Chapter 5
Direct-Attached Storage and Introduction to SCSI

Direct-Attached Storage (DAS) is an architecture where storage connects directly to servers. Applications access data from DAS using block-level access protocols. The internal HDD of a host, tape libraries, and directly connected external HDD packs are some examples of DAS.

KEY CONCEPTS

- Internal and External DAS
- SCSI Architecture
- SCSI Addressing

Although the implementation of storage networking technologies are gaining popularity, DAS has remained ideal for localized data access and sharing in environments that have a small number of servers. For example, small businesses, departments, and workgroups that do not share information across enterprises find DAS to be an appropriate solution. Medium-size companies use DAS for file serving and e-mail, while larger enterprises leverage DAS in conjunction with SAN and NAS.

This chapter details the two types of DAS along with their benefits and limitations. A major part of this chapter is devoted to detailing the types of storage devices used in DAS environments and SCSI—the most prominent protocol used in DAS.

5.1 Types of DAS

DAS is classified as internal or external, based on the location of the storage device with respect to the host.

5.1.1 Internal DAS

In *internal DAS* architectures, the storage device is internally connected to the host by a serial or parallel bus. The physical bus has distance limitations and can only be sustained over a shorter distance for high-speed connectivity. In addition, most internal buses can support only a limited number of devices, and they occupy a large amount of space inside the host, making maintenance of other components difficult.

5.1.2 External DAS

In *external DAS* architectures, the server connects directly to the external storage device (see Figure 5-1). In most cases, communication between the host and the storage device takes place over SCSI or FC protocol. Compared to internal DAS, an external DAS overcomes the distance and device count limitations and provides centralized management of storage devices.

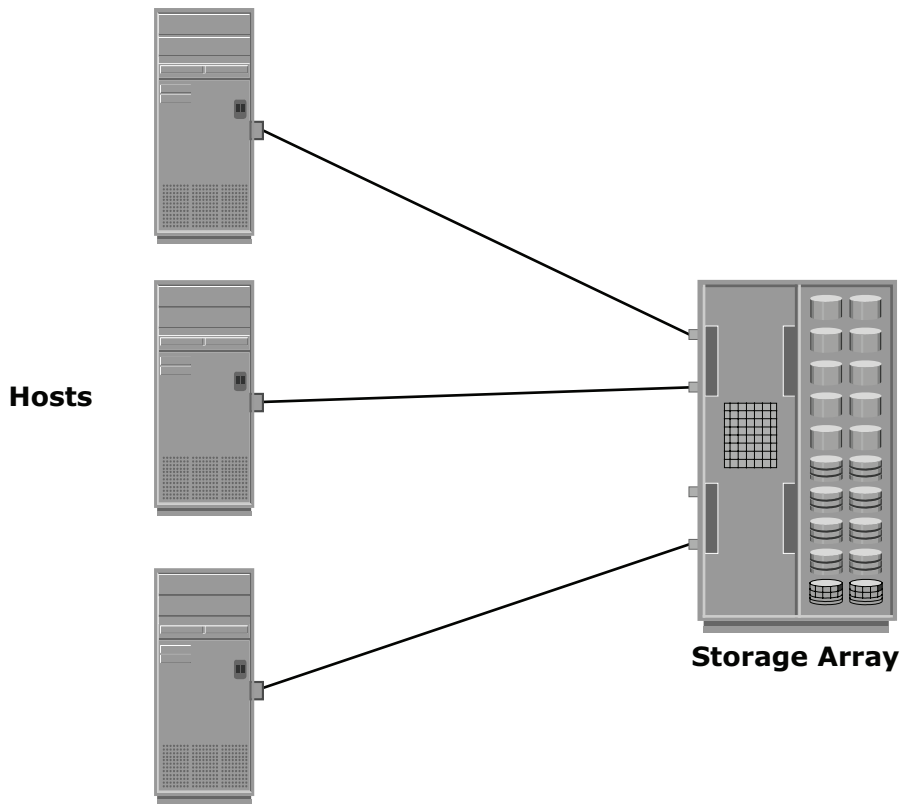


Figure 5-1: External DAS architecture

5.2 DAS Benefits and Limitations

DAS requires a relatively lower initial investment than storage networking. Storage networking architectures are discussed later in this book. DAS configuration is simple and can be deployed easily and rapidly. Setup is managed using host-based tools, such as the host OS, which makes storage management tasks easy for small and medium enterprises. DAS is the simplest solution when compared to other storage networking models and requires fewer management tasks, and less hardware and software elements to set up and operate.

However, DAS does not scale well. A storage device has a limited number of ports, which restricts the number of hosts that can directly connect to the storage. A limited bandwidth in DAS restricts the available I/O processing capability. When capacities are being reached, the service availability may be compromised, and this has a ripple effect on the performance of all hosts attached to that specific device or array. The distance limitations associated with implementing DAS because of direct connectivity requirements can be addressed by using Fibre Channel connectivity. DAS does not make optimal use of resources due to its limited ability to share front end ports. In DAS environments, unused resources cannot be easily re-allocated, resulting in islands of over-utilized and under-utilized storage pools.

Disk utilization, throughput, and cache memory of a storage device, along with virtual memory of a host govern the performance of DAS. RAID-level configurations, storage controller protocols, and the efficiency of the bus are additional factors that affect the performance of DAS. The absence of storage interconnects and network latency provide DAS with the potential to outperform other storage networking configurations.

5.3 Disk Drive Interfaces

The host and the storage device in DAS communicate with each other by using predefined protocols such as IDE/ATA, SATA, SAS, SCSI, and FC. These protocols are implemented on the HDD controller. Therefore, a storage device is also known by the name of the protocol it supports. This section describes each of these storage devices in detail.

5.3.1 IDE/ATA

An Integrated Device Electronics/Advanced Technology Attachment (IDE/ATA) disk supports the IDE protocol. The term IDE/ATA conveys the dual-naming conventions for various generations and variants of this interface. The IDE component in IDE/ATA provides the specification for the controllers connected

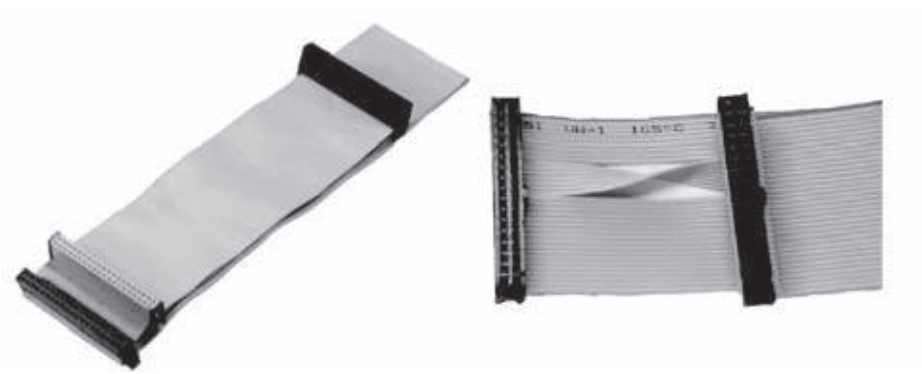
to the computer's motherboard for communicating with the device attached. The ATA component is the interface for connecting storage devices, such as CD-ROMs, floppy disk drives, and HDDs, to the motherboard.

IDE/ATA has a variety of standards and names, such as ATA, ATA/ATAPI, EIDE, ATA-2, Fast ATA, ATA-3, Ultra ATA, and Ultra DMA. The latest version of ATA—Ultra DMA/133—supports a throughput of 133 MB per second.

In a master-slave configuration, an ATA interface supports two storage devices per connector. However, if the performance of the drive is important, sharing a port between two devices is not recommended.

Figure 5-2 shows two commonly used IDE connectors attached to their cables. A 40-pin connector is used to connect ATA disks to the motherboard, and a 34-pin connector is used to connect floppy disk drives to the motherboard.

An IDE/ATA disk offers excellent performance at low cost, making it a popular and commonly used hard disk.



40-pin connector 34-pin connector

Figure 5-2: Common IDE connectors

5.3.2 SATA

A SATA (Serial ATA) is a serial version of the IDE/ATA specification. SATA is a disk-interface technology that was developed by a group of the industry's leading vendors with the aim of replacing parallel ATA.

A SATA provides point-to-point connectivity up to a distance of one meter and enables data transfer at a speed of 150 MB/s. Enhancements to the SATA have increased the data transfer speed up to 600 MB/s.

A SATA bus directly connects each storage device to the host through a dedicated link, making use of *low-voltage differential signaling (LVDS)*. LVDS is an electrical signaling system that can provide high-speed connectivity over

low-cost, twisted-pair copper cables. For data transfer, a SATA bus uses LVDS with a voltage of 250 mV.

A SATA bus uses a small 7-pin connector and a thin cable for connectivity. A SATA port uses 4 signal pins, which improves its pin efficiency compared to the parallel ATA that uses 26 signal pins, for connecting an 80-conductor ribbon cable to a 40-pin header connector.

SATA devices are *hot-pluggable,* which means that they can be connected or removed while the host is up and running. A SATA port permits single-device connectivity. Connecting multiple SATA drives to a host requires multiple ports to be present on the host. Single-device connectivity enforced in SATA, eliminates the performance problems caused by cable or port sharing in IDE/ATA.

5.3.3 Parallel SCSI

SCSI is available in a variety of interfaces. Parallel SCSI (referred to as SCSI) is one of the oldest and most popular forms of storage interface used in hosts. SCSI is a set of standards used for connecting a peripheral device to a computer and transferring data between them. Often, SCSI is used to connect HDDs and tapes to a host. SCSI can also connect a wide variety of other devices such as scanners and printers. Communication between the hosts and the storage devices uses the SCSI command set, described later in this chapter.

Since its inception, SCSI has undergone rapid revisions, resulting in continuous performance improvements. The oldest SCSI variant, called SCSI-1 provided data transfer rate of 5 MB/s; SCSI Ultra 320 provides data transfer speeds of 320 MB/s. Other variants of SCSI and transfer speeds are listed in Table 5-2.

Table 5-1 provides a comparison between the features of IDE/ATA and SCSI, the two most popular hard disk interfaces.

Table 5-1: Comparison of IDE/ATA with SCSI

FEATURE	IDE/ATA	SCSI
Speed	100, 133, 150 MB/s	320 MB/s
Connectivity	Internal	Internal and external
Cost	Low	Moderate to high
Hot-pluggable	No	Yes
Performance	Moderate to low	High
Ease of configuration	High	Low to moderate
Maximum number of devices supported	2	16

SERIAL ATTACHED SCSI DISKS

Serial Attached SCSI (SAS) is the evolution of SCSI beyond SCSI Ultra 320. SAS addresses the scalability, performance, reliability, and manageability requirements of a data center while leveraging a common electrical and physical connection interface with SATA. SAS uses SCSI commands for communication and is pin compatible with SATA. SAS supports data transfer rate of 3 Gb/s (SAS 300). It supports dual porting, full-duplex, device addressing, and uses a simplified protocol to minimize interoperability issues between controllers and drives. It also enables connectivity to multiple devices through expanders and is commonly preferred over SCSI in high-end servers for faster disk access.

5.4 Introduction to Parallel SCSI

Shugart Associates and NCR developed a system interface in 1981 and named it Shugart Associates System Interface (SASI). SASI was developed to build a proprietary, high-performance standard primarily for use by these two companies. However, to increase the acceptance of SASI in the industry, the standard was updated to a more robust interface and renamed SCSI. In 1986, the American National Standards Institution (ANSI) acknowledged the new SCSI as an industry standard.

SCSI, first developed for hard disks, is often compared to IDE/ATA. SCSI offers improved performance and expandability and compatibility options, making it suitable for high-end computers. However, the high cost associated with SCSI limits its popularity among home or business desktop users.

FC DISKS

FC disks use the FC-AL topology (FC-AL2 over copper). FC is the specification in storage networking for gigabit speed network technology. Although FC disks are used extensively with SAN technology, they can also be implemented for DAS. Faster access speeds of Fibre Channel (8.5 Gb/s) for 8 GFC (Gigabit Fibre Channel) are used in high-end storage system.

5.4.1 Evolution of SCSI

Prior to the development of SCSI, the interfaces used to communicate with devices varied with each device. For example, an HDD interface could only be used with a hard disk drive. SCSI was developed to provide a device-independent

mechanism for attaching to and accessing host computers. SCSI also provided an efficient peer-to-peer I/O bus that supported multiple devices. Today, SCSI is commonly used as a hard disk interface. However, SCSI can be used to add devices, such as tape drives and optical media drives, to the host computer without modifying the system hardware or software. Over the years, SCSI has undergone radical changes and has evolved into a robust industry standard. Various SCSI standards are detailed in this section.

SCSI-1

SCSI-1, renamed to distinguish it from other SCSI versions, is the original standard that the ANSI approved. SCSI-1 defined the basics of the first SCSI bus, including cable length, signaling characteristics, commands, and transfer modes. SCSI-1 devices supported only single-ended transmission and *passive termination*. SCSI-1 used a narrow 8-bit bus, which offered a maximum data transfer rate of 5 MB/s.

SCSI-1 implementations resulted in incompatible devices and several subsets of standards. Due to these issues, work on improving the SCSI-1 standard began in 1985, a year before its formal approval.

SCSI-2

To control the various problems caused by the nonstandard implementation of the original SCSI, a working paper was created to define a set of standard commands for a SCSI device. This set of standards, called the *common command set (CCS)*, formed the basis of the SCSI-2 standard.

SCSI-2 was focused on improving performance, enhancing reliability, and adding additional features to the SCSI-1 interface, in addition to standardizing and formalizing the SCSI commands. The ANSI withdrew the SCSI-1 standard and, in 1994, approved SCSI-2 as one large document: X3.131-1994. The transition from SCSI-1 to SCSI-2 did not raise much concern because SCSI-2 offered backward compatibility with SCSI-1.

SCSI-3

In 1993, work began on developing the next version of the SCSI standard, SCSI-3. Unlike SCSI-2, the SCSI-3 standard document is comprised different but related standards, rather than one large document.

5.4.2 SCSI Interfaces

Along with the evolving SCSI standards, SCSI interfaces underwent several improvements. Parallel SCSI, or SCSI parallel interface (SPI), was the original

SCSI interface (Table 5-2 lists some of the available parallel SCSI interfaces). The SCSI design is now making a transition into Serial Attached SCSI (SAS), which is based on a serial point-to-point design, while retaining the other aspects of the SCSI technology.

In addition to the interfaces listed in Table 5-2, many interfaces are not complete SCSI standards, but still implement the SCSI command model.

Table 5-2: SCSI Interfaces

INTERFACE	STANDARD	BUS WIDTH	CLOCK SPEED	MAX THROUGHPUT	MAX DEVICES
SCSI-1	SCSI-1	8	5 MHz	5 MB/s	8
Fast SCSI	SCSI-2	8	10 MHz	10 MB/s	8
Fast Wide SCSI	SCSI-2; SCSI-3 SPI	16	10 MHz	20 MB/s	16
Ultra SCSI	SCSI-3 SPI	8	20 MHz	20 MB/s	8
Ultra Wide SCSI	SCSI-3 SPI	16	20 MHz	40 MB/s	16
Ultra2 SCSI	SCSI-3 SPI-2	8	40 MHz	40 MB/s	8
Ultra2 Wide SCSI	SCSI-3 SPI-2	16	40 MHz	80 MB/s	16
Ultra3 SCSI	SCSI-3 SPI-3	16	40 MHz DDR	160 MB/s	16
Ultra320 SCSI	SCSI-3 SPI-4	16	80 MHz DDR	320 MB/s	16
Ultra640 SCSI	SCSI-3 SPI-5	16	160 MHz DDR	640 MB/s	16

FIBRE CHANNEL PROTOCOL

FCP (Fibre Channel Protocol) is the implementation of SCSI-3 over Fibre Channel networks. A new SCSI design, iSCSI, implements the SCSI-3 standard over Internet Protocol (IP) and uses TCP as a transport mechanism.

5.4.3 SCSI-3 Architecture

The SCSI-3 architecture defines and categorizes various SCSI-3 standards and requirements for SCSI-3 implementations. (For more information, see Technical Committee T10 "SCSI Architecture Model-3 (SAM-3)" document from `www.t10.org`.) The SCSI-3 architecture was approved and published as the standard X.3.270-1996 by the ANSI. This architecture helps developers, hardware designers, and users to understand and effectively utilize SCSI. The three major components of a SCSI architectural model are as follows:

- **SCSI-3 command protocol:** This consists of primary commands that are common to all devices as well as device-specific commands that are unique to a given class of devices.
- **Transport layer protocols:** These are a standard set of rules by which devices communicate and share information.
- **Physical layer interconnects:** These are interface details such as electrical signaling methods and data transfer modes.

Common access methods are the ANSI software interfaces for SCSI devices. Figure 5-3 shows the SCSI-3 standards architecture with interrelated groups of other standards within SCSI-3.

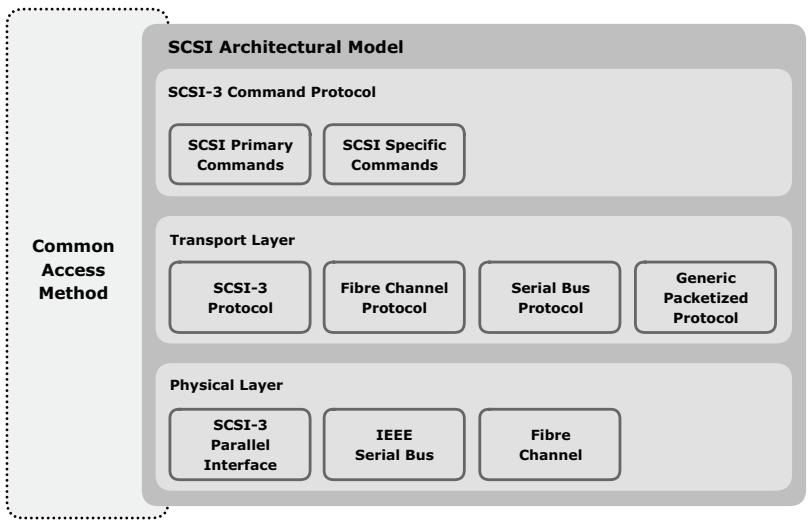


Figure 5-3: SCSI-3 standards architecture

SCSI-3 Client-Server Model

SCSI-3 architecture derives its base from the client-server relationship, in which a client directs a service request to a server, which then fulfills the client's request. In a SCSI environment, an initiator-target concept represents the client-server model. In a SCSI-3 client-server model, a particular SCSI device acts as a SCSI target device, a SCSI initiator device, or a SCSI target/initiator device. Each device performs the following functions:

- **SCSI initiator device:** Issues a command to the SCSI target device, to perform a task. A SCSI host adaptor is an example of an initiator.
- **SCSI target device:** Executes commands to perform the task received from a SCSI initiator. Typically a SCSI peripheral device acts as a target device. However, in certain implementations, the host adaptor can also be a target device.

Figure 5-4 displays the SCSI-3 client-server model, in which a SCSI initiator, or a client, sends a request to a SCSI target, or a server. The target performs the tasks requested and sends the output to the initiator, using the protocol service interface.

A SCSI target device contains one or more logical units. A logical unit is an object that implements one of the device functional models as described in the SCSI command standards. The logical unit processes the commands sent by a SCSI initiator. A logical unit has two components, a *device server* and a *task manager*, as shown in Figure 5-4. The device server addresses client requests, and the task manager performs management functions.

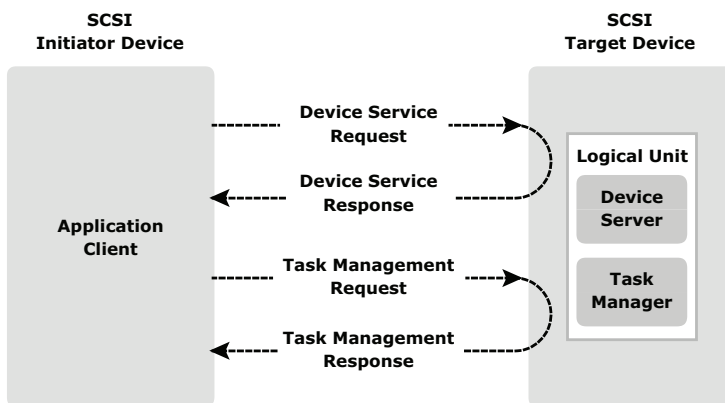


Figure 5-4: SCSI-3 client-server model

The SCSI initiator device is comprised of an application client and task management function, which initiates device service and task management requests. Each device service request contains a *Command Descriptor Block (CDB)*. The CDB defines the command to be executed and lists command-specific inputs and other parameters specifying how to process the command. The application client also creates tasks, objects within the logical unit, representing the work associated with a command or a series of linked commands. A task persists until either the "task complete response" is sent or the task management function or exception condition ends it.

The SCSI devices are identified by a specific number called a SCSI ID. In narrow SCSI (bus width=8), the devices are numbered 0 through 7; in wide (bus width=16) SCSI, the devices are numbered 0 through 15. These ID numbers set the device priorities on the SCSI bus. In narrow SCSI, 7 has the highest priority and 0 has the lowest priority. In wide SCSI, the device IDs from 8 to 15 have the highest priority, but the entire sequence of wide SCSI IDs has lower priority than narrow SCSI IDs. Therefore, the overall priority sequence for a wide SCSI is 7, 6, 5, 4, 3, 2, 1, 0, 15, 14, 13, 12, 11, 10, 9, and 8.

When a device is initialized, SCSI allows for automatic assignment of device IDs on the bus, which prevents two or more devices from using the same SCSI ID.

SCSI Ports

SCSI ports are the physical connectors that the SCSI cable plugs into for communication with a SCSI device. A SCSI device may contain target ports, initiator ports, target/initiator ports, or a target with multiple ports. Based on the port combinations, a SCSI device can be classified as an initiator model, a target model, a combined model, or a target model with multiple ports (see Figure 5-5).

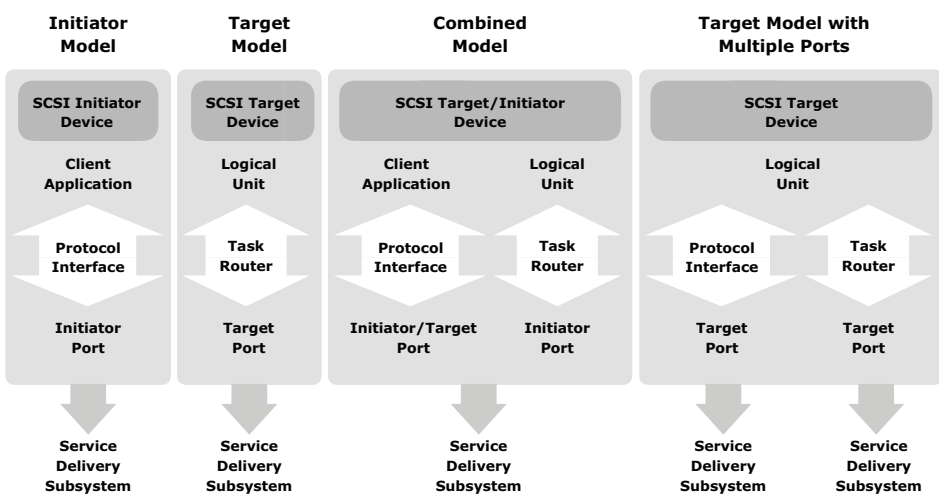


Figure 5-5: SCSI device models with different port configurations

In an initiator model, the SCSI initiator device has only initiator ports. Therefore, the application client can only initiate requests to the service delivery subsystem and receive confirmation. This device cannot serve any requests, and therefore does not contain a logical unit.

Similarly, a SCSI target device with only a target port can serve requests but cannot initiate them. The SCSI target/initiator device has a target/initiator port that can switch orientations depending on the role it plays while participating in an I/O operation. To cater to service requests from multiple devices, a SCSI device may also have multiple ports of the same orientation (target).

SCSI Communication Model

A SCSI communication model (see Figure 5-6) is comprised of three interconnecting layers as defined in the SAM-3 and is similar to the OSI seven-layer model. Lower-level layers render their services to the upper-level layers. A high-level layer communicates with a low-level layer by invoking the services that the low-level layer provides. The protocol at each layer defines the communication between peer layer entities.

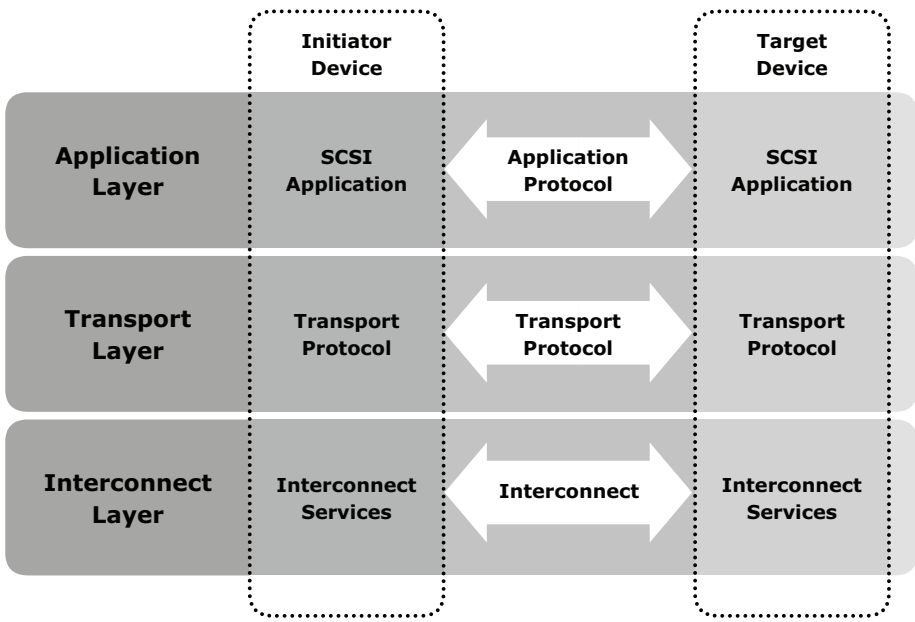


Figure 5-6: SCSI communication model

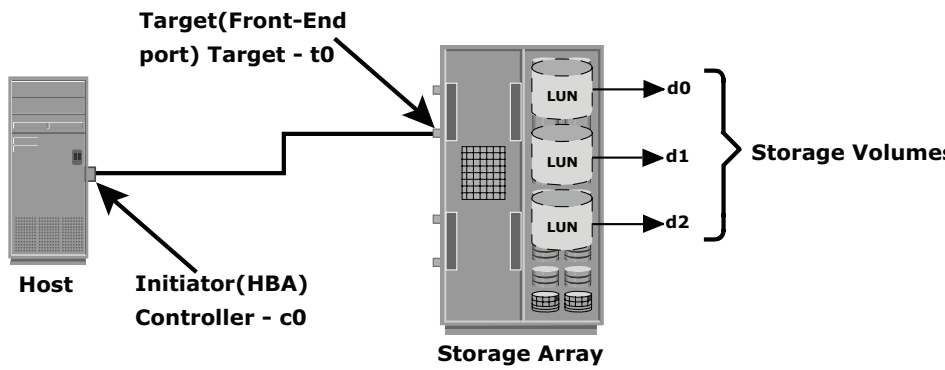
There are three layers in the SCSI communication model:

- **SCSI application layer (SAL):** This layer contains both client and server applications that initiate and process SCSI I/O operations using a SCSI application protocol.

- **SCSI transport protocol layer (STPL):** This layer contains the services and protocols that allow communication between an initiator and targets.
- **Interconnect layer:** This layer facilitates data transfer between the initiator and targets. The interconnect layer is also known as the *service delivery subsystem* and comprises the services, signalling mechanisms, and interconnects for data transfer.

5.4.4 Parallel SCSI Addressing

In the Parallel SCSI Initiator-Target communication (see Figure 5-7), an initiator ID uniquely identifies the initiator and is used as an originating address. This ID is in the range of 0 to 15, with the range 0 to 7 being the most common. A target ID uniquely identifies a target and is used as the address for exchanging commands and status information with initiators. The target ID is in the range of 0 to 15.



Host Addressing :
Storage Volume 1 - c0 t0 d0
Storage Volume 2 - c0 t0 d1
Storage Volume 3 - c0 t0 d2

Figure 5-7: SCSI Initiator-Target communication

SCSI addressing is used to identify hosts and devices. In this addressing, the UNIX naming convention is used to identify a disk and the three identifiers—initiator ID, target ID, and a LUN—in the **c**n|**t**n|**d**n format, which is also referred as *ctd addressing*. Here, cn is the initiator ID, commonly referred to as the controller ID; tn is the target ID of the device, such as t0, t1, t2, and so on; and dn is the device number reflecting the actual address of the device unit, such as d0, d1, and d2. A LUN identifies a specific logical unit in a target. The implementation of SCSI addressing may differ from one vendor to another. Figure 5-7 shows ctd addressing in the SCSI architecture.

5.5 SCSI Command Model

In the SCSI communication model (regardless of interface type: Parallel SCSI, SAS, or FC-AL2), the initiator and the target communicate with each other using a command protocol standard. The original SCSI command architecture was defined for parallel SCSI buses and later adopted for iSCSI and serial SCSI with minimal changes. Some of the other technologies that use the SCSI command set include ATA Packet Interface, USB Mass Storage class, and FireWire SBP-2. The SCSI command model is defined with the CDB. The CDB structure is detailed next.

5.5.1 CDB Structure

The initiator sends a command to the target in a CDB structure. The CDB defines the operation that corresponds to the initiator's request to be performed by the device server. The CDB consists of a 1-byte *operation code* followed by 5 or more bytes containing *command-specific parameters* and ending with a 1-byte *control field* (see Figure 5-8). The command specification is less than or equal to 16 bytes. The length of a CDB varies depending on the command and its parameters.

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Command-Specific Parameters							
n-1								
n	Control							

Figure 5-8: CDB structure

5.5.2 Operation Code

The operation code consists of group and command code fields (see Figure 5-9).

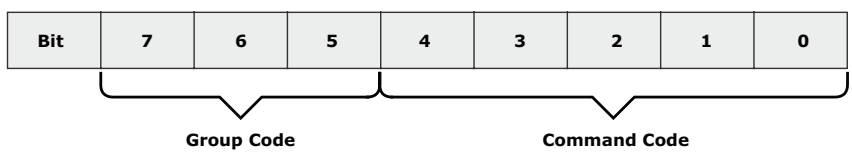


Figure 5-9: Operation code field

The *group code field* is a 3-bit field that specifies the length of the command-specific parameters shown in Table 5-3.

Table 5-3: Group Codes

GROUP CODE	COMMAND-SPECIFIC PARAMETERS
0	6 bytes
1 and 2	10 bytes
3	Reserved
4	16 bytes
5	12 bytes
6 and 7	Vendor specific

The *command code field* is a 5-bit field that allows 32 command codes in each group, for a total of 256 possible operation codes (refer to Figure 5-9). However, there are only about 60 different SCSI commands that facilitate communication between an initiator and a target. Some of the commonly used SCSI commands are shown in Table 5-4.

Table 5-4: Common SCSI Commands

COMMAND	DESCRIPTION
READ	Reads data from a device
WRITE	Writes data to a device
TEST UNIT READY	Queries the device to check whether it is ready for data transfer
INQUIRY	Returns basic information, which is also used to ping the device
REPORT LUNS	Lists the logical unit numbers
SEND AND RECEIVE DIAGNOSTIC RESULTS	Runs a simple self-test or a specialized test defined in a diagnostic page
FORMAT UNIT	Sets all sectors to all zeroes and allocates logical blocks, avoiding defective sectors
LOG SENSE	Returns current information from log pages
LOG SELECT	Used to modify data in the log pages of a SCSI target device
MODE SENSE	Returns current device parameters from mode pages
MODE SELECT	Sets device parameters on a mode page

5.5.3 Control Field

The *control field* is a 1-byte field and is the last byte of every CDB. The control field implements the *Normal Auto Contingent Allegiance (NACA)* and *link bits*. The control field structure is shown in Figure 5-10.

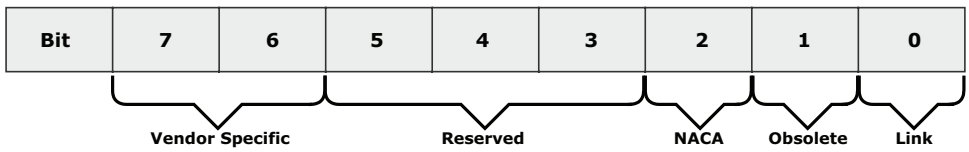


Figure 5-10: Control field

The NACA bit and associated ACA mechanism are almost never used. The NACA bit specifies whether an *auto contingent allegiance (ACA)* is established if the command returns with `CHECK CONDITION` status.

The link bit is unused in practice. This bit can be used to continue the task across multiple commands. A link bit of 1 indicates that the initiator has requested continuation of the task across two or more SCSI commands. Bits 3 to 5 are reserved and the last two bits are vendor-specific bits.

5.5.4 Status

After command execution, the logical unit sends the status along with the flag to the application client. The status, except `INTERMEDIATE` or `INTERMEDIATE-CONDITION MET`, indicates the end of the task. Table 5-5 shows the hexadecimal (h) codes and the associated status.

Table 5-5: SCSI Status Codes

STATUS BYTE CODES	STATUS
0h	GOOD
2h	CHECK CONDITION
4h	CONDITION MET
8h	BUSY
10h	INTERMEDIATE
14h	INTERMEDIATE-CONDITION MET
18h	RESERVATION CONFLICT
22h	COMMAND TERMINATED
28h	TASK SET FULL
30h	ACA ACTIVE
All other codes	Reserved

Summary

DAS offers several advantages, such as simplicity, ease of configuration, and manageability, but its limitations in scalability and availability restrict its use as an enterprise storage solution. DAS is still used in small and medium enterprises for localized data access and sharing and in environments that leverage DAS in conjunction with SAN and NAS. Storage devices such as IDE/ATA disks are popularly used to build DAS. SATA, SAS, SCSI, and FC are other protocols implemented on disk controllers also used in DAS environments.

SAN and NAS are preferred storage solutions for enterprises due to the limitations of DAS. The SCSI protocol is the basic building block of SAN. SCSI as a client-server model communicates between the initiator and the target using SCSI commands and SCSI port addresses. FCP (Fibre Channel Protocol), used in SAN, is an implementation of SCSI-3 over FC. NAS uses TCP/IP as a transport protocol between the hosts and NAS devices. The next chapter describes the storage networking technology architectures based on SCSI.

EXERCISES

1. **DAS provides an economically viable alternative to other storage networking solutions. Justify this statement.**
2. **How is the priority sequence established in a wide SCSI environment?**
3. **Why is SCSI performance superior to that of IDE/ATA? Explain the reasons from an architectural perspective.**
4. **Research blade server architecture and discuss the limitations of DAS for this architecture.**
5. **What would you consider while choosing serial or parallel data transfer in a DAS implementation? Explain your answer and justify your choice.**
6. **If three hard disk drives are connected in a daisy chain and communicate over SCSI, explain how the CPU will perform I/O operations with a particular device.**
7. **A UNIX host has a path to a storage device that shows as c0 t1 d3. Draw a diagram to show the path and explain what it means.**