Ladder Diagrams for Error Recovery For FCP -2 Rev 04 Out-Of-Order Delivery- Annex D

Carl Zeitler

Compaq Computer Corporation

May 31, 2000

T10/00-137r4

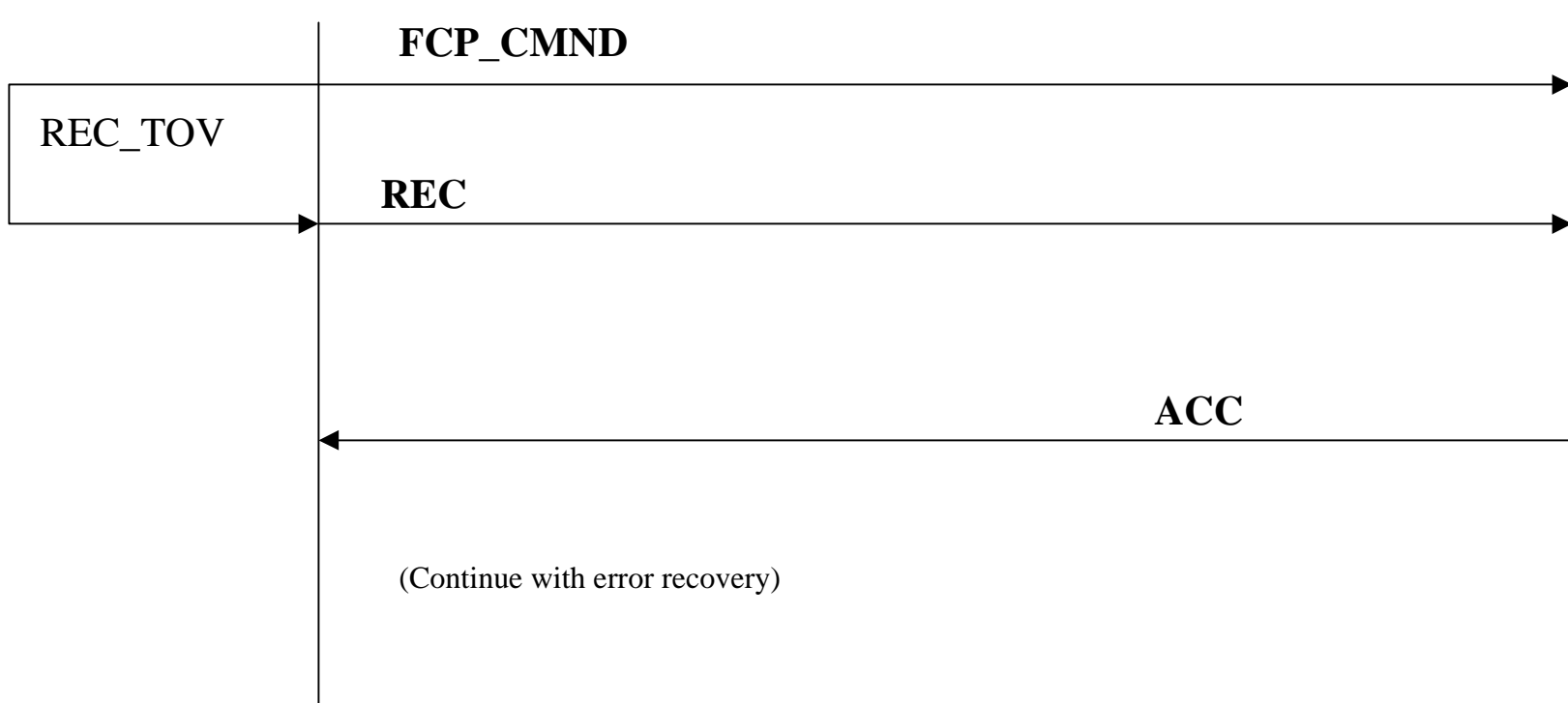Reference: T11/00-133r1, r2 and r3

Summary of Major Changes

- Made minor modifications per the May meeting to D.3, D.4, D.5, D.5a, D.6a, D.10, D.12a changed to D.12.

- D.?? Class 2 FCP_CONF Lost (Possibility 2) removed.

- Minor modifications to D.5?, D.5??

- Substituted the diagrams, D13 and D.13a, from T11/00-145v2 for REC, changing the timings from 2* R_A_TOV to R_A_TOV per discussions on R_R_TOV at the May meeting.

- Added diagram, D.13?, from 00-145v2 where 2 RECs are lost, aborting the original Exchange.

- Substituted the diagrams from T11/00-145v2 for SRR to abort the Exchange in the event of an SRR failure and to change the recovery actions where the response to SRR is lost.

- Added text to applicable Class 3 diagrams showing the difference between out-of-order vs in-order (waiting R_A_TOV before beginning error recovery vs no wait)
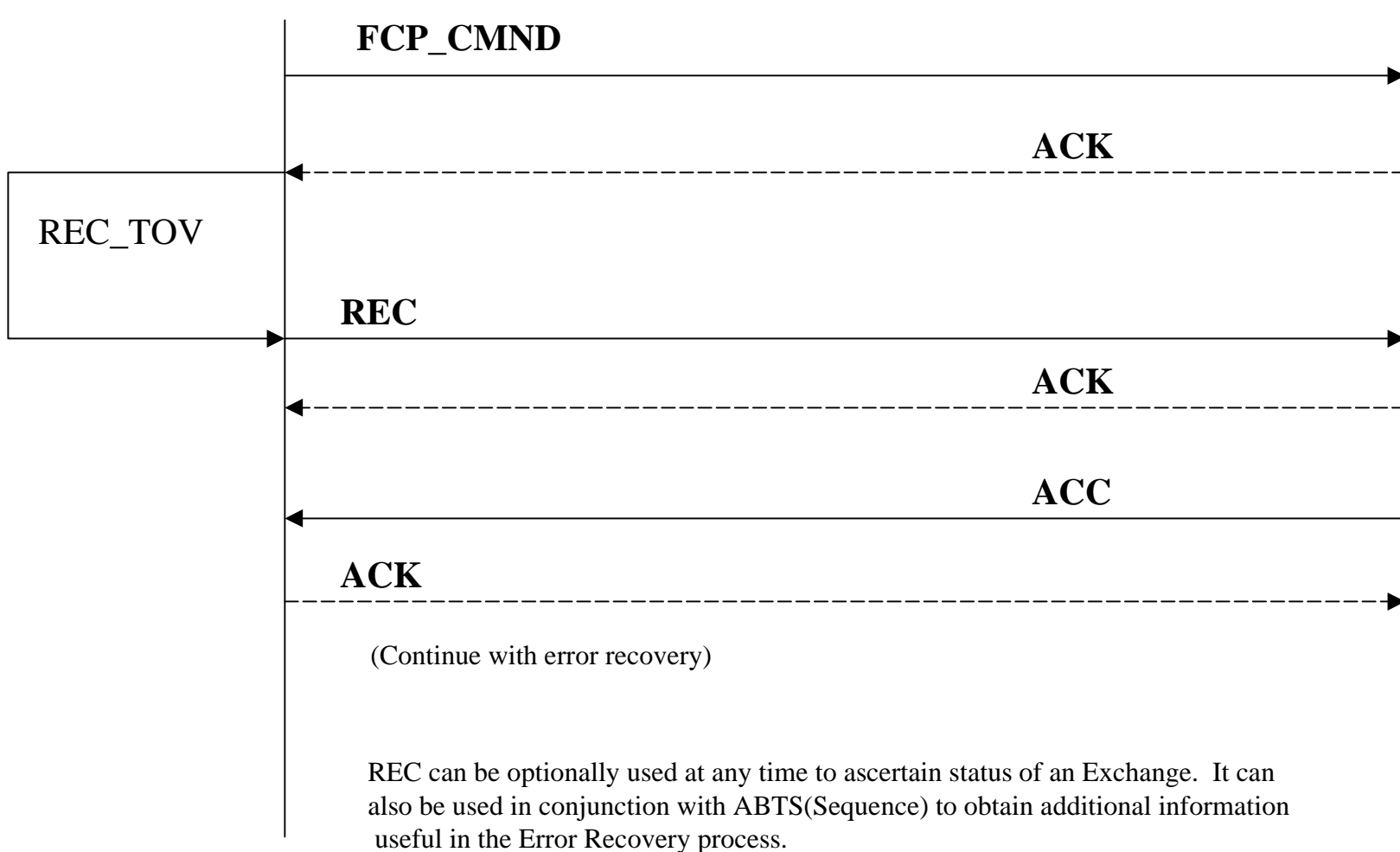
Summary of Major Changes continued

- Changed out-of-order Class 2 Diagrams to remove requirement for hacking a space out of the Sequence Count. Retransmitted Sequences can now begin with a Sequence Count of 0. The solution was to begin out-of-order recovery after the Recovery Qualifier was retired, the same way as in-order for Class 2. The recovery time is longer than necessary for Class 2, but at this point in time, commonality is more important . Recovery for Class 2 can be modified, if necessary in the future, based on the optional continuously increasing Sequence Count Bit in N_Port Common Service Parameters, to shorten the time.

- As a result, all Ladder Diagrams for in and out of order are exactly the same (or should be). The differences between in and out of order are minimal.

- Separate Annexes for in-order and out-of-order are out of order (pardon the pun) if there is agreement.

D.1 Class 3 Error Detection

FCP_CMND

REC_TOV

REC

ACC

(Continue with error recovery)

D.1 Class 2 Error Detection

FCP_CMND →

ACK ⟵

REC_TOV

REC →

ACK ⟵

ACC ⟵

ACK →

(Continue with error recovery)

REC can be optionally used at any time to ascertain status of an Exchange. It can also be used in conjunction with ABTS(Sequence) to obtain additional information useful in the Error Recovery process.

D.2 Class 3 FCP_CMD Lost

FCP_CMND

X

REC_TOV

REC

LS_RJT

(Continue with error recovery)

D.2 Class 2 FCP_CMD Lost

FCP_CMND

→ **X**

| E_D_TOV | **ABTS** |

ACK

BA_ACC

ACK

FCP_CMND*

ACK

R_A_TOV

RRQ

ACK

ACC

ACK
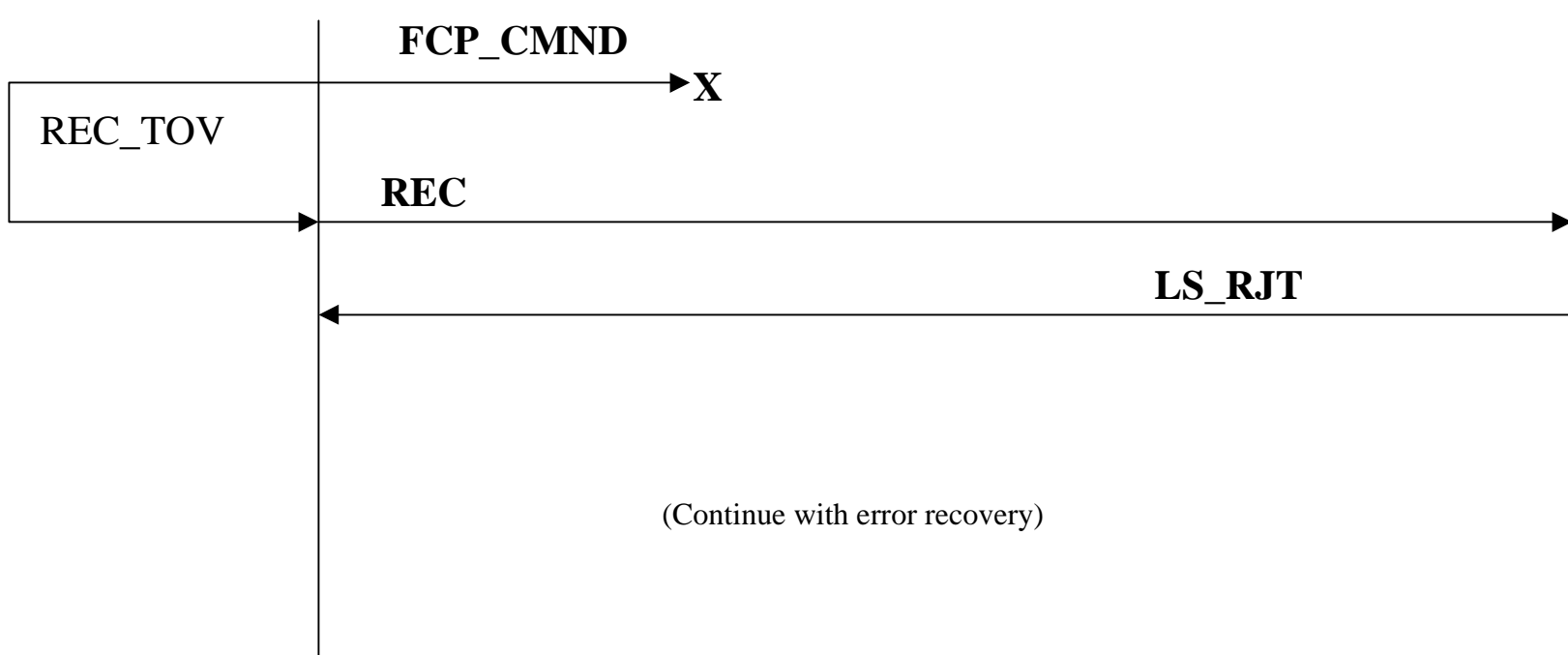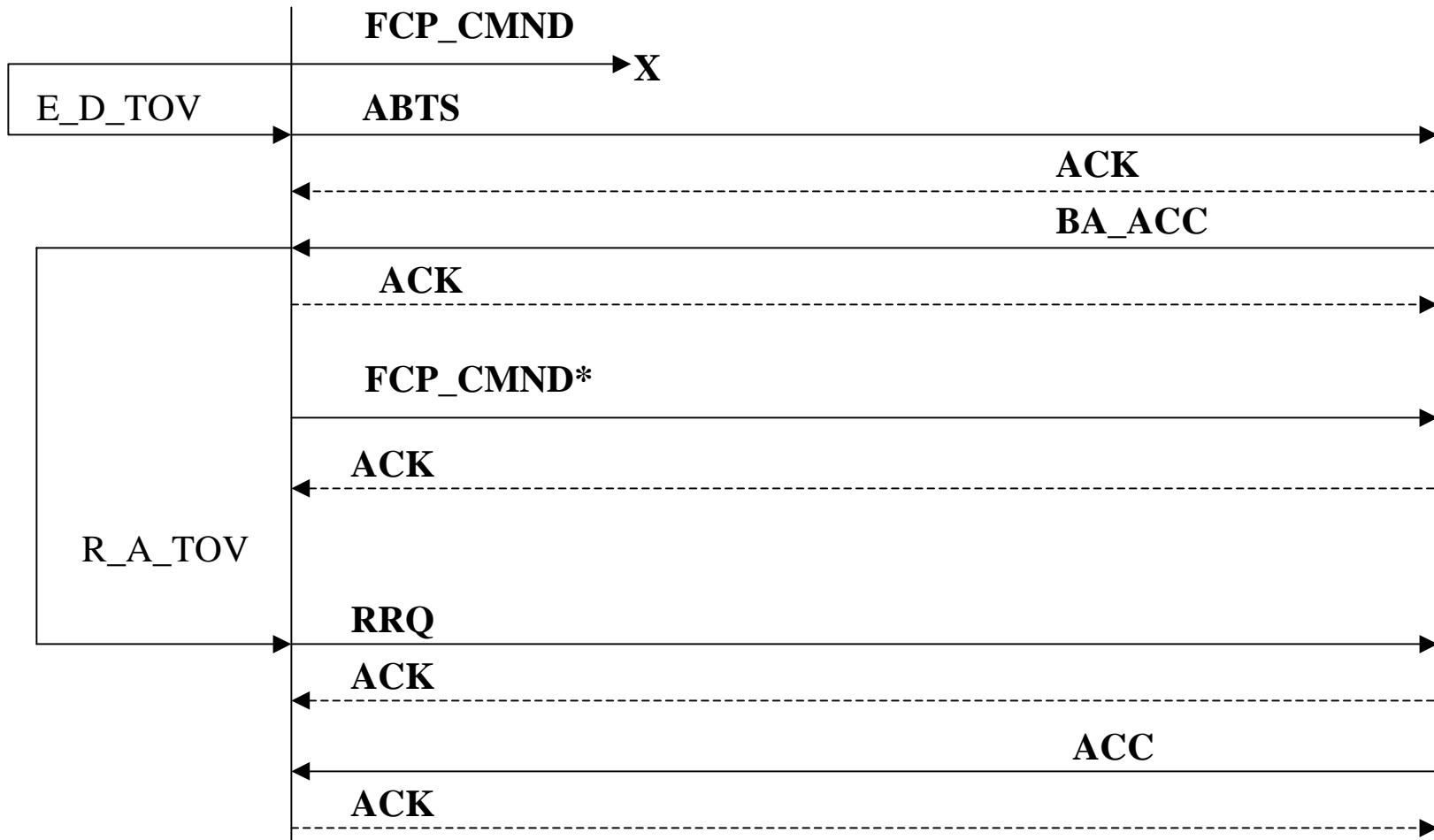
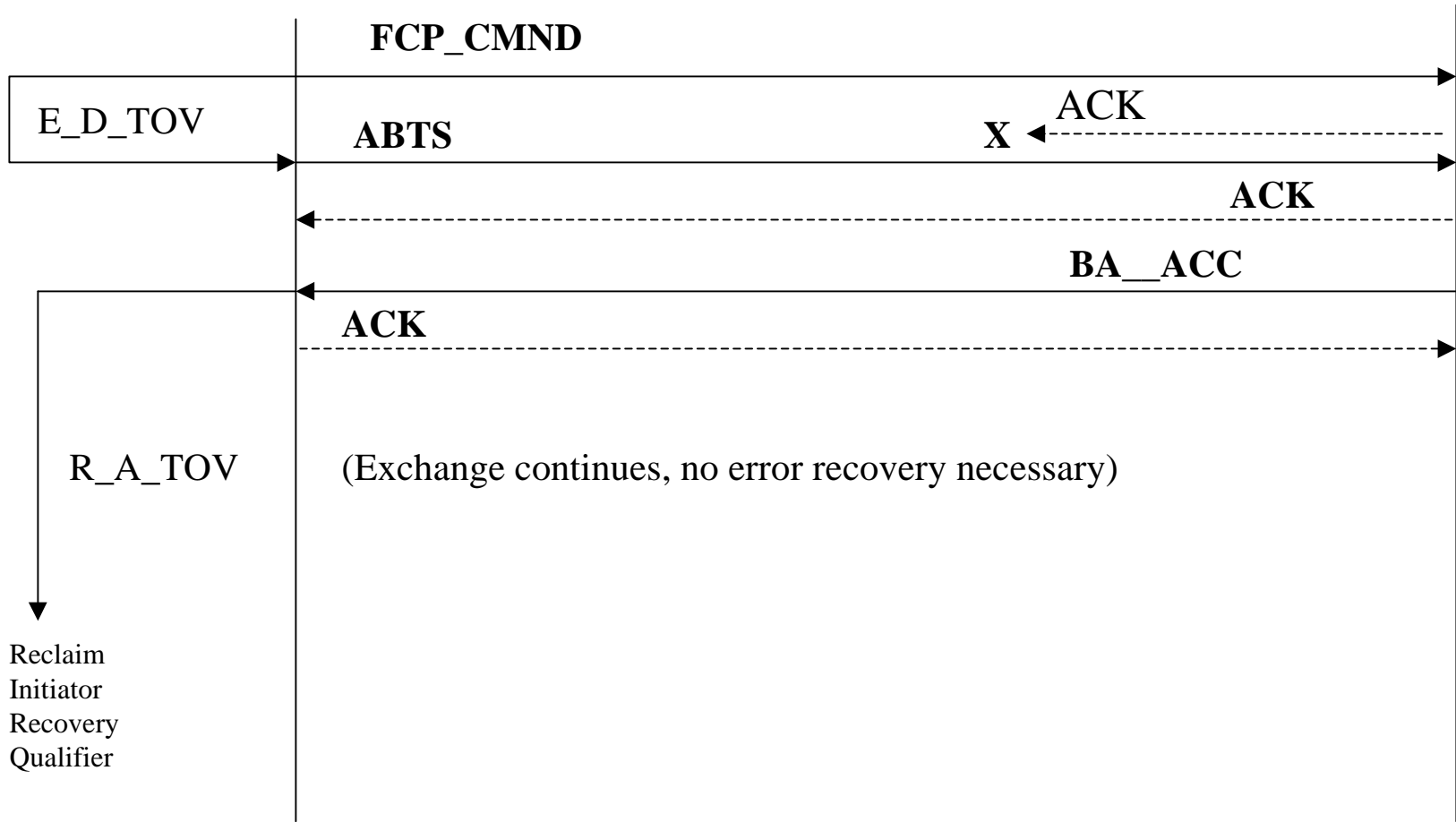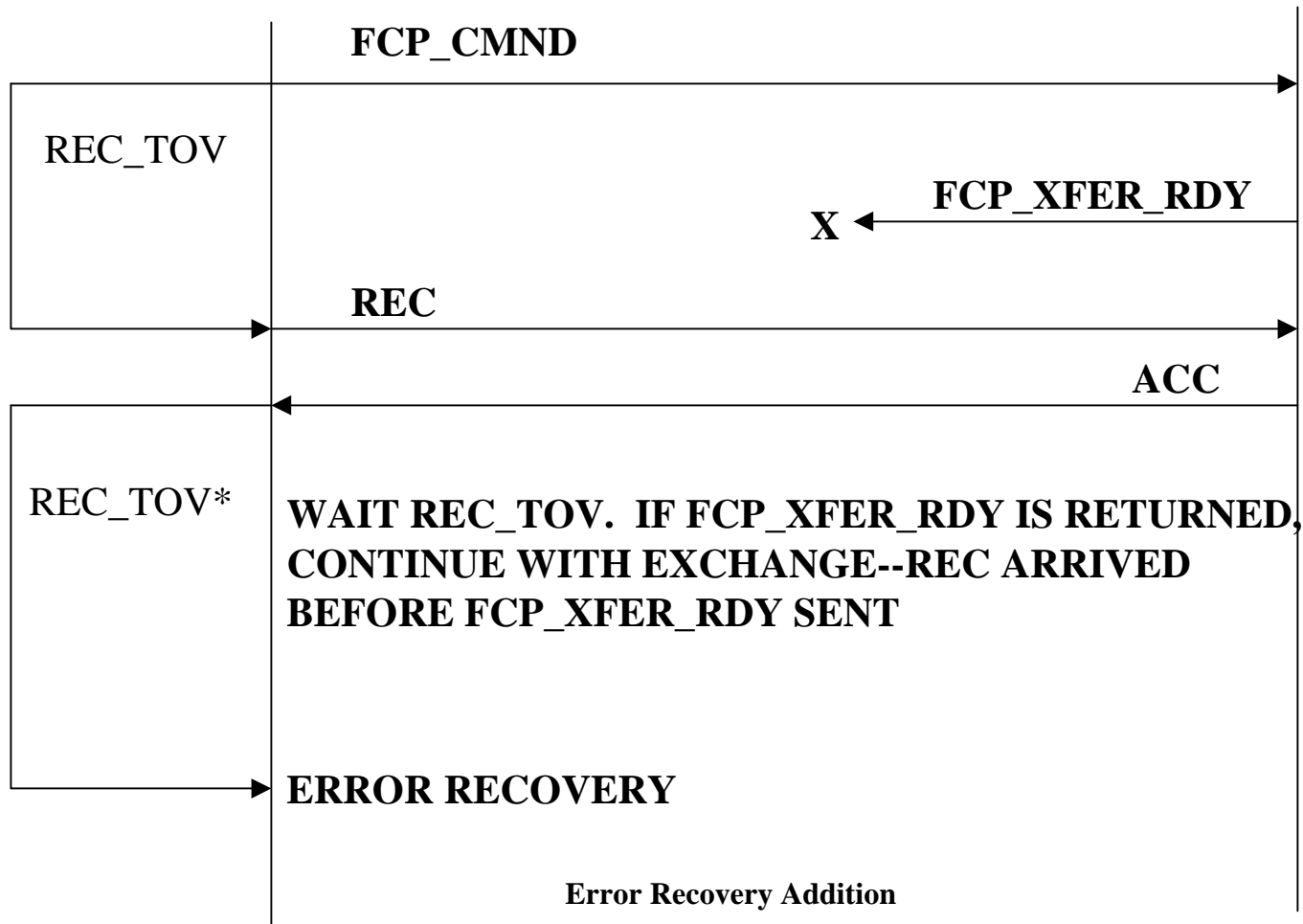**Error Recovery**
The BA_ACC indicates that the FCP_CMD was never received. (BA-ACC payload :
SEQ_ID Validity = invalid, Low SEQ_CNT= 0, High SEQ_CNT= SEQ_CNT of
ABTS frame.) Both the Initiator and Target establish Recovery Qualifiers. FCP_CMD*
is resent in a new Exchange with a new SEQ_ID. The use of REC to determine status
for error recovery is optional.

D.2.1 Class 2 Lost ACK on FCP_CMND

FCP_CMND

E_D_TOV

ABTS   **ACK**

X

ACK

BA__ACC

ACK

R_A_TOV   (Exchange continues, no error recovery necessary)

Reclaim
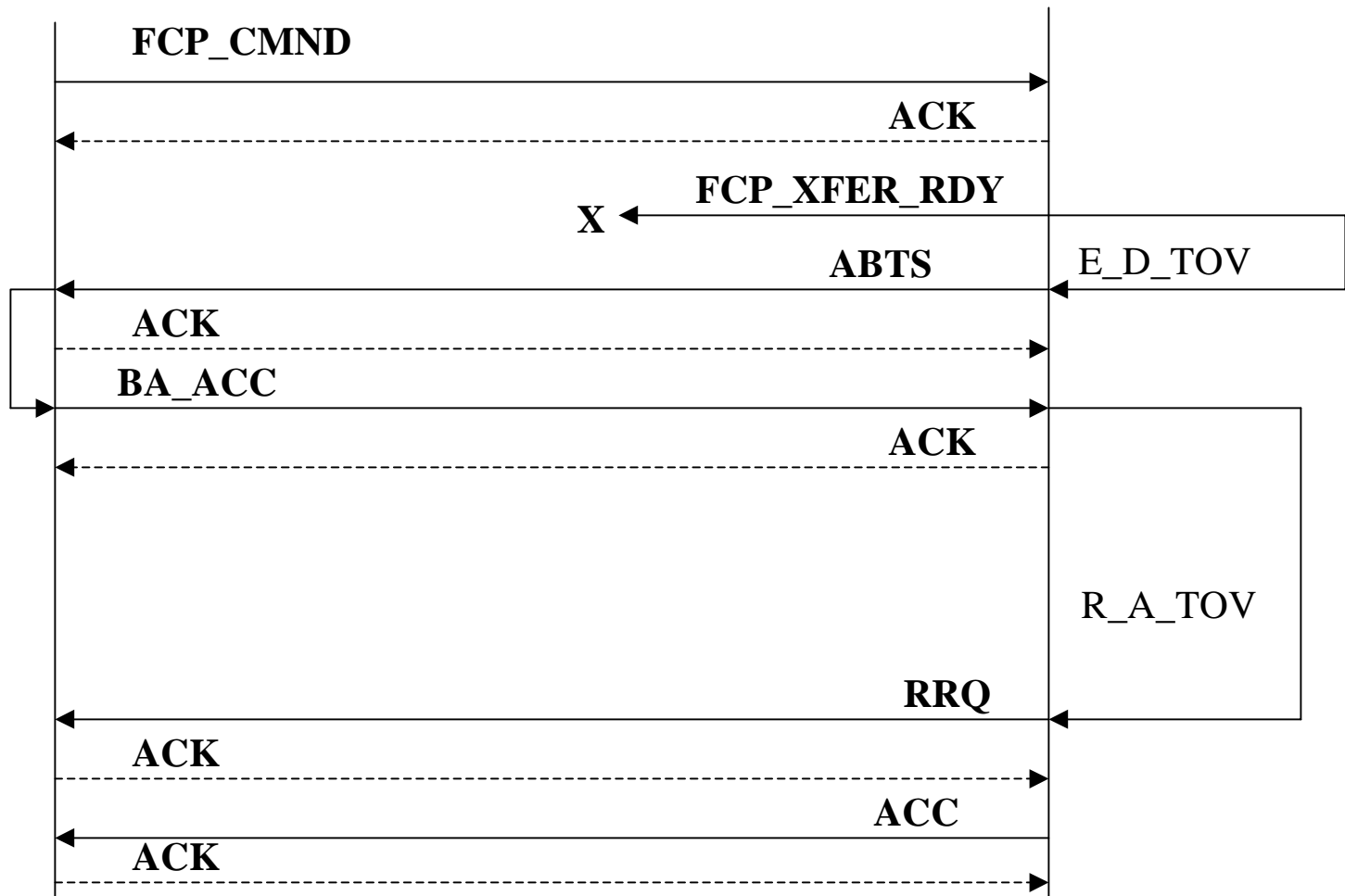Initiator
Recovery
Qualifier

No error recovery. The BA_ACC Payload indicates that FCP_CMD was received. The BA_ACC payload is SEQ_ID Valid, SEQ_ID value of FCP_CMND, low SEQ_CNT=high SEQ_CNT=SEQ_CNT of FCP_CMND. Note that the issuance of RRQ is not necessary in this case, since the Target will not have established a Recovery Qualifier. However, the Initiator cannot reclaim the resources associated with its Recovery Qualifier until R_A_TOV time-out expires. The use of REC to determine status for error recovery is optional.

D.3 Class 3 FCP_XFER_RDY Lost

FCP_CMND

REC_TOV

FCP_XFER_RDY

X

REC

ACC

REC_TOV* **WAIT REC_TOV. IF FCP_XFER_RDY IS RETURNED, CONTINUE WITH EXCHANGE--REC ARRIVED BEFORE FCP_XFER_RDY SENT**

ERROR RECOVERY

Error Recovery Addition

A new SEQ_ID must be used for the retransmission of the FCP_XFER_RDY.
For in-order delivery, the value of REC_TOV* is 0.

D.3 Class 2 FCP_XFER_RDY Lost
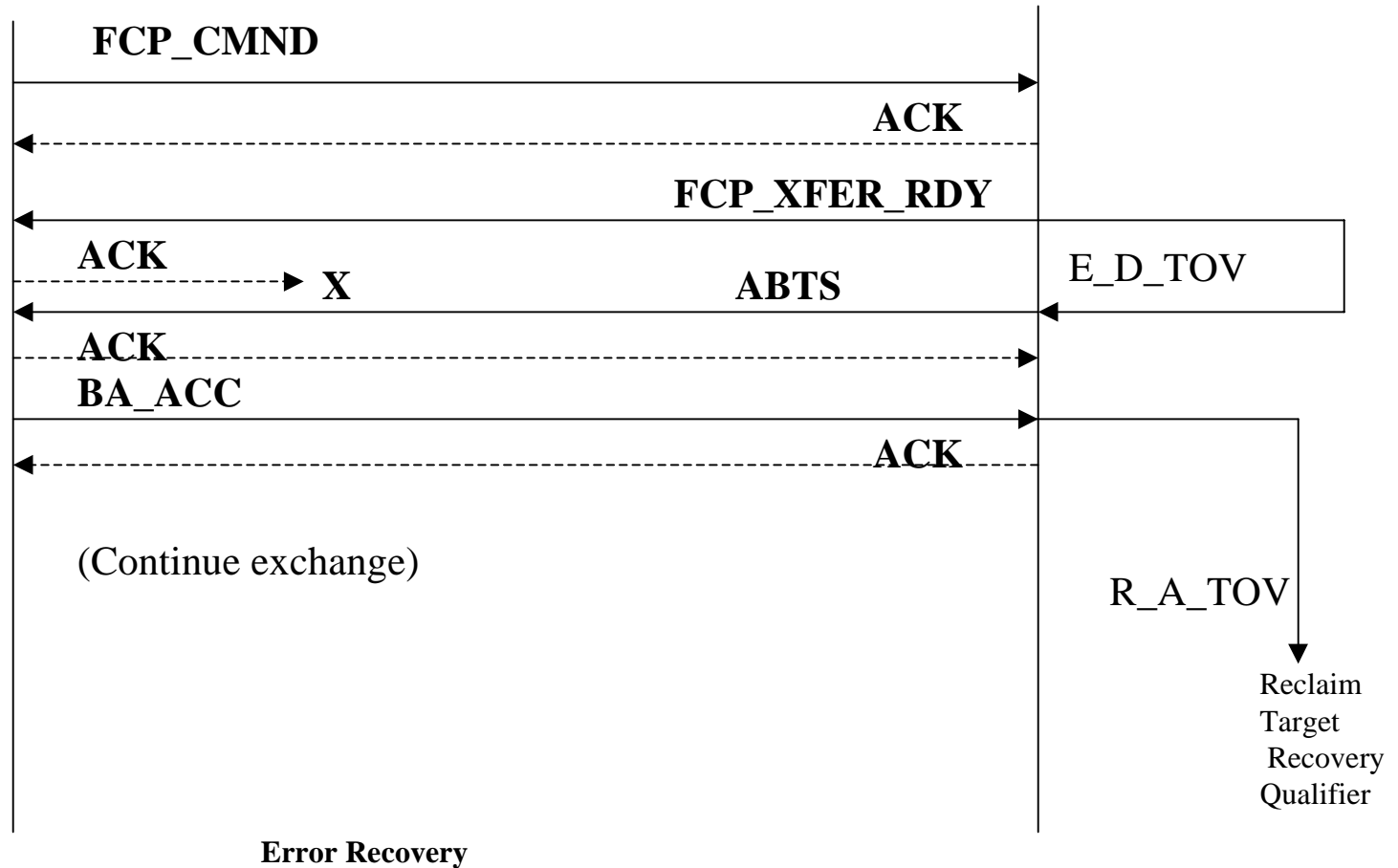
FCP_CMND

ACK

FCP_XFER_RDY

X

ABTS   E_D_TOV

ACK

BA_ACC

ACK

R_A_TOV

RRQ

ACK

ACC

ACK

Error Recovery

For Class 2 the BA_ACC indicates that the FCP_XFER_RDY was never received by the Initiator. (BA_ACC Payload is SEQ_ID invalid, low SEQ_CNT=0, high SEQ_CNT =SEQ_CNT in ABTS frame = 1.)   Both Target and Initiator must establish Recovery Qualifiers. A new SEQ_ID must be used in the retransmission of FCP_XFER_RDY. The use of REC to determine status for error recovery is optional.
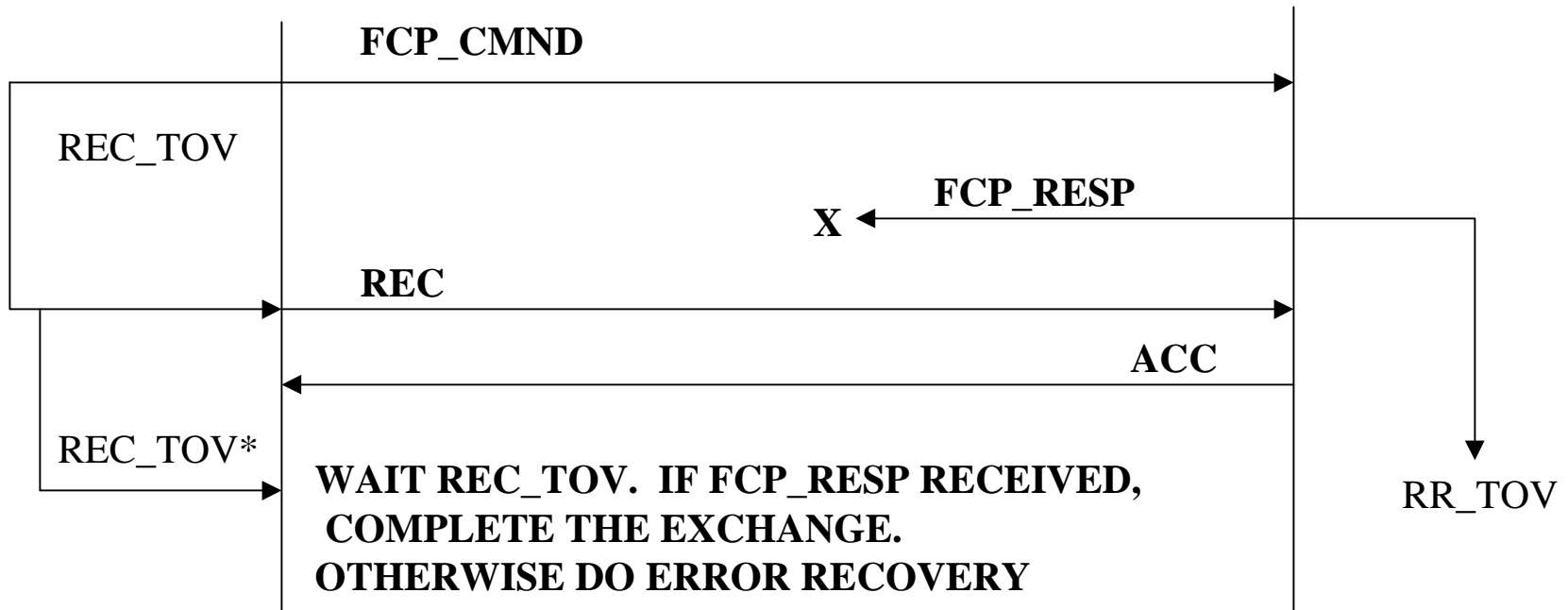
D.4 Class 2 FCP_XFER_RDY Rcvd, ACK Lost

FCP_CMND

ACK

FCP_XFER_RDY

ACK

X   **ABTS**   E_D_TOV

ACK

BA_ACC

ACK

(Continue exchange)

R_A_TOV

Reclaim
Target
Recovery
Qualifier

Error Recovery

None:
 The BA_ACC indicates that the FCP_XFER_RDY was received by the Inititator, ACC Payload of SEQ_ID valid, SEQ-ID
=SEQ_ID of the FCP_XFER_RDY frame, low SEQ_CNT=high SEQ_CNT=SEQ_CNT of the ABTS frame. No error
 recovery is required. Note: There is no need for the Target to issue the RRQ since no Recovery Qualifier was
 established by the Initiator in this case. It must still let R_A_TOV expire before reclaiming the resources associated
 with its Recovery Qualifier. The use of REC to determine status for error recovery is optional.

D.5 Class 3 FCP_RESP Lost, No FCP_CONF Req.

FCP_CMND

REC_TOV

FCP_RESP

X

REC

ACC

REC_TOV*

**WAIT REC_TOV. IF FCP_RESP RECEIVED,
COMPLETE THE EXCHANGE.
OTHERWISE DO ERROR RECOVERY**

RR_TOV

For Sequence level recovery, 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_RESP, or for at least RR_TOV, in order to
preserve the FCP_RESP information. This long timeout can be avoided by
using FCP_CONF. The FCP_RESP retransmission must use a new SEQ_ID.

For Exchange level recovery, the context of the Exchange can be purged by the
Target after FCP_RESP is transmitted.

For in-order delivery, the value of R_A_TOV* is 0.

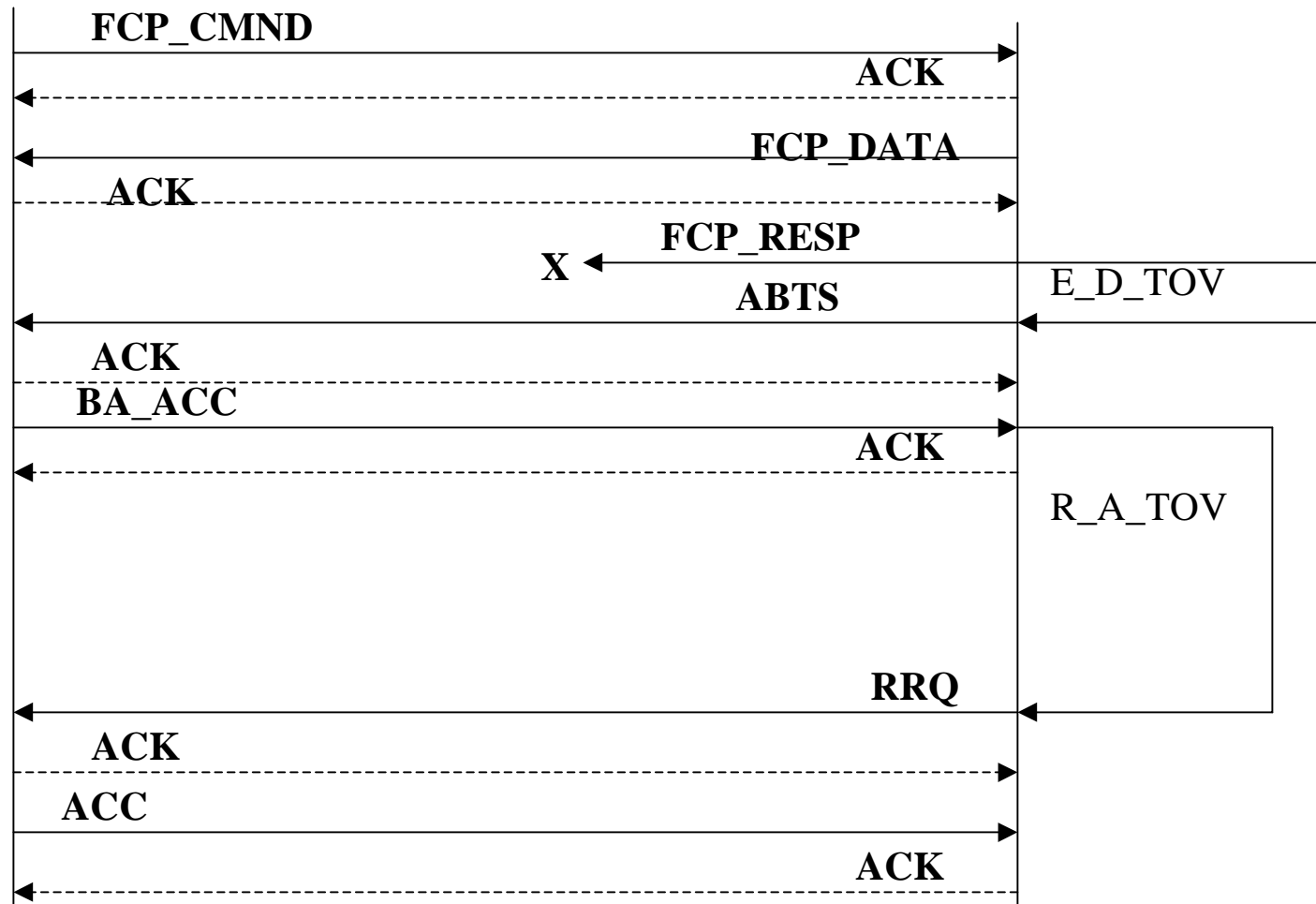# D.5 Class 2 FCP_RESP Lost, No FCP_CONF Req
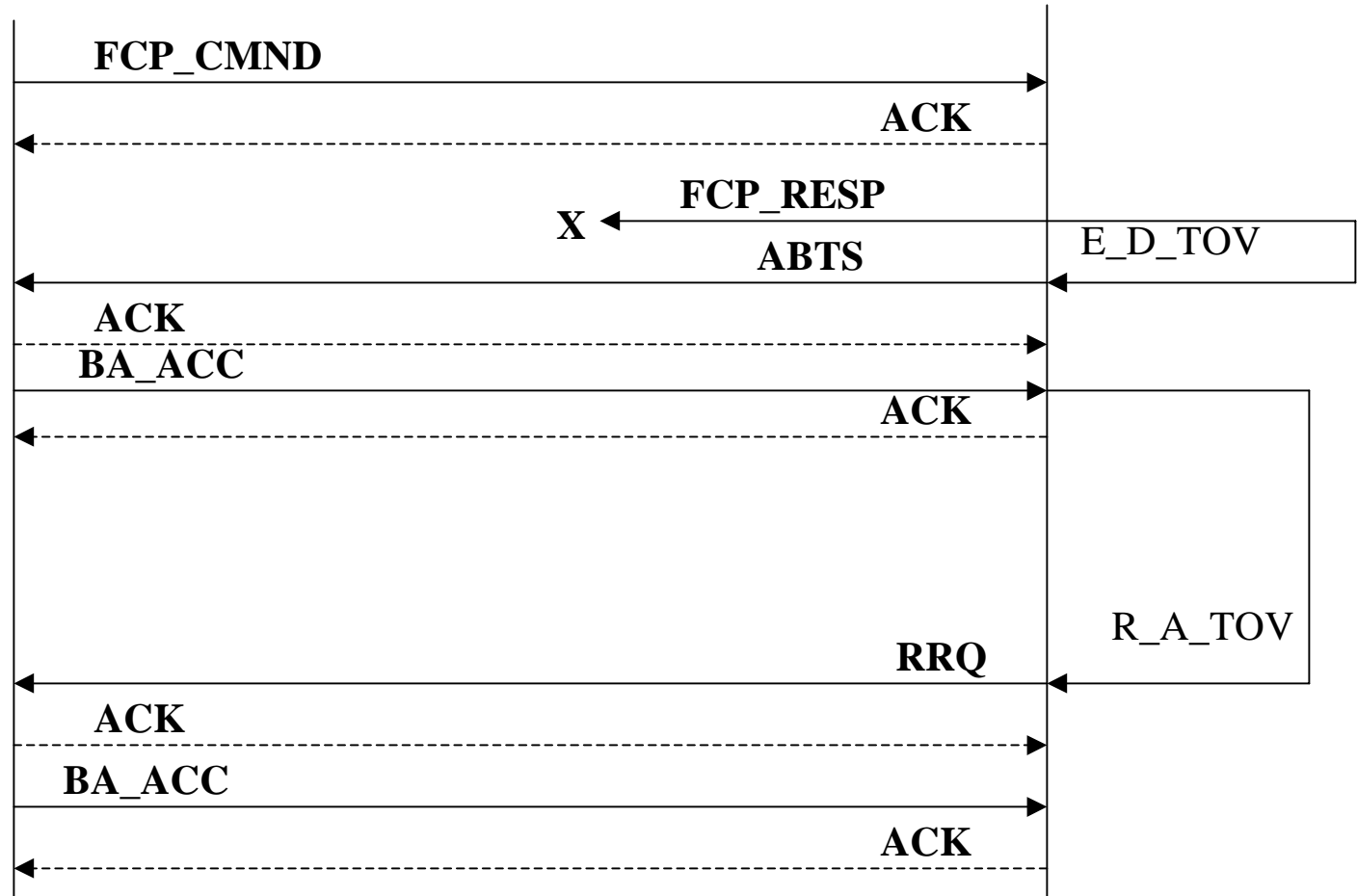
Error Recovery
BA_ACC indicates that FCP_RESP was never received by the Initiator (Payload is SEQ_ID invalid,
low SEQ_CNT=0, high SEQ_CNT=SEQ_CNT in ABTS frame=1.) Both Initiator and Target must
establish Recovery Qualifiers. A new SEQ_ID must be used in the retransmission of FCP_RESP.
 The use of REC to determine status for error recovery is optional.

D.5a Class 2 FCP_RESP Lost after a Read Cmnd, No FCP_CONF Req.

FCP_CMND

ACK

FCP_DATA

ACK

FCP_RESP

X

E_D_TOV

ABTS

ACK

BA_ACC

ACK

R_A_TOV

RRQ

ACK

ACC

ACK

Error Recovery
BA_ACC indicates that FCP_RESP was never received by the Initiator (Payload is SEQ_ID valid,
SEQ_ID=SEQ_ID of FCP_DATA Sequence, low SEQ_CNT=x, high SEQ_CNT=SEQ_CNT in
ABTS frame.) Both Initiator and Target must establish Recovery Qualifiers. A new SEQ_ID must
be used in the retransmission of FCP_RESP. The use of REC to determine status for error recovery
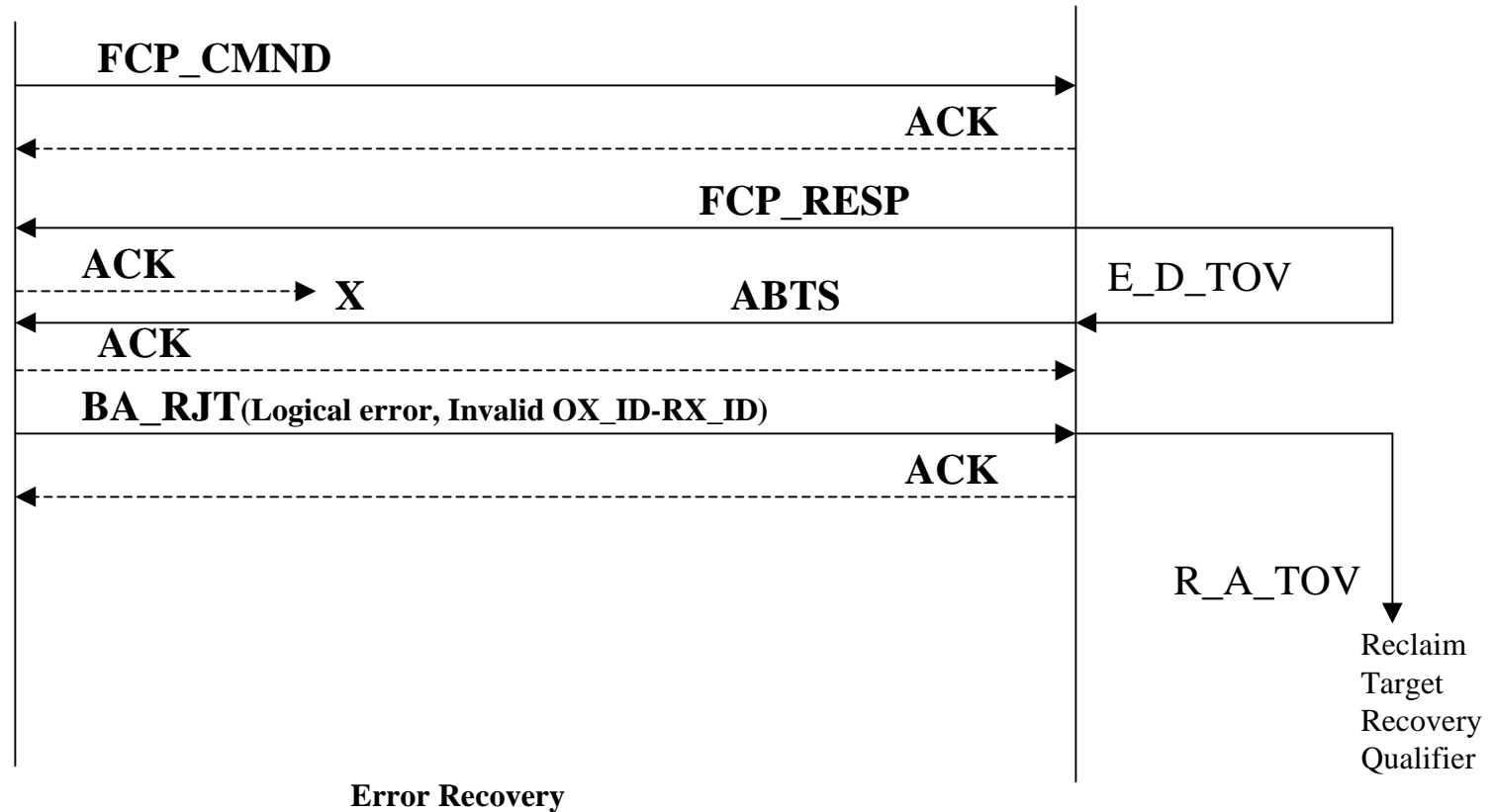is optional.

D.6 Class 2 FCP_RESP Rcvd, ACK Lost

FCP_CMND

ACK

FCP_RESP

ACK X **ABTS**

E_D_TOV

ACK

BA_RJT(Logical error, Invalid OX_ID-RX_ID)

ACK

R_A_TOV

Reclaim
Target
Recovery
Qualifier

Error Recovery

None:
The BA_RJT for the ABTS indicates the Exchange is unknown and therefore complete and no error recovery is required. The Target must establish a Recovery Qualifier. The associated resources cannot be reused for a period of R_A_TOV. (See D.5?? where the same OX_ID is used before the expiration of R_A_TOV)
Note: The Target need not issue RRQ as no Recovery Qualifier was established by the initiator.

D.6a Class 2 FCP_RESP Rcvd, ACK Lost, Case 2

FCP_CMND, OX_ID=j

ACK

FCP_RESP

ACK → **X**

E_D_TOV

E_D_TOV

ABTS

FCP_CMND, OX_ID=j

ACK

ACK

X

BA_RJT(Logical error, Invalid OX_ID-RX_ID)

E_D_TOV Timer
Stopped

R_A_TOV

ACK

Reclaim
Target
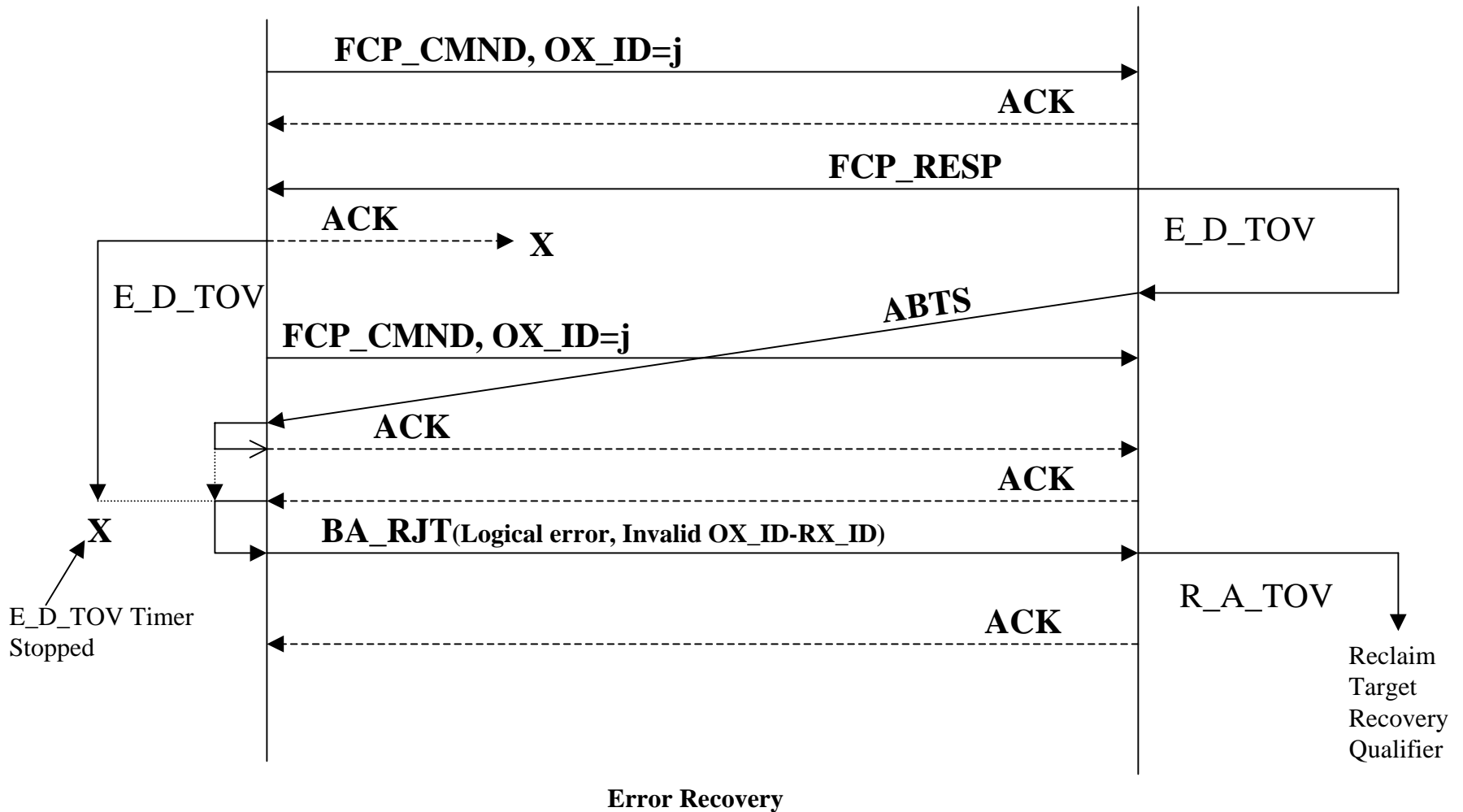Recovery
Qualifier

Error Recovery

None:
The BA_RJT for the ABTS indicates the Exchange is unknown and therefore complete and no
error recovery is required. The Target must establish a Recovery Qualifier. The associated
resources cannot be reused for a period of R_A_TOV. Note that no action is taken on ABTS
until the ACK to the outstanding Sequence has been received, allowing the analysis to take into
consideration the RX_ID, to eliminate ambiguity. (See D. 5?? For another case.)
Note: The Target need not issue RRQ as no Recovery Qualifier was established by the initiator.

D.7 Class 3 Lost Write Data, Last Frame of Seq.
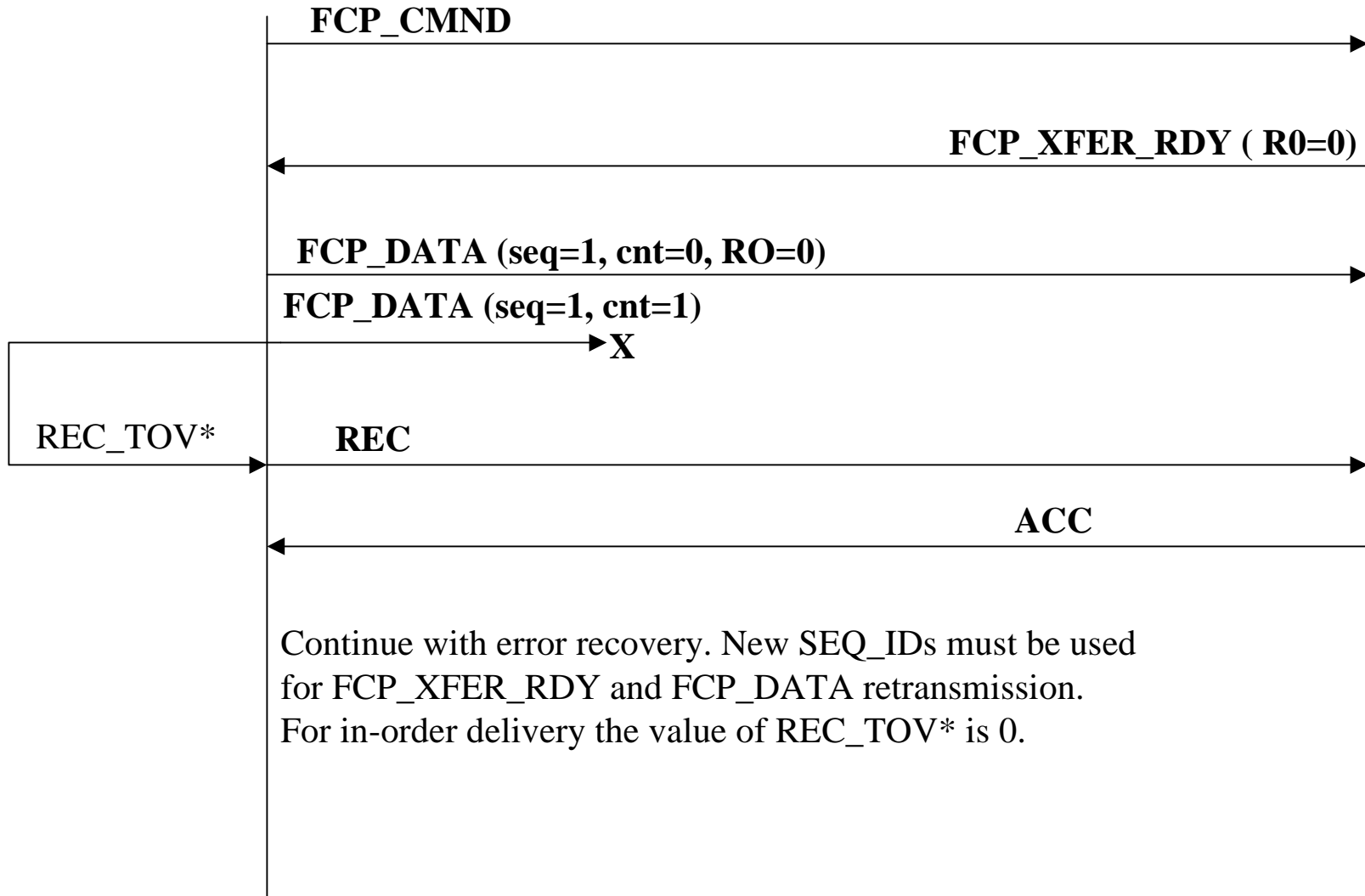
FCP_CMND

FCP_XFER_RDY (R0=0)

FCP_DATA (seq=1, cnt=0, RO=0)

FCP_DATA (seq=1, cnt=1)

X

REC_TOV*

REC

ACC

Continue with error recovery. New SEQ_IDs must be used
for FCP_XFER_RDY and FCP_DATA retransmission.
For in-order delivery the value of REC_TOV* is 0.

D.7 Class 2 Lost Write Data, Last Frame of Seq.

FCP_CMND

ACK

FCP_XFER_RDY (R0=0)

ACK

FCP_DATA (seq=1, cnt=0, RO=0)

FCP_DATA (seq=1, cnt=1)

X

E_D_TOV

ABTS

ACK

BA_ACC

ACK

REC

ACC

R_A_TOV

RRQ

ACK

ACC

ACK
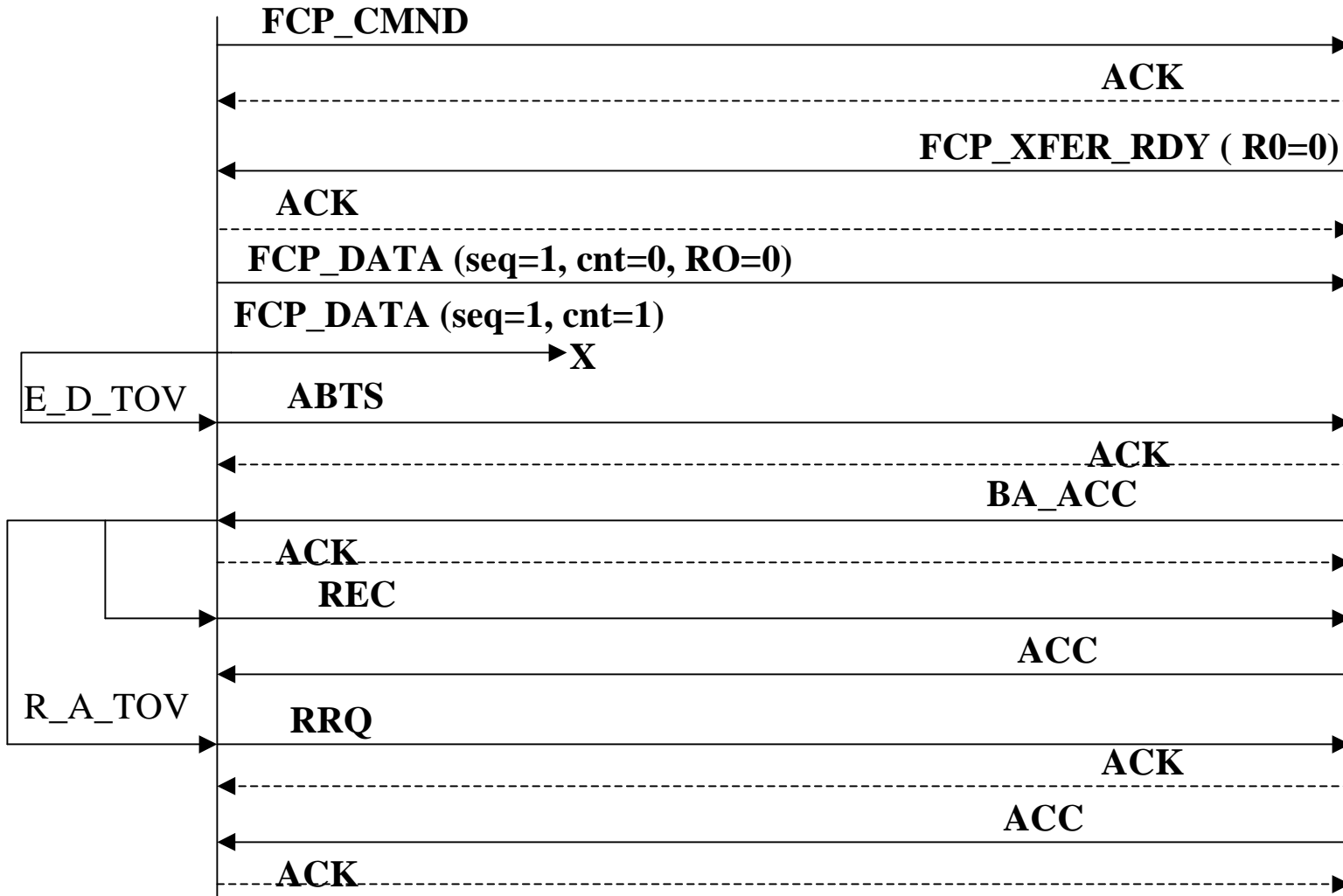
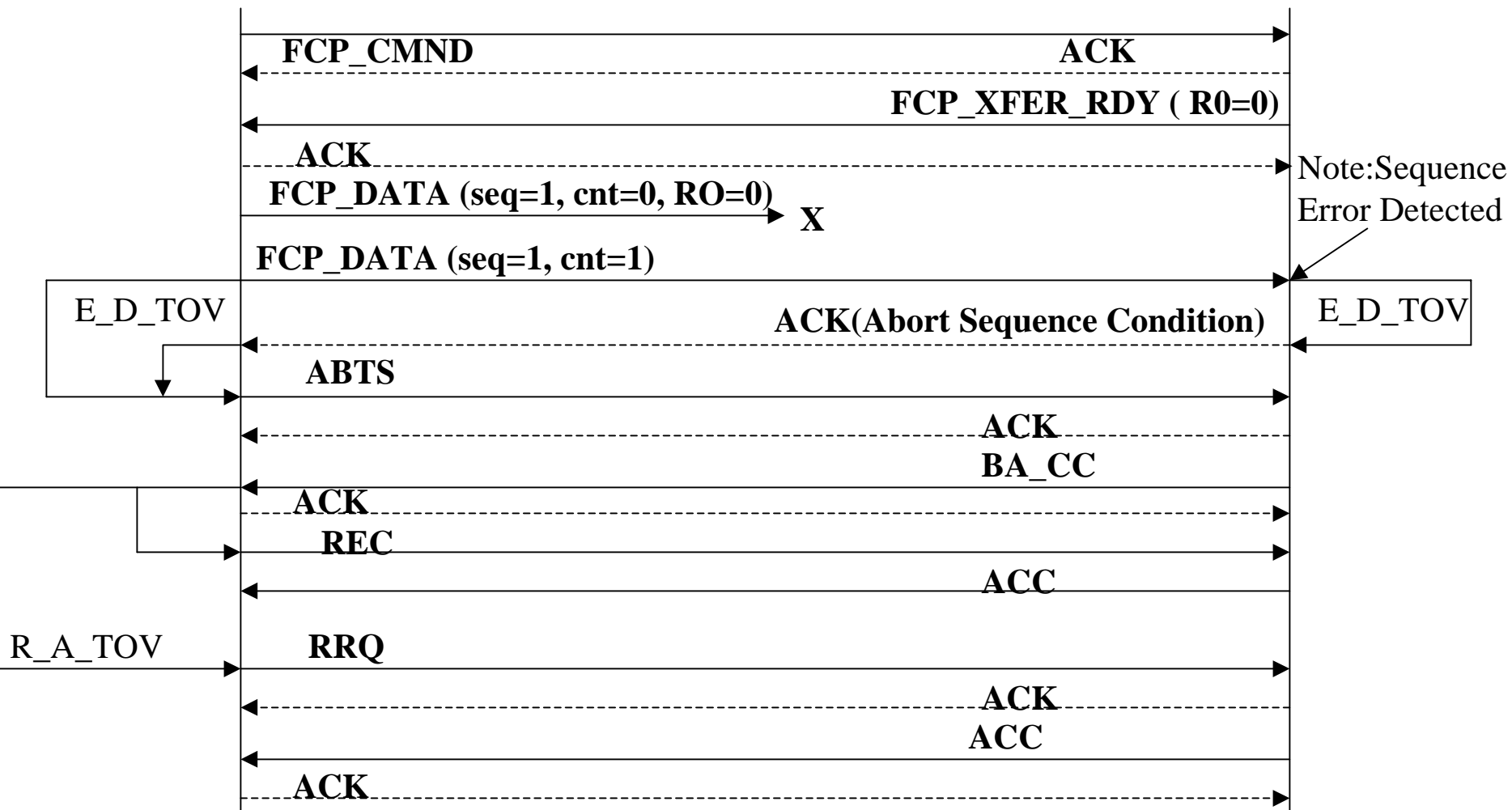**Error Recovery Addition**

BA_ACC indicates by its payload, that FCP_DATA Sequence was not completely received(Payload: SEQ_ID valid, SEQ_ID value =SEQ_ID of FCP_CMND, low SEQ_CNT of 0 not equal to high SEQ_CNT of 2). New SEQ_IDs shall be used for the retransmission of FCP_XFER_RDY and FCP_DATA. The ACKs for REC/ACC are not shown.

D.8 Cl 3, Lost Write Data, Not Last Fr. of Seq.

FCP_CMND

FCP_XFER_RDY (R0=0)

FCP_DATA (seq=1, cnt=0, RO=0) X

FCP_DATA (seq=1, cnt=1)

(If the lost data frame arrives at the Target prior to the expiration of
REC_TOV, then the Exchange proceeds and no recovery is required.)

REC_TOV* **REC**

ACC

Continue error recovery. New SEQ_IDs must be used
for FCP_XFER_RDY and FCP_DATA retransmission.
For in-order delivery the value of REC_TOV* is 0
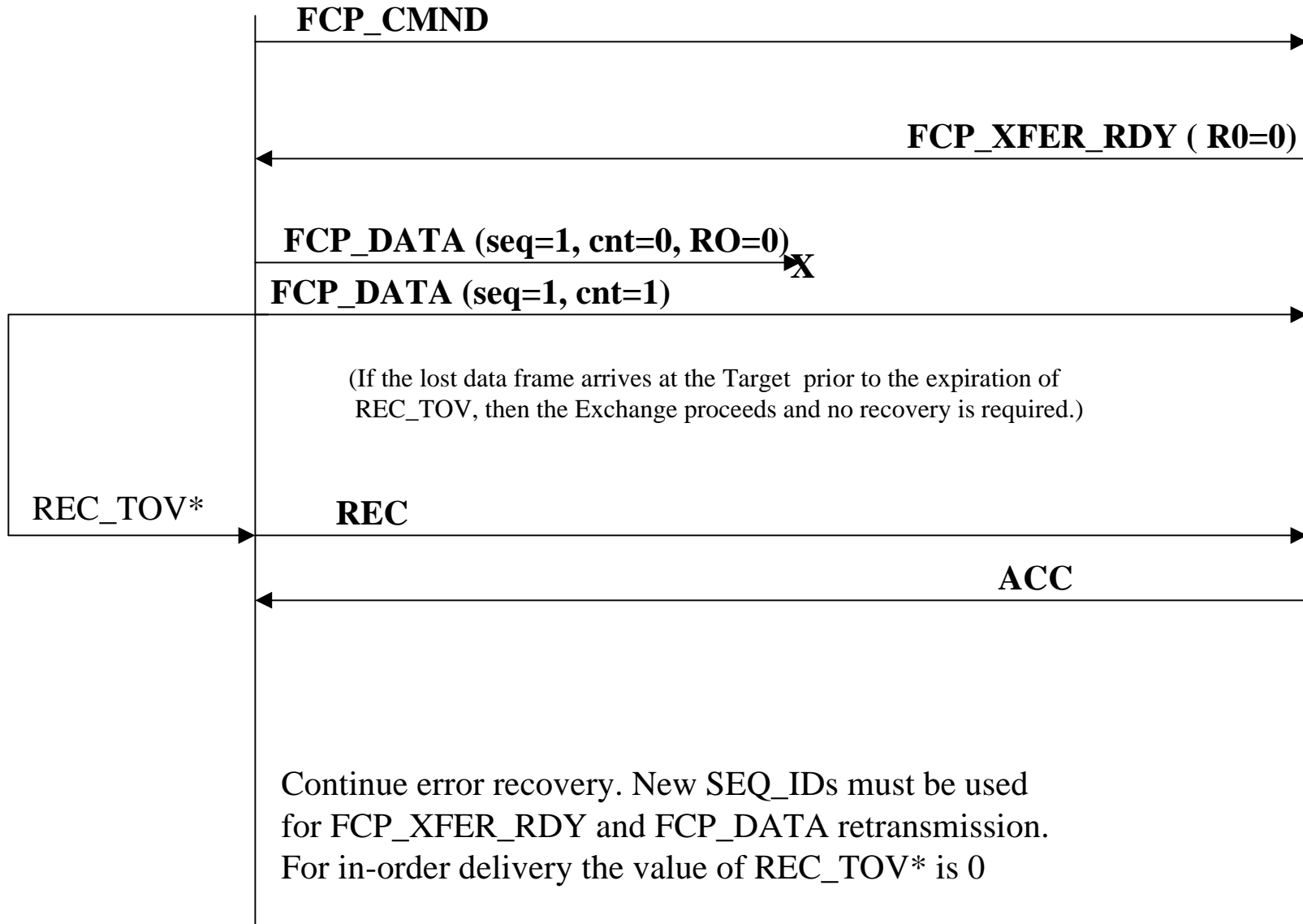
D.8 Class 2 Lost Write Data, Not Last Frame of Seq.

FCP_CMND .... **ACK**

FCP_XFER_RDY (R0=0)

ACK ....

FCP_DATA (seq=1, cnt=0, RO=0) → **X**

Note:Sequence
Error Detected

FCP_DATA (seq=1, cnt=1)

E_D_TOV

ACK(Abort Sequence Condition)   E_D_TOV

ABTS

ACK ....

BA_CC

ACK ....

REC

ACC

R_A_TOV   **RRQ**

ACK ....

ACC

ACK ....

Error Recovery Addition

BA_ACC indicates by its payload, that the FCP_DATA Sequence was not completely received(Payload: SEQ_ID valid,
SEQ_ID value =SEQ_ID of FCP_CMND, low SEQ_CNT of 0 not equal to high SEQ_CNT of 2). New SEQ_IDs shall
be used for retransmission of FCP_XFER_RDY and FCP_DATA. Note that if all data frames arrive at the Target prior to
the expiration of E_D_TOV(out-of-order), then there is no error and no recovery is necessary. ACKs for REC/ACC are
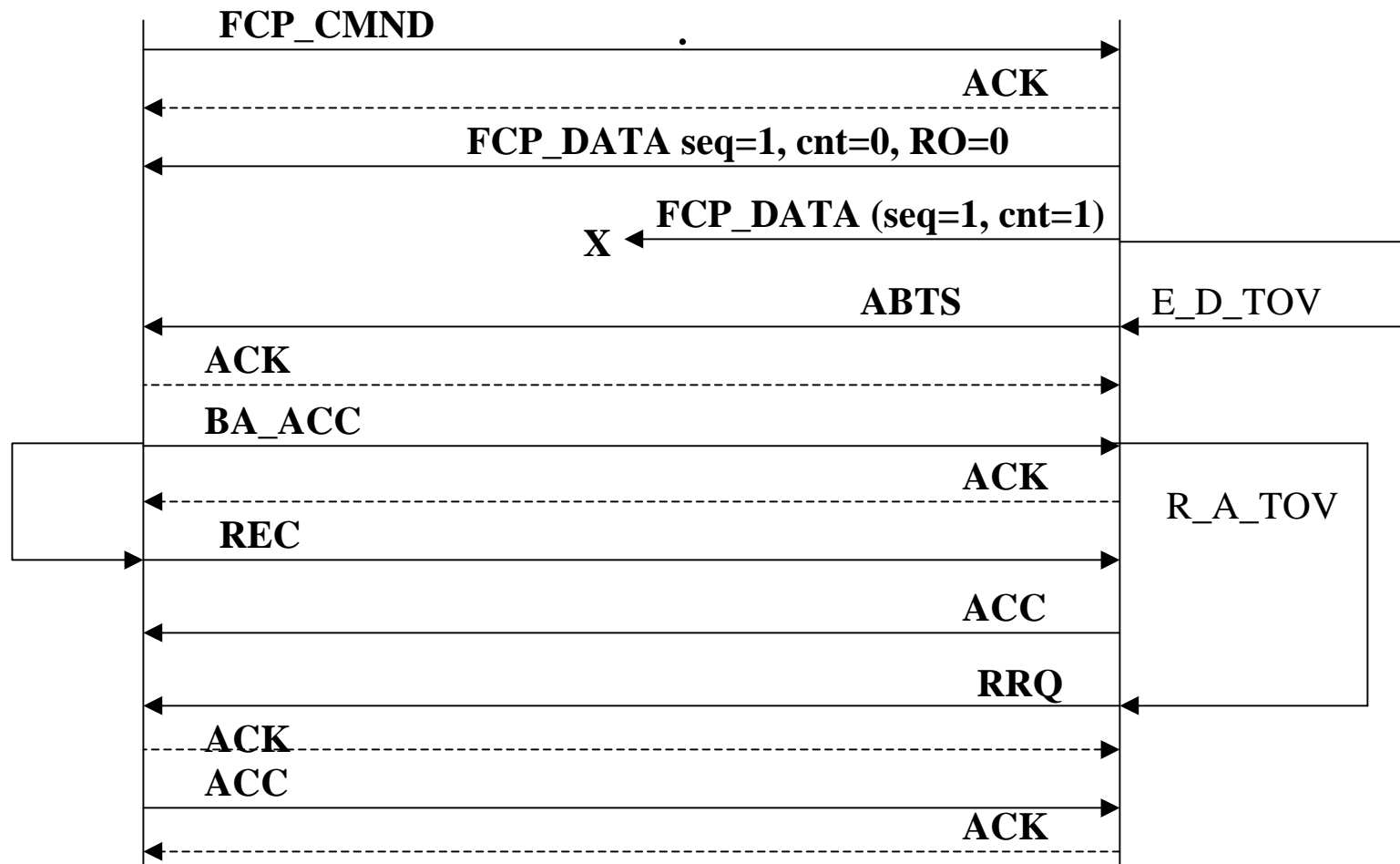not shown.

D.9 Class 3 Lost Read Data, Last Frame of Seq.

FCP_CMND

FCP_DATA seq=1, cnt=0, RO=0

FCP_DATA (seq=1, cnt=1)

X

FCP_RESP

REC_TOV*

IF FCP_DATA (seq=1,cnt=1) is received during REC_TOV, then complete the Exchange.

REC

ACC
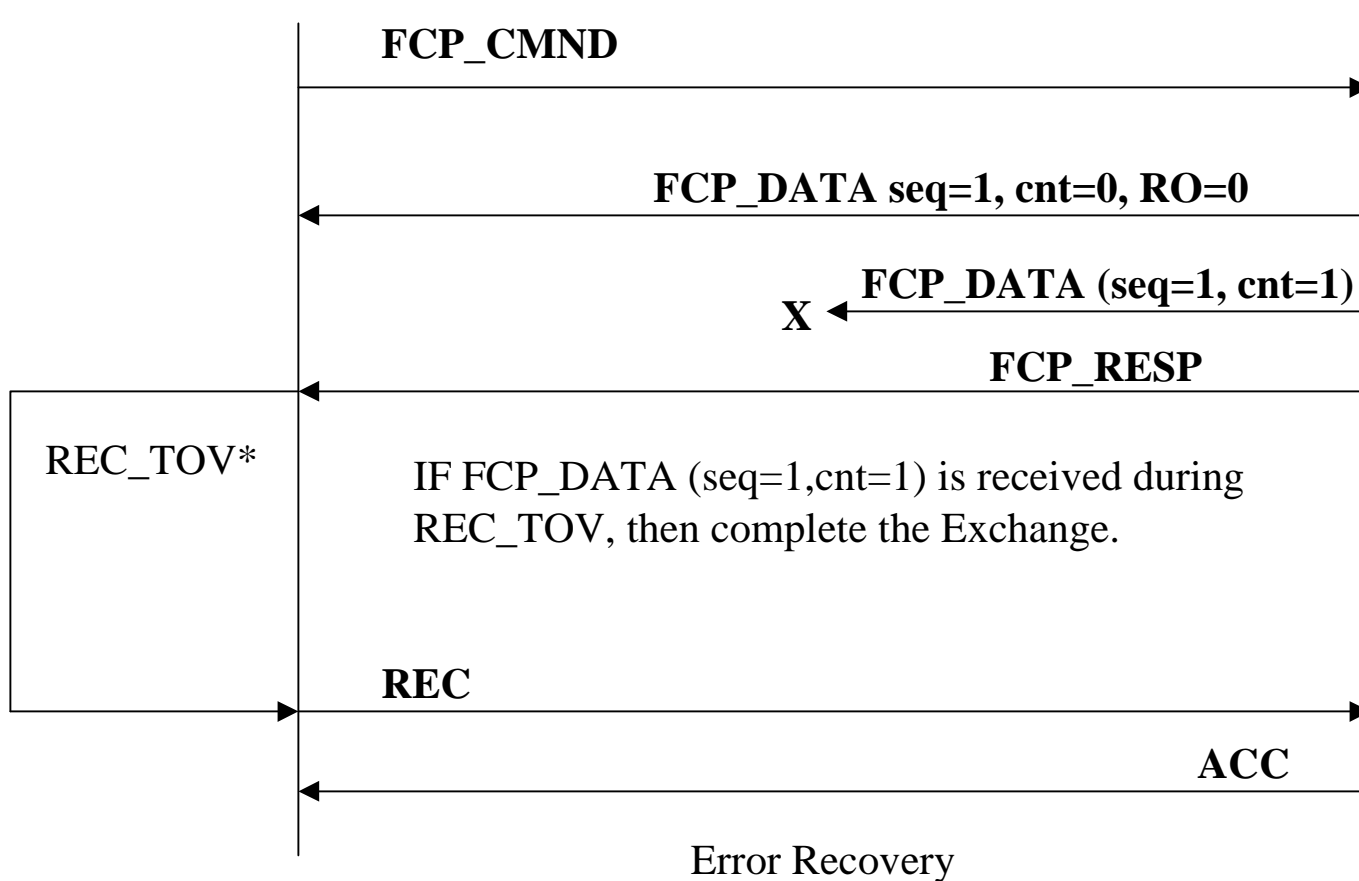
Error Recovery

Continue error recovery. FCP_DATA retransmission must use a new SEQ_ID.
For in-order delivery, the value of REC_TOV* is 0.
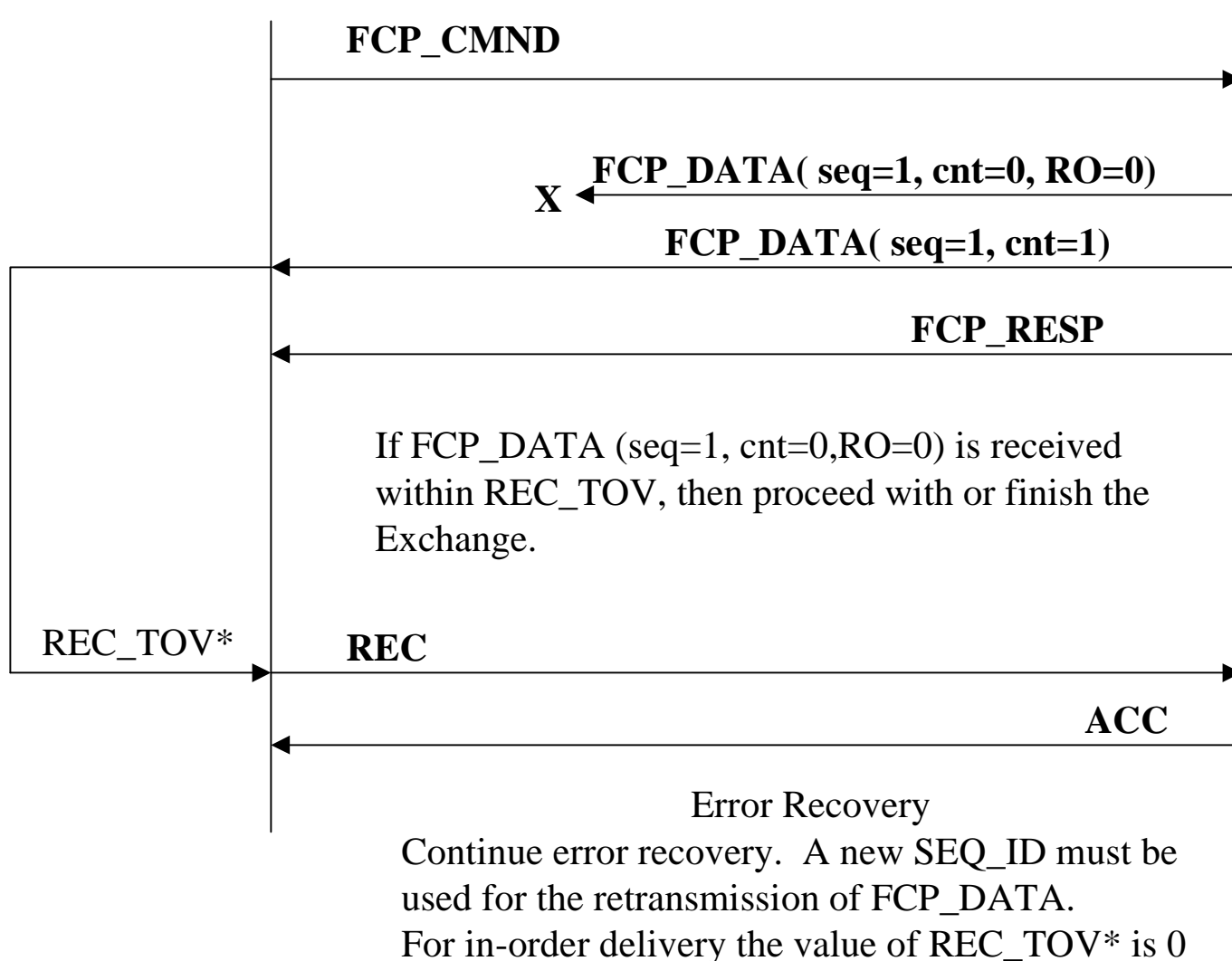
D.9 Class 2 Lost Read Data, Last Frame of Seq

FCP_CMND .

ACK

FCP_DATA seq=1, cnt=0, RO=0

FCP_DATA (seq=1, cnt=1)

X

ABTS   E_D_TOV

ACK

BA_ACC

ACK   R_A_TOV

REC

ACC

RRQ

ACK

ACC

ACK

Error Recovery Addition

BA_ACC indicates by its payload, that the FCP_DATA Sequence was not completely received(Payload:
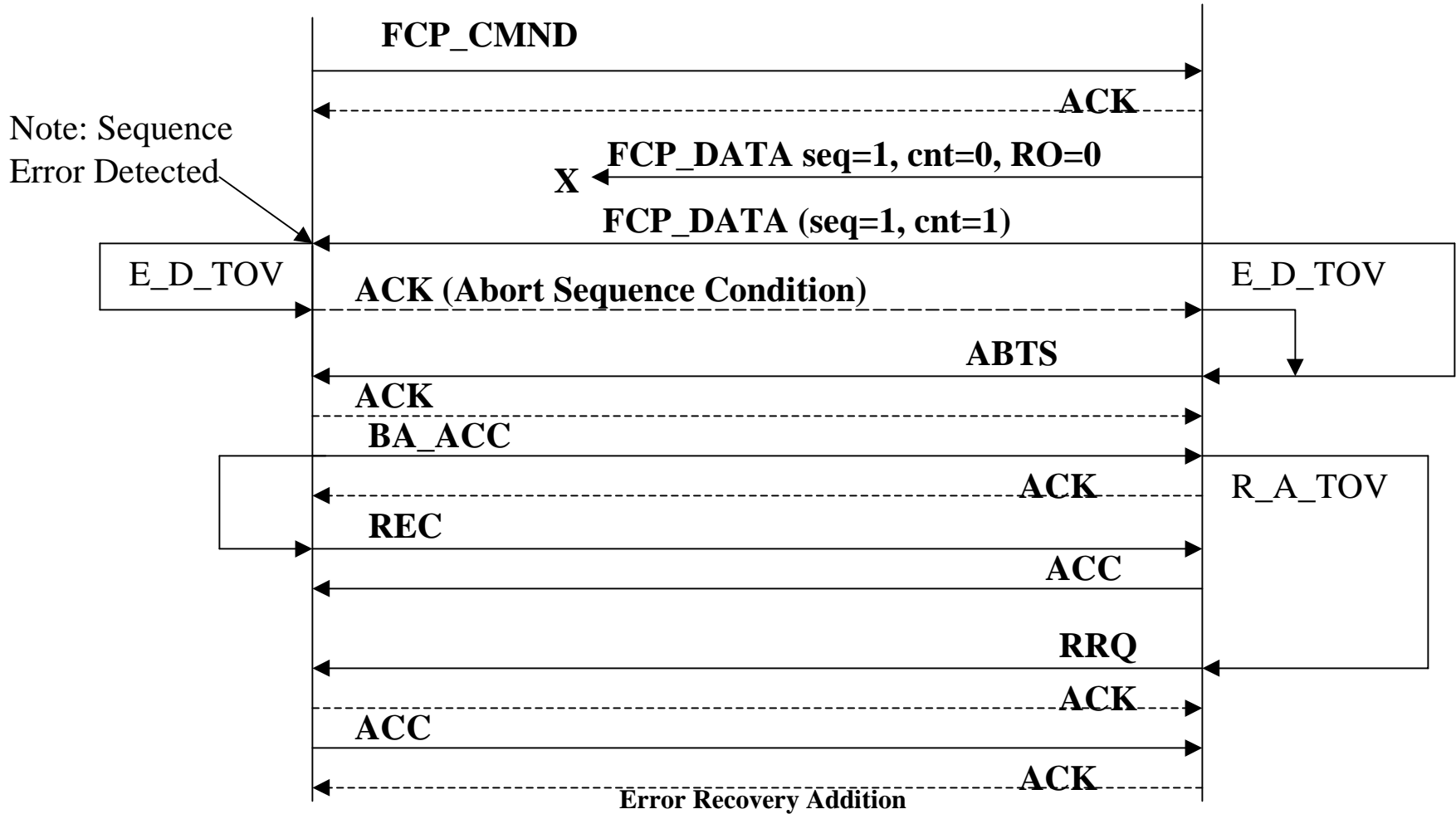SEQ_ID invalid, low SEQ_CNT of 0 not equal to high SEQ_CNT of the ABTS frame of 2).
A new Sequence ID shall be used for retransmission of FCP_DATA.
The ACKs for REC/ACC are not shown.

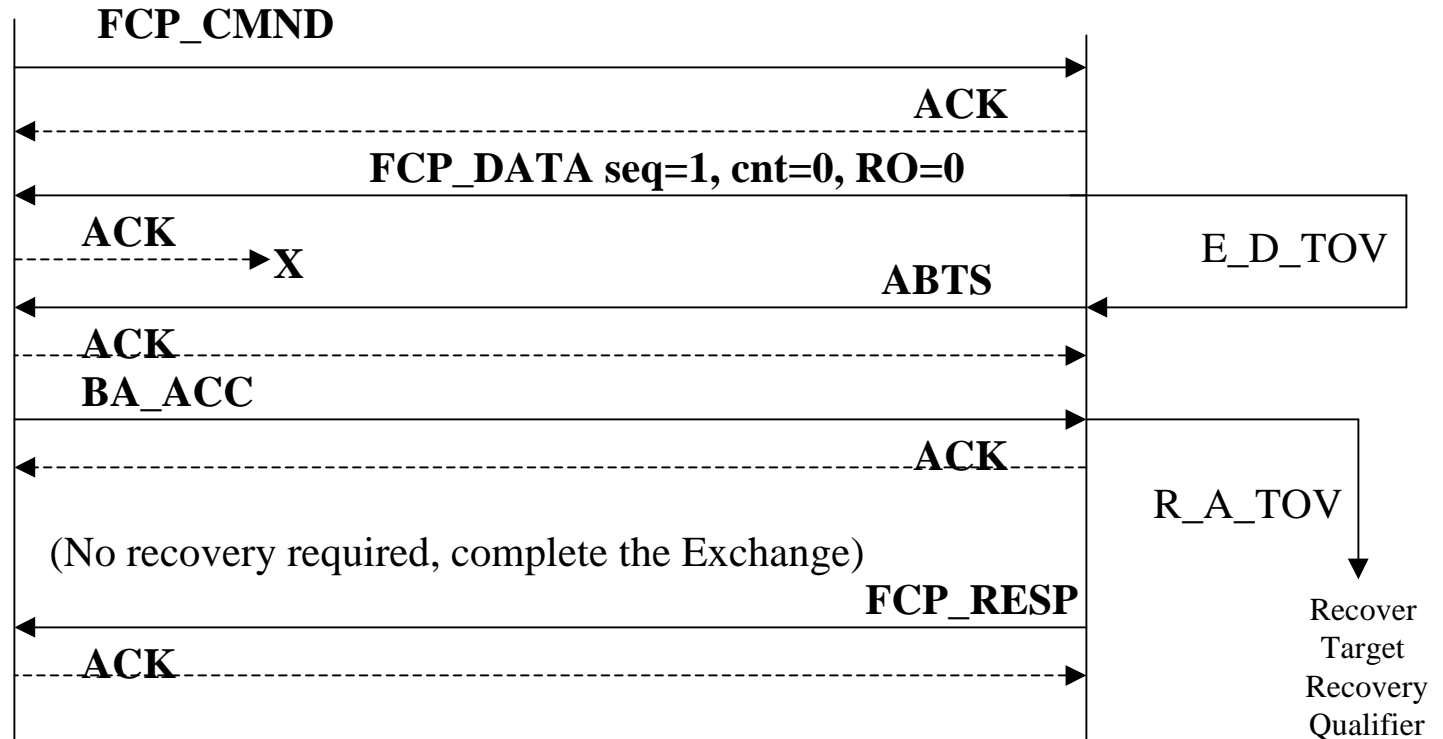# D.10 Class 3 Lost Read Data, Not Last Frame of Seq

FCP_CMND

FCP_DATA(seq=1, cnt=0, RO=0)   X

FCP_DATA(seq=1, cnt=1)

FCP_RESP

If FCP_DATA (seq=1, cnt=0,RO=0) is received within REC_TOV, then proceed with or finish the Exchange.

REC_TOV*   **REC**

ACC

Error Recovery
Continue error recovery. A new SEQ_ID must be used for the retransmission of FCP_DATA.
For in-order delivery the value of REC_TOV* is 0

D.10 Class 2 Lost Read Data, Not Last Frame of Seq

FCP_CMND

ACK

Note: Sequence
Error Detected

FCP_DATA seq=1, cnt=0, RO=0

X

FCP_DATA (seq=1, cnt=1)

E_D_TOV

ACK (Abort Sequence Condition)

E_D_TOV

ABTS

ACK

BA_ACC

ACK

R_A_TOV

REC

ACC

RRQ

ACK

ACC

ACK

Error Recovery Addition

BA_ACC indicates by its payload, that the FCP_DATA Sequence was not completely received(Payload: SEQ_ID invalid, low SEQ_CNT of 0 not equal to high SEQ_CNT of the ABTS frame of 2). A new Sequence ID shall be used for retransmission of FCP_DATA. Note that if all data frames arrive at the initiator before E_D_TOV expires, then no recovery is required; a frame or frames arrived out-of-order. The ACKs for REC/ACC are not shown.

D.11 Class 2 ACK Lost on Read

FCP_CMND

ACK

FCP_DATA seq=1, cnt=0, RO=0

ACK X

E_D_TOV

ABTS

ACK

BA_ACC

ACK

R_A_TOV

(No recovery required, complete the Exchange)

FCP_RESP

ACK
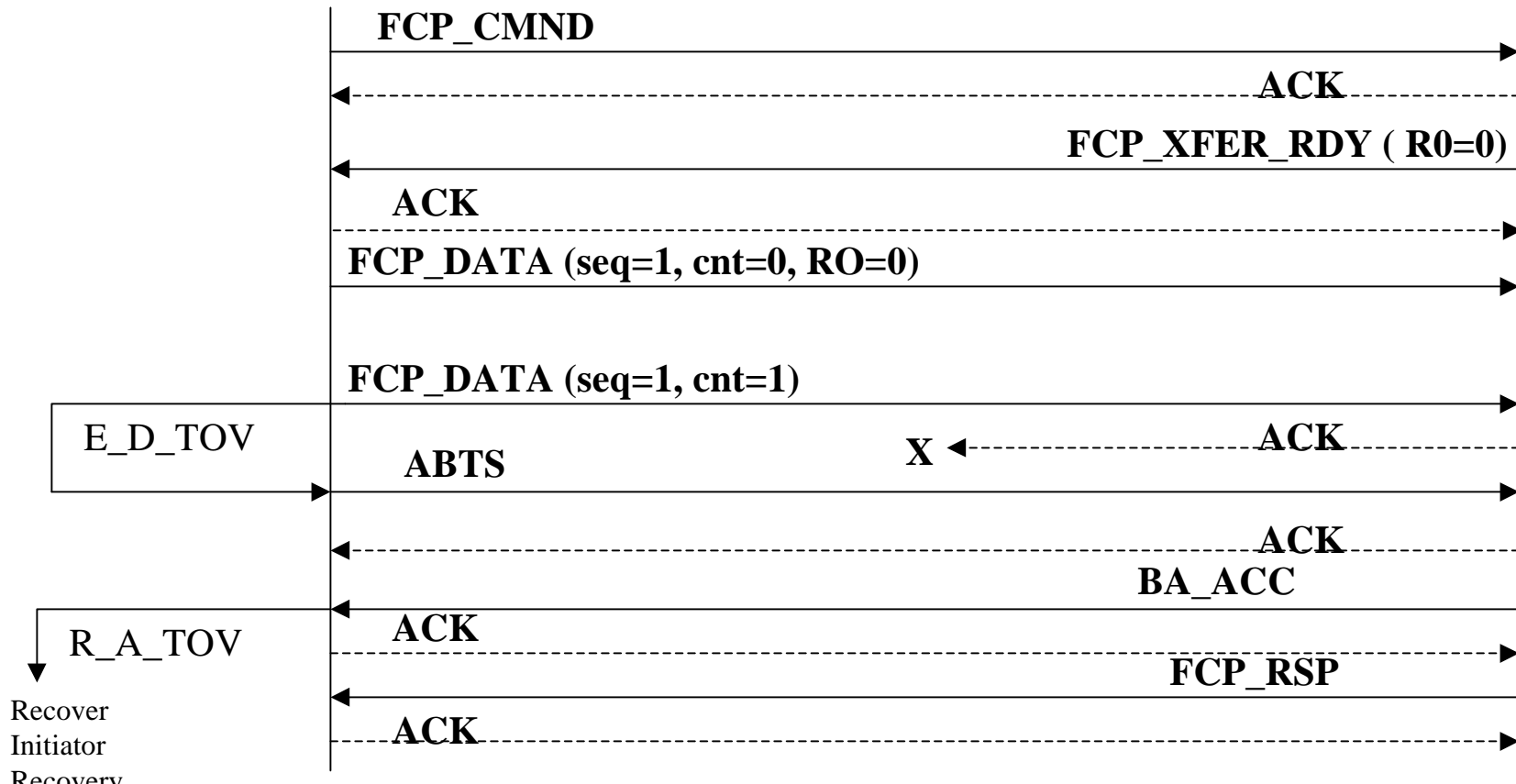
Recover
Target
Recovery
Qualifier

Error Recovery

None:

The initiator has received the FCP_DATA Sequence. No error recovery is required.

Note: The BA_ACC indicates the FCP_DATA sequence was received (Payload is SEQ_ID valid, SEQ_ID value
 =SEQ_ID value of FCP_DATA Sequence, low SEQ_CNT=high SEQ_CNT= SEQ_CNT of ABTS frame).

Note: The Target must establish its Recovery Qualifier. The resources associated with the Recovery Qualifier
can be reclaimed after R_A_TOV. The issuance of RRQ is optional as no Recovery Qualifier was established
 by the Initiator in this case. FCP_RESP can be received anytime after the transmission of FCP_CMD due to
out-of-order delivery.

D.12 Class 2 ACK Lost on Write

FCP_CMND

ACK

FCP_XFER_RDY (R0=0)

ACK

FCP_DATA (seq=1, cnt=0, RO=0)

FCP_DATA (seq=1, cnt=1)

E_D_TOV

ABTS

X ◄------- **ACK**

ACK

BA_ACC

R_A_TOV

ACK

FCP_RSP

Recover
Initiator
Recovery
Qualifier

ACK

Error Recovery

None: The Target received the FCP_DATA sequence. No error recovery is required.

Note: The BA_ACC indicates the data sequence was received (Payload is SEQ_ID valid, SEQ_ID value =SEQ_ID value of the FCP_DATA Sequence, low SEQ_CNT=high SEQ_CNT=SEQ_CNT of ABTS frame). The Target and Initiator continue the Exchange. The Initiator must establish its Recovery Qualifier. The resources associated with the Recovery Qualifier can be reclaimed after R_A_TOV. The issuance of the RRQ is optional as no Recovery Qualifier was established by the Target. FCP_RESP can be received at any time after the last FCP_DATA frame has been transmitted.

D.?1 Class 3 FCP_CONF Lost

FCP_CMND

FCP_RESP

FCP_CONF   X

R_A_TOV

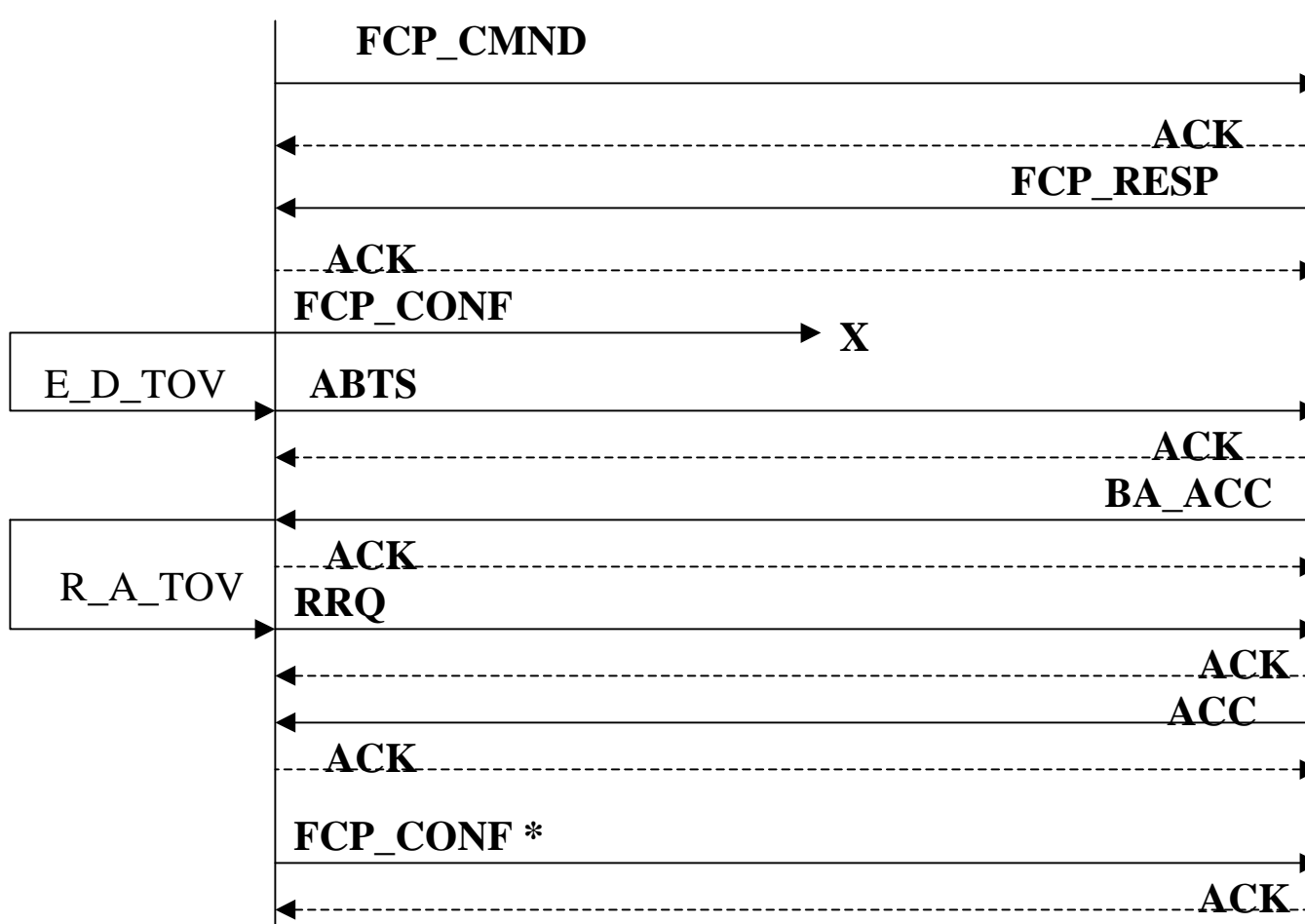**REC**

LS-RJT

R_A_TOV

Error Recovery
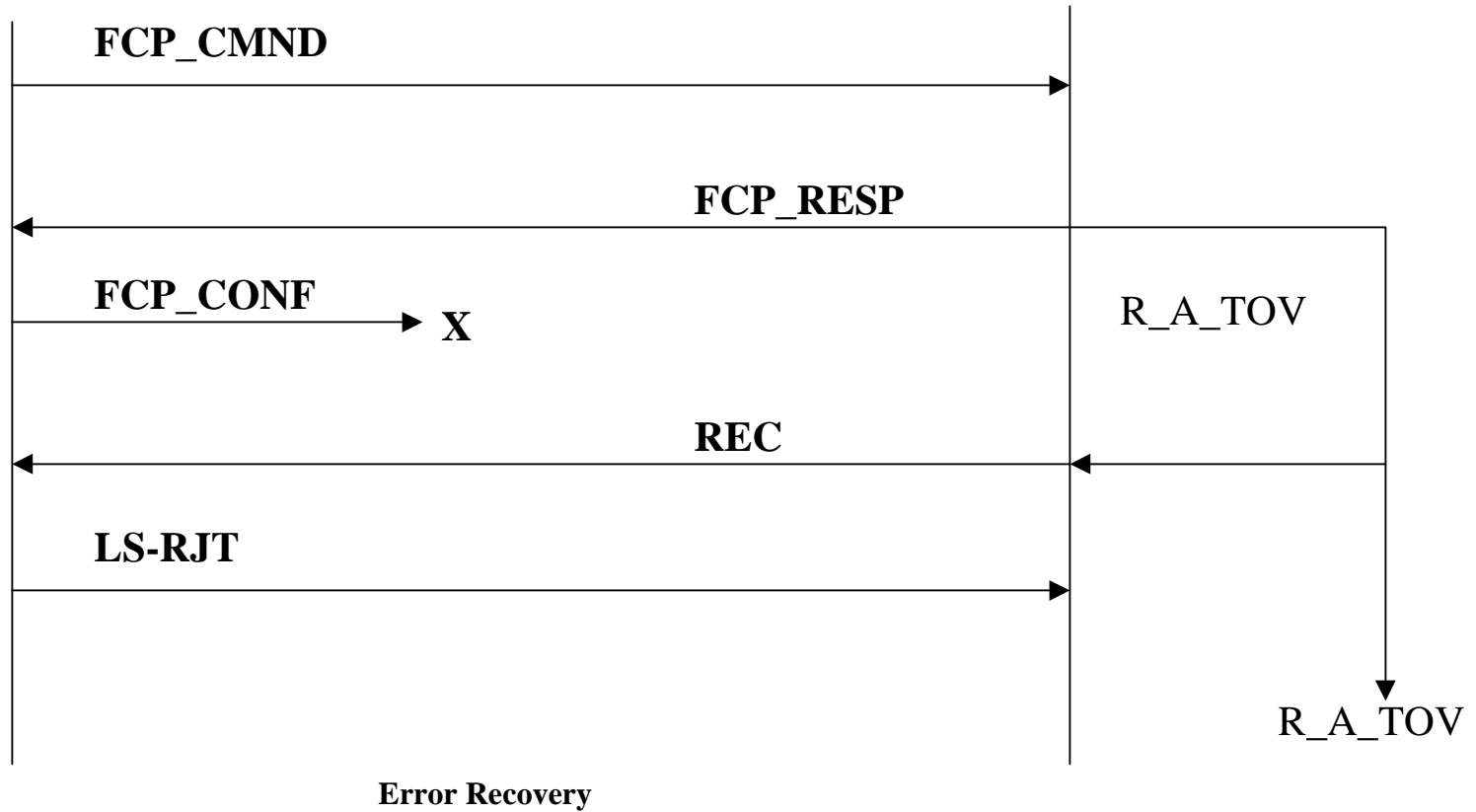
None.
LS-RJT implicitly indicates that the Initiator received FCP_RESP and sent FCP_CONF,
since no context exists for the Exchange.
The context for the Exchange in the Target must be preserved for another R_A_TOV to
prevent possible aliasing. If FCP_CONF is received after REC is transmitted, it is accepted
and discarded and the context for the Exchange can be purged.

D.?? Class 2 FCP_CONF Lost

FCP_CMND

ACK

FCP_RESP

ACK

FCP_CONF → **X**

E_D_TOV **ABTS**

ACK

BA_ACC

R_A_TOV **ACK**

RRQ

ACK
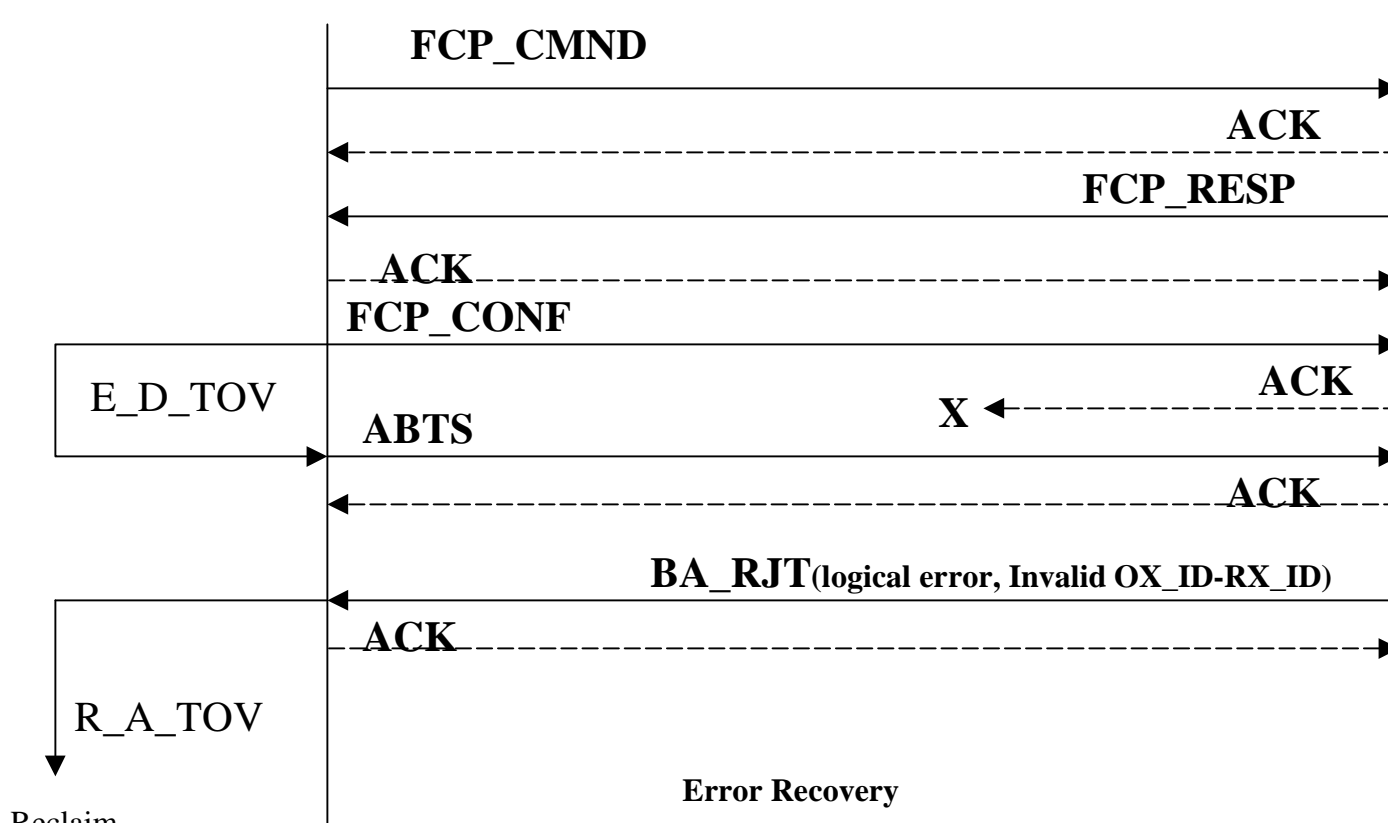
ACC

ACK

FCP_CONF *

ACK

Error Recovery

BA_ACC payload indicates that FCP_CONF was not received (low SEQ_CNT=0, not equal to high SEQ_CNT =1, SEQ_ID valid, SEQ_ID value =SEQ_ID of FCP_CMND)

FCP_CONF* must be sent with a different SEQ_ID

D.??? Class 2 ACK Lost on FCP_CONF

FCP_CMND

ACK

FCP_RESP

ACK

FCP_CONF

E_D_TOV

ACK

X ◄

ABTS

ACK

BA_RJT(logical error, Invalid OX_ID-RX_ID)

ACK

R_A_TOV

Error Recovery

Reclaim
Initiator
Recovery
Qualifier

None:
BA_RJT is the response to the ABTS since no context exists for this Exchange
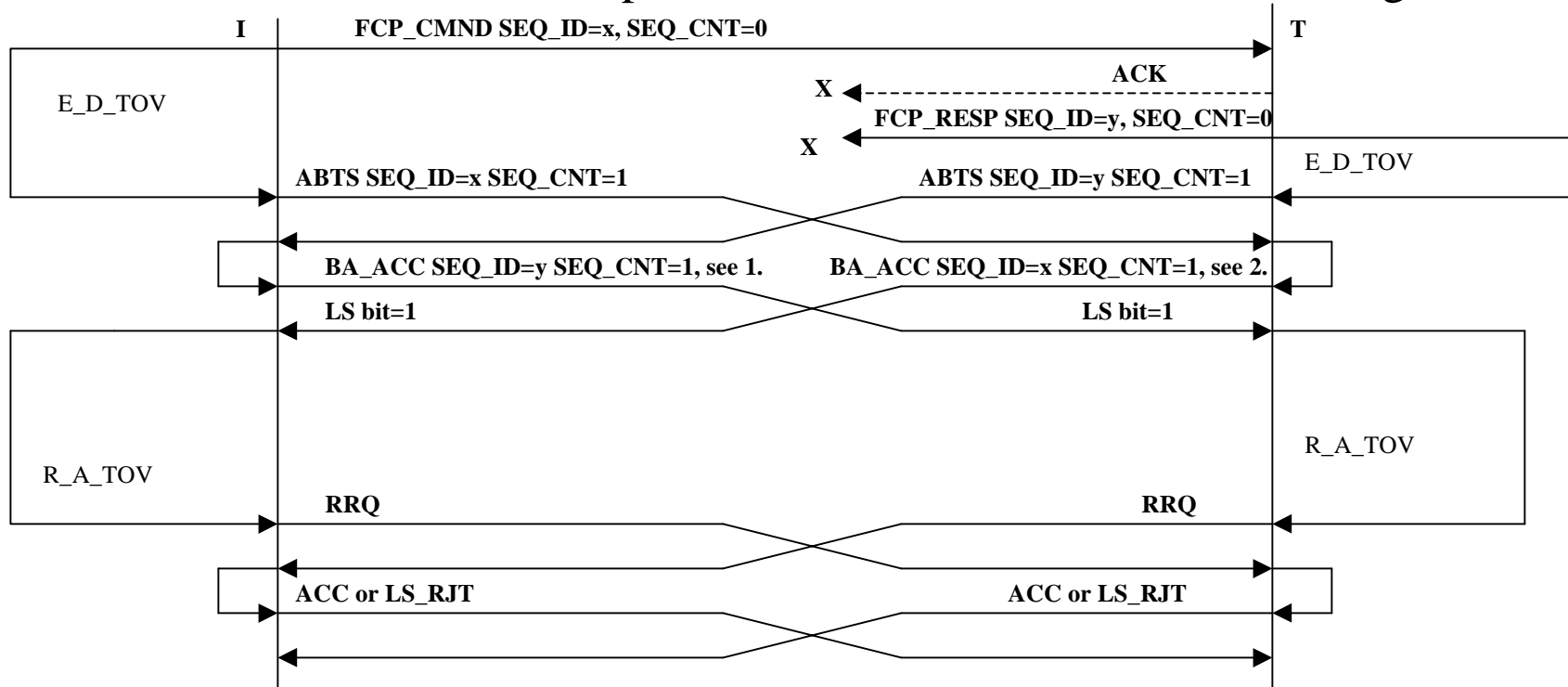and the ABTS was not issued on the first Sequence of a new Exchange.
The Initiator must establish a Recovery Qualifier on receipt of the BA_RJT.
The resources associated with the Recovery Qualifier can be reclaimed
when R_A_TOV expires. Note that the issuance of RRQ is optional as no
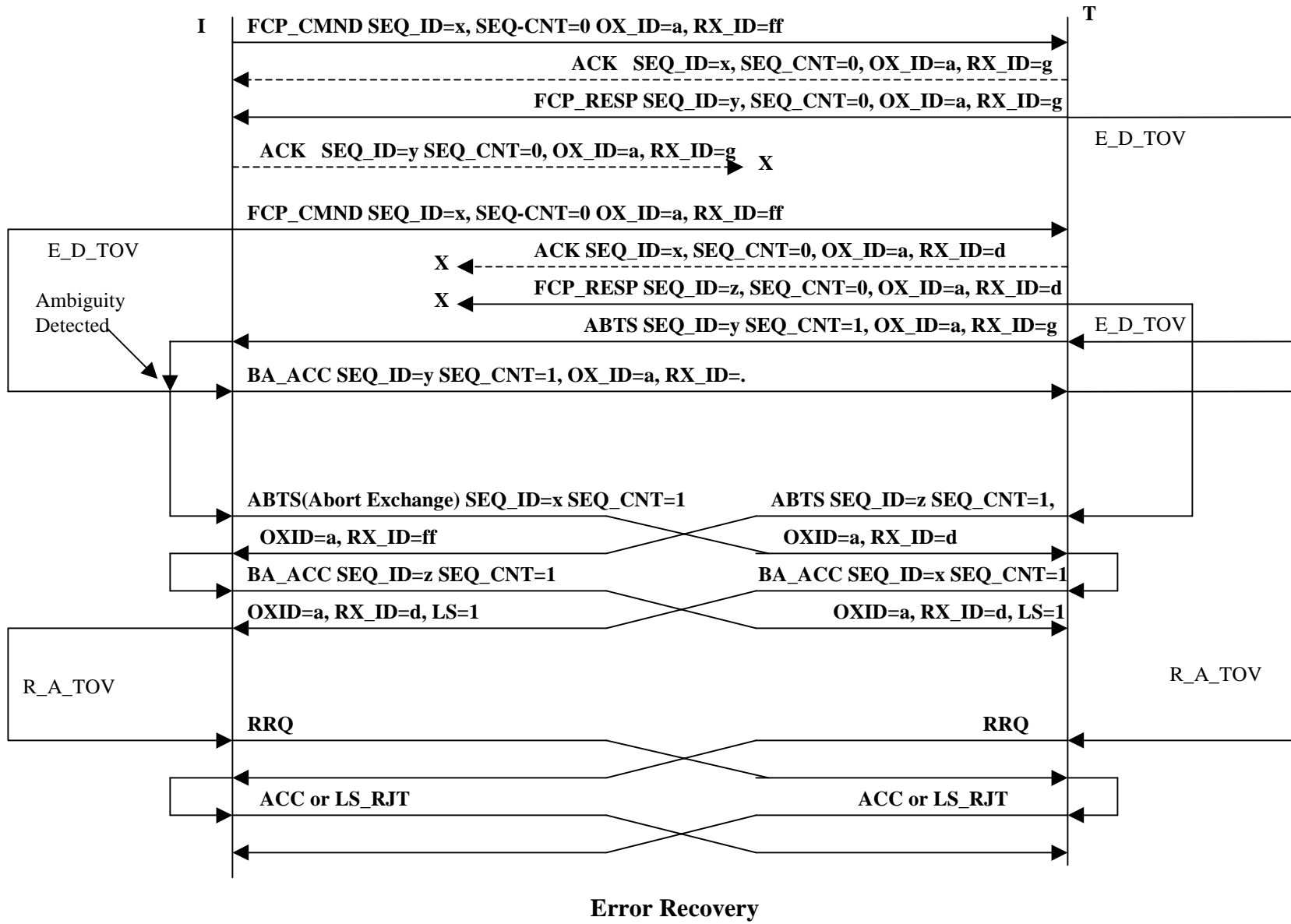Recovery Qualifier was established by the Target.

D.5? Class 2 Multiple Error Condition, Abort the Exchange

I | **FCP_CMND SEQ_ID=x, SEQ_CNT=0** | **T**

E_D_TOV

X ◄------------------------------- **ACK**

FCP_RESP SEQ_ID=y, SEQ_CNT=0

X

E_D_TOV

ABTS SEQ_ID=x SEQ_CNT=1   **ABTS SEQ_ID=y SEQ_CNT=1**

BA_ACC SEQ_ID=y SEQ_CNT=1, see 1.   **BA_ACC SEQ_ID=x SEQ_CNT=1, see 2.**

LS bit=1   **LS bit=1**

R_A_TOV

R_A_TOV

RRQ   **RRQ**

ACC or LS_RJT   **ACC or LS_RJT**

Error Recovery

Error recovery shall not be attempted if multiple errors, or the appearance of multiple errors, have occurred in an Exchange, such as both the ACK to command and the response to the command being lost. A multiple error arises when a Recovery Qualifier has been established and then either an ABTS is received or the recovery action indicates the need to send an ABTS. The Exchange shall be aborted by issuing ABTS(Abort Sequence) if called for in the recovery process or by setting the Last_Sequence bit to one in BA_ACC with payload of SEQ_ID Validity = invalid, Low SEQ_CNT=hex '0000', High SEQ_CNT= hex 'FFFF', or both.

D.5?? Mutliple Error Condition, Exchange Ambiguities, Abort the Exchange

I | FCP_CMND SEQ_ID=x, SEQ-CNT=0 OX_ID=a, RX_ID=ff   T

ACK   SEQ_ID=x, SEQ_CNT=0, OX_ID=a, RX_ID=g

FCP_RESP SEQ_ID=y, SEQ_CNT=0, OX_ID=a, RX_ID=g

E_D_TOV

ACK   SEQ_ID=y SEQ_CNT=0, OX_ID=a, RX_ID=g   X

FCP_CMND SEQ_ID=x, SEQ-CNT=0 OX_ID=a, RX_ID=ff

E_D_TOV   ACK SEQ_ID=x, SEQ_CNT=0, OX_ID=a, RX_ID=d   X

FCP_RESP SEQ_ID=z, SEQ_CNT=0, OX_ID=a, RX_ID=d   X

Ambiguity   ABTS SEQ_ID=y SEQ_CNT=1, OX_ID=a, RX_ID=g   E_D_TOV
Detected

BA_ACC SEQ_ID=y SEQ_CNT=1, OX_ID=a, RX_ID=.

ABTS(Abort Exchange) SEQ_ID=x SEQ_CNT=1   ABTS SEQ_ID=z SEQ_CNT=1,

OXID=a, RX_ID=ff   OXID=a, RX_ID=d

BA_ACC SEQ_ID=z SEQ_CNT=1   BA_ACC SEQ_ID=x SEQ_CNT=1

OXID=a, RX_ID=d, LS=1   OXID=a, RX_ID=d, LS=1

R_A_TOV   R_A_TOV

RRQ   RRQ

ACC or LS_RJT   ACC or LS_RJT

Error Recovery

Reuse of Exchange IDs can produce ambiguities. In the event of multiple errors, the current Exchange shall be aborted.

1. The Initiator shall not act on the ABTS until either the ACK to FCP_CMND is received or E_D_TOV expires.
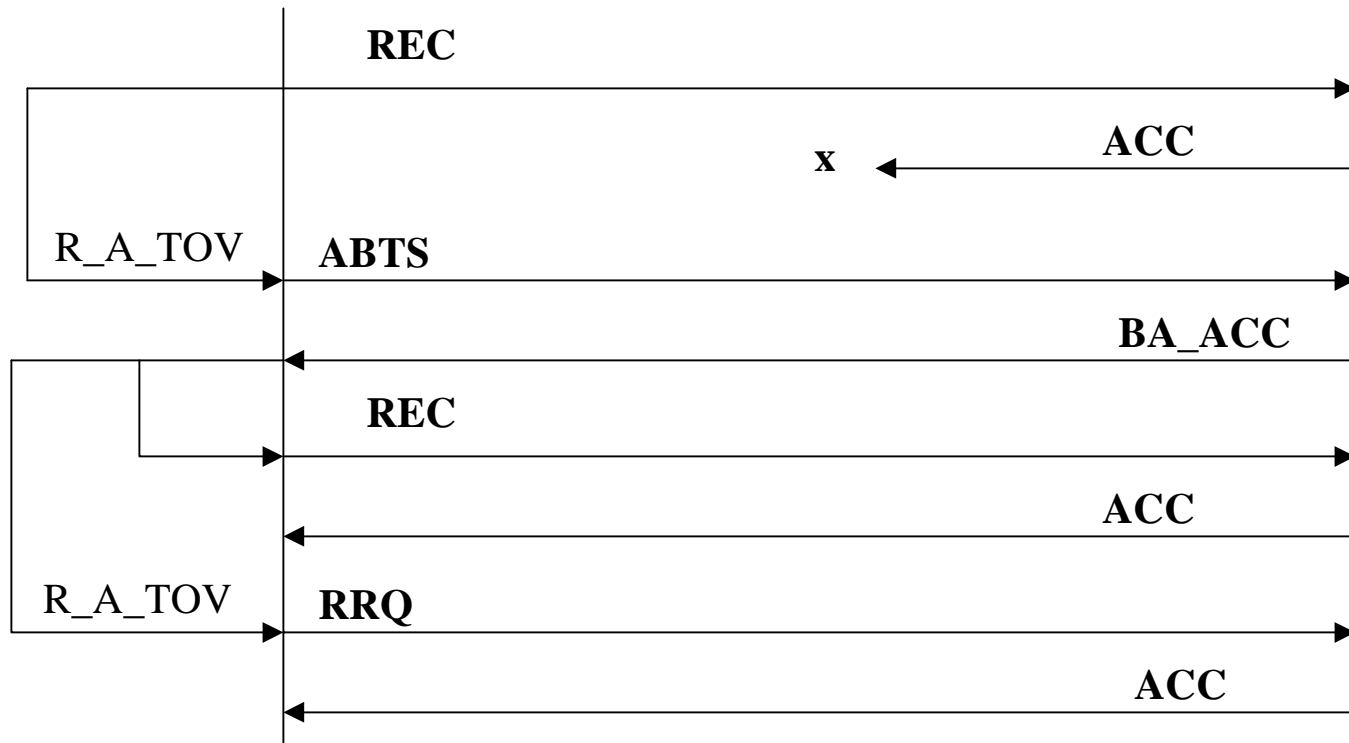
D.13 Cl 3, REC Lost

REC

x

R_A_TOV

ABTS

BA_ACC

REC

ACC

R_A_TOV

RRQ

ACC
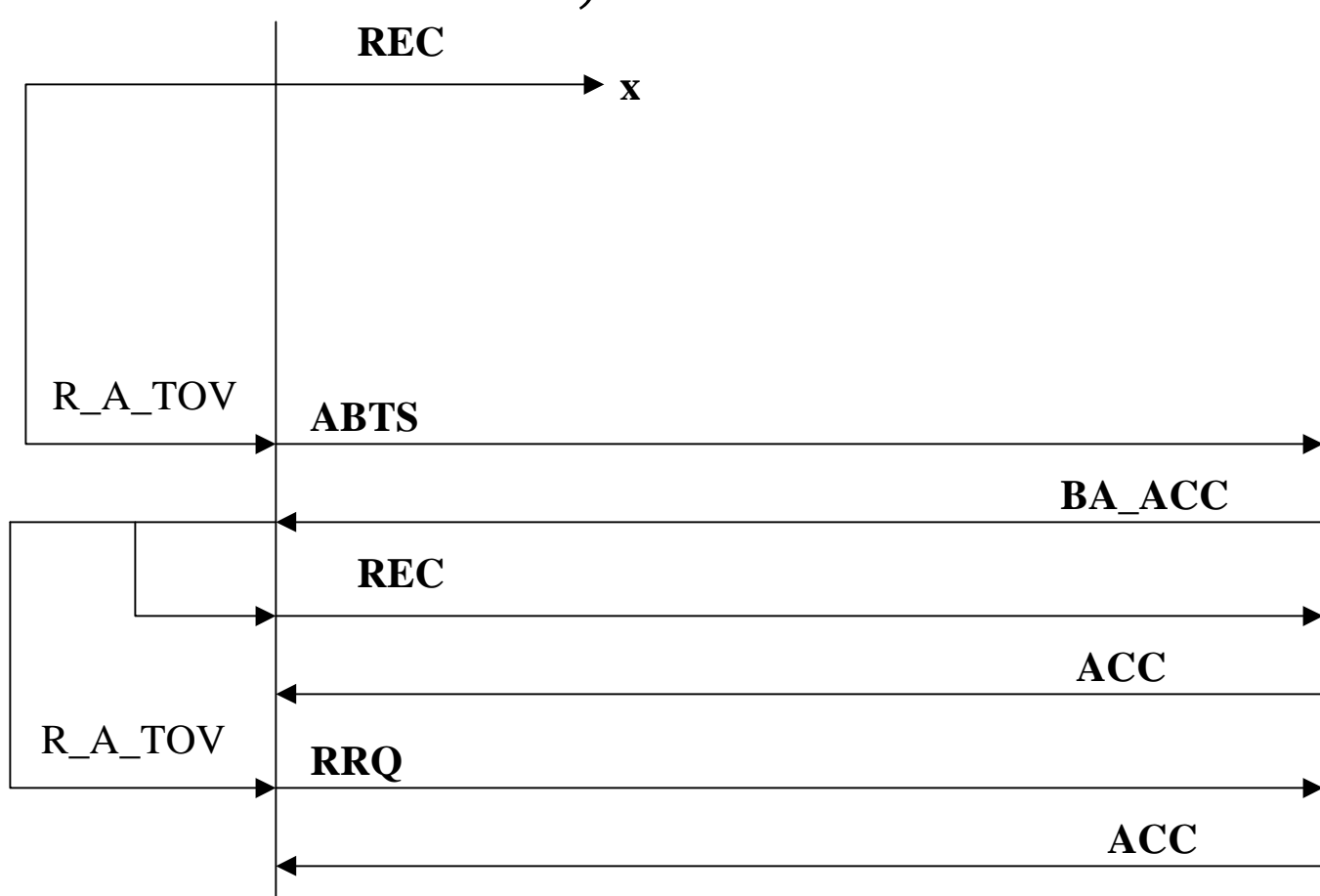
Since the REC was never received by the Target, the
BA_ACC payload is SEQ_ID invalid, low SEQ_CNT =0, high SEQ_CNT=
SEQ_CNT of ABTS=1. Recovery qualifiers are established on both sides
The second REC must be issued in a new Exchange

Change E_D_TOV in the text to R_A_TOV in12.6.2

D.13a Cl 3, REC Response Lost

Since the Target has already responded with ACC then no context exists for the Exchange. The Target would view the ABTS as being issued on a new Exchange, establish a Recovery Qualifier and respond with BA_ACC (Payload is SEQ_ID invalid, low SEQ_CNT=high SEQ_CNT= SEQ_CNT of ABTS). Since REC does not change any state, it can be reissued unconditionally. The second REC must be issued in a new Exchange.

Change E_D_TOV in the text to R_A_TOV in 12.6.2

D.13? Cl 3, Two RECs Lost, Abort the original Exchange
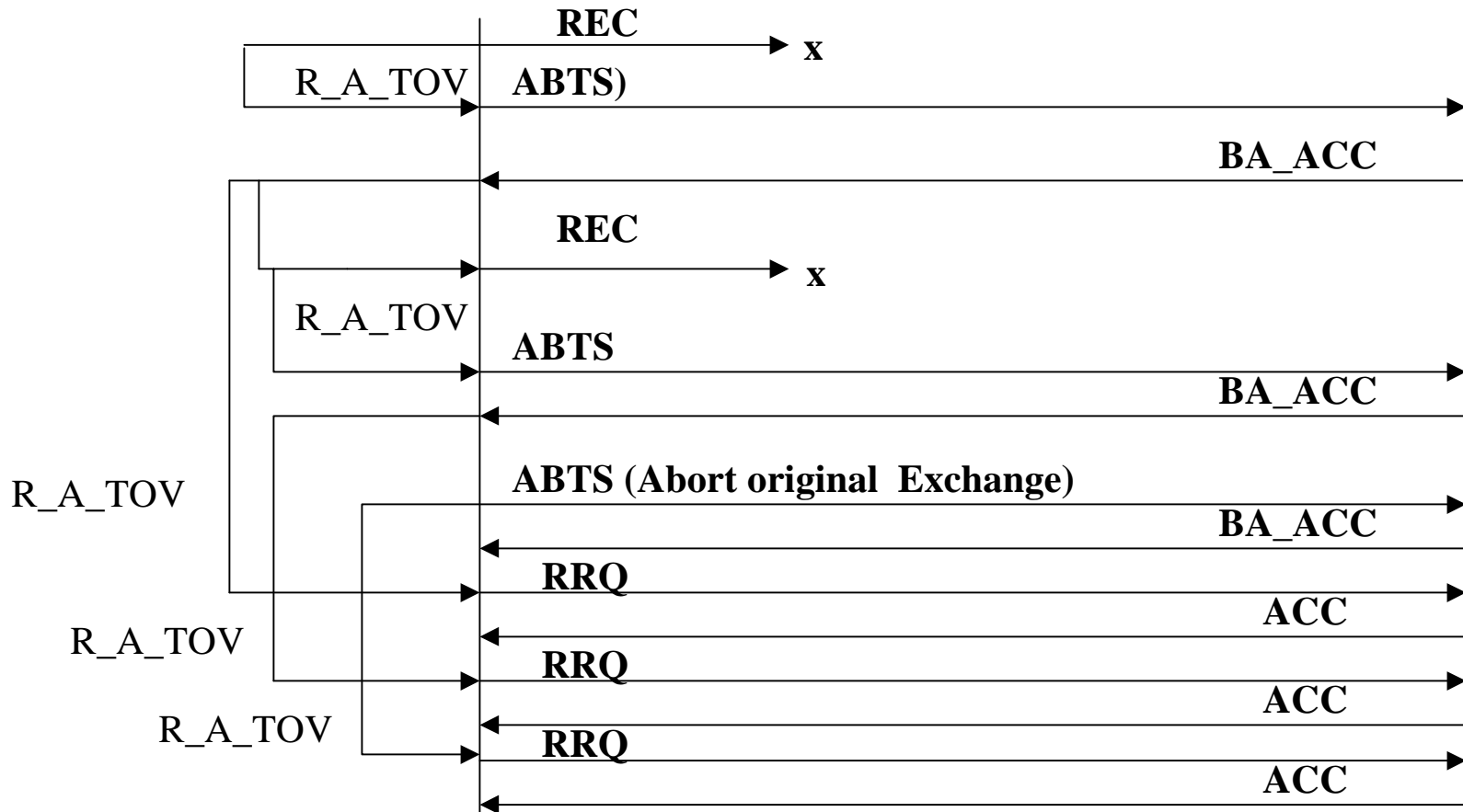
The failure of two RECs issued against the same Exchange, causes all associated Exchanges to be be aborted. The ABTS for the original Exchange uses the previous SEQ_ID and a SEQ_CNT one greater than the count used in the previous Sequence and Bit 0 = 0 set in the Parameter Field. The Payload for the BA_ACC is: SEQ_ID Validity =valid, SEQ_ID = SEQ_ID of last deliverable Sequence received, low SEQ_CNT=hex '0000" and high SEQ_CNT=hex 'FFFF'.
Recovery Qualifiers are established on both sides for each Exchange
(In the text 12.6.2. add sentence specifying that if two RECs fail, the original Exchange is aborted, regardless of Class.)

(Out of context: Need to put somewhere in the text, that any ABTS or RRQ failure causes
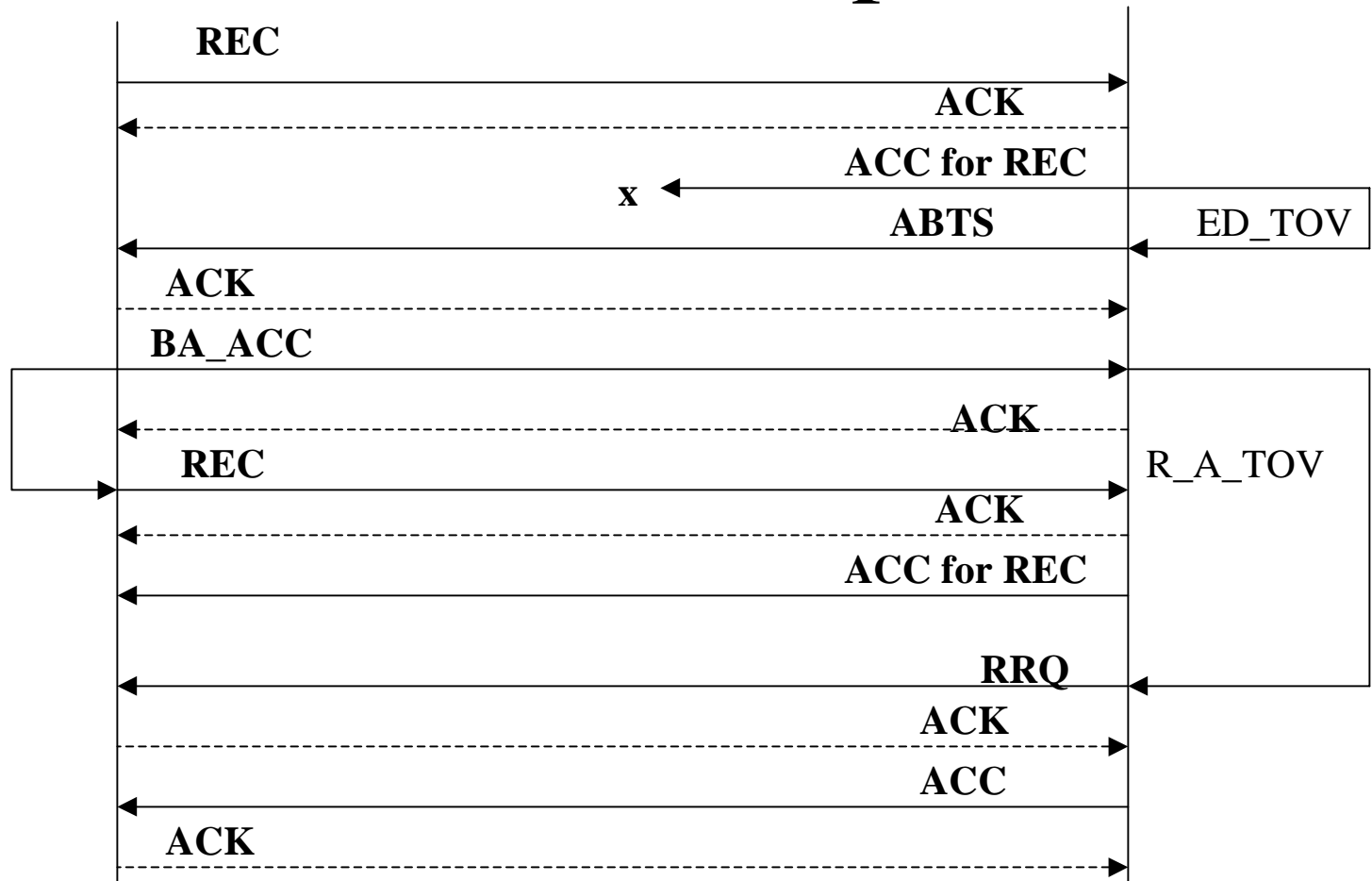*the Exchange to be aborted.)*

D.13b Class 2, REC Lost

REC

x

ED_TOV ABTS

ACK

BA_ACC

ACK

REC

ACK

ACC

ACK

R_A_TOV RRQ

ACK

ACC

ACK
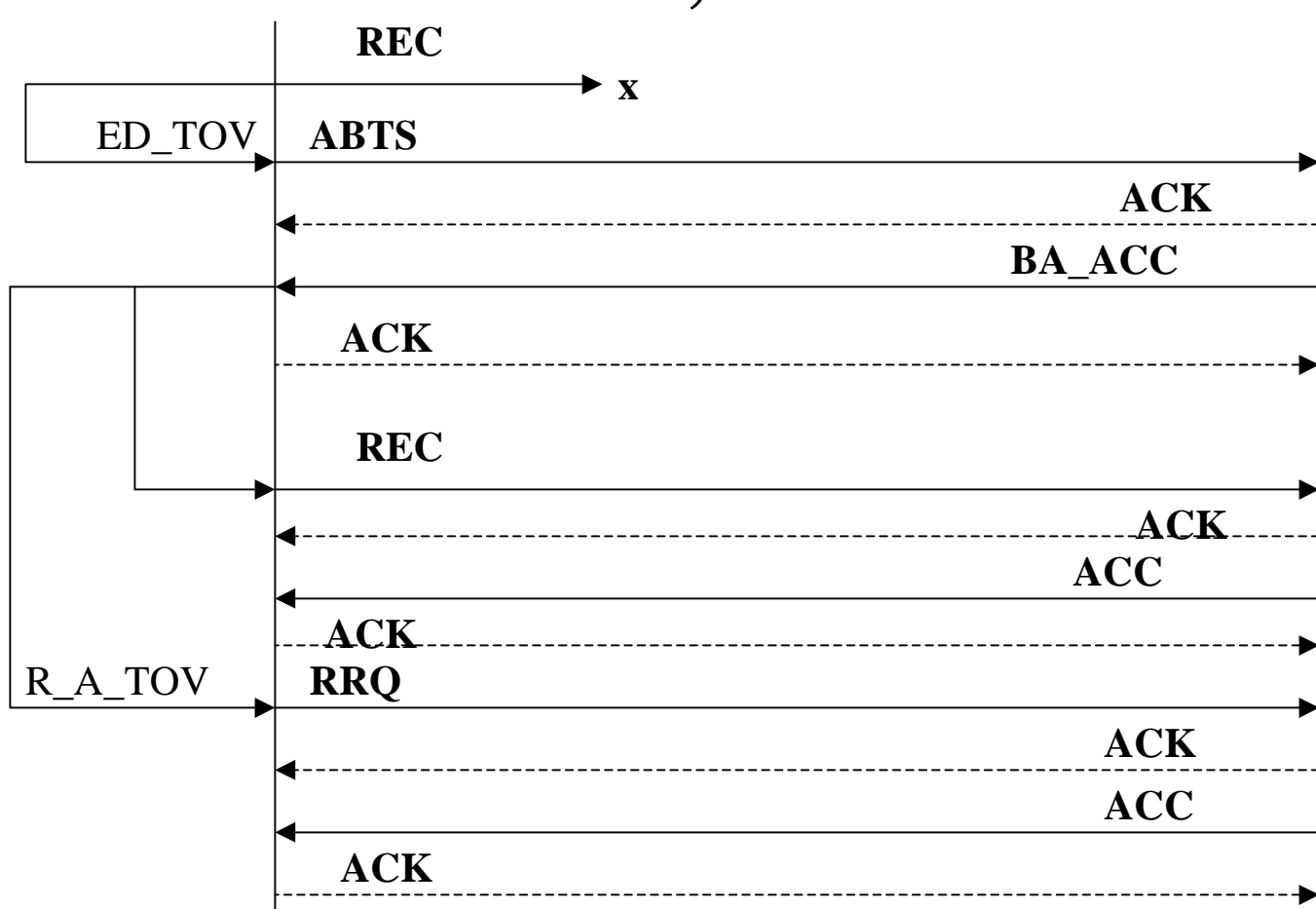
The BA_ACC payload indicates that the REC was never received by the Target (Payload is SEQ_ID invalid, low SEQ_CNT=0, high SEQ_CNT= SEQ_CNT in ABTS frame) Recovery Qualifiers are established on each side.
The second REC is issued using a new Exchange.

D.13c Class 2, REC Response Lost

REC

ACK

ACC for REC
x

ABTS   ED_TOV

ACK

BA_ACC

ACK

REC   R_A_TOV

ACK

ACC for REC

RRQ

ACK

ACC

ACK

Note: The Target determines that the ACC was never received by the Initiator;
BA_ACC payload is SEQ_ID invalid, Low SEQ_CNT=0, High SEQ_CONT=
SEQ_CNT of ABTS. After responding to the ABTS, the Initiator reissues the
REC in a new Exchange.

D.14a Class 3, SRR Lost, Abort the Original Exchange

SRR → x

2* R_A_TOV

ABTS

BA_ACC

ABTS (Abort original Exchange)

R_A_TOV

BA_ACC

RRQ

R_A_TOV

ACC

RRQ

ACC
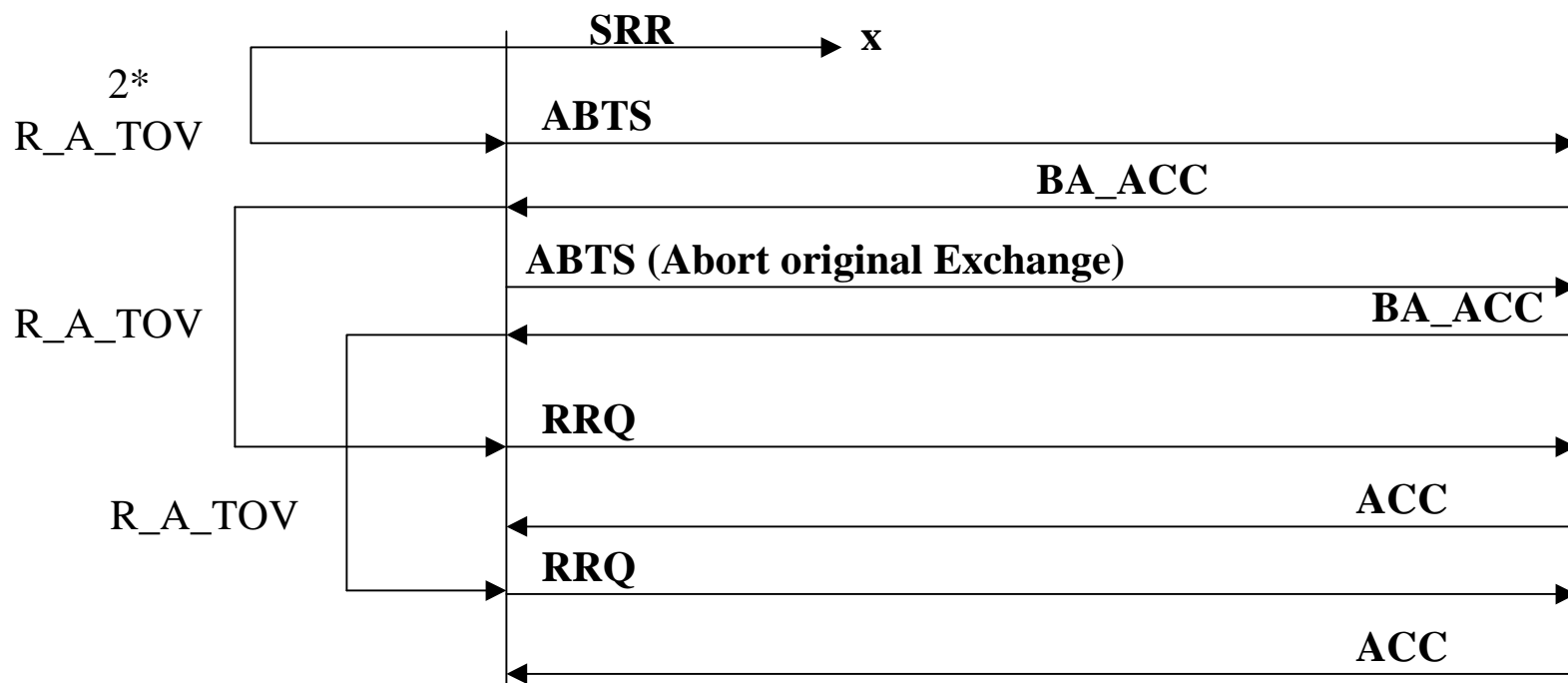
Note:The SRR BA_ACC Payload is: SEQ_ID Validity = valid, low SEQ_CNT=0, high SEQ_CNT = SEQ_CNT of the ABTS frame.

The ABTS for the original Exchange uses the previous SEQ_ID and a SEQ_CNT one greater than the count used in the previous Sequence and Bit 0 = 0 set in the Parameter Field. The Payload for the BA_ACC is: SEQ_ID Validity =valid, SEQ_ID=SEQ_ID of last deliverable Sequence received, low SEQ_CNT=hex '0000" and high SEQ_CNT=hex 'FFFF'.

Recovery Qualifiers are established on both sides for each Exchange.

(In the text in 12.6.3, add a note that for Class 3 that an SRR failure causes the original Exchange to be aborted. In Class 2, 2 failures of SRR cause the original Exchange to be aborted
Change E_D_TOV in the text to 2*R_A_TOV to agree with the text in 12.6.3)
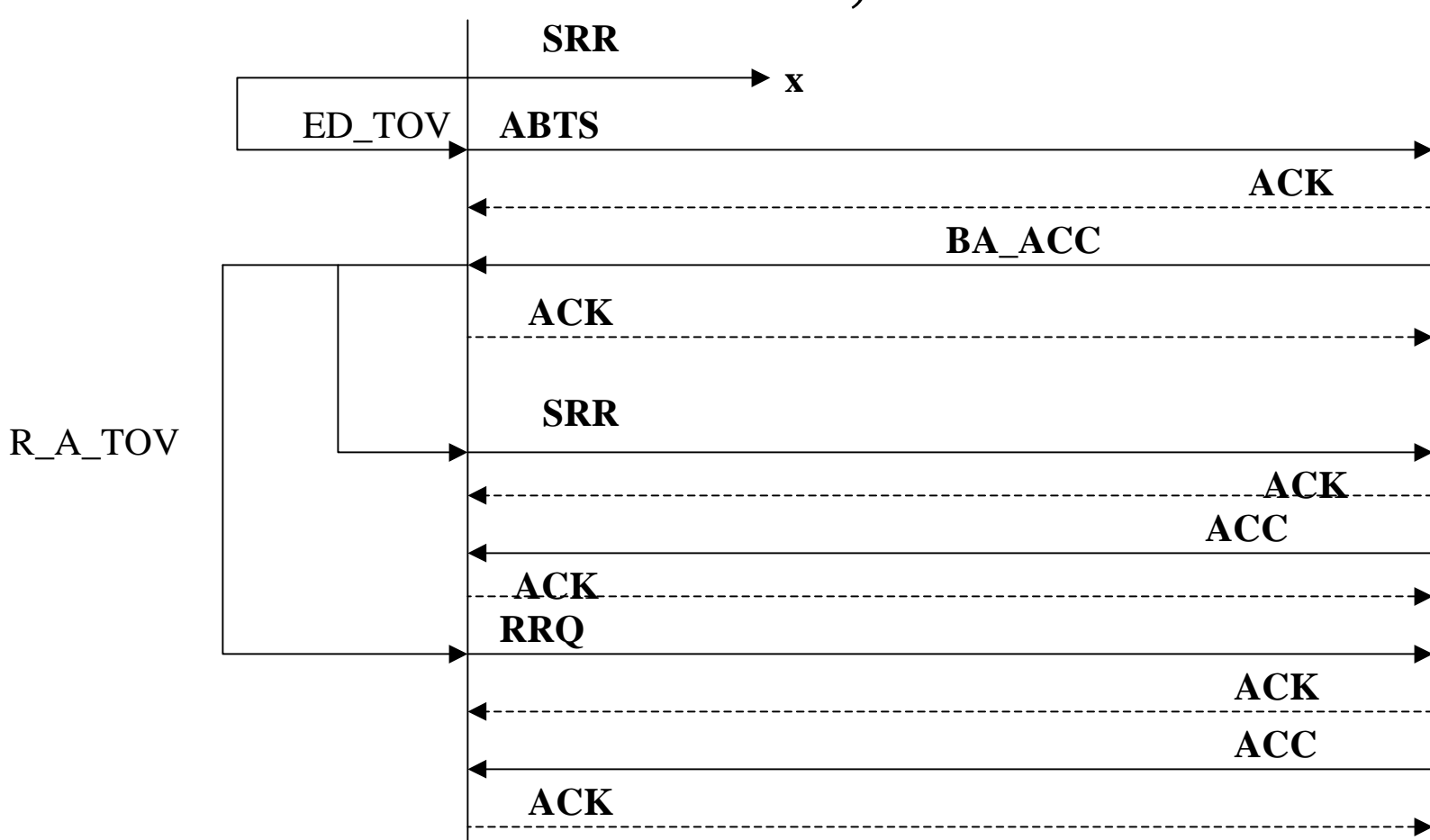
D.14b Class 3, SRR Response Lost
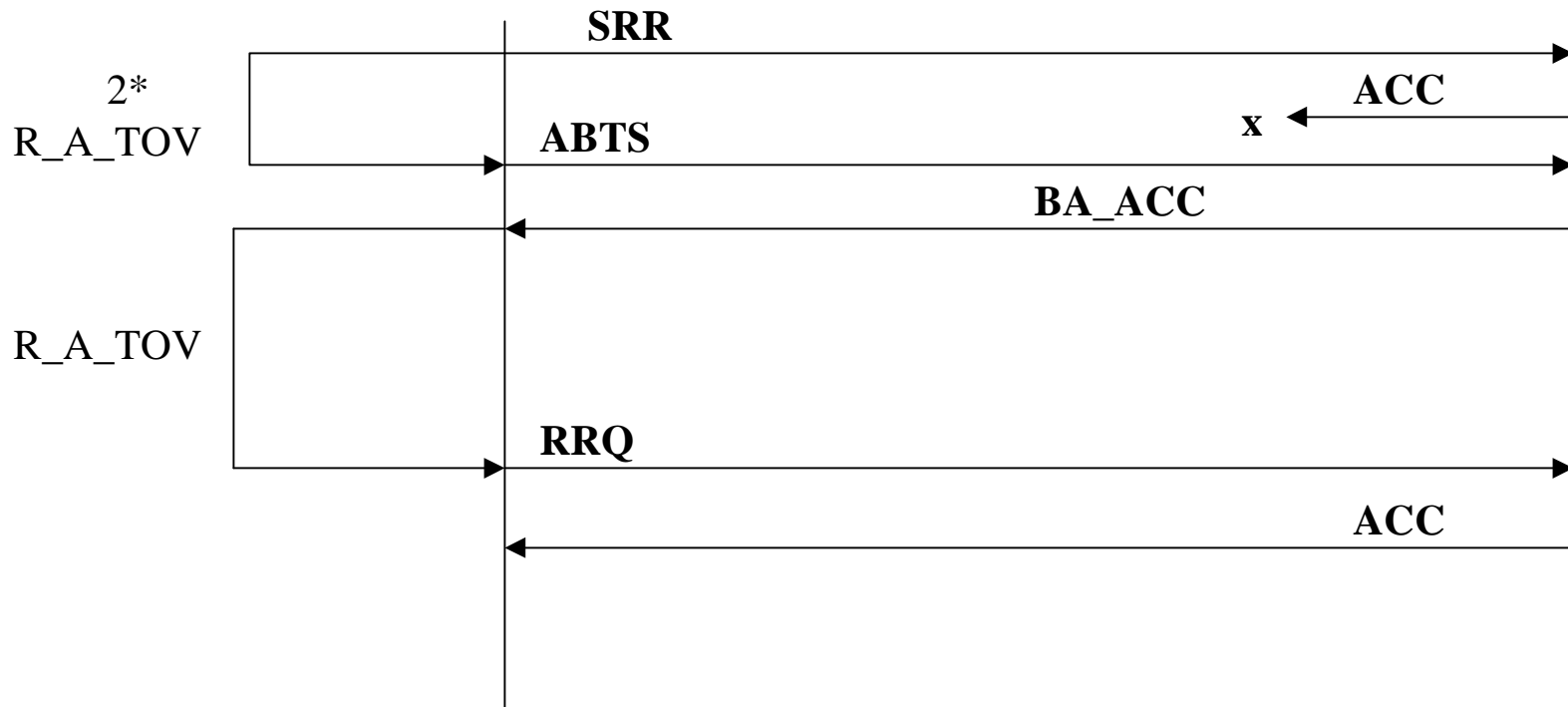
Note: If the Exchange is unknown to the Recipient (Exchange completed and context purged) the Payload for the BA_ACC is : low SEQ_CNT =hex'0000', high SEQ_CNT =hex 'FFFF, SEQ_ID Validity = invalid. Recovery Qualifiers are established on both sides. Otherwise if the Recipient has knowledge of the SRR(last deliverable Sequence), the Payload is SEQ_ID Validity = valid, and SEQ_ID = SEQ_ID of the SRR, low SEQ_CNT =high SEQ_CNT = SEQ_CNT of the ABTS frame. Since no Recovery Qualifier is established on the Target side, RRQ need not be issued. The Recovery Qualifier is established on the Initiator side and must be timed out for R_A_TOV.
 In either case **the original Exchange need not be aborted.**
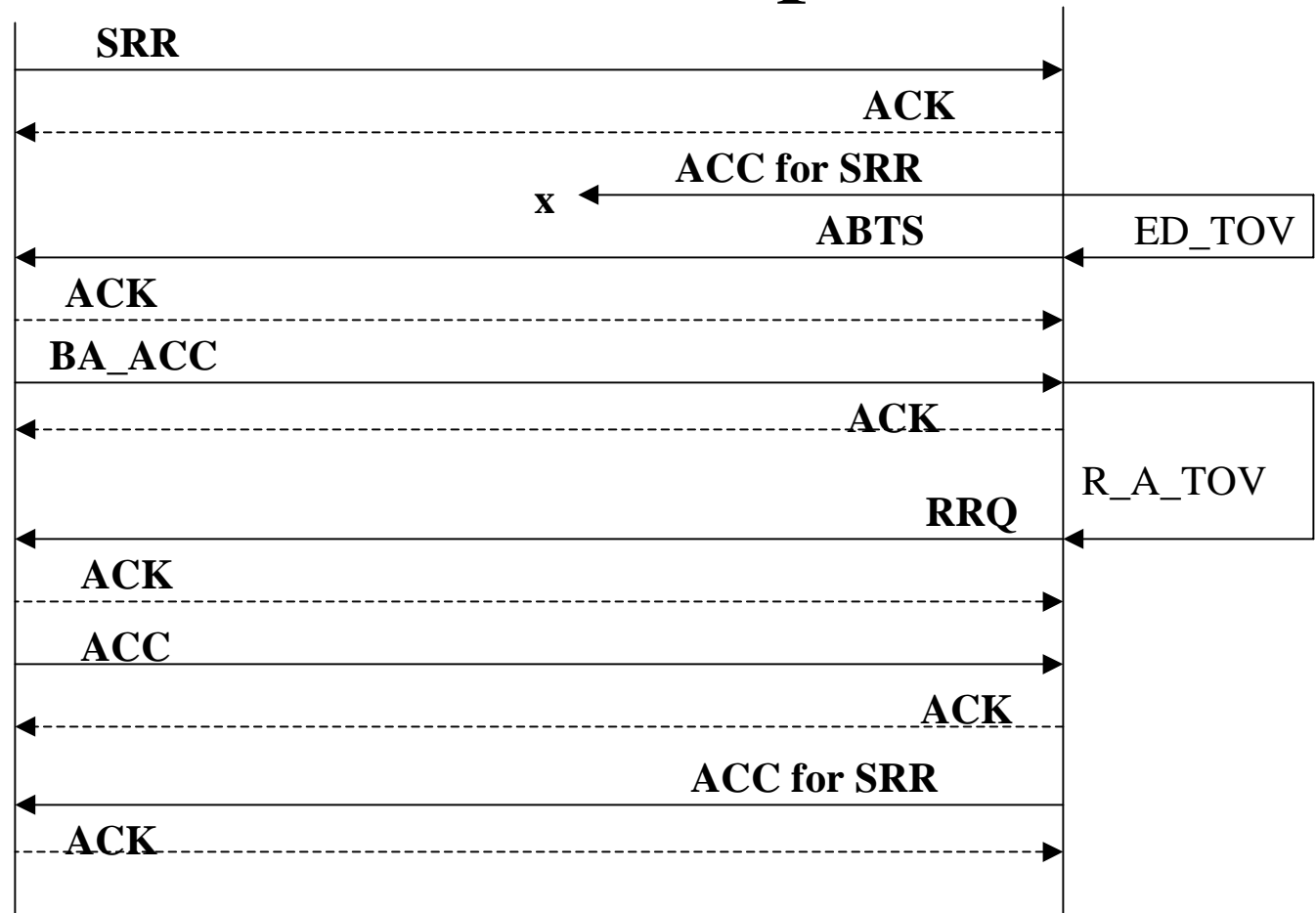.
{Note out of context: If the response to SRR is LS_RJT, then the original Exchange shall be aborted or treated as if it had been aborted by the Exchange Initiator, if it is still active--add to 8.3text, hard copy page 33.})

D.14c Class 2, SRR Lost

Since this is an ABTS on a new Exchange, Recovery Qualifiers
must be established. BA-ACC indicates Invalid SEQ_ID, low
SEQ_CNT= 0 and high SEQ_CNT = SEQ_CNT of the ABTS. The second
SRR is issued using a new Exchange.

D.14d Class 2, SRR Response Lost

SRR

ACK

ACC for SRR

x

ABTS

ED_TOV

ACK

BA_ACC

ACK

R_A_TOV

RRQ

ACK

ACC

ACK

ACC for SRR

ACK

Note: The BA_ACC payload indicates SEQ_ID invalid, low SEQ_CNT=0
and high SEQ_CNT=SEQ_CNT of the ABTS, which indicates that
the ACC for SRR was not received and will be discarded if it is received.
Recovery Qualifiers are established on both sides. The ACC for SRR is
issued with a new SEQ_ID.