

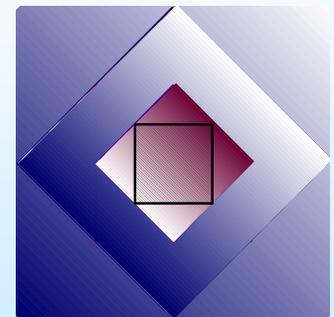
iSCSI – a SCSI over TCP mapping

IETF - 48



Julian Satran

IBM Research Lab in Haifa





Objective

- ◆ Performance
 - ◆ Interconnect will meet or exceed current storage needs and enable growth
 - ◆ High bandwidth, minimum latency
- ◆ Availability
 - ◆ Enable various levels of recovery
- ◆ Cost
 - ◆ Reuse whatever available
- ◆ A good network citizen
 - ◆ Congestion control
 - ◆ Security



Design goals

- ◆ Minimalist design
- ◆ Enable efficient implementations
- ◆ No or minimal options
- ◆ Features can be ignored without affecting interoperability – no negotiation or setting beyond that mandated by SCSI for basic functions
- ◆ Clear layering of functions
 - ◆ SCSI
 - ◆ iSCSI
 - ◆ transport/delivery network



Basic mechanisms

- ◆ Sessions and connections
- ◆ Login, authentication and security
- ◆ Commands, messages, tasks and tags
- ◆ Ordered delivery – the numbering scheme
- ◆ The response numbering scheme
- ◆ Recovery
- ◆ Each of the basic mechanisms has a minimal functionality that must be implemented



Sessions and connections

- ◆ A session is a set of TCP connections linking an initiator and a target
- ◆ It is long lived
- ◆ It is meant to provide bandwidth and availability
 - ◆ several connections = more bandwidth
 - ◆ Several connections = better availability provided fail over is enabled
- ◆ The initiator is supposed to be able to use any connection to execute a command and keep all the command related packets on the connection it started the command (connection allegiance)
- ◆ *Minimum requirement – 1 TCP connection/session*



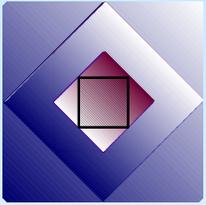
Login, authentication and security

- ◆ Login has multiple functions:
 - ◆ session building
 - ◆ authentication and security
 - ◆ some parameter negotiations
- ◆ Authentication and security
 - ◆ none
 - ◆ authentication only
 - ◆ authentication & encryption – IPsec & TLS
- ◆ *Minimal implementation – session building*



Commands, messages, tasks and tags

- ◆ Commands & Responses are mappings of SCSI and carry an initiator-wide unique tag
- ◆ The initiator tag helps relate all the elements of a command (command, optional data and response)
- ◆ Messages are used to:
 - ◆ manipulate tasks or check/set the whole SCSI/iSCSI path (in which case they carry also a tag)
 - ◆ Check only the iSCSI transport in which case they don't carry a tag
- ◆ *A compliant implementation must support all command and message types but can choose to ignore parameters as outlined in the draft or reject them*



3 stages in the life of a SCSI command

- ◆ Initiation – delivery from initiator SCSI layer to target SCSI layer
- ◆ Execution – optional data transfer and RTT
 - ◆ To keep latency at a minimum iSCSI permits unsolicited data transmission as immediate data (attached to the command) or in separate packets
 - ◆ *To enable recovery iSCSI mandates honoring target issued RTT*
- ◆ Status transmission from target SCSI to initiator SCSI (*status recovery is enabled but not mandated*)



Ordered delivery – the numbering scheme

- ◆ iSCSI enables ordered delivery of commands and tagged messages from initiator to target over several distinct TCP connections
- ◆ The model used by a target that chooses to implement ordered execution is that of an iSCSI staging-area from which commands are delivered to a SCSI device-server only after all preceding commands have been delivered
- ◆ A sliding window mechanism is used to keep the staging-area within bounds



Ordered delivery – the numbering scheme (cont.)

- ◆ The command numbers are significant only while commands are within the staged area
- ◆ The only unique identifier for the life of a command is the initiator tag
- ◆ *Ordered delivery is not mandatory – although initiators supporting several connections per session should implement it*
- ◆ *Targets supporting several connections per session should indicate the support/lack-of-support for ordered delivery (how?)*



The response numbering scheme

- ◆ Responses are also subject to numbering
- ◆ The main purpose of response numbering is bulk response acknowledgement
- ◆ Response acknowledgement enables a target to discard whatever residual information it has about a task after the response is acked
- ◆ *Response numbering and acknowledgement is mandatory*



Recovery

- ◆ Several levels of recovery are enabled by iSCSI:
 - ◆ error notification – broken connections lead to SCSI command failure
 - ◆ command restart – commands pending or in progress on a broken connection can be restarted on a new or one of the remaining connections in a session
 - ◆ data transmission restart – commands can be restarted from a known point in the data transfer (mainly important for long operations)



Recovery (cont.)

- ◆ Failure to handle data and deadlock avoidance – data can be dropped by targets and reacquired by RTT
- ◆ The design aim is to enable a session to stay operational as long as a single TCP link can be maintained/established
- ◆ *The mandatory recovery support is error notification and data replay on request (RTT)*



Additional mechanisms

- ◆ Text commands
 - ◆ enable parameter negotiations and vendor unique extensions
- ◆ Mapping
 - ◆ an aliasing mechanism for string mapping into 8 byte “normal” SCSI addresses
 - ◆ to be used for third party naming and access control
- ◆ *Text commands and mapping may be rejected(not implemented)*



Not yet polished enough

- ◆ Login / authentication / security
 - ◆ Completely specify including all or elements of new submitted documents (SANRAD)
 - ◆ Register a profile with SLAS?
- ◆ Text parameters
- ◆ Rationale/explanations/implementer notes/state diagrams
- ◆ RDMA/Synch Recovery