Quantum Corporation
4001 Discovery Drive, Suite 1100
Boulder, CO 80303
720-406-5611
Jim.Jones@quantum.com

Document: T10/02-487r0
Date: December 20, 2002
To: T10 Committee Membership
From: Jim Jones, Quantum Corporation
Subject: SAS Transport Layer Retries

Related Documents

- SAS-r03
- SSC-2 (ssc2r08f)
- T10/02-323r2 – SAS Data Corruption Problem [Robert Sheffield, Intel Corporation]
- T10/02-449r1 – SAS Simple Relative Offset [Bill Galloway, BREA Technologies]

Overview of the Issue

As defined in SAS-r03, there is a severe limitation for some devices in the handling of frame transmission errors that will result in an unacceptably high frequency of failed write or read commands as compared to existing parallel-SCSI solutions. The frequency of failed commands results from a combination of the specified bit error rate (BER), the supported link rate, and the policy for handling frame errors detected during transmission through a structure of expanders.

The BER, defined to be 10^{-12}, indicates that some kind of transmission error will occur for every 100 GBytes of data transferred. This is less than the amount of data that can be contained on one tape with today's tape drive technology.

Pushing the recovery of transmission errors to the application layer may be unrealistic for applications that can't just reissue the failing command for reasons that will be noted below.

Transmission Error Handling

A variety of error conditions, many related to transmission errors, result in a frame being discarded.

From SAS-r02c section 7.16.7.9:

> The frame (i.e., all the dwords between an SOF and EOF) shall be discarded if any of the following conditions are true:
>
> a) the number of data dwords between the SOF and EOF is less than 7;
> b) the number of data dwords after the SOF is greater than 263 data dwords;
> c) the Rx Credit Status (Credit Exhausted) parameter is received; or
> d) the DONE Received parameter is received.
>
> If consecutive SOF Received parameters are received without an intervening EOF Received parameter (i.e., SOF, data dwords, SOF, data dwords, and EOF instead of SOF, data dwords, EOF, SOF, data dwords, and EOF) then this state shall discard all dwords between those SOFs.

From SAS-r02c section 9.2.4.3:

> If a target port transmits an XFER_RDY frame and does not receive an ACK or NAK, it shall close the connection with DONE (ACK/NAK TIMEOUT) and return a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of ACK/NAK TIMEOUT (see 10.1.2).

If a target port transmits an XFER_RDY frame and receives a NAK, it shall return a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of NAK RECEIVED (see 10.1.2).

From SAS-r02c section 9.2.4.4:

If a target port transmits a DATA frame and does not receive an ACK or NAK, it shall close the connection with DONE (ACK/NAK TIMEOUT) and return a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of ACK/NAK TIMEOUT (see 10.1.2).

If a target port transmits a DATA frame and receives a NAK, it shall return a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of NAK RECEIVED (see 10.1.2).

If an initiator port transmits a DATA frame and does not receive an ACK or NAK, it shall abort the command with ABORT TASK (see 10.1.3).

If an initiator port transmits a DATA frame and receives a NAK, it shall abort the command with ABORT TASK (see 10.1.3).

Consequences of a Discarded DATA Frame

Consider the situation where a transmission error occurs when a port is sending a DATA frame. For the sake of argument, assume that a SOF is corrupted so that a frame is completely lost. However, the result is substantially similar with other errors, e.g. coding violations or a CRC error.

The addition of a Simple Relative Offset (T10/02-449r1) to the SSP frame can help the port receiving DATA frames detect that a frame is missing for all but the last DATA frame by comparing the Relative Offset of the received frame with that of the expected Relative Offset. The last DATA frame is detected as missing by timing out the data's arrival (ACK/NAK TIMEOUT).

Unfortunately, there is no way of knowing how many DATA frames are missing, since DATA frames can be any size – the only requirement is that DATA frame lengths are between 1 and 1024 bytes and a multiple of 4 bytes (except for the last one).

Also, with the ability for some devices to have large logical block sizes (tape devices can be to 16MB), it's possible to have a partial data block written onto the media at the time a DATA frame is detected as missing. As noted above, when a DATA frame is detected as missing, the write or read command will fail with CHECK CONDITION status with a sense key of ABORTED COMMAND.

Applications receiving this type of CHECK CONDITION may not be able to just reissue the command as is possible with disk drives. For example, a sequential application such as tape would need to reposition to the beginning of the failing block and reattempt the write operation. Some tape devices will only allow a write operation at "end of data" – this is no longer the case if there is a partial block on the media. So due to the complexity of error recovery and it not being possible it in certain situations, the application will fail unless error recovery is performed at the transport layer.

Changes to Enable Transport Layer Retries

The following is a proposal to enable retransmission of XFER_RDY and DATA frames.

If the EMDP bit is 1 in the Disconnect-Reconnect mode page, the target will try to recover from data errors (taking advantage of relative offset).

In order to ensure that non-tape devices are not affected by this proposal, a port requesting retransmission in a new connection will use one of the currently reserved DONE primitives (i.e.

DONE (RESERVED TIMEOUT 0)). This proposal renames this to DONE (RETRANSMIT). This special DONE tells the port to not assume success for the current data burst (in case it received the frame and sent an ACK, but the ACK got lost). A port receiving a DONE (RETRANSMIT) can treat it the same as a DONE (ACK/NAK TIMEOUT) and act accordingly if no retries are desired (i.e. it shall send a DONE (ACK/NAK TIMEOUT) indicating it does not support data retransmission).

XFER_RDY and DATA frames that are retransmitted in the same connection will have the RETRANSMIT bit set in the frame header.

Read DATA frames

- If a read DATA frame is NAKed, retry it and subsequent (previously sent, in flight, or even sent later) read DATA frames as needed. The new DATA frames have the appropriate relative offsets.
- If a read DATA frame encounters an ACK/NAK timeout, close the connection with DONE (RETRANSMIT). Retry it and subsequent read DATA frames in the next connection.

XFER_RDY frames

- If an XFER_RDY frame is NAKed, retry it.
- If an XFER_RDY frame encounters an ACK/NAK timeout, close the connection with DONE (RETRANSMIT). Retry the XFER_RDY in the next connection. Note: The initiator could reply with ACK and then send write DATA frames immediately. The target should ignore these (at its transport layer) and only accept them after a successful XFER_RDY.

Write DATA frames

- If a write DATA frame is NAKed, retry it and subsequent (previously sent, in flight, or even sent later) read DATA frames as needed. The new DATA frames have the appropriate relative offsets.
- If a write DATA frame encounters an ACK/NAK timeout, close the connection with DONE (RETRANSMIT). Retry it and subsequent write DATA frames in the next connection.
- If the target did send an ACK, it might move on to the next XFER_RDY before getting the DONE (RETRANSMIT). Marking the retried write DATA frames with a RETRANSMIT bit might be helpful here (the target could ignore the retries, since it is happy with the original set).

To illustrate the recovery mechanism, examples of the following are described in the next section.

Read Operations

- read DATA frame NAK
- read DATA frame ACK lost
- read DATA frame lost (no ACK or NAK)
- read DATA frame NAK lost

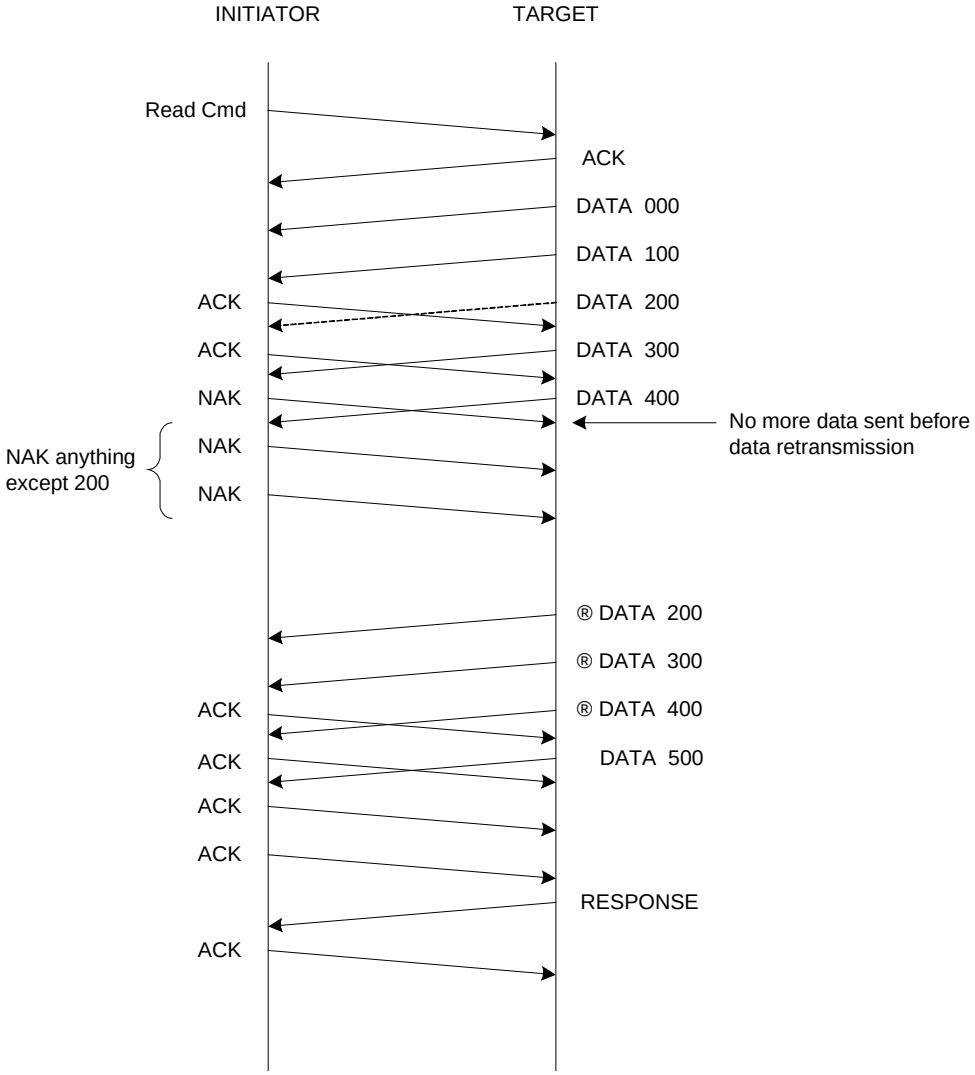
Write Operations

- XFER_RDY frame NAK
- XFER_RDY DATA frame ACK lost (note: initiator might start sending DATA frames after ACK sent)
- XFER_RDY DATA frame lost (no ACK or NAK)
- XFER_RDY DATA frame NAK lost
- write DATA frame NAK
- write DATA frame ACK lost
- write DATA frame lost (no ACK or NAK)
- write DATA frame NAK lost

Read DATA Frame NAK

The following is an example of a Read operation in which a DATA frame is NAKed. The initiator NAKs the following DATA frames as well. The target will retransmit the data beginning at the relative offset of the first frame NAKed.

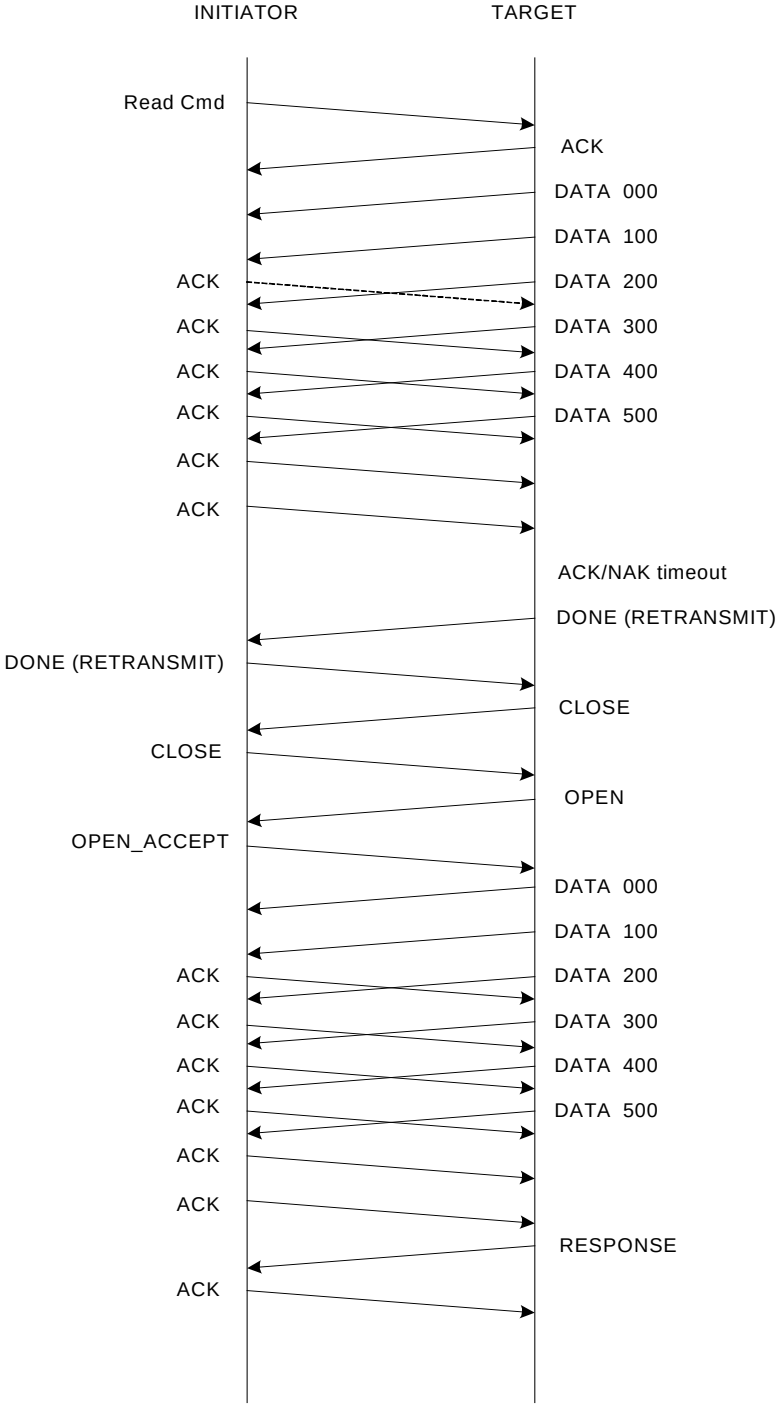
NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.



Read DATA Frame ACK Lost

The following is an example of a Read operation in which the ACK is lost. The target times out with an ACK/NAK imbalance. The target indicates that it wants to retransmit the data in a new connection by sending the DONE (RETRANSMIT) primitive. Data retransmission must restart from the beginning since there's no way to know if a DATA frame or an ACK/NAK was lost.

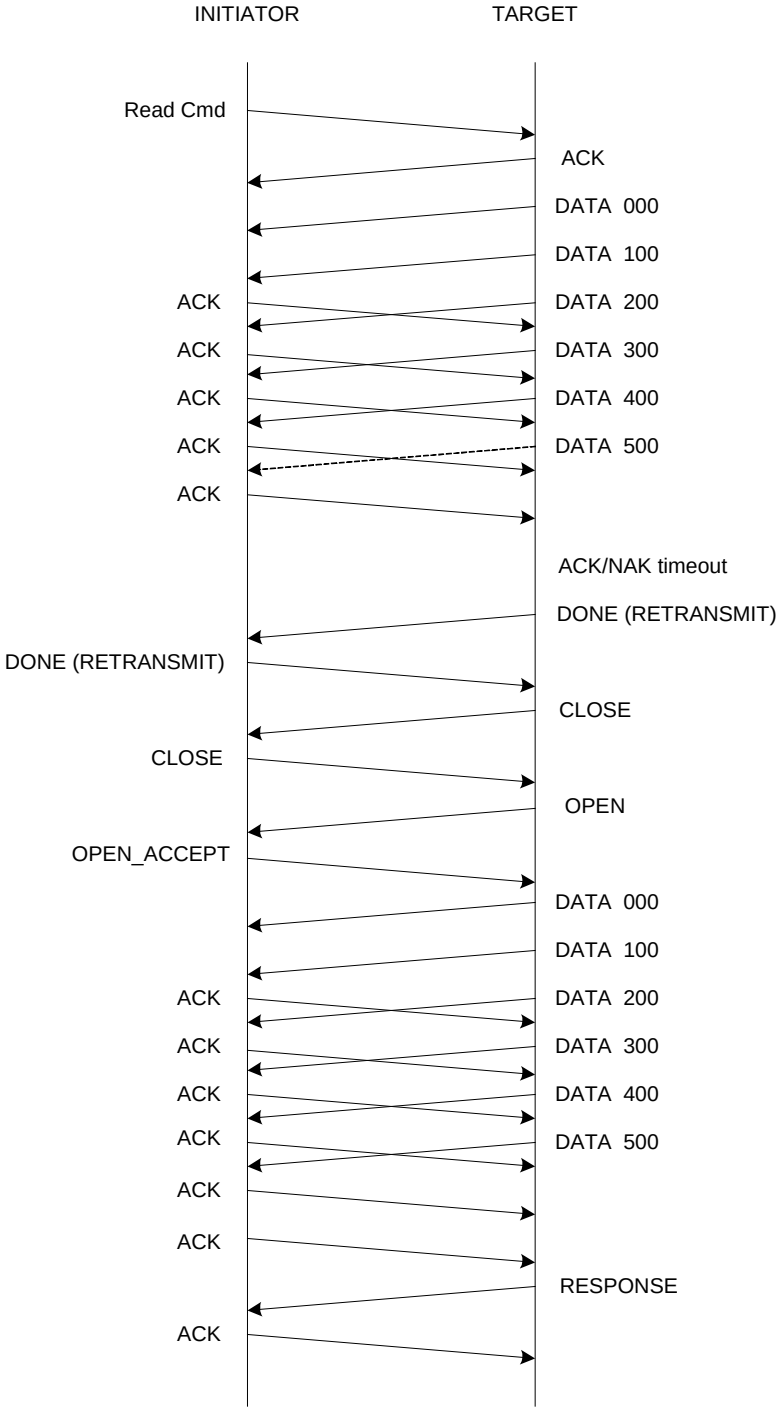
NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.



Read DATA Frame Lost (no ACK or NAK)

The following is an example of a Read operation in which the DATA frame is lost. The target times out with an ACK/NAK imbalance. The target indicates that it wants to retransmit the data in a new connection by sending the DONE (RETRANSMIT) primitive. Data retransmission must restart from the beginning since there's no way to know if a DATA frame or an ACK/NAK was lost.

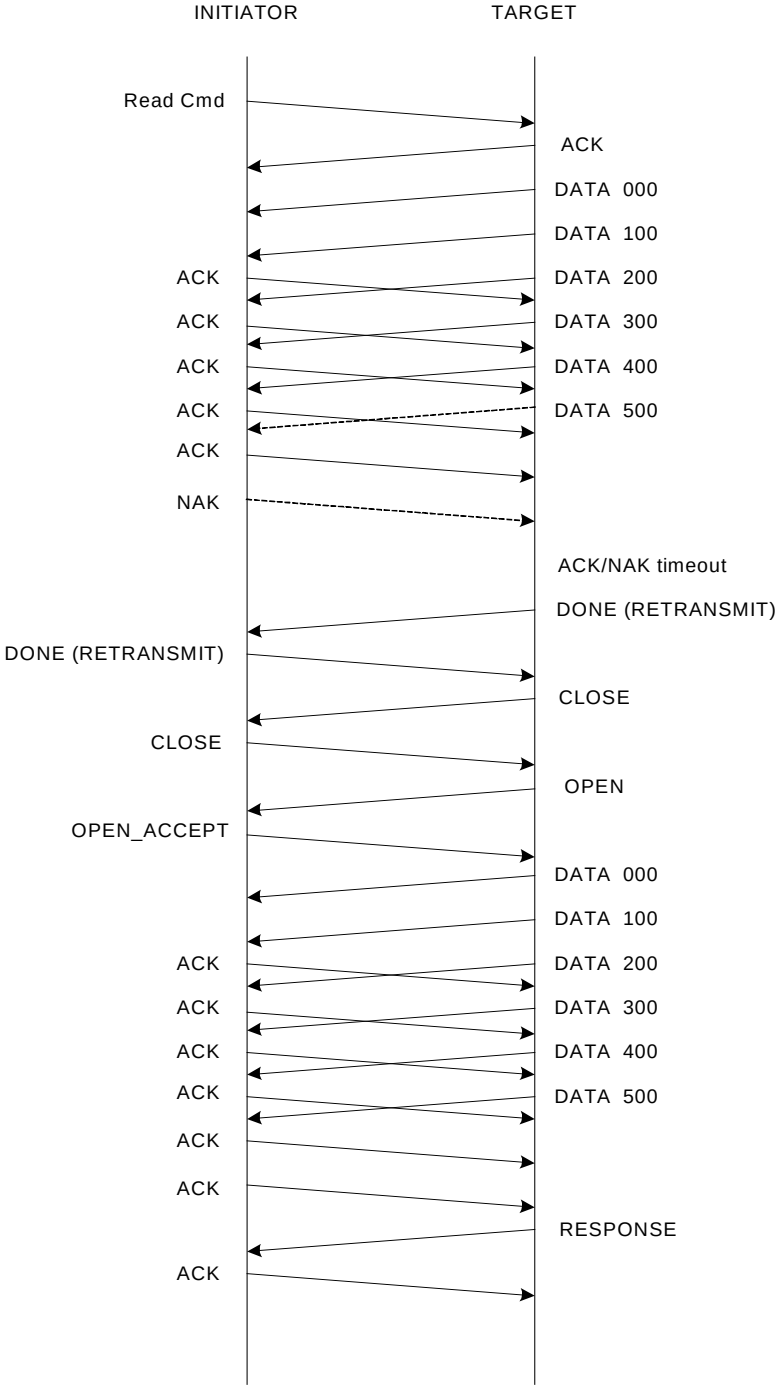
NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.



Read DATA Frame NAK Lost

The following is an example of a Read operation in which the NAK is lost. The target times out with an ACK/NAK imbalance. The target indicates that it wants to retransmit the data in a new connection by sending the DONE (RETRANSMIT) primitive. Data retransmission must restart from the beginning since there's no way to know if a DATA frame or an ACK/NAK was lost.

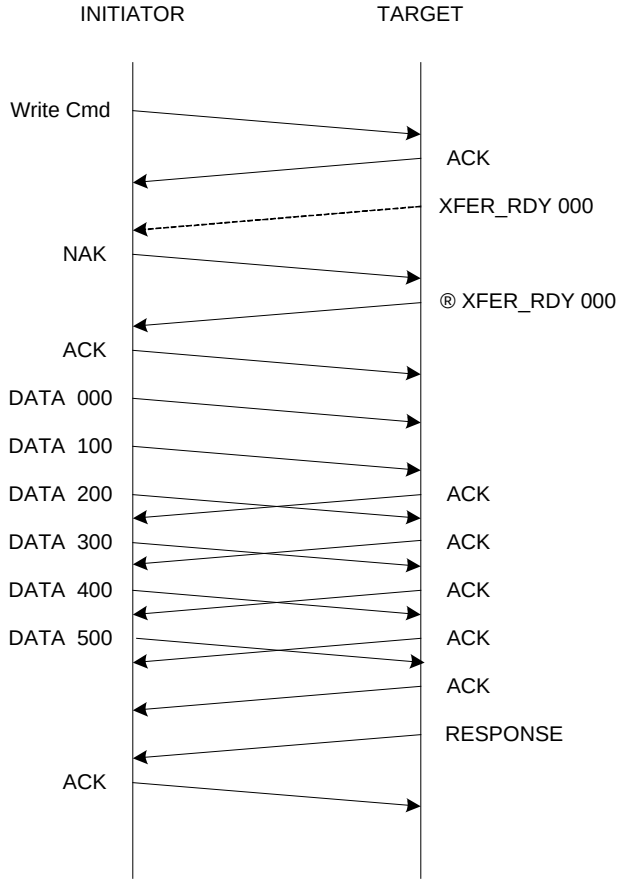
NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.



Write XFER_RDY NAK

The following is an example of a Write operation in which the XFER_RDY frame is NAKed. The target will retransmit the XFER_RDY frame.

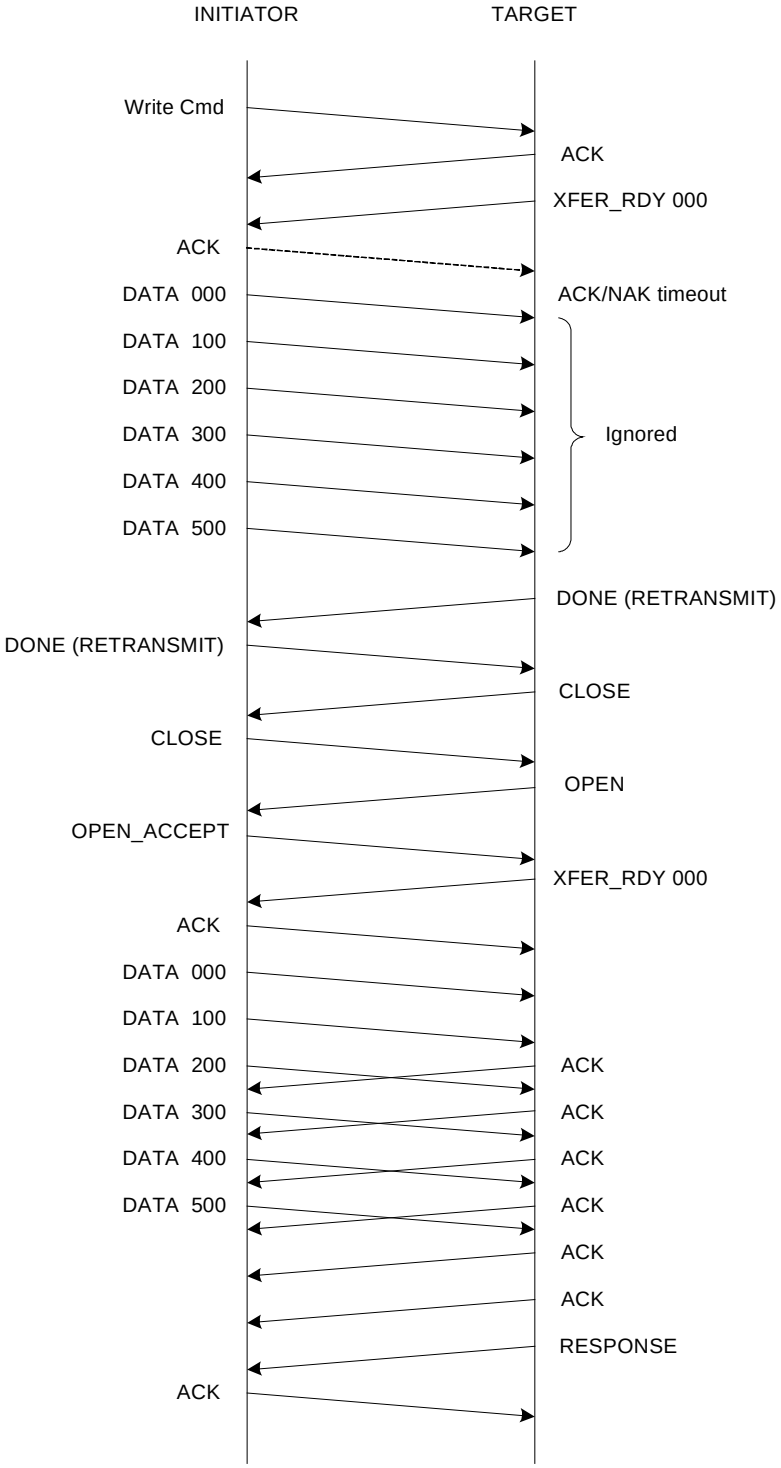
NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.



Write XFER_RDY ACK Lost

The following is an example of a Write operation in which the ACK for the XFER_RDY frame is lost. The initiator doesn't know this and begins transmitting DATA frames. The target times out waiting for the ACK to the XFER_RDY frame. The target indicates that it wants data retransmission in a new connection by sending the DONE (RETRANSMIT) primitive.

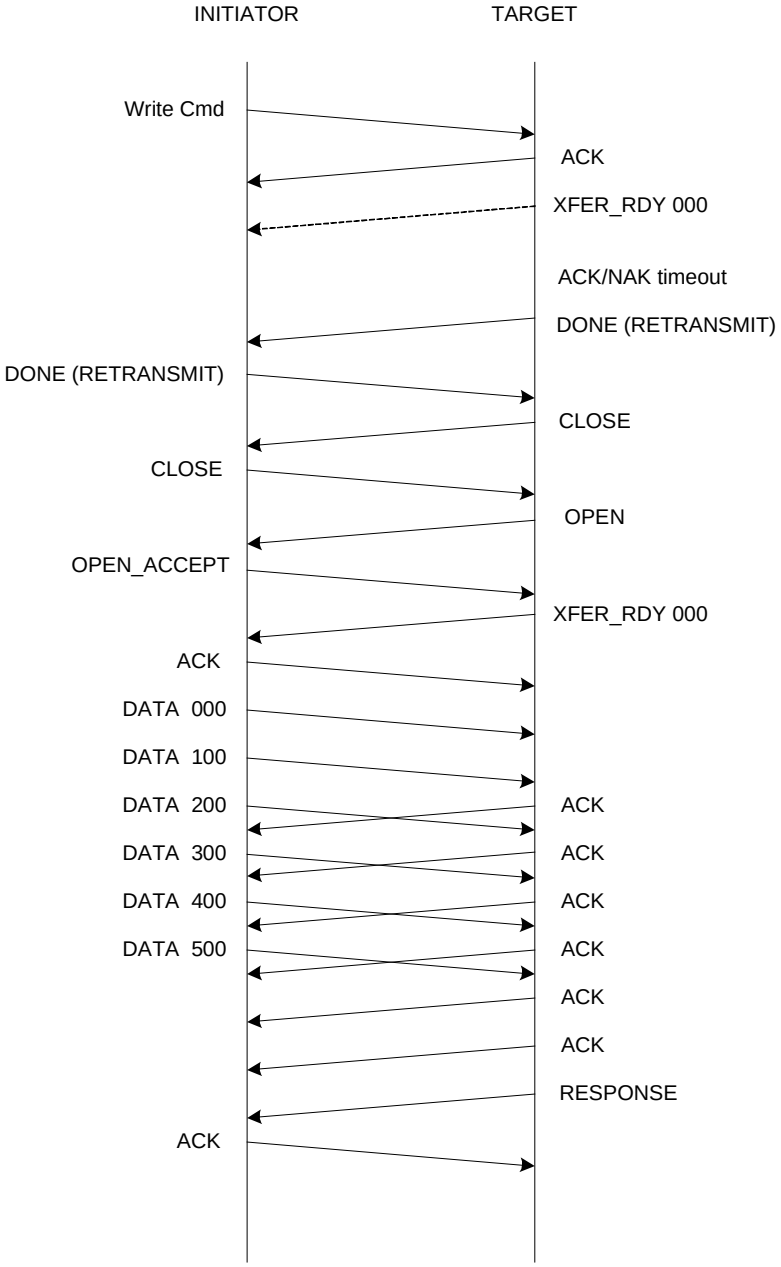
NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.



Write XFER_RDY Frame Lost

The following is an example of a Write operation in which the XFER_RDY frame is lost. The target times out waiting for the ACK to the XFER_RDY frame. The target indicates that it wants data retransmission in a new connection by sending the DONE (RETRANSMIT) primitive.

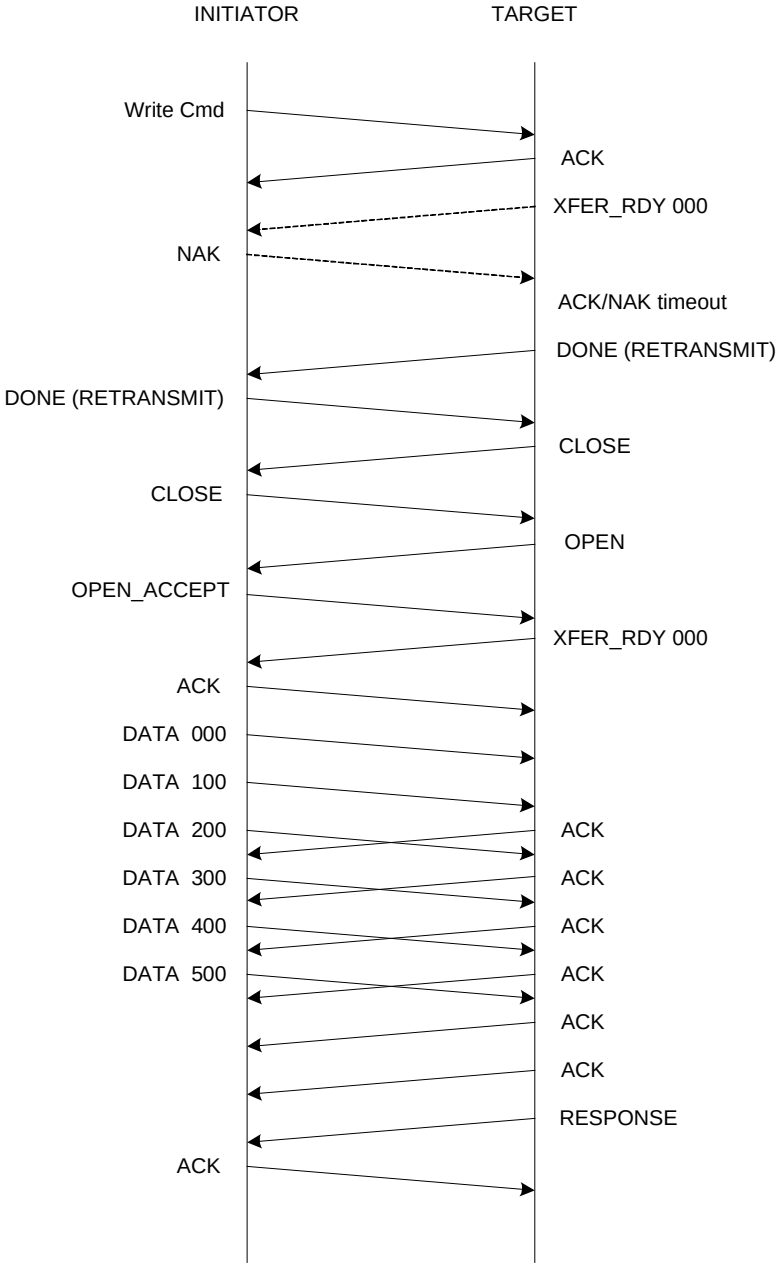
NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.



Write XFER_RDY NAK Lost

The following is an example of a Write operation in which the NAK for the XFER_RDY frame is lost. The target times out waiting for the ACK/NAK to the XFER_RDY frame. The target indicates that it wants data retransmission in a new connection by sending the DONE (RETRANSMIT) primitive.

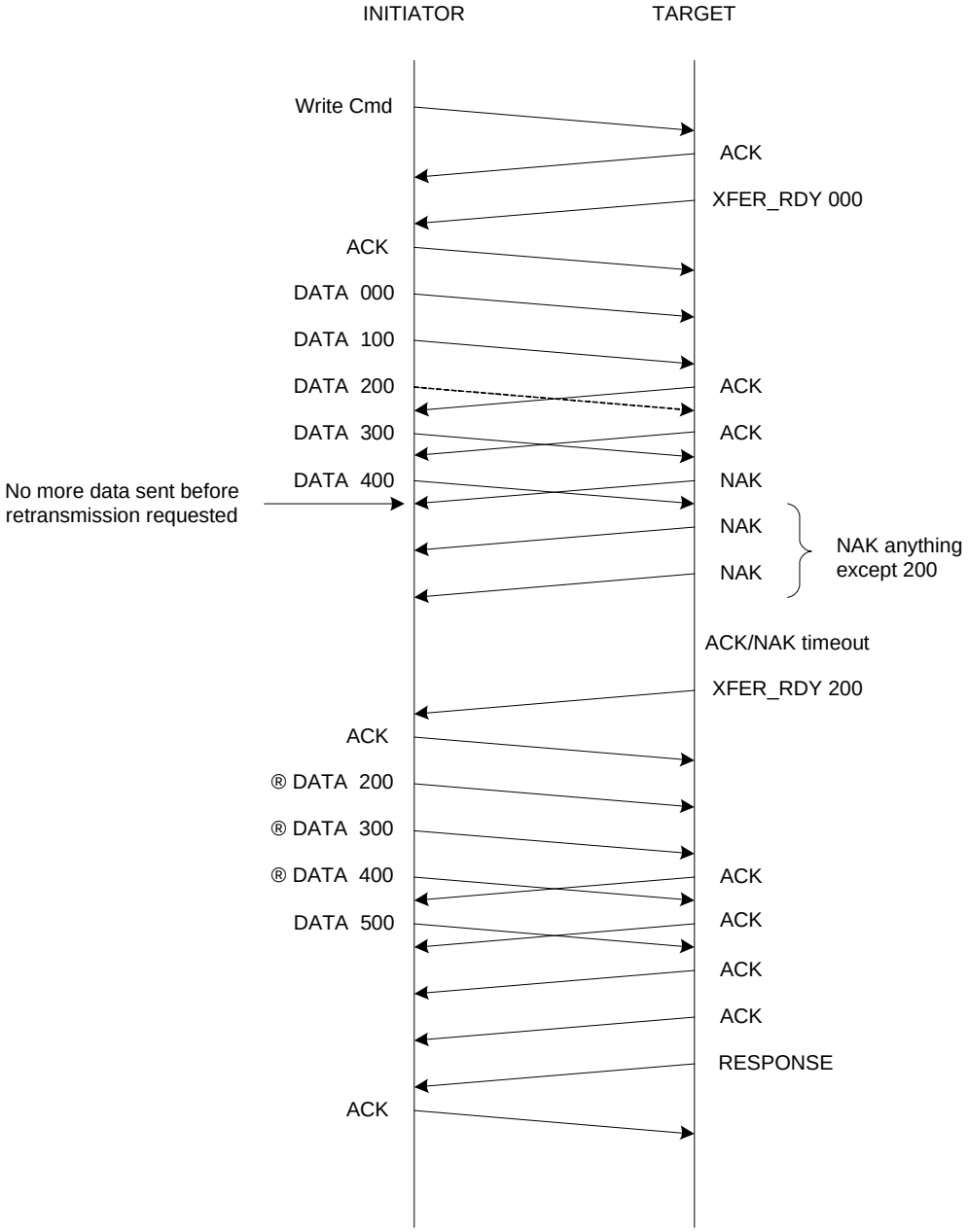
NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.



Write DATA Frame NAK

The following is an example of a Write operation in which the DATA frame is NAKed. The target also NAKs following DATA frames as well. The target will request data retransmission beginning at the relative offset of the first frame NAKed.

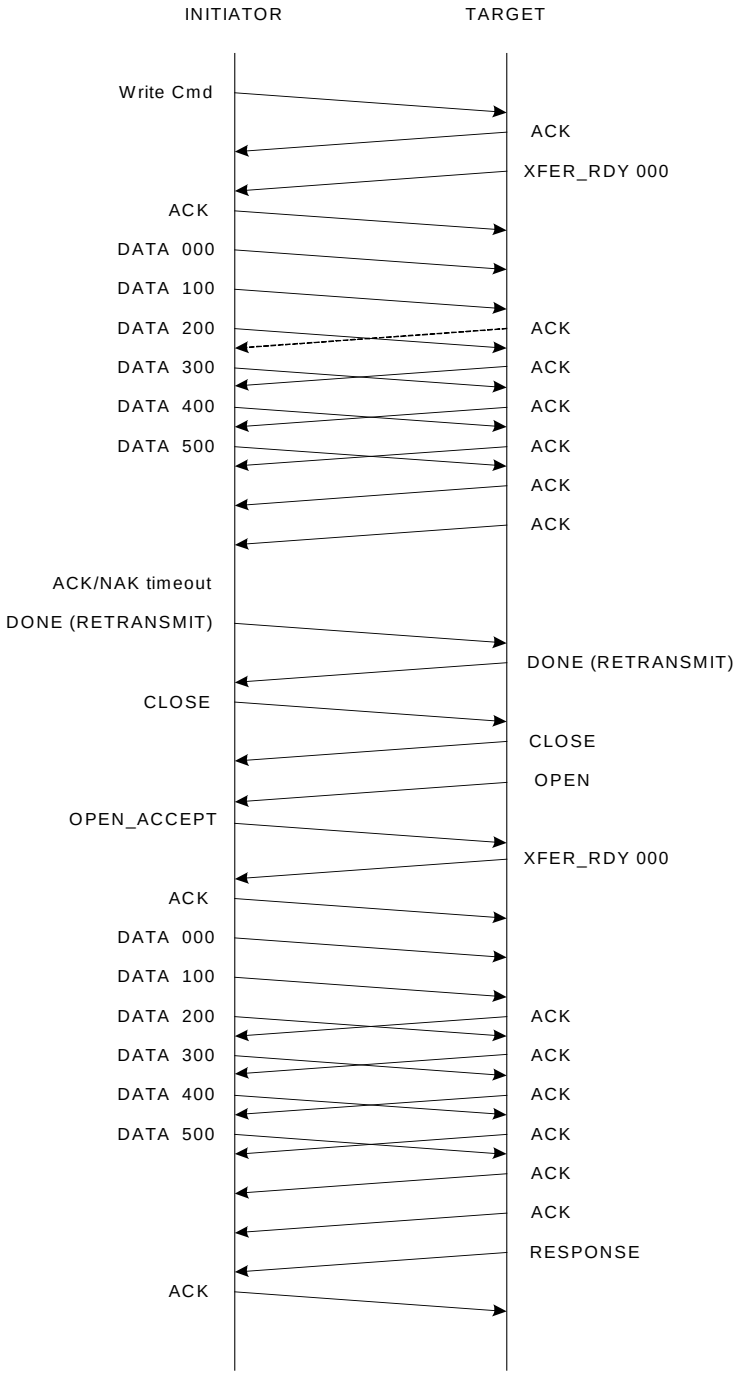
NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.



Write DATA Frame ACK Lost

The following is an example of a Write operation in which the ACK is lost. The initiator times out with an ACK/NAK imbalance. The initiator indicates that it wants to the data retransmitted in a new connection by sending the DONE (RETRANSMIT) primitive. Data retransmission must restart from the beginning since there's no way to know if a DATA frame or an ACK/NAK was lost.

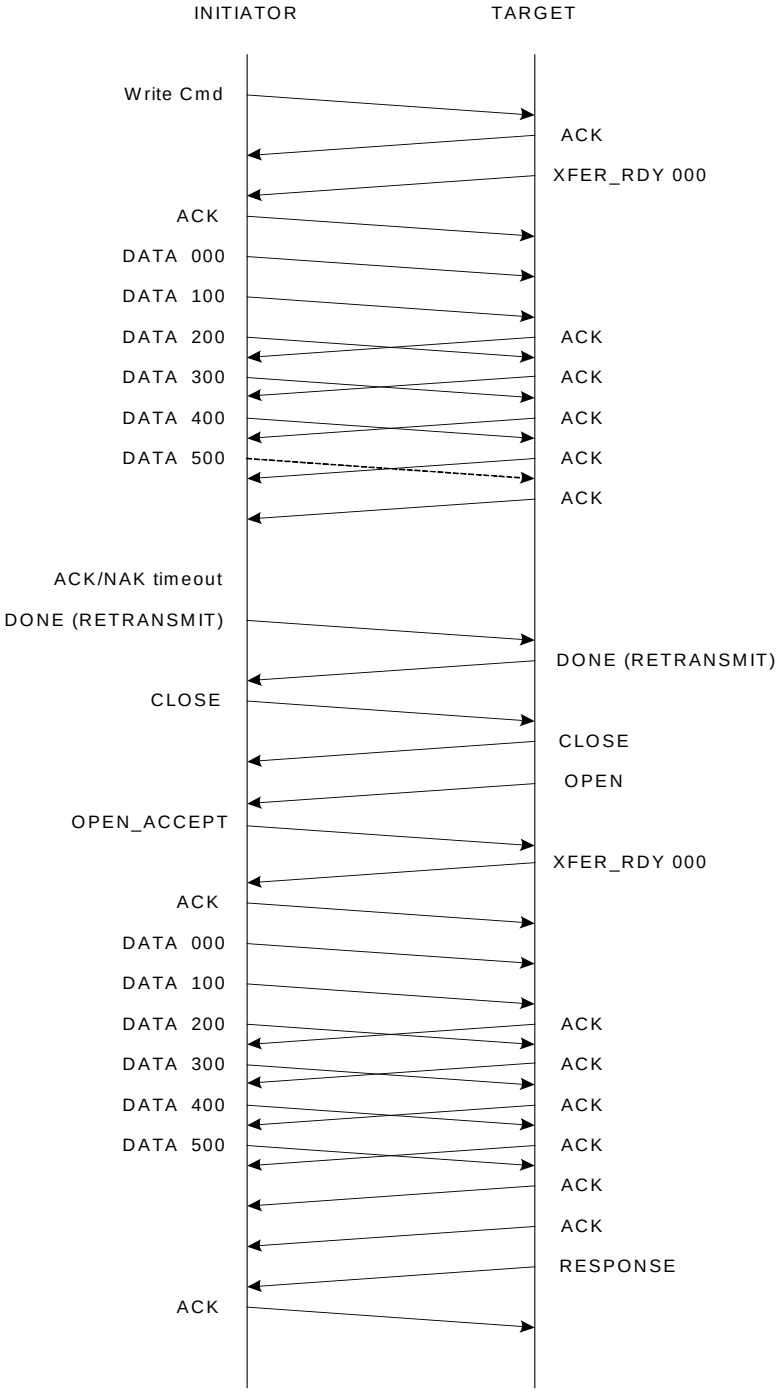
NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.



Write DATA Frame Lost (no ACK or NAK)

The following is an example of a Write operation in which the DATA frame is lost. The initiator times out with an ACK/NAK imbalance. The initiator indicates that it wants to the data retransmitted in a new connection by sending the DONE (RETRANSMIT) primitive. Data retransmission must restart from the beginning since there's no way to know if a DATA frame or an ACK/NAK was lost.

NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.



Write DATA Frame NAK Lost

The following is an example of a Write operation in which the NAK is lost. The initiator times out with an ACK/NAK imbalance. The initiator indicates that it wants to the data retransmitted in a new connection by sending the DONE (RETRANSMIT) primitive. Data retransmission must restart from the beginning since there's no way to know if a DATA frame or an ACK/NAK was lost.

NOTE: in the example below, the number following the XFER_RDY or DATA frame is the relative offset. ® is used to indicate that the frame is retransmitted within the same connection. The dotted line indicates lost or corrupted data.