Date: Jan 09, 2001

To: T10 Committee (SCSI)

From: George Penokie (Tivoli Systems)

Subject: Defining Tragets/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.

What follows are suggested changes to SAM that would define targets and initiators as ports and allow multiples on a single SCSI device.

Multiple port target SCSI Device

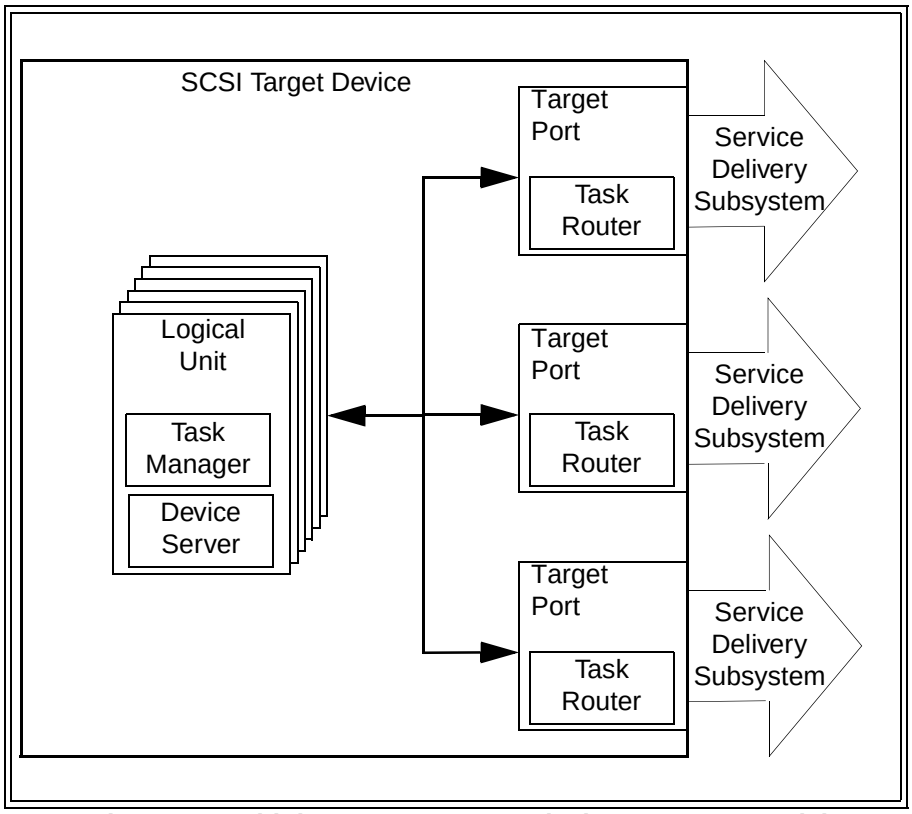

Figure 1 - Multiple port target SCSI device structure model

Multiple port initiator SCSI Device

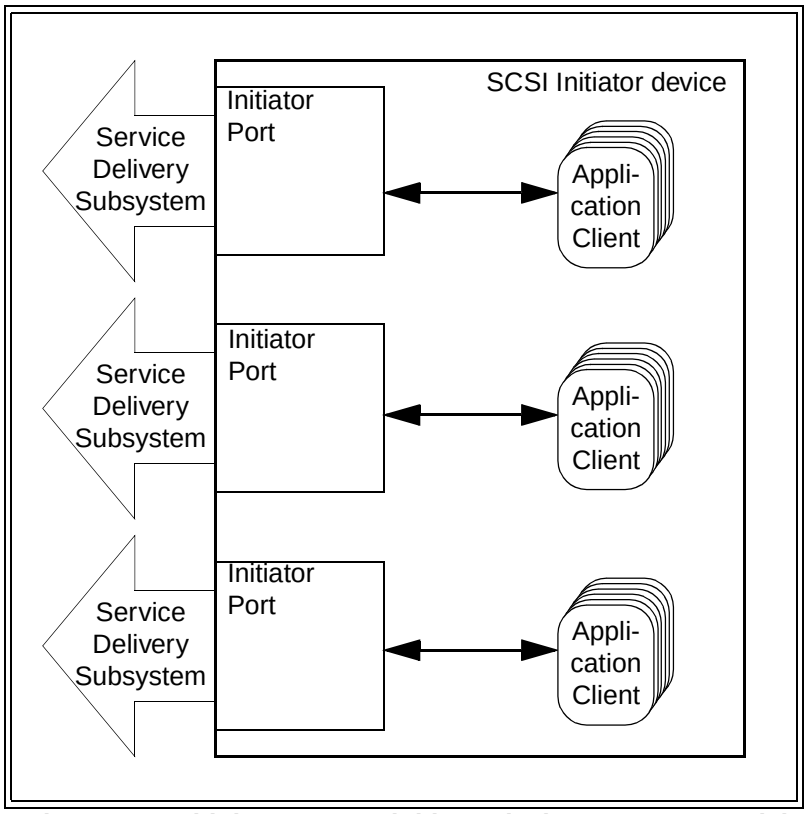

Figure 2 - Multiple port SCSI initiator device structure model

Multiple port target/initiator SCSI Device

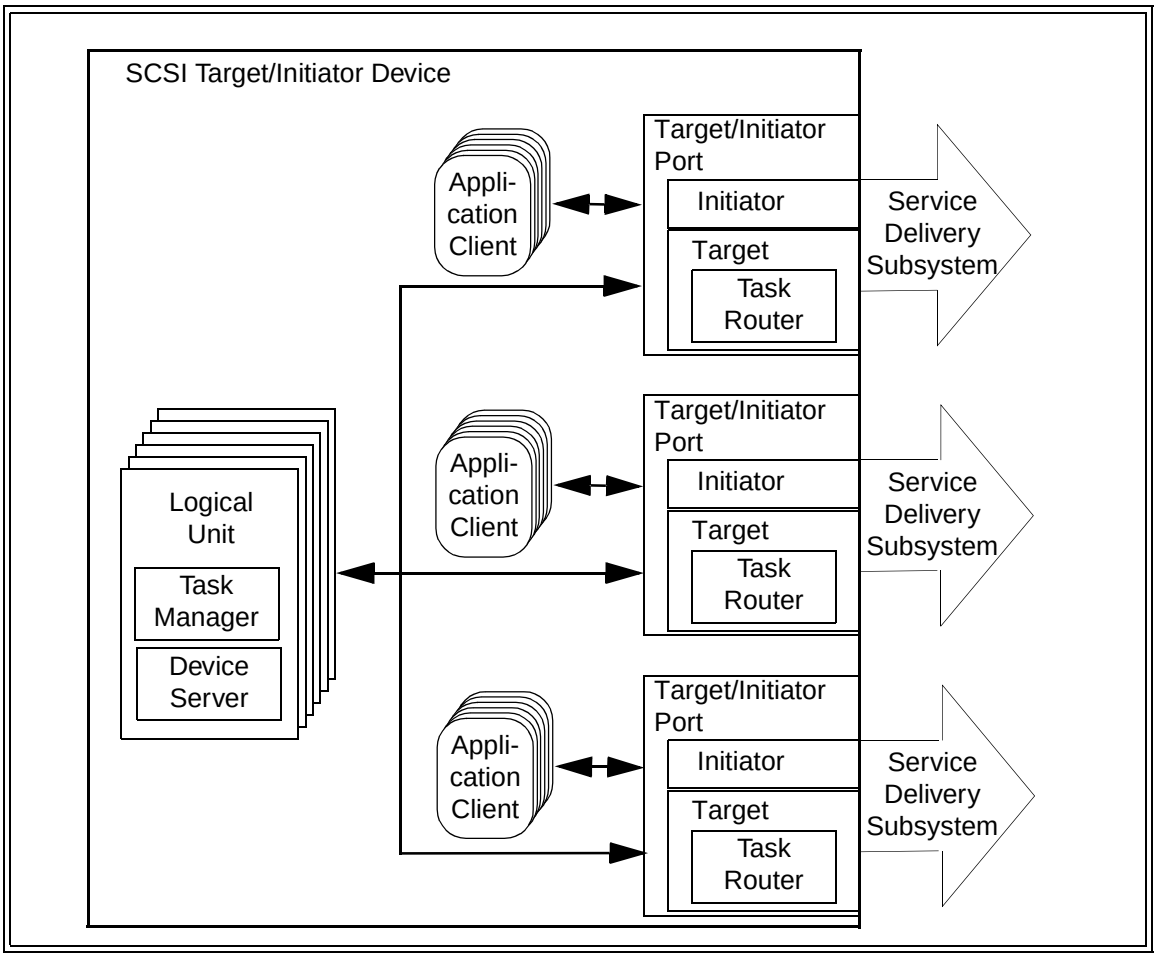

Figure 3 - Multiple port target/initiator SCSI device structure model

Multiple port SCSI Device model

The following show the changes in section 4 of SAM-2 that would be required to change to support a port = target/initiator model.

SAM-2 Section 4.3 The SCSI client-server model

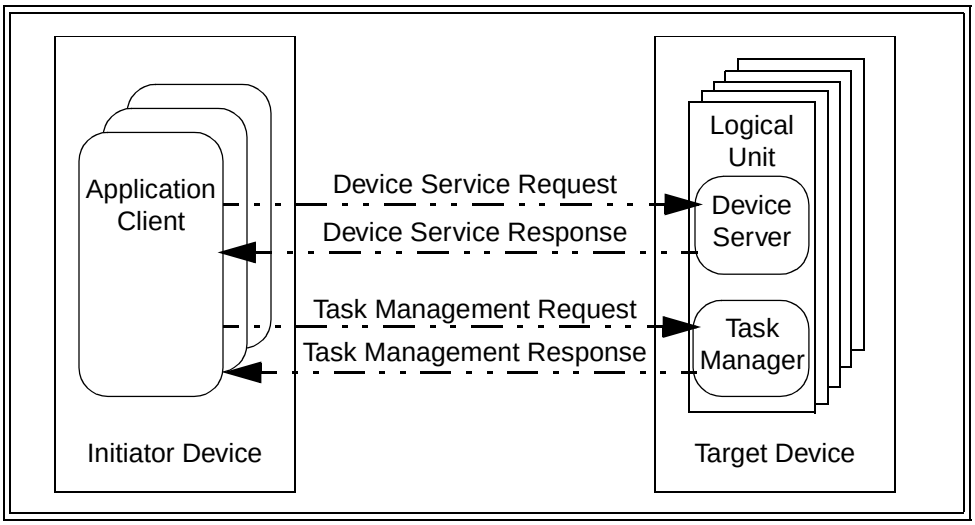

Figure 4 — SCSI client-server model

SAM-2 Section 4.4 The SCSI structural model

The SCSI structural model represents a view of the elements comprising an SCSI I/O system as seen by the application clients interacting with the system through the service delivery port. In an implementation, this view is similar to that seen by a CAM device driver interacting with the system through the CAM SIM layer. This model is defined as a hierarchy of objects. As shown in figure 5, the fundamental object is the SCSI domain, which represents an I/O system. A domain is made up of SCSI devices and a service delivery subsystem, which transports commands and data. An SCSI device, in turn, may consist of logical units and so forth.

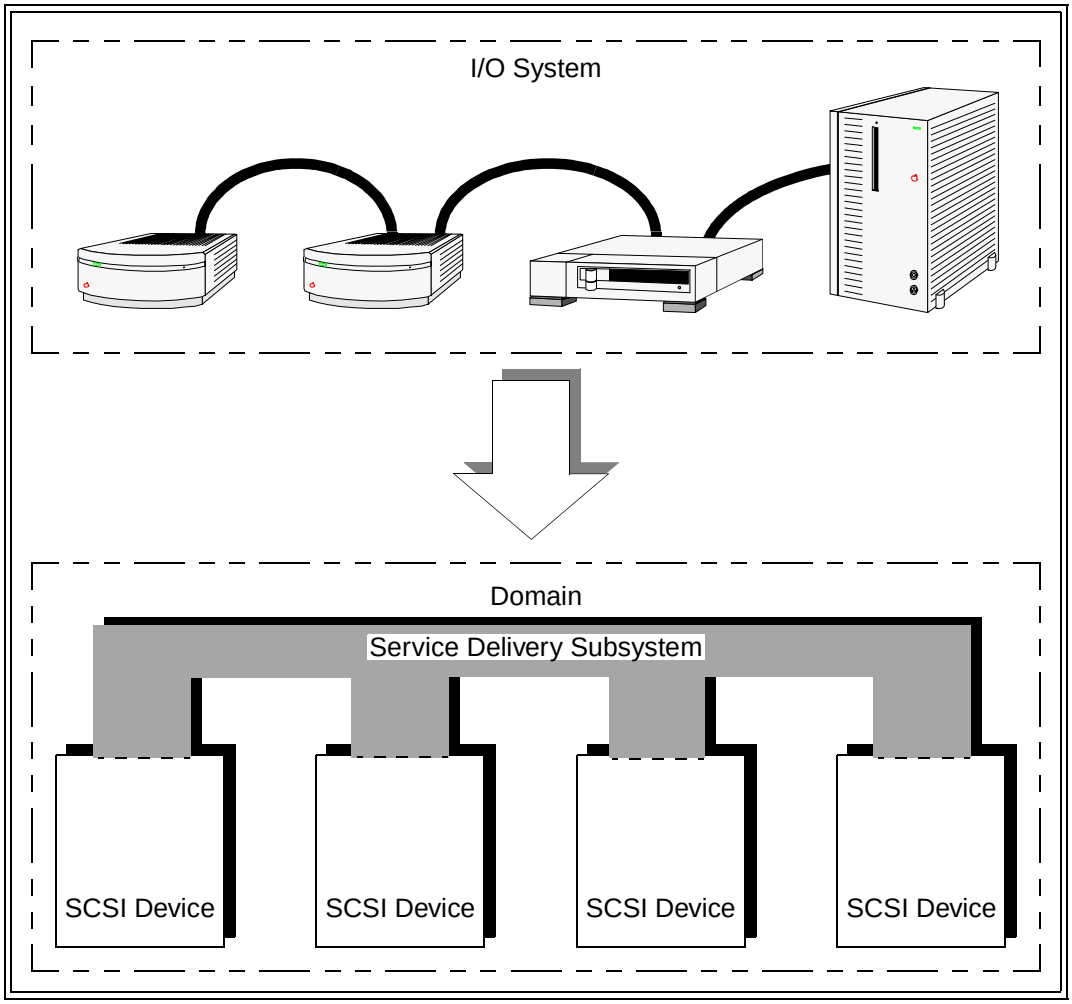

Figure 5 — SCSI I/O system and domain model

The SCSI structural model is organized as follows:

a) a basic structural model (see 4.4 through 4.9);
b) a TBD enhancement for multiple port devices (see); and
c) an enhancement for logical units with dependent logical units (see).

The basic structural model applies to all SCSI devices and domains.

Figure 6 shows the main functional components of the SCSI hierarchy. The following clauses define these components in greater detail.

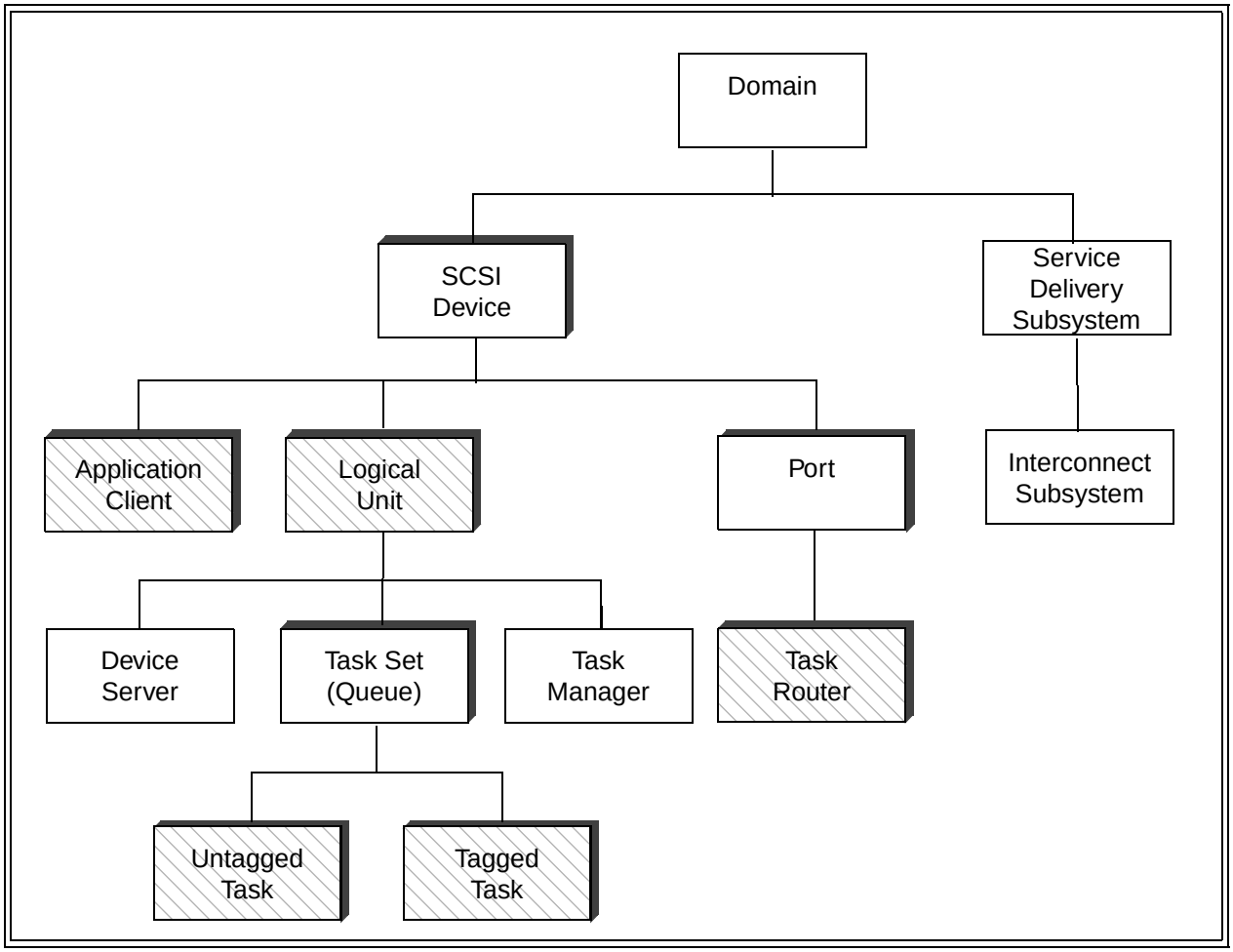

Figure 6 - SCSI hierarchy

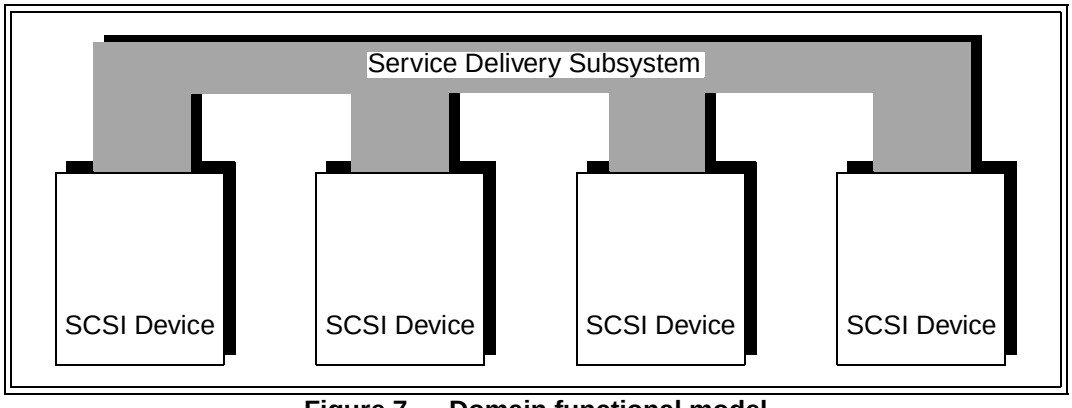

Figure 7 — Domain functional model

SAM-2 Section 4.5 SCSI domain

An SCSI domain is composed of two or more SCSI devices and a Service Delivery Subsystem. Figure 7 shows a functional model of a SCSI domain corresponding to the SCSI I/O system model shown in figure 5. Figure 8 shows the hierarchy of SCSI domain objects.

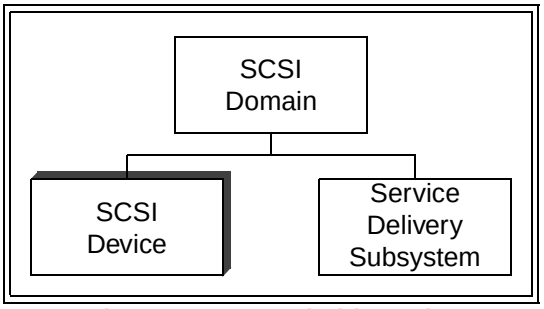

Figure 8 — Domain hierarchy

A SCSI device is an object that originates or services SCSI commands. As described in , when a SCSI device originates a command that command goes through an initiator port or a target/initiator port; a SCSI device containing logical units that service commands receives commands through a target port or a target/initiator port. The service delivery subsystem connects all the ports in the SCSI domain, providing a subsystem through which application clients and device servers communicate (see 4.6). 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 is composed of an interconnect subsystem (see figure 9).

.

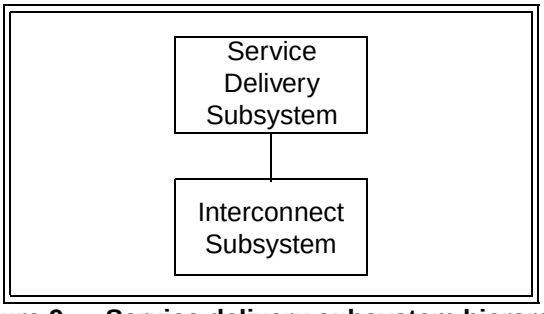

Figure 9 — Service delivery subsystem hierarchy

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 execute 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 port for transmission; it is "in transit" until delivered and "received" when it has been forwarded to the receiver via the destination device's 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 devices, 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 often important whenever 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 specification makes no assumption about, or places any requirement on the ordering of requests or responses between one sending SCSI device and several receiving SCSI devices.

SAM-2 Section 4.7 SCSI Device models

A SCSI device's (see figure 10) consists of a SCSI target device, a SCSI initiator device, or a SCSI target/initiator device.

.

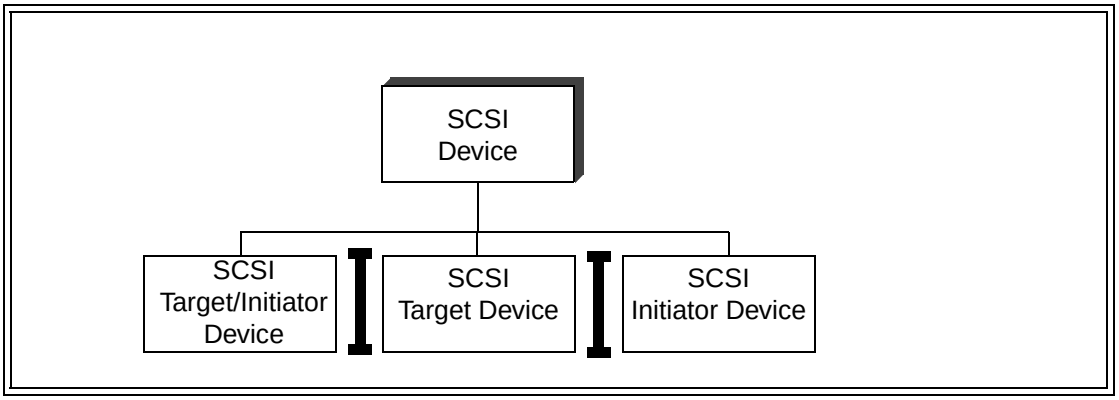

Figure 10 - SCSI Device hierarchy diagram

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

SAM-2 section 4.7.1 SCSI initiator

A SCSI initiator device (see figure 11) contains one or more initiator ports each having an initiator identifier and zero or more application clients.

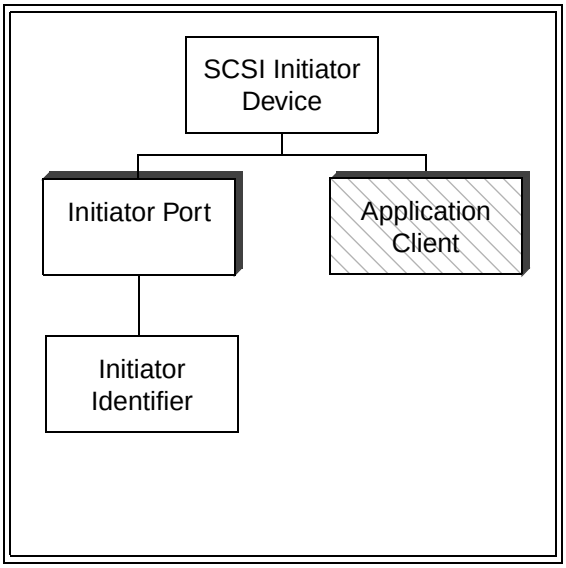

Figure 11 - Initiator hierarchy diagram

An initiator identifier is a value containing up to 64 bits that is the SCSI device identifier for an initiator port. An application client is the source of commands and task management functions. A SCSI initiator device contains is one application client for each pending command or task management function.

SAM-2 section 4.7.2 SCSI target

A SCSI target device (see figure 12) contains of one more target ports composed of a task router and target identifier and one or more logical units.

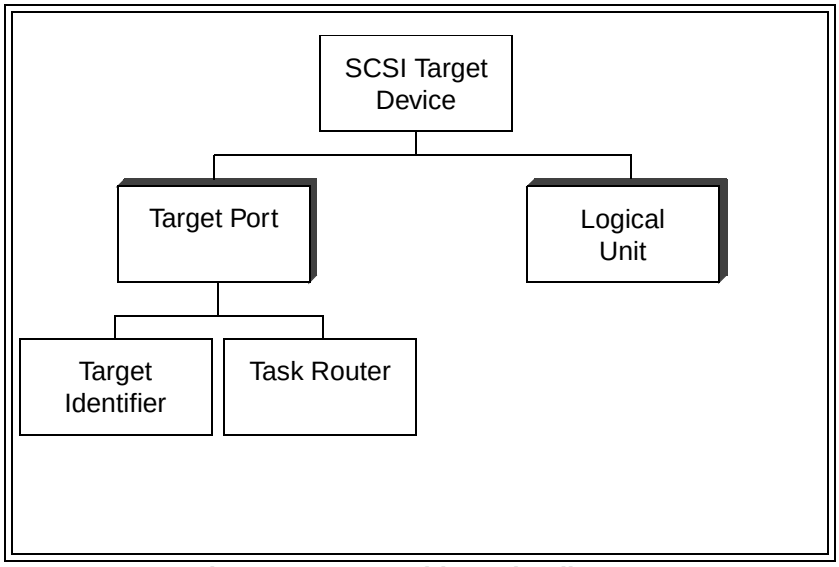

Figure 12 - Target hierarchy diagram

A target identifier is a value containing up to 64 bits that is a SCSI device identifier for a target port. A task router moves the commands and task management functions between the service delivery subsystem and the appropriate logical unit. A logical unit (see) is the object to which SCSI commands are addressed. One of the logical units within the SCSI target device shall be addressed using the logical unit number zero. SCSI target/initiator model.

See xxx for a description of the logical unit.

A SCSI target/initiator device (see figure 12) contains of one or more target/initiator ports composed of a task router and target/initiator identifier and one or more logical units.

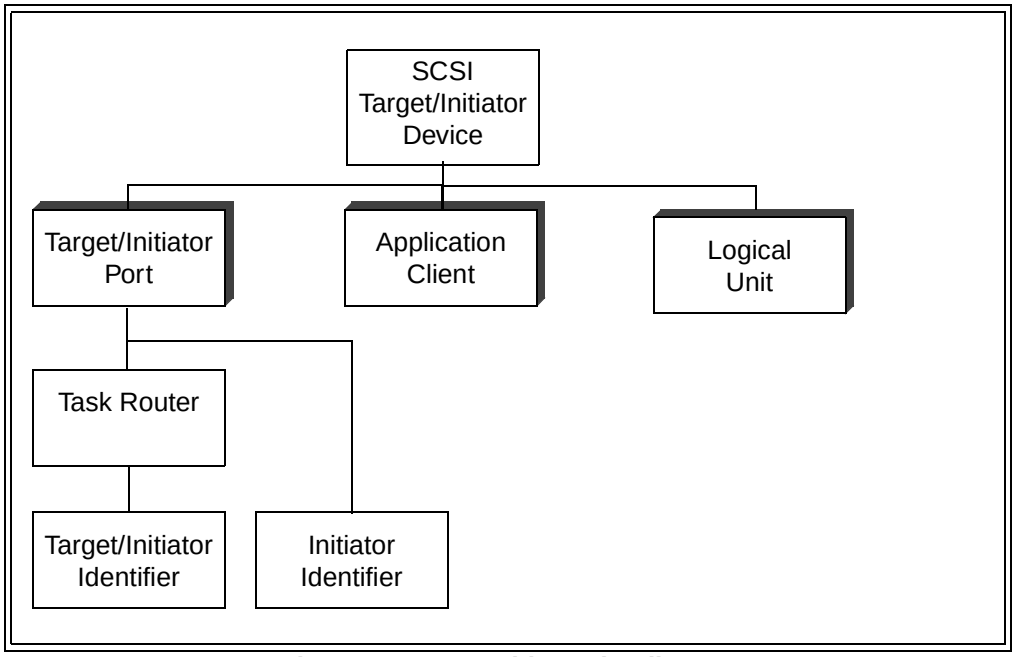

Figure 13 - Target hierarchy diagram

The target/initiator identifier is a value containing up to 64 bits that is a SCSI device identifier for a target/initiator port.

When the SCSI target/initiator device is operating as a SCSI target device a task router moves the commands and task management functions between the service delivery subsystem and the appropriate logical unit. A logical unit (see) is the object to which SCSI commands are addressed. One of the logical units within the SCSI target/initiator device shall be addressed 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. There is one application client for each pending command or task management function.

See xxx for a description of the logical unit.

SAM-2 section 4.9.3 SCSI device identifier

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

New SAM-2 section 4.9.5 Task router

The task router routes received tasks and task management functions to the addressed logical unit. Any task that is addressed to a logical unit that is not know to the task router shall be routed to LUN 0. Any task management function that is not addressed to a logical unit shall be broadcast to all logical units known to the task router.

New SAM-2 section 4.7.6 SCSI device/port models

There are several models for implementing ports in an SCSI device, some are shown in figure 14. A SCSI device may contain only target ports, only initiator ports, only target/initiator ports or any combination there of.

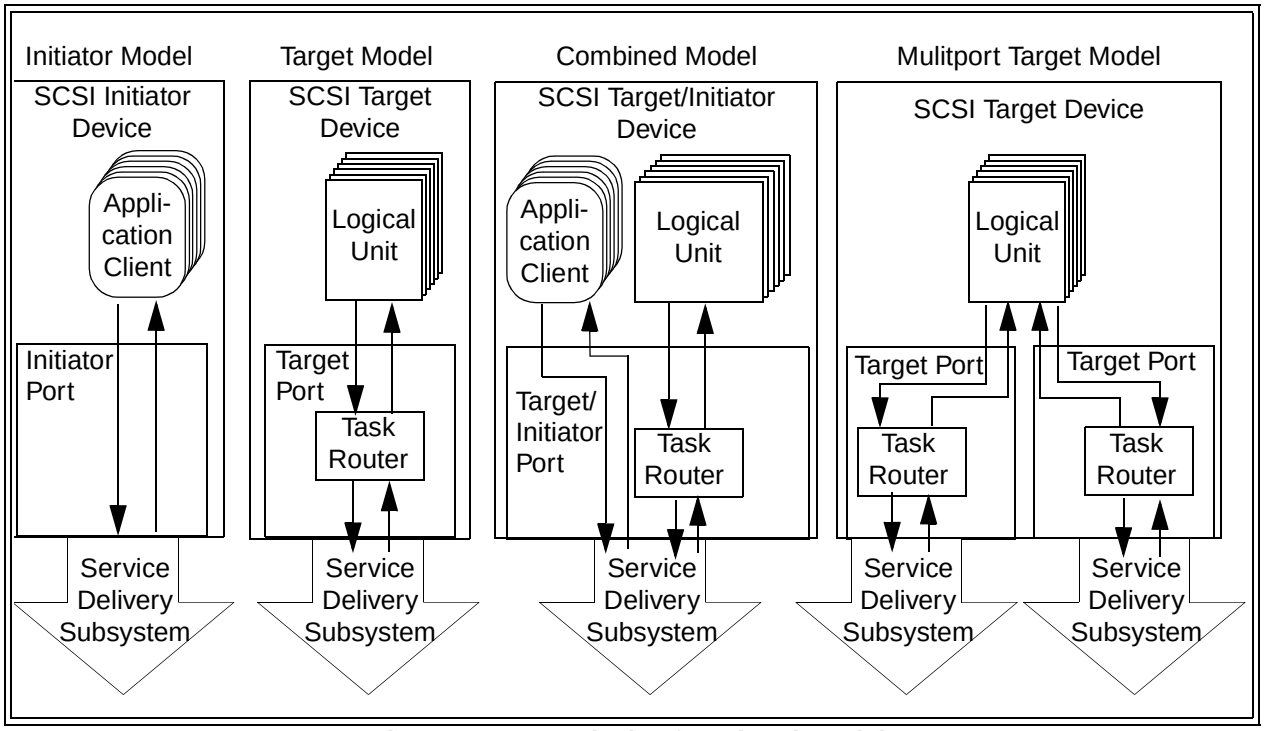

Figure 14 - SCSI device functional models

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

SAM-2 section 4.8 Logical units

A logical unit (see figure 15) is composed of a logical unit number, a device server, a task manager, and one or more task sets.

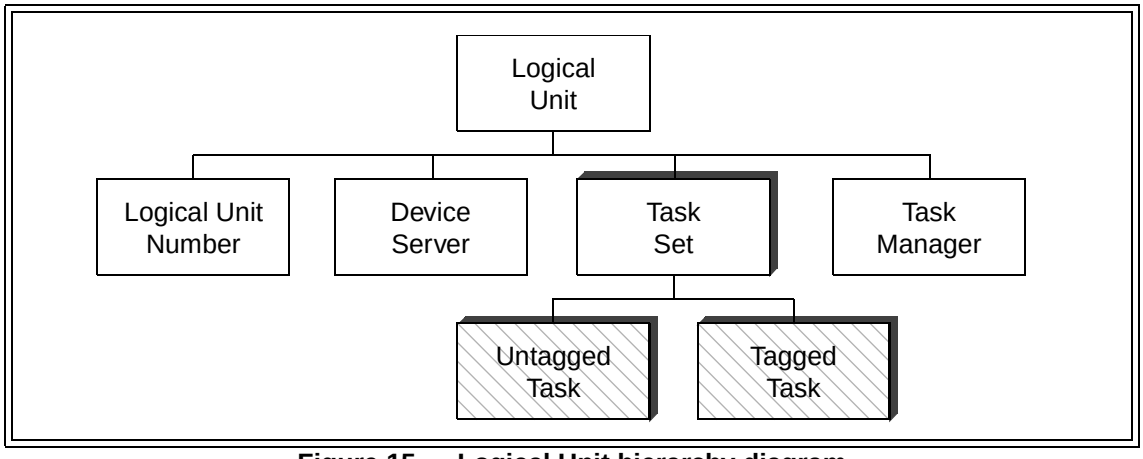

Figure 15 — Logical Unit hierarchy diagram

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), then its logical unit numbers shall have the format shown in table 1, which is a single level subset of the format described in .

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)					
3								(LSB)
4	(MSB)		Null third level LUN (0000h)					
5								(LSB)
6	(MSB)		Null forth level LUN (0000h)					
7								(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 addressing for a logical unit at the current level (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 .

A device server is the object that carries out 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 executed is not specified by this standard. This standard does not require in-order delivery of such requests, as defined in , or execution by the task manager in the order received. To guarantee the execution order of task management requests referencing a specific logical unit, an initiator should, therefore, 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 two SCSI Devices and the initiator and target ports within those SCSI devices.

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

a) one initiator port to one target port (an I_T nexus);
b) one initiator port to one target port to one logical unit (an I_T_L nexus);
c) one initiator port to one 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 Identifier Target Identifier	
I_T_L	Initiator Identifier Target Identifier Logical Unit Number	
I_T_L_Q	Initiator Identifier Target Identifier Logical Unit Number Tag	4.9.1

SAM-2 section 4.11 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 device, SCSI initiator device, or SCSI target/initiator device may respond to multiple SCSI device identifiers. The model for such a SCSI device also is one of multiple SCSI target devices, SCSI initiator devices, or SCSI target/initiator devices, one for each SCSI device identifier.

The multiple SCSI device identifiers representing the ports shall meet the requirements for Initiator Identifiers (see xxx) or Target Identifiers (see xxx) or both. SCSI target/initiator devices with multiple ports implement both target and initiator models and combine the 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 an initiator port is examining a target port or target/initiator port, or a target port is servicing an initiator port or target/initiator port. The structures and views of SCSI devices are asymmetric for target ports, initiator ports, and target/initiator ports.

New SAM-2 section 4.11.1 Multiple port target SCSI device structure

Figure 10 shows the structure of a SCSI target device with multiple target ports. Each target port consists of a task router and is shared by a collection of logical units. Each logical unit contains a single task manager and a device server.

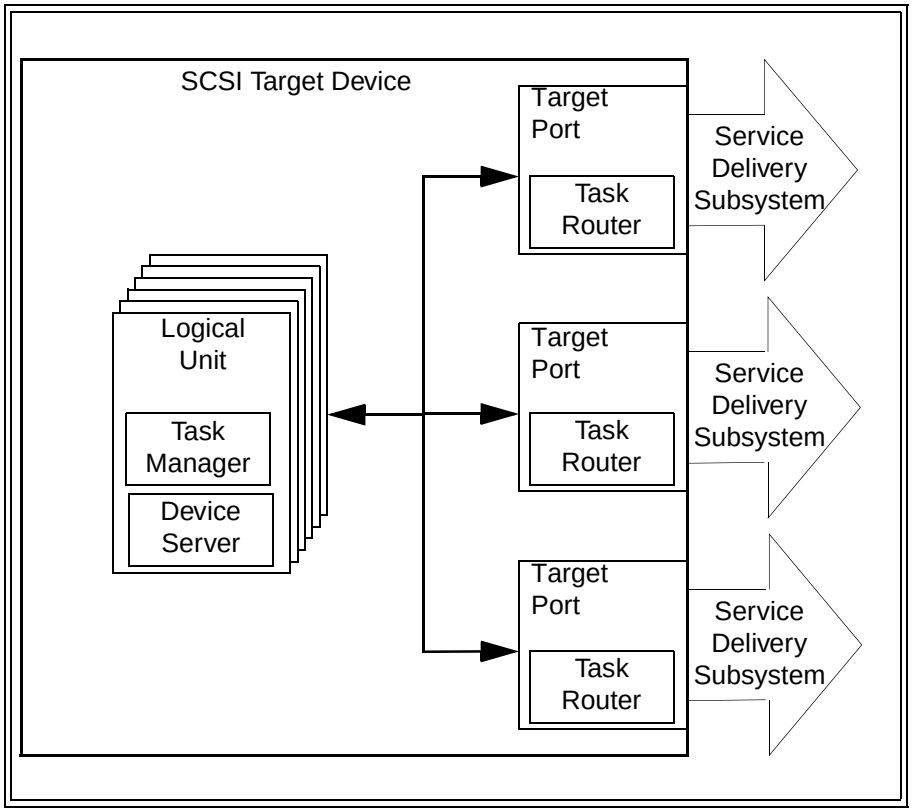

Figure 16 - Multiple port target SCSI device structure model

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

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

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

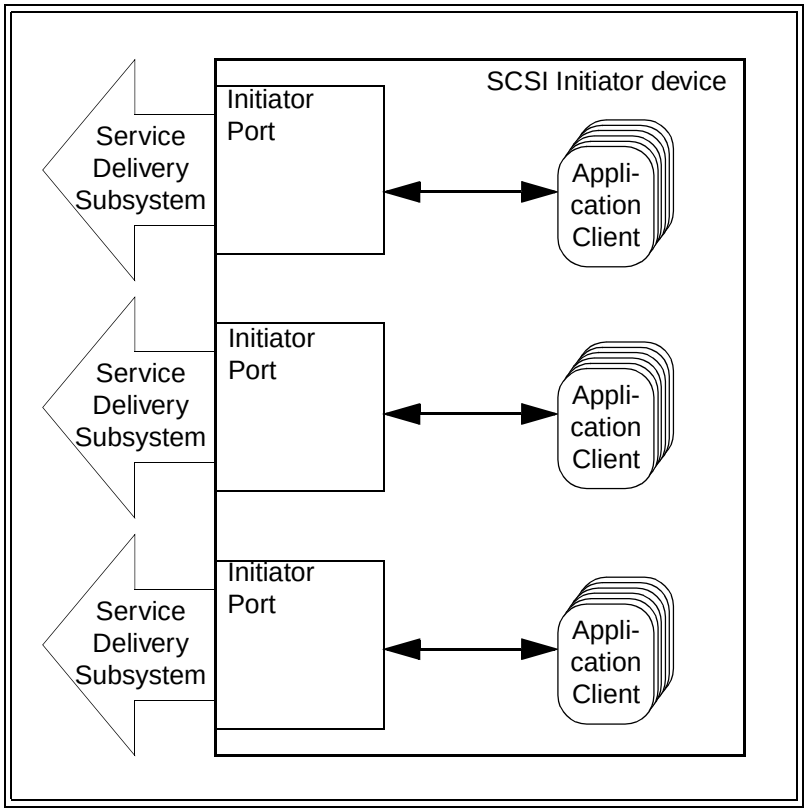

Figure 17 - Multiple port SCSI initiator device structure model

Two-way communications shall be possible between an application client and it's associated initiator port, however, communications may occasionally be inactive. Mechanisms by which a SCSI target device would have the ability to discover that it is communicating with an initiator port are beyond the scope of any standards in the SCSI family of standards.

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

Figure 13 shows the structure of a SCSI target/initiator device with multiple target/initiator ports. Each target/initiator port consists of a task router and is shared by a collection of logical units. Each logical unit contains a single task manager, a device server, and an application client.

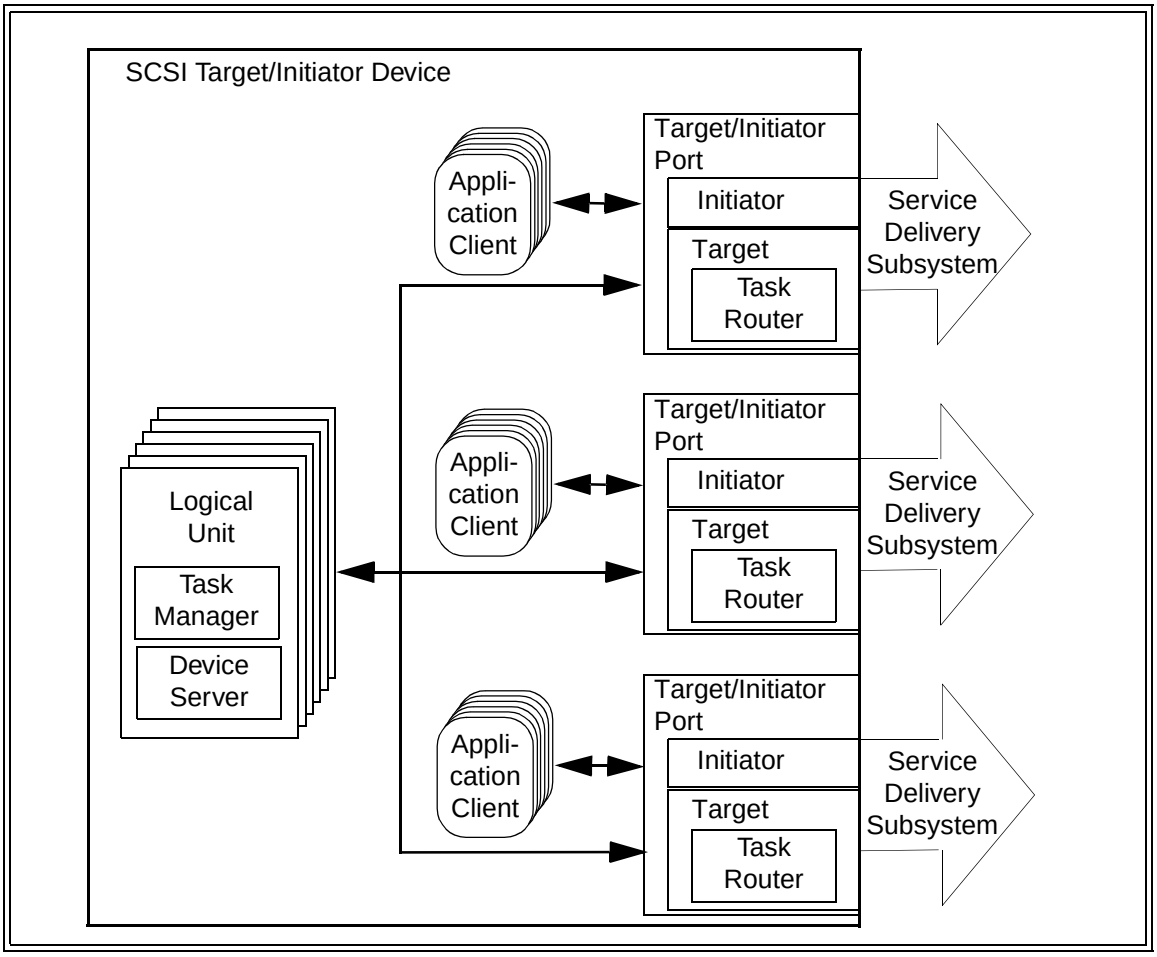

Figure 18 - Multiple port target/initiator SCSI device structure model

Two-way communications shall be possible between all logical units and all target/initiator ports, however, communications between any logical unit and any target/initiator port may occasionally be inactive. Two-way communications shall be available between an application client and a target/initiator port. Each target/initiator port shall accept commands addressed to LUN 0 and the task router shall forward them to a device server for processing. The REPORT LUNS commands (see SPC-2) shall be accepted by logical unit 0 from any target port and shall return the logical unit inventory available via that target port. The availability of a the same logical unit through multiple target/initiator ports is discovered by matching device 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 a target/initiator port are beyond the scope of any standards in the SCSI family of standards.

New SAM-2 section 4.11.4 SCSI initiator device view of a multiple port SCSI target device

In all cases when the term SCSI target device is used it refers to SCSI target devices or SCSI target/initiator devices that are using the target/initiator port as a target port. In all cases when the term target port is used it refers to a target port or a target/initiator port operation as a target.

In all cases when the term SCSI initiator device is used it refers to SCSI initiator devices or SCSI target/initiator devices that are using the target/initiator port as an initiator port. In all cases when the term initiator port is used it refers to an initiator port or a target/initiator port operation as an initiator.

A SCSI target device may be connected to multiple independent service delivery subsystems through target ports in such a way that no single initiator port can communicate with a logical unit using two or more of the target ports in the SCSI target device. In this case, the application clients in the SCSI initiator device cannot determine if a SCSI target device has multiple ports.

However, SCSI target devices with multiple ports may be configured where application clients have the ability to discover that one or more logical units are accessible via multiple target ports. Figure 19 and figure 20 show two examples of such configurations.

Figure 19 shows a SCSI target device with multiple target ports participating in a single SCSI Domain with two SCSI initiator devices. There are three SCSI devices, one of which has two target ports, one with one initiator port, and one with one initiator port. There are two target identifiers and two initiator 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 that have the ability to discover the logical units in the SCSI target devices are accessible via multiple target identifiers (i.e., target ports) and map the configuration of the SCSI target devices.

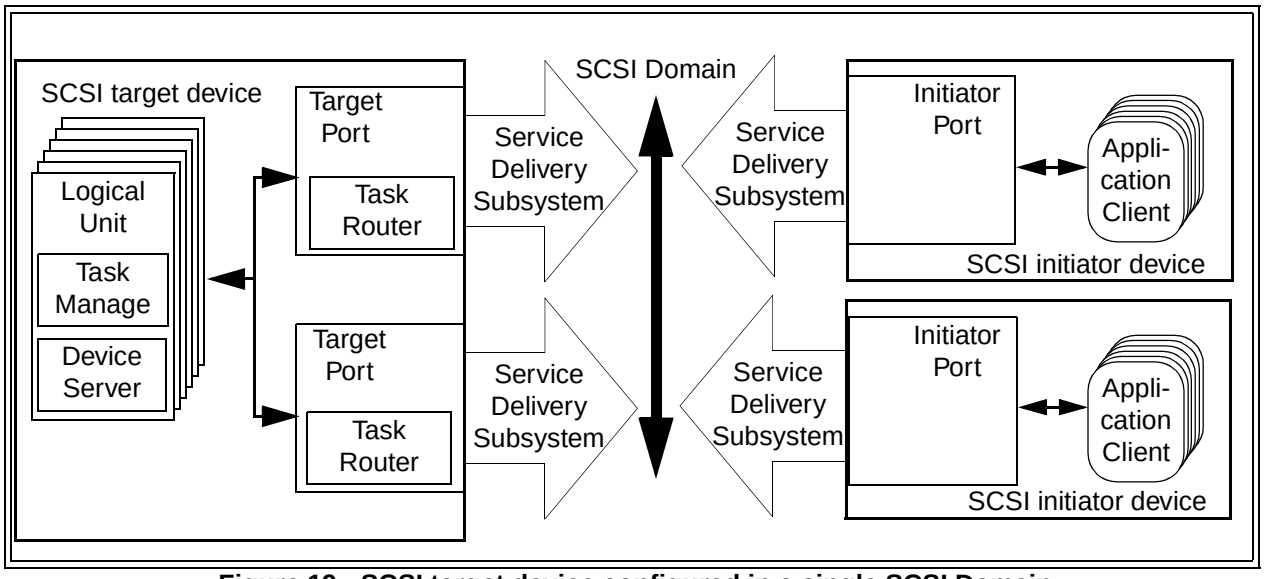

Figure 19 - SCSI target device configured in a single SCSI Domain

Figure 20 shows a SCSI target device with multiple target ports participating in two SCSI domains and a SCSI initiator device with multiple initiator ports participating in the same two SCSI domains. There is one SCSI target device two target ports and the a SCSI initiator device with two initiator ports. There is one target identifier and one initiator 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 20 and the configuration shown in figure 22 may be beyond the scope of the SCSI family of standards.

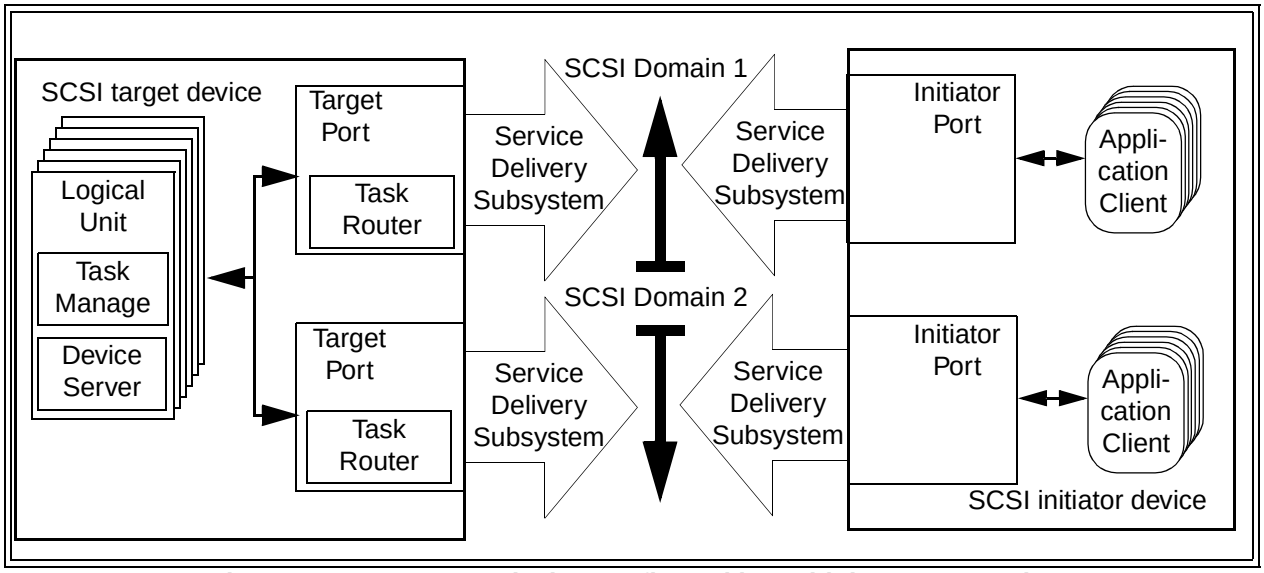

Figure 20 - SCSI target device configured in multiple SCSI Domains

Figure 21 shows the same configuration as figure 20 except that the two SCSI domains have been replaced by a single SCSI domain.

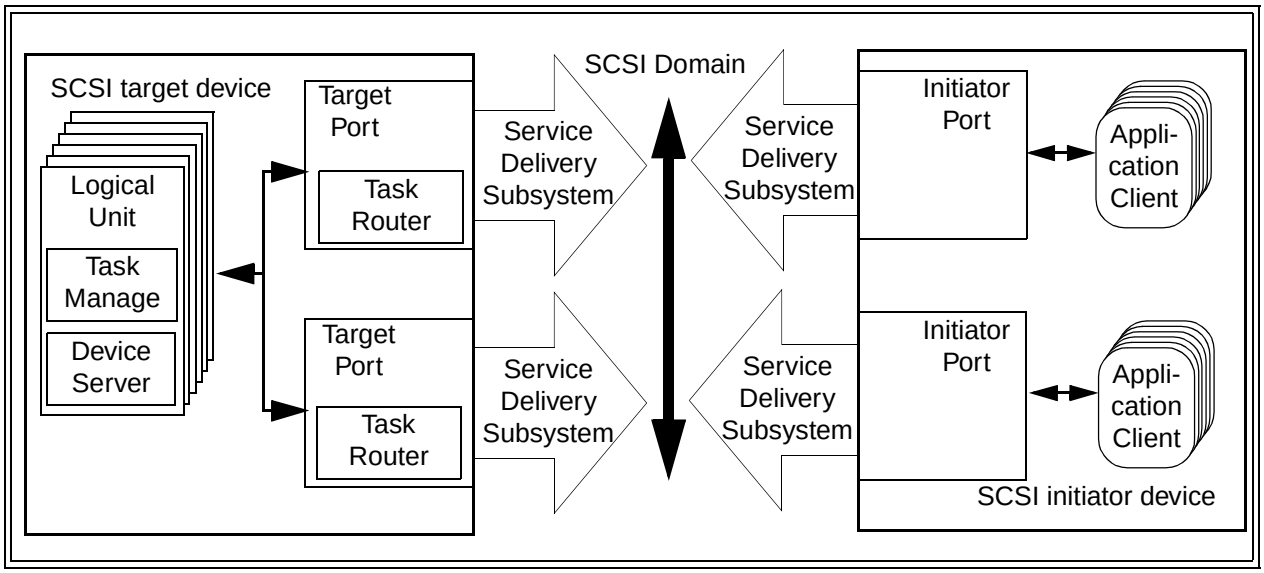

Figure 21 - SCSI target device and SCSI initiator device configured in a single SCSI Domain

This model for application client determination of multiple target port configurations relies on information that is available only to the application clients via SCSI commands. The initiator ports in the SCSI initiator devices (figure 19) or SCSI initiator device (figure 20 and figure 21) are unable to distinguish multiple target ports from individual target ports in two separate SCSI target devices.

new SAM-2 section 4.11.5 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 initiator ports. Therefore, a SCSI target device handles a SCSI initiator device with multiple initiator ports exactly as it would handle multiple separate SCSI initiator devices each with a single initiator port. For example, a SCSI target device handles the configurations shown in figure 20 and figure 21 in exactly the same way it handles the configuration show in figure 19.

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 initiator ports makes an exclusive access reservation via one initiator port, then access is denied to the other initiator port(s) on that same SCSI initiator device.

new SAM-2 section 4.11.6 Multiple port considerations for task management functions

Although, the task manager for each logical unit handles all the target ports and 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) may be enhanced to include one or more unique logical units embedded within another logical unit. In such cases, the model hierarchy diagram in is enhanced to become the diagram shown in figure 15 and the embedded logical units are called dependent logical units (see 3.1.22).

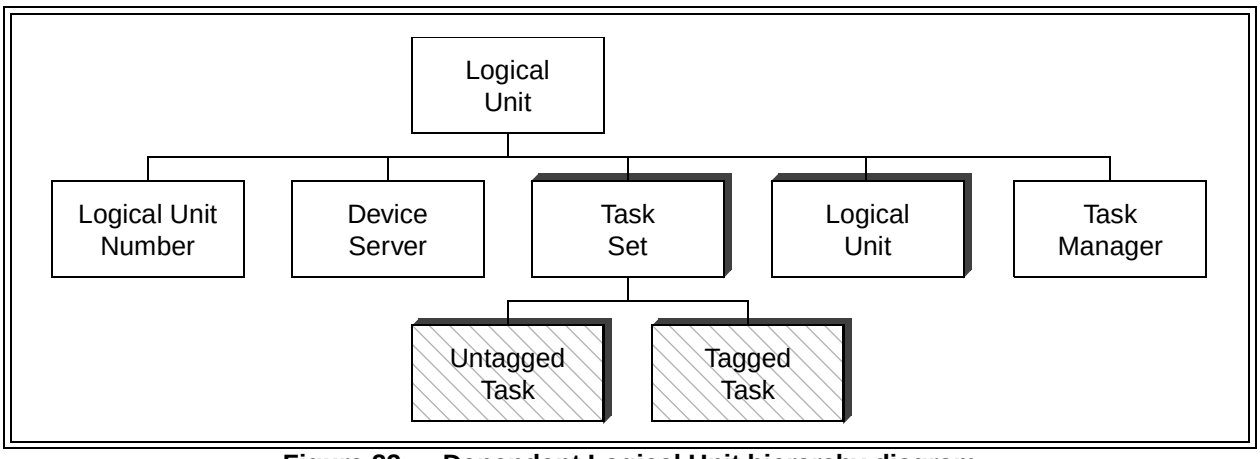

Figure 22 — Dependent Logical Unit hierarchy diagram

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 clause. 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 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 clause.