Annex D  Error detection and recovery action examples
(Informative) [Draft, based on T10/00-137r5]

D.1 Introduction
This annex diagrams various error detection and recovery procedures for SCSI devices conforming to this profile.

<table>
<thead>
<tr>
<th>Drawing Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>- - - - - - - - - -</td>
<td>Acknowledged or Unacknowledged Frame</td>
</tr>
<tr>
<td>- - - - - - - - - -</td>
<td>Acknowledgement Frame</td>
</tr>
<tr>
<td>_ _ _ _ _ _</td>
<td>Time-out value exceeded, caused transmission of IU or ELS</td>
</tr>
<tr>
<td></td>
<td>IU or ELS received is processed to transmit IU or ELS</td>
</tr>
<tr>
<td></td>
<td>Frame lost or dropped</td>
</tr>
<tr>
<td>X</td>
<td>Error detection complete. Operation continues with specified Error Recovery.</td>
</tr>
</tbody>
</table>

Table D.1 - Diagram Drawing Conventions
The REC ACC indicates the Exchange is open and the target holds Sequence Initiative. No error recovery is required.
Upon receipt of the LS_RJT for the REC (indicates the Exchange is unknown) the initiator performs error recovery.

The FCP_CMND is retransmitted in a new Exchange using the same CRN.

The initiator establishes a Recovery Qualifier. The value of R_A_TOV* for in-order topologies is 0.
Upon expiration of E_D_TOV, the initiator performs error recovery.

The use of REC to determine status for error recovery shall not be used.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
Upon expiration of E_D_TOV, the initiator performs error recovery. Although it is possible for the Exchange to continue when the ACK is lost, in the interest of simplicity, error recovery is performed.

The use of REC to determine status for error recovery shall not be used.

It was not decided how to proceed if the exchange “continues” or “completes” before the E_D_TOV expires.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
Figure D.5 - FCP_XFER_RDY Lost, Unacknowledged Classes

Error Detection

The REC ACC indicates the initiator holds Sequence Initiative and the Exchange is open.

Wait REC_TOV* before performing error recovery. If FCP_XFER_RDY is received before REC_TOV* expires, continue with the Exchange (REC ACC arrived before FCP_XFER_RDY, out of order). Otherwise continue recovery. For in-order topologies, the value of REC_TOV* is 0.

Error Recovery

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
After receiving the ABTS, the initiator performs error recovery. Does the initiator send BA_ACC or go directly to recovery and send ABTS(retry)?

The use of REC to determine status for error recovery shall not be used.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
Although it is possible for the Exchange to continue when the ACK is lost, in the interest of simplicity, error recovery is performed when the initiator receives the ABTS.

It was not decided how to proceed when the target receives the FCP_DATA before the E_D_TOV expires (initiator does not know ACK was lost and sends the FCP_DATA). Will the target send an ACK for the data? Will it send FCP_RSP? Does the initiator send BA_ACC or go directly to recovery and send ABTS(retry)? If BA_ACC is sent, target might send FCP_RSP.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
The REC ACC indicates the initiator holds Sequence Initiative and the Exchange is complete. The target must keep the context of this Exchange until the OX_ID value is reused in a new command (implicitly validating the receipt of FCP_RSP), or for at least RR_TOV in order to preserve the FCP_RSP information. This long time-out can be avoided by using FCP_CONF.

Wait REC_TOV* before performing error recovery. If FCP_RSP is received before REC_TOV* expires, continue with the Exchange (REC ACC arrived before FCP_RSP, out of order). Otherwise perform error recovery.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
After receiving the ABTS, the initiator performs error recovery. Does the initiator send BA_ACC or go directly to recovery and send ABTS(retry)?

The use of REC to determine status for error recovery shall not be used.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
After receiving the ABTS, the initiator performs error recovery. Does the initiator send BA_ACC or go directly to recovery and send ABTS(retry)?

The use of REC to determine status for error recovery shall not be used.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
Error Detection

No Error Recovery

If the OX_ID value is reused in a new command before E_D_TOV expires, this implicitly validates the initiator received the FCP_RSP (no need to send ABTS). If E_D_TOV expires, the target sends ABTS and the receipt of BA_RJT indicates the Exchange is unknown and therefore complete. No error recovery is required.

The target establishes a Recovery Qualifier. The issuance of the RRQ is optional, as no Recovery Qualifier was established by the initiator. For in-order topologies, the value of R_A_TOV* is 0.
Figure D.12 - FCP_RSP Received, ACK Lost, Acknowledged Classes, Example 2

Error Detection

If the OX_ID value is reused in a new command before E_D_TOV expires, this implicitly validates the initiator received the FCP_RSP (no need to send ABTS). If E_D_TOV expires, the target sends ABTS and the receipt of BA_RJT indicates the Exchange is unknown and therefore complete. No error recovery is required.

The target establishes a Recovery Qualifier. The issuance of the RRQ is optional, as no Recovery Qualifier was established by the initiator. For in-order topologies, the value of R_A_TOV* is 0.

For out-of-order topologies, if the initiator sent a FCP_CMND with the same OX_ID as the one in the received ABTS, the initiator takes no action on the ABTS until the ACK to the outstanding FCP_CMND Sequence has been received or E_D_TOV* expires, allowing the analysis to take into consideration the RX_ID to eliminate ambiguity.
Figure D.13 - Lost Write Data, Last Frame of Sequence, Unacknowledged Classes

**Error Detection**

The REC ACC indicates the target does not hold Sequence Initiative and the Exchange is open. The initiator performs error recovery.

**Error Recovery**

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
Upon expiration of E_D_TOV, the initiator performs error recovery.

The use of REC to determine status for error recovery shall not be used.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
The REC ACC indicates the target does not hold Sequence Initiative and the Exchange is open. The initiator performs error recovery.

**Error Recovery**

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
Upon expiration of E_D_TOV or the receipt of ABTS, the initiator performs error recovery. If ABTS is received, does the initiator send BA_ACC or go directly to recovery and send ABTS(retry)?

The use of REC to determine status for error recovery shall not be used.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
After receiving the FCP_RSP, the initiator waits REC_TOV* before performing error recovery to wait for possible out of order FCP_DATA frames. For in-order topologies, the value of REC_TOV* is 0.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
Figure D.18 - Lost Read Data, Last Frame of Sequence,Acknowledged Classes

Error Detection

(Sequence Error Detected)
If sequence is using ACK_1 model, and sequence recipient detects sequence error due to E_D_TOV expiration, it may send ACK (abort sequence)

Init Targ

After receiving the ABTS, the initiator performs error recovery. Does the initiator send BA_ACC or go directly to recovery and send ABTS(retry)?

The use of REC to determine status for error recovery shall not be used.

Error Recovery

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
After receiving the FCP_RSP, the initiator waits REC_TOV* before performing error recovery to wait for possible out of order FCP_DATA frames. For in-order topologies, the value of REC_TOV* is 0.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
Figure D.20 - Lost Read Data, Not Last Frame of Sequence, Acknowledged Classes

Error Detection

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCP_CMND (crn=a, oxid=x)</td>
<td>Error Detection</td>
</tr>
<tr>
<td>E_D_TOV</td>
<td>If sequence error is not detected, sequence timeout will be detected with same result</td>
</tr>
<tr>
<td>FCP_DATA (seq=1, cnt=0)</td>
<td></td>
</tr>
<tr>
<td>FCP_DATA (seq=1, cnt=1)</td>
<td></td>
</tr>
<tr>
<td>ACK</td>
<td></td>
</tr>
<tr>
<td>ABTS (seq)</td>
<td></td>
</tr>
<tr>
<td>ACK</td>
<td></td>
</tr>
<tr>
<td>BA_ACC (LS=0)</td>
<td></td>
</tr>
<tr>
<td>ACK</td>
<td></td>
</tr>
<tr>
<td>Init</td>
<td>Continue</td>
</tr>
</tbody>
</table>

(Sequence Error Detected)
If sequence recipient detects sequence error due to out of order frames or E_D_TOV expiration, it may send ACK (abort sequence)

After receiving the ABTS, the initiator performs error recovery. Does the initiator send BA_ACC or go directly to recovery and send ABTS(retry)?

The use of REC to determine status for error recovery shall not be used.

Error Recovery

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABTS (retry) (oxid=x)</td>
<td>Error Recovery</td>
</tr>
<tr>
<td>ACK</td>
<td></td>
</tr>
<tr>
<td>BA_ACC (LS=1)</td>
<td></td>
</tr>
<tr>
<td>ACK</td>
<td></td>
</tr>
<tr>
<td>FCP_CMND (retry) (crn=a, oxid=y)</td>
<td></td>
</tr>
<tr>
<td>ACK</td>
<td></td>
</tr>
<tr>
<td>Init</td>
<td>Targ</td>
</tr>
</tbody>
</table>

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
Although it is possible for the exchange to continue when the ACK is lost, in the interest of simplicity, error recovery is performed when the initiator receives the ABTS. Does the initiator send BA_ACC or go directly to recovery and send ABTS(retry)? If BA_ACC is sent, target might send FCP_RSP.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of $R_{A\_TOV^*}$ for in-order topologies is 0.
Upon expiration of E_D_TOV, the initiator performs error recovery. It was not decided how to proceed if FCP_RSP is received before the E_D_TOV expires.

The use of REC to determine status for error recovery shall not be used.

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
The LS_RJT for the REC indicates the Exchange is unknown. That implicitly indicates that the initiator received FCP_RSP and sent FCP_CONF.

The context for the Exchange in the target must be preserved for another R_A_TOV to prevent possible aliasing. If FCP_CONF is received before LS_RJT received, it is accepted and the context for the Exchange can be purged. For in-order topologies, the value of R_A_TOV* = 0.
Upon expiration of E_D_TOV, the initiator performs error recovery.

The use of REC to determine status for error recovery shall not be used.

**Error Recovery**

The initiator sends an ABTS(retry) requesting the Exchange be retried. The FCP_CMND is retransmitted in a new Exchange using the same CRN.

Both initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
Upon expiration of E_D_TOV, the initiator performs error recovery.

The use of REC to determine status for error recovery shall not be used.

BA_RJT is the response to the ABTS, since no context exists for this Exchange and the ABTS was not issued on the first sequence of a new Exchange.

The initiator establishes a Recovery Qualifier. The issuance of the RRQ is optional, as no Recovery Qualifier was established by the target. For in-order topologies, the value of R_A_TOV* = 0.
Upon expiration of 2 times $R_{A\_TOV}^*$, the initiator performs error recovery.

Initiator and target establish Recovery Qualifiers. For in-order topologies, the value of $R_{A\_TOV}^*$ is 0.

The initiator re-issues the REC in a new Exchange.
Initiator and target establish Recovery Qualifiers. For in-order topologies, the value of R_A_TOV* is 0.

The initiator re-issues the REC in a new Exchange.
Upon receipt of ABTS, the initiator performs error recovery.

Initiator and target establish Recovery Qualifiers. For in-order topologies, the value of $R_A\_TOV^*$ is 0.

The initiator re-issues the REC in a new Exchange.
The BA_ACC payloads indicate that the RECs were not received by the target. The failure of two RECs issued against the same Exchange indicates a "double error" and causes all associated Exchanges to be aborted.

The initiator transmits ABTS for the original FCP exchange with Bit 0 = 0 set in the Parameter field (abort exchange).

Initiator and target establish Recovery Qualifiers. The value of R_A_TOV* for in-order topologies is 0.
Upon receipt of ABTS, the initiator performs error recovery.

**Error Recovery**

Initiator and target establish Recovery Qualifiers. For in-order topologies, the value of R_A_TOV* is 0.

**ISSUE:** If after the ACC to the original REC is received, the initiator sends another REC using the same OX_ID (it does not know that the ACK was lost), the E_D_TOV timer on the target could expires and send the ABTS against the new REC Exchange.