FCP Revision 01 Letter Ballot Comment Database

Person making comment + number	tech/edit	Page	Sec/table/fig locator	Comment	Proposed Solution	Resolution	Status
BrianHart-001	T	85 (69 hardcopy)	12.2.2 first paragraph, item b)	Target requirement for FCP_RESID_UNDER is missing:: There is no requirement for a target to set FCP_RESID_UNDER if a read operation results in the transfer of fewer than FCP_DL bytes. The 4th paragraph of section 9.4.2, requires: "Because there were fewer bytes provided than required by FCP_DL, the FCP_RESID_UNDER bit...shall be set to one in the FCP_RSP IU...." But this occurs in the context of a discussion of a write operation. There is no similar requirement that FCP_RESID_UNDER be set appropriately in the context of read operations. Section 12.2.2 first paragraph bullet (b) requires the initiator to detect underrun. This may imply a requirement for the target, but it would be better explicitly stated.	1) Break section 9.4 paragraph 4 after "...the target FCP_Port shall discard the excess bytes.", -and- 2) Amend the following sentence to replace "Because there were fewer bytes provided than required...." with "If an operation results in the transfer of fewer bytes than required....".		
BrianHart-002	T	43 (27 hardcopy)	6.3.4	Sequence level recovery is not defined:: Every usage of the phrase "Sequence level recovery" has the indicated capitalization. This is a marked usage and suggests that the phrase is being used as a term of art. However, the phrase is not defined by the standard, so is left to assume its normal English meaning. It is not clear how the normal meaning of the phrase relates to the concepts of the standard. Specifically, it is not clear when an FCP_Port "ha[s] agreed to Sequence level recovery". What constitutes this agreement should be clearly defined as it qualifies several sections describing recovery. This has ramifications for data integrity (see, e.g., issue (4) below).	In section 6.3.4, subsection "Word 3, Bit 8: RETRY", add a sentence following the first sentence of the third paragraph: "....in both the request payload and in the accept payload. In this case the initiator and target shall have agreed to Sequence level recovery."		

BrianHart-003	T	89 (73 hardcopy)	12.4.1.5	Recovery is insufficiently required:: Several recovery sections (e.g. 12.4.1.5) are qualified by: "This procedure shall be used only by FCP devices that have agreed to Sequence level recovery". That is, agreement to Sequence level recovery is necessary but not sufficient to imply that an initiator or target will perform the defined recovery. The standard provides no mechanism for an agreeable FCP_Port to communicate its actual intent to follow the recovery procedures, so it is possible that an initiator and target might make opposite choices. There are cases, though, where either both or neither initiator and target must perform the recovery in order to preserve data integrity. A target, for example, might agree to Sequence level recovery but elect not to perform the FCP_RSP IU recovery described in section 12.4.1.5. Not being subject, then, to the restrictions in 12.4.1.5, the target would be at liberty to discard exchange information as soon as an FCP_RSP was sent. If the FCP_RSP were lost, an otherwise timely REC by the initiator would be rejected by the ta	Replace the qualifications at the heads of sections 12.4.1.3, 12.4.1.4, 12.4.1.5, 12.4.1.6, and 12.4.1.7 with: "This procedure shall be used by and only by FCP devices that have agreed to Sequence level recovery." Note the larger effect on 12.4.1.3 than on the others.		
BrianHart-004	T	82 (66 hardcopy)	11.3	R_A_TOV (re)definitions drop vital guarantee:: Section 11.3 states: "R_A_TOV has two separate components, labeled R_A_TOVseq_qual and R_A_TOVels." FC-FS-2 contains no mention of separate components of R_A_TOV. It's unclear whether FCP's R_A_TOV component timers inherit substance or merely name from FC-FS-2. FC-FS-2 section 20.2.1.4 provides a guarantee: "R_A_TOV represents E_D_TOV plus twice the maximum time that a frame may be delayed within a Fabric and still be delivered." The notion that R_A_TOV encompasses the maximum fabric delivery time is vital to the definition of RR_TOVseq_init (Table 30) and the recovery mechanisms that depend on it (e.g. section 12.4.1.5). If R_A_TOVels does not inherit substantially from FC-FS-2 R_A_TOV then this vital guarantee is dropped. Even if R_A_TOVels does inherit substantially from FC-FS-2 R_A_TOV, Table 30 flatly redefines the duration of R_A_TOVels as 2 or 10 seconds without mention of maximum fabric delivery time,	Amend Table 30 - Timer summary NOTE 1 to add a sentence: "R_A_TOV for ELS shall encompass the maximum time that a frame may be delayed within a Fabric and still be delivered." Note that boundedness of R_A_TOVels directly affects boundedness of RR_TOVseq_init, and so has implications for boundedness of REC_TOV.		

BrianHart-005	T	88 (72 hardcopy)	12.4.1.3	REC_TOV floor allows REC vs FCP_CMND race:: Section 12.4.1.3 equates REC reject (with "Logical error"/"Invalid OX_ID-RX_ID combination") to the loss of the FCP_CMND and prescribes retransmission of the FCP_CMND. But an initiator would see the same reject in the case where the REC merely arrived at the target ahead of the FCP_CMND. In that case retransmission of the FCP_CMND could result in a loss of data integrity. Arrival of REC ahead of FCP_CMND could be prevented by ensuring that REC is not transmitted until it is certain that the FCP_CMND is either delivered or lost. FC-FS-2 section 20.2.1.3 limits to three the actions whose duration is bounded by E_D_TOV; frame delivery across a fabric is not among those. Rather, FC-FS-2 section 20.2.1.4 describes R_A_TOV as the timer that encompasses the maximum frame delivery time. In order to ensure REC is not sent prematurely, REC_TOV's range must therefore encompass R_A_TOV rather than E_D_TOV.	Replace REC_TOV range of ">= E_D_TOV + 1s" with ">= R_A_TOV" in Table 30 - Timer summary. -or- Replace section 12.4.1.3 paragaph 2 with: 'If the target reports the exchange invalid (i.e. the initiator FCP_Port receives an LS_RJT for the REC with the reason code of "Logical error" and reason code explanation set to "Invalid OX_ID-RX_ID combination"), the initiator shall not retransmit the FCP_CMND and shall notify the application client appropriately.' Note that if 12.4.1.3 is allowed to stand a modification may still be in order. Verb "retransmit" following the parenthetical is in the imperative mood and would better be declarative: "...), the initiator shall retransmit...."		
BrianHart-006	T	81 (65 hardcopy)	Table 33	Lack of REC_TOV ceiling allows REC vs exchange discard race:: REC_TOV is described in the timer summary table (Table 33) as a range with a floor but no ceiling. No mechanism is provided to communicate the choice of REC_TOV between initiator and target. This allows the possibility that an initiator may choose a REC_TOV that is arbitrarily large and that differs from the REC_TOV chosen by the target. Further, section 11.5 describes REC_TOV as the "minimum polling interval" for REC and states that a duration of "at least" REC_TOV occurs before REC may be sent. REC_TOV is not a ceiling on the REC polling interval. Section 12.4.1.5 attempts to ensure that a target will maintain exchange information until a timely REC arrives by requiring that the target retain the information for up to RR_TOVseq_init after sending the FCP_RSP. Table 30 suggests RR_TOVseq_init should be ">= REC_TOV + 2xR_A_TOVels + 1s" (in the RETRY case), but this is insufficient. The target must necessarily base its RR_TOVseq_init on its own REC_TOV since it has no	All three of: 1) Modify section 11.5 first paragraph to add a sentence encouraging prompt polling by initiators: "....first polling for Exchange status with the REC ELS. Initiators should transmit REC promptly after REC_TOV expiration. Table 31...." -and- 2) Modify Table 30 to set an appropriate ceiling for REC_TOV, perhaps one of: "<= R_A_TOV", "<= R_A_TOV + E_D_TOV", or "<= 2xR_A_TOV". -and- 3) Modify Table 30 to set a floor for RR_TOVseq_init based on the REC_TOV ceiling, making RR_TOVseq_init's range: ">= ceil(REC_TOV) + R_A_TOV + 1s" (with "R_A_TOV" allowing time for the REC to traverse the fabric and "1s" as an allowance for initiator promptness failings). -- Or just: -- Replace section 12.4.1.3 paragaph 2 with: 'If the target reports the exchange invalid (i.e. the initiator FCP_Port receives an LS_RJT for the REC with the		

BrianHart-007	T			The target must retain the exchange information not only long enough to ensure that any REC from the initiator will have arrived before discard, but additionally long enough to allow time for the target to transmit the REC ACCEPT, the ACCEPT to cross the fabric and reach the initiator, the initiator to process the ACCEPT and respond with SRR, and the SRR to cross the fabric to arrive again at the target. Practically, I think that means the target must retain the exchange information for an additional "2 x R_A_TOV + 1s" (two fabric traversals and some grace for promptness of target and initiator) _beyond_ what I had considered originally.			
KevinButt-001	T		12.4.2.3 & 4.7 - General comment	There needs to be a shall statement prohibiting consecutive exchanges with the same OX_ID as well as making the clear statement that "rapid" (whatever that means) reuse of OX_ID is highly frowned upon. If a shall statement cannot be made, then a statement is needed that clearly acknowledges that rapid OX_ID reuse is dangerous behavior. It seems like rapid OX_ID reuse would create a colossal mess if command queuing were enabled.			