Class 3 Error Detection and Recovery
ANSI T10 Working Document 97-189RO

Error Detection and Recovery for Sequential Access Devices in FCP

Scope

Problems exist in PLDA in detecting and correcting error conditions on sequential access devices (tapes) in PLDA. The basic causes of these problems are due to the lack of a guaranteed delivery protocol and the implicit state information intrinsic to sequential access devices. More specifically, lost frames in FCP can result in FC information units being lost. ULP recovery is not sufficient for a variety of reasons, including an inability to detect such errors, the effort required to implement recovery mechanisms, and the extended time required to detect and recover from error conditions.

Requirements

An ideal solution will incorporate the following characteristics:

Provide the ability to recover from lost frames in FCP for sequential access devices
Interoperability with block and sequential access devices
No or minimal changes to FC-PH and PLDA
No additional protocol overhead for normal operation
Can be implemented with existing silicon

Problem Analysis

On stream and media changer devices there are two classes of commands for which it is critical to know whether the command was accepted by the target, and then whether successful completion of the command occurred.

The first class, unique to these devices, are those that alter the media state or content in a way that simply re-executing the command will not recover the error. These include read/write/position/write filemarks (the tape is repositioned past the referenced block(s) or files only if the operation started; how far the operation continued is critical to proper recovery) and move medium/load/unload medium (which may have actually hanged the medium in the target). Unfortunately, these comprise most of the commands issued during normal operation of the subsystem.

The second class, which is not unique to these devices, are those in which information is lost if it is presumed sent by the target, but not received by the initiator. These commands include request sense and read/reset log. Loss of sense data also may affect error recovery from failed commands of the aforementioned media move/change class, but it may also affect proper error recovery for cached/RAID disk controllers as well.

On a parallel SCSI bus, the host adapter has positive confirmation that the target accepted the command by the fact that the target requested all bytes of the CDB and continued to the next phase without a Restore Pointers message. Such confirmation is only implicit in a serial protocol by receipt of a response message, such as Transfer_Ready or Response. In cases of some commands, this implicit confirmation may require a lengthy period of time, during which mechanical movement requiring several multiples of E_D_TOV occurs (in FLA environments, R_A_TOV may be the appropriate value). Similarly, the target has positive confirmation that the host has accepted sense or log data immediately upon completion of the data and status phases; this data may now be reset. In a serial environment, this is only implicit by receipt of the next command. Note that a change to the target to only clear sense/log data on receipt of a command other than request sense or read/reset log would eliminate this problem.

In summary, the errors that are of concern are where FCP information units are lost in transit between an FCP initiator and target. The cause for such loss is not specific, but is assumed to be cases where a link level connection is maintained between the target and initiator, and some number of FCP IU's are dropped. Other cases are either handled by PLDA through existing methods, or may be generally classified as unrecoverable and treated in a fashion similar to a SCSI bus reset.

In order to meet the defined requirements, any proposed solution must enable the initiator to make the following determinations:

An error condition occurred (an FCP IU is expected and not received, or not responded to)
If FCP_CMND, was it received by the target
If FCP_DATA, was it received or sent by target
If FCP XFER_RDY or FCP_RSP, was it sent by target

Note that the solution must work in a Class 3 environment, preferably with no change to existing hardware.

Tools For Solution

The tools prescribed in FC-PH for FC-2 recovery are the Read Exchange Status (RES), and Read Sequence Status (RSS) Extended Link Services, and the Abort Sequence (ABTS) Basic Link Service.

RES is an appropriate tool for the host adapter to use; its function is to inquire of the status of an operation during and for some period of time after its life. Unfortunately, in several of the cases of interest, the RX_ID is unknown to the exchange initiator. In these cases, the initiator must use an RX_ID of 0xFFFF, which, combined with the FC_PH wording that "...the Responder destination N_Port would use RX_ID and ignore the OX_ID", means that if the Responder had not received the command frame, the RES would be rejected, and if the Responder had received the command and sent the FCP_RSP response frame, the RES would be rejected, in both cases with the same reason code; only in the case where the command was in process but no FCP_RSP response frame had been sent by the Responder would a useful response be sent. Real implementations appear to search for the S_ID - OX_ID pair when the RX_ID is set to 0xFFFF in the RES request, and this behavior needs to become required.

Further, even if this change is implemented, in the case of a non-transfer command, it is impossible to detect the difference between a command that was never received and a command whose response was lost unless the target retains ESB information for a period of R_A_TOV after the exchange is closed.

Further clarification of the text in the standard (FC-PH) is required, and requirements specified in profile documents.

Similar arguments apply to the use of the RSS, though the wording of the applicable section uses the word "may" rather than "would".

ABTS, while recommended in FC-PH for use in polling for sequence delivery, is always interpreted as an abort of the exchange in FC-PLDA, and is therefore not useful for this purpose.

Proposed Solution

After (2 x R_A_TOV) with no reply received:

Issue RES for the exchange issued (using new OX_ID). If no ACC response to RES within (timeout-period), send ABTS to abort the RES exchange, and allow retries at reasonable intervals. If all of the issued RES's never receives a response, allow ULP timeout to occur, along with ULP recovery.

If the FCP_CMND was not received by the target (i.e., the initiator receives an LS_RJT for the RES, with a reason code indicating that the OX_ID is unknown), send ABTS to abort the original sequence/exchange. Resend the command (using a new OX_ID).

Note that it is assumed here that the target retains ESB information for the error detection timeout period after the response has been sent. In this way the initiator may determine the difference between a command that was never received and one whose reply sequence(s) were lost.

If an FCP_XFER_RDY was sent by the target , but not received by the initiator, issue an SRR Extended Link Service (see below for details) frame to request sequence retransmission. When the FCP_XFER_RDY is successfully received, the data is sent, and the operation continues normally. No error is reported to the ULP, though the error counters in the LESB should be updated. If the SRR receives a LS_RJT, perform sequence error recovery as documented in PLDA section 9.1, 9.3.

If an FCP_DATA sequence was sent by the target, but not all received by the initiator, issue an SRR Extended Link Service frame to request retransmission of the sequence (or portion thereof) that was not successfully received. The received data is delivered to the ULP, and no error is reported. If the SRR receives a LS_RJT, perform sequence error recovery as documented in PLDA section 9.1, 9.3.

If an FCP_RSP sequence was sent by the target, but not received by the initiator, issue an SRR Extended Link Service frame to request retransmission of the sequence. The response is delivered to the ULP, and no error is reported. If the SRR receives a LS_RJT, perform sequence error recovery as documented in PLDA section 9.1, 9.3.

If an FCP_DATA sequence was sent by the initiator, but not successfully received by the target, the remainder of the data sequence is retransmitted. The operation should complete with no error indication to the ULP. This action may optionally use the SRR service to inform the target that data is being retransmitted.

It is the responsibility of the initiator to determine the appropriate action (retry, allow ULP time out, or return status to ULP) required based on the information determined by RES and other internal state.

Note that link recovery should be treated as the equivalent of a bus reset.

SRR Extended Link Service

The SRR (Sequence Resend Request) Extended link service frame follows the rules for extended link services as defined in FC-PH Rev 4.3, Section 23.1. A new Link Service command code in R_CTL needs to be added to FC_PH. The next available value is 0001 0011b.

The SRR payload and reject codes are defined below. The Accept does not require a payload. The direction flag indicates to the target that the initiator is requesting sequence data transfer to (0) or from (1) the target. All other fields are as defined in FC-PH.

Item	Size Bytes
SEQ_ID	1
Direction	1
OX_ID	2
RX_ID	2
Low SEQ_CNT	2
High SEQ_CNT	2

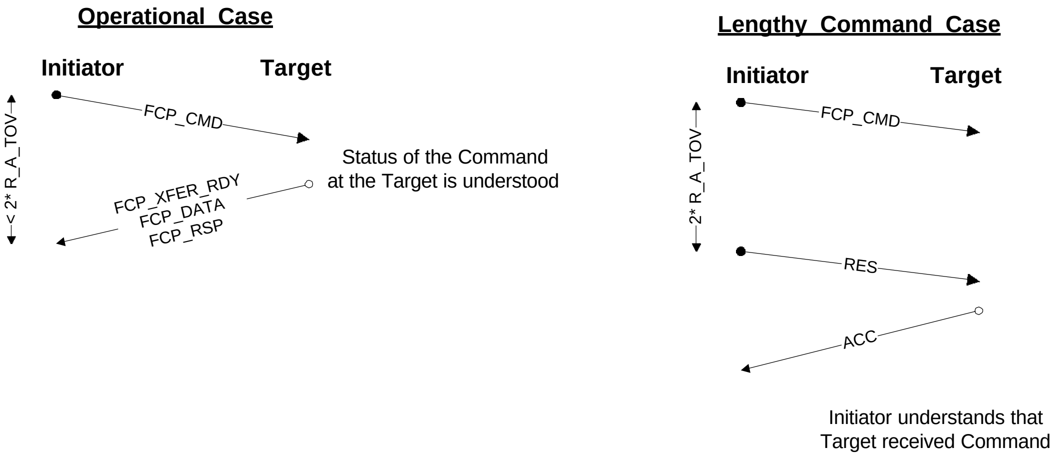
SRR Payload

Encoded Value	**LS_RJT Reason code explanation**
0x00052A00	Cannot resend last sequence
Reserved	

SRR LS_RJT Reason Codes

Example Data Flow Diagrams

Class 3 Operation for Tape Devices on FC-AL using FCP



Operational Case

Status of the Command at the Target is understood

Lengthy Command Case

Initiator understands that Target received Command

LS_RJT Codes for RES command:

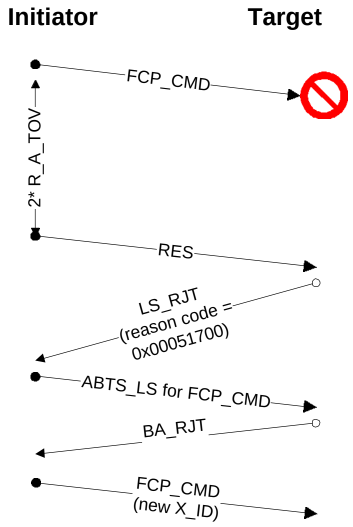
0x00051700 - Invalid Exchange
0x00052A00 - Cannot provide Sequence Information
0x000B0000 - Does not Support Command

LS_RJT Codes for SRR command:

0x00052A00 - Cannot resend last Sequence

Example Data Flow Diagrams Continued

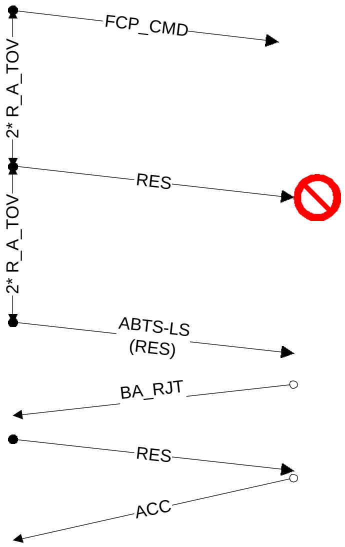
FCP_CMD Lost



Both Initiator & Target Understand CMD NOT Received

Command Resent and Processed Normally

RES Lost

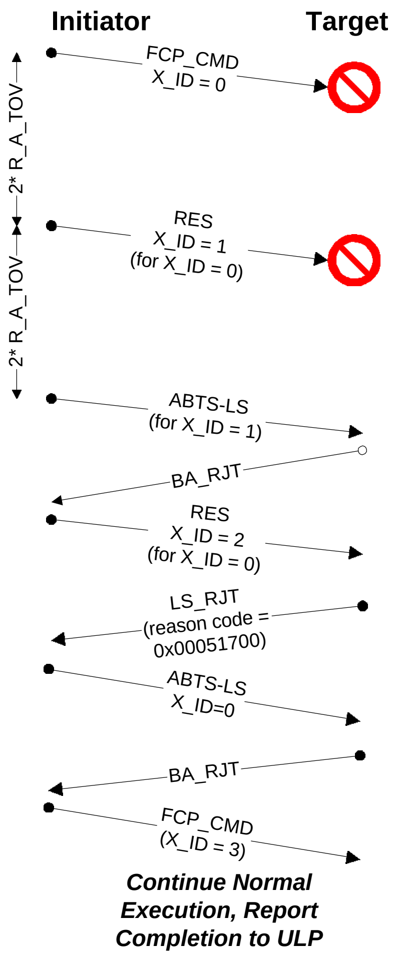


Both Initiator & Target Understand CMD Received

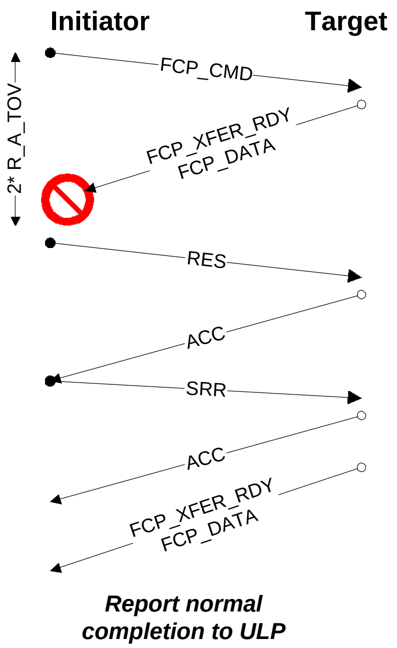
Command Processed Normally

Example Data Flow Diagrams Continued

CMD & RES Lost

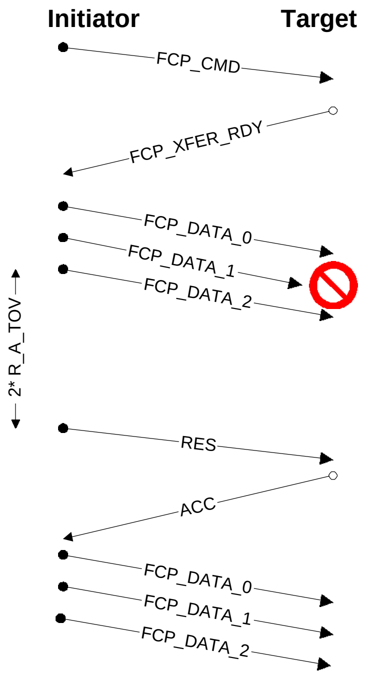


Lost Reply Sequence

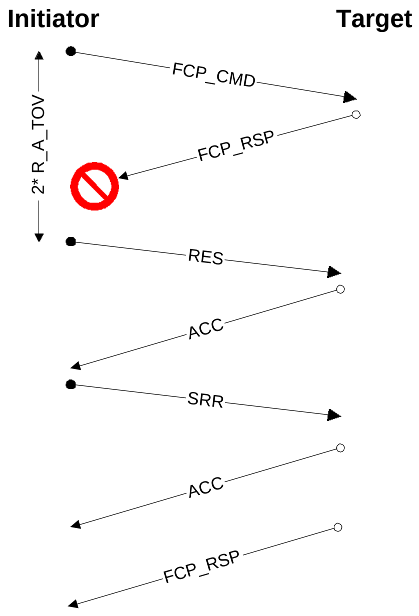


Example Data Flow Diagrams Continued

Lost Write Data



Lost RSP



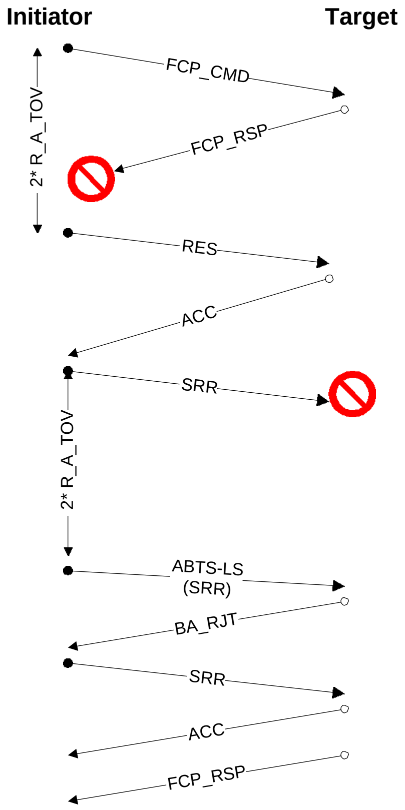
Report Normal Completion to ULP

Requires Persistence of ESB Data

Note **-** Need to request change sequence initiative before sequence resend (use SRR with direction flag to inform target that sequence resend is happening, or other link level request) in lost write example

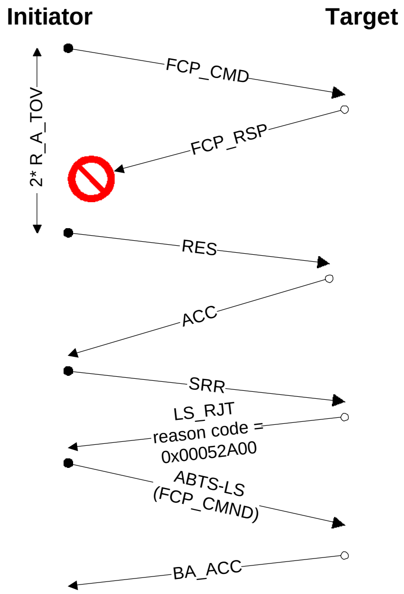
Example Data Flow Diagrams Continued

Lost SRR



SRR is lost. Retry SRR one more time and if still unsuccessful, then abort command and notify ULP.

No FCP_RSP data

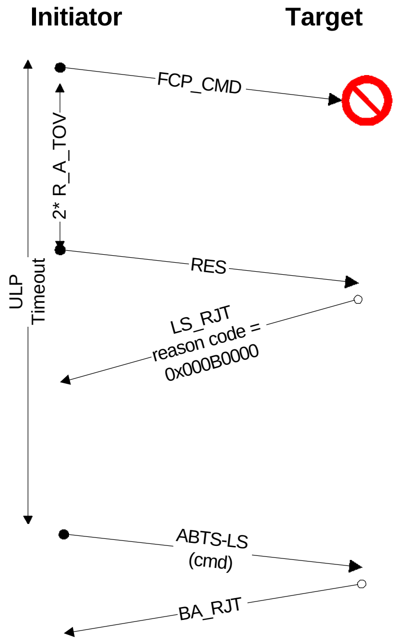


Target cannot resend the last sequence. The initiator is free to abort the command and notify the host with status information. NOTE: if an FCP_RSP was dropped then an BA_RJT will be returned from the ABTS-LS.

Example Data Flow Diagrams Continued

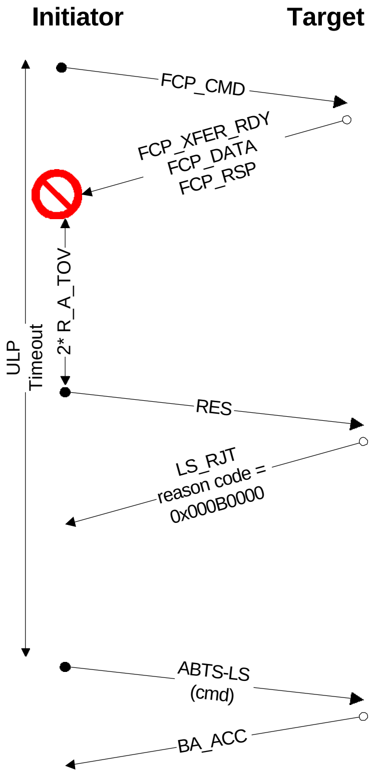
Error Recovery Disk Device

Lost Command



Device is not a Tape class device. Let ULP handle error recovery.

Lost Reply

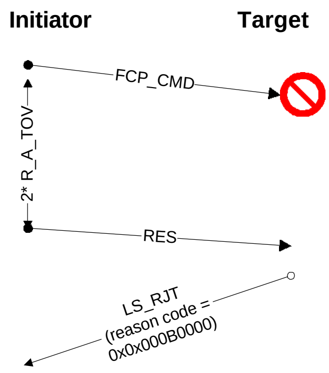


Device is not a Tape class device. Let ULP handle error recovery.

Example Data Flow Diagrams Continued

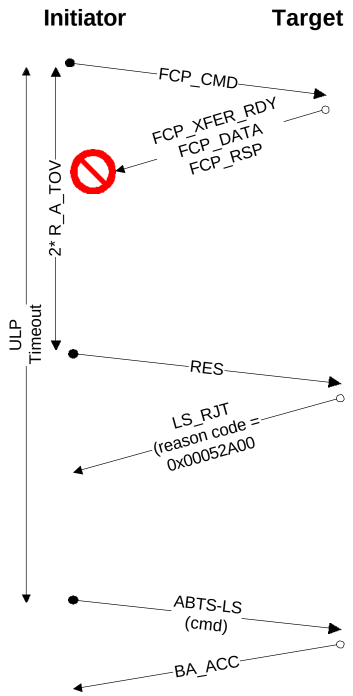
Tape Device Lacking Functionality

Lost Command



Device does not support this error recovery. Let ULP handle error recovery.

Lost Reply

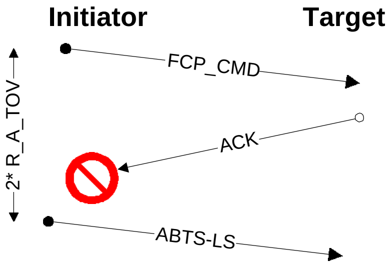


The Target does not have enough sequence information to respond to the RES. At this point the Initiator may not be able to determine the state of the command and should fall back on using the ULP timeout for error recovery. If the Initiator can verify that the command is broken then it can abort the command and notify the ULP before the ULP timeout occurs.

Example Data Flow Diagrams Continued

Why Class 2 Will Not Solve The Problem

Class 2 Error Case

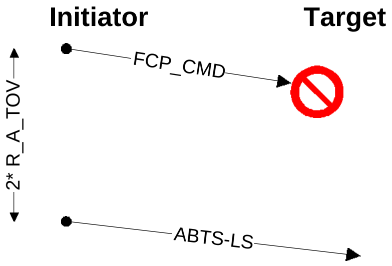


Host Follows Normal ABTS-LS
Error Recovery from here

However,
Don't Know if Command Arrived

Report Error to ULP
Device and Host out of Sync

Class 2 Error Case



Host Follows Normal ABTS-LS
Error Recovery from here

However,
Don't Know if Command Arrived

Report Error to ULP
Device and Host out of Sync

Brian R. Smith

Crossroads Systems, Inc.

April 30, 1997