

Date: December 11, 2003

To: T10 Committee (SCSI)

From: George Penokie (IBM/Tivoli)

Subject: SBC-2: Problems with application client modifiable logical block reference tag

1 Overview

Within the end-to-end protection that was recently approved there are three fields defined each provides a different type of protection:

- a) The logical block guard protects the user data within each logical block as the logical blocks transit from the source application client to the destination device server and back. Once generated, it remains constant at all times;
- b) The logical block reference tag protects the relationship between logical blocks as those blocks transit from the an application client to a device server. Once generated it remains constant until unless another application client in the transit path is required to modify it (e.g., any device that issues SCSI commands to different logical blocks than the logical blocks indicated in the received SCSI commands). But because the tag is tightly associated with the LBA (i.e., LBA locked data) the modification by any SCSI device does not cause false read errors as the LBA is globally known to every device; and
- c) The logical block application tag allows an application client a place to store values that it may check on a read of that logical block (e.g., to determine if the logical block being returned is the logical block that was last written). The knowledge of how to interrupt this tag is only known to the application client and to others that application client chooses to share that knowledge with.

A proposal has been submitted that would allow the logical block reference tag to contain a value determined by an application client (i.e., with no requirement that the tag be related to an LBA or any other specific value). If accepted this proposal would allow devices that would normally have to modify the logical block reference tag to pass it through without medication (i.e., non-LBA locked data). This would work because the value to check against the tag in the logical block would be placed into the CDB of the SCSI command.

However, there are several problems in placing an application defined value in the logical block reference tag that the proposal does not address. This proposal defines those problems and requests that those problems be resolved before accepting an application client modifiable logical block reference tag.

2 Overlapping logical block reference tag values

2.1 Problem description

Anytime an application client transmits non-LBA locked data the checking of logical block reference tags on reads may cause false read errors, except in tightly controlled environments. This occurs, over time, when multiple write operations are sent to overlapping ranges of logical blocks. The result of such writes is that the values within the logical block reference tags in consecutive logical blocks may no longer be incrementing properly (e.g., instead of $n, n+1, n+2$, etc. it could be $n, n+1, m, m+1$, etc.). The result of a read that crosses the overlapped boundary would be an error if the logical block reference tag is checked during the read.

Overlapping logical blocks may occur:

- a) anytime writes of non-LBA locked data in mixed with writes of LBA locked data; and
- b) when writes of non-LBA locked data overlap logical blocks with other writes of non-LBA locked data.

2.2 Example of non-overlapping configuration

The overlapping problem can be controlled such that reads do not cause a problem if the paths between all the originating initiators and all the destination targets pass through a single device that controls the address space such that the reference tags are always sequential. See figure 1 for an example of such a configuration.

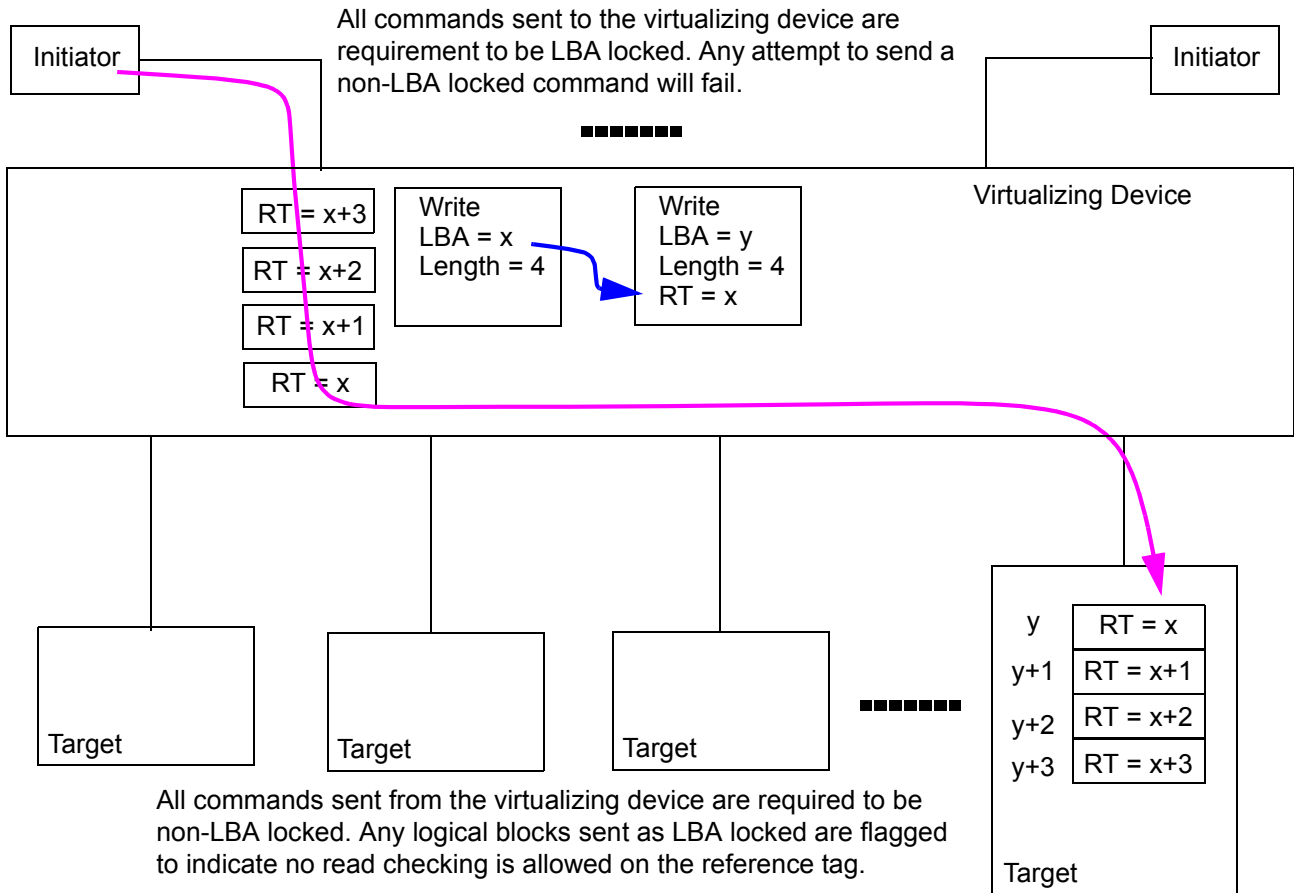


Figure 1 — Non-overlapping configuration

3 Coalescing read and write operations

3.1 Problem description

Anytime an application client transmits non-LBA locked data to a device that stripes logical blocks across multiple targets the ability for that device to coalesce write or read operations to a single target is limited. The problem is that the reference tag in the logical blocks transferred in the coalesced command are, by definition, not sequential and, therefore, can't be checked based off information received in the command.

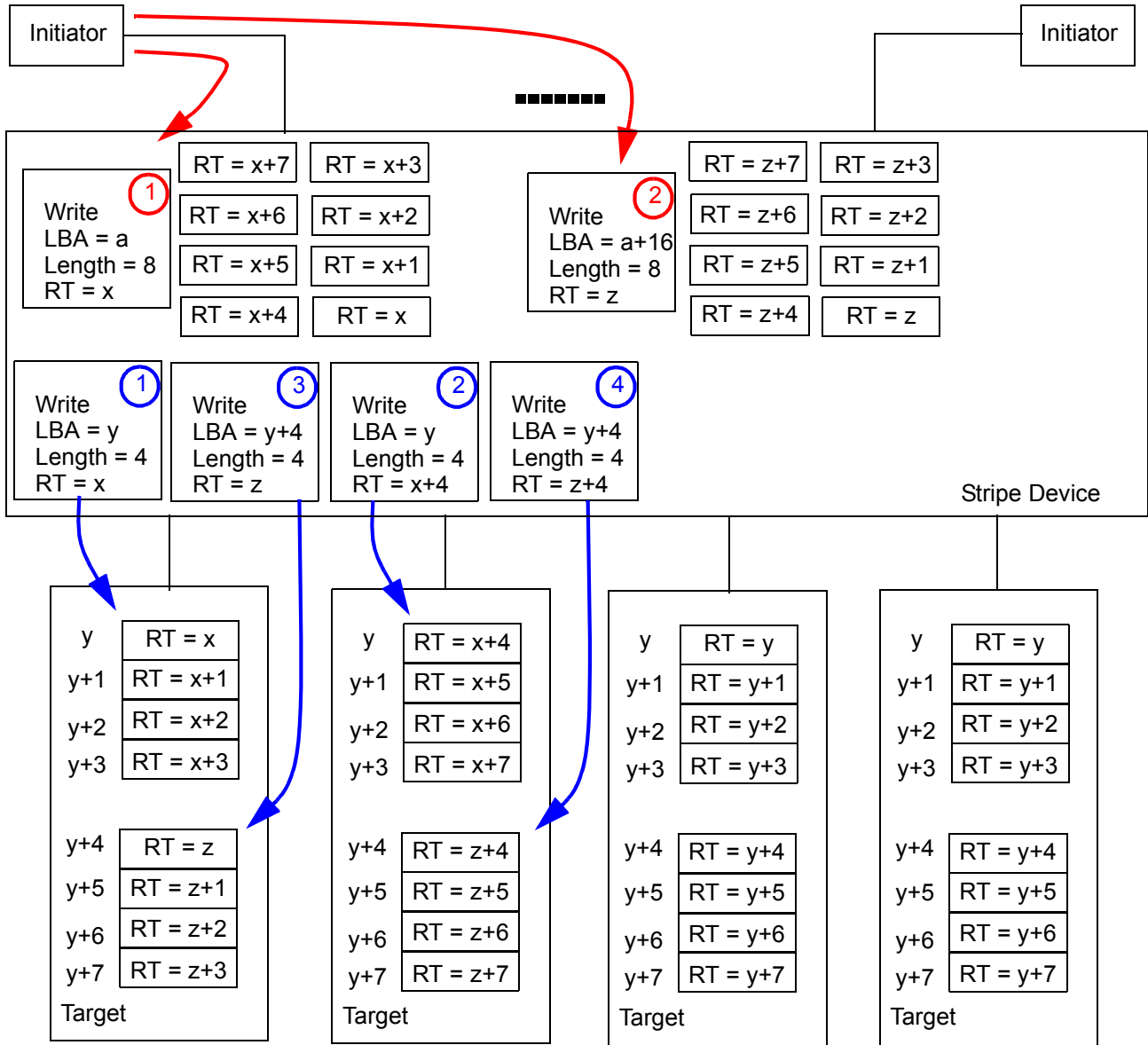


Figure 2 — Strips with no coalescing

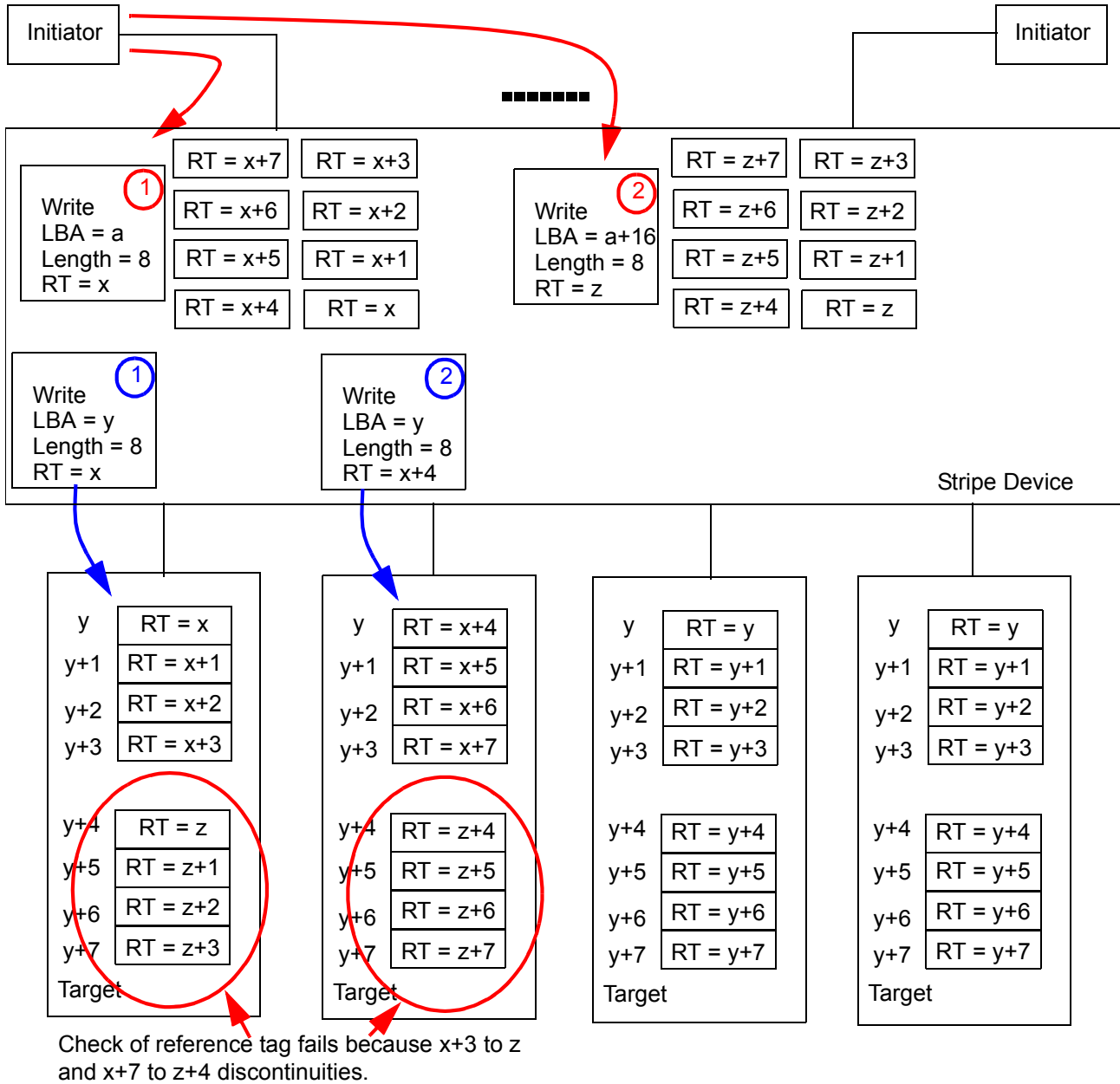


Figure 3 — Strips with coalescing