FCP-3: Revsion 3a Discussion Items

Dave Peterson, CNT

(T10/04-132r0)

From an email sent T10 Refelctor (4/26/2004) by Kevin Butt, IBM:

I have some suggested discussion items for FCP WG related to FCP-3r3a:

1)

In the description of the SRR:

<<

Addressing:

The S_ID field designates the initiator requesting the information retransmission. The D_ID field designates the target that is to receive the request. In the event that the target responds to the SRR with an FCP FC-4 Link Service Reject, the target shall return CHECK CONDITION status with the sense key set to HARDWARE ERROR and an additional sense code of INITIATOR DETECTED ERROR MESSAGE RECEIVED.

>>

Calling out a sense key of HARDWARE ERROR is a bad idea. When customers see a sense key 4 they assume bad hardware and automatically return the drive. This creates increased field support costs resulting in No Defect Found conditions. A different Sense Key needs to be used, and I suggest COMMAND ABORTED.

2)

Please add a statement to SRR that indicates that when a target receives an SRR request for status that it is allowed for the target to retransmit the data as well as the status and that the host shall replace the data with the new data received. This request stems from conditions where the target can only detect certain errors (e.g. CRC Error) after the data has already been transmitted. If we can resolve the method for getting the correct data to the host then we can increase the likelihood of success.

The new text in 12.4.1.5 FCP_RSP IU Recovery would be that indicated by << ... >>:

When an error in transmitting an FCP_RSP IU is detected, the initiator shall issue an SRR FC-4 Link Service frame in a new Exchange to request retransmission of the FCP_RSP IU. The target shall first transmit the ACC for the SRR, then shall retransmit << either >> the FCP_RSP IU << or FCP_DATA IUs and FCP_RSP IU >> in a new Sequence.

3)

<<

11.5 Read Exchange Concise Time-out Value (REC_TOV) REC_TOV is used by the initiator to provide a minimum polling interval for REC and by the target for FCP_CONF IU error detection. The REC_TOV timer shall be implemented such that at least one REC_TOV period passes between transmission of a command and the first polling for Exchange status with the REC Extended Link Service.

>>

Can we state that the first REC may optionally be sent prior to REC_TOV but that subsequent RECs will be REC_TOV?

The reason for this is that many HBAs will use E_D_TOV for their first REC and Targets may wish to use a short time for response to an FCP_CONF IU.

The reason for short timeouts for FCP_CONF IUs would be that they are almost instantaneous and if not then we are in an ERP state. Also, if we change the requirement such that if an initiator receives a REC when it has an open exchange, but before it receives an FCP_RSP_IU that it shall initiate a REC to determine the state of the Exchange, then the target would be able to tell the initiator that it needs recovered. We noticed that some initiators would respond to the REC in such a way that the target would end up waiting RR_TOV before being able to free up the exchange.

4)
The following text in 12.4.1.5 FCP_RSP IU Recovery is a little confusing. I read it to mean that an exchange is being recovered where a check condition ocurred but the FCP_RSP IU was lost. In this case the target is being told to try an FCP_XFER_RDY IU unless the original command did not intend to transfer data.
I am confused as to how the target can attempt to send an FCP_XFER_RDY IU after it has already terminated the transfer with a CHECK CONDITION. Does this paragraph intend to talk about a single exchange or multiple exchanges that get confused?
I think this needs to be modified such that the intent of this paragraph is clear.

An Exchange carrying a command that was terminated by a CHECK CONDITION requesting an FCP_CONF IU prior to transferring data may have the same REC values as an Exchange carrying a command having an FCP_XFER_RDY IU not received by the initiator. For a command transferring data from the initiator to the target with a non-zero FCP_DL, the parameters for the SRR shall indicate that an FCP_XFER_RDY IU is expected from the target. The target is aware of the actual present state of the transfer and response and shall either retry the FCP_XFER_RDY IU or, if the actual data transfer length for the command was zero, retry the FCP_RSP IU.

5)
In 12.4.1.5 FCP_RSP IU Recovery clause the changes inside << ... >> need to be made:

The Exchange information retained shall include data transfer information, data descriptors, and FCP_RSP IU information.
If retransmission is enabled between the initiator and target, FCP_RSP IU information shall be:
a) discarded RR_TOV after the FCP_RSP IU was transmitted to the initiator; or,
b) discarded after a new Exchange with the same OX_ID << change 'and' to ',' >> S_ID << and task retry identification >> is received.

6)
Due to differing values of RR_TOV in various versions of FCP_2, it is unsafe to use the larger value of RR_TOV in some areas and unsafe to use the smaller value in others. This has resulted in some vendors being unwilling to use the most current RR_TOV values and thereby causing an inability to perform second level error recovery.
Additionally, RR_TOV may be modified by a Mode Page at any time. A method is needed to ensure that both the initiator and target are using the same RR_TOV. Since the mode page is involved and can be modified at any time by an application there is no safe way for the initiator to use the mode page to get this value.
I suggest we add a new FC4LS either:
a) specifically for RR_TOV information called Request RR_TOV Value (RRV) that returns the RR_TOV in seconds or
b) for an exchange of FC4 support information and call the request Request FC4 Information (RFI) and the response
FC4 Information (FI).
The RFI would request the FI be sent. If no response in REC_TOV the error recovery would be to ABTS is then resend a vendor specific number of times.
The FI would contain the RR_TOV value and any other FC4 information we decide on - perhaps with spare fields.
The FI would be sent in response to an RFI and in response to any FC4 event that causes a change in the supported fields.

7)
Table 20 - task management Flags does not mention QUERY TASK. A comment should

be made as to support or non-support of the QUERY TASK function. I would suggest a comment to the effect that the QUERY TASK function is not supported as QUERY TASK since the REC mechanism performs this function in FCP, but REC give the HBA a method but it does not give the application a method. The QUERY TASK could provide that. I have had several customers ask me if something like this is available.

8)
In 12.4.1.7 FCP_DATA IU Recovery - Read, indicating a HARDWARE ERROR sense key is a bad idea as previously described.

9)
12.5.2 REC
If a response to an REC is not received within 2 times R_A_TOVELS, the initiator shall:
1) send an ABTS(Exchange) for the REC followed by an RRQ if a BA_ACC is received for the ABTS; and
2) send another REC in a new Exchange.
If the response to the second REC is not received within 2 times R_A_TOVELS, the initiator should send an
ABTS(Exchange) for the REC followed by an RRQ if a BA_ACC is received for the ABTS;

RR_TOV was defined as:
If RETRY bit is set to 1:
>= REC_TOV +
2xR_A_TOVELS + 1 sec.
for the reason of being able to successfully perform second level error recovery. However, the above description for REC has the initiator wait 2xR_A_TOVELS leaving only REC_TOV+1 sec for step 2. The final sentence quoted above specifies 2xR_A_TOVELS for the second REC. That adds up to 4*R_A_TOVELS for second level error recovery.
In seconds that means there is specified 40 seconds for second level error recovery to be performed but RR_TOV is defined as >= 24 seconds.

In FCP-2r7 the time to wait is only R_A_TOVELS and RR_TOV is defined as 3*REC_TOV+1:
12.5.2 REC
If a response to an REC is not received within R_A_TOVELS, the initiator shall:
1) send an ABTS(Exchange) for the REC followed by an RRQ if a BA_ACC is received for the ABTS; and
2) send another REC in a new Exchange.
If the response to the second REC is not received within R_A_TOVELS, the initiator should send an
ABTS(Exchange) for the REC followed by an RRQ if a BA_ACC is received for the ABTS; Other retry mechanisms after the second REC fails are optional and, if implemented, shall comply with FC-FS
ABTS(Exchange) may be required to clear resources associated with the original failing Exchange if the retry mechanisms are not successful.

The changes to clause 12.5.2 REC in Second Level Error Recovery and the definitions of RR_TOV seem to be fighting each other. Why was 12.5.2 time changed to 2xR_A_TOVELS? It should be changed back to 1xR_A_TOVELS so RR_TOV does not clean up the exchange prior to the conclusion of this. Extending RR_TOV to greater than it already is, is not a viable solution. Existing devices have ULP timeout values as low as 30 seconds. Also, there are application clients that will not wait 45 seconds for response to some commands (e.g. INQ). If those commands fail the device is considered offline and no further communication will be attempted.