# Class 3 Error Detection and Recovery for Sequential Access Devices
# Preliminary ANSI T10 Working Document 97-189R22

## 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
- Don't turn fiber transport errors into tape drive recovery
- Optimize for single sequence errors
- Don't add inefficiencies for multiple-sequence errors

## 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 changed 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.

We have identified several functions providing some of the needed functionality, these are listed below, along with some deficiencies we have identified in each of the existing mechanisms. We are proposing additional ELS functions to provide the required functionality.

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 an 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 RA_TOV after the exchange is closed.

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

In view of these difficulties, we are proposing the addition of the Read Exchange Concise (REC), a new ELS which returns all of the required information, without including superfluous information at a frame level. In light of this, our proposed solution assumes that the REC becomes part of the standard.

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.

Additionally, there needs to be a mechanism for requesting retransmission of sequences that were not received at the destination. We are proposing the addition of the Sequence Retransmission Request (SRR), a new ELS which provides information to the sequence initiator about which sequences were not received by the sequence recipient.

## Proposed Solution

A method is proposed where the initiator determines the state of an exchange and initiates appropriate sequence level recovery. A timer is used in conjunction with internal driver state information to determine if a target response is overdue, indicating that packet information ma~~a~~y have been lost. The initiator will then request exchange and sequence state information from the target~~,~~ from which it can be determined if corrective action is necessary. The initiator can then resend sequence information, request that the target resend sequence information , or provide early indication to the ULP that an error has occurred.

The timer is based on the maximum frame propagation delivery time through the fabric. This is significantly less than typical ULP time out values, providing the capability to detect and correct errors before ULP actions take effect. The suggested time out for FLA environments is twice R_A_TOV (~~, which is~~ currently 2 seconds).

The suggested method of determining target sequence state is by using the REC extended link service. The target device must maintain transmitted sequence status information as well as received sequence information. The REC ELS command is described later in this document.

Details of the recovery mechanism are as follows:

After (2 x R_A_TOV) with no reply sequence received to the FCP_CMND_IU:

Issue REC~~S~~ for the exchange containing the FCP_CMND. The REC is issued in a new exchange. If there is no ACC or LS_RJT response to the REC within 2*R_A_TOV, send ABTS to abort the exchange containing the REC. The REC shall (optionally?) be retried at a rate not to exceed once per 2*R_A_TOV for at least 3 times. If none of the RECs receives a response, the initiator shall report an error condition to the ULP.

If the response is an LS_RJT, with a reason code indicating that the function is not supported, as is required in PLDA for block devices, treat the target as a disk or other device not supporting this proposal and allow normal ULP recovery to occur.

If the FCP_CMND was not received by the target (i.e., the initiator receives an LS_RJT for the REC, 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).

The target shall retain ESB information for 3~~2~~*R_A_TOV 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 the ACC for an REC indicates that the FCP_CMND was received by the target, and that no reply sequence has been sent, the command is in process and no recovery is needed at this time. At intervals of 2*R_A_TOV the REC shall be retransmitted. This is to ensure that no reply sequences have been lost. If at any time, there is no reply to the REC, an ABTS is sent for the REC, and it is retried as specified above.

If the ACC for an REC indicates that 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. The target retransmits the FCP_XFER_RDY, with F_CTL bit 9 set, indicating that this is a retransmitted frame. 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.

The target shall return information about the status of sequences sent, as well as the status of sequences received. This is optional behavior in FC-PH for the RES Link Service, and would need to be made mandatory in FC-PLDA if RES were used instead of REC. Additionally, sequence-id usage in Class 3 must be specified in such a way that the target does not reuse sequence-ids within an exchange.

If an ACC for and REC indicates that an FCP_DATA sequence was sent by the target, but not successfully received by the initiator, issue an SRR Extended Link Service frame to request retransmission of the sequence that was not successfully received. The target retransmits the FCP_DATA sequence, with F_CTL bit 9 set in each data frame, indicating that this is a retransmitted frame. The received data is delivered to the ULP, and no error is reported. Note that the sequence that was received in error is not delivered to the ULP. If the target responds to the SRR with an LS_RJT and a reason code indicating that the function could not be performed, the target shall present an FCP_RSP IU with an appropriate error status (e.g., Sense key 4, ASC/ASQ of 48/00 (initiator detected error)).

If an ACC for an REC indicates that 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 target retransmits the FCP_RSP sequence, with F_CTL bit 9 set, indicating that this is a retransmitted frame. 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.

Note that detecting a lost sequence is unambiguous, as a lost sequence followed by a successfully received sequence would not have a continuously incrementing sequence count, as required for streamed sequences (i.e., in the case of an FCP_RSP following the last FCP_DATA sequence, or in the case of multiple FCP_DATA sequences). Thus the SEQ_ID could not have been reused.

If the ACC for an REC indicates that an FCP_DATA sequence was sent by the initiator, but not successfully received by the target, the initiator sends an RSI Extended Link Service to request sequence initiative. As documented in PLDA Sec. 9.2, the target discards the sequence in error, but does not initiate any recovery action. When the ACC is received for the RSI, the data sequence is retransmitted with F_CTL bit 9 set in each frame, indicating that this is a retransmitted frame. None of the sequence that was not completely received is delivered to the ULP. The operation should complete with no error indication to the ULP.

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 REC and other internal state. As described in PLDA, the target does not initiate recovery action.

Note that link recovery should be treated as the equivalent of a bus reset. All open exchanges will be terminated and a unit attention condition shall be generated.

**SRR Basic 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.

In the event that the target cannot accept this request, the target shall present a check condition as if it had not responded to an Initiator Detected Error with a Restore Pointers message (i.e., Sense Key = 4, ASC/ASQ = 48/00). The target shall not reject requests for retransmission of FCP_XFER_RDY or FCP_RSP frames unless the SRR is not supported.

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<br>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 | Can't resend last sequence |
| Reserved | |

**SRR LS_RJT Reason Codes**

## Read Exchange Concise (REC)

The REC Links Service request Sequence requests an N_Port to return information on completed sequences for the RX_ID or OX_ID originated by the S_ID specified in the Payload of the request Sequence. The specification of OX_ID and RX_ID may be useful or required information for the destination N_Port to locate the status information requested. A Responder destination N_Port would use the RX_ID and ignore the OX_ID, unless the RX_ID was undetermined (i.e., RX_ID = 0xffff). An Originator N_Port would use the OX_ID and ignore the RX_ID. This function provides the N_Port transmitting the request with information regarding the current status of the Exchange specified.

If the ~~-~~destination N_Port of the RES request determines that the SEQ_ID, Originator S_ID , OX_ID, or RX_ID ~~-~~are inconsistent, then it shall reply with an LS_RJT Sequence with a reason code that it is unable to perform ~~-~~the command request.

## Protocol:

Read ~~-~~Exchange Concise request Sequence
Accept ~~-~~(ACC) reply Sequence

## Format: ~~F~~FT_1

## Addressin~~g~~g:

The S_ID ~~–~~field designates the source N_Port requesting the Exchange information. The D_ID field designates ~~-~~the destination N_Port to which the request is being made.

## Paylo~~a~~d:

The for~~m~~mat of the Payload is shown in the following table.~~table XX.~~ The Payload shall include an Association Header for the Exchang~~e~~e if the destination N_Port requires X_ID reassignment.

| **~~Table xe~~x – REC Payload** | |
|---|---|
| Item | Size<br>-Bytes |
| Hex ~~-~~'13000000' | 4 |
| Reserv~~e~~ed | 1 |
| Originat~~o~~or S_ID | 3 |
| OX_I~~D~~D | 2 |
| RX_I~~D~~D | 2 |
| Associ~~a~~ation Header<br>(optiona~~l~~lly required) | 32 |

## Reply Link ~~-~~Service Sequence

Se~~r~~rvice Reject (LS_RJT)

Signifies ~~-~~rejection of the REC command.

ACC~~ceptt~~

S~~i~~ignifies that the N_Port has transmitted the requested data.

- Accep~~t~~t payload:
  - Th~~e~~e format of the Accept Payload is shown in the table below~~table 78~~. The format of the Concise Exc~~h~~hange Status is specified in below.~~xx.xx.xx~~

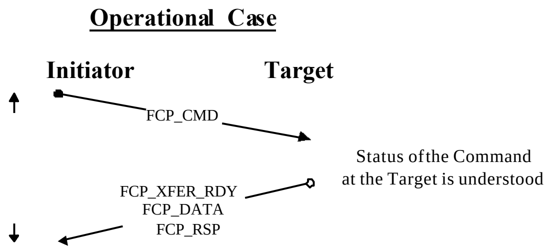
N~~o~~ote that for a sequence to be reported as received, the entire sequence must have been suc~~c~~cessfully received. For a sequence to be reported as transmitted, the entire sequence must have been -successfully transmitted.

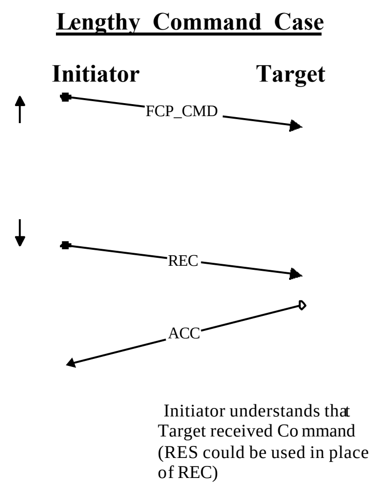
| **~~Table yy~~ REC Accept Payload** | |
|---|---|
| Ite~~m~~m | Size<br>-Bytes |
| Hex -‘02000000’ | 4 |
| Concise -Exchange Status (see 24.8.xx) | N |
| Association -Header<br>(optionally -required) | 32 |

| **~~Table zz~~ -Concise Exchange Status** | |
|---|---|
| Ite~~m~~m | Size<br>-Bytes |
| OX_~~I~~ID | 2 |
| RX_~~I~~ID | 2 |
| Originator -Address Identifier (High order byte – reserved) | 4 |
| Responder -Address Identifier (High order byte – reserved) | 4 |
| E_STA~~T~~T | 4 |
| Number -of sequences received (m) | 4 |
| Number -of sequences transmitted (n) | 4 |
| R_SEQ_ID -0 | 1 |
| : | : |
| R_SEQ_ID -m-1 | 1 |
| X_SEQ_ID -0 | 1 |
| : | : |
| X_SEQ_ID -n-1 | 1 |

## Example Data Flow Diagrams

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




**LS_RJT Codes for RES command:**

0x00051700 - Invalid Exchange

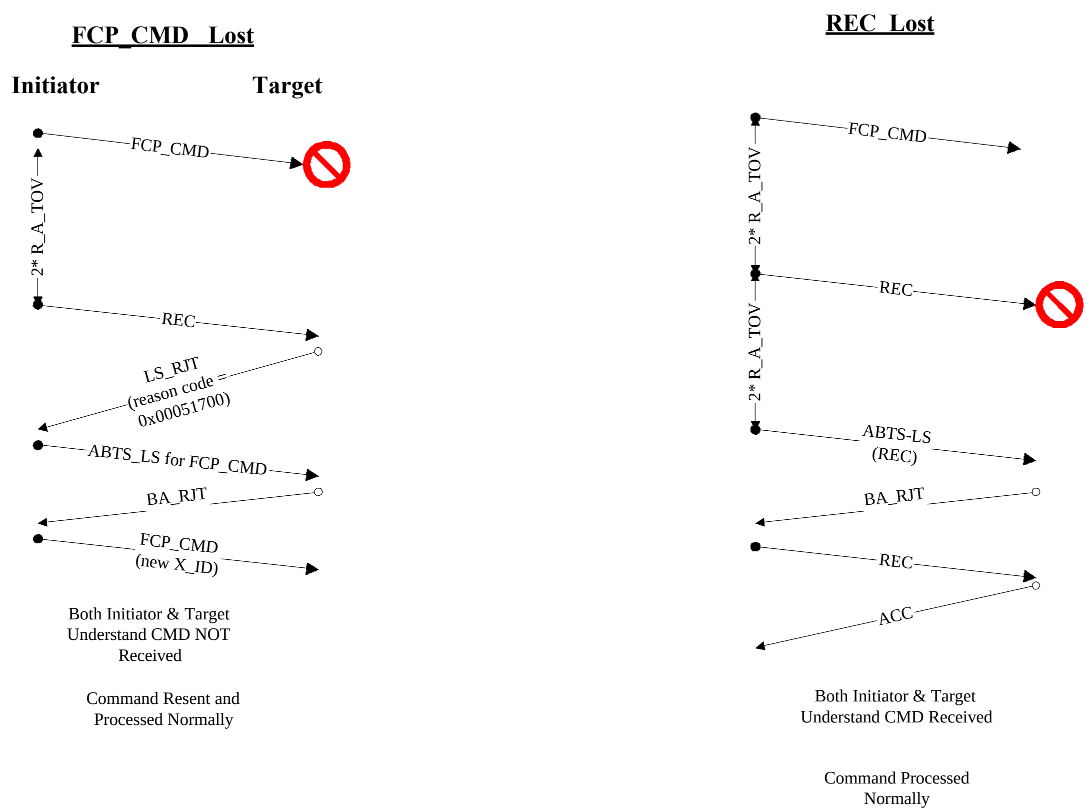
0x00052A00 - Can't provide Sequence Information

0x000B0000 - Don't Support Command

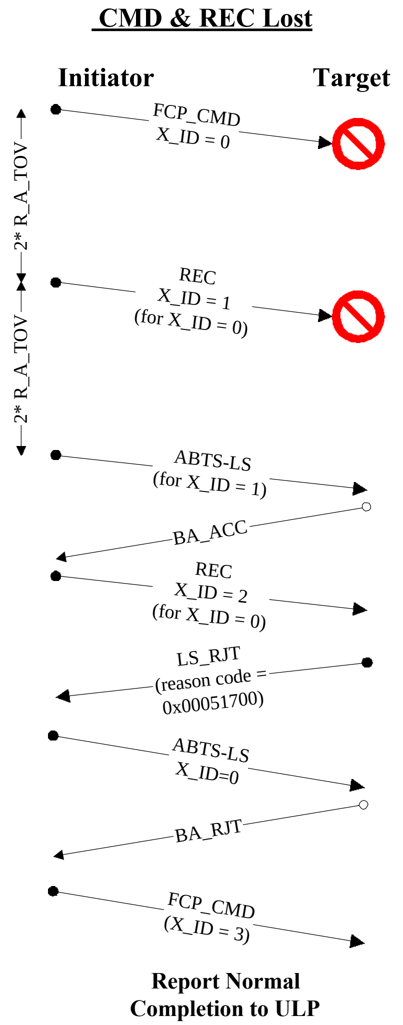
**LS_RJT Codes for SRR command:**

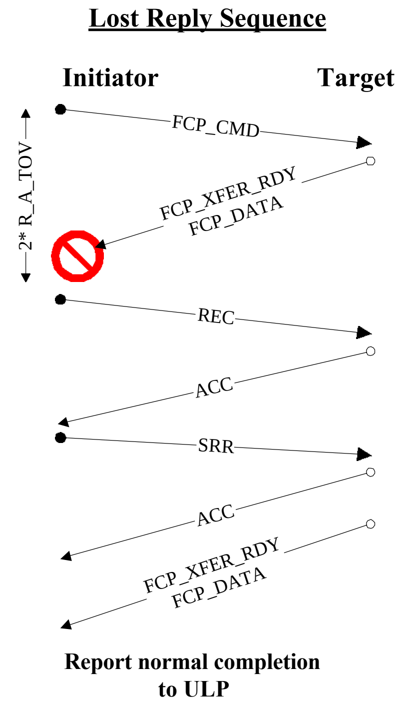
0x00052A00 - Can't resend last Sequence

**Example Data Flow Diagrams Continued**



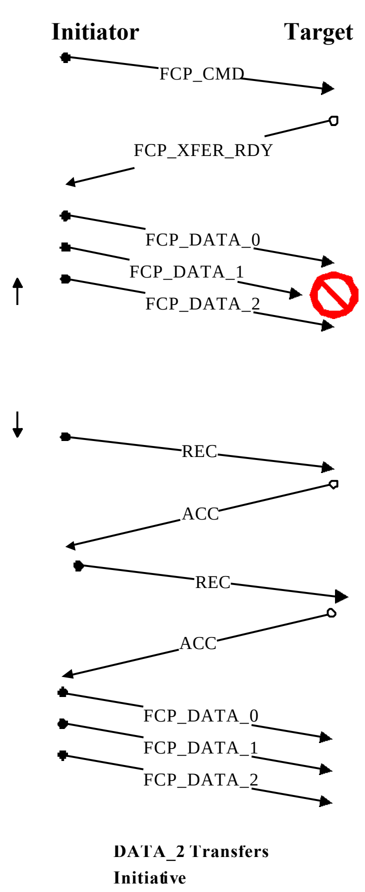
**Example Data Flow Diagrams Continued**



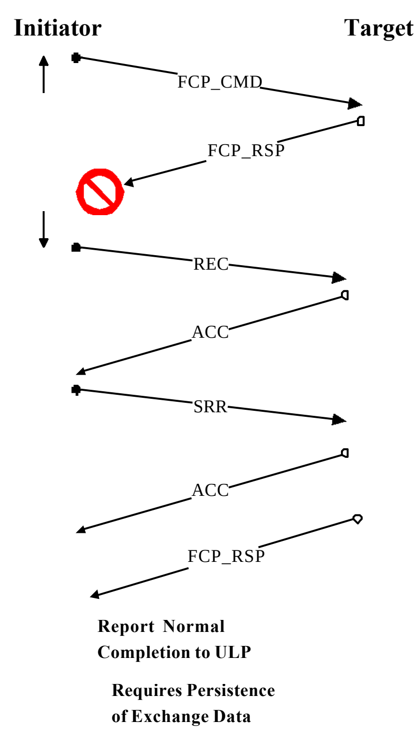



## Example Data Flow Diagrams Continued

**Lost Write Data**

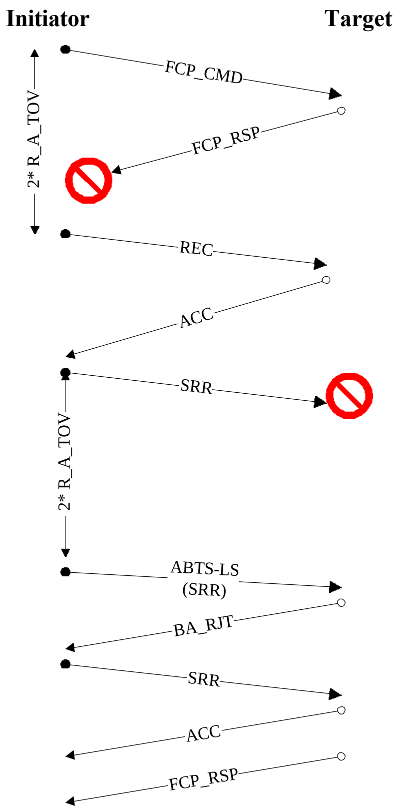



**Lost RSP**



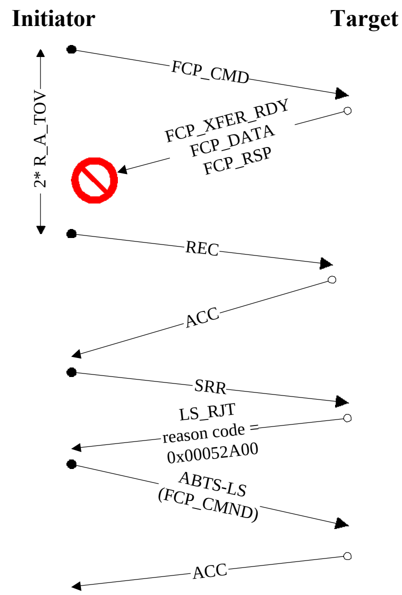
## Example Data Flow Diagrams Continued

**Lost SRR**



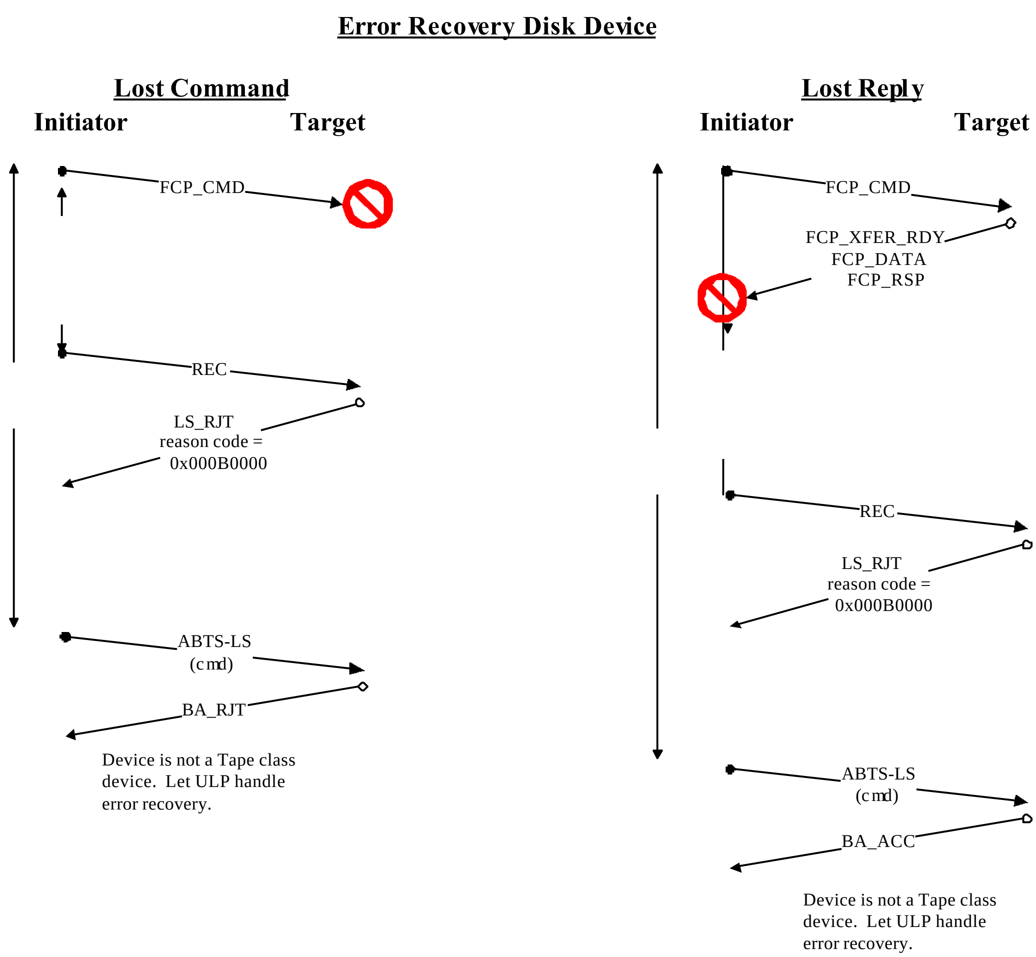

SRR is lost. Retry SRR one more time and if still unsuccessful, then abort command and notify ULP.

**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.

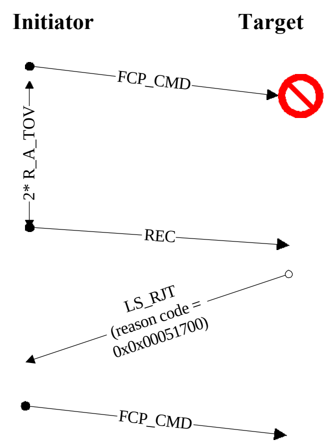
**Example Data Flow Diagrams Continued**



## Example Data Flow Diagrams Continued

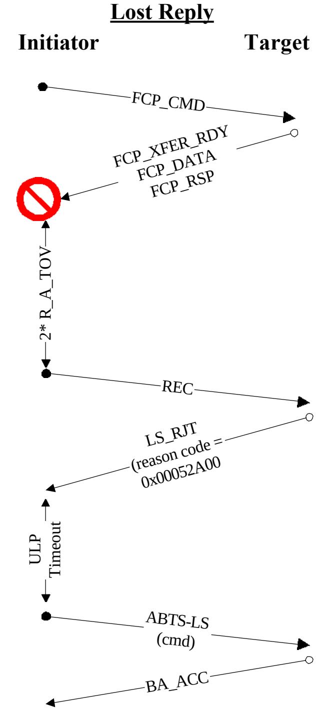
**Tape Device Lacking Functionality**

**Lost Command**




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

**Lost Reply**

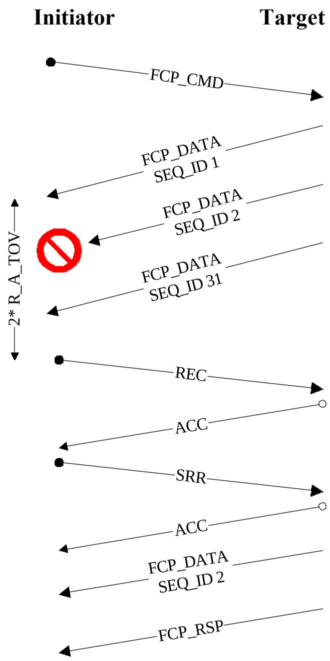



The Target does not have enough sequence information to respond to the RES. At this point the Initiator may not be able to determine the state of the command and should fall back on using the ULP timeout for error recovery. If the Initiator can verify that the command is broken then it can abort the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

## ~~Example Data Flow Diagrams Continued~~

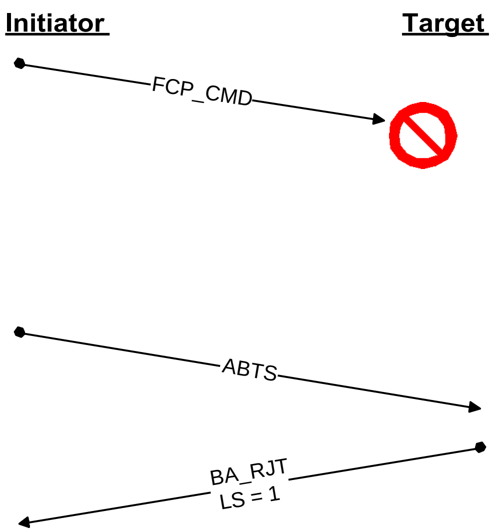
## Example Data Flow Diagrams Continued

**One Lost Data Sequence Of Multiple Data Sequences**

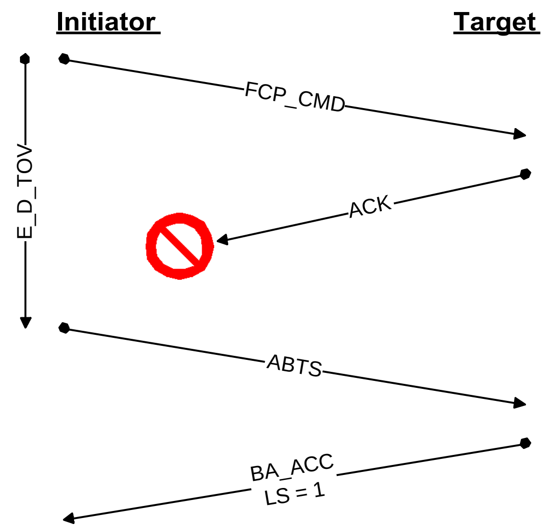



**Report Normal Completion To ULP**

## Class 2 Error Detection of FCP_CMD




At this point, the Initiator knows that the command was NOT received by the Target because the BA_RJT indicates that this FX_ID is unknown.




At this point, the Initiator knows that the command was received by the Target because the BA_ACC indicates that this FX_ID was known.