TO: T10 Membership

RE: Use of Class 2 for Fibre Channel Tapes

FR: Doug Hagerman

This paper describes the use of the Fibre Channel Class 2 protocol when communicating to a tape device that implements the SCSI-3 Streaming Commands (SSC) device model. Some preliminary discussion on this topic is in document T10/97-155R3.TXT.

1. Scope

1.1 The basic proposal here is to "use Class 2 for tapes".

1.2. The protocol is intended to work using the FC-PH Class 2 behavior as it applies to switched Fibre Channel, FC-AL, and FCL environments. This has numerous implications such as, for example, the possibility of out-of-order delivery of frames within a sequence.

1.3. While this discussion is narrowly targeted at SSC devices, it is believed that the protocol described here would work for all SCSI-3 device models.

2. Other possibilities

Refer to 97-xxxRx.TXT [the Crossroads proposal] for discussion of a protocol for the use of Class 3 for tapes. The Class 3 proposal is to poll the target device at certain times to determine the status of a transfer.

3. FC/FCP/SCSI Reminder

According to the FCP mapping of SCSI to Fibre Channel, each SCSI command is completely processed within a single Fibre Channel exchange.

Throughout this discussion it is held as a fundamental assumption that when a SCSI command, contained within the bounds of an exchange, is issued by the initiator every possible attempt shall be made to successfully transfer that command, and its associated data and return status, between the initiator and the target.

Proposals that require commands to be retried at the ULP level, except in the most severe and rare cases, are not under discussion.

4. Proposal

4.1. Class 2 Concepts and Rules

- Use Class 2 ACK 0 model. This is one ACK per sequence.
- Use E_D_TOV to detect most errors.
- Rely on ULP timeout for all of the few remaining errors.
- Use existing FCP Information Units (IU) and FC-PH features.

- If E_D_TOV expires before a given sequence expires, retransmit the entire IU in a new sequence using new sequence ID and counts.
- Follow all existing Class 2 rules regarding the use of ABTS and RRQ.

The basic rules are as follows. It is intended that these not conflict with the normal use of Class 2 as described in FC-PH.

a.) For each exchange, the sender starts a ULP timer (SCSI command timeout). If the timer expires before the SCSI status is successfully returned in the FCP_RSP IU, then the exchange and SCSI command have failed and this is reported to the user's program.

b.) For each sequence within the exchange, the sender starts an E_D_TOV timer upon the transmission of the last frame of the sequence. If the sequence ACK is not received by the time the timer expires, then send the same IU in a new sequence. Repeat the wait-resend procedure until until ULP timer expires.

The same rule is followed by the initiator and the target. Whoever starts a sequence applies the E_D_TOV timeout test. Context for each sequence is held until the timer expires or the ACK is received.

c.) If the target is acting as the Sequence Initiator and it is unable to successfully send the sequence and get the associated ACK, then the sequence timeout (also E_D_TOV) will cause the sequence to be aborted.

4.2. Examples

The following examples show the case of processing an error frame that occurs during the transfer of an FCP_DATA sequence. Errors that occur during the many other frames and sequences are discussed in the notes following the examples.

==

Regardless of whether the command is a READ or a WRITE, there are really only about three cases that need to be handled:

- first sequence in an exchange, including exchange initiation
- mid-exchange sequences going in either direction
- last sequence in an exchange, including exchange completion

The next version of this document will cover these three cases without specifying whether a given sequence is for READ data or WRITE data, since they are really the same when considered at the Fibre Channel level.

4.2.1. First Sequence in an Exchange

In this case, in addition to the normal sequence processing there is some overhead related to starting up the exchange.

[stuff needed here]

[but note that it's all just regular Class 2 from FC-PH]

4.2.2. Mid-Exchange Sequences

This is the "normal" case.

[stuff needed here]

4.2.3. Last Sequence in an Exchange

In this case the primary issue is running down the exchange and releasing various resources.

[stuff needed here]

==

4.3. Mappings of FCP to Class 2

This section shows how FCP is used with Class 2.

4.3.1. WRITE

Example: Normal case (no error). Transfer of 2 data sequences, each containing 4 frames of data. Target indicates its ability to accept data by use of FCP_XFR_RDY IUs.

```
Initiator          Target
--------------------------------------------

FCP_CMD ---------->

        <----------  ACK
                     Receipt of ACK by initiator indicates FCP_CMD is
ok.
                     (FCP_CMD is a single-frame sequence.)

                     A long period of time may be required here
                     for the target to find space for the data or to
                     do some preliminary media positioning.

        <----------  FCP_XFR_RDY
                     Target indicates readiness to accept one
                     sequence of data
ACK     ---------->

        ---------->  DATA sequence ID = 1, CNT = 1 (frame 1)
        ----->X      DATA sequence ID = 1, CNT = 2 (frame 2)
                     Error occurs on interconnect at "X". The frame is
lost.
```

```
        ---------->  DATA sequence ID = 1, CNT = 3 (frame 3)
        ---------->  DATA sequence ID = 1, CNT = 4 (frame 4)

Initiator has sent all the data frames,
so it starts an E_D_TOV timer for this sequence.

                     Target does not get all the frames, so it doesn't
                     send an ACK. [Does it experience a sequence
                     timeout? Probably E_D_TOV.]

Initiator's timer expires. ACK not received,
so the sequence has failed.
Initiator sends ABTS (with LS bit set) to make sure that this
sequence is aborted. [Required by FC-PH but of no apparent value.]
Initiator retransmits the data in a new sequence.

[Note requirement for support of non-ascending tranfers.]

        ---------->  DATA sequence ID = 2, CNT = 1 (frame 1)
        ---------->  DATA sequence ID = 2, CNT = 2 (frame 2)
        ---------->  DATA sequence ID = 2, CNT = 3 (frame 3)
        ---------->  DATA sequence ID = 2, CNT = 4 (frame 4)

        <----------  ACK

Receipt of ACK by initiator indicates that the sequence is ok.
Initiator may discard context for this sequence.

                     If no additional data for this [mumble] is
                     received by E_D_TOV after sending the ACK,
                     the target may discard context for this sequence.

                     When the tape is ready to receive additional
                     data, it sends another FCP_XFR_RDY.

[The wording of this example is not meant to imply that streaming is not
allowed,
but we must verify that streaming works properly.]

        <----------  FCP_XFR_RDY
                     Target indicates readiness to accept one
                     sequence of data
ACK     ---------->

        ---------->  DATA sequence ID = 3, CNT = 1 (frame 1)
        ---------->  DATA sequence ID = 3, CNT = 2 (frame 2)
        ---------->  DATA sequence ID = 3, CNT = 3 (frame 3)
        ---------->  DATA sequence ID = 3, CNT = 4 (frame 4)

        <----------  ACK

                     Receipt of ACK indicates that the sequence is ok.
                     Proceed to next sequence.
                     Target observes that it has enough data to
                     satisfy the requirements of the SCSI WRITE command,
                     so it sends the SCSI status back.

        <----------  FCP_RSP
                     With SCSI Status.
ACK     ---------->
                     Target closes exchange and deletes command context.
```

Initiator waits for E_D_TOV after sending ACK to make sure no more sequences are coming. (This would be a resend of the FCP_RSP if the last ACK had been lost on its way to the target.) After timout expires, Initiator discards context for this exchange.

==

4.3.2. READ

Example: Transfer of 2 data sequences, each containing 4 frames of data. Assume the host can accept all the data specified in the command.

```
Initiator          Target
--------------------------------------------

FCP_CMD ---------->
        <----------  ACK
                     Receipt of ACK by initiator indicates FCP_CMD is
ok.
                     (FCP_CMD is a single-frame sequence.)

                     A long period of time may be required here
                     for the target to get the data from the media.

        <----------  DATA sequence ID = 1, CNT = 1 (frame 1)

             X<----  DATA sequence ID = 1, CNT = 2 (frame 2)
                     Error occurs on interconnect at "X". The frame is
lost.

        <----------  DATA sequence ID = 1, CNT = 3 (frame 3)
        <----------  DATA sequence ID = 1, CNT = 4 (frame 4)

                     Target has sent all the data frames,
                     so it starts an E_D_TOV timer for this sequence.
```

Initiator does not get all the frames, so it doesn't send an ACK.
[Does it experience a sequence timeout? Probably E_D_TOV.]

```
                     Target's timer expires. ACK not received,
                     so the sequence has failed. Retransmit sequence.
```

[Does frame header give enough info to allow Initiator to put the re-sent data in the right place?]

[Note requirement for support of non-ascending tranfers.]

```
        <----------  DATA sequence ID = 2, CNT = 1 (frame 1)
        <----------  DATA sequence ID = 2, CNT = 2 (frame 2)
        <----------  DATA sequence ID = 2, CNT = 3 (frame 3)
        <----------  DATA sequence ID = 2, CNT = 4 (frame 4)

ACK     ---------->
                     Receipt of ACK indicates that the sequence is ok.
                     Proceed to next sequence. [Not meant to imply
                     that streaming is not allowed.]

        <----------  DATA sequence ID = 3, CNT = 1 (frame 1)
        <----------  DATA sequence ID = 3, CNT = 2 (frame 2)
        <----------  DATA sequence ID = 3, CNT = 3 (frame 3)
        <----------  DATA sequence ID = 3, CNT = 4 (frame 4)

ACK     ---------->
                     Receipt of ACK indicates that the sequence is ok.
                     Proceed to next sequence.

        <----------  FCP_RSP
                     With SCSI Status.
ACK     ---------->
                     Target closes exchange and deletes command context.
```

Initiator waits for E_D_TOV after sending ACK to make sure no more sequences are coming. (This would be a resend of the FCP_RSP if the last ACK had been lost on its way to the target.) After timout expires, Initiator discards context for this exchange.

==

4.4.

4.5. Notes on Examples

Any frame in the exchange may fail. The following lists the handling of each possible case. This is intended to follow the FC-PH protocol exactly.

4.5.1. Loss of FCP_CMD single-frame sequence. In this case the target never sees the command. When initiator's E_D_TOV timer expires it first sends an ABTS with the LS bit set. This has no effect on the target (which simply ignores this attempt to abort a sequence in an exchange it has never heard of).
Then the initiator resends the FCP_CMD IU using the same information and the same exchange identifier in a new sequence.

[See FC-PH 29.7.1.1. for discussion of ABTS at start of exchange.
[Note need for BA_ACC frame to acknowledge the ABTS. What's the point?]

4.5.2. Loss of ACK after FCP_CMD. In this case the target saw the command

and has constructed exchange context. When initiator's E_D_TOV timer expires it first sends an ABTS with the LS bit set; this aborts the sequence but maintains the exchange. Then it resends the FCP_CMD IU using the same information and the same exchange identifier in a new sequence.

4.5.3 Loss of ABTS after loss of FCP_CMD sequence. In this case the target has not seen the command anyway. [Check for ACCEPT needed for handshake on ABTS.]

4.5.4. Loss of ABTS after loss of ACK after FCP_CMD sequence. In this case the target has constructed exchange context. When the FCP_CMD is resent by the initiator, the target observes that the exchange identifier is for an existing exchange and thus ignores the redundant information.

5. Advantages of Class 2

- Uses existing FC-PH features.
- Uses existing FCP Information Units.
- Good opportunity to automate the processing of each exchange.
- Sequence Initiative management is already defined in FC-PH.
- Sequence streaming is already defined in FC-PH.