SNIA

STORAGE NETWORKING INDUSTRY ASSOCIATION

EDUCATION

SCSI – The Protocol for all Storage Architectures

David Deming, Solution Technology

April 12, 2005

Abstract

SCSI: The Protocol for All Storage Architectures

This session will appeal to System Administrators, Storage Administrators, Storage Architects, and those that are seeking a fundamental understanding of SCSI Protocol and how it benefits your IT storage applications. The session will delve into the SCSI model, it’s protocol, and how storage applications benefit from having a single high level protocol. The audience will receive the fundamental understanding of why SCSI is used as the storage industry’s main language.

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA.
- Member companies and individuals may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced without modification
  - The SNIA must be acknowledged as source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.

Objectives

- Brief understanding of SCSI and its history
- Understanding of industry standards and the role of the SCSI Architecture
- Understanding the SCSI Architecture model
- Understanding the I/O Operation and the importance behind SCSI Protocol
- Learn how parallel SCSI, Fibre Channel, and iSCSI rely on the SCSI Architecture to support storage applications

Brief History

- SCSI-1: In the beginning – 1987
  - **SCSI was adapted from the SASI Interface (1979)**
  - **Specified a physical interface, transport protocol, and standard command for disk, tape, and other storage devices**
- SCSI-2: Second coming of SCSI
  - **Speed and bus width enhancements**
  - **Multi-threading commands and transport protocol cleanup**
  - **Added more storage device type command sets**
- SCSI-3: Today and beyond
  - **More speed: Ultra-2/Ultra-160/Ultra-320**
  - **Separation of Physical Interface, Transport Protocols, and the SCSI Command Sets**

SCSI Today and the Future

- SCSI Language
  - **Every architecture uses the SCSI command sets**
  - **Every architecture uses the SCSI upper layer protocol model**
  - **SCSI is the Language of Love**
- SCSI Transport
  - **Defines protocol mapping and transfer of SCSI language**
  - **Fibre Channel Protocol (FCP), Serial Storage Architecture (SSA), Serial Bus Protocol (SBP), iSCSI, Serial ATA (SATA), Serial Attached SCSI (SAS)**
- SCSI Physical Architectures
  - **Serial Attached SCSI – next generation SCSI physical interface**
  - **Fibre Channel, SSA, 1394, ATAPI, TCP/IP**

Introduction to Standards

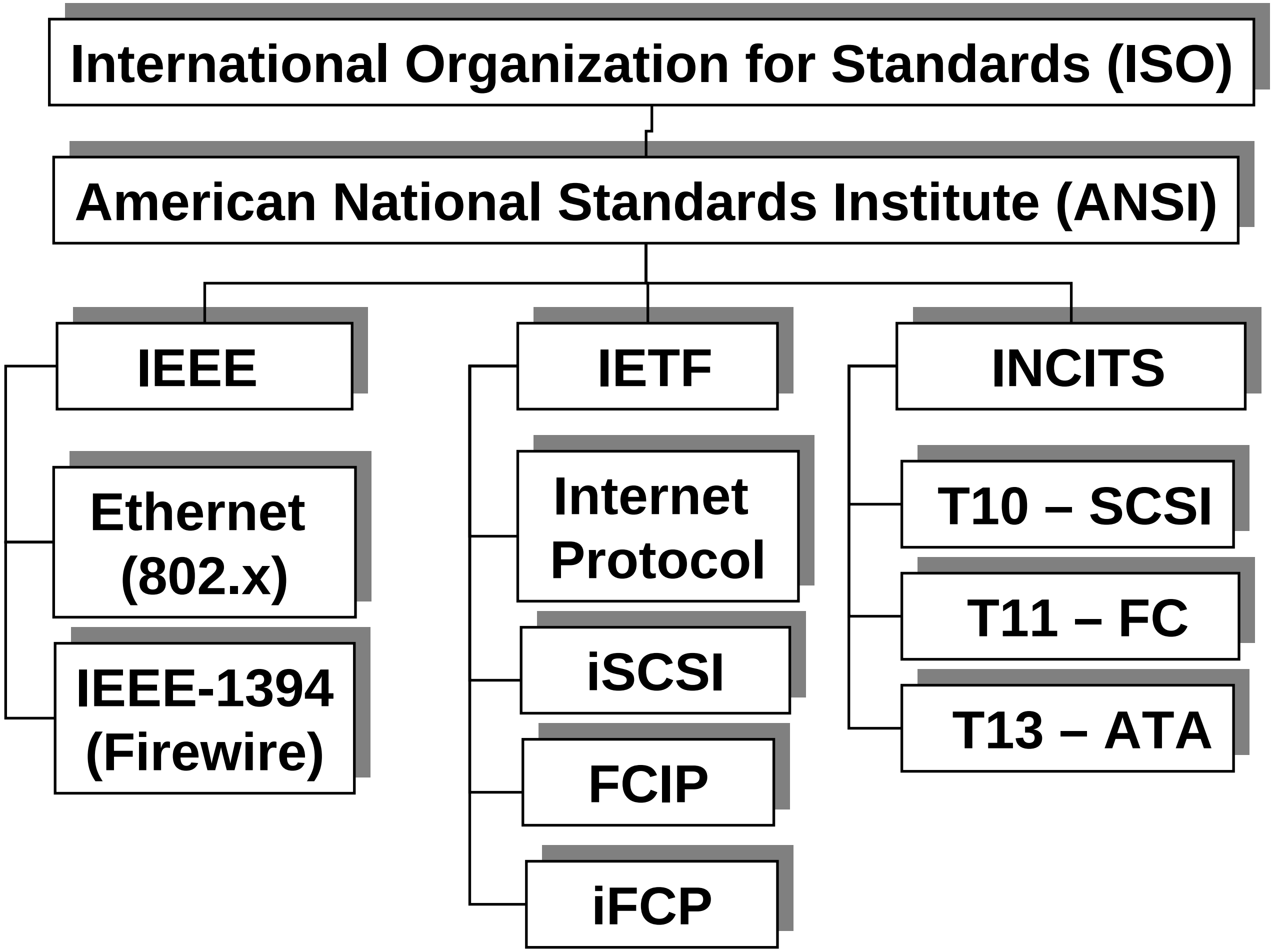
SNIA

STORAGE NETWORKING INDUSTRY ASSOCIATION

EDUCATION

SCSI Standards and their impact on all storage interfaces

Standards Organizations


International Organization for Standards (ISO)

American National Standards Institute (ANSI)

IEEE

Ethernet (802.x)

IEEE-1394 (Firewire)

IETF

Internet Protocol

iSCSI

FCIP

iFCP

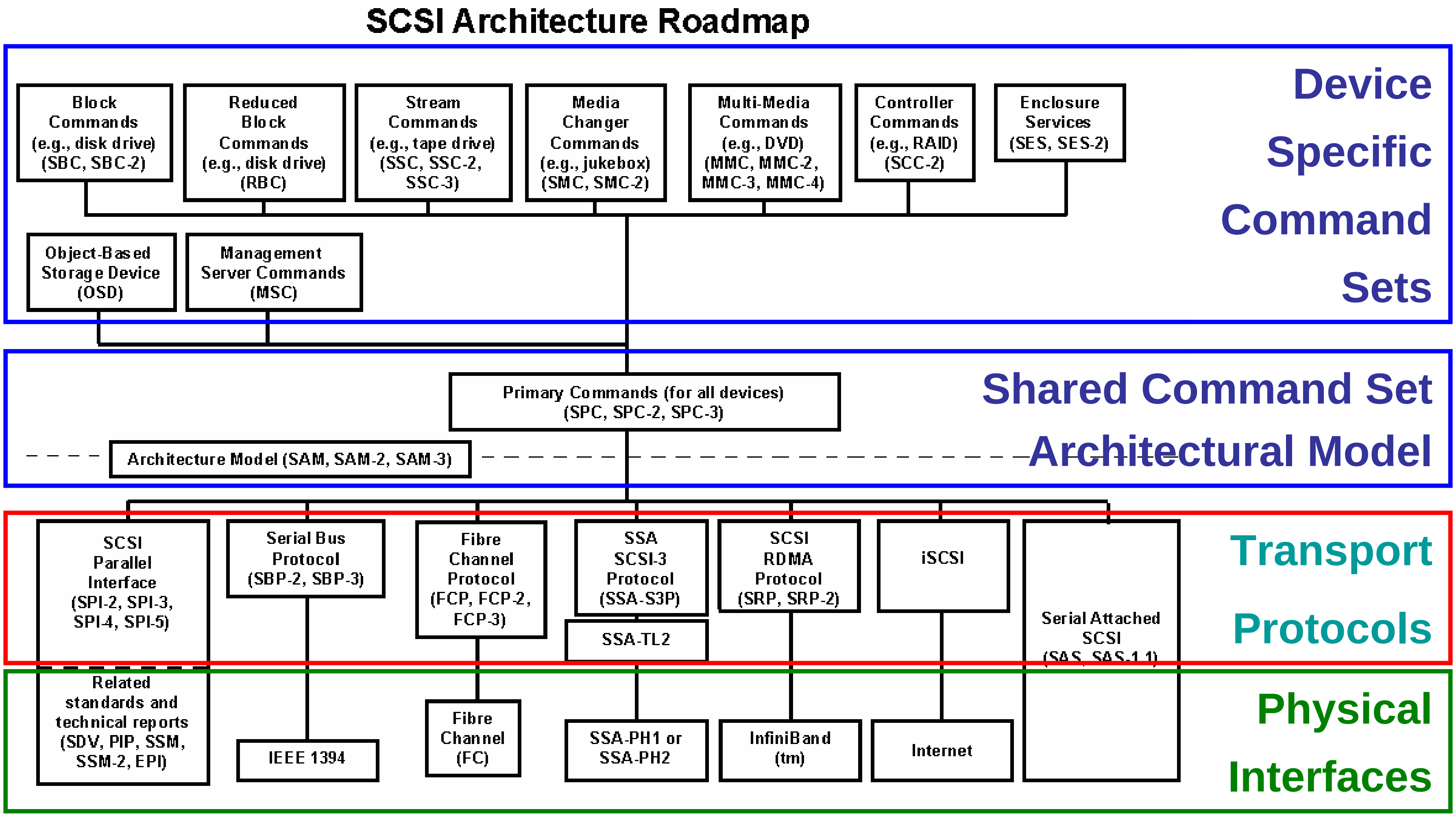
INCITS

T10 – SCSI

T11 – FC

T13 – ATA

SCSI Standards



SCSI Architecture Roadmap

Device Specific Command Sets

Block Commands (e.g., disk drive) (SBC, SBC-2)

Reduced Block Commands (e.g., disk drive) (RBC)

Stream Commands (e.g., tape drive) (SSC, SSC-2, SSC-3)

Media Changer Commands (e.g., jukebox) (SMC, SMC-2)

Multi-Media Commands (e.g., DVD) (MMC, MMC-2, MMC-3, MMC-4)

Controller Commands (e.g., RAID) (SCC-2)

Enclosure Services (SES, SES-2)

Object-Based Storage Device (OSD)

Management Server Commands (MSC)

Shared Command Set

Primary Commands (for all devices) (SPC, SPC-2, SPC-3)

Architectural Model

Architecture Model (SAM, SAM-2, SAM-3)

Transport Protocols

SCSI Parallel Interface (SPI-2, SPI-3, SPI-4, SPI-5)

Serial Bus Protocol (SBP-2, SBP-3)

Fibre Channel Protocol (FCP, FCP-2, FCP-3)

SSA SCSI-3 Protocol (SSA-S3P)

SSA-TL2

SCSI RDMA Protocol (SRP, SRP-2)

iSCSI

Serial Attached SCSI (SAS, SAS-1.1)

Physical Interfaces

Related standards and technical reports (SDV, PIP, SSM, SSM-2, EPI)

IEEE 1394

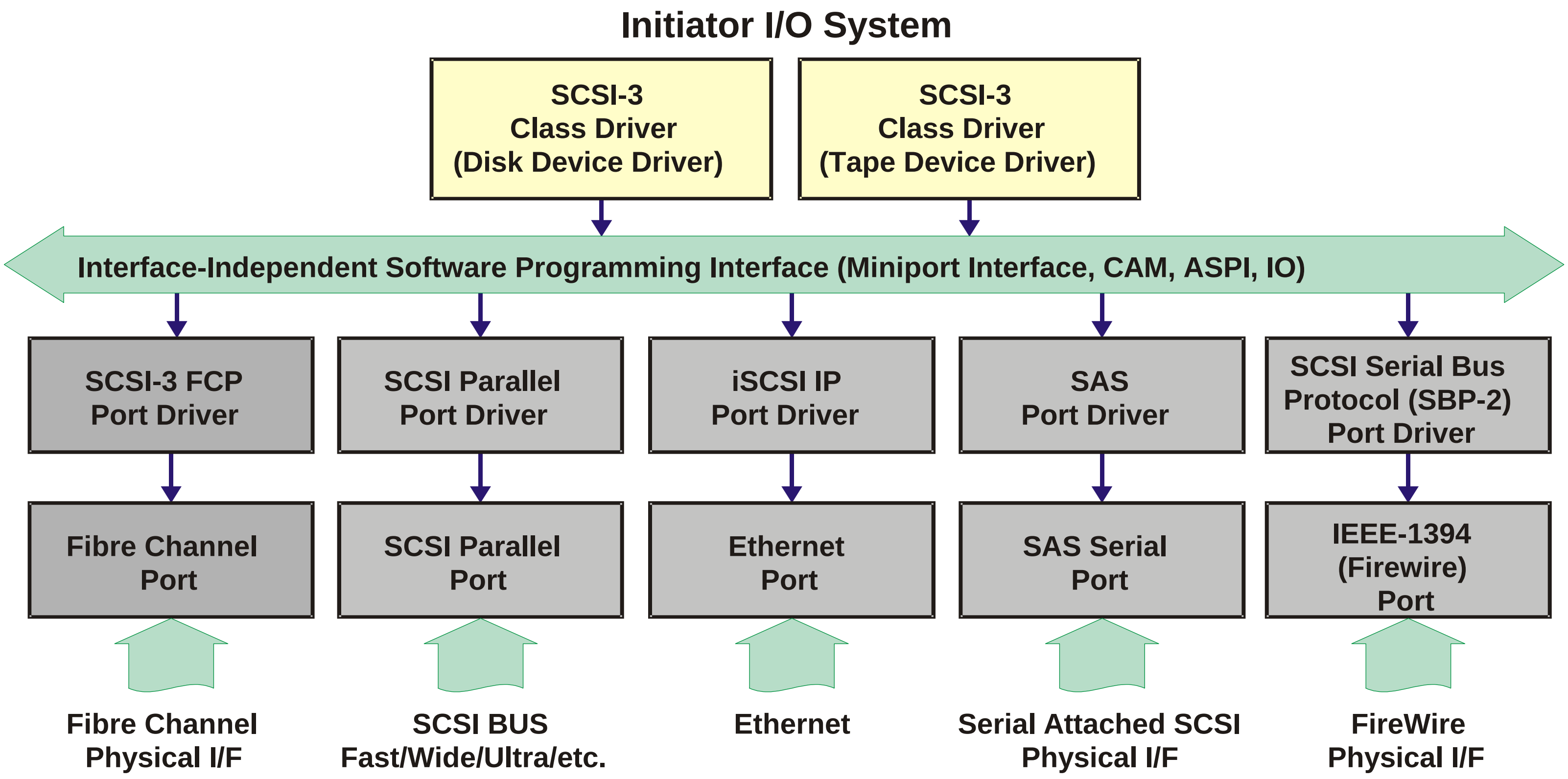
Fibre Channel (FC)

SSA-PH1 or SSA-PH2

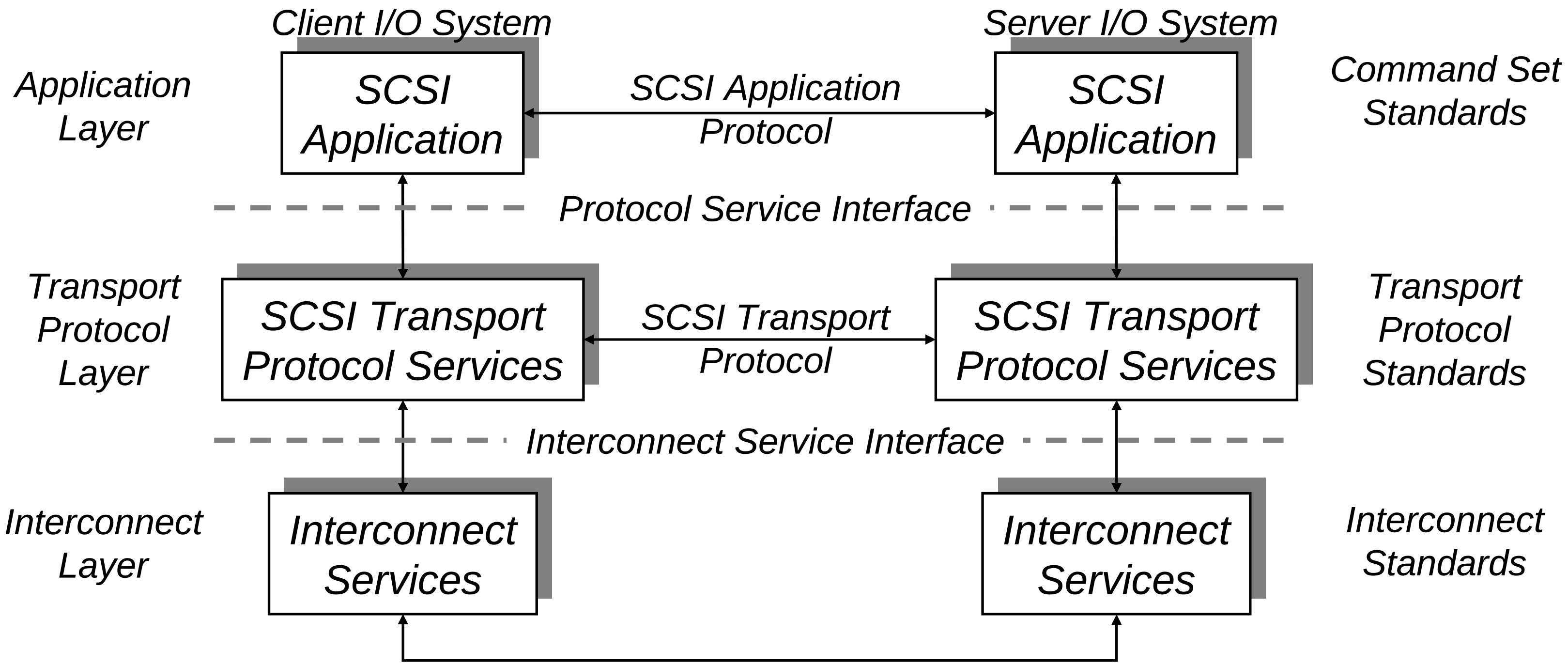
InfiniBand (tm)

Internet

SCSI-3 Physical Interfaces


Initiator I/O System
SCSI-3 Class Driver (Disk Device Driver)
SCSI-3 Class Driver (Tape Device Driver)
Interface-Independent Software Programming Interface (Miniport Interface, CAM, ASPI, IO)
SCSI-3 FCP Port Driver
SCSI Parallel Port Driver
iSCSI IP Port Driver
SAS Port Driver
SCSI Serial Bus Protocol (SBP-2) Port Driver
Fibre Channel Port
SCSI Parallel Port
Ethernet Port
SAS Serial Port
IEEE-1394 (Firewire) Port
Fibre Channel Physical I/F
SCSI BUS Fast/Wide/Ultra/etc.
Ethernet
Serial Attached SCSI Physical I/F
FireWire Physical I/F

Protocol Service Model



- Application layer: Clients and servers that originate and process SCSI I/O operations by means of a SCSI application protocol
- Transport protocol layer: Services and protocols through which clients and servers communicate
- Interconnect layer: Services, signaling mechanism and interconnect subsystem needed for the physical transfer of data from sender to receiver.

SCSI Command Sets

- The most appealing features behind the SCSI Architecture, i.e. the SCSI language
- Includes every type of storage device manufactured today; disk, tape, CD, storage enclosures, etc…
- Commands are manufacturer independent and are common for all types of devices
- SCSI device drivers and operating systems have utilized SCSI Command Sets since the late 80’s
- Can be used in any storage architecture including Fibre Channel, iSCSI, InfiniBand, ATA/IDE, SATA (Serial ATA), SAS (Serial Attached SCSI), 1394 (Firewire)

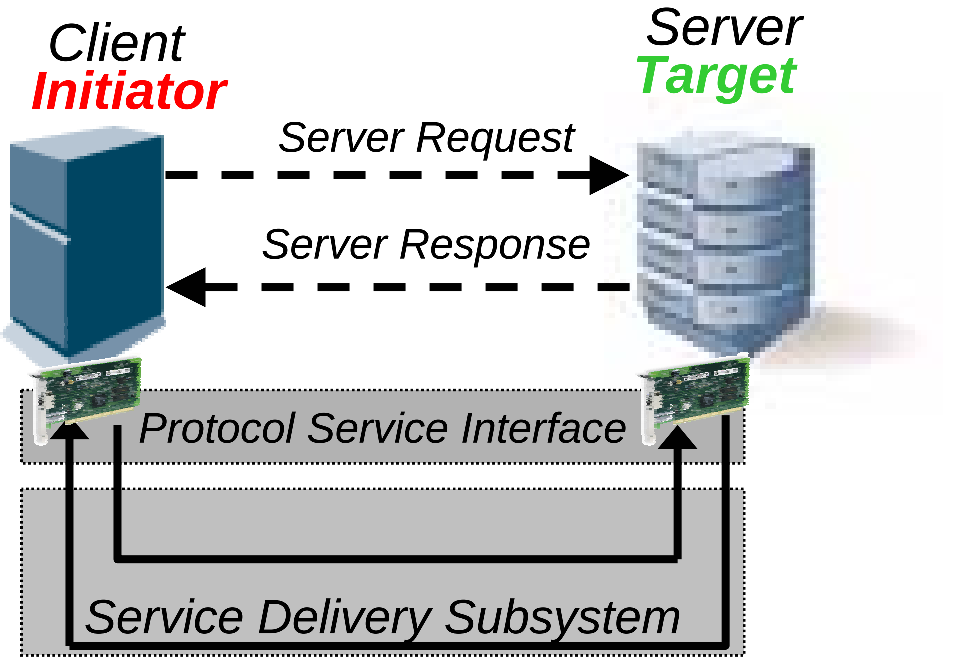
SCSI Architectural Model

Characteristics of the SCSI-3 Architecture

The SCSI Distributed Model

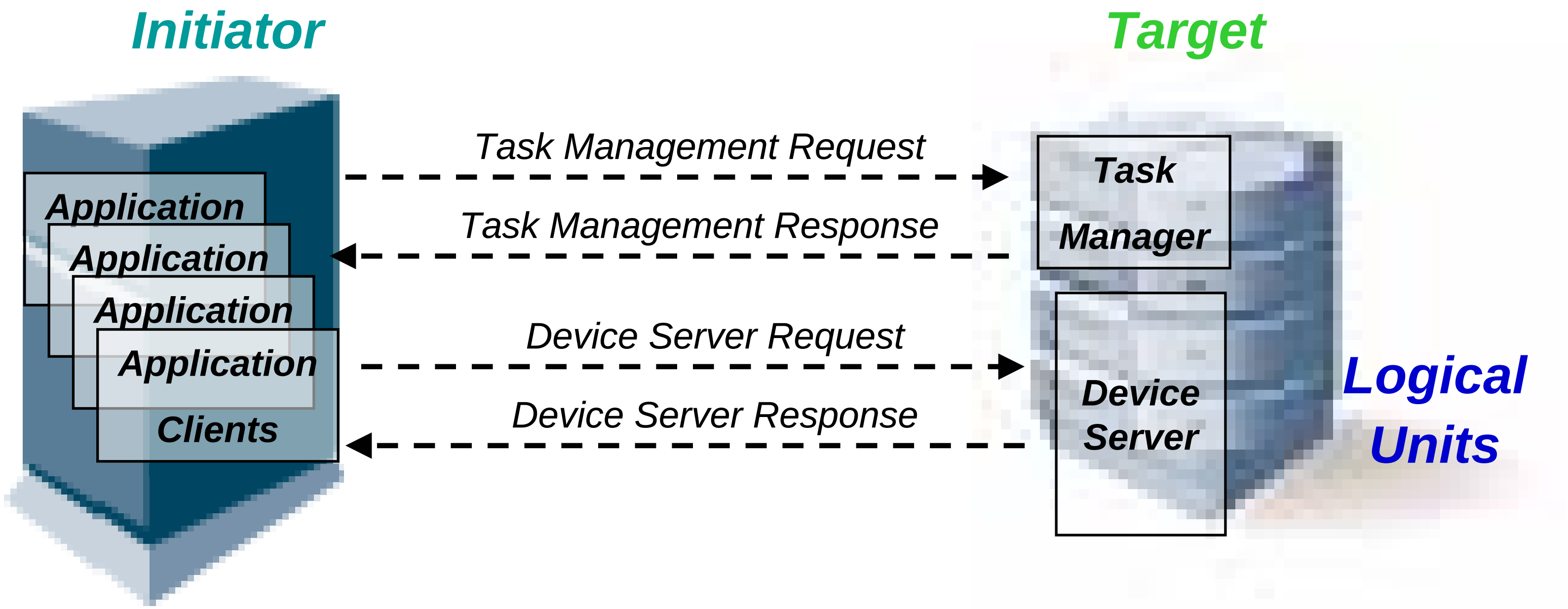
- SCSI is a client-server protocol.
- The *client* is called the ***Initiator*** (usually the OS I/O subsystem) and issues requests to the server.
- The *server* is called the ***Target*** (usually the SCSI controller that is part of a storage device) and receives, executes and returns Initiator requests and their associated responses.

Client-Server Transaction



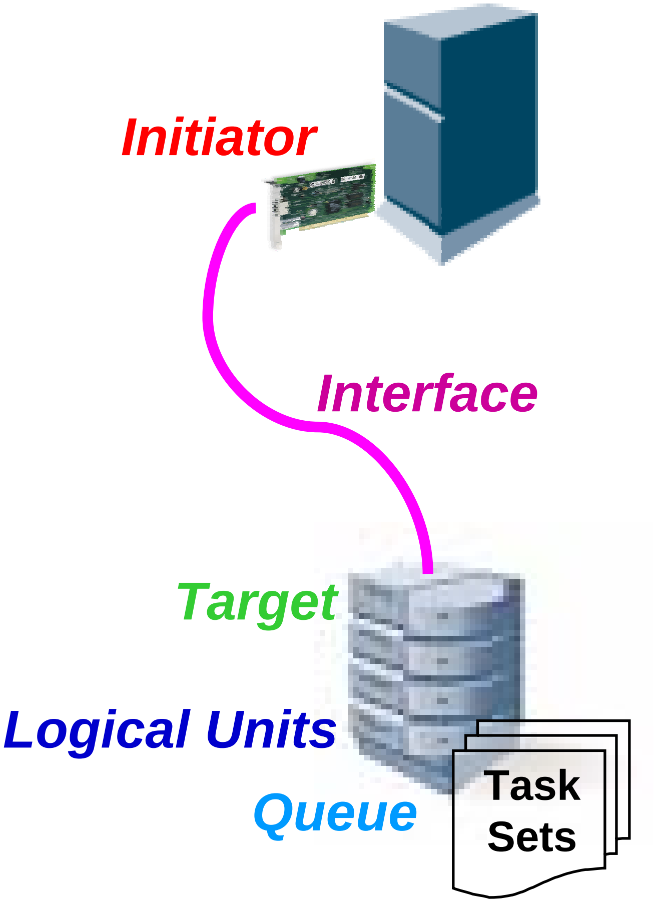
Client-Server Model

- A single Initiator can have multiple Application Clients.
- Targets have ONE Task Manager and one or more Logical Units (LU), which are numbered (LUN). The Task Manager:
  - **controls the sequencing of one or more tasks within a LU**
  - **carries out the task management functions**
  - **has the authority to modify Service Requests that have already been received by the target**
- The Device Server processes operations and directs them to a specific LUN.



Re-cap SCSI Terminology

- SCSI is a *standard* that defines an interface between an ***Initiator*** (usually a computer) and a ***Target*** (usually a storage device such as a hard disk, tape backup, or storage array).
- ***Interface*** refers to connectors, cables, electrical signals, optical signals and the command protocol that allow initiators and targets to communicate.
- ***Logical Units*** are a subset of Target devices which can allow for scalability.
- The ***Queue*** (or Task Set) is used to hold pending Commands (Tasks) that the Target will execute



Task Attributes

- There are four types of Task attributes that can affect how the Target executes each task (Command)
  - **Simple**
    - **Target can execute in any order**
    - **Target will typically apply a performance algorithm to numerous simple tasks**
  - **Ordered**
    - **Target must execute all ordered tasks in the order they are received**
    - **Any task prior to ordered must be executed before ordered task**
  - **Head of queue**
    - **Informs Target to insert the task into the front of the queue**
  - **Auto Contingent Allegiance (ACA)**
    - **Used when the Target enters into an error condition for a command that has previously executed**

I/O Operations

SNIA

STORAGE NETWORKING INDUSTRY ASSOCIATION

EDUCATION

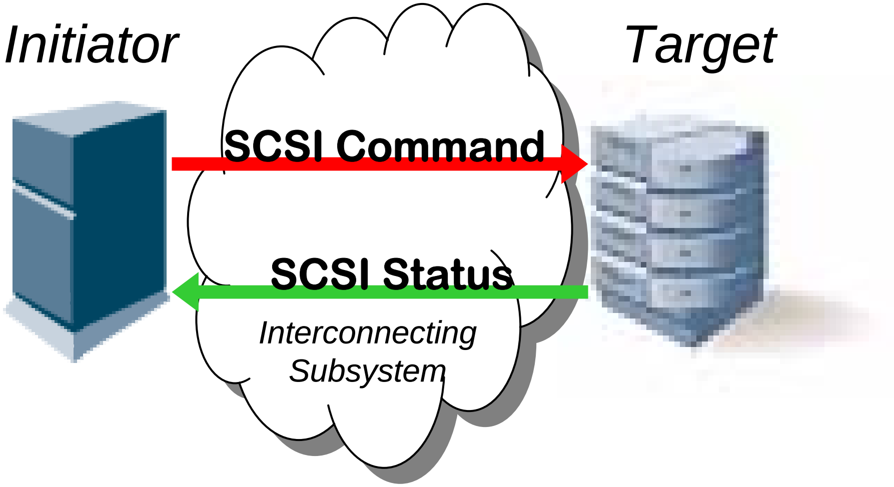
Characteristics of SCSI I/O Operations

I/O Operation Model

- There are two categories of Protocol Services:
  - **Execute Command and Confirmation Services**
  - **Data Transfer Services**
- There are three main phases of an I/O operation that includes a data transfer:
  1. **Command: Send required command and parameters via a Command Descriptor Block (CDB)**
  2. **Data: Transfer data in accordance with the command**
  3. **Status: Receive confirmation of command execution**

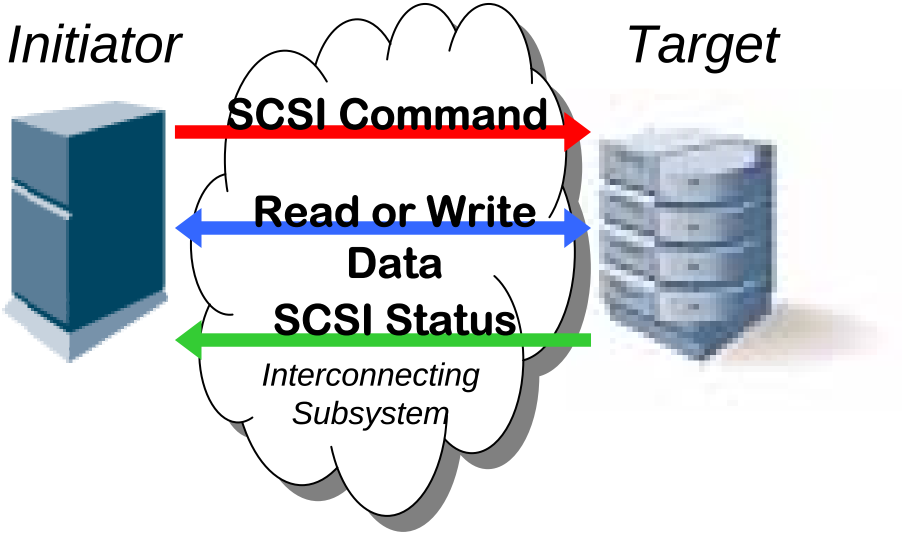
SCSI I/O Operations

- At a minimum, SCSI I/O Operations consist of:
  - **An Initiator issuing a SCSI Command**
  - **A Target returning completion Status**
  - **There is no “Data” transfer between Initiator and Target**
- Types of Commands that do not move Data
  - **Test Unit Ready**
  - **Start/Stop Unit**
  - **Rewind**



Operations that move Data

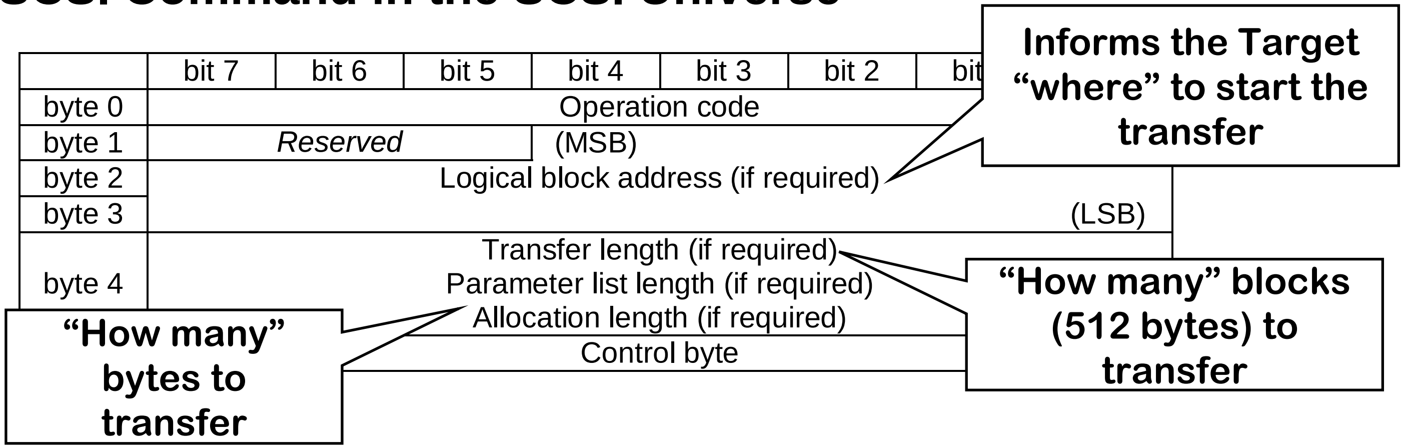
- When an Initiator and Target need to exchange information
  - **They utilize a Data phase**
  - **Data In transmits information from the Target to the Initiator**
  - **Data Out transmits information from the Initiator to the Target**
- Data can be transmitted all at once or take numerous Data phases to complete information transfers
- Types of Commands
  - **Read or Write**
  - **Inquiry**



Command Descriptor Block

- A Command is executed by sending a Command Descriptor Block (CDB) to the Target
- For each CDB
  - **The first byte of the CDB is the Operation Code**
  - **The last byte of the CDB is the Control Byte**
  - **The format of the Operation Code and Control Byte is identical for every SCSI Command in the SCSI Universe**

Example Six Byte CDB



	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit
byte 0	Operation code						
byte 1	*Reserved*			(MSB)			
byte 2	Logical block address (if required)						
byte 3	(LSB)						
byte 4	Transfer length (if required) Parameter list length (if required) Allocation length (if required)						
	Control byte						

Informs the Target "where" to start the transfer

"How many" blocks (512 bytes) to transfer

"How many" bytes to transfer

Other CDB Formats

CBDs can be:

- **10 bytes**
- **12 bytes**
- **16 bytes**
- **or even variable bytes in length**

Ten Byte CDB

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
byte 0	Operation code							
byte 1	*Reserved*			*Service Action (if required)*				
byte 2	(MSB)							
byte 3	Logical block address (if required)							
byte 4								
byte 5								(LSB)
byte 6	Reserved							
byte 7	(MSB) Transfer length (if required) Parameter List Length (if required)							
byte 8	Allocation length (if required)							(LSB)
byte 9	Control byte							

Twelve Byte CDB

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
byte 0	Operation code							
byte 1	*Reserved*			*Service Action (if required)*				
byte 2	(MSB)							
byte 3	Logical block address (if required)							
byte 4								
byte 5								(LSB)
byte 6	(MSB)							
byte 7	Transfer length (if required)							
byte 8	Parameter list length (if required)							
byte 9	Allocation length (if required)							(LSB)
byte 10	Reserved							
byte 11	Control byte							

SCSI Commands all Devices

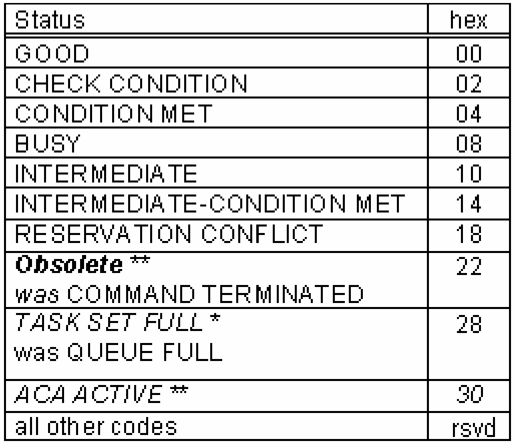
Standard SCSI Commands:

- **Disk**
- **Tape**
- **Storage Enclosures**
- **Disk Arrays**
- **CD**
- **WORM**
- **Media Changers**
- **Every device imaginable or not yet created**

Op. Code	Command Name	Type
00h	TEST UNIT READY	M
03h	REQUEST SENSE	Z
12h	INQUIRY	M
15h	MODE SELECT(6)	Z
18h	COPY	O
1Ah	MODE SENSE(6)	Z
1Ch	RECEIVE DIAGNOSTIC RESULTS	O
1Dh	SEND DIAGNOSTIC	O
39h	COMPARE	O
3Ah	COPY AND VERIFY	O
3Bh	WRITE BUFFER	Z
3Ch	READ BUFFER	O
4Ch	LOG SELECT	O
4Dh	LOG SENSE	O
55h	MODE SELECT(10)	Z
5Ah	MODE SENSE(10)	Z

SCSI Status

- Any time a SCSI Command is sent to a Target
  - **The Initiator expects a completion Status**
  - **This status can reflect successful or unsuccessful completion of the command**
- The Status may indicate
  - **Busy or Not Ready**
  - **Error condition exists for another command**
  - **Targets task set is full**



Status	hex
GOOD	00
CHECK CONDITION	02
CONDITION MET	04
BUSY	08
INTERMEDIATE	10
INTERMEDIATE-CONDITION MET	14
RESERVATION CONFLICT	18
Obsolete ** was COMMAND TERMINATED	22
*TASK SET FULL ** was QUEUE FULL	28
*ACA ACTIVE ***	30
all other codes	rsvd

EDUCATION

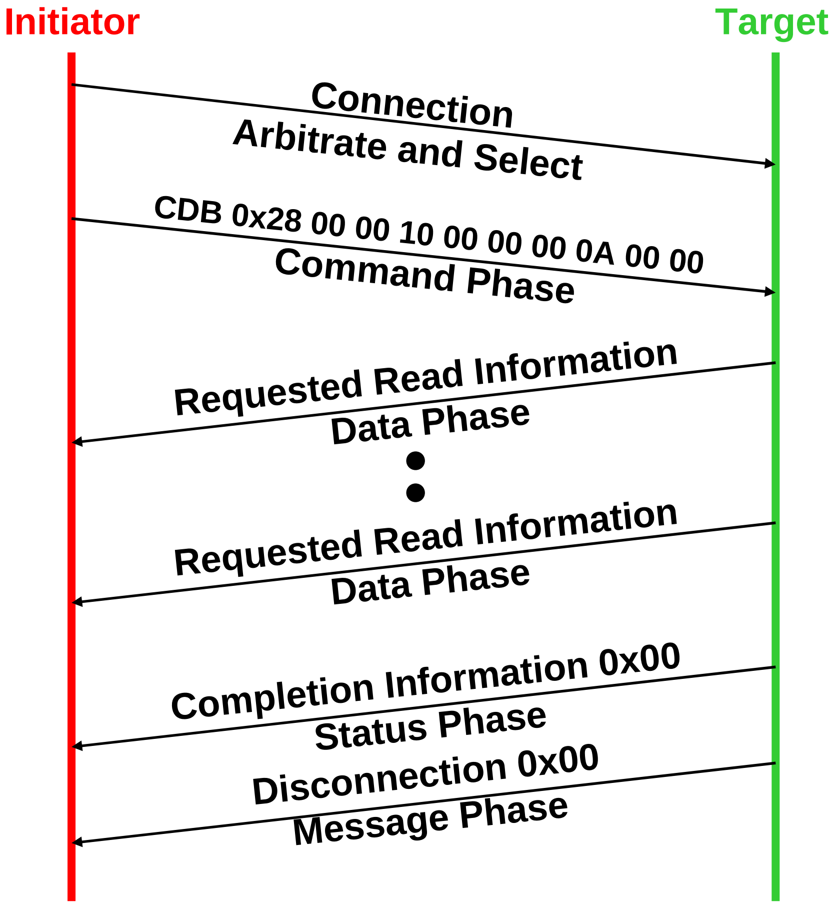
Parallel SCSI

Parallel SCSI Characteristics

- Information can only go in one direction at a time
  - **Each piece (byte) of information is acknowledged**
  - **Information transfers are interlocked**
- A connection protocol is used before any information transfers
  - **Arbitration, Selection, and Message phases**
  - **Uses SCSI ID’s to identify devices**
- Utilizes protocol “bus” phases to accomplish information transfers
  - **Command Phase to deliver the CDB**
  - **Data Phase to deliver customer data**
  - **Status Phase to deliver completion status**
- Can multi-task by disconnecting

Read Command Example

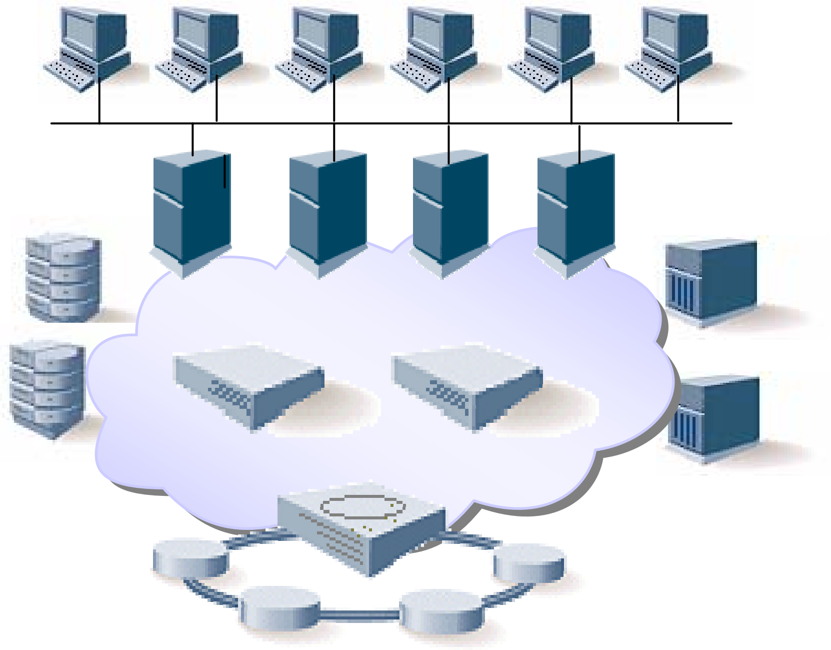
- Initiator connects to Target
  - **Once Initiator connects to Target, the Target is in control of the I/O Process**
- Initiator sends CDB information via Command Phases
- Target returns requested information via Data Phase
- Target returns completion information via Status Phase
- Target Disconnects via Message Phase



SNIA

STORAGE NETWORKING INDUSTRY ASSOCIATION

EDUCATION

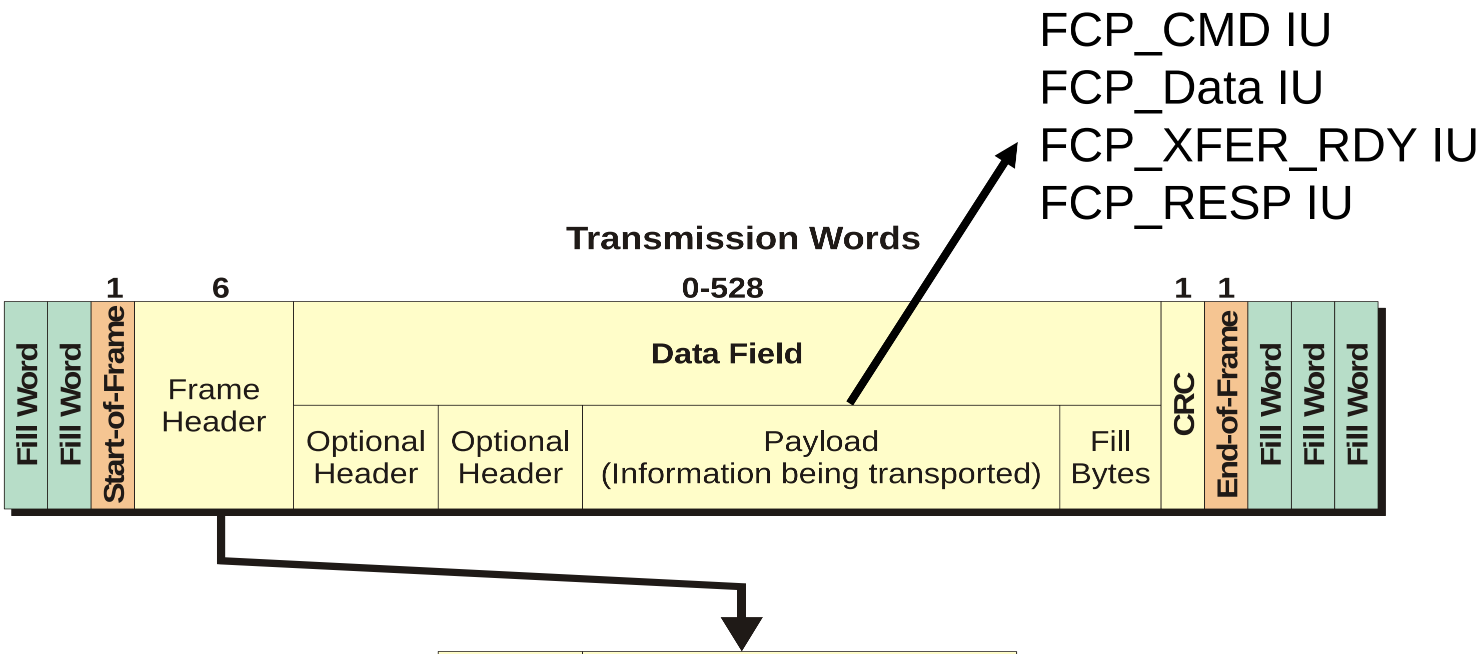


Fibre Channel

Fibre Channel Protocol Characteristics

- Fibre Channel creates frames to deliver SCSI Commands, Data, and Status information units (IU). FCP Frames include:
  - FCP_CMD frame
    - **Equivalent to parallel SCSI Command Phase**
  - FCP_XFER_RDY frame
    - **No actual parallel SCSI Equivalent**
  - FCP_DATA frame
    - **Equivalent to parallel SCSI Data Phase**
  - FCP_RSP frame
    - **Equivalent to parallel SCSI Status Phase**

Fibre Channel Frame



FCP_CMD IU
FCP_Data IU
FCP_XFER_RDY IU
FCP_RESP IU

Transmission Words

1 6 0-528 1 1

Fill Word | Fill Word | Start-of-Frame | Frame Header | Data Field: Optional Header | Optional Header | Payload (Information being transported) | Fill Bytes | CRC | End-of-Frame | Fill Word | Fill Word | Fill Word

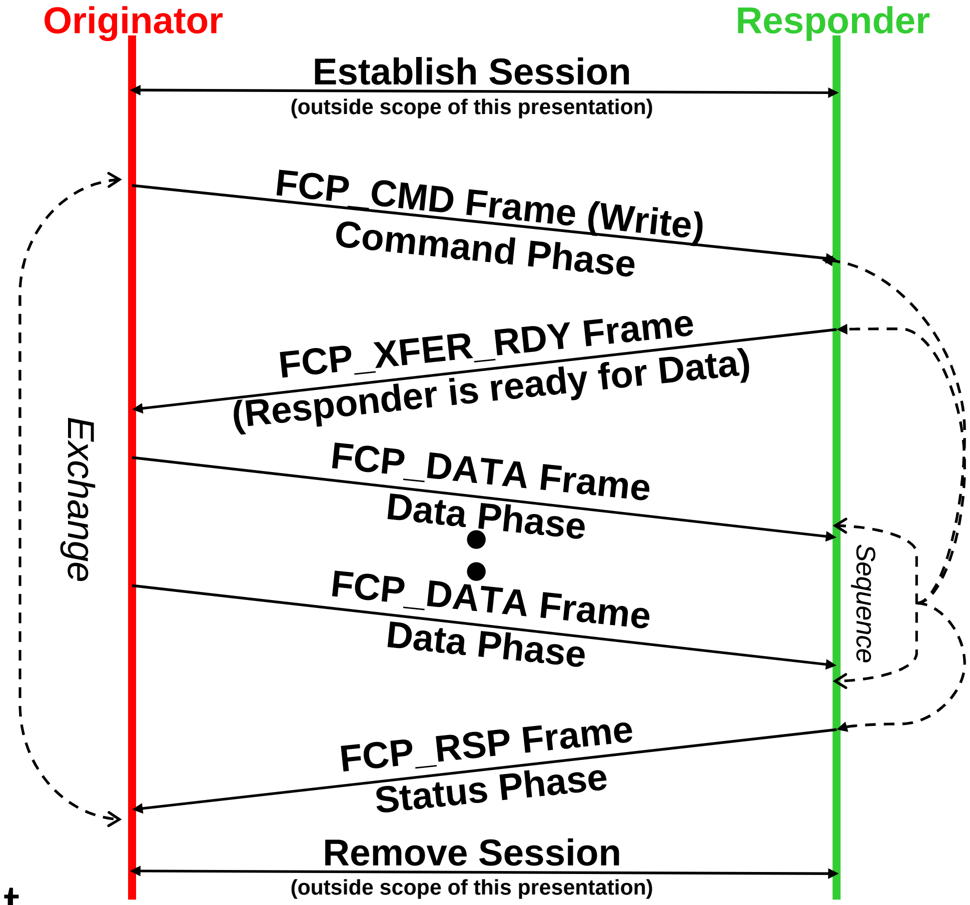
Frame type and content/function	R_CTL	Destination Address (D_ID)		*Where frame is being sent to*
Class-specific control information	CS_CTL	Source Address (S_ID)		*Where the frame came from*
Protocol Type in this frame	TYPE	Frame Control (F_CTL)		*Frame Control field*
Sequence this frame belongs to	SEQ_ID	DF_CTL	SEQ_CNT	*Sequential count of frames*
Originator Exchange ID	OX_ID		RX_ID	*Responder Exchange ID*
Multi-purpose parameter field	Parameter Field (PARM)			

Write Command Example

SNIA STORAGE NETWORKING INDUSTRY ASSOCIATION

EDUCATION

- Before devices can communicate in FC they must establish a session via login protocol
- Originator sends Write command to responder
- Responder replies with a ready to receive response
- Originator sends Data frames
- When all Data is transferred, Responder sends Status to Originator
- Session could end at this point but typically doesn't

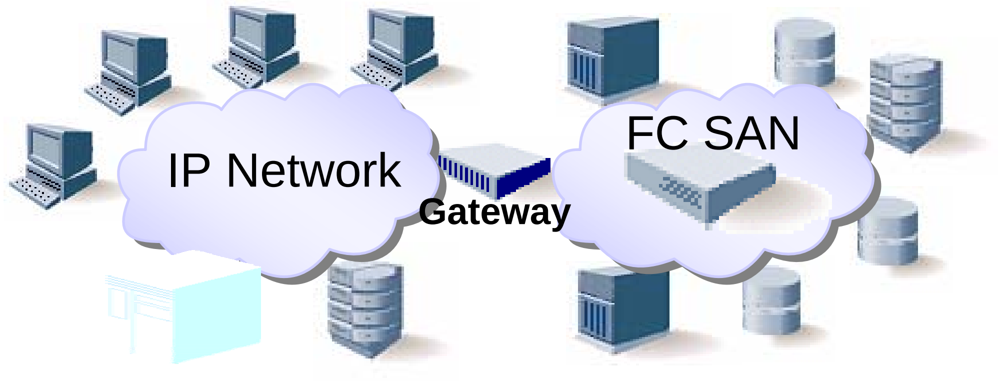


SNIA

STORAGE NETWORKING INDUSTRY ASSOCIATION

EDUCATION

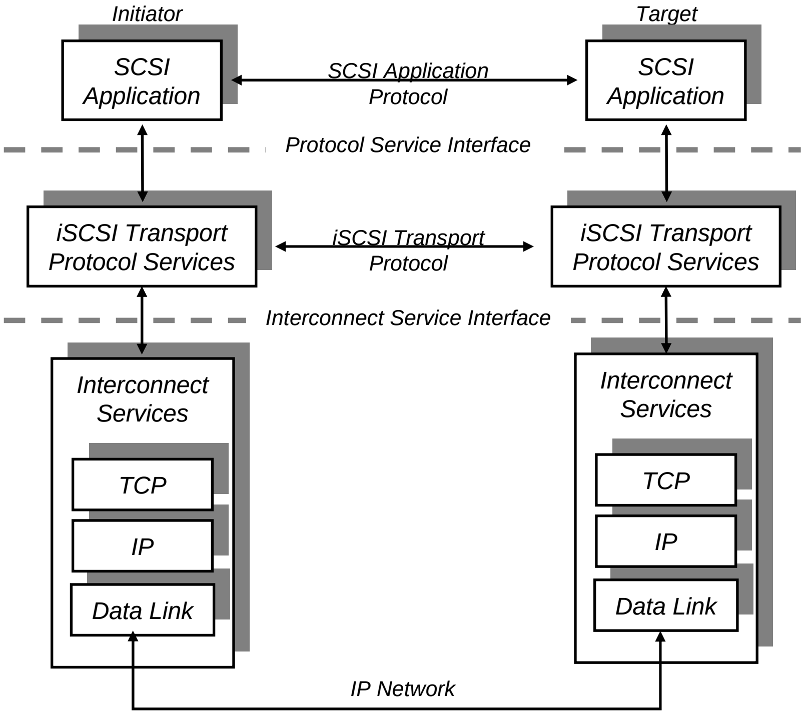
iSCSI

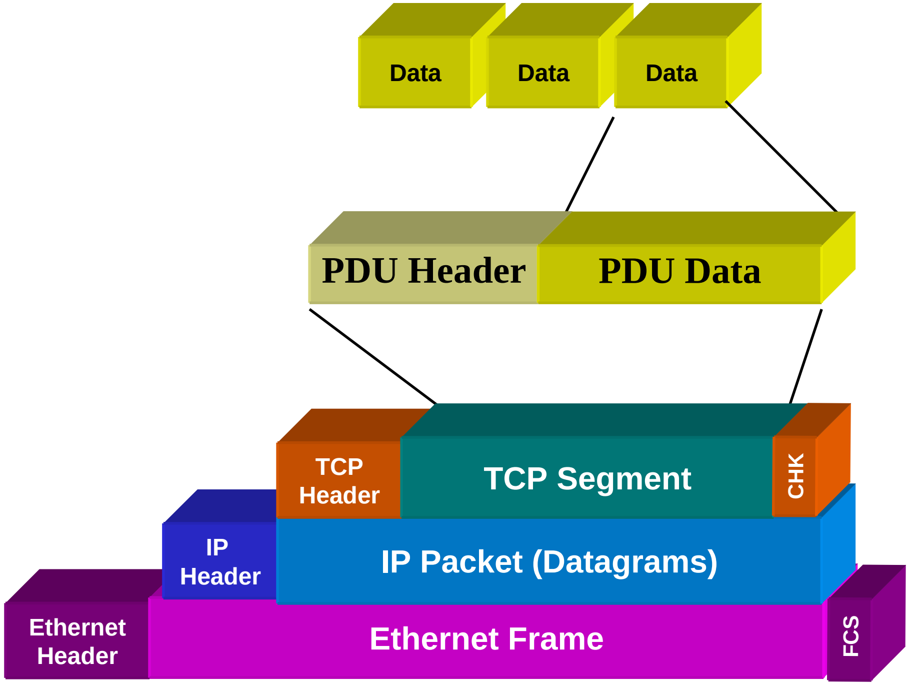


iSCSI Characteristics

- iSCSI is a SCSI transport protocol for mapping of block oriented storage data over TCP/IP networks.
- Storage generally identified for this application include disk, tape arrays and tape libraries.
- IP networks most applicable for this purpose are Gigabit Ethernet and in the future 10 Gigabit Ethernet, however for low performance applications 10/100bT will work.
- The iSCSI layer encapsulates the SCSI CDB into a iSCSI Protocol Data Unit (PDU) and forwards it to the Transmission Control Protocol (TCP) layer.
- The communications between the Initiator and Target will occur over one or more TCP connections.
- The TCP connections form a session and will carry the iSCSI PDU’s. The sessions are given an ID called a Connection ID (CID). There are two parts of the ID, Initiator Session ID (ISID) and Target ID (TSID) and together make up an “I_T nexus”.

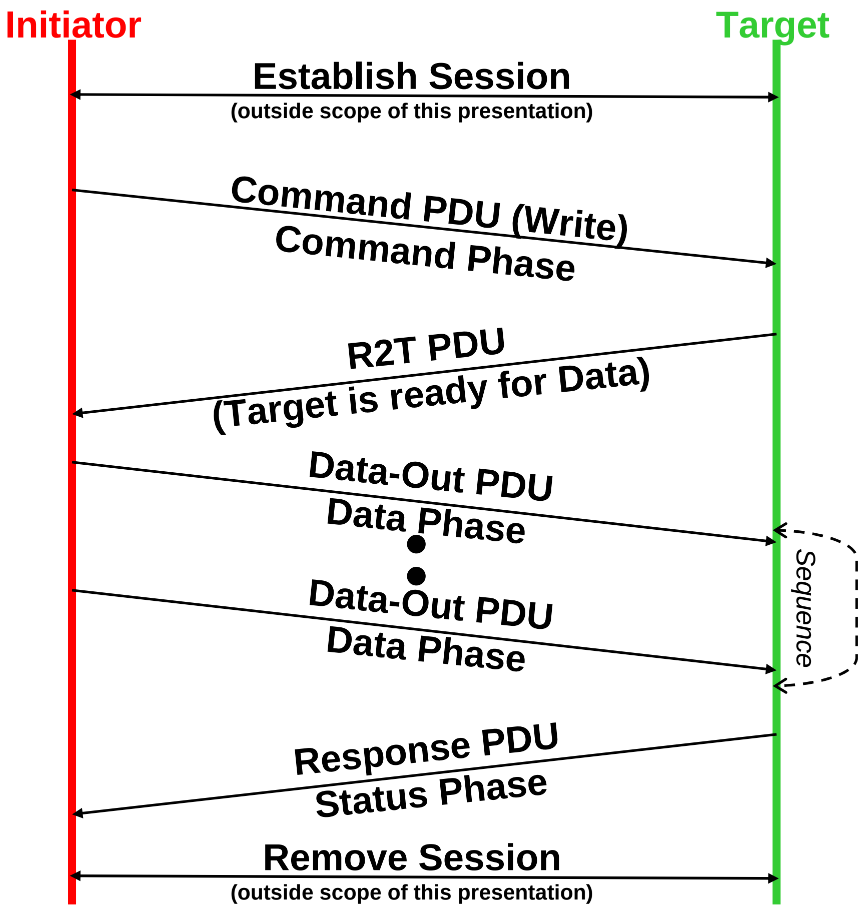
iSCSI Encapsulation





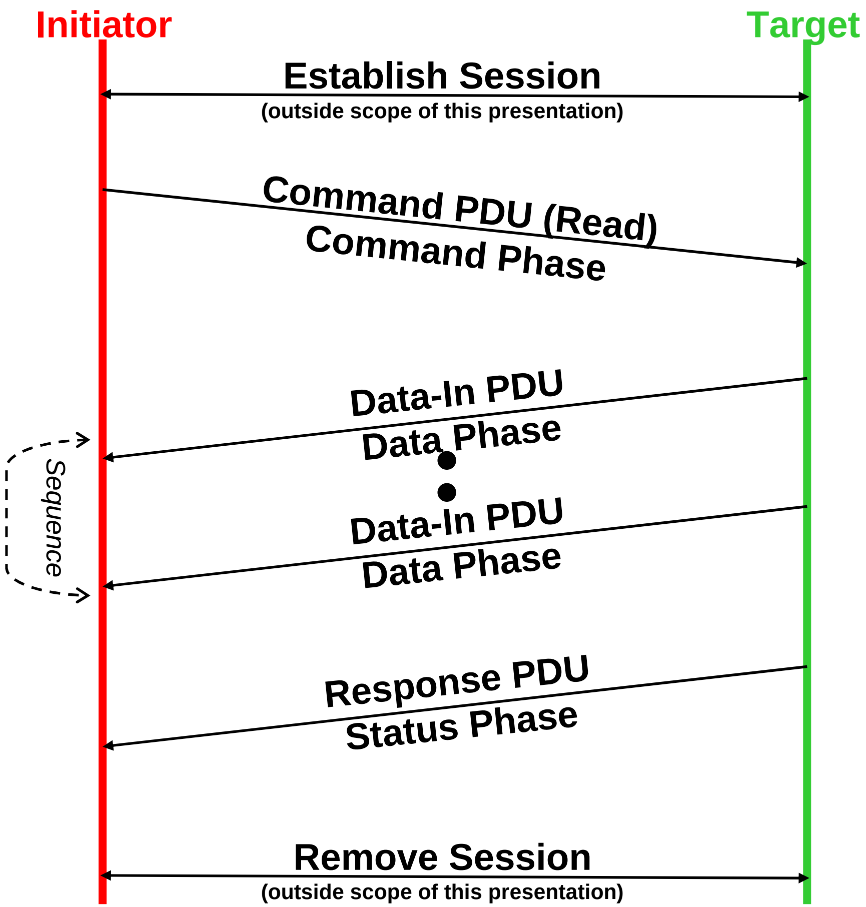
Write Command Example

- Before devices can communicate in iSCSI they must establish a session via login protocol
- Initiator sends Write command to Target
- Target replies with a ready to transmit R2T response
- Initiator sends Data frames
- When all Data is transferred, Target sends Status to Initiator
- Session could end at this point but typically doesn't



Read Command Example

- Before devices can communicate in iSCSI they must establish a session via login protocol
- Initiator sends Read command to Target
- Target sends Data frames
- When all Data is transferred, Target sends Status to Initiator
- Session could end at this point but typically doesn't



Summary

- SCSI is the language of LOVE
- Every storage architecture utilizes the SCSI model
- SCSI Commands sets are available for every type of storage device even including host-to-host
- An I/O Process is interface independent and is made up of:
  - Command Phase
  - Optional Data Phase
  - and a Status Phase
- SCSI language is here to stay and will be used in every storage architecture including:
  - SATA
  - SAS
- For more information on standards see: incits.org, t10.org, t11.org, t13.org, ietf.org, scsita.org, fibrechannel.org

Q&A / Feedback

- Please send any questions or comments on this presentation to SNIA: track-storage@snia.org

Many thanks to the following individuals for their contributions to this tutorial.

SNIA Education Committee

David Deming
Elaine Silber