

Date: May 02, 2001

To: T10 Committee (SCSI)

From: George Penokie (Tivoli Systems)

Subject: Defining Targets/Initiators as Ports

Overview

In looking over the issue of how SCSI should deal with multiple port devices it appears a reasonable solution would be to change the notion of what a target and initiator is defined as. If you confine a target or initiator to only contain a service delivery port and allow a SCSI device to contain any number of targets or initiators then defining multi-port devices becomes an easier task. In many ways the current definitions and descriptions already support this idea.

Multiple port SCSI Device model

The following show the changes in SAM-2 that would be required to change to support a port = target/initiator model.

SAM-2 Glossary changes:

0.0.1 device identifier: Synonymous with SCSI port identifier (see 0.0.14).

0.0.2 device model: The description of a type of SCSI target device (e.g., block, stream).

0.0.3 initiator: Synonymous with SCSI initiator port (see 0.0.12).

0.0.4 initiator identifier: Synonymous with initiator port identifier (see 4.7.3).

0.0.5 initiator port identifier: A value by which a SCSI initiator port is referenced within a domain (see 4.7.3).

0.0.6 name: A label of an object that is unique within a specified context and should never change (e.g., the term name, and world wide identification (WWID) may be interchangeable).

0.0.7 port: Synonymous with SCSI port (see 0.0.13).

0.0.8 SCSI device: A device that contains one or more SCSI port that are connected to a service delivery subsystem and supports a SCSI application protocol.

0.0.9 SCSI device identifier: Synonymous with SCSI port identifier (see 0.0.14).

0.0.10 SCSI identifier: Synonymous with SCSI port identifier (see 0.0.14).

0.0.11 SCSI initiator device: A SCSI device containing application clients and SCSI initiator ports that originate device service and task management requests to be processed by a target SCSI device. When used this term refers to SCSI initiator devices or SCSI target/initiator devices that are using the SCSI target/initiator port as a SCSI initiator port.

0.0.12 SCSI initiator port: A SCSI initiator device object acts as the connection between application clients and the service delivery subsystem through which requests and responses are routed. In all cases when this term is used it refers to an initiator port or a SCSI target/initiator port operating as a SCSI initiator port.

0.0.13 SCSI port: A device-resident object that connects the application client, device server or task manager to the service delivery subsystem through which requests and responses are routed. The SCSI port is synonymous with port and either a SCSI initiator port (see 0.0.12) or a SCSI target port ((see 0.0.16)).

0.0.14 SCSI port identifier: A value by which a SCSI port is referenced within a domain. The SCSI port identifier is either an initiator port identifier (see 0.0.7) or a target port identifier (see 0.0.21).

0.0.15 SCSI target device: A SCSI device containing logical units and SCSI target ports that receives device service and task management requests for processing. When used this term refers to SCSI target devices or SCSI target/initiator devices that are using the SCSI target/initiator port as a SCSI target port.

0.0.16 SCSI target port: A SCSI target device object that contains a task router and acts as the connection between device servers and task managers and the service delivery subsystem through which requests and responses are routed. When this term is used it refers to a SCSI target port or a SCSI target/initiator port operating as a SCSI target port.

0.0.17 SCSI target/initiator device: A device that has all the characteristics of a SCSI target device and a SCSI initiator device.

0.0.18 SCSI target/initiator port: A device-resident object that has all the characteristics of a SCSI target port and a SCSI initiator port.

0.0.19 target: Synonymous with SCSI target port (see 0.0.16).

0.0.20 target identifier: Synonymous with target port identifier (see 4.7.3).

0.0.21 target port identifier: A value by which a SCSI target port is referenced within a domain (see 4.7.3).

SAM-2 Section 4.3 The SCSI client-server model

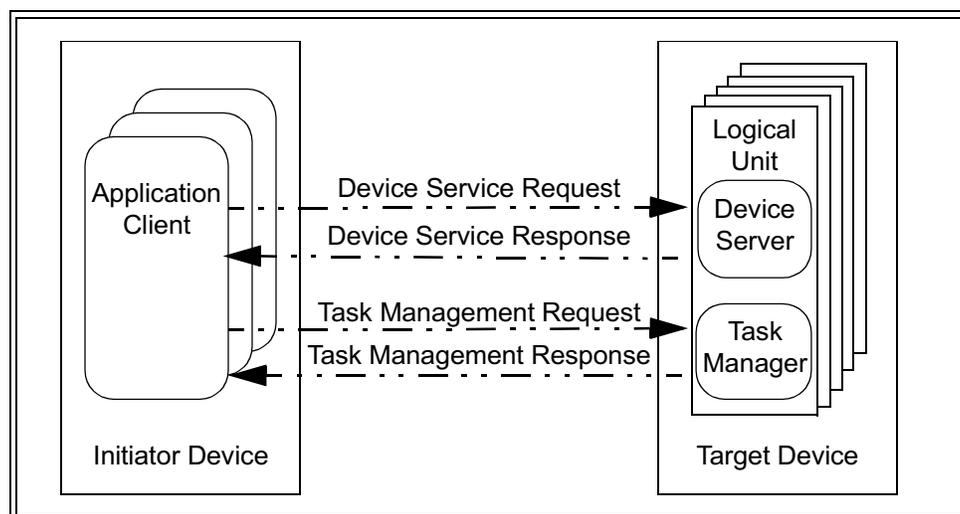


Figure 1 — SCSI client-server model

SAM-2 Section 4.4 The SCSI structural model

The SCSI structural model represents a view of the elements comprising a SCSI I/O system as seen by the application clients interacting with the system. This view is similar to that seen by a CAM device driver interacting with the system through the CAM SIM layer. As shown in figure 2, the fundamental object is the SCSI domain, that represents an I/O system. A domain is made up of SCSI devices and a service delivery subsystem, that transports commands and data. A SCSI device contains application clients or device servers or both and the infrastructure to support them.

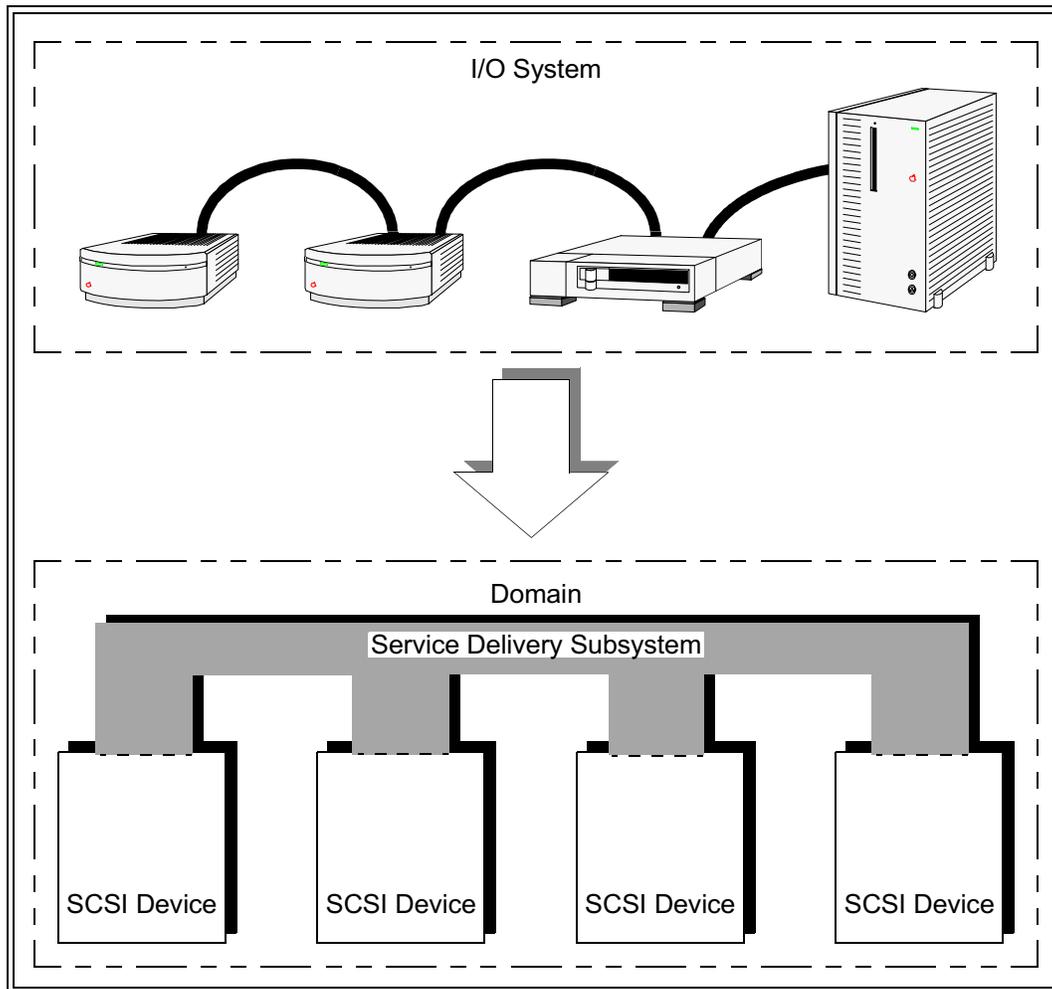


Figure 2 — SCSI I/O system and domain model

Figure 3 shows the main functional components of the SCSI domain. The following subclauses define these components in greater detail.

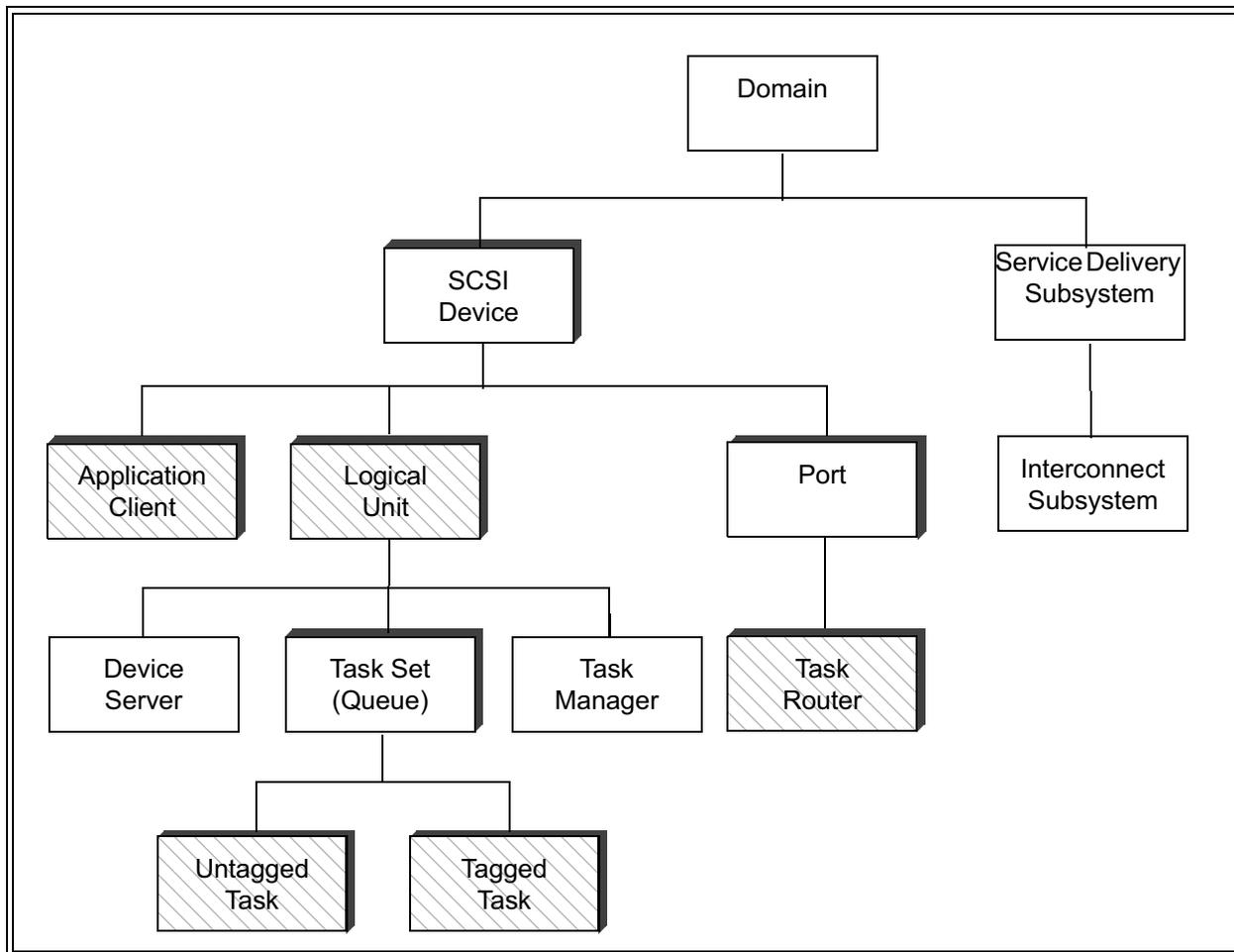


Figure 3 — SCSI domain model

SAM-2 Section 4.5 SCSI domain

A SCSI domain is composed of at least one SCSI device, at least one target port and at least one initiator port interconnected by a service delivery subsystem (see figure 4).

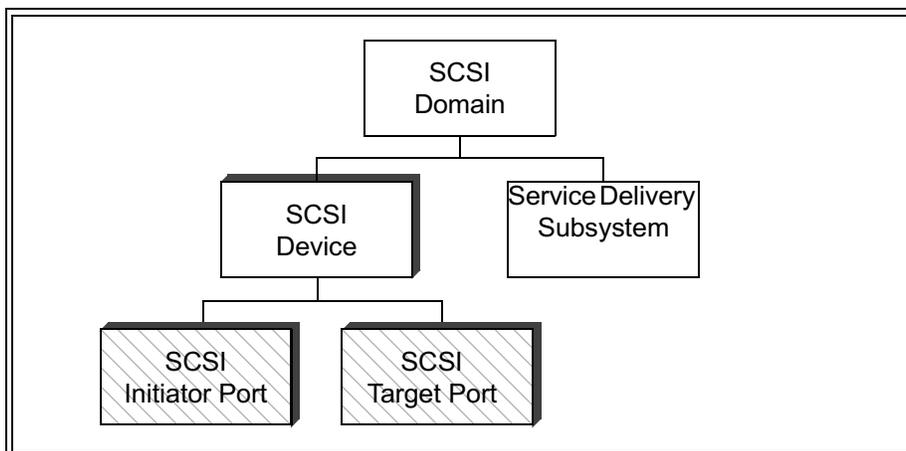


Figure 4 — SCSI Domain model

A SCSI device is an object that originates or services SCSI commands. As described in xxx, when a SCSI device originates a command it is called a SCSI initiator device and that command is transmitted through an initiator port or a SCSI target/initiator port. A SCSI device containing logical units that service commands is called a SCSI target device and receives commands through a SCSI target port or a SCSI target/initiator port. The service delivery subsystem connects all the SCSI ports in the SCSI domain, providing a subsystem through which application clients and device servers communicate (see xxx). The boundaries of a SCSI domain are established by the system implementor, within the constraints of a specific SCSI protocol and interconnect standards.

SAM-2 Section 4.6 The service delivery subsystem

The service delivery subsystem connects SCSI ports and is composed of an interconnect subsystem (see figure 5).

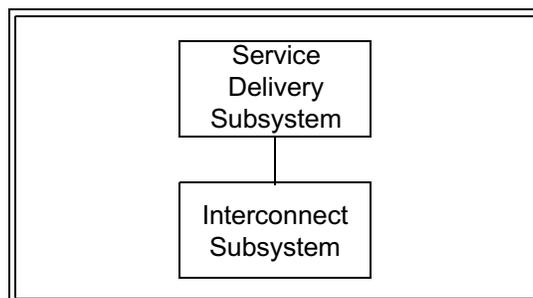


Figure 5 — Service delivery subsystem model

The interconnect subsystem is a set of one or more physical interconnects that appear to a client or server as a single path for the transfer of requests, responses, and data between SCSI devices.

The service delivery subsystem is assumed to provide error-free transmission of requests and responses between client and server. Although a device driver in a SCSI implementation may perform these transfers through several interactions with its SCSI protocol layer, the architecture model portrays each operation, from the viewpoint of the application client, as occurring in one discrete step. In this model, the data comprising an outgoing request is sent in a single package containing all the information required to process the remote procedure call. Similarly, an incoming server response is returned in a package enclosing the output data and status. The request or response package is sent when it is passed to the a SCSI port for transmission; it is in transit until delivered and received when it has been forwarded to the receiver via the destination device's SCSI port.

SAM-2 Section 4.6.1 Synchronizing client and server states

The client is usually informed of changes in server state through the arrival of server responses. In the architecture model such state changes occur after the server has sent the associated response and possibly before the response has been received by the SCSI initiator device. Some SCSI protocols, however, may require the SCSI target device to verify that the response has been received successfully before completing a state change. State changes controlled in this manner are said to be synchronized. Since synchronized state changes are not assumed or required by the architecture model, there may be a time lag between the occurrence of a state change within the SCSI target device and the SCSI initiator device's awareness of that change.

The model assumes that state synchronization, if required by a SCSI protocol standard, is enforced by the service delivery subsystem transparently to the server. That is, whenever the server invokes a protocol service to return a response as described in 6.7 and 5.3, it is assumed that the service delivery subsystem for such a protocol will not return control to the server until the response has been successfully delivered to the SCSI initiator device.

SAM-2 Section 4.6.2 Request/Response ordering

In this standard, request or response transactions are said to be in order if, relative to a given pair of sending and receiving SCSI ports, transactions are delivered in the order they were sent.

A sender may occasionally require control over the order in which its requests or responses are presented to the receiver (e.g., the sequence in which requests are received is important when a SCSI initiator device issues a series of SCSI commands with the ORDERED attribute to a logical unit as described in clause 7). In this case, the order in which these commands are completed, and hence the final state of the logical unit, may depend on the order in which these commands are received. Similarly, the SCSI initiator device acquires knowledge about the state of pending commands and task management functions and may subsequently take action based on the nature and sequence of SCSI target device responses (e.g., if the SCSI initiator device aborts a command whose completion response is in transit and the abort response is received out of order, the SCSI initiator device could incorrectly conclude that no further responses are expected from that command).

The manner in which ordering constraints are established is vendor-specific. An implementation may choose to delegate this responsibility to the application client (e.g., the device driver). In some cases, in-order delivery may be an intrinsic property of the service delivery subsystem or a requirement established by the SCSI protocol standard.

The SCSI architecture model assumes in-order delivery to be a property of the service delivery subsystem. This assumption is made to simplify the description of behavior and does not constitute a requirement. This standard makes no assumption about, or places any requirement on the ordering of requests or responses between tasks or task management functions received from different SCSI initiator ports.

SAM-2 Section 4.7 SCSI devices

A SCSI device is a SCSI target device, a SCSI initiator device, or a SCSI target/initiator device.

A SCSI initiator device contains at least one SCSI initiator port and is capable of originating SCSI commands and task management requests (see xxx). A SCSI target device contains at least one SCSI target port is capable of processing SCSI commands and task management requests (see xxx). A SCSI target/initiator device contains at least one SCSI target/initiator port and is capable of originating and processing SCSI commands and task management requests (see xxx). To be functional, a SCSI domain needs to contain a SCSI target port or a SCSI target/initiator port operating as a SCSI target port and another SCSI initiator port or SCSI target/initiator port operating as a SCSI initiator port.

SAM-2 section 4.7.1 SCSI initiator device

A SCSI initiator device (see figure 6) contains:

- a) one more SCSI initiator ports each containing an initiator port identifier; and
- b) zero or more application clients.

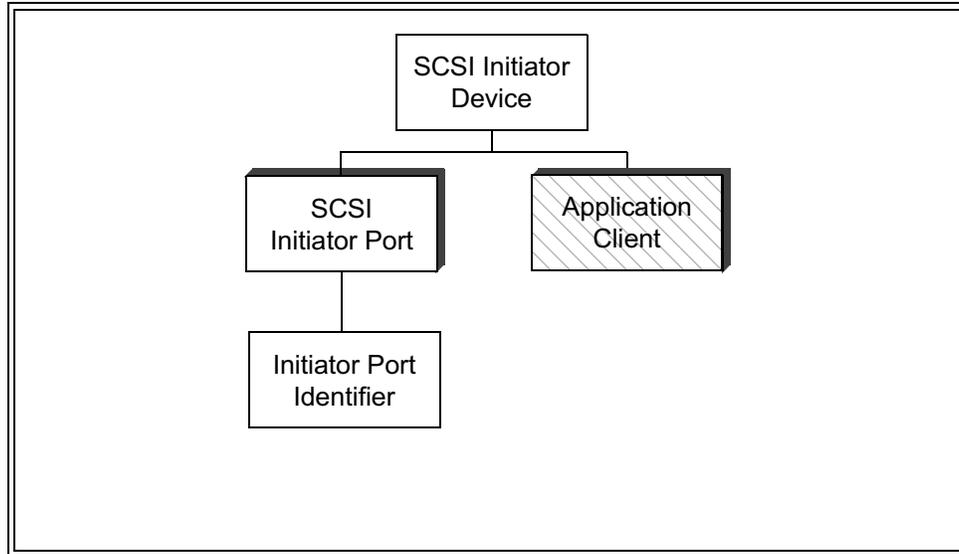


Figure 6 — Initiator model

An initiator port identifier is a value that is the SCSI port identifier for an initiator port.

An application client is the source of commands and task management functions. This model assumes a SCSI initiator device contains one application client for each pending command or task management function.

SAM-2 section 4.7.2 SCSI target device

A SCSI target device (see figure 7) contains:

- a) one more SCSI target ports each containing a task router and SCSI target port identifier; and
- b) one or more logical units.

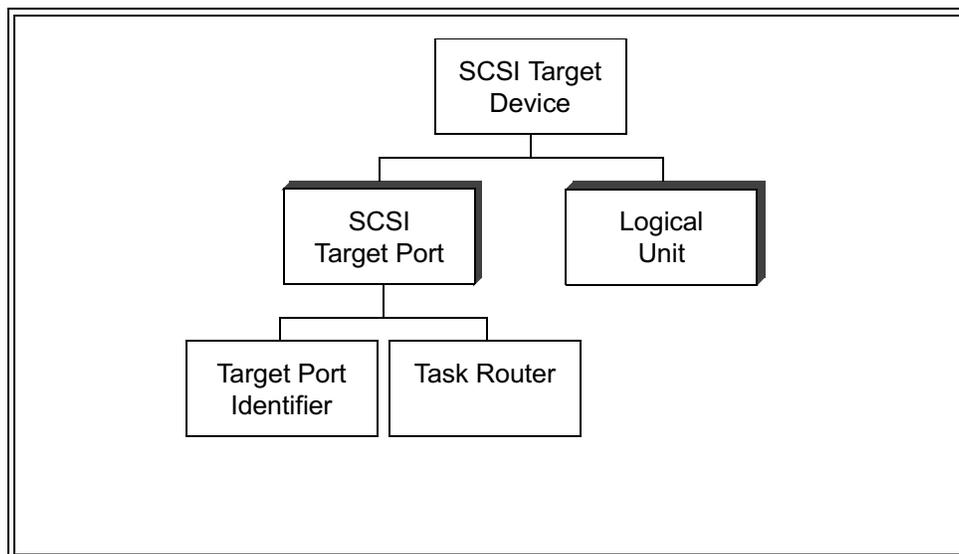


Figure 7 — Target model

A SCSI target port identifier is a value that is a SCSI port identifier for a SCSI target port.

A task router routes commands and task management functions between the service delivery subsystem and the appropriate logical unit's task manager.

A logical unit is the object to which SCSI commands are sent. One of the logical units within the SCSI target device shall be accessed using the logical unit number zero. See xxx for a description of the logical unit.

New SAM-2 section 4.7.3 SCSI target/initiator device

A SCSI target/initiator device (see figure 7) contains:

- a) one more SCSI target/initiator ports each containing a task router, target port identifier, and an initiator port identifier;
- b) one or more logical units; and
- c) zero or more application clients.

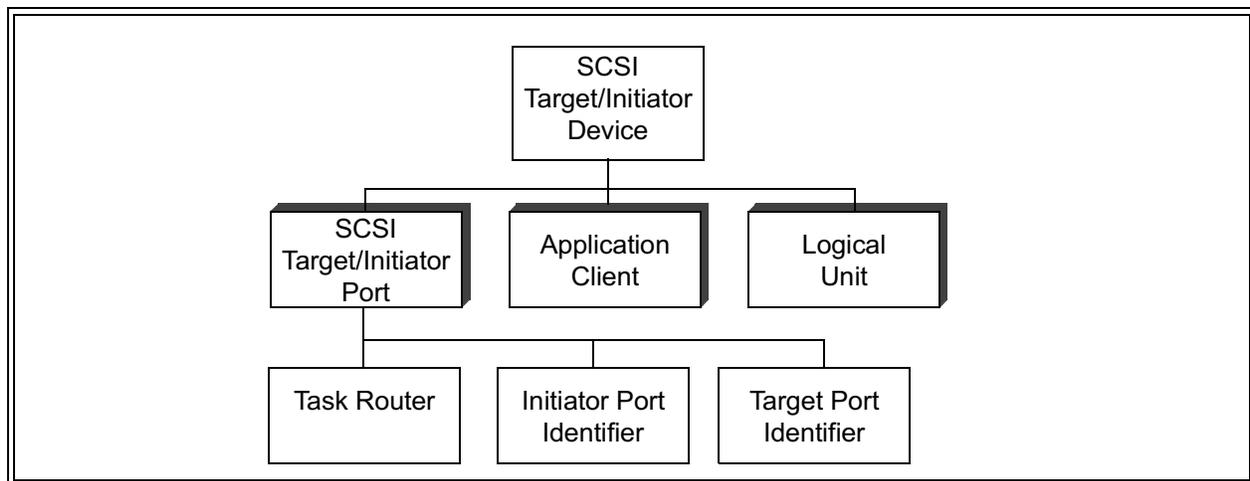


Figure 8 — Target/Initiator model

The target port identifier and the initiator port identifier are values containing a SCSI identifier for a SCSI target/initiator port. The target port identifier and the initiator port identifier may or may not be identical.

When the SCSI target/initiator device is operating as a SCSI target device a task router routes the commands and task management functions between the service delivery subsystem and the appropriate logical unit (see xxx). A logical unit is the object to which SCSI commands are sent. One of the logical units within the SCSI target/initiator device shall be accessed using the logical unit number zero.

When the SCSI target/initiator device is operating as a SCSI initiator device an application client is the source of commands and task management functions. This model assumes there is one application client for each pending command or task management function.

SAM-2 section 4.7.4 SCSI identifier

The SCSI identifier is equivalent to SCSI port identifier and is the object name used to represent either an initiator port identifier for an initiator, or a target port identifier for a target. A SCSI identifier is used when either a SCSI initiator port or SCSI target port might be applicable or when other context in the description identifies the SCSI initiator port or SCSI target port usage.

New SAM-2 section 4.7.5 Task router

The task router routes tasks and task management functions to the selected logical unit. Any task that is sent to a logical unit that is not known to the task router shall be routed to a default logical unit (e.g., LUN 0). Any task management function that is not sent to a specific logical unit shall be broadcast to all logical units known to the task router.

SAM-2 section 4.8 Logical units

A logical unit (see figure 9) contains:

- a) A logical unit number;
- b) a device server;
- c) a task manager; and
- d) one or more task sets each may contain zero or more untagged tasks or a combination of zero or more tagged tasks and zero or more untagged tasks.

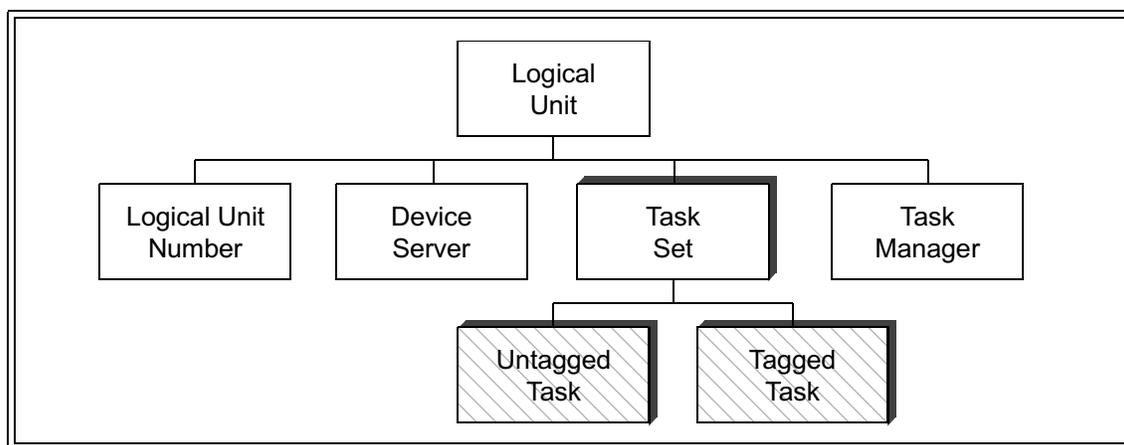


Figure 9 — Logical Unit model

A logical unit number is a value containing up to 64 bits that identifies the logical unit within a SCSI target device. If a SCSI target device contains 256 or fewer logical units none of which are dependent logical units (see xxx), then its logical unit numbers shall have the format shown in table 1, which is a single level subset of the format described in xxx.

Table 1 — Single Level LUN structure

Bit Byte	7	6	5	4	3	2	1	0
0	ADDRESS METHOD (00b)		BUS IDENTIFIER (00h)					
1	SINGLE LEVEL LUN (00h to FFh, inclusive)							
2	(MSB)		Null second level LUN (0000h)				(LSB)	
3	(MSB)		Null third level LUN (0000h)				(LSB)	
4	(MSB)		Null forth level LUN (0000h)				(LSB)	
5	(MSB)		Null forth level LUN (0000h)				(LSB)	
6	(MSB)		Null forth level LUN (0000h)				(LSB)	
7	(MSB)		Null forth level LUN (0000h)				(LSB)	

In the single level subset format, all LUN structure fields shall be zero except the SINGLE LEVEL LUN field (see table 1). The value in the single level LUN field shall be between 0 and 255. The 00b in the ADDRESS METHOD field and the 00h in the BUS IDENTIFIER field indicate a logical unit at the current level is to be accessed (see 4.12.2). When the single level subset format is used, the HiSUP bit shall be set in the standard inquiry data returned by logical unit 0 (see SPC-2).

If any logical unit within the scope of a SCSI target device includes dependent logical units in its composition, all logical unit numbers within the scope of the SCSI target device shall have the format described in xxx.

A device server is the object that processes the operations requested by the received commands.

The task manager controls the sequencing of one or more tasks within a logical unit. The task manager also carries out the task management functions specified in clause 6. There is one task manager per logical unit.

The order in which task management requests are processed is not specified by this standard. This standard does not require in-order delivery of such requests, as defined in xxx, or processing by the task manager in the order received. To guarantee the processing order of task management requests referencing a specific logical unit, an initiator should, not have more than one such request pending to that logical unit.

A task set is composed of zero or more untagged tasks or a combination of zero or more tagged tasks and zero or more untagged tasks. See 4.9 for additional restrictions on the untagged tasks and tagged tasks in a Task Set.

For convenience, task (see 4.9) refers to either a tagged task or an untagged task. The interactions among the tasks in a task set are determined by the rules for task set management specified in clause 7 and the auto contingent allegiance and contingent allegiance rules specified in 5.7.1 and SCSI-2. The number of task sets per logical unit and the boundaries between task sets are governed by the TST field in the control mode page (see SPC-2).

SAM-2 section 4.10 The nexus object

The nexus object is a relationship between a SCSI initiator port, a SCSI target port, optionally a logical unit, and optionally a task.

The nexus object may refer to any one or all of the following relationships:

- a) one SCSI initiator port to one SCSI target port (an I_T nexus);
- b) one SCSI initiator port to one SCSI target port to one logical unit (an I_T_L nexus);
- c) one SCSI initiator port to one SCSI target port to one logical unit to one tagged task (an I_T_L_Q nexus); or
- d) either an I_T_L nexus or an I_T_L_Q nexus (denoted as an I_T_L_x nexus).

Table 2 relates the Nexus object to several identifier objects presented elsewhere in this standard.

Table 2 — Mapping Nexus to SAM-2 Identifiers

Nexus	Identifiers That Form Nexus	Reference
I_T	Initiator Port Identifier Target Port Identifier	
I_T_L	Initiator Port Identifier Target Port Identifier Logical Unit Number	
I_T_L_Q	Initiator Port Identifier Target Port Identifier Logical Unit Number Tag	4.9.1

New SAM-2 section 4.11 SCSI ports

New SAM-2 section 4.11.1 SCSI port configurations

A SCSI device may contain only SCSI target ports, only SCSI initiator ports, only SCSI target/initiator ports or any combination of ports. Some of the port configurations possible for a SCSI device are shown in figure 10.

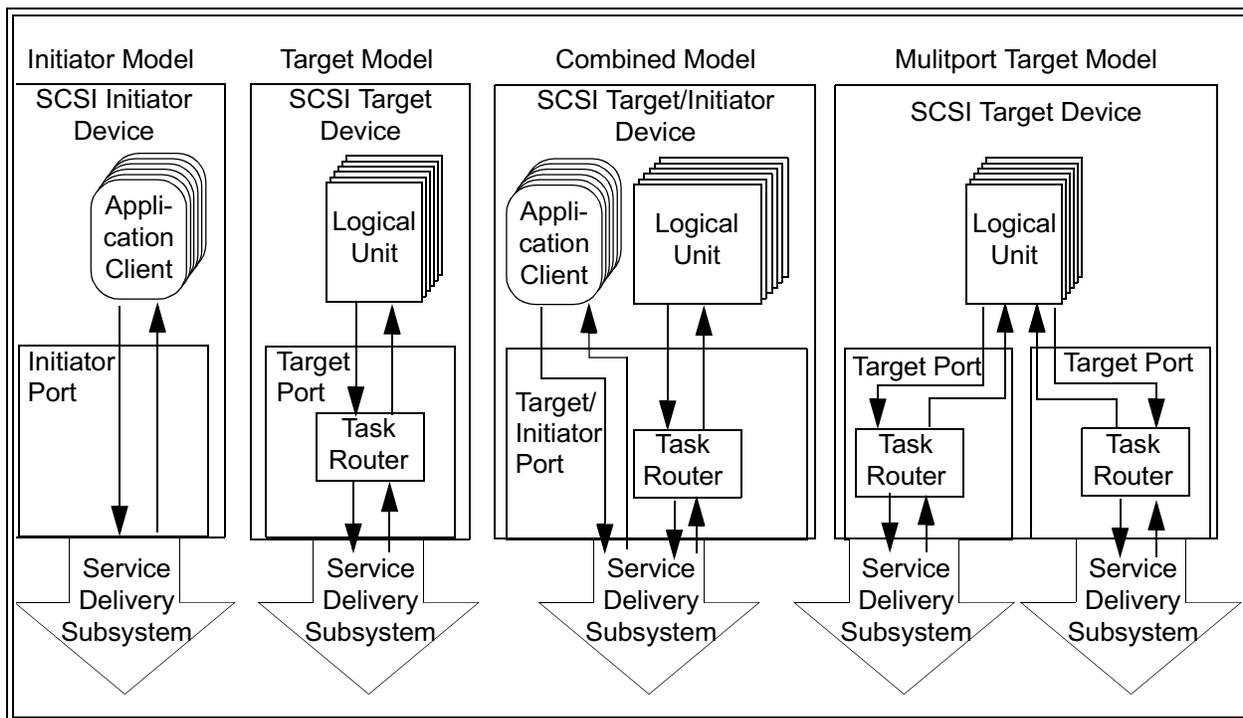


Figure 10 — SCSI device functional models

A target/initiator SCSI device is referred to by the role it's port takes when it participates in an I/O operation. When a SCSI target/initiator device receives SCSI commands or task management functions, the SCSI target/initiator device takes on the characteristics of and is referred to as a SCSI target device. When a SCSI target/initiator device issues SCSI commands or task management functions, the SCSI target/initiator device takes on the characteristics of and is referred to as a SCSI initiator device.

SAM-2 section 4.11.2 SCSI devices with multiple ports

The model for a SCSI device with multiple ports is a single SCSI target device, SCSI initiator device, or SCSI target/initiator device (see xxx) with multiple ports. Similarly, a single SCSI target port or SCSI initiator port may respond to multiple SCSI identifiers. The model for such a SCSI device also is one of multiple SCSI target ports or SCSI initiator ports, one for each SCSI identifier.

The SCSI identifiers representing the ports shall meet the requirements for initiator port identifiers (see xxx) or target port identifiers (see xxx) or both. SCSI target/initiator devices with multiple ports implement both target and initiator models and combine the SCSI target/initiator port structures in vendor-specific ways that meet product requirements while maintaining the multi-port model for the target and initiator functions performed by the product. How a multiple port SCSI device is viewed by counterpart SCSI devices in the SCSI domain also depends on whether a SCSI initiator port is examining a SCSI target port or SCSI target/initiator port, or a SCSI target port is servicing a SCSI initiator port or SCSI target/initiator port. The structures and views of SCSI devices are asymmetric for SCSI target ports, and SCSI initiator ports.

New SAM-2 section 4.11.2 Multiple port target SCSI device structure

Figure 11 shows the structure of a SCSI target device with multiple SCSI target ports. Each SCSI target port consists of a task router that is shared by a collection of logical units. Each logical unit contains a single task manager and a device server.

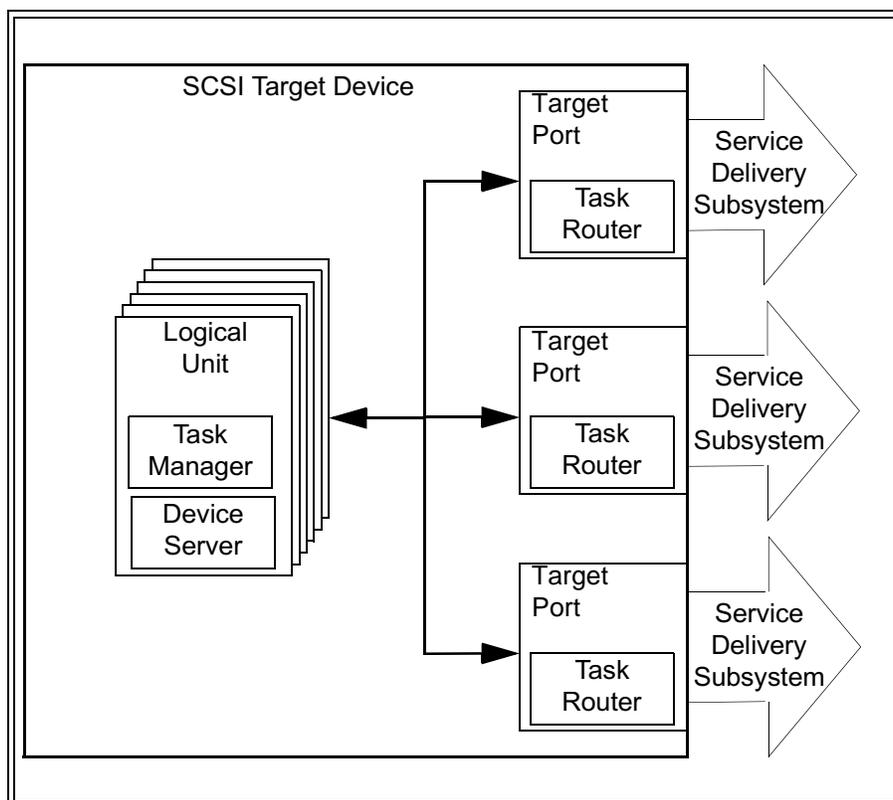


Figure 11 — Multiple port target SCSI device structure model

Two-way communications shall be possible between all logical units and all SCSI target ports, however, communications between any logical unit and any SCSI target port may occasionally be inactive. Two-way communications shall be available between each task manager and all task routers. Each SCSI target port shall accept commands sent to LUN 0 and the task router shall route them to a device server for

processing. The REPORT LUNS commands (see SPC-2) shall be accepted by logical unit 0 from any SCSI target port and shall return the logical unit inventory available via that SCSI target port. The availability of the same logical unit through multiple SCSI target ports is discovered by matching SCSI port identifier values in the INQUIRY command vital product data page (see SPC-2).

New SAM-2 section 4.11.3 Multiple port initiator SCSI device structure

Figure 11 shows the structure of a SCSI initiator device with multiple SCSI initiator ports. Each SCSI initiator port is shared by a collection of application clients.

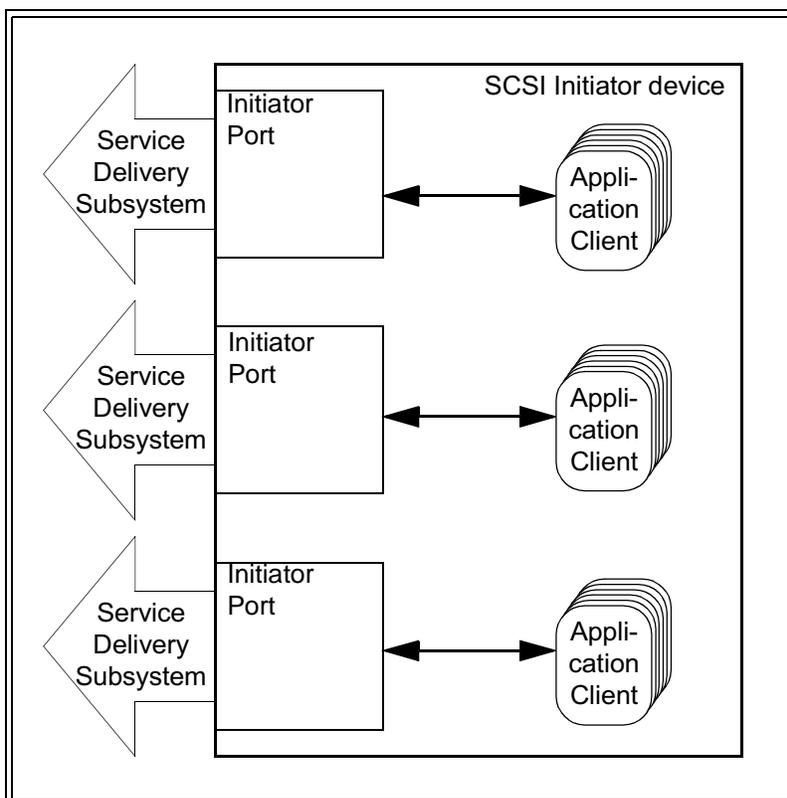


Figure 12 — Multiple port SCSI initiator device structure model

Two-way communications shall be possible between an application client and its associated SCSI initiator port. Mechanisms by which a SCSI target device would have the ability to discover that it is communicating with multiple ports on a single SCSI initiator device are beyond the scope of any standards in the SCSI family of standards.

New SAM-2 section 4.11.4 Multiple port target/initiator SCSI device structure

Figure 11 shows the structure of a SCSI target/initiator device with multiple SCSI target/initiator ports. Each SCSI target/initiator port consists of a task router and is shared by a collection of logical units and application clients. Each logical unit contains a single task manager and a device server.

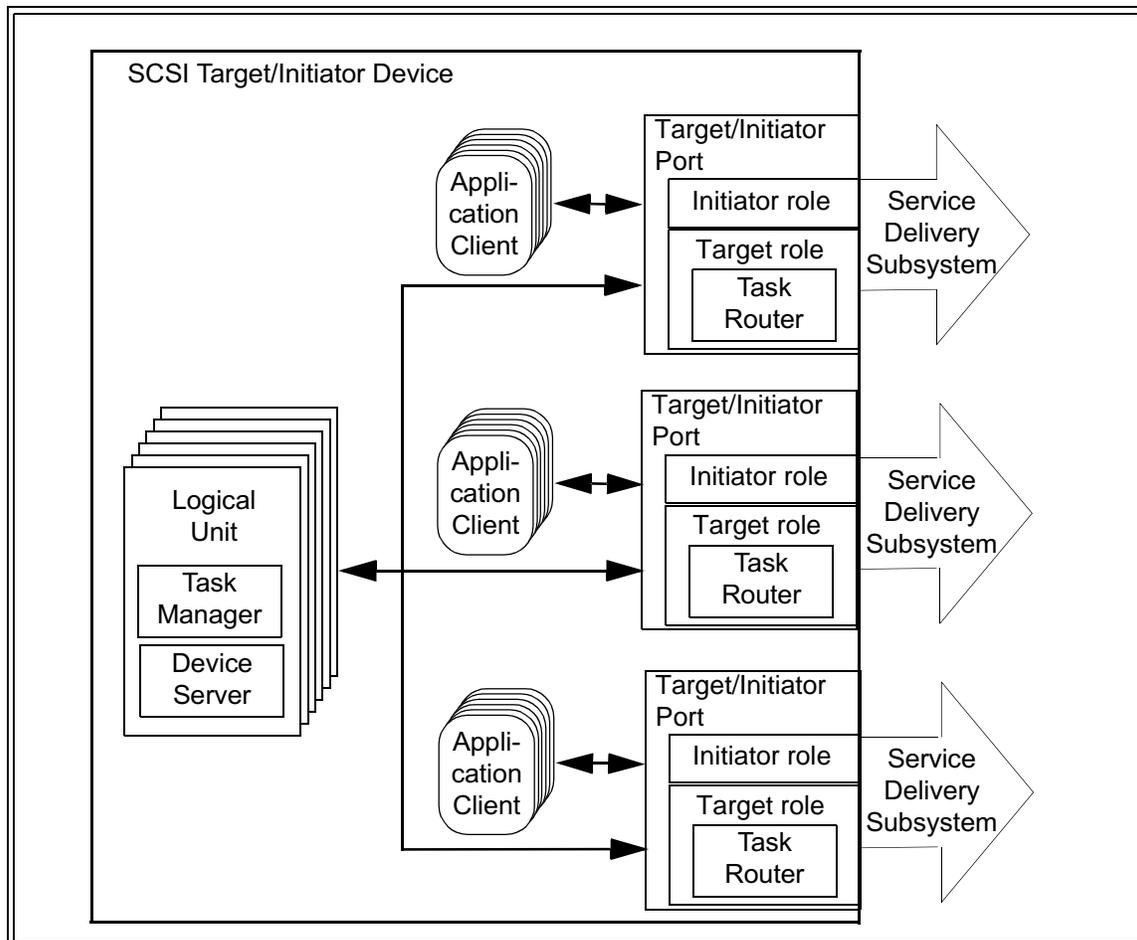


Figure 13 — Multiple port target/initiator SCSI device structure model

Two-way communications shall be possible between all logical units and all SCSI target/initiator ports, however, communications between any logical unit and any SCSI target/initiator port may occasionally be inactive. Two-way communications shall be possible between an application client and its associated SCSI target/initiator port. Each SCSI target/initiator port shall accept commands sent to LUN 0 and the task router shall route them to a device server for processing. The REPORT LUNS commands (see SPC-2) shall be accepted by logical unit 0 from any SCSI target/initiator port and shall return the logical unit inventory available via that SCSI target/initiator port. The availability of the same logical unit through multiple SCSI target/initiator ports is discovered by matching SCSI port identifier values in the INQUIRY command vital product data page (see SPC-2).

Mechanisms by which a SCSI target device would have the ability to discover that it is communicating with multiple ports on a SCSI target/initiator device are beyond the scope of any standards in the SCSI family of standards.

New SAM-2 section 4.11.5 SCSI initiator device view of a multiple port SCSI target device

A SCSI target device may be connected to multiple domains such that a SCSI initiator port is only allowed to only communicate with logical units using a single SCSI target port. This would restrict application clients from determining if a SCSI target device has multiple SCSI ports.

However, SCSI target devices with multiple SCSI ports may be configured where application clients have the ability to discover that one or more logical units are accessible via multiple SCSI target ports. Figure 14 and figure 15 show two examples of such configurations.

Figure 14 shows a SCSI target device with multiple SCSI target ports participating in a single SCSI domain with two SCSI initiator devices. There are three SCSI devices, one of which has two SCSI target ports, one with one SCSI initiator port, and one with one SCSI initiator port. There are two SCSI target port identifiers and two initiator port identifiers in this SCSI domain. Using the INQUIRY command vital product data page (see SPC-2), the application clients in each of the SCSI initiator devices have the ability to discover the logical units in the SCSI target devices are accessible via multiple SCSI target port identifiers (i.e., SCSI target ports) and map the configuration of the SCSI target devices.

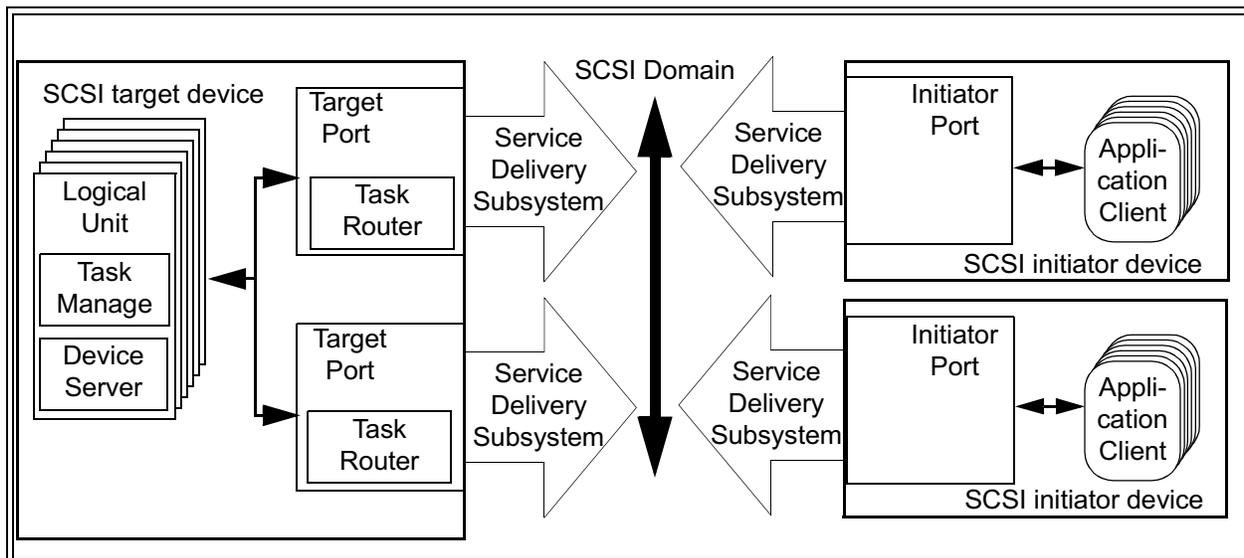


Figure 14 — SCSI target device configured in a single SCSI Domain

Figure 15 shows a SCSI target device with multiple SCSI target ports participating in two SCSI domains and a SCSI initiator device with multiple SCSI initiator ports participating in the same two SCSI domains. There is one SCSI target device with two SCSI target ports and one SCSI initiator device with two SCSI initiator ports. There is one SCSI target port identifier and one initiator port identifier in each of the two SCSI domains. Using the INQUIRY command vital product data page (see SPC-2), the application clients in the SCSI initiator device have the ability to discover that logical units in the SCSI target device are accessible via multiple ports and map the configuration. However, the methods available to application clients to distinguish between the configuration shown in figure 15 and the configuration shown in figure 14 are beyond the scope of the SCSI family of standards.

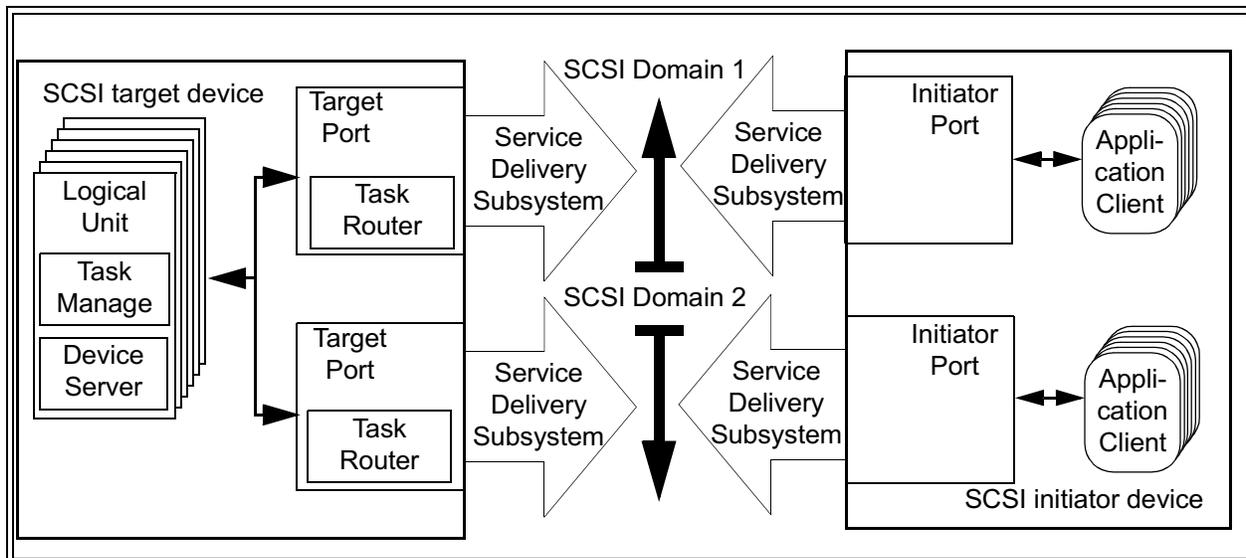


Figure 15 — SCSI target device configured in multiple SCSI Domains

Figure 16 shows the same configuration as figure 15 except that the two SCSI domains have been replaced by a single SCSI domain.

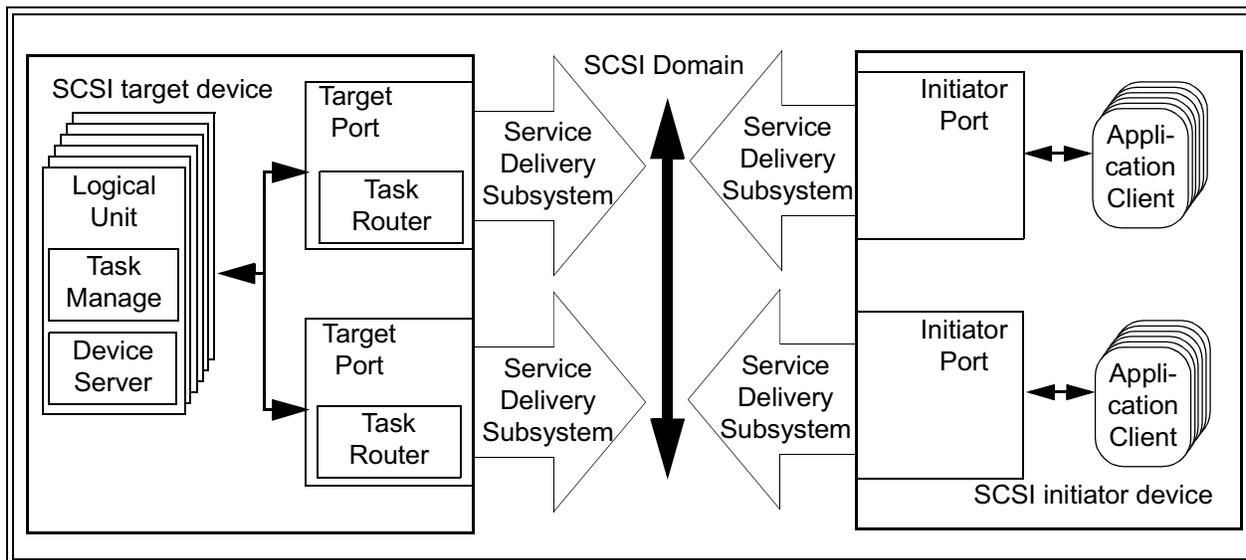


Figure 16 — SCSI target device and SCSI initiator device configured in a single SCSI Domain

This model for application client determination of multiple SCSI target port configurations relies on information that is available only to the application clients via SCSI commands. The SCSI initiator ports in the SCSI initiator devices (figure 14) or SCSI initiator device (figure 15 and figure 16) are unable to distinguish multiple SCSI target ports from individual SCSI target ports in two separate SCSI target devices.

new SAM-2 section 4.11.6 SCSI target device view of a multiple port SCSI initiator device

A SCSI target device does not have the ability to detect the presence of a SCSI initiator device with multiple SCSI initiator ports. Therefore, a SCSI target device handles a SCSI initiator device with multiple SCSI initiator ports exactly as it would handle multiple separate SCSI initiator devices each with a single SCSI initiator port. For example, a SCSI target device handles the configurations shown in figure 15 and figure 16 in exactly the same way it handles the configuration show in figure 14.

NOTE 1 The implications of this view of a SCSI initiator device are more far reaching than are immediately apparent. For example, if a SCSI initiator device with multiple SCSI initiator ports makes an exclusive access reservation via one SCSI initiator port, then access is denied to the other SCSI initiator port(s) on that same SCSI initiator device.

new SAM-2 section 4.11.7 Multiple port considerations for task management functions

Although, the task manager for each logical unit handles all the SCSI target ports and SCSI target/initiator ports in a SCSI target device, the task manager in each logical unit shall observe the requirements described in the following clauses in addition to the requirements placed on a task set manager by the SCSI architecture structural model.

SAM-2 section 4.12 Hierarchies of dependent logical units

Optionally, the model for a logical unit (see xxx) may be enhanced to include one or more unique logical units embedded within another logical unit. In such cases, the model shown in xxx is enhanced to become the model shown in figure 9 and the embedded logical units are called dependent logical units (see 3.1.22).

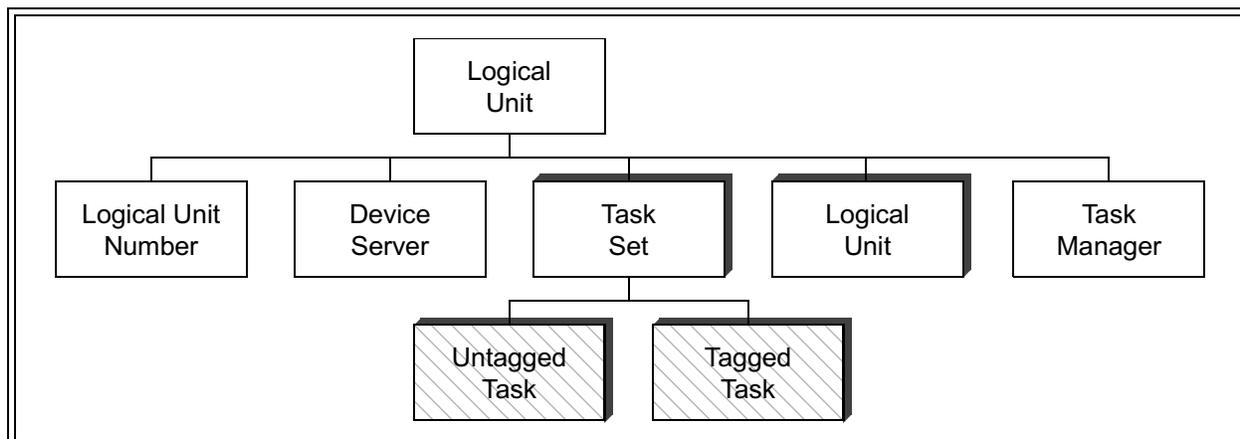


Figure 17 — Dependent Logical Unit model

When the dependent logical unit model enhancement is utilized, the hierarchical logical unit structure defined here shall be used. If any logical unit within the scope of a SCSI target device includes dependent logical units, all logical unit numbers within the scope of the SCSI target device shall have the format described in this subclause. A device server that implements the hierarchical structure for dependent logical units described here shall set the HiSUP bit in the STANDARD INQUIRY DATA returned by logical unit 0 (see SPC-2). Clause xxx defines cases when SCSI target devices that do not implement dependent logical units are required to implement a subset of the logical unit structure described in this subclause.