ENDL
TEXAS

Date: 28 February 2000
To: T10 Technical Committee
From: Ralph O. Weber
Subject: Bug in SAM-2 Task Identifier Definition

While reviewing FCP-2, I discovered a problem in the SAM-2 definition of Task Identifier. The problem adversely affects protocol standards such as SPI-4 and FCP-2 when they attempt to define a mapping between SAM-2 terms and protocol specific entities. As things stand right now, a protocol standard cannot show a mapping to the Task Identifier object with wording that will read sensibly.

There is an additional, less severe, problem with the SAM-2 object called Logical Unit Identifier, the actual definition of the object is not what most people think of when the name 'Logical Unit Identifier' is mentioned. While it is possible to describe the Logical Unit Identifier object sensibly in protocol standards, the lack of an intuitive definition for Logical Unit Identifier forces careful review and consideration in order to get the definitions right.

Review of the Current Object Definitions

SAM-2 (and SAM) define several objects for identifying tasks:

Targets use
Task Identifier
Untagged Task Identifier = Initiator Identifier + Logical Unit Identifier
Tagged Task Identifier = Untagged Task Identifier + Tag

Initiators use
Task Address
Untagged Task Address = Logical Unit Identifier
Tagged Task Address = Untagged Task Identifier + Tag

See SAM-2 clauses 4.9.2 and 4.9.3 (PDF page 49 in sam2r12.pdf).

Note that the Target's Task Identifier requires an Initiator Identifier while the Initiator's Task Address does not require a Target Identifier. This is because the Logical Unit Identifier is defined to include the Target Identifier, as follows:

Logical Unit Identifier = Target Identifier + Logical Unit Number

See SAM-2 clauses 4.8 (PDF page 48 in sam2r12.pdf).

It is difficult to judge the intentions behind these choices of definitions, so the analysis that follows may be incomplete and comments from the SCSI CAP (Commands, Architecture, and Protocols) Working Group are welcomed.

Inspection of the SAM-2 task management function definitions shows that the Logical Unit Identifier definition is more than a little fortuitous. By compounding the Target Identifier and Logical Unit Number in a single object, most of the task set management function definitions require only one argument. (Note: other compound objects are defined for other cases.)

The Problems

Replacing the Logical Unit Identifier object with its definition in the Target's Task Identifier definition we see that:

Task Identifier
Untagged Task Identifier = Initiator Identifier + Target Identifier + Logical Unit Number
Tagged Task Identifier = Untagged Task Identifier + Tag

That is, the target (and a protocol's description of a target's operation) is required to define a Task Identifier in terms of both the initiator and target identifiers. FCP-2 got this wrong, and might very well have an arduous task getting it right. As far as I can tell, there is no reason to require Task Identifier to have the definition currently in SAM-2 (and SAM). From a target's perspective, Initiator Identifier plus Logical Unit Number is a sufficient identifier for a task.

Furthermore, everybody or nearly everybody equates Logical Unit Identifier with Logical Unit Number, whereas Logical Unit Identifier is really a bookkeeping object defined to allow convenient constructions for the definitions of the task set management functions. It is even possible to view confusion between the Logical Unit Identifier and Logical Unit Number definitions during the development of SAM as the source of the task identifier problems.

A Simple Correction

The easiest way to correct these problems is to eliminate the Logical Unit Identifier object, or at least to restrict its usage to the definitions of the task set management functions. This would have the effect of changing the task identifier and task address objects as follows:

Task Identifier (target)
Untagged Task Identifier = Initiator Identifier + Logical Unit Number
Tagged Task Identifier = Untagged Task Identifier + Tag

Task Address (initiator)
Untagged Task Address = Target Identifier + Logical Unit Number
Tagged Task Address = Untagged Task Identifier + Tag

Note: These algebraic descriptions would need to be translated to English wording in SAM-2, but that task is within the editor's abilities.

Also, 5.6.3 (PDF page 77 in sam2r12.pdf) uses Logical Unit Identifier incorrectly as a substitute for Logical Unit Number in the following statements:

> The target's response to an incorrect logical unit identifier is described in the following paragraphs.
> The logical unit identifier may be incorrect because:

The only other uses of Logical Unit Identifier are in the tasks set management functions definitions in clause 6. It is doubtful that the definitions can be kept readable without using the Logical Unit Identifier object. So localizing the definition to clause 6 is recommended.

A More Complex Correction

While reviewing these issues, I couldn’t help wondering if good reasons existed for the removal of the I_T_L_Q nexus concept from SAM. Reinstating the I_T_L_Q nexus concept in SAM-2 would be a substantial challenge. Would it be worth the effort? This is an issue for SCSI CAP Working Group discussion. It should be noted that a change to I_T_L_Q nexus notation would allow the single task management function argument to be 'nexus' with individual task management functions using I_T, I_T_L, or I_T_L_Q nexus as appropriate.