TO: T10 Membership
FROM: Paul A. Suhler, Certance
DATE: 19 November 2003
SUBJECT: T10/03-367r4, ADT Annex: Error detection and recovery action examples

Revision 4:

- Changes from 17 November working group teleconference (T10/04-002r0)

Revision 3:

- Changes from 3 November working group meeting (T10/03-378r0)

Revision 2:

- Various changes for 3 November working group meeting

Revision 1:

- Changes to clarify transmission of multiple IUs without receiving Acknowledgement IUs
- Fixed incorrect labels
- Added P4:Initiating Recovery state and added labels for sender actions.
- This is not entirely consistent with 03-355r2, which does not yet show a P3:P2 transition upon receipt of an Initiate Recovery IU.

Revision 0:

- Initial version

1.1 Background

In the process of preparing proposals for error detection and recovery, we found that drawings made comprehension easier. This document offers some such as an informative annex for ADT.

1.2 Changes

Add the following as an Annex.

Annex A

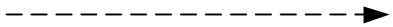
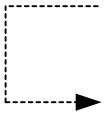
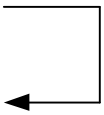
(Informative)

Error detection and recovery action examples

A.1 Introduction

This informative annex diagrams various error detection and recovery procedures for ADT devices conforming to this standard. The conventions for the diagrams are shown in Table A.1.

Table A.1 – Diagram Drawing Conventions

Drawing Convention	Meaning
	Acknowledged or Unacknowledged IU
	IU with error
	Acknowledgement IU
IU Request(A)	Service request from upper layer of sender
IU Indication(A)	Service indication to upper layer of receiver
IU Confirmation(A)	Service confirmation to upper layer of sender
	Time-out value exceeded
	IU received is processed to transmit IU
X	IU lost or dropped
PR	Value of the Pending Recovery (PR) bit in the NAK IU.
EFN	Value of the port's Expected Frame Number counter
NFTS	Value of the port's Next Frame To Send counter.
a = b	Counter a is set to the value of expression b
(a == b)	Expression a equals expression b
(a != b)	Expression a does not equal expression b
(condition) => action	Because the condition is true, the action is performed

A.2 Receiver-detected retryable error

Figure A.1 shows the detection of a retryable error by the receiver and the subsequent recovery.

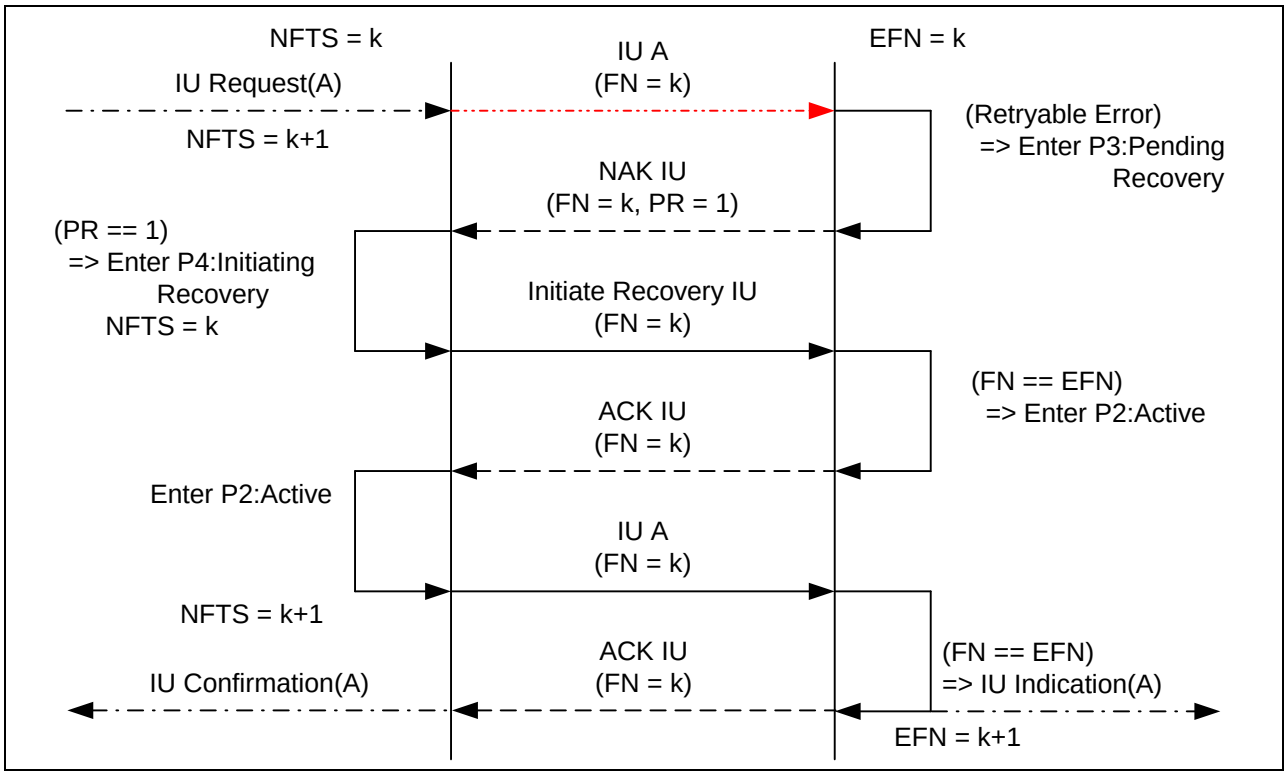


Figure A.1 – Receiver-detected retryable error

A.3 Receiver-detected retryable error with multiple active IUs

Figure A.2 shows the detection of a retryable error by the receiver, when the IU in error is followed by a good IU.

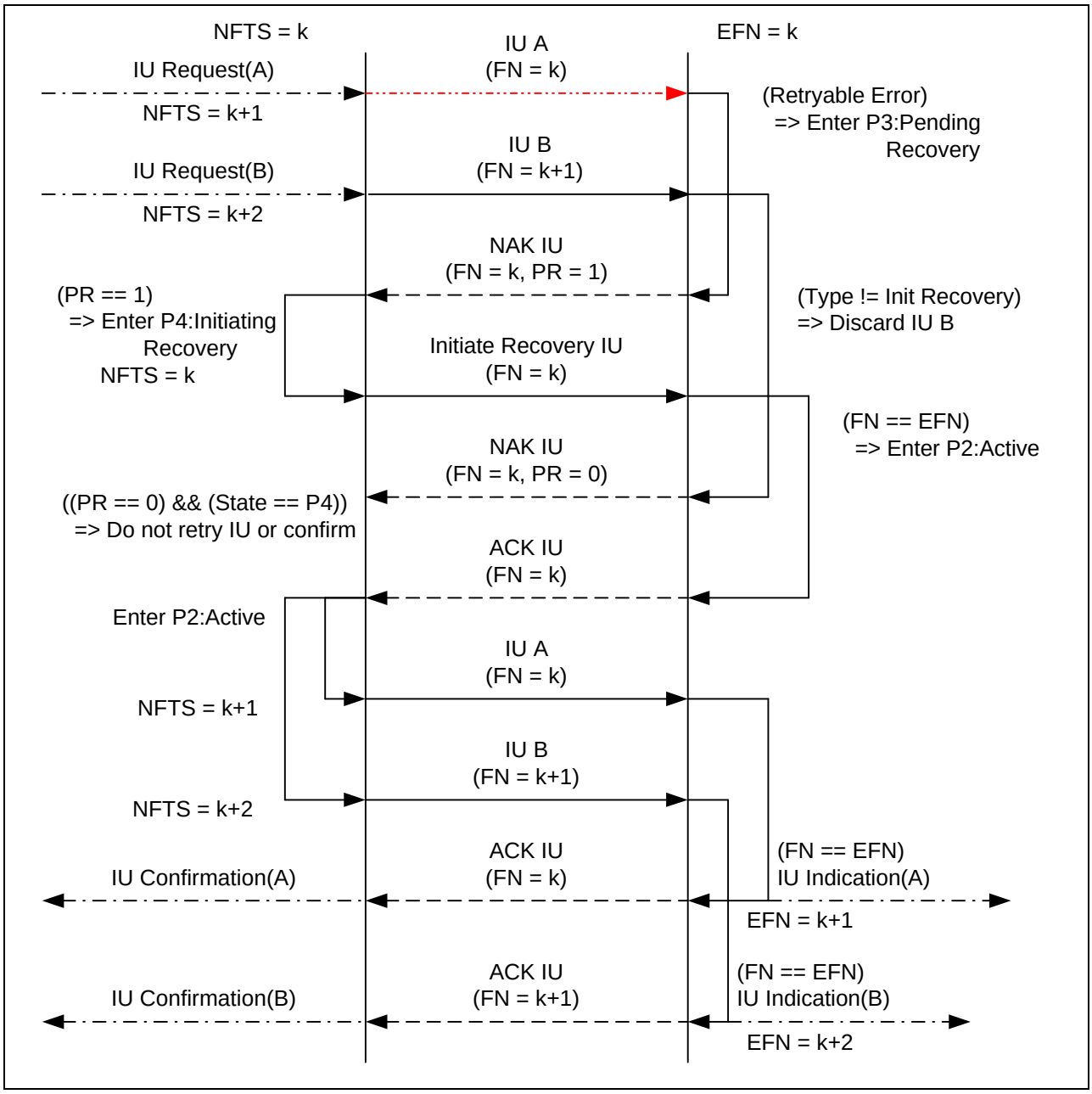


Figure A.2 – Receiver-detected retryable error with multiple active IUs

A.4 Lost IU with no further traffic

Figure A.3 shows a lost IU (e.g., bad checksum), in which there is not further traffic from the sender. The sender detects the error when a timeout occurs without receipt of an Acknowledgement frame.

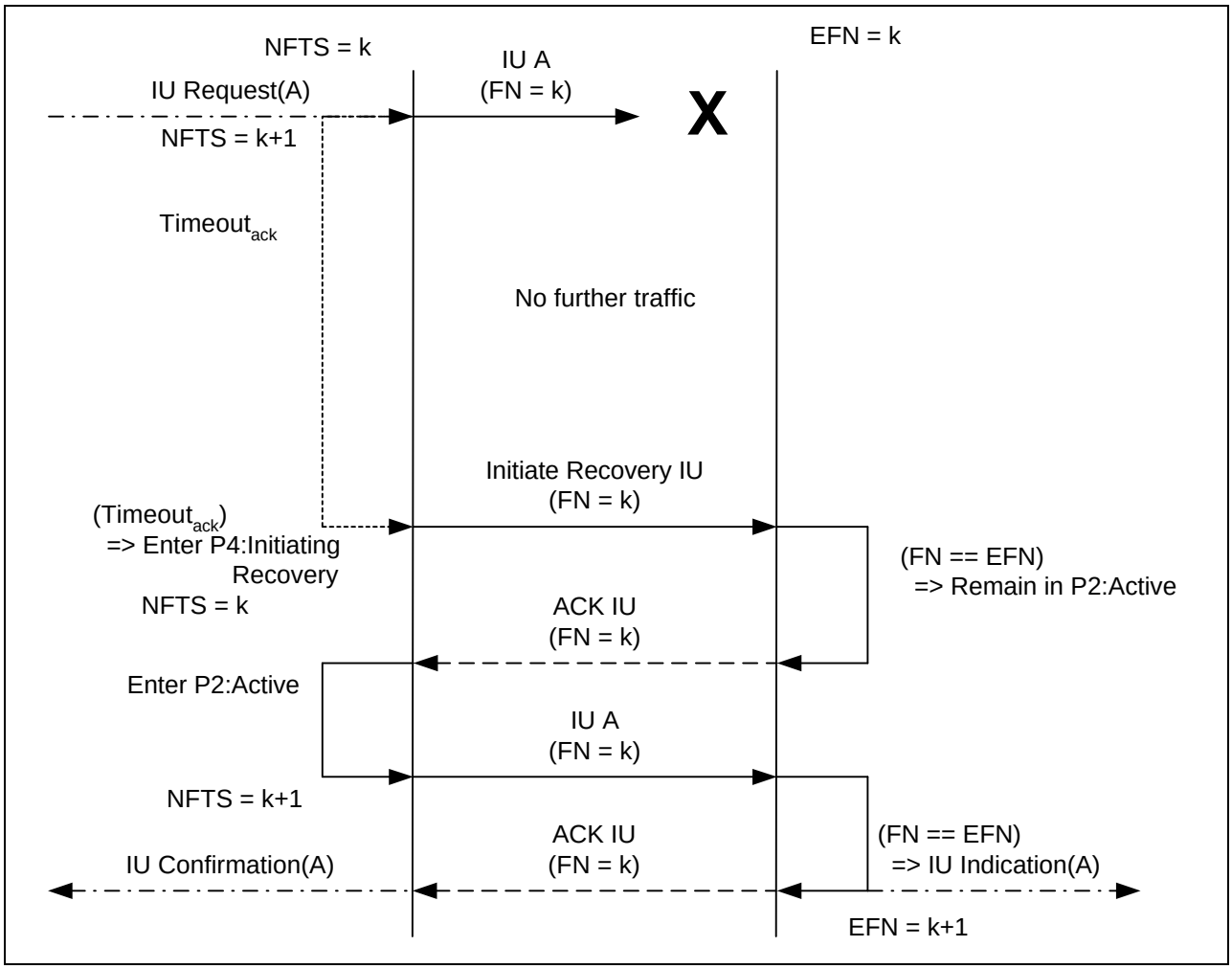


Figure A.3 – Lost IU with no further traffic

A.5 Lost ACK with recovery driven by out-of-order ACK

Figure A.4 illustrates how the requirement for Acknowledgement IUs to be sent in the same order as the original IUs are received allows detection of the missing acknowledgement for FN k. There is no need to wait for timer expiration.

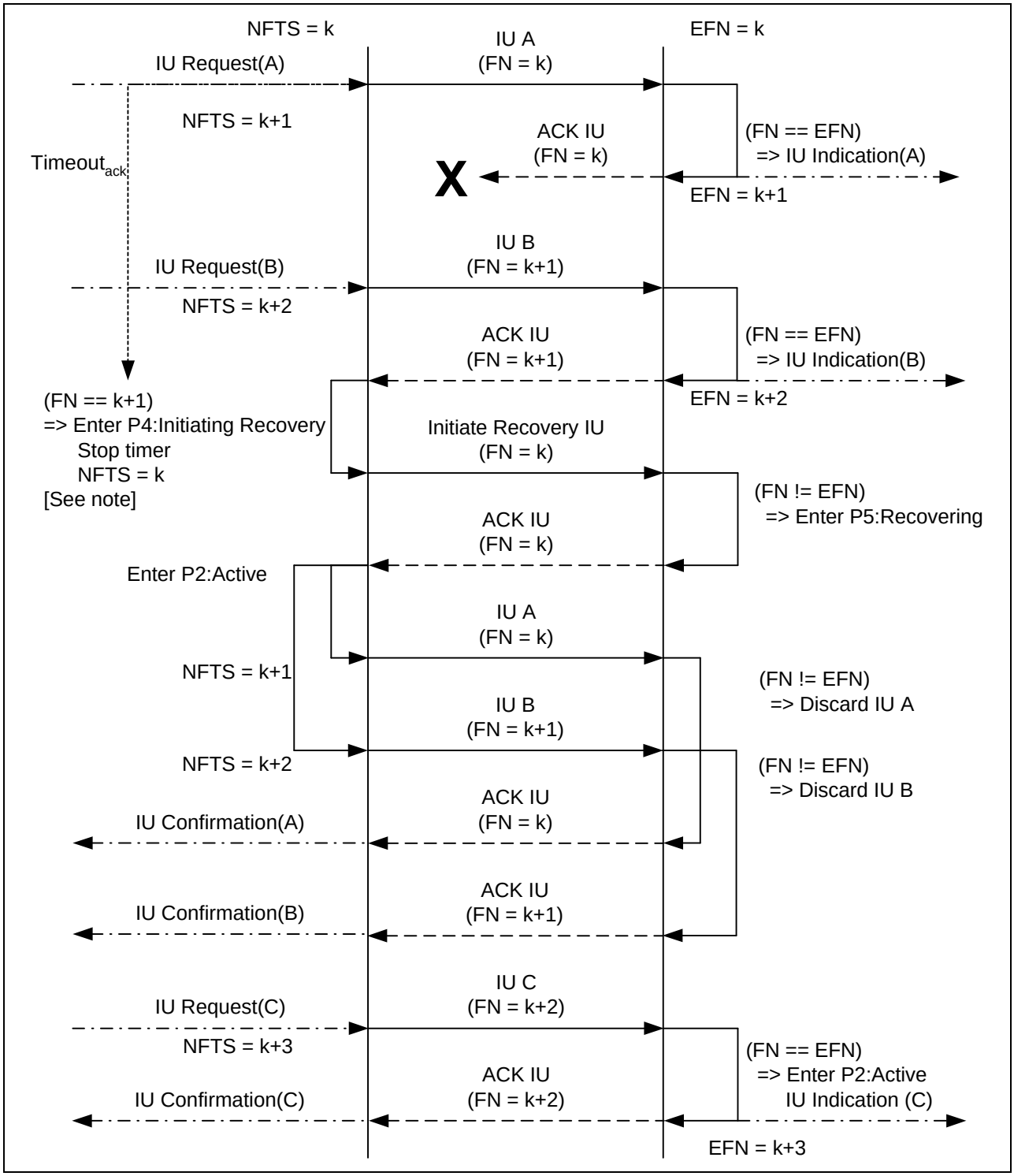


Figure A.4 – Lost ACK with recovery driven by out-of-order ACK

A.6 Lost IU with recovery driven by out-of-order NAK

Figure A.5 is similar to the previous one, but the second IU receives a NAK instead of an ACK. Again, there is no need to wait for timer expiration.

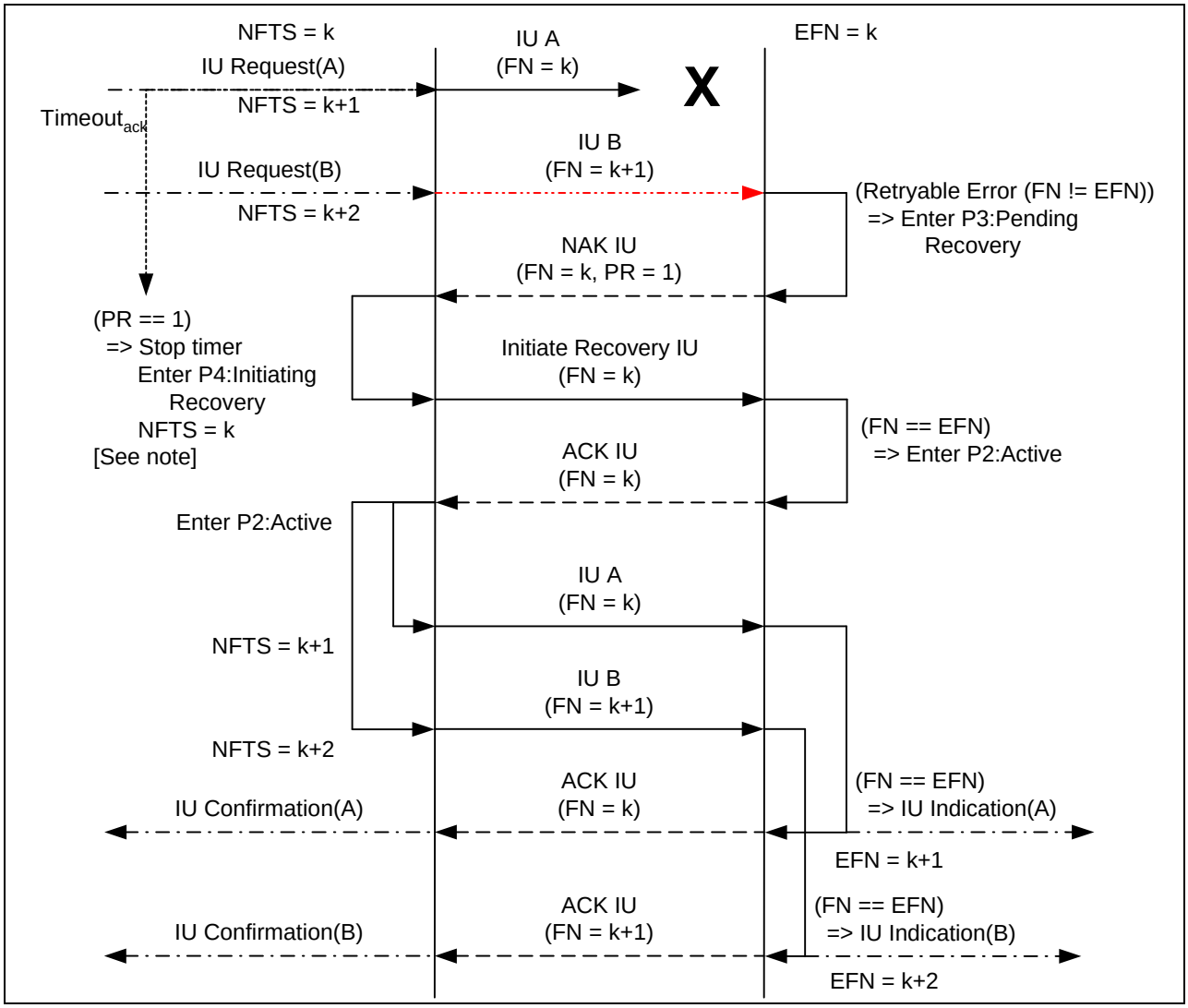


Figure A.5 – Lost IU with recovery driven by out-of-order NAK

A.7 Lost NAK with recovery driven by timeout

In the example in Figure A.6 – unlike the previous ones – the sender does not use an out-of-order Acknowledgement IU to infer that an earlier Acknowledgement IU was lost. Instead, it waits for the $Timeout_{ack}$ on the earlier Acknowledgement IU.

This diagram would also apply similarly if IU A received an ACK instead of a non-retryable NAK.

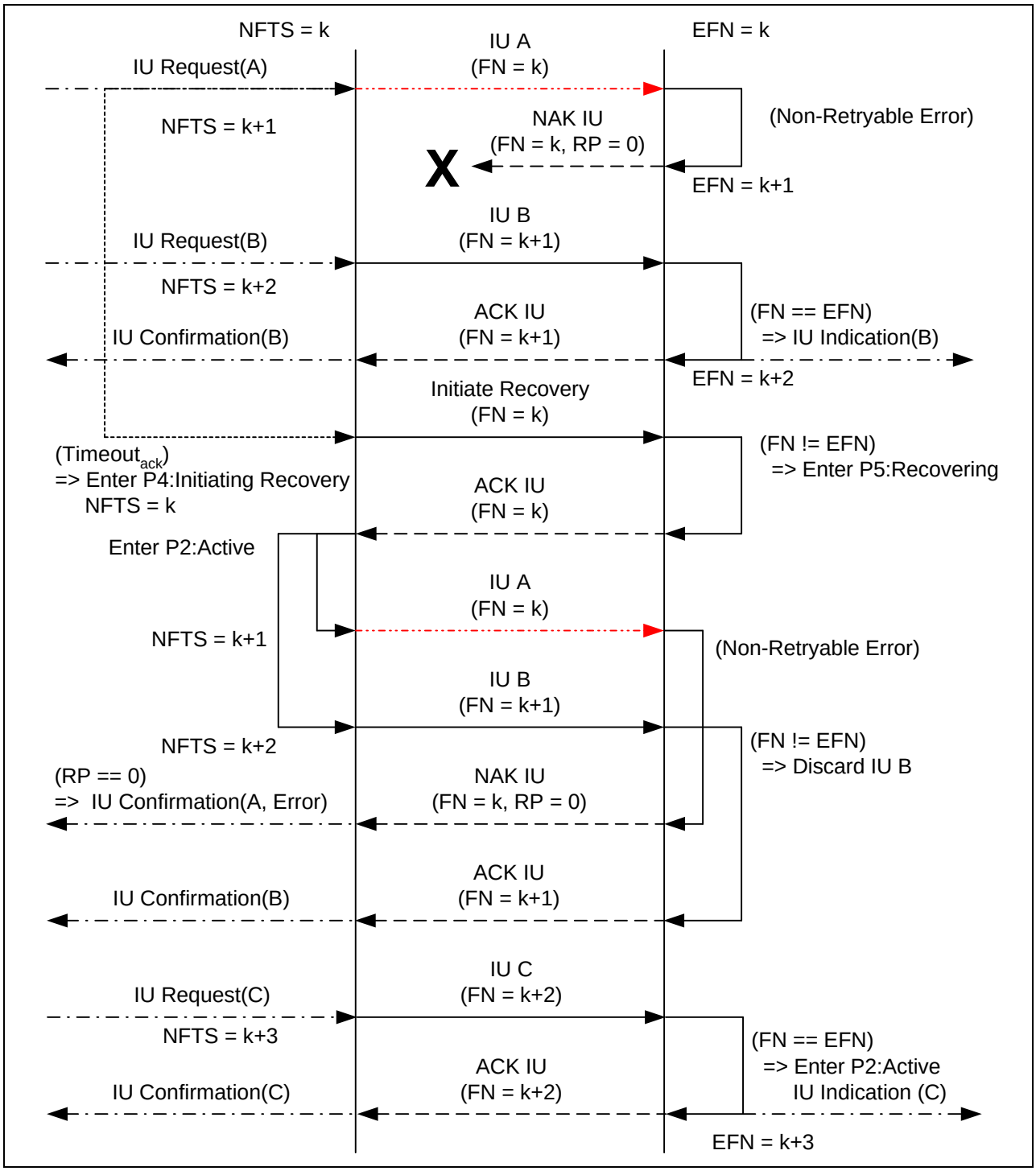


Figure A.6 – Lost NAK with recovery driven by timeout

A.8 Non-retryable error

In Figure A.7, the receiver detects a non-retryable error and sends a NAK IU with a value of zero in the PENDING RECOVERY (PR) field. The error is reported to the sender's upper layer and when transmission of the next IU is requested, it is sent with the next frame number in sequence.

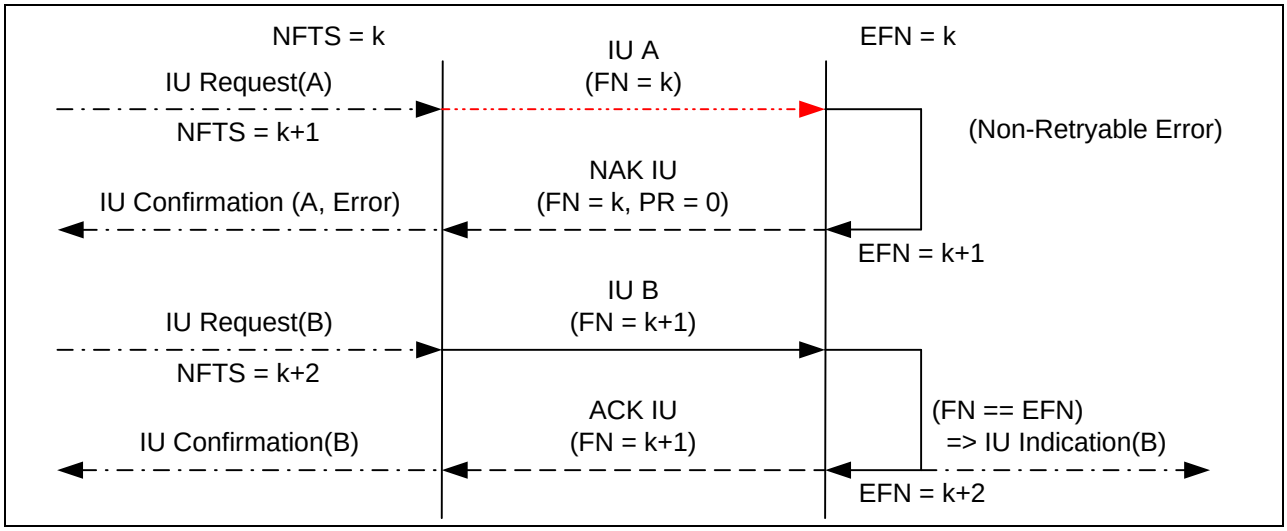


Figure A.7 – Non-retryable error

A.9 Lost ACK with errors on next IU

Figure A.8 shows a succession of three errors: a lost ACK, a retryable error, and a lost NAK for the retryable error. It is the timeout on the original lost ACK which begins the error recovery sequence.

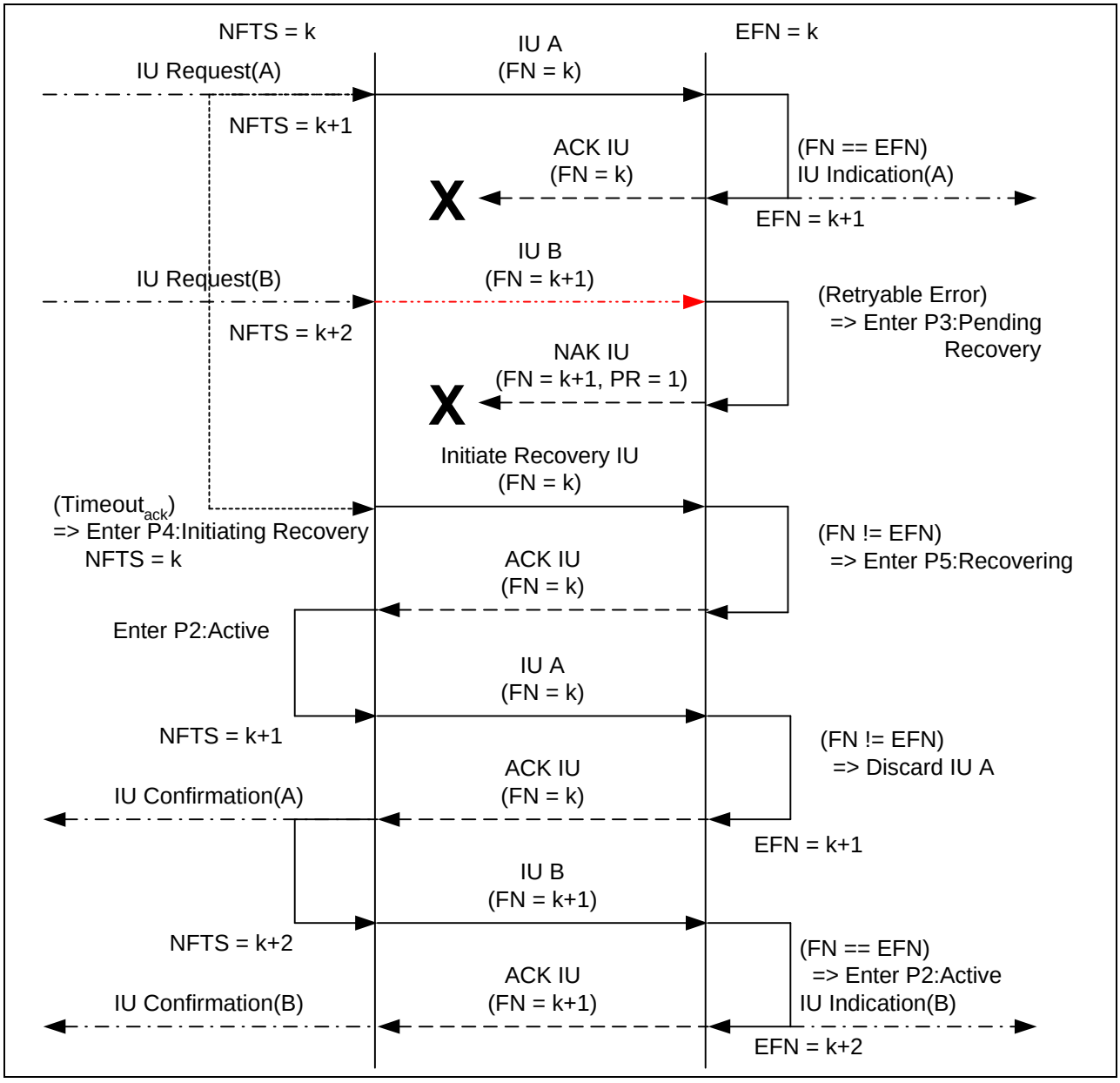


Figure A.8 – Lost ACK with errors on next IU