Date: June 30, 2003

To: T10 Committee (SCSI)

From: George Penokie (IBM/Tivoli)

Subject: End-to-End Error Cases

1 Overview

As part of the discussion on end-to-end data protection the question of what errors are being detected has come up several times. This document describes the error cases which end-to-end data protection protects against and the mechanism for detecting those errors.

2 Storage device LBA translation error

This is a case where a storage device (e.g., RAID 5 controller) that receives a write to an LBA that is translated into another LBA and then transmitted to another storage device (e.g., disk drive) and that translation fails. An example is shown in figure 1. Although this example is specific to a RAID controller it is valid for any target device the translates received logical unit numbers from the received value to a new target port/logical unit destination.

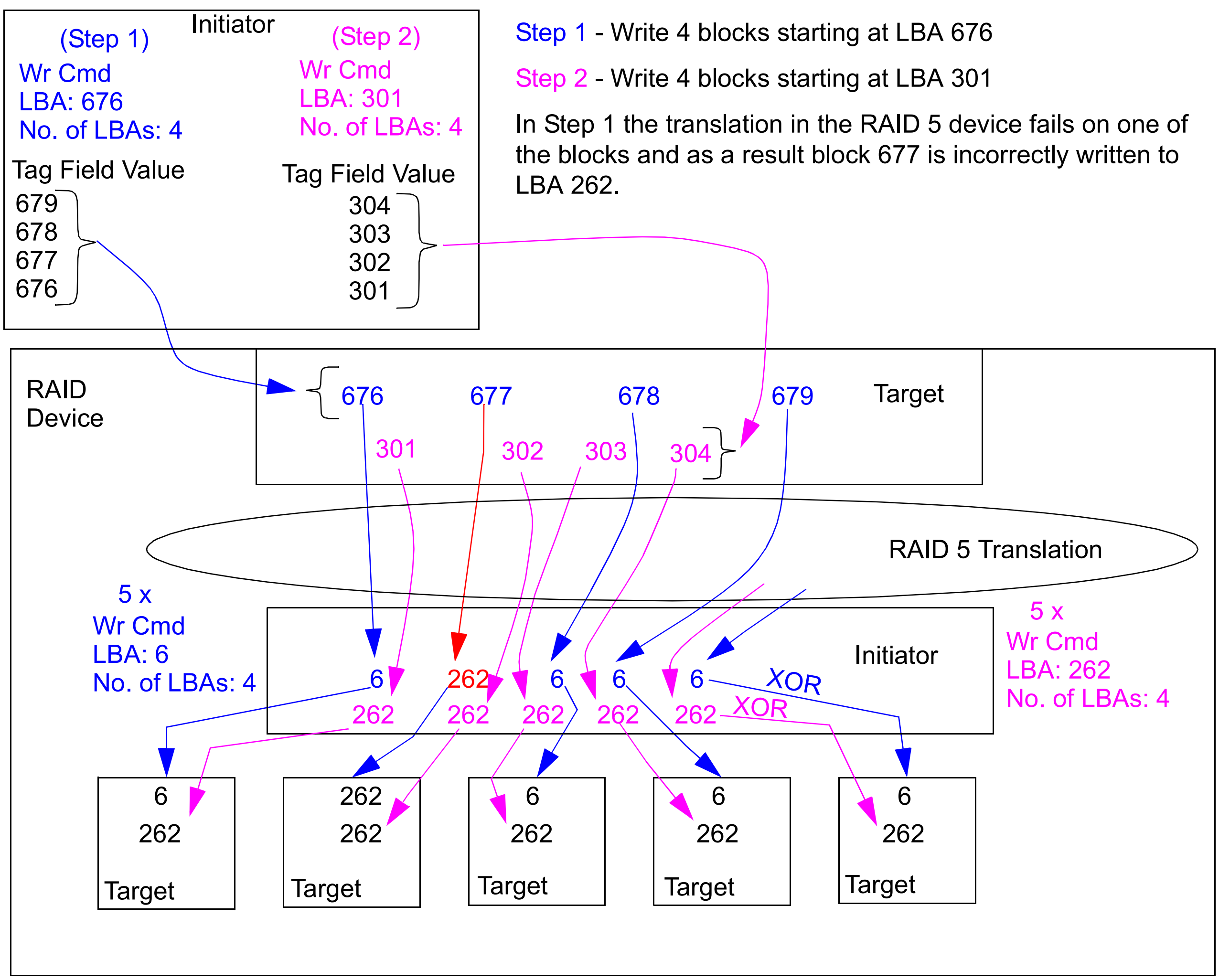

Figure 1 — Storage device LBA translation error example

Table 1 describes the methods for detecting the error case shown in figure 1 if only step 1 occurs.

Table 1 — LBA translation error detection (Step 1 only case)

Error [a]	Result of Error	Detection Method during Write	Detection Method during Read
Fixed translation error (i.e., LBA x always translates to LBA z instead of LBA y)	All requests to read or write LBA x results in a read or write of LBA z	Not possible as everything looks good to the targets	Not possible as everything looks good to the initiators. This is not really an error as the data read is the data that was written.
Intermittent or random translation error (i.e., LBA x sometimes translates into something other that LBA y)	Requests to read or write LBA x results in a read or write some LBA possibly not LBA y (i.e., read or write of wrong data).	Not possible as everything looks good to the targets	This can be detected if: [b] a)there is a field that is written on each block of data (e.g., generation information); b)the value in the field is not determined by the data in the block; c)is not changed as the block moves from the original source to the final destination; and d).the device reading the block knows the algorithm used to write the field.
An intermittent write error with a read of the correct LBA	Request to read the LBA may result in old data being returned (i.e., read of stale data)	Not possible as everything looks good to the targets	This can be detected if: [b] a)there is a field that is written on each block of data (e.g., generation information); b)the value in the field is not determined by the data in the block; c)is not changed as the block moves from the original source to the final destination; and d).the device reading the block knows the algorithm used to write the field.

[a] An example of a error would be if a pointer into the cache was misdirected.
[b] A solution may be to define a vendor specific field of size n. Then assume that only applications that have to have knowledge of the fields contents should use it.

Table 2 describes the methods for detecting the error case shown in figure 1 if step 1 and step 2 occur.

Table 2 — LBA translation error detection (Step 1 and 2 case)

Step	Error	Result of Error	Detection Method during Write	Detection Method during Read
1	Fixed translation error.	All requests to read or write LBA x results in a read or write of LBA z	Not possible as everything looks good to the targets	This can be detected if: a) there is a field that is written on each block of data; b) the value in the field is not determined by the data in the block; c) is not changed as the block moves from the original source to the final destination; and d). the device reading the block knows the algorithm used to write the field.
2	No errors but the data is written to the same LBA as was incorrectly written to in step 1			

2.1 Mishandling of blocks within a Device

This is a case where part of the data is lost due to improper handling within a device that temporarily stores that data. An example would be a device that improperly caches received data before retransmitting. An example is shown in figure 2.

This type of a device would look like a target to the device sending the data and would look like an initiator when transmitting the data. There is currently no method defined in SCSI that would allow this type of lost data to be detected. However, as shown in the example, placing an incrementing tag value with each block could detect this class of error.

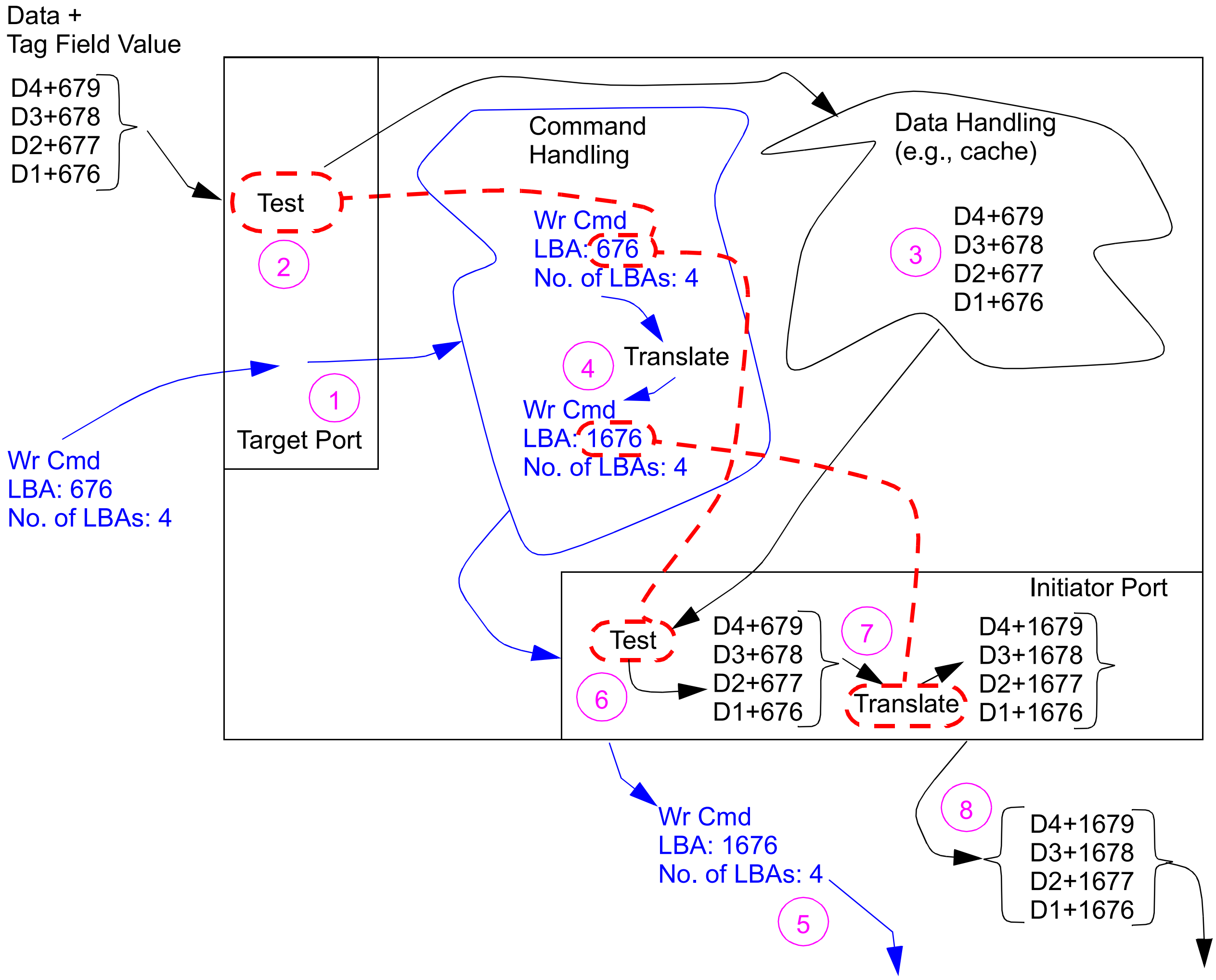

Figure 2 — Write command cache data loss example

The steps in figure 2 are:

1) Receive and save write command;
2) Receive data and test tag against information in write command. This test is not required because the transport should find this class of error;
3) Store data in temporary storage (e.g., cache);
4) Translate LBA to the destination LBA and create a new write command;
5) Transmit the new write command to the destination logical unit;
6) Move data from temporary storage and test tag again information from original write command;
7) Translate tag to match LBA in new write; and
8) Transmit data and tag to destination.

3 Bit Errors Inside a Device

This is a case where the data in a block is changed while it is being processed and/or stored in a device (e.g., a switch) that receives and then retransmits the block. The protection (e.g., CRC) that is used to protect the transmission of the block is generated at the transmitter and checked/stripped at the receiver. As a result there is no standardized protection on the block between receivers and transmitters. An example is shown in figure 3.

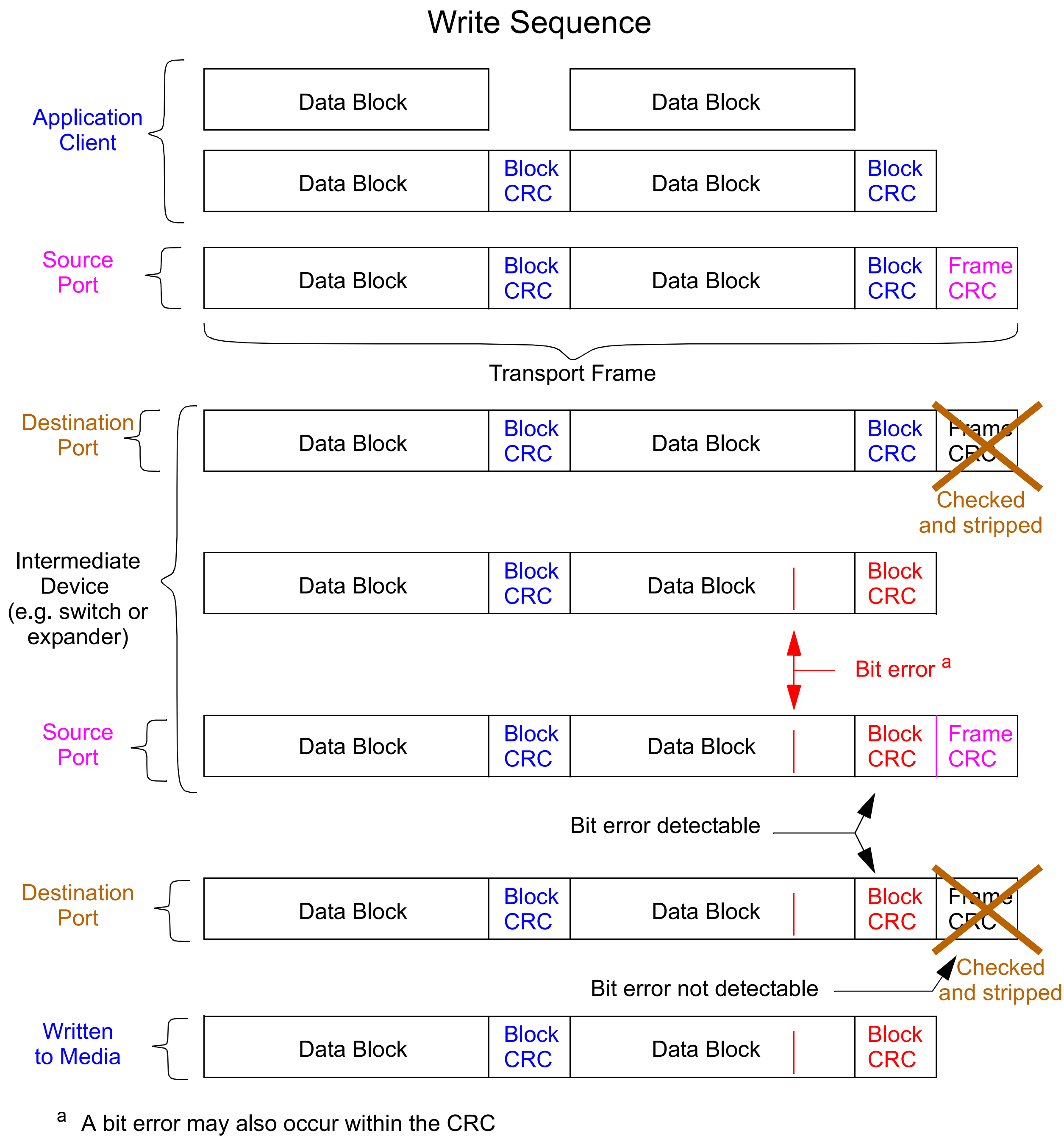

Figure 3 — Device bit error example

Table 3 describes the methods for detecting the error case shown in figure 3.

Table 3 — Bit error detection

Error	Result of Error	Detection Method during Write	Detection Method during Read
A bit transposition occurs in a device in the data path that retransmits data	A block of data is corrupted.	Detectable at the write destination if: a) there is protection placed on each block at the source based on the contents of the block; b) the protection is not regenerated as the block moves from the original source to the final destination; and c) the protection is checked at a write destination.	Detectable at the read destination if: a) there the protection placed on each block at the original source (i.e., the source of the write) was based on the contents of the block; b) the protection is not regenerated as the block moves between the original source to the final destination; c) the protection was written, unchanged, to the media; and d) the protection is checked during the read.

4 Lost Frame

This is a case where a frame is lost during transmission. The frame level protection (e.g., offset) that is used to protect the transmission of the frame is generated at the original transmitter and checked at the final receiver. This should provide enough protection on the application client side as the offset value is set by the application client, if the transport protocol supports this feature. An example is shown in figure 4.

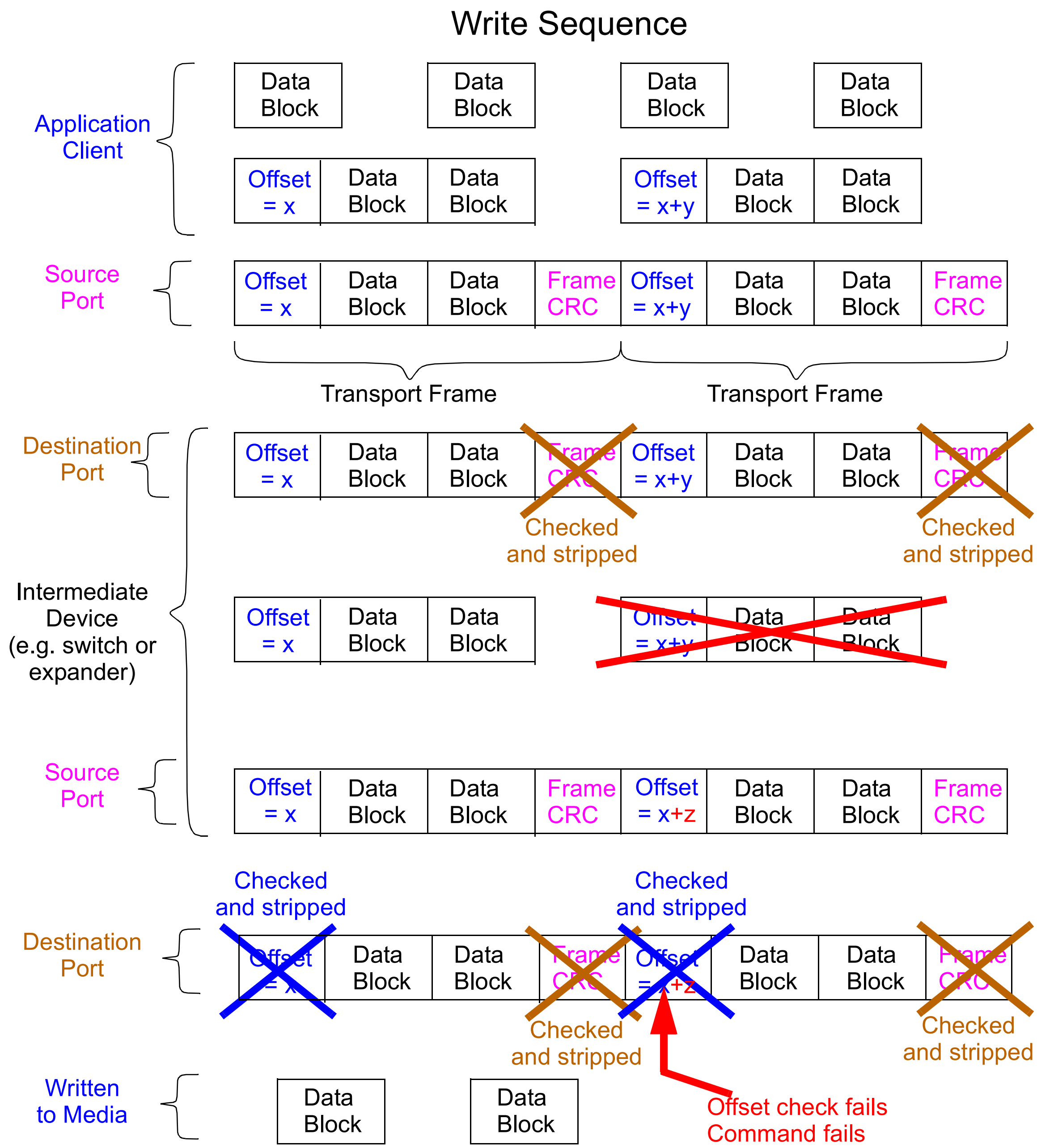

Figure 4 — Lost frame example

Table 4 describes the methods for detecting the error case shown in figure 4.

Table 4 — Lost frame detection

Error	Result of Error	Detection Method during Write	Detection Method during Read
A frame is lost in a device in the data path.	A frame of data is lost.	Detectable at the write destination if: a) there is protection placed on or in each frame at the source that starts at a known value and increments on every frame boundary; b) the protection is not regenerated as the frame moves from the original source to the final destination; and c) the protection is checked at a write destination.	Detectable at the read destination if: a) there is protection placed on or in each frame at the source that starts at a known value and increments on every frame boundary; b) the protection is not regenerated as the frame moves from the original source to the final destination; and c) the protection is checked at a read destination.