

Target-to-Initiator Error Recovery **Enhancement Proposal**

Akihiro Shimura, CANON INC.

August 12, 1998

This document proposes enhancement to the target-to-initiator error recovery based on whole buffer re-transmission.

1. Background

The buffer writes from the target are performed by reliable IEEE 1394 asynchronous transactions. In error recovery process, it is preferable to re-send only failed 1394 transactions from the bus's point of view because it saves bandwidth. Re-sending whole buffer requires same sized re-sending buffer in the target (which will limit maximum buffer size in the initiator) in addition to wasting bandwidth.

To recover from the error by only re-sending failed transactions, the initiator is required to keep the contents of the buffer across the error.

Because SBP-2 only specifies that buffer shall be accessed via write transactions when direction bit is set to one, whether the initiator keeps the contents of buffer or not depends on the underlying implementation on the initiator (i.e. interpretation on the buffer's "direction" bit set to one).

2. Recovery with whole buffer re-transmission

This method does not require integrity of buffer contents on the initiator, but it wastes bandwidth in recovery. It requires another same size of re-send buffer in target, and target resource restricts the usable size of the buffers associated with the ORBs.

3. Recovery with transaction(s) re-sending

This method saves bandwidth optimally in recovery.

It saves target resource for recovery, and allows efficient use of large buffers associated with ORBs.

But, it requires the integrity of buffer contents across the error.

4. Enhancement Proposal

We propose an enhancement to the recovery model based on the whole buffer re-transmission to take advantage of shared memory model while keeping independence of implementations.

The enhancement is based on negotiated MAX_T2I_DATA_SIZE reflecting initiator's preference.

The definition of MAX_T2I_DATA_SIZE will look like following;

MAX_T2I_DATA_SIZE provided by the target indicates the maximum unit of target-to-initiator data the target can guarantee to hold and re-send in case of recovering re-queued TRANSPORT_T2I_DATA command.

Initiator may set this parameter value smaller than the value provided by the target. The value zero specified by the initiator indicates that the target needs not to guarantee to hold and re-send data beyond 1394 transaction.

Target shall guarantee to hold and re-send data up to this (negotiated) parameter size.

Responsibility to hold the contents of buffer beyond this parameter size belongs to the initiator.

The initiator shall avoid to issue TRANSPORT_T2I_DATA command which has associated buffer larger than this parameter in case that the buffer contents may not be held.

5. Conclusion

This enhancement proposal archives both benefits in “Recovery with whole buffer re-transmission” and “Recovery with transaction(s) re-sending”. In other words, this enhancement works without requiring integrity of buffer contents and works efficiently with large buffer that provides integrity of buffer contents.

Alterable MAX_T2I_DATA_SIZE parameter allows both ends to optimize resource/efficiency balance.

Note: Though this enhancement will not be optimal in the spirit of shared memory model, almost the same result will be derived by this enhancement.

6. Example

At first, target reports the maximum amount of data target can hold and re-send in recovery. For example, target reports this parameter as 32KB.

If the initiator prefers to keep the contents of buffer by itself, the initiator alters this parameter to zero. In this case, re-transmission unit in recovery becomes 1394 transaction(s), and the target can release 32KB of re-sending buffer, and both ends can efficiently use the buffer size larger than 32KB (e.g., 1MB or so).

If the initiator prefers to make target to hold re-send data, initiator limits the size of the buffer associated with the ORBs up to 32KB. If initiator never uses the size of the buffer associated with the ORBs beyond 4KB, initiator alters this parameter to 4KB. As a result, target can release 28KB of unused re-sending buffer and re-transmission unit becomes 4KB.