Revision History
Revision 1 (11 August 2003) second revision
Revision 0 (25 June 2003) first revision

Related Documents
sas-r05 - Serial Attached SCSI revision 5
03-186r2 - SAS-1.1 Transport Layer Retries

Overview
This presents ladder diagrams for all the error recovery cases in 03-186.

Conventions:
® DATA means resend the DATA frame
For certain frame types, the RETRANSMIT bit is set to 1 when retransmitting.
RESPONSE Frame NAK Received

<table>
<thead>
<tr>
<th>INITIATOR</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAK</td>
<td>RESPONSE</td>
</tr>
<tr>
<td>ACK</td>
<td>RESPONSE</td>
</tr>
</tbody>
</table>

- NAK received for RESPONSE Frame - resend RESPONSE Frame with RETRANSMIT=1
- The RETRANSMIT bit is not strictly necessary here; it doesn’t help the initiator. It helps debugging with logic analyzers.

Tag is not ready for reuse (this ACK could still be lost)
RESPONSE Frame ACK Lost

<table>
<thead>
<tr>
<th>INITIATOR</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>RESPONSE</td>
</tr>
<tr>
<td></td>
<td>ACK/NAK timeout</td>
</tr>
<tr>
<td>DONE</td>
<td>DONE (ACK/NAK TIMEOUT)</td>
</tr>
<tr>
<td></td>
<td>resend RESPONSE Frame with RETRANSMIT=1</td>
</tr>
<tr>
<td>CLOSE</td>
<td>CLOSE</td>
</tr>
<tr>
<td>OPEN_ACCEPT</td>
<td>OPEN</td>
</tr>
<tr>
<td>ACK</td>
<td>RESPONSE</td>
</tr>
</tbody>
</table>

Tag is not ready for reuse (this ACK could still be lost)
RESPONSE Frame NAK Lost

<table>
<thead>
<tr>
<th>INITIATOR</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAK</td>
<td>RESPONSE</td>
</tr>
<tr>
<td></td>
<td>ACK/NAK timeout</td>
</tr>
<tr>
<td>DONE</td>
<td>DONE (ACK/NAK TIMEOUT)</td>
</tr>
<tr>
<td></td>
<td>resend RESPONSE Frame with RETRANSMIT=1</td>
</tr>
<tr>
<td>CLOSE</td>
<td>CLOSE</td>
</tr>
<tr>
<td>OPEN_ACCEPT</td>
<td>OPEN</td>
</tr>
<tr>
<td>ACK</td>
<td>© RESPONSE</td>
</tr>
</tbody>
</table>

Tag is not ready for reuse (this ACK could still be lost)
RESPONSE Frame Not Delivered

Tag is not ready for reuse (this ACK could still be lost)
COMMAND Frame NAK Received

Same as in SAS-1

INITIATOR | TARGET
---|---
COMMAND | NAK
NAK received for COMMAND Frame - resend COMMAND Frame
© COMMAND | ACK
COMMAND Frame ACK lost (command running, but target hasn’t sent a frame yet). If first burst is enabled, the initiator can start sending the first burst data after running query task.

Same as in SAS-1

**Diagram:**

- **INITIATOR**
  - COMMAND
  - ACK/NAK timeout
  - DONE (ACK/NAK TIMEOUT)
  - CLOSE
  - OPEN
  - QUERY TASK

- **TARGET**
  - ACK
  - DONE
  - CLOSE
  - OPEN_ACCEPT
  - RESPONSE of FUNCTION SUCCEEDED

Command active at target
COMMAND Frame ACK Lost, target processes command returning RESPONSE, XFER_RDY, or DATA

The initiator still runs DONE (ACK/NAK Timeout) and has to participate in a new connection at some point.

These pictures just show what the transport layer deals with.
COMMAND Frame NAK Lost (command not running)

Command not active at target - initiator can resend the command
COMMAND Frame Not Delivered

<table>
<thead>
<tr>
<th>INITIATOR</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td></td>
</tr>
<tr>
<td>ACK/NAK timeout</td>
<td>DONE (ACK/NAK TIMEOUT)</td>
</tr>
<tr>
<td>DONE</td>
<td>DONE</td>
</tr>
<tr>
<td>CLOSE</td>
<td>CLOSE</td>
</tr>
<tr>
<td>OPEN</td>
<td>OPEN_ACCEPT</td>
</tr>
<tr>
<td>QUERY TASK</td>
<td>RESPONSE of FUNCTION COMPLETE</td>
</tr>
<tr>
<td>@COMMAND</td>
<td>ACK</td>
</tr>
</tbody>
</table>

Command not active at target - initiator can resend the command.
Task Frame NAK Received

Same as in SAS-1

NAK received for TASK Frame - resend TASK Frame
TASK Frame ACK lost - resend

Same as in SAS-1

Initiator can resend the TASK Frame

® TASK

ACK/NAK timeout

DONE (ACK/NAK TIMEOUT)

CLOSE

OPEN

ACK

Target
TASK Frame ACK Lost, target processes TASK Frame returning RESPONSE

The initiator still runs DONE (ACK/NAK Timeout).
TASK Frame NAK Lost

Same as in SAS-1

Initiator can resend the TASK Frame
XFER_RDY Frame NAK Received
SAS-1.0: target aborts command

INITIATOR | TARGET
---|---
NAK | XFER_RDY RO=000, TPTT=123, Retry DATA Frames=1

NAK received for XFER_RDY Frame - resend XFER_RDY Frame with RETRANSMIT=1 and new TPTT

ACK | XFER_RDY RO=000, TPTT=124, Retry DATA Frames=1
XFER_RDY Frame ACK Lost
SAS-1.0: target aborts command

, Retry DATA Frames=1

DATA RO=000, TPTT=123
DATA RO=100, TPTT=123
DATA RO=200, TPTT=123
DATA RO=300, TPTT=123
DATA RO=400, TPTT=123

ACK

DONE

CLOSE

OPEN_ACCEPT

Could still send more data for TPTT=123; ignored by the target. No more sent once TPTT=124 is used

ACK

DATA RO=000, TPTT=124
DATA RO=100, TPTT=124
DATA RO=200, TPTT=124
DATA RO=300, TPTT=124
DATA RO=400, TPTT=124

Target can reuse TPTT=123 when data for TPTT=124 is received

ACK

© XFER_RDY RO=000, TPTT=124, Retry DATA Frames=1

ACK/NACK timeout

DONE (ACK/NACK TIMEOUT)

resend XFER_RDY Frame with RETRANSMIT=1 and a new TPTT

CLOSE

OPEN

resend XFER_RDY Frame with RETRANSMIT=1

ACK

Data ACKed but discarded

ACK (000)
ACK (100)
ACK (200)
ACK (300)
ACK (400)
XFER_RDY Frame NAK Lost

INITIATOR | TARGET
---|---
NAK | XFER_RDY RO=000, TPTT=123, Retry DATA Frames=1
ACK/NAK timeout
DONE | DONE (ACK/NAK TIMEOUT)
resend XFER_RDY Frame with RETRANSMIT=1 and new TPTT
CLOSE | CLOSE
OPEN_ACCEPT | OPEN
ACK | XFER_RDY RO=000, TPTT=124, Retry DATA Frames=1
XFER_RDY Frame Not Delivered

- **INITIATOR**
  - **ACK**
  - **OPEN**
  - **DONE**
  - **CLOSE**
  - **OPEN_ACCEPT**

- **TARGET**
  - XFER_RDY RO=000, TPTT=123, Retry DATA Frames=1
  - ACK/NAK timeout
  - DONE (ACK/NAK TIMEOUT)
  - resend XFER_RDY Frame with RETRANSMIT=1 and new TPTT
  - CLOSE
  - OPEN
  - XFER_RDY RO=000, TPTT=124, Retry DATA Frames=1
Write DATA Frame NAK Received

Go back to the beginning of XFER_RDY.
No need to close connection
Write DATA Frame ACK Lost

- **ACK**
- **DATA RO=000, TPTT=123**
- **DATA RO=100, TPTT=123**
- **DATA RO=200, TPTT=123**
- **DATA RO=300, TPTT=123**
- **DATA RO=400, TPTT=123**
- **ACK (000)**
- **ACK (100)**
- **ACK (200)**
- **ACK (300)**
- **ACK (400)**

- **ACK/NAK timeout**

**DONE (ACK/NAK TIMEOUT)**

**OPEN**

OPEN_ACCEPT

- **DATA RO=000, TPTT=123, Changing Data Pointer = 1**
- **DATA RO=100, TPTT=123**
- **DATA RO=200, TPTT=123**
- **DATA RO=300, TPTT=123**
- **DATA RO=400, TPTT=123**
- **ACK (000)**
- **ACK (100)**
- **ACK (200)**
- **ACK (300)**
- **ACK (400)**
Write DATA Frame NAK Lost

DATA RO=000, TPTT=123
DATA RO=100, TPTT=123
DATA RO=200, TPTT=123
DATA RO=300, TPTT=123
DATA RO=400, TPTT=123

DATA  RO=000, TPTT=123, Retry DATA Frames=1

DATA  RO=100, TPTT=123, Retry DATA Frames=1

DATA  RO=200, TPTT=123, Retry DATA Frames=1

DATA  RO=300, TPTT=123, Retry DATA Frames=1

DATA  RO=400, TPTT=123, Retry DATA Frames=1

ACK

ACK

ACK

ACK

ACK

DATA RO=000, TPTT=123, Changing Data Pointer = 1

DATA RO=100, TPTT=123, Changing Data Pointer = 1

DATA RO=200, TPTT=123, Changing Data Pointer = 1

DATA RO=300, TPTT=123, Changing Data Pointer = 1

DATA RO=400, TPTT=123, Changing Data Pointer = 1

ACK (000)

ACK (100)

ACK (200)

ACK (300)

ACK (400)

ACK/NAK timeout

DATA 400 is unexpected due to the gap, and shall be discarded by the target transport layer even though the link layer ACKs it.

DONE (ACK/NAK TIMEOUT)

Go back to the beginning of XFER_RDY.

CLOSE

DONE

CLOSE

OPEN

OPEN_ACCEPT

Changing Data Pointer=1 signals that the target can go back and start accepting data again. It might internally discard the new writes up to the gap, or rewrite.
Write DATA Frame Not Delivered

<table>
<thead>
<tr>
<th>INITIATOR</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>XFER_RDY RO=000, TPTT=123, Retry DATA Frames=1</td>
</tr>
<tr>
<td>DATA RO=000, TPTT=123</td>
<td></td>
</tr>
<tr>
<td>DATA RO=100, TPTT=123</td>
<td></td>
</tr>
<tr>
<td>DATA RO=200, TPTT=123</td>
<td>ACK (000)</td>
</tr>
<tr>
<td>DATA RO=300, TPTT=123</td>
<td>ACK (100)</td>
</tr>
<tr>
<td>DATA RO=400, TPTT=123</td>
<td>ACK (200)</td>
</tr>
<tr>
<td>ACK/NAK timeout</td>
<td>ACK (400)</td>
</tr>
<tr>
<td>DONE (ACK/NAK TIMEOUT)</td>
<td></td>
</tr>
<tr>
<td>CLOSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GO back to the beginning of XFER_RDY.</td>
</tr>
<tr>
<td>OPEN</td>
<td></td>
</tr>
<tr>
<td>OPEN_ACCEPT</td>
<td></td>
</tr>
<tr>
<td>© DATA RO=000, TPTT=123, Changing Data Pointer = 1</td>
<td></td>
</tr>
<tr>
<td>© DATA RO=100, TPTT=123</td>
<td></td>
</tr>
<tr>
<td>© DATA RO=200, TPTT=123</td>
<td></td>
</tr>
<tr>
<td>© DATA RO=300, TPTT=123</td>
<td></td>
</tr>
<tr>
<td>© DATA RO=400, TPTT=123</td>
<td>ACK (000)</td>
</tr>
<tr>
<td></td>
<td>ACK (100)</td>
</tr>
<tr>
<td></td>
<td>ACK (200)</td>
</tr>
<tr>
<td></td>
<td>ACK (300)</td>
</tr>
<tr>
<td></td>
<td>ACK (400)</td>
</tr>
</tbody>
</table>

DATA 400 is unexpected due to the gap, and shall be discarded by the target transport layer even though the link layer ACKs it.

Changing Data Pointer=1 signals that the target can go back and start accepting data again. It might internally discard the new writes up to the gap, or rewrite.
Write DATA Frame ACK Lost crossing RESPONSE

Command is done, but an ACK was apparently missed

Initiator should stop transmitting data on receipt of RESPONSE.

Target cannot reuse TPTT=123 yet

Target can reuse TPTT=123 after receipt of next command.
Write DATA Frame ACK Lost crossing XFER_RDY

The XFER_RDY burst is done, but an ACK was apparently missed because another XFER_RDY just arrived

Initiator must stop servicing the old XFER_RDY burst before sending data frames for TPTT=124.

Target can now reuse TPTT=123 when data frames for TPTT=124 arrive
Read DATA Frame NAK Received

Although DATA 400 is ACKed, the transport layer is not happy with the RO gap. For this logical unit, it does not abort the command, however; it could discard DATA 400.

When DATA with Changing Data Pointer=1 arrives, the initiator resumes accepting read DATA at the correct destination.
Read DATA Frame ACK Lost

Got 4 ACKs, but target doesn’t know what was lost (an ACK or NAK, or which one since the last ACK/NAK balance point). Must restart from last ACK/NAK balance state (RO=000 here).

Initiator honors Changing Data Pointer=1 from this logical unit

© DATA RO=000, Changing Data Pointer=1
© DATA RO=100
© DATA RO=200
© DATA RO=300
© DATA RO=400
Read DATA Frame NAK Lost

Initiator honors Changing Data Pointer=1 from this logical unit
Read DATA Frame Not Delivered

Initiator shall not abort the command just because it detected an RO gap. It could discard all the data after the gap. No need to start NAKing.

Changing Data Pointer indicates the initiator can modify its data pointers and resume accepting data.

Got 4 ACKs, but don’t know what was lost. Must restart from last ACK/NAK balance state (RO=000 here).