To: T11
From: Dal Allan
Date: 2-9-98
Subject: FC-AL Loop Initialization

An email I received from Barry Reinhold this morning contains information of value to us in the FC-AL-2 review which is on our current agenda. I have extracted the relevant material below, which consists of:

- some recommendations relative to the FC-AL-2 draft specifications
- a proposed starting point for a state diagram addressing loop initialization and recovery
- a set of comments on rev 5.8 (December 24, 1997) of the FC-AL-2 draft standard.

Inclusions from Barry's email:

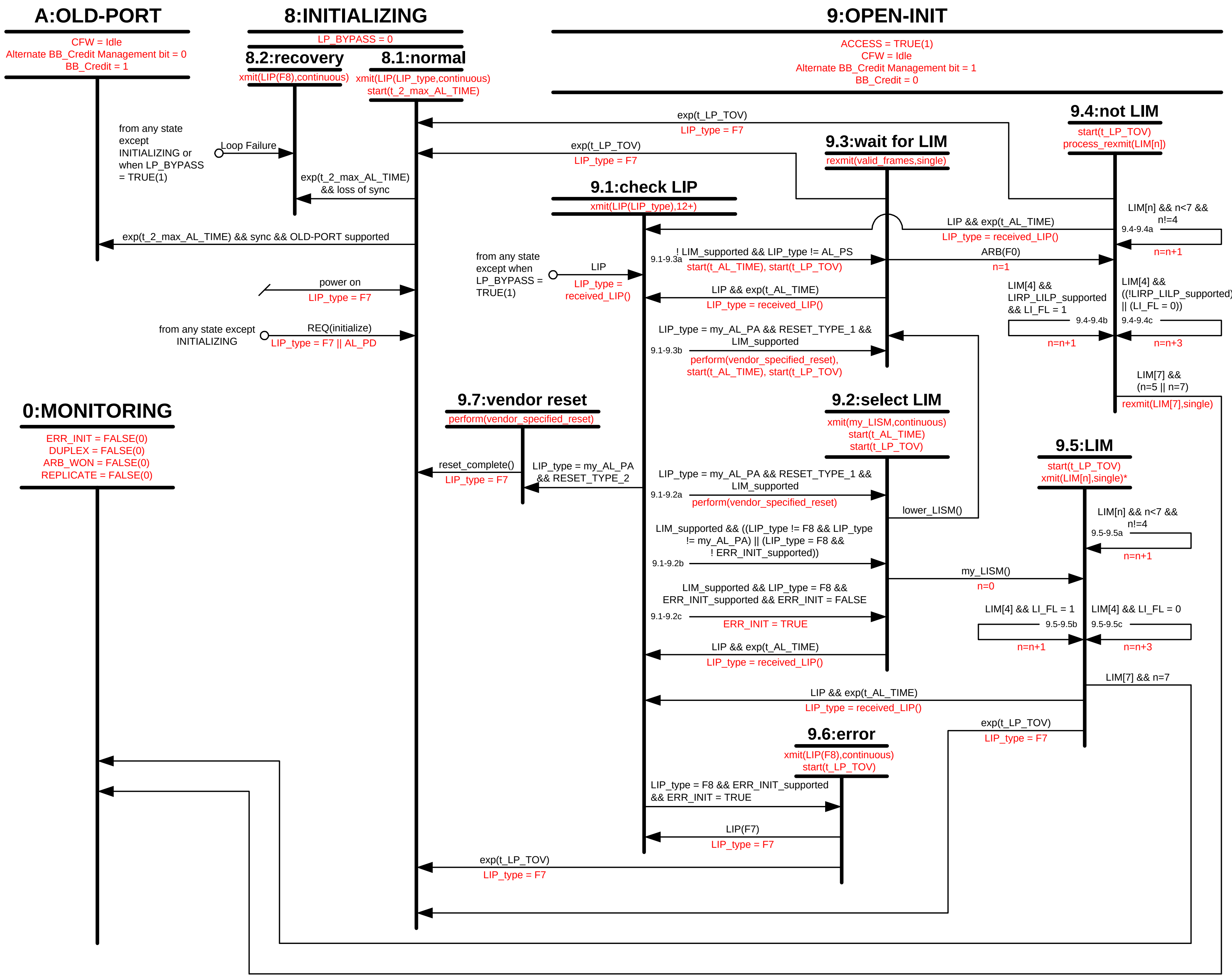
I (Barry Reinhold) would like to propose that the FC-AL-2 standards body address the following issues:

1. Document the proper external behavior of a L_Port during the loop initialization and recovery process using a normative state diagram with labeled transitions and a table of notes attached which references the transitions. The state diagram should be the highest level of priority in terms of conflict resolution with the transition notes being second and lose text the lowest priority. Examples of this nature are common and can be found in IEEE 1394 documentation, almost all IEEE 802 documentation as well as ANSI FDDI. A potential starting point for a state diagram is included in this message. The diagram is not an attempt to document the current state of FC-AL-2. After significant work we abandoned that effort as we felt the current draft was too ambiguous to document without significant changes in the external behavior. The provided state diagram makes a number of simplifications that we believe would significantly enhance the operation of arbitrated loops. However, our real goal is to achieve precise specification of external behavior.
2. Remove all references to Annex I.2.2 loop recovery. There are a number of reasons for this, based on the current state of the standard, current field experience with FC-AL products and the limited effectiveness of the trace operation in FDDI. Let me elaborate on each of these points:
   - The degree of specification of Annex I.2.2 is so loose that it is impossible to understand what proper behavior is. It is poorly integrated into the behavior of the LPSM. Actual use of this process in an arbitrated loop is very unlikely to be successful and could easily leave nodes in a bypassed state for no reason. The whole concept of removing devices from a loop using a set of AL_PAs that may no longer be valid is questionable – but if allowed should not be up to the discretion of an implementor but carefully documented by the standard.
   - To our knowledge no product has actually implemented this process and there is not a need to address backwards compatibility. Its existence causes needless complexity in an area where the standard should be simple and robust. The process of selecting a loop master in order to get a loop going is not that complex and very similar processes have been done in both Token Ring and FDDI. These solutions can't be adopted but the process can be simplified and made more robust.
   - Field experience has shown that the FDDI trace mechanism though well defined and documented, is seldom useful in resolving problems even though FDDI can easily bound the fault domain and has the ability to reach the faulting node through a second path. The basic problem is that when the station has gone bad, and not the physical connection itself, it has seldom responded to trace. In FC-AL we do not really have a good mechanism to isolate the fault domain and we do not have a second path to get to the fault. Our chances of actually recovering from a loop failure condition through the use of mechanism outlined in Annex I.2.2 are almost nil.
   - If Annex I.2.2 recovery is removed it is reasonable to eliminate all state transitions in INITIALIZING and OPEN-INIT associated with LPB and LPE. This would help simplify the standard significantly and not impact currently running loops. This would help in the robustness of loops as the use of LPB and LPE when the AL_PA is not solid can lead to unexpected behaviors.
3. Define the external behavior of a station when it enters the loop during power up. If this proves to be unacceptable at least limit the duration of time during which arbitrary behavior can take place. There are too many cases in the field where stations do not behave in a defined manner until disk power up cycling has completed or drivers loaded.

4. Although we realize that an effort has been made and rejected for establishing a standard for hubs there is still a need to bring legacy devices into a FC-AL loop in a manner that will allow stable operation. Concentrating devices that shield the loop from the destructive behavior of current L_Ports appear to offer a realistic avenue towards stable loop operation. The goals for the concentrating device should be basic:
   - Properly insert current legacy L_Ports into the loop. This can be done in any number of ways but the standard should provide a set of specifications that will ensure proper and consistent behavior during loop insertion.
   - A clear market segment has developed for concentrating devices and vendors are creating products without guidance of a standard. This is currently causing a new type of interoperability problem and as additional features are added the potential for new problems will grow.

The following section contains the state diagram and associated documentation:

General Notes – State Diagrams are expressed using vertical shafts to represent states and horizontal lines to denote transitions between states.



To maintain consistency with FC-AL 4.5 the INITIALIZING and OPEN-INIT states are broken into sub-states. When entering a sub-state the from outside of the parent state the entry actions for both the parent state and the sub-state are performed. The combined entry actions of the state and sub-state are assumed to start simultaneously. When entering a different sub-state within the same parent state only the entry actions of the sub-state are to be performed.

Transitions are illustrated with the triggering condition located above the horizontal line and any actions to be performed with the transitions below the line. All transitions are performed while remaining in the previous state, before entry to the new state.

All times are assumed to be global and are only reset by explicit action. An expired timer remains expired until it is reset with the start() procedure.

The following rules are associated with event processing:
1. All conditions are evaluated in the context of the current state.
2. If the conditions for a transition are satisfied then: (a) perform the actions associated with the transition in the current state, (b) enter the new state, (c) perform the entry actions for the new state, and (d) evaluate conditions for exiting the new state.
3. No two transitions in a given state may be true at the same time or the state diagram is in error.

State Variables
1. ERR_INIT – Set to true to indicate that the L_Port has already attempted to initialize the loop by sending LISM frames while receiving LIP F8.
2. ERR_INIT_supported – Set to true to indicate that the L_Port will send LIP F8 when receiving LIP F8 after an attempt has been made to bring up the loop by sending LISM frames. If set to false the L_Port will not send continuous LIP F8 frames.
3. LI_FL – The value of bit 8 of the 16 bit LI_FL field as defined in section 10.4.1.
4. LIM[0..7] – An array containing the different sequences that are used in the loop initialization process. This array is logically initialized using the following C programming language notation: LIM[ ] = {ARB F0, LIFA, LIPA,LIHA,LISA,LIRP,LILP,CLS}
5. LIM_supported – This is true if the L_PORT can perform the operations required of a loop master.
6. LIP_type – The type of LIP to be transmitted in the current or next state. May take on the value of F8, F7 or any valid AL_PD as defined in Annex K.
7. RESET_TYPE_1 – This is true if the implementation supports a type of reset that when externally observed would be identical to the external behavior of receiving LIP F7. The station shall continue in the OPEN-INIT sub state 9.3
8. RESET_TYPE_2 – This shall be the logical opposite of RESET_TYPE_1. It indicates that the vendor specific reset causes behavior different than that expected from a LIP F7
9. T_AL_TIME – A timer set to the value of AL_TIME.
10. T_LP_TOV – A timer set to the value of LP_TOV.

State Procedures (do not return a value)
1. Lower_LISM() – Returns true if (rx_D_ID < my_D_ID) or ((rx_D_ID = my_D_ID) && (rx_S_ID < my_S_ID)) or ((rx_D_ID = my_D_ID) && (rx_S_IS = my_S_ID) && (rx_WWN < my_WWN)).
2. My_LISM() – Returns true if the received LISM frame's D_ID, S_ID, and WWN match the D_ID, S_ID, and WWN of my LISM frame.
3. Perform(arg) – This procedure is a notational convenience that allows a complex action to be expressed. The actions being performed by this function do not alter that behavior of the LPSM.
4. Process_rexmit() – Performs the necessary actions on the received frame before retransmitting it. The actions are defined in section 10.4.3.
5. Rexmit(arg) – Transmit the last logical item received.
6. Start(arg) – This procedure starts the timer passed in as an argument. It is logically equivalent to starting a count down timer by loading the timer with the time value specified as an argument.
7. Xmit(arg1,arg2) – Transmit the first argument in the manner described by the second argument. The second argument may be either single, 12+, or continuous.

State Functions (return a value)
Expire() – This function is true when the timer associated with the argument reaches zero.

Receive_LIP() – Returns the type of LIP received in accordance with the FC-AL 3.1.22. This function can take on the values F7,F8 or a valid AL_PD as defined for the reset LIP.
Reset_complete() – True when the L_Port has completed the vendor specific reset process and is ready to reenter the INITIALIZING state.

Transitions – The following text provides additional information about particular transitions.

1. X-8.1 – The value of LIP_type is established by the LPSM as defined in section 7.8. It shall be either LIP F7 or LIPr.
2. 9.1-8.1a – The L_Port shall respond to the LIPr by performing a vendor specific reset. The station shall remain in state 9.7, Vendor Reset, until it is ready to reenter the initializing state. The external behavior of the station is as if it were bypassed until the station is ready to reenter the loop.
3. 9.1-9.2a - If the first character following the LIP is the AL_PA of this L_Port the L_Port is to perform a vendor specific reset procedure. The external behavior of the L_Port shall be the same as if a LIP F7 was received.
4. 9.1-9.3a and 9.1-9.3b – If the first character following the LIP is the AL_PA of this L_Port the L_Port is to perform a vendor specific reset procedure. The external behavior of the L_Port shall be the same as if a LIP F7 was received.
5. 9.4-0 – The L_Port transitions into monitoring only when CLS is received LIM[7]. This transitions is further constrained by checking to see that six or eight sequences have been sent. If these conditions are not met the station will time out on LP_TOV.
6. 9.4-9.4a – Each of the loop initialization sequences must be received in order to be processed by this transition. The exception to this rule is the LISA frame which is processed by 9.4-9.4b and 9.4-9.4c. If a loop initialization sequence is out of order the frame will not be forwarded and the LIM is expected to time out on LP_TOV.
7. 9.1-9.7 – This transition is taken if the reset process will cause external behavior different from that of receiving a LIP F7. This expected behavior of this type of reset is that of a power on reset.
8. 9.5-9.5a – When n=0 the xmit(LIM[n],single) function shall not transmit a single item but shall transmit ARB F0 continuously.

FC-AL-2 v5.8 comments

INITIALIZING state

- The cases of normal initialization and Loop recovery should be two separate states or substates. The behavior is different depending on which operation is performed.
- Annex I.2.2 should be removed from the standard. It causes needless complexity and ambiguity.
- The expected behavior for an L_Port detecting Loop Failure is unclear. Some interpretations of the behavior for a device powering up receiving no signal:
  1. Enter normal initializing section and transmit LIP(F7,F7) for up to 2*max AL_TIME. If Annex I.2.2 is supported, transmit LIP(F8,F7) for 2*AL_TIME followed by LBPyx for up to 2*AL_TIME for each AL_PA (recovery). Then what?
  2. Enter normal initializing section and transmit LIP(F7,F7) for up to 2*max AL_TIME. If Annex I.2.2 is not supported, transmit LIP(F8,F7) for 2*AL_TIME. Then what?
  3. Enter Loop recovery and transmit LIP(F8,F7) for 2*AL_TIME. Then what?
- If a device powers up receiving no signal, and it directly enters recovery, it cannot perform Annex I.2.2.
- The last paragraph of INITIALIZING states that if a device begins to either bypass or enable another port by transmitting LPB or LPE, it shall transmit the primitive until it is received or until the request has been dropped. By definition of what the port is transmitting, it is in the recovery section at this point. What if the port receives LIP? Should it continue to send the LPB or LPE or should LIP cause a transition into OPEN-INIT?

OPEN-INIT state

- The vendor specific reset mentioned in the first bullet point should not cause the external behavior of an L_Port to be undefined. The device then *may* continue with the initialization procedure. What happens if it does not? After the reset, the L_Port should go to the INITIALIZING state, as if from power up.
- It is not specified in the text what an L_Port should do if the received LIP is a reset LIP and y≠AL_PA of the L_Port. This behavior should be the same as for a LIP(F7).
- The third to last bullet point on page 41 (reception of ARB(F0)) should be clarified. What should happen if a device is transmitting its own LISM frames and receives ARB(F0) before receiving its LISM back? This should never happen, but if it does, shouldn't the L_Port ignore the ARB?
- The second to last bullet point on page 41 (reception of CLS) should be constrained. Receiving CLS at any point during the initialization process should not cause a transition into MONITORING. This transition should only be allowed after receiving LISA or LILP.
- On page 64, the last paragraph states "The frame header shall not be used to validate the Loop Initialization Sequences." What exactly does this mean?
- In section 10.3, P63, the last paragraph should have all but the first sentence should be removed. The entire paragraph should read, "The L_Port that is attempting to initialize shall make the transition to the INITIALIZING state (REQ(initialize)) (see 8.4.3, item 21)."[end]
- The second bullet point on page 41 states "…the L_Port shall ignore LIPs for AL_TIME" after receiving LIP(F7). Is this timer started upon entering the OPEN-INIT state or after the L_Port has finished transmitting LIP?
- The transmission and reception of LPB and LPE should not be allowed during the Loop initialization procedure (i.e., the L_Port is in the INITIALIZING or OPEN-INIT state).

MONITORING state

- On page 32, "If the L_Port detects Loop Failure, on its inbound fibre and LP_BYPASS is TRUE(1), the LPSM shall transmit LIP(F8), but remain in the MONITORING state." There is not a case when LP_BYPASS is FALSE(0). Although this transition exists in Table 4, it should be in the text as well. There should also be an explicit statement requiring the L_Port to stop transmitting LIP and continue retransmitting received transmission words once the Loop Failure condition has disappeared.