**TO:** T10 Membership
**FROM:** Paul A. Suhler, Certance
**DATE:** 12 February 2004
**SUBJECT:** T10/04-053r0, Revision of ADT Error Recovery Ladder Diagrams

**Revision 0:**

- Initial version

### 1.1 Background

T10/369r3 has changed the error recovery state machines. This proposal updates ADT Revision 10 Annex B to change the state names and certain transitions to match the new state machines. It also adds a ninth diagram to illustrate the new TE2:Retry Initiate Recovery state..

### 1.2 Changes

All sub-clauses of the Annex are changed, except for Non-retryable error (Figure A.7). It is included here for simplicity.

### A.2 Receiver-detected retryable error

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

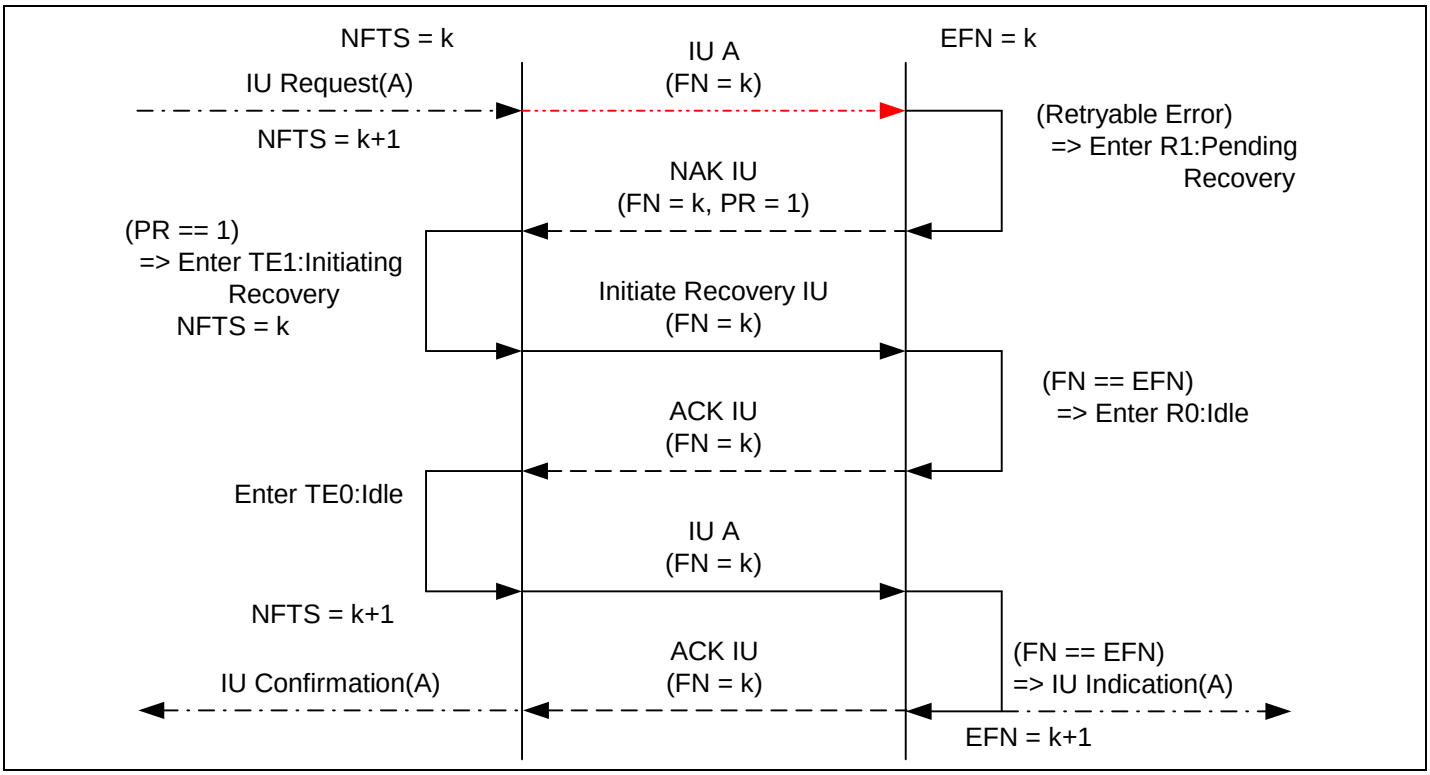


**Figure A.1 – Receiver-detected retryable error**

### 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.

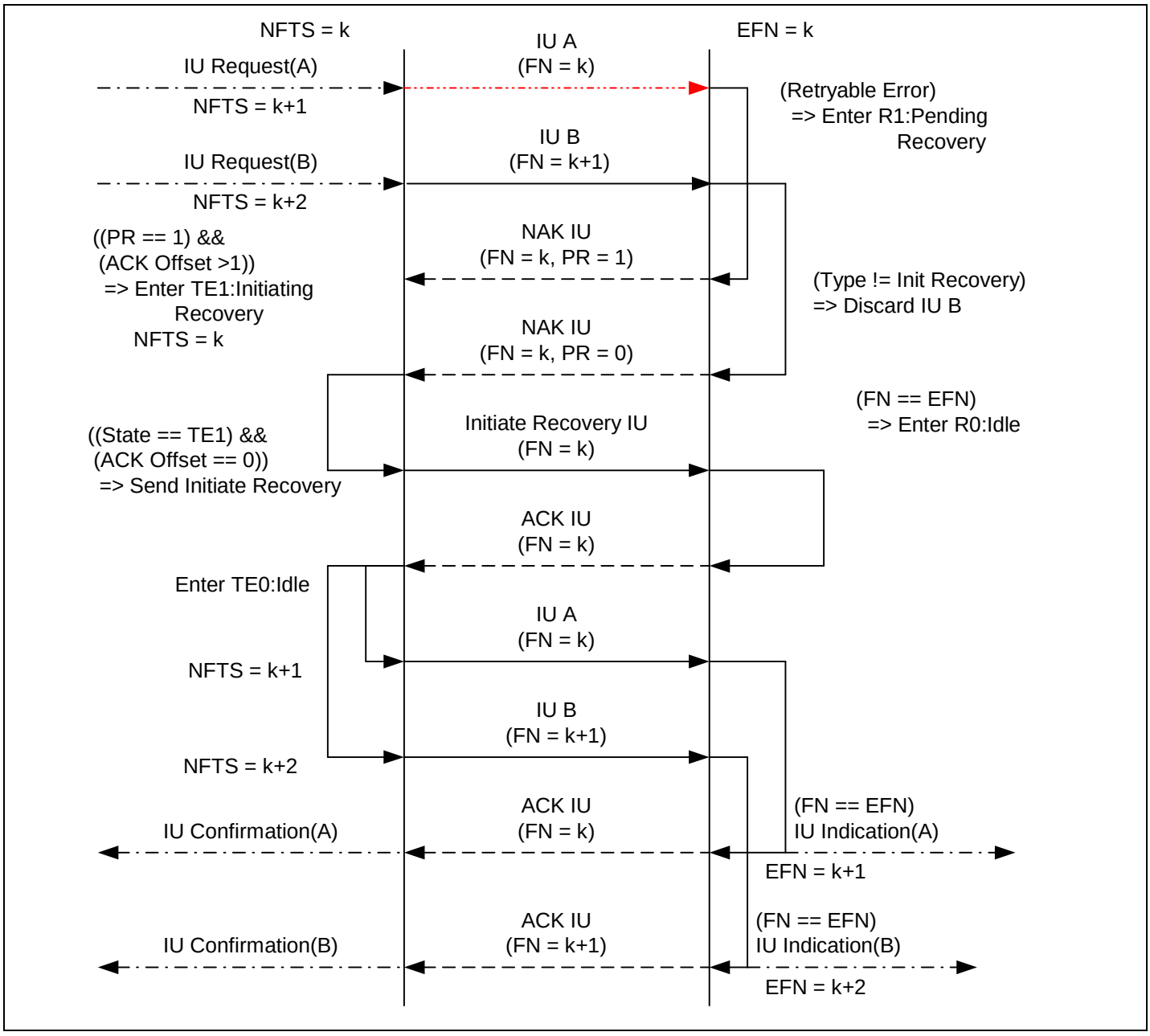


**Figure A.2 – Receiver-detected retryable error with multiple active IUs**

### 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.

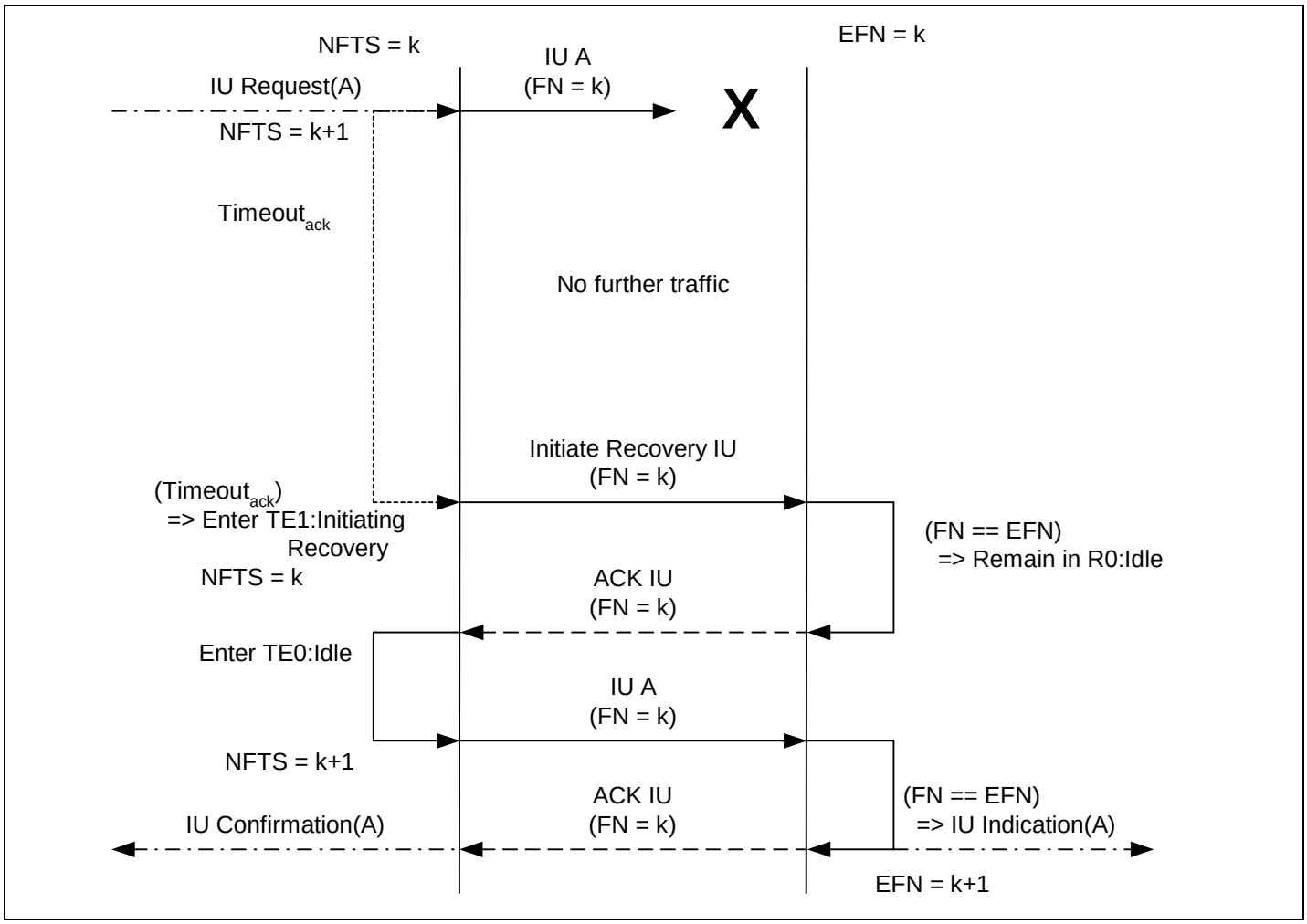


**Figure A.3 – Lost IU with no further traffic**

### 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.

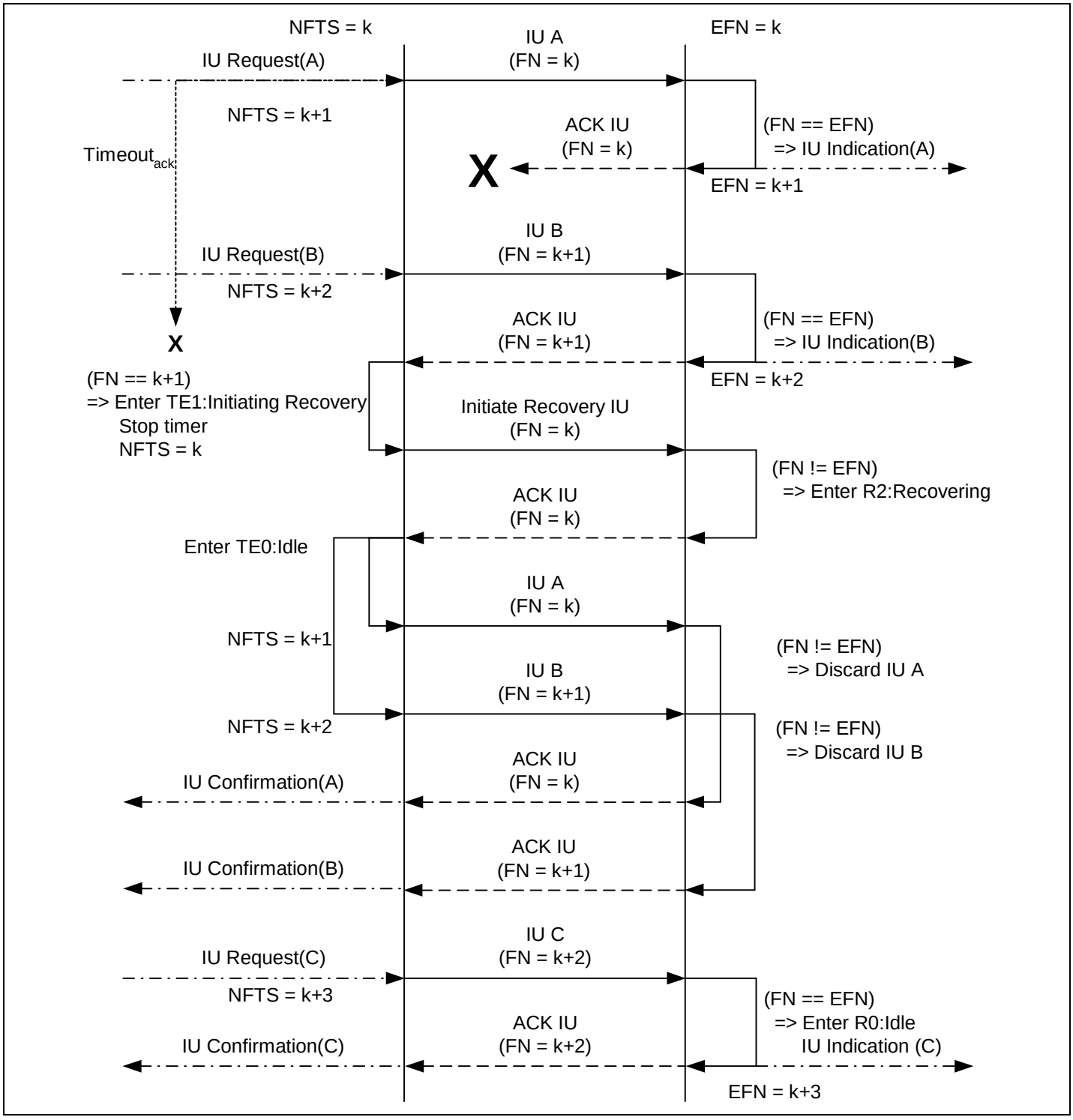


**Figure A.4 – Lost ACK with recovery driven by out-of-order ACK**

### 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.

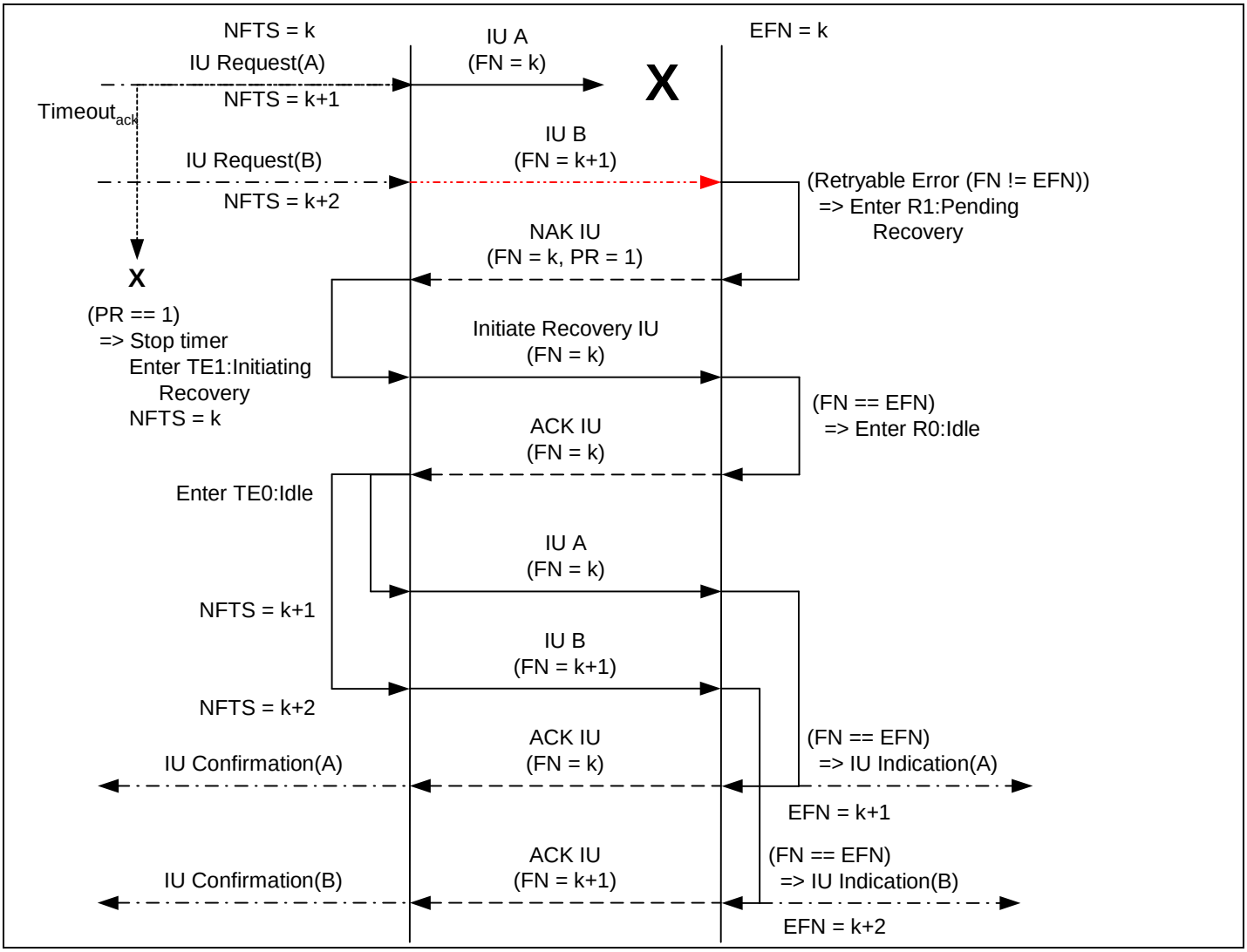


**Figure A.5 – Lost IU with recovery driven by out-of-order NAK**

### 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.

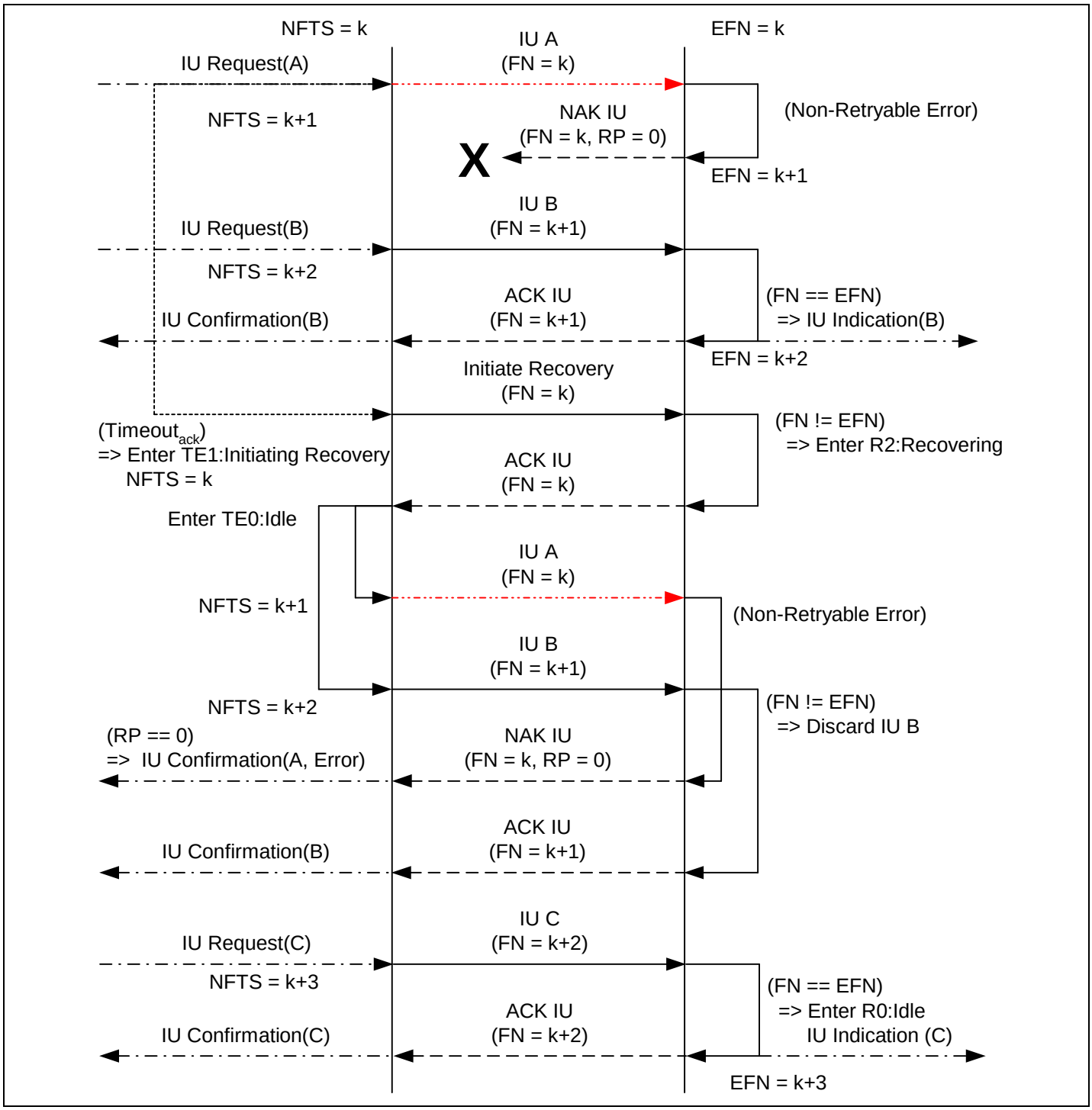


**Figure A.6 – Lost NAK with recovery driven by timeout**

### 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.

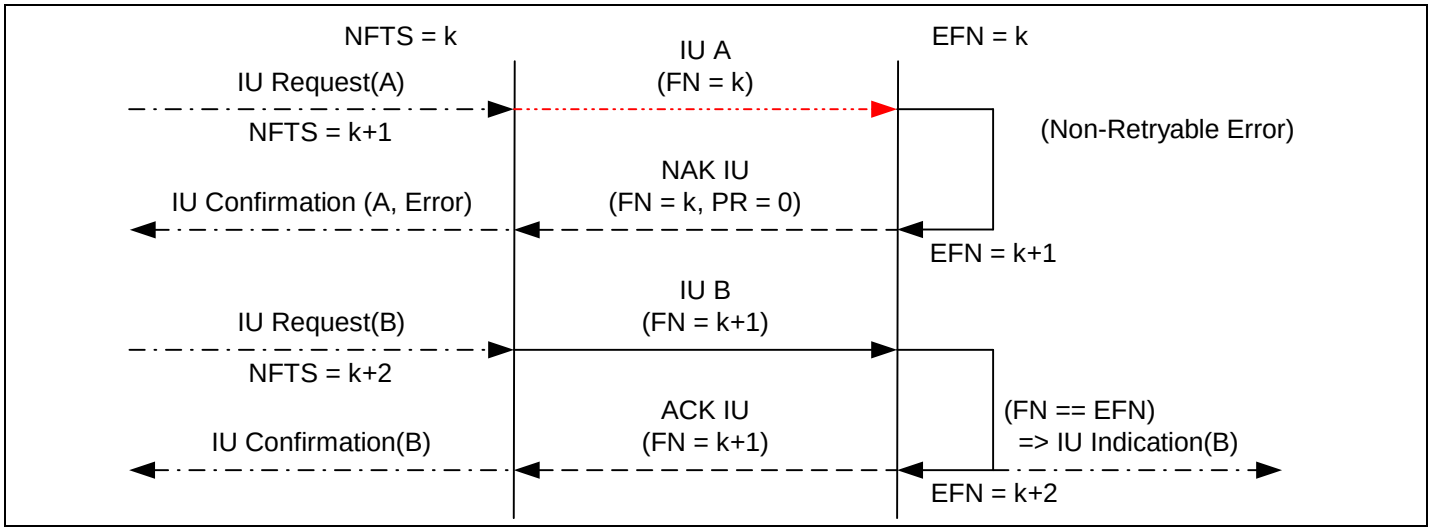


**Figure A.7 – Non-retryable error**

### 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.

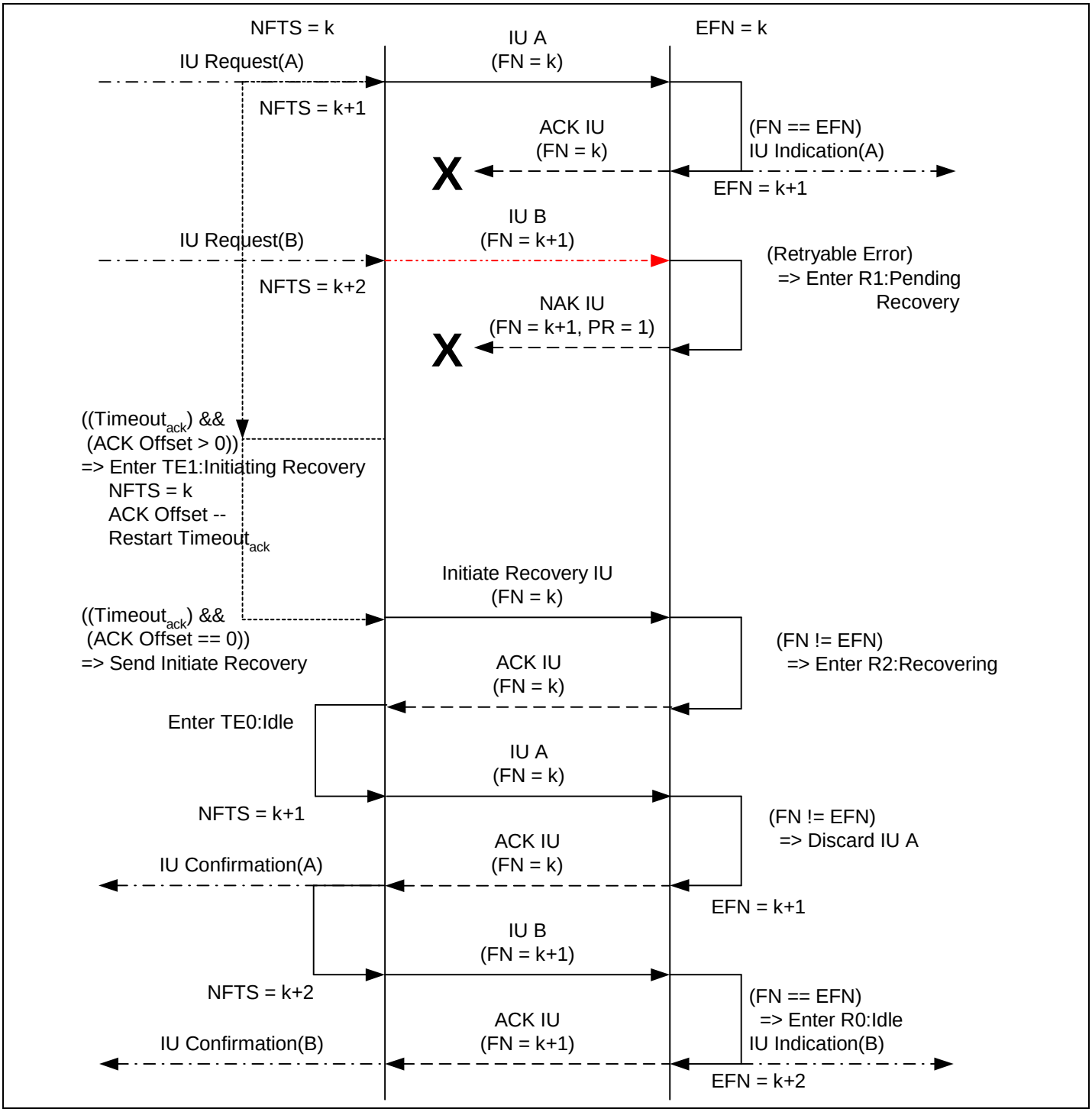


**Figure A.8 – Lost ACK with errors on next IU**

### A.10 Error on Initiate Recovery IU

Figure A.9 shows the detection of a retryable error by the receiver, detection of an error on the Initiate Recovery IU, and the subsequent recovery.

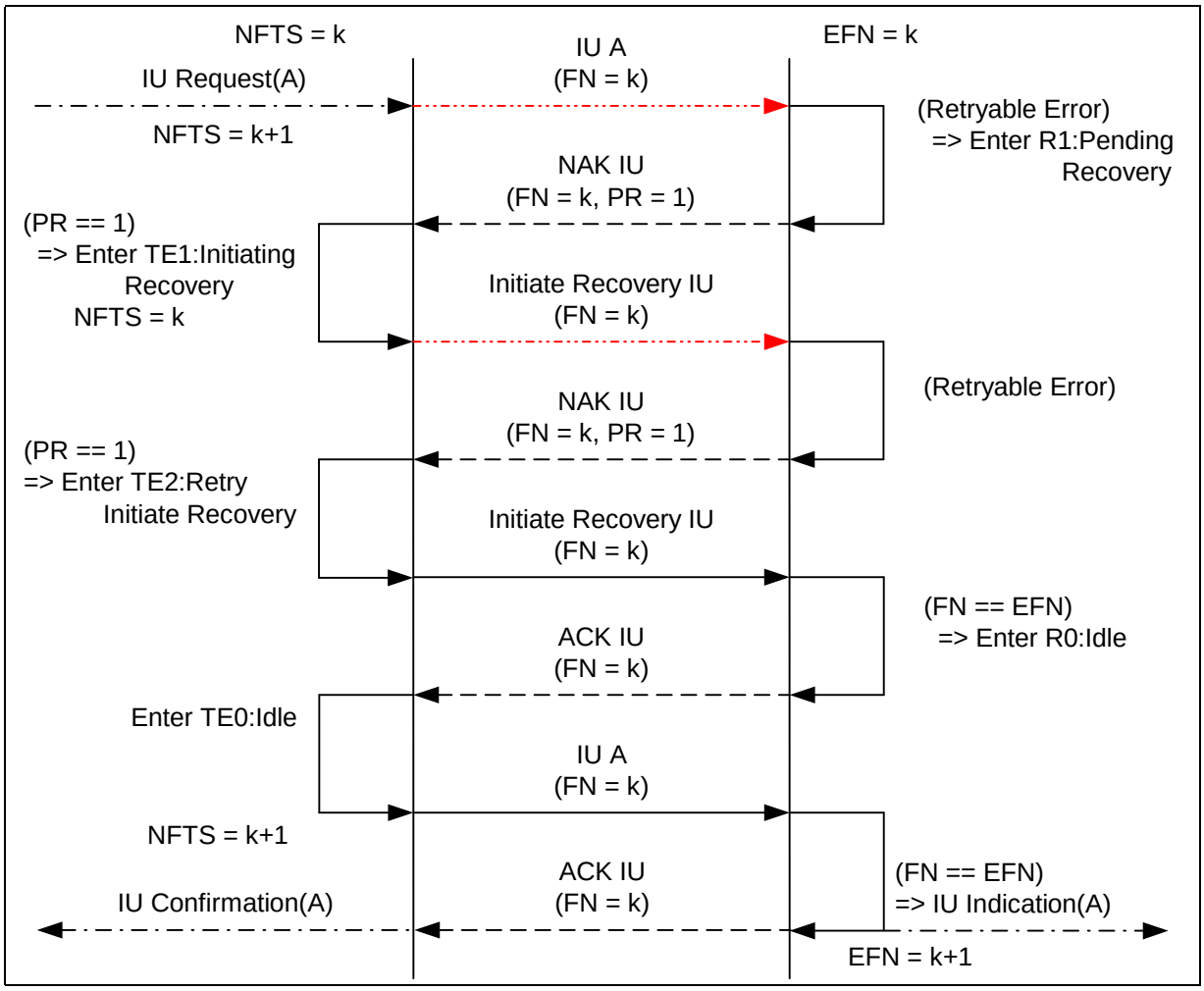


**Figure A.9 – Error on Initiate Recovery IU**