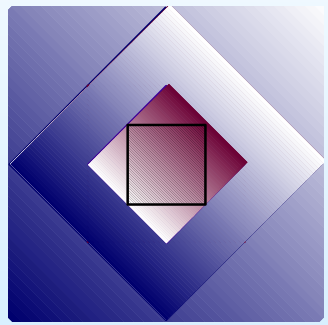
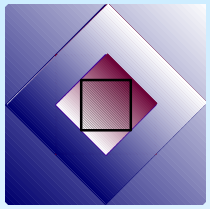
iSCSI – a brief introduction

April 2002

Julian Satran

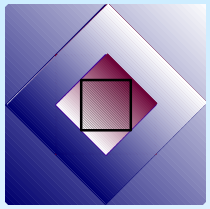
IBM Research Lab in Haifa





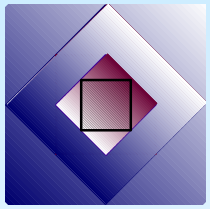
Why?

- Standard networking infrastructure is good enough for storage
- Leverage existing networking infrastructure
  - equipment
  - management
  - skills



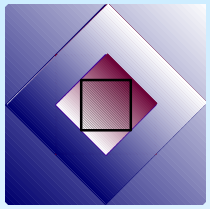
What standards handle IP Storage

- iSCSI native SCSI over IP
- FCIP – connect FC islands via TCP connections – requires FC switches within islands
- iFCP connect FCP devices to an IP network (enable mixture of devices) and may require special switches (if devices are not iFCP switch ports may be)
- Companion standards – for naming (NDT, iSNS, SLP), management (iSNS, SNMP, MIBs), common encapsulation (FCP, iFCP)



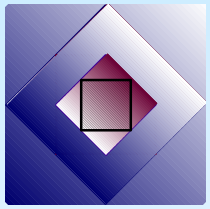
Objective

- 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 guidelines

- 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

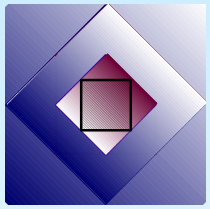


Basic mechanisms

- 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

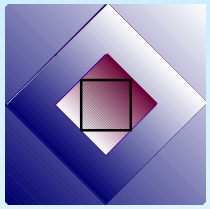
Sessions and connections

- A session is a set of TCP connections linking an initiator and a target
- It is long lived – logical equivalent of a cable
- It is meant to provide bandwidth and availability
  - several connections = more bandwidth
  - several connections = better availability provided fail over is enabled
  - several connections = some complexity
- 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*



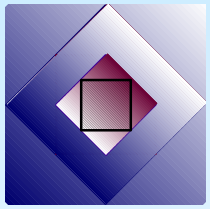
Login, authentication and security

- Login has multiple functions:
  - session building – using names
  - authentication and security
  - some parameter negotiations
- *minimal implementation – session building*



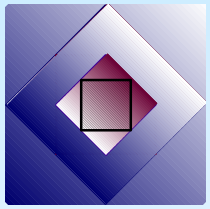
iSCSI naming

- iSCSI uses a naming structure involving:
  - Node names
  - Portal groups
  - Session identifiers (ISID)
  - Session handles
- Clearly separates entity names from addresses
- Multihoming possible but not apparent in the protocol
- Accessed entities are named rather than addressed



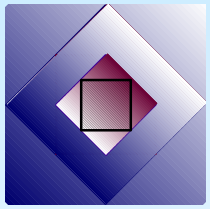
Text key=value negotiation

- Text key=value negotiation is performed at Login and during Full Feature Phase
- Negotiation rules are simple and require a single request response exchange
- Can be initiated by initiator or target
- Most parameters have defaults but negotiations are stateless – i.e., outcome is dictated only by values presented
- A negotiation session is atomic and results are committed/aborted at the sequence-end
- Sequences are ended by “consensus” (the F bit)



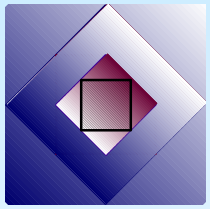
Security

- Security elements:
  - Authentication – mandatory to implement(SRP or xx)
  - IPsec mandatory to implement tunnel mode
- Good structure for current and future networks
- Non cryptographic data integrity iSCSI - CRC



Requests, Responses, messages, tasks and tags

- Requests & Responses are used for SCSI and iSCSI functions and carry an initiator-wide unique tag
- The initiator task tag helps relate all the elements of a command (command, optional data and response)
- iSCSI requests and responses 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
- Unidirectional messages are used for notifications
- *A compliant implementation must support all command and message types but can choose to ignore parameters as outlined in the draft or reject them*



iSCSI PDU structure

- A basic, fixed length (48 bytes) header
  - Good enough for most of the requests and all responses
  - A variable number/variable length set of additional header segments (optional)
  - An optional data segment carrying payload and/or parameters
  - The header-set and data are optionally followed by a CRC
- All common elements are located at the same location in all headers

3 stages in the life of a SCSI command

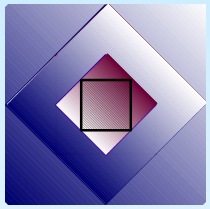
- Initiation – delivery from initiator SCSI layer to target SCSI layer
- Execution – optional data transfer and R2T
  - to keep latency at a minimum iSCSI permits unsolicited data transmission as immediate data (attached to the command) or in separate packets
  - *to enable recovery iSCSI mandates honoring target issued R2T*
- 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 command 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

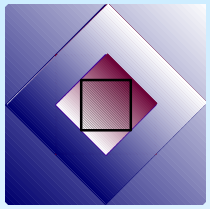
Ordered delivery – the numbering scheme (cont.)

- 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*



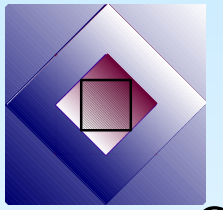
Alternatives considered

- An asymmetric scheme
- 1 control connection + several data connections
- The command connection can fail over on another connection



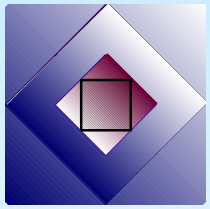
The response numbering scheme

- 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*



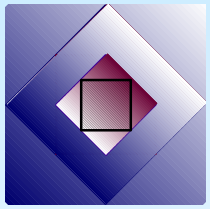
Recovery

- 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)



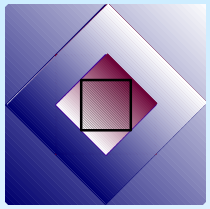
Recovery (cont.)

- failure to handle data and deadlock avoidance – data can be dropped by targets and reacquired by R2T
- 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



Additional mechanisms

- Synch and Steering
  - Data loss or reordering may require resynchronization (recover iSCSI PDU boundary)
  - A simple scheme based on Fixed Interval Markers was selected to accomplish it