

## SVP Overview

SCSI VI Protocol Overview

Permission is granted to members of NCITS, its technical committees, and their associated task groups to reproduce this document for the purposes of NCITS standardization activities without further permission, provided this notice is included. All other rights reserved. Any commercial or for-profit duplication is strictly prohibited.

Edward A. Gardner  
Ophidian Designs  
1262 Hofstead Terrace  
Colorado Springs, CO 80907

eag@ophidian.com  
+1 719 593-8866


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    1



## Agenda

- **VI Architecture Summary**
- **Command Mapping**
- **Flow Control**
- **Miscellaneous Issues**
- **Future Schedule**


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    2



## VI Architecture Summary

- **Virtual Interfaces**
- **VI Connections**
- **Memory Registration**
- **Data Transfer Operations**
- **Reliability Levels**
- **References**


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    3



## Virtual Interfaces

- A VI (Virtual Interface) consists of a send Work Queue and a receive Work Queue.
- Send Work Queue descriptors identify requests.
- Receive Work Queue descriptors identify receive buffers.
- Completion Queues (optional) allow shared processing of multiple Work Queues.
  - Not relevant to SVP.


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    4



## VI Connections

- An association between a pair of VIs that allows messages (packets) sent on one VI to be received at the other.
- Connection establishment:
  - Asymmetric: client requests connection to server.
  - Symmetric: defined as an extension (post v1.0).
    - Not relevant to SVP.
- Connection operation:
  - Symmetric: either endpoint may request any data transfer operation supported on that connection.


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    5



## Memory Registration

- VI Consumer (e.g. SVP Initiator) identifies virtual memory regions to VI Provider (NIC).
  - Locks pages in physical memory.
- Returns memory handle for region.
  - Memory access requires handle, virtual address (equivalent to offset) and length.
- Memory regions may be registered once, then used many times (e.g. by many SCSI commands).


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    6



### Data Transfer Operation

- Send / Receive
  - Receiving endpoint posts receive buffer descriptor to its receive Work Queue.
  - Sending endpoint posts send buffer descriptor (send request) to its send Work Queue.
  - VI Providers transfer data.
  - Both endpoints notified on completion (both descriptors marked Done).


Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 7



### Data Transfer Operation

- RDMA Write setup:
  - Local endpoint is source, remote is destination.
  - Remote endpoint registers a memory region (data transfer destination), then sends the region's memory handle and virtual address to local endpoint.
  - Local endpoint registers a memory region (data transfer source).


Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 8



### Data Transfer Operation

- RDMA Write operation:
  - Local endpoint posts RDMA Write descriptor (RDMA Write request) to its send Work Queue.
    - Includes memory handles and virtual addresses for local buffer (data transfer source) and remote buffer (data transfer destination) plus transfer length.
  - VI Providers transfer data.
  - Only local endpoint notified on completion (descriptor marked Done).


Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 9



### Data Transfer Operation

- RDMA Write with Immediate Data
  - Similar to RDMA Write without Immediate Data.
  - Transfers 32-bit Immediate Data in addition to buffer-to-buffer data transfer.
  - Consumes receive Work Queue descriptor at remote (destination) endpoint.
  - Both endpoints notified on completion (both descriptors marked Done).


Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 10



### Data Transfer Operation

- RDMA Read setup:
  - RDMA Read is optional.
  - Local endpoint is destination, remote is source.
  - Remote endpoint registers a memory region (data transfer source), then sends the region's memory handle and virtual address to local endpoint.
  - Local endpoint registers a memory region (data transfer destination).


Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 11



### Data Transfer Operation

- RDMA Read operation:
  - Local endpoint posts RDMA Read descriptor (RDMA Read request) to its send Work Queue.
    - Includes memory handles and virtual addresses for local buffer (data transfer destination) and remote buffer (data transfer source) plus transfer length.
  - VI Providers transfer data.
  - Only local endpoint notified on completion (descriptor marked Done).
  - RDMA Read is optional.


Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 12



### Reliability Levels

- **Unreliable Delivery**
  - Sends and RDMA Write data delivered at most once, but may be lost or delivered out of order.
  - RDMA Reads prohibited.
  - Errors (e.g. lost messages) do not break connections.


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    13



### Reliability Levels

- **Reliable Delivery**
  - Sends and RDMA Write data delivered exactly once and in order (or connection is broken).
  - Send Work Queue descriptor marked Done while data is still in flight.
  - Send / Receive: send Work Queue descriptor marked Done before receive Work Queue descriptor is marked Done.
  - RDMA Reads optional; may complete out of order relative to Sends and RDMA Writes.
  - Any error breaks the connection.

Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    14



### Reliability Levels

- **Reliable Reception**
  - Sends and RDMA Write data delivered exactly once and in order (or connection is broken).
  - Send Work Queue descriptor marked Done after data delivery has been confirmed.
  - Send / Receive: receive Work Queue descriptor marked Done before send Work Queue descriptor is marked Done.
  - RDMA Reads optional. The specification is not clear on whether there are any ordering guarantees beyond those for Reliable Delivery.
  - Any error breaks the connection.

Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    15



### References

- **Intel VI web site:**
  - [www.intel.com/design/servers/vi/index.htm](http://www.intel.com/design/servers/vi/index.htm)
- **VI Architecture web site:**
  - [www.viarch.org](http://www.viarch.org)
  - Appears to be broken.
- **VI Developer Forum web site:**
  - [www.vidf.org](http://www.vidf.org)
  - Under development.
- **Orca web site:**
  - [www.orca.com/via/docs/](http://www.orca.com/via/docs/)


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    16



### Agenda

- VI Architecture Summary
- **Command Mapping**
- Flow Control
- Miscellaneous Issues
- Future Schedule

Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    17



### Command Mapping

- Basic Concepts
- Reliability Level Use
- SCSI Read command choices
- SCSI Write command choices
- Direct Device Access

Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    18

Ophidian Designs

### Basic Concepts

- VI Connections between SVP (SCSI) Initiators and SVP (SCSI) Targets.
- SCSI Commands, Status, Task Management use Send / Receive operations.
  - Based on FCP formats
- Each VI Connection is independent:
  - Each VI Connection to a SVP (SCSI) Target is a separate SVP (SCSI) Initiator.
  - Commands and Task Management share the same VI Connection.

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 19

Ophidian Designs

### Reliability Level Use

- Assume Reliable Delivery is predominant.
  - Want to use RDMA Read and in-order delivery.
- If special circumstances dictate, may use Reliable Reception.
  - E.g., where FCP uses FCP\_CONF.
- Neither prohibit use of Unreliable Delivery nor try to make it reliable.
  - Might be usable for video data or other so-called isochronous data.
  - Only recovery for many errors is to disconnect and start over (equivalent to Abort Task Set).

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 20

Ophidian Designs

### SCSI Read Command Choice 1

- Initiator sends Read command using Send.
  - Includes Read data buffer's memory handle and virtual address.
- Target returns Read data using RDMA Write.
- Target returns status using separate Send.
  - Target may post Send before RDMA Write completes.
  - Reliable Delivery and Reliable Reception guarantee in order delivery.

```

sequenceDiagram
    participant I as Initiator
    participant T as Target
    I->>T: Send: command
    T-->>I: RDMA Write: data
    T-->>I: Send: status
    
```

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 21

Ophidian Designs

### SCSI Read Command Choice 2

- Initiator sends Read command using Send.
  - Includes Read data buffer's memory handle and virtual address.
- If success and no autosense data:
  - Target returns Read data using RDMA Write.
  - Immediate data is Task Tag.
- If error or autosense data:
  - Target follows choice 1.

```

sequenceDiagram
    participant I as Initiator
    participant T as Target
    I->>T: Send: command
    T-->>I: RDMA Write: data
    T-->>I: Immed: Task Tag
    
```

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 22

Ophidian Designs

### SCSI Read Command Choices

- No difference in round-trip communication with Reliable Delivery.
  - Choice 1 might require an extra round-trip with Reliable Reception (specification ambiguous).
- No difference in receive Work Queue descriptors used.
- Choice 1 required regardless (for errors).
- Choice 2 might be advantageous with Unreliable Delivery connections.
- Recommendation: only allow choice 1.

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 23

Ophidian Designs

### SCSI Write Command Choice 1

- Initiator sends Write command using Send.
  - Includes Write data buffer's memory handle and virtual address.
- Target fetches Write data using RDMA Read.
- Target returns status using Send.
- Problem: RDMA Read is optional, might not be available with all VI Providers.

```

sequenceDiagram
    participant I as Initiator
    participant T as Target
    I->>T: Send: command
    T-->>I: RDMA Read: data
    T-->>I: Send: status
    
```

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 24

### SCSI Write Command Choice 2

- Emulate RDMA Read.
- Initiator sends Write command using Send.
- Target requests Write data using Send.
- Initiator returns Write data using RDMA Write.
- Target returns status using Send.
- Requires Initiator software intervention to fetch Write data.

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 25

### SCSI Write Command Choices

- If RDMA Read is available, Choice 1 is always better than Choice 2.
  - Write data supplied by NIC hardware rather than Initiator software.
- If RDMA Read is not available, Choice 2 is only option.
- Can we mandate RDMA Read?
- Recommendation: assume we can mandate RDMA Read, then see if anyone objects.

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 26

### Direct Device Access

- In principle, RDMA Read / Write requests could be used to read / write a device (LU).
- Would require setup or map device request.
- Serious flow control problems and distortion of SCSI device models.
- RDMA operations do memory access, not device access.
- Recommendation: **DON'T DO THIS**

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 27

### Agenda

- VI Architecture Summary
- Command Mapping
- **Flow Control**
- Miscellaneous Issues
- Future Schedule

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 28

### Flow Control

- Virtual Interface Architecture Specification, Version 1.0, page 15:
  - VI Consumers are responsible for managing flow control on a connection. The VI Consumer on the receiving side must post a Receive Descriptor of sufficient size before the sender's data arrives. If the Receive Descriptor at the head of the queue is not large enough to handle the incoming message, or the Receive Queue is empty, an error will occur. The connection may be broken if it is intended to be reliable. See section 2.5.

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 29

### Flow Control

- SCSI Initiators may issue an arbitrarily large number of commands, expecting to receive Task Set Full status if the Initiator issues more commands than the SCSI Target or Logical Unit can accept.
- This is common or normal behavior in some system environments.
- This behavior is guaranteed to overrun the SVP (SCSI) Target's receive Work Queue and break the VI Connection between the SVP (SCSI) Initiator and SVP (SCSI) Target.

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 30

Ophidian Designs

### Flow Control Requirements

- Requirements:
  - An SVP Target must post some minimum number of receive buffers on each connection (i.e., for each initiator).
  - Information about this minimum must be available to the SVP Initiator.
  - The SVP Initiator must not send more commands or task management requests than this minimum.
- The converse is also required: an SVP Target must not overrun an SVP Initiator.

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 31

Ophidian Designs

### SAM Communication Model

- SAM-2 draft revision 12, page 18:

Figure 8 - SAM client-server model

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 32

Ophidian Designs

### SVP Communication Model

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 33

Ophidian Designs

### SVP Communication Model

- SVP Flow Control operates between SVP Initiators and SVP Targets.
  - Analogous to R\_RDY flow control in FCP.
  - Independent of Task Set Full mechanism.
- A command is transferred from SVP Initiator to SVP Target using SVP Flow Control.
- The SVP Target then transfers the command to a Logical Unit or Device Server, where the command may incur Task Set Full status.

Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 34

Ophidian Designs

### SVP I-T Flow Control

- SVP Initiators maintain TRD\_Count:
  - Target Receive Descriptor Count.
  - Minimum number of descriptors the Initiator knows are present in the Target's receive Work Queue.
  - Initiator decrements TRD\_Count whenever it sends (posts a send descriptor containing) a command or task management request to the target.
  - Initiator may not send commands or task management requests when TRD\_Count is zero.


Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 35

Ophidian Designs

### SVP I-T Flow Control

- TRD\_Count:
  - Initial value determined during VI Connection establishment. (E.g., passed in first message on a new connection).
  - Decremented whenever Initiator sends a command or task management request to Target.
  - Each message from target contains a value that the Initiator uses to adjust TRD\_Count (e.g. adds to TRD\_Count):
    - Command status.
    - Task management responses.
    - Special purpose "Adjust TRD\_Count" message.


Copyright ©1999 by Ophidian Designs T10/99-316r0 November 3, 1999 36



### SVP I-T Flow Control

- If TRD\_Count reaches zero, Target should promptly provide a receive descriptor.  
Example:
  - If an Initiator sends a command that consumes the Target's last receive Work Queue descriptor (TRD\_Count = 0), and the Target is unable to allocate additional descriptors, the Target should (shall?) immediately return Task Set Full status and specify TRD\_Count\_Increment = 1.
    - Final specification of Task Set Full vs. Busy status deferred to Ralph Weber.


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    37



### SVP T-I Flow Control

- TRD\_Count ensures the Initiator does not overrun the Target's receive Work Queue.
- We also need a mechanism to ensure the Target does not overrun the Initiator's receive Work Queue.
- We could use a similar mechanism (e.g. IRD\_Count), with some special rules to avoid deadlock (TRD\_Count = IRD\_Count = 0).


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    38



### SVP T-I Flow Control

- Alternative to IRD\_Count:  
An SVP Initiator shall post to its receive Work Queue at least as many descriptors as the sum of:
  - One receive descriptor for each of its outstanding commands.
  - One receive descriptor for any outstanding task management request.
  - One additional receive descriptor to allow the target to send an "Adjust TRD\_Count" message.
- Recommendation: use this alternative.


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    39



### Agenda

- VI Architecture Summary
- Command Mapping
- Flow Control
- **Miscellaneous Issues**
- **Future Schedule**


Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    40



### Miscellaneous Issues

- Task Tag size:
  - Recommend 32 bits.
- Untagged Task support:
  - Recommend not supported.
  - Alternative: reserve Task Tag value zero to designate an untagged task.
- Linked commands:
  - Recommend not supported.

Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    41



### Deferred Issues

- Recommend deferring the following issues, hoping that VIDF or someone else will address them:
  - Discovery: how does an SVP Initiator determine whether a potential VI Connection supports SVP or some other application protocol.
  - Security: how does an SVP Target determine that a VI Connection is from an authorized SVP Initiator rather than from an arbitrary application program.

Copyright ©1999 by Ophidian Designs    T10/99-316r0    November 3, 1999    42



Ophidian  
Designs

### Future Schedule

- No plans to discuss SVP during the January, 2000 T10 plenary week (Brisbane, Australia).
- Anticipate first draft of SVP available in late January, 2000.
- Request review via email and during the subsequent T10 plenary week (March 6-10, 2000, Dallas, TX).
- Goal is completion by end of year 2000.

Copyright ©1999 by Ophidian Designs      T10/99-316r0      November 3, 1999 43