Class 3 Error Detection and Recovery for Sequential Access Devices Preliminary ANSI T10 Working Document 97-189R5c

Scope

Problems exist in PLDA in detecting and correcting error conditions on sequential access devices (tapes). 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
- Support Out-of-Order Operation

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 deficiencies we have identified in each of the existing mechanisms. We are proposing additional ELS functions to provide the required functionality.

RES (Read Exchange Status) provides some of the required functionality; 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 R_A_TOV after the exchange is closed.

A third issue is the fact that the standard only requires that the Responder to the RES need only provide information on sequences received. Provision of status on sequences transmitted is optional.

A final issue is the fact that many existing silicon implementations hide sequence information from the firmware. Provision of information about SEQ_IDs, SEQ_CNTs, etc., is not available, and SEQ_IDs may not be unique within exchanges. This means that it is not possible to provide sequence-level information sufficient to perform error recovery.

Similar arguments apply to the use of the RSS (Read Sequence Status), though the wording of the applicable section indicates that "…the Responder destination N_PORT may use RX_ID and ignore the OX_ID…"

ABTS, while recommended in FC-PH for use in polling for sequence delivery in Class 3, is always interpreted as an abort of the exchange in FC-PLDA, and is therefore not useful for this purpose. In addition, the ABTS requires the use of sequence identification and delivery information which is not made visible to firmware in some existing silicon implementations.

In view of these difficulties, we are proposing the addition of the Read Exchange Concise (REC), a new ELS which returns only the information required for error recovery.

Additionally, there needs to be a mechanism for requesting retransmission of information that was not received at the destination. We are proposing the addition of the System Retransmission Request (SRR), a new ELS which provides a mechanism for a sequence recipient to request retransmission of a missing IU. This is modified from previous proposals by requesting this on the basis of IU type and relative offset, rather than by SEQ_ID and SEQ_CNT.

Proposed Solution

A method is proposed where the initiator determines the state of an exchange and initiates appropriate recovery. A timer is used in conjunction with internal driver state information to determine if a target response is overdue, indicating that frames or sequences may have been lost. The initiator will then request exchange information from the target from which it can be determined if corrective action is necessary. The initiator may then request that the target retransmit an IU, 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 is twice R_A_TOV, (currently 2 seconds in PLDA environments (and moving to 200 usec), and 10 seconds in FLA environments).

The method of determining target sequence state is by using the REC extended link service. A target device supporting this ELS must maintain a limited amount of exchange context until the next command is received, making this appropriate for devices not supporting queueing. The REC ELS Request and Reply sequences are described later in this document.

Details of the recovery mechanism follow. For illustrative purposes, an FCP exchange is discussed; the mechanisms would apply to any FC-4, with the appropriate changes to the IU names and protocols. In order to reduce verbosity, the term “initiator” is used to denote Exchange Originator, and “target” to denote Exchange Responder.

Case 1

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

Issue REC 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, followed by an RRQ if the ABTS is ACCepted. The REC shall 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 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, followed by an RRQ if the ABTS is ACCepted. Resend the command, using a new OX_ID.

The target shall retain exchange information until the next command has been received. 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, followed by an RRQ if the ABTS is ACCepted, and the REC is retried as specified above.

If the ACC for an REC indicates that an FCP_XFER_RDY was sent by the target (by indicating that the initiator holds sequence initiative, and that the exchange is not complete), but not received by the initiator, issue an SRR Extended Link Service (see below for details) frame to request retransmission of the FCP_XFER_RDY (R_CTL = data descriptor). The target retransmits the FCP_XFER_RDY in a new sequence, and containing the appropriate Relative Offset. 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 error recovery as documented in PLDA section 9.1, 9.3.

If an ACC for an REC indicates that an FCP_RSP sequence was sent by the target, but not received by the initiator (E_STAT indicates the exchange is complete), issue an SRR Extended Link Service frame to request retransmission of the FCP_RSP IU. The target retransmits the FCP_RSP in a new sequence. The response is delivered to the ULP, and no error is reported. If the SRR receives a LS_RJT, perform error recovery as documented in PLDA section 9.1, 9.3.

Case 2:

If the ACC for an REC indicates that an FCP_DATA sequence was sent by the initiator, but not received by the target (the data received count in the REC response is smaller than the initiator sent, and the target indicates that he does not hold sequence initiative), the responder issues an SRR Extended Link Service frame to request retransmission of an FCP_XFER_RDY to request the missing data. As documented in PLDA Sec. 9.2, the target discards the sequence in error, but does not initiate any recovery action. After transmitting the ACC for the SRR, the target transmits an FCP_XFER_RDY with the appropriate Relative Offset parameter, and the initiator responds with the requested data. The operation should complete with no error indication to the ULP. No ABTS is required for the missing data, as it cannot be received by the target after the REC, as R_A_TOV has expired since the data was transmitted, and the fabric can no longer deliver it. In addition, the ABTS is interpreted as an Abort Tag Task Management function, and would terminate the operation.

Case 3:

If a read-type command completes with a data count smaller than the CDB indicated:

If the command completes with an FCP_RSP frame indicating no data underrun (i.e., FCP_RESID_UNDER set in the FCP_STATUS field of the FCP_RSP), but the received data count is not correct for the operation, the initiator shall initiate an REC to determine how much data was sent by the target. If the ACC for the REC indicates that data was sent by the target, but not successfully received by the initiator (by indicating a data sent count greater than the initiator has successfully received), the initiator issues an SRR Extended Link Service frame to request retransmission of the FCP_DATA (R_CTL = Solicited Data) that was not successfully received. The initiator passes the Relative Offset of the next data requested. The target retransmits the data in a new sequence. The received data is delivered to the ULP, and no error is reported 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 message received)). See the diagram “Lost Read Data Relative Offset Recovery” for an illustration of this procedure.

If the command completes with an FCP_RSP frame which indicates that there is a data underrun condition (i.e., that the target did not transmit all of the data requested by the CDB), the initiator shall issue an REC and compare the quantity of data transmitted against the quantity of data received. If these are the same, no recovery is required. See the diagram “Underrun Condition, No Err” for an illustration of this. If they differ, perform recovery for lost read data as indicated in the previous paragraph. See the diagram “Lost Read Data, Underrun Indication” for an illustration.

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.

SRR Extended Link Service

The SRR (System Retransmission Request) Extended Link Service request Sequence requests an N_Port to retransmit information for the RX_ID or OX_ID originated by the S_ID originating 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 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 the ability to request that information not correctly received by either the exchange originator or exchange recipient to be retransmitted so that the exchange may be completed normally.

If the destination N_Port of the SRR request determines that the 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:

System Retransmission request Sequence
Accept (ACC) reply Sequence

Format: FT_1

Addressing:

The S_ID field designates the source N_Port requesting the information retransmission. The D_ID field designates the destination N_Port to which the request is being made.

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.

Payload:

The format of the Payload is shown in the following table. The Payload shall include an Association Header for the Exchange if the destination N_Port requires X_ID reassignment. The amount of data to transfer is implicitly the remainder of that for the exchange.

SRR Payload	
Item:	Size (bytes):
OX_ID	2
RX_ID	2
Relative Offset	4

R_CTL for IU	1
Reserved	3

The R_CTL encoding is as in FC-PH section 18.2, Tables 28, 29; i.e., 06 = Unsolicited Command, 05 = Data Descriptor (FCP_XFER_RDY), 07 = Command Status, 01 = Solicited Data.

Reply Link Service Sequence

Service Reject (LS_RJT)

Signifies rejection of the SRR request.

ACC

Signifies that the information will be retransmitted.

- Accept payload:
  - The accept has no payload.

The new SRR reject reason code is defined below.

Encoded Value	**LS_RJT Reason code explanation**
0x00052A00	Can’t resend requested sequence
Reserved	

SRR LS_RJT Reason Codes

Read Exchange Concise (REC)

The REC Extended Link Service request Sequence requests an N_Port to return exchange information 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). If the RX_ID is unspecified in the request, the target must identify the exchange by means of the S_ID and OX_ID. 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 REC request determines that the 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: FT_1

Addressing:

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.

Payload:

The format of the Payload is shown in the following table. The Payload shall include an Association Header for the Exchange if the destination N_Port requires X_ID reassignment.

REC Payload	
Item	Size -Bytes
Hex ‘13000000’	4
Reserved	1
Exchange Originator S_ID	3
OX_ID	2
RX_ID	2
Association Header (optionally required)	32

Reply Link Service Sequence

Service Reject (LS_RJT)

Signifies rejection of the REC command.

ACC

Signifies that the N_Port has transmitted the requested data.

- Accept payload:
  - The format of the Accept Payload is shown in the table below. The format of the Concise Exchange Status is specified below.

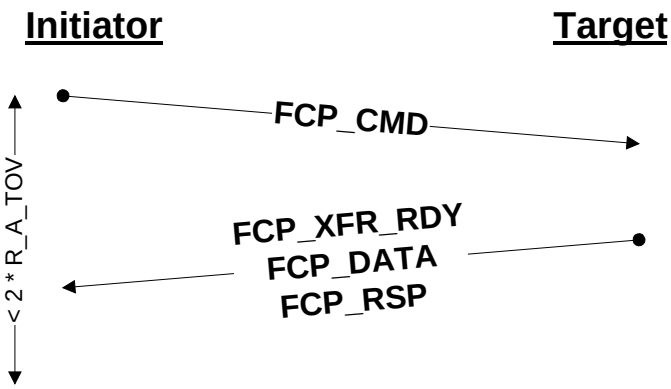
REC Accept Payload	
Item	Size -Bytes
Hex '02000000'	4
Concise Exchange Status (see 24.8.xx)	N
Association Header (optionally required)	32

Concise Exchange Status	
Item	Size -Bytes
OX_ID	2
RX_ID	2
Originator Address Identifier (High order byte – reserved)	4
Responder Address Identifier (High order byte – reserved)	4
Data transfer count	4
E_STAT	4

E_STAT is as defined in FC-PH Sec. 24.8.1. for the Exchange Status Block.

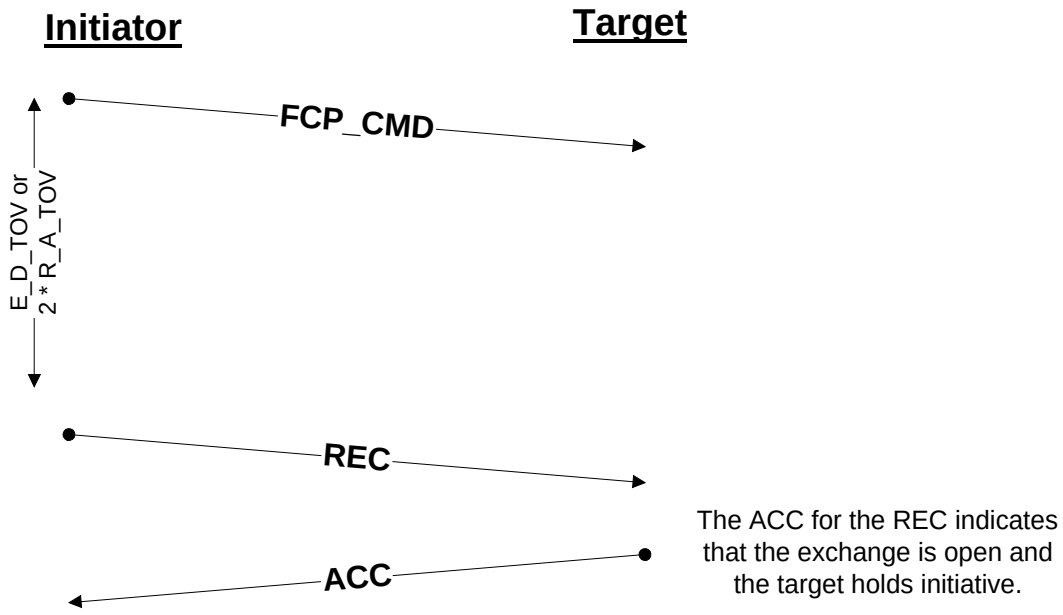
Class 3 Operation for Tape Devices on FC-AL using FCP

Operational Case



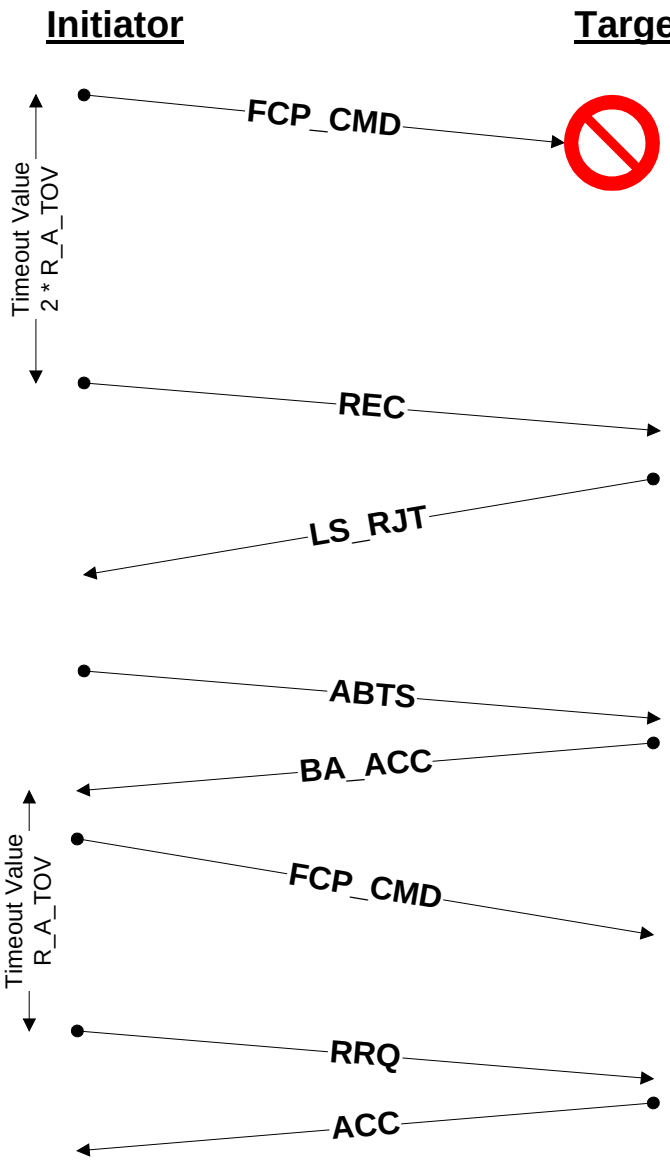
In this case, some response is received before the timeout period, so no probes are done.

Lengthy Command Case



Initiator understands that the Target received the FCP_CMD

FCP CMD Lost



The LS_RJT indicates the exchange is unknown to the target.

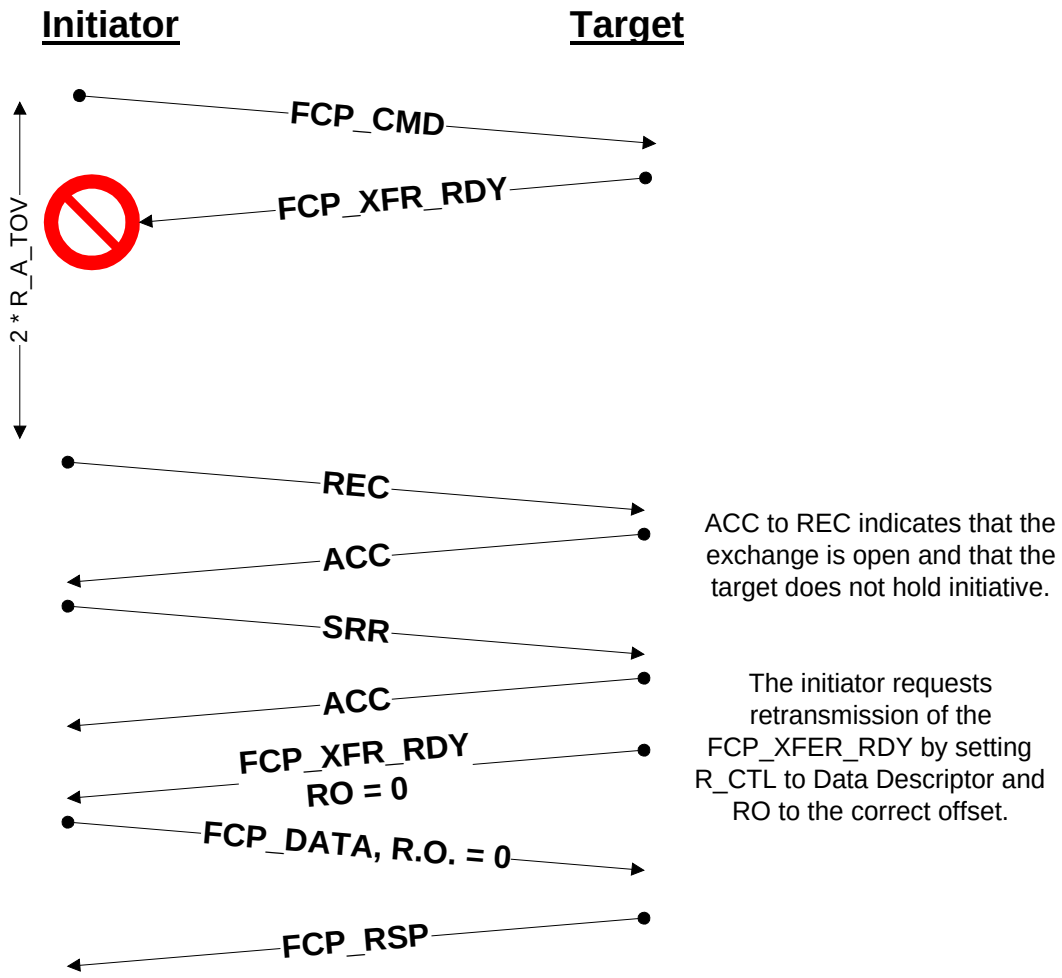
The ABTS is for the FCP_CMD, and is sent as part of the original exchange.

The BA_ACC occurs as specified in PLDA section 9.3.2

The command may now be retransmitted, in a new exchange (the old is still blocked by the recovery qualifier.

After the BA_ACC to the ABTS, and the expiration of RA_TOV, we need to restore the recovery qualifier.

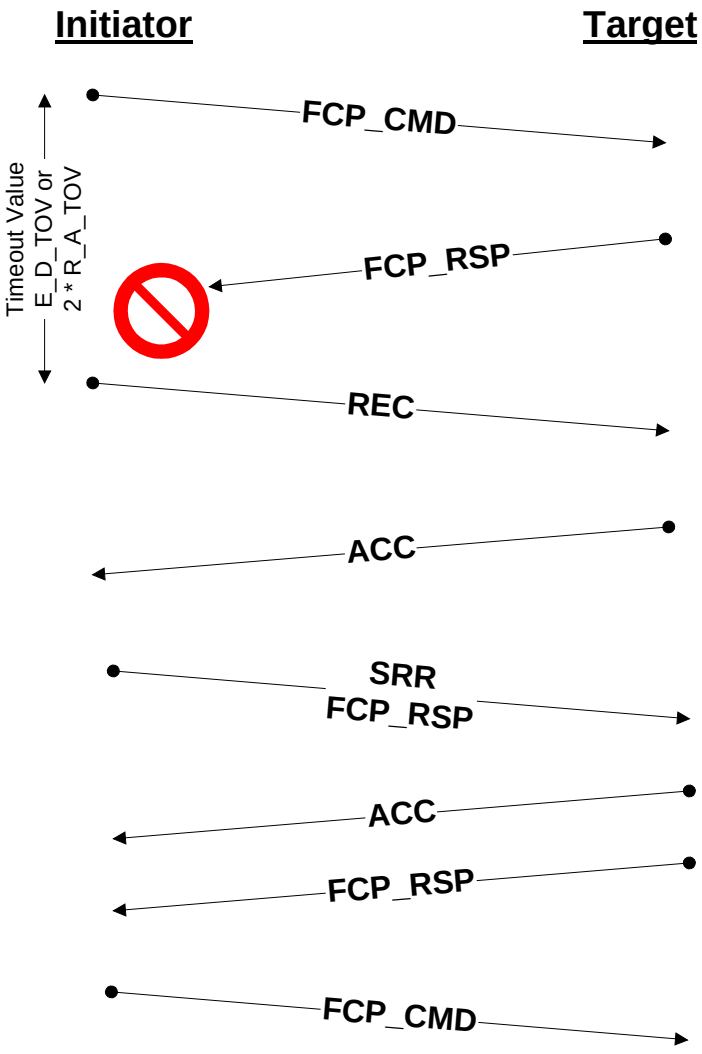
Lost FCP_XFER_RDY



ACC to REC indicates that the exchange is open and that the target does not hold initiative.

The initiator requests retransmission of the FCP_XFER_RDY by setting R_CTL to Data Descriptor and RO to the correct offset.

FCP_RSP Lost

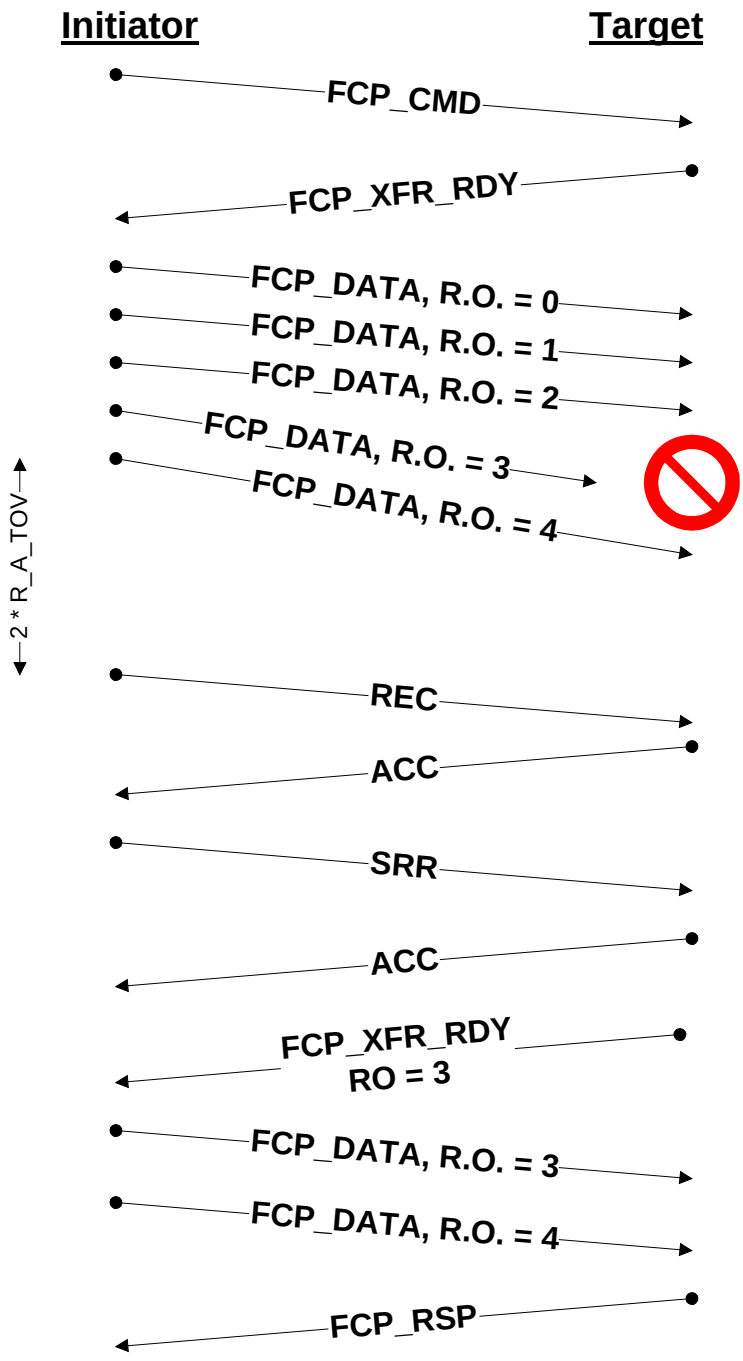


The ACC indicates that the target does not hold initiative, and that the exchange was completed (E_STAT bits). The target has retained exchange status and response data since no new command has been received.

The initiator requests the FCP_RSP be retransmitted by setting R_CTL to be Command Status.

At this point, the information about the prior exchange is discarded.

Lost Write Data - Relative Offset Recovery



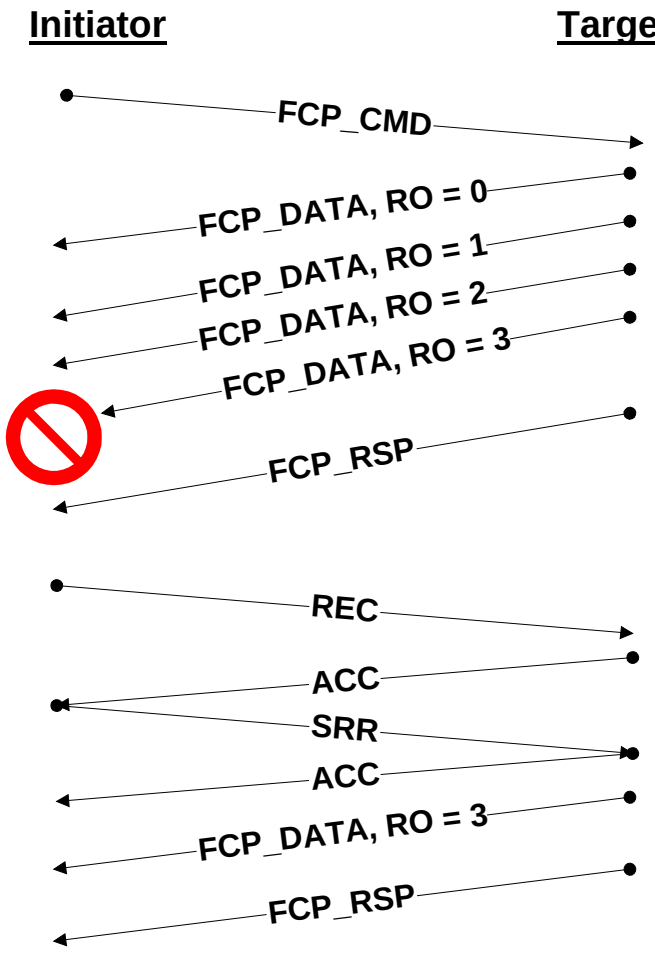
This data is discarded as the SEQ_CNT is incorrect.

The ACC indicates the amount of data received was less than the amount transmitted. No ABTS is required, as the missing frames can't pop out of fabric any more.

The initiator requests retransmission of an FCP_XFER_RDY by setting R_CTL to Data Descriptor and the correct relative offset to match the unreceived data.

In response to the FCP_XFER_RDY, the initiator retransmits the data that was not received by the target.

Lost Read Data - Relative Offset Recovery

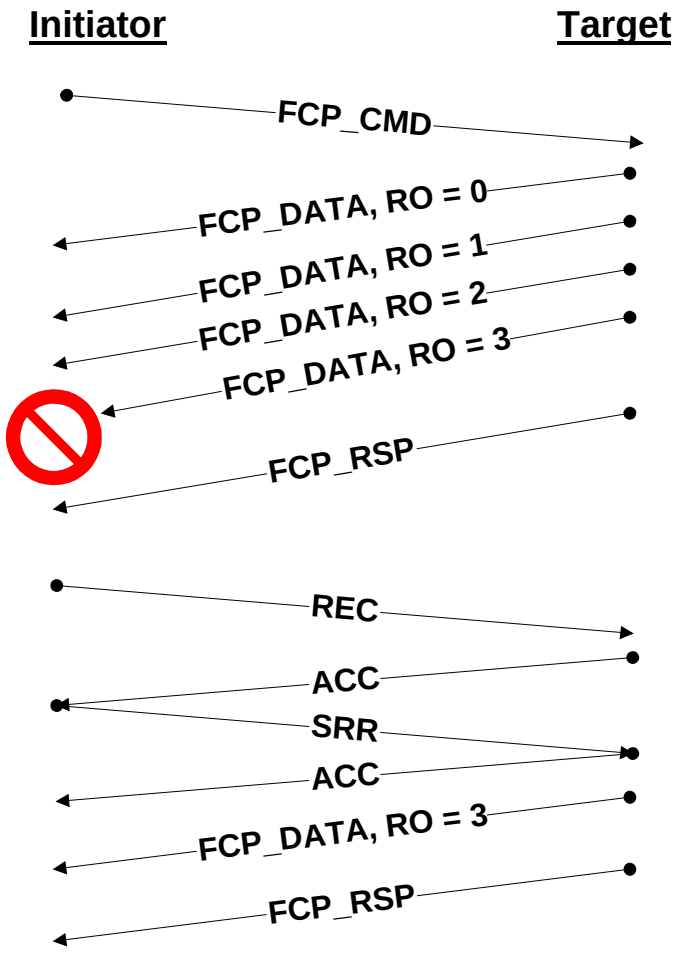


Response frame indicates no data underrun. Transfer count does not match CDB.

ACC indicates that all data was transmitted by target.

Initiator requests retransmission of missing data by setting R_CTL to Solicited Data and relative offset to that of the unreceived data.

Lost Read Data - Underrun Indication

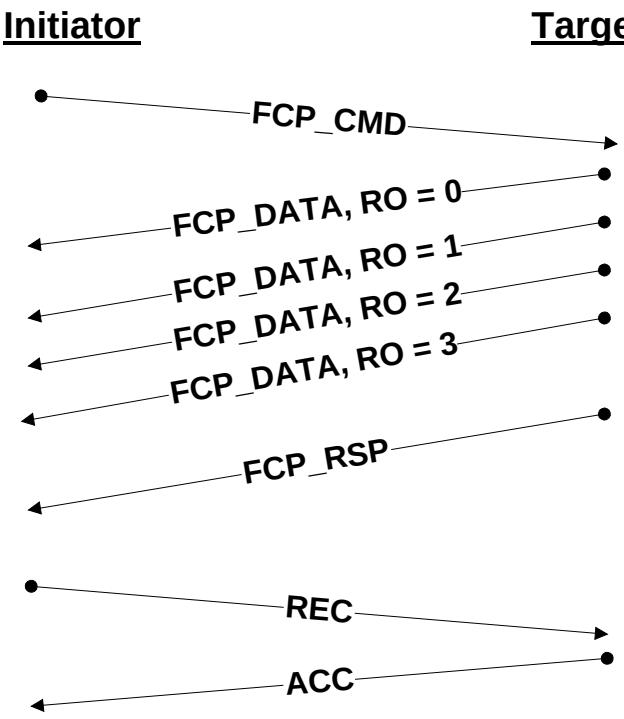


Response frame indicates data underrun occurred.

ACC indicates the quantity of data that was actually transmitted.

Initiator requests retransmission of missing data by setting R_CTL to Solicited Data and relative offset to that of the unreceived data.

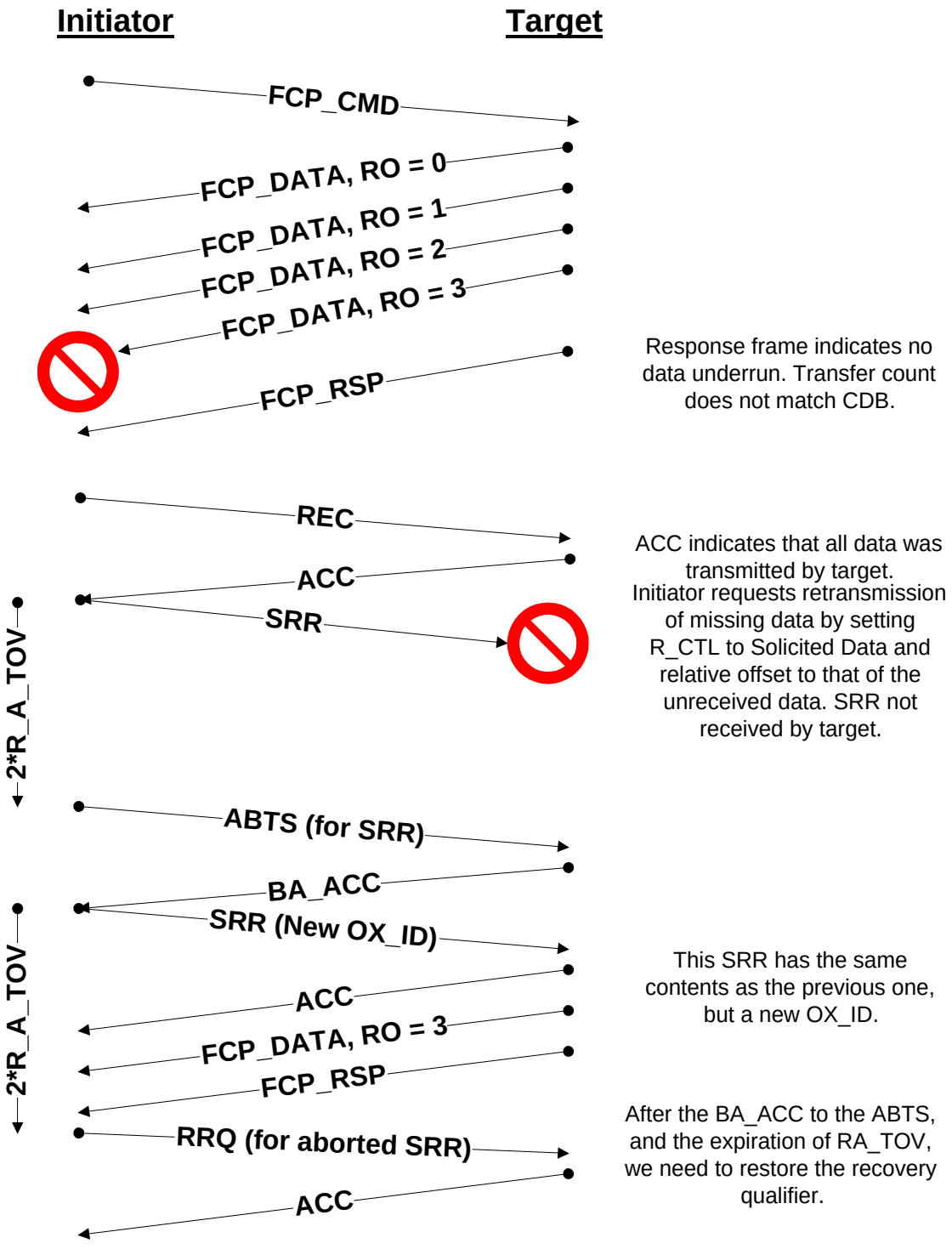
Underrun Indication, no err



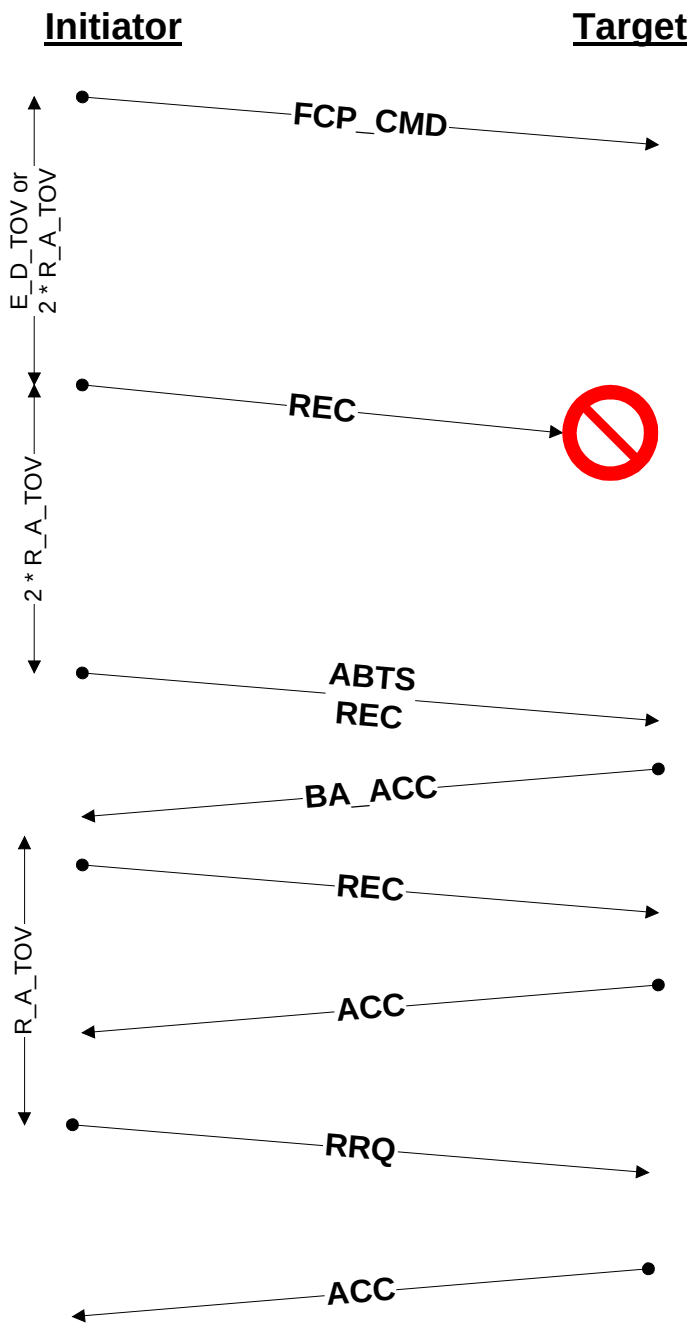
Response frame indicates data underrun occurred.

ACC indicates the amount of data that was transferred. As the quantity of data received matches the quantity transmitted, there is no error. No recovery is required.

Lost SRR After Lost Read Data



REC Lost



The REC needs to be aborted. The ABTS will get a BA_ACC, as RX_ID = 0xffff.

This ACC indicates that the exchange is open, and that the target holds initiative. No recovery is needed.

The BA_ACC to the ABTS established a recovery qualifier; after RA_TOV the recovery qualifier must be reinstated.