Robust training process timing

in support of comment #i-113

Adee Ran, Intel Corp.

Supporters

Status

- As of D3.0, all three PMDs use the control function defined in Clause 72, with the following additional requirement:

  > In addition to the coefficient update process specified in 72.6.10.2.5, after responding to the first request after training begins, the period from receiving a new request to responding to that request shall be less than 2 ms.

- This addition is aimed at ensuring that the training process is not stalled by an unresponsive partner.
- As will be shown, the text does not achieve this goal, and may even encourage implementations that don't promote interoperability.

Problem statement

- The condition "after responding to the first request after training begins" for the 2 ms requirement, assumes that this first response indicates a stable control channel… (TRAIN_LOCAL)
- But the control channel depends on having frame lock – and frame_lock can be lost and re-acquired.
- The time for acquiring or re-acquiring frame_lock is currently unspecified. It may take more than 2 ms.

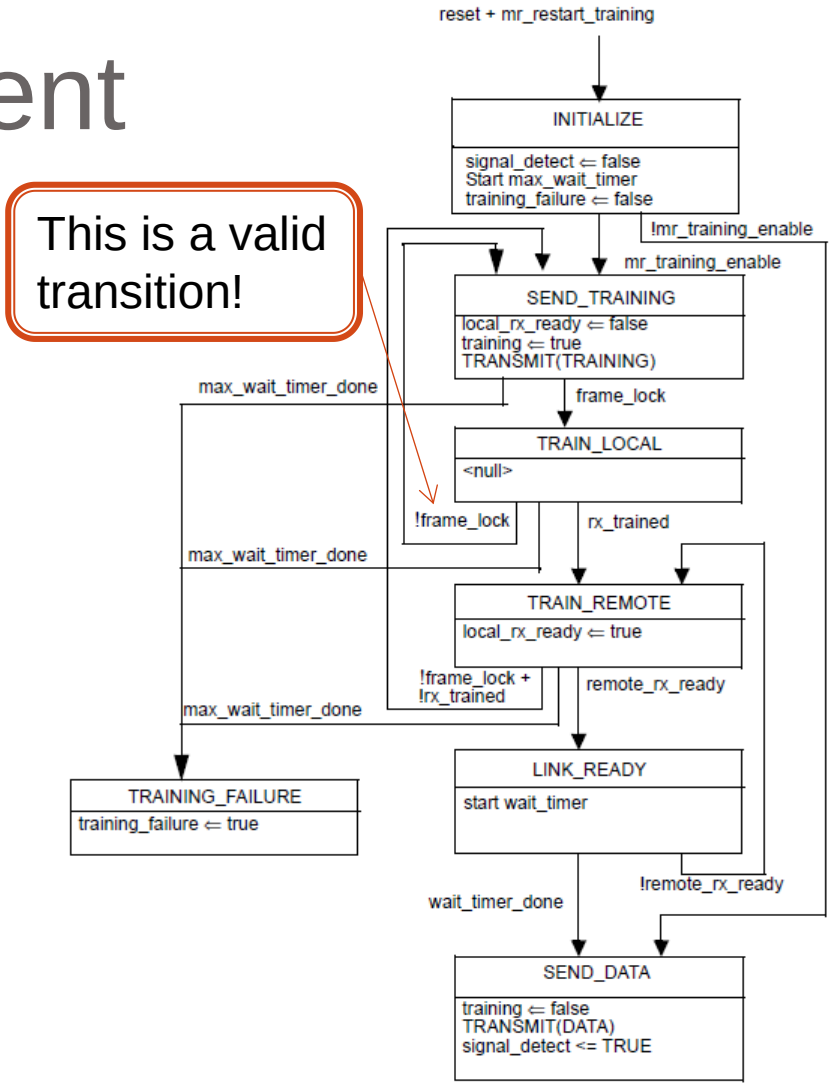

Figure 72–5—Training state diagram

Problem statement

- Typically, handshake processes include timeouts, with a specified “escape path” when timeouts occur. Examples include
  - 10GBASE-R LPI Receive state diagram (Figure 49-13) – timeouts cause transition to RX_LINK_FAIL
  - Auto-negotiation arbitration state diagram (Figure 73-11) – timeouts cause transition to TRANSMIT_DISABLE
  - Figure 72-5 itself has TRAINING_FAILURE
- However, for a receiver that cannot respond within 2 ms after responding to the first request, there is no specified behavior. Since it is a normative requirement, there is no compliant behavior in this case.

Problem statement

- A safe way to avoid the problem (and be compliant) could be “don’t respond to the first request!”
  - Or simply delay the response until signal quality is good enough…
  - A receiver may even send its own requests to improve signal quality while delaying the responses (e.g. start by requesting “preset” and wait for completion in order to continue).
  - Nothing in the current text prevents using this strategy. Obviously, if both sides use it, we get a deadlock.
- Even without a deadlock, delaying the first response deprives the partner of control channel usage. Without limiting this delay, the problem is not solved.

Proposed solution

- A minimal change which prevents deadlock and control channel starvation is:
  1. Allow a reasonable period for possible RX initialization and TX transient effects. During this time the RX is not required to respond.
  2. Start measuring the time when AN pages stop being transmitted. To limit transient effects, specify maximum time for valid TX signal.
  3. After the initialization period, requests should be acknowledged within 2 ms.

Detailed proposal

- **Change the third paragraph of 92.7.12 as follows:**

  "In addition to the coefficient update process specified in 72.6.10.2.5, ~~after responding to the first request after training begins~~ within 50 ms of beginning training (as demarked by the entry to the AN_GOOD_CHECK state in Figure 73-11), the period from receiving a new request to responding to that request shall be less than 2 ms. The start of the period is the frame marker of the training frame with the new request and the end of the period is the frame marker of the training frame with the corresponding response. A new request occurs when the coefficient update field is different from the coefficient field in the preceding frame. The response occurs when the coefficient status report field is updated to indicated that the corresponding action is complete."

- **Change the last sentence of 73.6.10 as follows:**

  When a PHY is connected to the MDI through the Transmit Switch function, the signals at the MDI shall conform to all of the PHY's specifications within 20 ms.

Backup

Alternative – full solution

- frame_lock is the essential status for the operation of the control channel:
  - Without frame_lock, the receiver cannot decode status messages, so it cannot send new (outgoing) requests:

    "A new request to increment or decrement shall not be sent before the incoming status messages for that tap revert to not_updated." (72.6.10.2.3.3)
  - Therefore there should be no motivation to delay frame_lock.
- A reasonable set of requirements is
  1. Timely response to incoming requests (within 2 ms) when frame_lock is true.
  2. frame_lock initial acquisition and re-acquisition within reasonable times.
  3. No change of outgoing requests when frame_lock is false.
- Specify maximum times to acquire and re-acquire frame lock, *with compliant escape paths*
  - Re-acquisition should be fast, to prevent starvation of the control channel.
  - Initial acquisition timer can be longer to allow start-up activity.
  - Expiration of timers leads to TRAINING_FAILURE.

Modified diagram

State diagram is based on the original (Figure 72-5).

New behavior marked in blue.

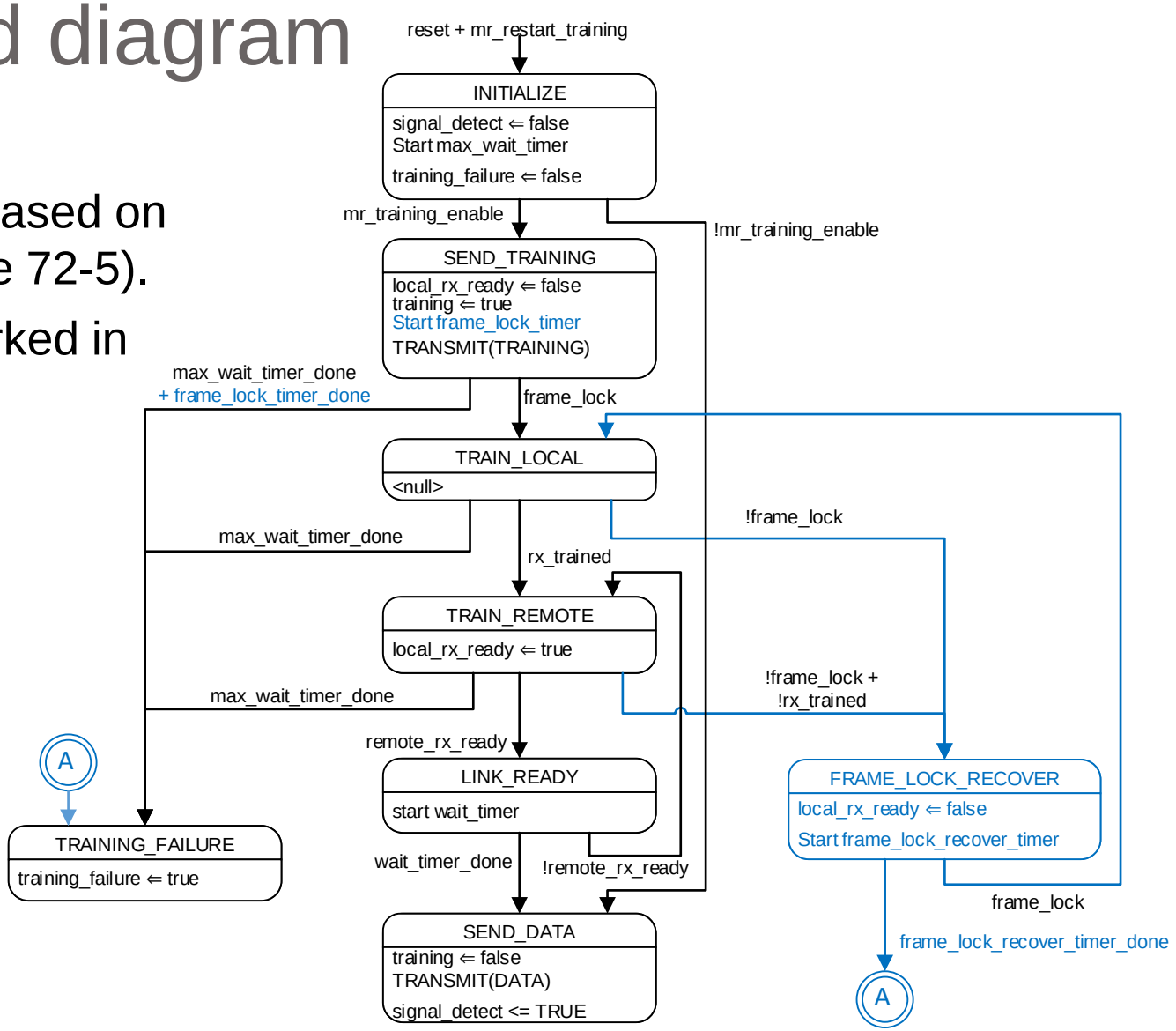

Figure 92-X – Training state diagram

Proposal

- Implement in subclause 92.7.12:
  - Add two new timers, frame_lock_timer and frame_lock_recover_timer
  - Use a modified training state diagram as shown.
  - Add explicit statement that the coefficient update field (72.6.10.2.3) is kept unchanged when frame_lock is false.
  - Make the 2 ms response time requirement conditional on frame_lock.
- Detailed text proposed for subclause 92.7.12 is submitted (ran_3bj_02_0314.pdf).
- Clause 93 uses an identical function, so refer to Clause 92.
- In clause 94, the changes should be split between 3 subclauses (94.3.10.6.4, 94.3.10.7.5, 94.3.10.11).
  - Refer to figure and timer definition in Clause 92, or repeat.
  - Using same timer value enables >4000 100GBASE-KP4 training frames – still safe.

Timer settings proposed

- Allow 50 ms for initialization activity
- Allow 2 ms for re-acquisition of frame lock
  - Same “quantum” used for required response
  - ~1700 training frames in 100GBASE-KP4
  - >11,000 training frames in 100GBASE-CR4 and 100GBASE-KR4
- How many requests does the training process enable in the worst case?
  - If timers are always fully consumed, the number of transactions before is link_fail_inhibit_timer expires is $\left\lfloor \frac{500-50}{2+2} \right\rfloor = 112$
  - If frame_lock isn’t lost, the number becomes 225

Comment against D3.0

Cl **94** *SC* **94.3.10.7.5** *P* **293** *L* **21** # i-113

RAN, ADEE — Intel Corporation

Comment Type **TR** *Comment Status* **X**

The additional requirement to respond to requests following the first acknowledged request in less than 2 ms may be impossible to fulfill if the frame_lock variable is set to false, e.g. due to SLIP function (see figure 72-4). There is currently way to abort the coefficient update state diagram or the training state diagram in that case; so there is no compliant behavior when this requirement can't be met.

It is unusual for such "handshake" related state diagram in the receiver not to have a compliant abort path. Examples include: TRAINING_FAILURE state in figure 72-5; several paths leading to TRANSMIT_DISABLE in figure 73-11; and RX_LINK_FAIL in figure 49-13.

It is possible that a designer wishing to avoid violating this requirement would defer its response to the first request (possibly, until the SLIP condition is unlikely). Such a delay is still compliant, but would undermine the purpose of the PMD control function.

Comment also applies to subclause 92.7.12 and 93.7.12.

SuggestedRemedy

A detailed remedy will be submitted separately.

Proposed Response *Response Status* O

Comment against D2.2

Cl **92** *SC* **92.7.12** *P* **197** *L* **13** # 83

Ran, Adee Intel

Comment Type **T** *Comment Status* **D** *control reponse time*

The required response time definition change from D2.1 creates a requirement that may not be possible to meet in practice, without providing a graceful abort option. Making this requirement normative is a real problem: we don't porvide a test definition and it's difficult to claim that this is correct by design.

With the curret text, a way to guarantee conformance by design is to never respond to any request; that might be the only way to ensure conformance (and we don't want that to happen).

The text in D1.1 was conditional on the state of frame_lock and a product could be designed to meet it (be correct by design). The change is part of the response to my comment #94 against D1.1, but neither the original text nor the suggested remedy for that comment involved a normative statement with the problems above.

Note that existing text in 72.6.10.2.3 and its prevents sending any update requests until the curresponding status is not_updated. This implies that frame_lock is set. Thus sending requests implies being able to timely respond to incoming requests (but not vice versa; therefore adding an indication in the status report is preferred).

Comment applies to clauses 93 and 94 as well.

SuggestedRemedy

Revert to D1.1 text and use the suggested remedy for comment #94 against D1.1 (indicate the value of frame_lock in the status report field).

Proposed Response *Response Status* **Z**

REJECT.

This comment was WITHDRAWN by the commenter.

Comment against D2.1

Cl **92** *SC* **92.7.12** *P* **193** *L* **18** # **94**

Ran, Adee Intel

Comment Type **T** *Comment Status* **A**

The response time requirement is dependent on the status of frame_lock_i which may be difficult to verify (e.g. if the MDIO interface is unavailable) and synchronize with a captured waveform. In addition, it is not available to the link partner.

It is relatively easy to make the lane frame lock state available as part of the status report field. This information would be very useful in analyzing link training issues and thus promote interoperability.

Comment applies to clauses 93 and 94 as well.

SuggestedRemedy

In clauses 92 and 93, assign cell 14 of the status report field (currently reserved) to represent the value of the PMD status variable frame_lock_i.

In clause 94, use cell 7 of the status report field instead of cell 14 (14 is already assigned, 7 is currently reserved).

Editorial license granted.

Response *Response Status* C

ACCEPT IN PRINCIPLE.

After initial frame lock the the response time should be 2 ms regardless of whether the receiver loses frame lock or not.

Replace:
"In addition to the coefficient update process specified in 72.6.10.2.5, when frame_lock_i is TRUE for lane i
(where i represents the lane number in the range 0 to 3), the period from receiving a new request to responding
to that request shall be less than 2 ms."

With:
In addition to the coefficient update process specified in 72.6.10.2.5, after responding to the first request after training begins, the period from receiving a new request to responding to that request shall be less than 2 ms.