consider that a network pipe at 10 Gbps x 200 msec holds 250 MB.
[Assume land-based communication with a spot half way around the
world at the equator. Ignore additional distance due to cable
routing. Ignore repeater and switching delays; consider only a
speed-of-light delay of 5 microsec/km. The circumference of the
globe at the equator is approx. 40000 km (round-trip delay must be
considered to keep the pipe full). 10 Gb/sec x 40000 km x 5
microsec/km x B / 8b = 250 MB]. In a conventional TCP
implementation, loss of a TCP segment means that stream processing
MUST stop until that segment is recovered, which takes at least a
time of <network round trip> to accomplish. Following the example
above, an implementation would be obliged to catch 250 MB of data
into an anonymous buffer before resuming stream processing; later,
this data would need to be moved to its proper location. Some
proponents of iSCSI seek some means of putting data directly where it
belongs, and avoiding extra data movement in the case of segment
drop. This is a key concept in understanding the debate behind
framing methodologies.

The framing of the iSCSI protocol impacts both the "interfacing
needs" and the "accelerated processing needs", however, while
including a length in a header may suffice for the "interfacing
needs", it will not serve the direct data placement needs. The
framing mechanism developed should allow resynchronization of packet
boundaries even in the case where a packet is temporarily missing in
the incoming data stream.

3.4. High bandwidth, bandwidth aggregation

   At today's block storage transport throughput, any single link can be
   saturated by the volume of storage traffic. Scientific data
   applications and data replication are examples of storage
   applications that push the limits of throughput.

   Some applications, such as log updates, streaming tape, and
   replication, require ordering of updates and thus ordering of SCSI
   commands. An initiator may maintain ordering by waiting for each
   update to complete before issuing the next (a.k.a. synchronous
   updates). However, the throughput of synchronous updates decreases
   inversely with increases in network distances.

   For greater throughput, the SCSI task queuing mechanism allows an
   initiator to have multiple commands outstanding at the target
   simultaneously and to express ordering constraints on the execution
   of those commands. The task queuing mechanism is only effective if
   the commands arrive at the target in the order they were presented to
   the initiator (FIFO order). The iSCSI standard must provide an
   ordered transport of SCSI commands, even when commands are sent along
   different network paths (see Section 5.2 SCSI). This is referred to
   as "command ordering".

   The iSCSI protocol MUST operate over a single TCP connection to
   accommodate lower cost implementations. To enable higher performance
   storage devices, the protocol should specify a means to allow
   operation over multiple connections while maintaining the behavior of
   a single SCSI port. This would allow the initiator and target to use
   multiple network interfaces and multiple paths through the network
   for increased throughput. There are a few potential ways to satisfy
   the multiple path and ordering requirements.

   A popular way to satisfy the multiple-path requirement is to have a
   driver above the SCSI layer instantiate multiple copies of the SCSI
   transport, each communicating to the target along a different path.
   "Wedge" drivers use this technique today to attain high performance.
   Unfortunately, wedge drivers must wait for acknowledgement of
   completion of each request (stop-and-wait) to ensure ordered updates.

   Another approach might be for iSCSI protocol to use multiple
   instances of its underlying transport (e.g. TCP). The iSCSI layer
   would make these independent transport instances appear as one SCSI
   transport instance and maintain the ability to do ordered SCSI
   command queuing. The document will refer to this technique as
   "connection binding" for convenience.

The iSCSI protocol SHOULD support connection binding, and it MUST be optional to implement.

In the presence of connection binding, there are two ways to assign features to connections. In the symmetric approach, all the connections are identical from a feature standpoint. In the asymmetric model, connections have different features. For example, some connections may be used primarily for data transfers whereas others are used primarily for SCSI commands.

Since the iSCSI protocol must support the case where there was only one transport connection, the protocol must have command, data, and status travel over the same connection.

In the case of multiple connections, the iSCSI protocol must keep the command and its associated data and status on the same connection (connection allegiance). Sending data and status on the same connection is desirable because this guarantees that status is received after the data (TCP provides ordered delivery). In the case where each connection is managed by a separate processor, allegiance decreases the need for inter-processor communication. This symmetric approach is a natural extension of the single connection approach.

An alternate approach that was extensively discussed involved sending all commands on a single connection and the associated data and status on a different connection (asymmetric approach). In this scheme, the transport ensures the commands arrive in order. The protocol on the data and status connections is simpler, perhaps lending itself to a simpler realization in hardware. One disadvantage of this approach is that the recovery procedure is different if a command connection fails vs. a data connection. Some argued that this approach would require greater inter-processor communication when connections are spread across processors.

The reader may reference the mail archives of the IPS mailing list between June and September of 2000 for extensive discussions on symmetric vs asymmetric connection models.

4. Ease of implementation/complexity of protocol

Experience has shown that adoption of a protocol by the Internet community is inversely proportional to its complexity. In addition, the simpler the protocol, the easier it is to diagnose problems. The designers of iSCSI SHOULD strive to fulfill the requirements of the creating a SCSI transport over IP, while keeping the protocol as simple as possible.

In the interest of simplicity, iSCSI SHOULD minimize optional
features. When features are deemed necessary, the protocol MUST
specify feature negotiation at session establishment (login). The
iSCSI transport MUST operate correctly when no optional features are
negotiated as well as when individual option negotiations are
unsuccessful.

5. Reliability and Availability

5.1. Detection of Data Corruption

   There have been several research papers that suggest that the TCP
   checksum calculation allows a certain number of bit errors to pass
   undetected [10] [11].

   In order to protect against data corruption, the iSCSI protocol MUST
   support a data integrity check format for use in digest generation.

   The iSCSI protocol MAY use separate digests for data and headers. In
   an iSCSI proxy or gateway situation, the iSCSI headers are removed
   and re-built, and the TCP stream is terminated on either side. This
   means that even the TCP checksum is removed and recomputed within the
   gateway. To ensure the protection of commands, data, and status the
   iSCSI protocol MUST include a CRC or other digest mechanism that is
   computed on the SCSI data block itself, as well as on each command
   and status message. Since gateways may strip iSCSI headers and
   rebuild them, a separate header CRC is required. Two header digests,
   one for invariant portions of the header (addresses) and one for the
   variant portion would provide protection against changes to portions
   of the header that should never be changed by middle boxes (eg,
   addresses).

   The iSCSI header format SHOULD be extensible to include other digest
   calculation methods.

5.2. Recovery

   The SCSI protocol was originally designed for a parallel bus
   transport that was highly reliable. SCSI applications tend to assume
   that transport errors never happen, and when they do, SCSI
   application recovery tends to be expensive in terms of time and
   computational resources.

   iSCSI protocol design, while placing an emphasis on simplicity, MUST
   lead to timely recovery from failure of initiator, target, or
   connecting network infrastructure (cabling, data path equipment such
   as routers, etc).

iSCSI MUST specify recovery methods for non-idempotent requests, such as operations on tape drives.

The iSCSI protocol error recover mechanism SHOULD take into account fail-over schemes for mirrored targets or highly available storage configurations that provide paths to target data through multiple "storage servers". This would provide a basis for layered technologies like high availability and clustering.

The iSCSI protocol SHOULD also provide a method for sessions to be gracefully terminated and restarted that can be initiated by either the initiator or target. This provides the ability to gracefully fail over an initiator or target, or reset a target after performing maintenance tasks such as upgrading software.

6. Interoperability

It must be possible for initiators and targets that implement the required portions of the iSCSI specification to interoperate. While this requirement is so obvious that it doesn't seem worth mentioning, if the protocol specification contains ambiguous wording, different implementations may not interoperate. The iSCSI protocol document MUST be clear and unambiguous.

6.1. Internet infrastructure

The iSCSI protocol MUST:

   -- be compatible with both IPv4 and IPv6.
   -- use TCP connections conservatively, keeping in mind there may
      be many other users of TCP on a given machine.

The iSCSI protocol MUST NOT require changes to existing Internet protocols and SHOULD minimize required changes to existing TCP/IP implementations.

iSCSI MUST be designed to allow future substitution of SCTP (for TCP) as an IP transport protocol with minimal changes to iSCSI protocol operation, protocol data unit (PDU) structures and formats. Although not widely implemented today, SCTP has many design features that make it a desirable choice for future iSCSI enhancement.

6.2. SCSI

In order to be considered a SCSI transport, the iSCSI standard must comply with the requirements of the SCSI Architecture Model [SAM-2] for a SCSI transport. Any feature SAM2 requires in a valid transport mapping MUST be specified by iSCSI. The iSCSI protocol document MUST

specify for each feature whether it is OPTIONAL, RECOMMENDED or
REQUIRED to implement and/or use.

The SCSI Architectural Model [SAM-2] indicates an expectation that
the SCSI transport provides ordering of commands on an initiator
target-LUN granularity. There has been much discussion on the IPS
reflector and in working group meetings regarding the means to ensure
this ordering. The rough consensus is that iSCSI MUST specify
strictly ordered delivery of SCSI commands over an iSCSI session
between an initiator/target pair, even in the presence of transport
errors. This command ordering mechanism SHOULD seek to minimize the
amount of communication necessary across multiple adapters doing
transport off-load. If an iSCSI implementation does not require
ordering it can instantiate multiple sessions per initiator-target
pair.

iSCSI is intended to be a new SCSI "transport" [SAM2]. As a mapping
of SCSI over TCP, iSCSI requires interaction with both T10 and IETF.
However, the iSCSI protocol MUST NOT require changes to the SCSI-3
command sets and SCSI client code except where SCSI specifications
point to "transport dependent" fields and behavior. For example,
changes to SCSI documents will be necessary to reflect lengthier
iSCSI target names and potentially lengthier timeouts. Collaboration
with T10 will be necessary to achieve this requirement.

The iSCSI protocol SHOULD track changes to SCSI and the SCSI
Architecture Model.

The iSCSI protocol MUST be capable of supporting all SCSI-3 command
sets and device types. The primary focus is on supporting 'larger'
devices: host computers and storage controllers (disk arrays, tape
libraries). However, other command sets (printers, scanners) must be
supported. These requirements MUST NOT be construed to mean that
iSCSI must be natively implementable on all of today's SCSI devices,
which might have limited processing power or memory.

ACA (Auto Contingent Allegiance) is an optional SCSI mechanism that
stops execution of a sequence of dependent SCSI commands when one of
them fails. The situation surrounding it is complex - T10 specifies
ACA in SAM2, and hence iSCSI must support it and endeavor to make
sure that ACA gets implemented sufficiently (two independent
interoperable implementations) to avoid dropping ACA in the
transition from Proposed Standard to Draft Standard. This implies
iSCSI SHOULD support ACA implementation.

The iSCSI protocol MUST allow for the construction of gateways to
other SCSI transports, including parallel SCSI [SPI-X] and to SCSI
FCP[FCP, FCP-2]. It MUST be possible to construct "translating"

gateways so that iSCSI hosts can interoperate with SCSI-X devices; so
that SCSI-X devices can communicate over an iSCSI network; and so
that SCSI-X hosts can use iSCSI targets (where SCSI-X refers to
parallel SCSI, SCSI-FCP, or SCSI over any other transport). This
requirement is implied by support for SAM-2, but is worthy of
emphasis. These are true application protocol gateways, and not just
bridge/routers. The different standards have only the SCSI-3 command
set layer in common. These gateways are not mere packet forwarders.

The iSCSI protocol MUST reliably transport SCSI commands from the
initiator to the target. According to [SAM-2, p. 17.] "The function
of the service delivery subsystem is to transport an error-free copy
of the request or response between the sender and the receiver"
[SAM-2, p. 22]. The iSCSI protocol MUST correctly deal with iSCSI
packet drop, duplication, corruption, stale packets, and re-ordering.

7. Security Considerations

In the past, directly attached storage systems have implemented
minimal security checks because the physical connection offered
little chance for attack. Transporting block storage (SCSI) over IP
opens a whole new opportunity for a variety of malicious attacks.
Attacks can take the active form (identity spoofing, man-in-the-
middle) or the passive form (eavesdropping).

7.1. Extensible Security

The security services required for communications depends on the
individual network configurations and environments. Organizations
are setting up Virtual Private Networks(VPN), also known as
Intranets, that will require one set of security functions for
communications within the VPN and possibly many different security
functions for communications outside the VPN to support
geographically separate components. The iSCSI protocol is applicable
to a wide range of internet working environments that may employ
different security policies. iSCSI MUST provide for strong
authentication when increased security is required. The protocol
SHOULD require minimal configuration and overhead in the insecure
operation, and allow integration of new security mechanisms without
breaking backwards compatible operation.

7.2. Authentication

The iSCSI protocol MAY support various levels of authentication
security, ranging from no authentication to secure authentication
using public or private keys.

The iSCSI protocol MUST support private authenticated login.

Authenticated login aids the target in blocking the unauthorized use
of SCSI resources. "Private" authenticated login mandates protected
identity exchange (no clear text passwords at a minimum). Since
block storage confidentiality is considered critical in enterprises
and many IP networks may have access holes, organizations will want
to protect their iSCSI resources.

The iSCSI authenticated login MUST be resilient against attacks since
many IP networks are vulnerable to packet inspection.

In addition, the iSCSI protocol MUST support data origin
authentication of its communications; data origin authentication MAY
be optional to use. Data origin authentication is critical since IP
networks are vulnerable to source spoofing, where a malicious third
party pretends to send packets from the initiator's IP address. These
requirements should be met using standard Internet protocols such as
IPsec or TLS. The endpoints may negotiate the authentication method,
optionally none.

7.3. Data Integrity

The iSCSI protocol SHOULD NOT preclude use of additional data
integrity protection protocols (IPSec, TLS).

7.4. Data Confidentiality

Block storage is used for storing sensitive information, where data
confidentiality is critical. An application may encrypt the data
blocks before writing them to storage - this provides the best
protection for the application. Even if the storage or
communications are compromised, the attacker will have difficulty
reading the data.

In certain environments, encryption may be desired to provide an
extra assurance of confidentiality. An iSCSI implementation MUST
provide for the use of a data encryption protocol such as TLS or
IPsec ESP to provide data confidentiality between iSCSI endpoints.

8. Management

iSCSI implementations SHOULD be manageable using standard IP-based
management protocols. However, the iSCSI protocol document MUST NOT
define the management architecture for iSCSI within the network
infrastructure. iSCSI will be yet another resource service within a
complex environment of network resources (printers, file servers,
NAS, application servers, etc). There will certainly be efforts to
design how the "block storage service" that iSCSI devices provide is
integrated into a comprehensive, shared model, network management

environment. A "network administrator" (or "storage administrator")
will desire to have integrated applications for assigning user names,
resource names, etc. and indicating access rights. iSCSI devices
presumably will want to interact with these integrated network
management applications. The iSCSI protocol document will not
attempt to solve that set of problems, or specify means for devices
to provide management agents. In fact, there should be no mention of
MIBs or any other means of managing iSCSI devices as explicit
references in the iSCSI protocol document, because management data
and protocols change with the needs of the environment and the
business models of the management applications.

8.1. Naming

   Whenever possible, iSCSI MUST support the naming architecture of
   SAM-2. Deviations and uncertainties MUST be made explicit, and
   comments and resolutions worked out between ANSI T10 and the IPS
   working group.

   The means by which an iSCSI resource is located MUST use or extend
   existing Internet standard resource location methods. RFC 2348 [12]
   specifies URL syntax and semantics which should be sufficiently
   extensible for the iSCSI resource.

   The iSCSI protocol MUST provide a means of identifying an iSCSI
   storage device by a unique identifier that is independent of the path
   on which it is found. This name will be used to correlate alternate
   paths to the same device. The format for the iSCSI names MUST use
   existing naming authorities, to avoid creating new central
   administrative tasks. An iSCSI name SHOULD be a human readable
   string in an international character set encoding.

   Standard Internet lookup services SHOULD be used to resolve names.
   For example, Domain Name Services (DNS) MAY be used to resolve the
   <hostname> portion of a URL to one or multiple IP addresses. When a
   hostname resolves to multiple addresses, these addresses should be
   equivalent for functional (possibly not performance) purposes. This
   means that the addresses can be used interchangeably as long as
   performance isn't a concern. For example, the same set of SCSI
   targets MUST be accessible from each of these addresses.

   An iSCSI device naming scheme MUST interact correctly with the
   proposed SCSI security architecture [99-245r9]. Particular attention
   must be directed to the proxy naming architecture defined by the new
   security model. In this new model, a host is identified by an
   Access ID, and SCSI Logical Unit Numbers (LUNs) can be mapped in a
   manner that gives each AccessID a unique LU map. Thus, a given LU
   within a target may be addressed by different LUNs.

The iSCSI naming architecture MUST address support of SCSI 3rd party operations such as EXTENDED COPY. The key issue here relates to the naming architecture for SCSI LUs - iSCSI must provide a means of passing a name or handle between parties. iSCSI must specify a means of providing a name or handle that could be used in the XCOPY command and fit within the available space allocated by that command. And it must be possible, of course, for the XCOPY target (the third party) to de-reference the name to the correct target and LU.

8.2. Discovery

iSCSI MUST have no impact on the use of current IP network discovery techniques. Network management platforms discover IP addresses and have various methods of probing the services available through these IP addresses. An iSCSI service should be evident using similar techniques.

The iSCSI specifications MUST provide some means of determining whether an iSCSI service is available through an IP address. It is expected that iSCSI will be a point of service in a host, just as SNMP, etc are points of services, associated with a well known port number.

SCSI protocol-dependent techniques SHOULD be used for further discovery beyond the iSCSI layer. Discovery is a complex, multi-layered process. The SCSI protocol specifications provide specific commands for discovering LUs and the commands associated with this process will also work over iSCSI.

The iSCSI protocol MUST provide a method of discovering, given an IP end point on its well-known port, the list of SCSI targets available to the requestor. The use of this discovery service MUST be optional.

Further discovery guidelines are outside the scope of this document and may be addressed in separate Informational documents.

9. Internet Accessibility

9.1. Denial of Service

As with all services, the denial of service by either incorrect implementations or malicious agents is always a concern. All aspects of the iSCSI protocol SHOULD be scrutinized for potential denial of service issues, and guarded against as much as possible.

9.2. NATs, Firewalls and Proxy servers

   NATs (Network Address Translator), firewalls, and proxy servers are a
   reality in today's Internet. These devices present a number of
   challenges to device access methods being developed for iSCSI. For
   example, specifying a URL syntax for iSCSI resource connection allows
   an initiator to address an iSCSI target device both directly and
   through an iSCSI proxy server or NAT. iSCSI SHOULD allow deployment
   where functional and optimizing middle-boxes such as firewalls, proxy
   servers and NATs are present.

   The iSCSI protocol's use of IP addressing and TCP port numbers MUST
   be firewall friendly. This means that all connection requests should
   normally be addressed to a specific, well-known TCP port. That way,
   firewalls can filter based on source and destination IP addresses,
   and destination (target) port number. Additional TCP connections
   would require different source port numbers (for uniqueness), but
   could be opened after a security dialogue on the control channel.

   It's important that iSCSI operate through a firewall to provide a
   possible means of defending against Denial of Service (DoS) assaults
   from less-trusted areas of the network. It is assumed that a
   firewall will have much greater processing power for dismissing bogus
   connection requests than end nodes.

9.3. Congestion Control and Transport Selection

   The iSCSI protocol MUST be a good network citizen with proven
   congestion control (as defined in [RFC2914]). In addition, iSCSI
   implementations MUST NOT use multiple connections as a means to avoid
   transport-layer congestion control.

10. Definitions

   Certain definitions are offered here, with references to the original
   document where applicable, in order to clarify the discussion of
   requirements. Definitions without references are the work of the
   authors and reviewers of this document.

   Logical Unit (LU): A target-resident entity that implements a device
   model and executes SCSI commands sent by an application client [SAM-
   2, sec. 3.1.50, p. 7].

   Logical Unit Number (LUN): A 64-bit identifier for a logical unit
   [SAM-2, sec. 3.1.52, p. 7].

SCSI Device: A device that is connected to a service delivery
subsystem and supports a SCSI application protocol [SAM-2, sec.
3.1.78, p. 9].

Service Delivery Port (SDP): A device-resident interface used by the
application client, device server, or task manager to enter and
retrieve requests and responses from the service delivery subsystem.
Synonymous with port (SAM-2 sec. 3.1.61) [SAM-2, sec. 3.1.89, p. 9].

Target: A SCSI device that receives a SCSI command and directs it to
one or more logical units for execution [SAM-2 sec. 3.1.97, p. 10].

Task: An object within the logical unit representing the work
associated with a command or a group of linked commands [SAM-2, sec.
3.1.98, p. 10].

Transaction: A cooperative interaction between two objects, involving
the exchange of information or the execution of some service by one
object on behalf of the other [SAM-2, sec. 3.1.109, p. 10].

11. References

1.   Bradner, S., "The Internet Standards Process -- Revision 3", BCP
      9, RFC 2026, October 1996.

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

3.   [SAM-2] ANSI NCITS. Weber, Ralph O., editor. SCSI Architecture
      Model -2 (SAM-2). T10 Project 1157-D. rev 23, 16 Mar 2002.

4.   [SPC-2] ANSI NCITS. Weber, Ralph O., editor. SCSI Primary
      Commands 2 (SPC-2). T10 Project 1236-D. rev 20, 18 July
      2001.

5.   [CAM-3] ANSI NCITS. Dallas, William D., editor. Information
      Technology - Common Access Method - 3 (CAM-3)). X3T10 Project
      990D. rev 3, 16 Mar 1998.

6.   [99-245r8] Hafner, Jim. A Detailed Proposal for Access
      Controls. T10/99-245 revision 9, 26 Apr 2000.

7.   [SPI-X] ANSI NCITS. SCSI Parallel Interface - X.

8.   [FCP] ANSI NCITS. SCSI-3 Fibre Channel Protocol [ANSI
      X3.269:1996].

   9.   [FCP-2] ANSI NCITS. SCSI-3 Fibre Channel Protocol - 2
        [T10/1144-D].

   10. Paxon, V. End-to-end internet packet dynamics, IEEE Transactions
        on Networking 7,3 (June 1999) pg 277-292.

   11. Stone J., Partridge, C. When the CRC and TCP checksum disagree,
        ACM Sigcomm (Sept. 2000).

   12. Malkin, G. and A. Harkin, "TFTP Blocksize Option", RFC 2348, May
        1998.

   13. Floyd, S., "Congestion Control Principles", BCP 14, RFC 2914,
        September 2000.

12. Acknowledgements

   Special thanks to Julian Satran, IBM and David Black, EMC for their
   extensive review comments.

13. Author's Addresses

   Address comments to:

   Marjorie Krueger
   Hewlett-Packard Corporation
   8000 Foothills Blvd
   Roseville, CA 95747-5668, USA
   Phone: +1 916 785-2656
   EMail: marjorie_krueger@hp.com

   Randy Haagens
   Hewlett-Packard Corporation
   8000 Foothills Blvd
   Roseville, CA 95747-5668, USA
   Phone: +1 916 785-4578
   EMail: Randy_Haagens@hp.com

   Costa Sapuntzakis
   Stanford University
   353 Serra Mall Dr #407
   Stanford, CA 94305
   Phone: 650-723-2458
   EMail: csapuntz@stanford.edu

   Mark Bakke
   Cisco Systems, Inc.
   6450 Wedgwood Road
   Maple Grove, MN 55311
   Phone: +1 763 398-1054
   EMail: mbakke@cisco.com

14. Full Copyright Statement

Acknowledgement