Refactoring APSU FSMs to separate ILT and RTS functions (comments #132-137, 140)

Tom Huber, Nokia
Marco Mascitto, Nokia
Luz Osorio, Nokia

Supporters

- Gary Nicholl, Cisco
- Mike Dudek, Marvell
- Matt Brown, Qualcomm
- Jeffery Maki, HPE

Introduction

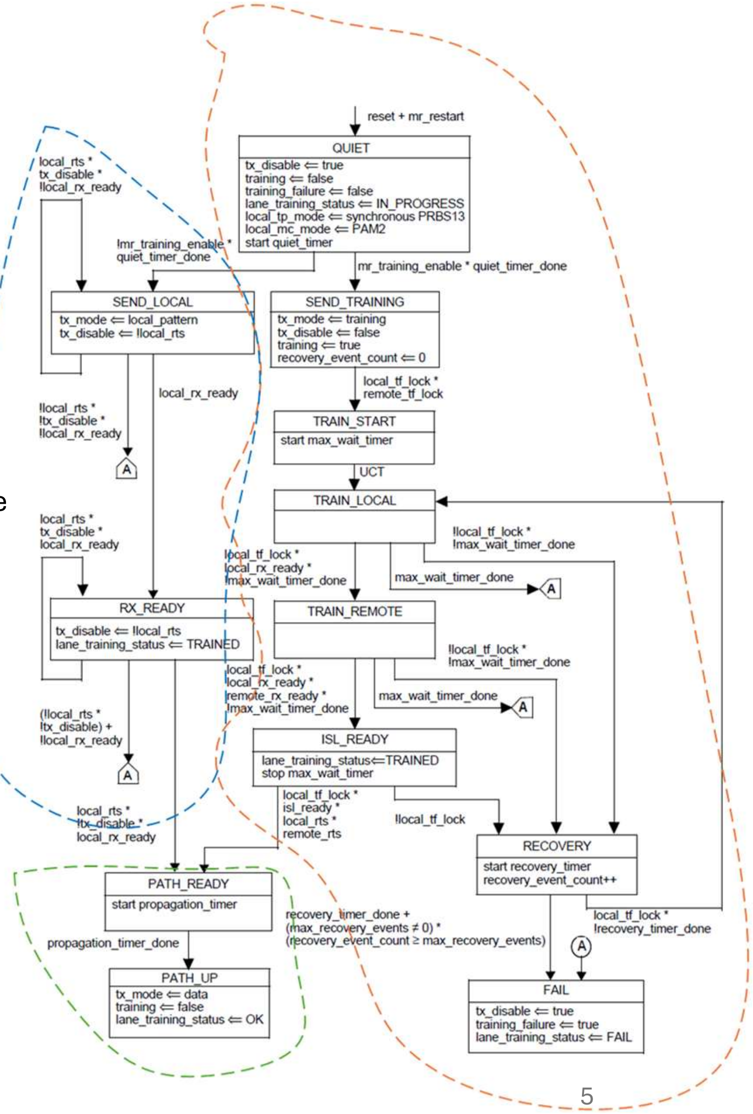
- Comments 132-137 and 140 all relate to the architecture of APSU, specifically, that the ILT and RTS functions and associated FSMs are intertwined in ways that make the description complex and make it complicated to extend the RTS function to PMDs that do not use the ILT function
  - Coherent PMDs in P802.3dj
  - Future PMDs?
- This presentation explains how the functions and associated FSMs can be refactored to reduce the coupling between them and simplify extension to PMDs that use RTS but not ILT *without changing the behavior that is already defined*
- Detailed proposals to resolve the comments are provided

Current architecture of APSU FSMs

- *Training control* FSM has multiple purposes, covering the ILT function and part of the RTS function:
  - Training for a single lane of an ISL (ILT function)
  - Signaling RTS status (Signal_OK status) downstream (part of RTS function)
    - When training is enabled, even though the training frame structure is used for the signaling, this signaling is <u>not</u> link training; it can only occur after training is completed. The training frames are the 'local pattern' that is used for signaling not-RTS and RTS
    - When training is disabled, the signaling is via squelching or transmitting a local pattern
  - Managing the transition to mission data (part of RTS function)
    - When training is enabled, this correlates to stopping transmission of training frames
    - When training is disabled, this correlates to stopping transmission of the local pattern
- *RTS Update* FSM provides part of the RTS function:
  - Manages the transition to mission clock for ISLs that are not always on mission clock
  - Sets the local_rts variable that drives the RTS status that is communicated downstream by the "training control" FSM

Figure 178B-10 “Training control state diagram” has three distinct purposes that should be separated

- The orange set of states (training mode) are related to ILT; these are *per-lane*
  - Two other FSMs (current figures 178B-11 and 178B-12) run when this FSM is in the TRAIN_START, TRAIN_LOCAL, and TRAIN_REMOTE states to control the details of lane training for 200G PAM4 ISLs via the local_tf_lock, remote_tf_lock, local_rx_ready, and remote_rx_ready variables
  - These states should **not** control tx_mode for the interface; that needs to be controlled by the RTS function
- The blue set of states (local_pattern mode) are related to not-RTS or RTS signaling for an *interface* that does is not using ILT for training
  - These states are showing transitions related to the RTS function (including control of what is transmitted)
  - If training is disabled (or maybe not supported at all) for an ISL, some other FSM or text should describe how isl_ready gets set to true, which is what the RTS function needs to see, and the RTS FSM should control the tx_mode transitions
  - *Keeping the same interface into the RTS FSM for all ISLs makes RTS extensible*
- The green set of states are entered when both directions of all lanes for the interface are RTS; these are *per-interface*
  - The RTS FSM is where tx_mode should be manipulated
  - *Using the training frame structure to signal RTS status does not make that signaling part of the ILT process*



Blue state behavior of current FSM

- In both SEND_LOCAL and RX_READY, the transmitter is squelched if local_rts is false and sends the local_pattern if local_rts is true, regardless of the status of the receiver
    - This is accomplished by the tx_disable ← !local_rts statement
- The difference between those two states is whether the local receiver is ready
    - Two states are needed because the local receiver being ready (local_rx_ready = true) and the upstream adjacent interface being ready (local_rts = true) can occur in either order
- The resulting behavior is:
    - Until the upstream adjacent interface is RTS, the tx is squelched
    - When the upstream adjacent is RTS, the local pattern is transmitted
    - When both the upstream adjacent interface is RTS and the local receiver is ready, transition to PATH_READY
- While remote_rts doesn't explicitly appear in this branch, it is implicitly true if you're receiving the local pattern, since that means the peer is RTS (which is what remote_rts means)

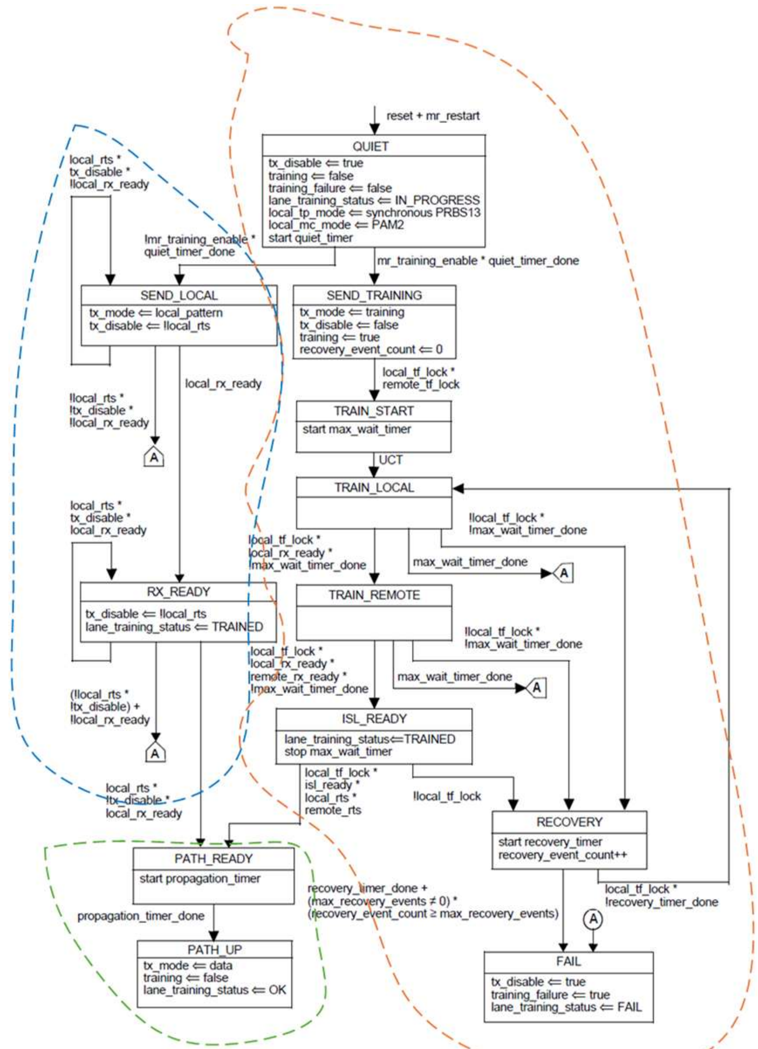
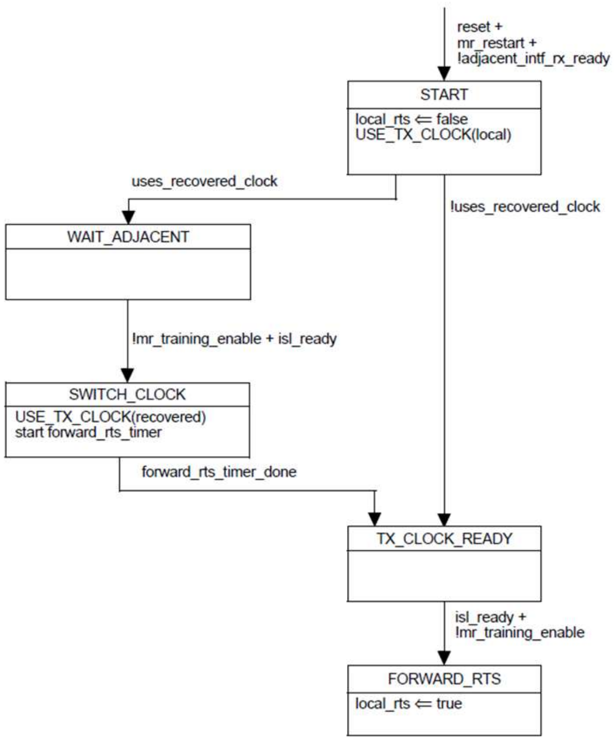


Figure 178B-9 “RTS update state diagram” should manage the overall RTS process, not just the transition to mission clock and setting local_rts



- There is nothing in this FSM related to the tx_mode for the interface
  - tx_mode manipulation needs to be here to enable RTS support on PMDs that don’t use ILT
  - Relates to the green states in figure 178B-10
- mr_training_enable should not appear in this FSM
  - The RTS function should not depend on whether or how training is done; it just needs an indication that training is done
  - Relates to the blue states in figure 178B-10
- There is no provision for local_rts or remote_rts becoming false after it has been true, other than resetting the interface
  - This topic is addressed in osorio_3dj_01_2605
- As slavick_3dj_03_2603 showed, some states in this diagram can be compressed

Goals of refactoring APSU functions

- Separate the ILT and RTS functions such that:
  - The ILT function manages only link training, i.e., achieving optimal configuration of the EQs at the two ends of the ISL
  - The RTS function manages the orderly transition from power-up through training, switching to mission clock, and switching to mission data
  - The FSMs are directly tied to those two functions, rather than having the RTS function split across multiple FSMs
- Enable use of RTS function and FSM by ISLs that do not use ILT
  - Simplifies extension of APSU feature to coherent PMDs in 802.3dj
  - Simplifies reuse of APSU feature in future projects and other SDOs
- Avoid changes to existing behavior

Characteristics of ISLs

Characteristic	200G/lane IMDD Electrical PMD/AUI	200G/lane IMDD Optical PMD	200G/lane IMDD PMD/AUI with ILT disabled	Coherent Optical PMD (LR1)	Coherent Optical PMD (ER1)	Future PMDs/AUIs
Uses ILT per 178B.7 for EQ adjustment	Y	N	N	N	N	???
Uses ILT per 178B.7 for precoder and modulation config	Y	Y	N	N	N	???
How is !RTS (SIGNAL_OK = IN_PROGRESS) signaled	E1 training frames control bit 10 = 1 status bit 15 = x	O1 training frames control bit 10 = 1 status bit 15 = x	Squelched signal (SEND_LOCAL state)	Squelched signal	Via bits in the ER1 frame overhead	???
How is RTS (SIGNAL_OK = READY) signaled	E1 training frames control bit 10 = 0 status bit 15 = 1	O1 training frames control bit 10 = 0 status bit 15 = 1	Local pattern (RX_READY state)	Local pattern	Via bits in the ER1 frame overhead	???
How is RTS (SIGNAL_OK = OK) signaled	Data frames	Data frames	Data frames	Data frames	Via bits in the ER1 frame overhead	???

control bit 10 (continue training) is !local_rts
status bit 15 (receiver ready) is local_rx_ready – so 1 if training is finished

To support all PMDs, RTS signaling needs to be decoupled from ILT

Fundamental assumptions (1)

- The RTS function provides orderly transition to mission mode across a set of concatenated ISLs that compose a path
  - RTS controls what is transmitted (tx_mode), the transition from local clock to mission clock, and the transition to mission data
  - It does this by signaling RTS status via service interfaces and across AUIs/PMDs; with refactoring, this signaling does not depend on ILT!
  - RTS is used on all ISLs in the path
- The ILT function only controls training, i.e., the exchange of information to configure the EQ correctly on both ends of the link
  - ILT is used only on ISLs that support ILT
- The only interaction between the two functions is ILT informing RTS that training is completed

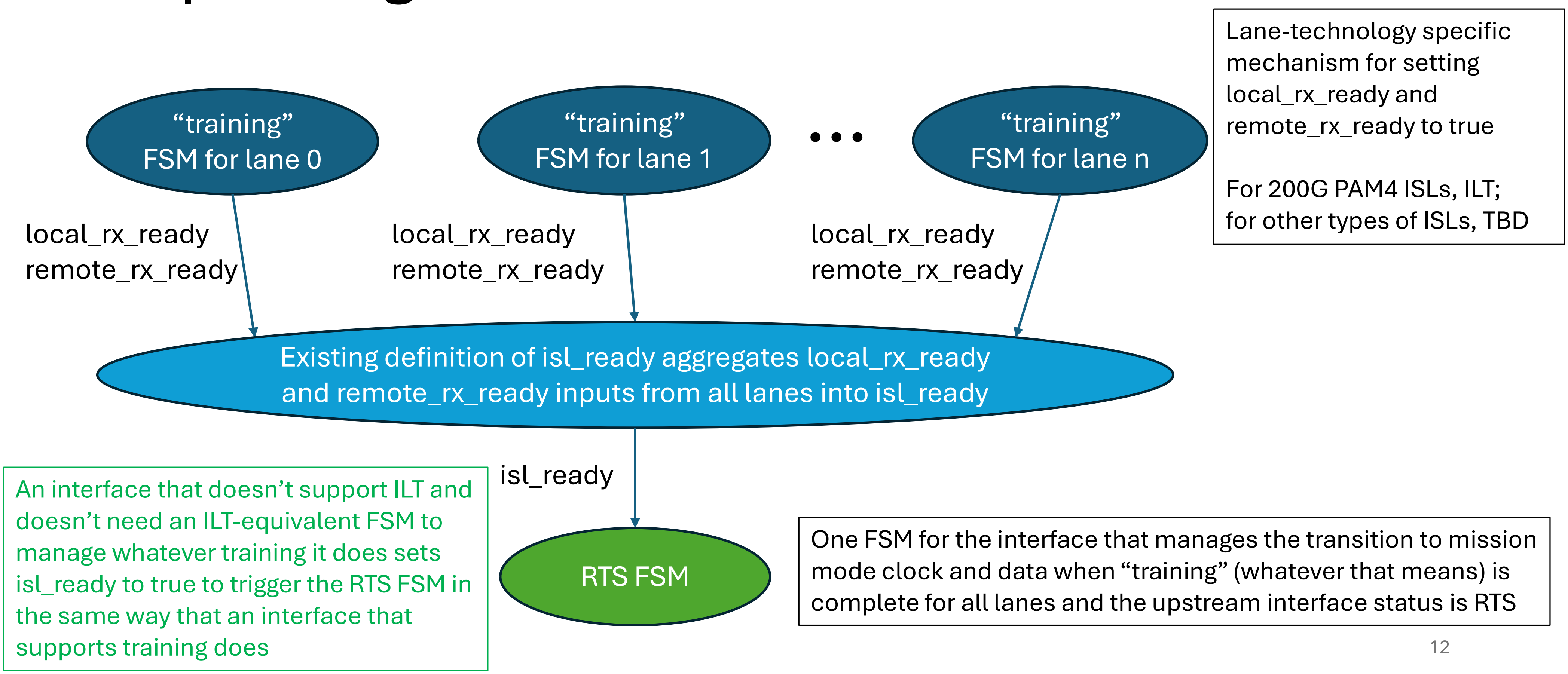
RTS signaling is not part of ILT, even if it uses the same frame structure as ILT!

Fundamental assumptions (2)

- Any form of signaling !RTS and RTS status can be abstracted as a "local pattern" that is transmitted based on the state of the ISL
  - E1 or O1 training frame with control bit 10 and status bit 15 set a particular way
  - PRBS31
  - Bits in the ER1 OH frame plus a pattern mapped into the payload area
  - Squelching is a (degenerate) local pattern

Characteristic	200G/lane IMDD Electrical PMD/AUI	200G/lane IMDD Optical PMD	200G/lane IMDD PMD/AUI with ILT disabled	Coherent Optical PMD (LR1)	Coherent Optical PMD (ER1)	Future PMDs/AUIs
How is !RTS (SIGNAL_OK = IN_PROGRESS) signaled	E1 training frames control bit 10 = 1 status bit 15 = x	O1 training frames control bit 10 = 1 status bit 15 = x	Squelched signal	Squelched signal	Via bits in the ER1 frame overhead	???
How is RTS (SIGNAL_OK = READY) signaled	E1 training frames control bit 10 = 0 status bit 15 = 1	O1 training frames control bit 10 = 0 status bit 15 = 1	Local pattern (PMD-specific)	Local pattern (PMD-specific)	Via bits in the ER1 frame overhead	???

High level vision for APSU architecture separating ILT and RTS functions



Setting isl_ready to true

ISLs that support ILT

- If ILT is enabled, isl_ready becomes true when local_rx_ready and remote_rx_ready are true for all lanes
- If ILT is disabled, isl_ready is always true

ISLs related to 800GBASE-LR1 or 800GBASE-ER1 PMD

- The endpoint of the ISL is more than just the PMD; it includes also the PMA and FEC sublayers, as these are monolithic for these PHYs
- isl_ready becomes true when the receive function has performed the processes that enable the link to work (e.g., aligning to DSP frame, aligning to FEC frame, etc.) and the RTS function to operate; this can be described in text or in FSMs specific to the PMA or FEC sublayers, as appropriate

Future PMDs

- TBD

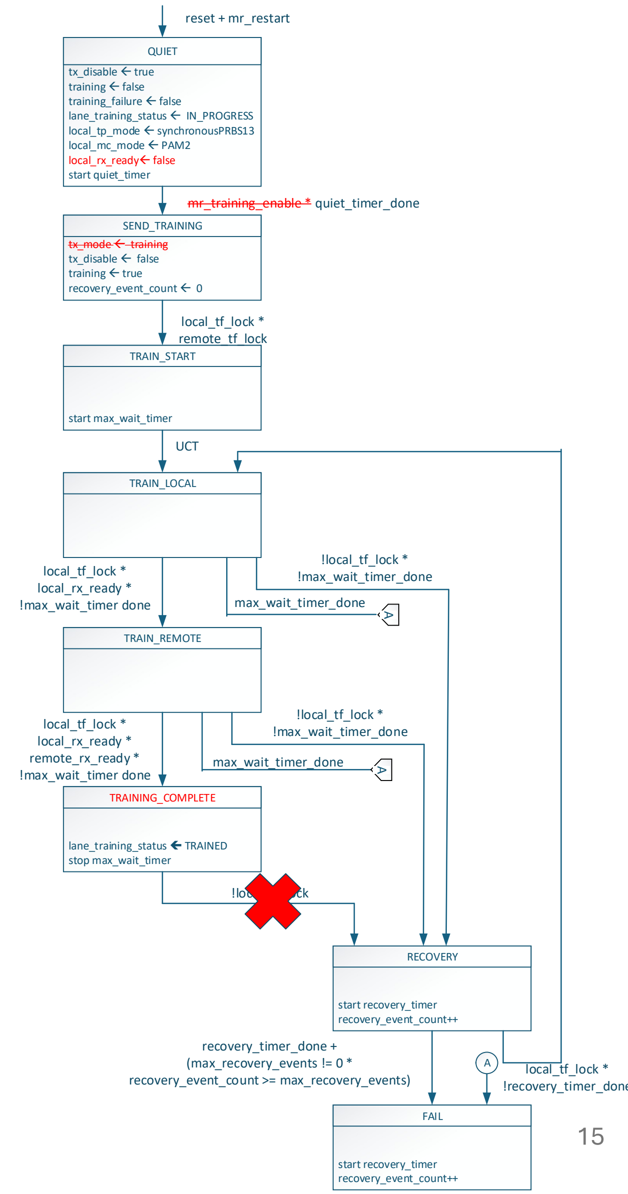
How to refactor the FSMs

- The ILT FSM should be only the orange states in Figure 178B-10
  - The purpose of this FSM is to control the process of training for a lane; that process ends when local_rx_ready and remote_rx_ready are both true
- Revise figure 178B-9 to cover all aspects of the RTS function
  - Include the green states from figure 178B-10
  - Include control of tx_mode, which, along with re-defining isl_ready to be “training is done”, effectively also replaces the blue states in figure 178B-10
  - Remove dependencies on mr_training_enable; RTS works the same way for all ISLs, whether or not they use ILT
- Revise definitions and related text to align to the new FSMs

This doesn’t change the behavior, only the description!

ILT FSM

- Retains the orange states shown on slide 5, except for the minor changes shown in red
  - local_rx_ready set to false in the QUIET state for clarity (existing definition means it must be false at this point)
  - mr_training_enable doesn’t need to be checked; this FSM only runs on interfaces with training enabled
  - tx_mode is an interface parameter, so it should not be set by a per-lane FSM; the RTS FSM handles this
  - Renamed “ISL_READY” to “TRAINING_COMPLETE” and made it a terminal state (removing transition to RECOVERY); this FSM is done once configuration of the EQs is done
    - Fault handling is addressed in osorio_3dj_01_2605



tx_mode changes

- tx_mode currently has three values that don't fully distinguish the status:
  - training means training frames are being transmitted, either for the purpose of training or for the purpose of indicating RTS status after training is completed; the interface may or may not be on mission clock
  - local_pattern means the interface is not using ILT; the interface may be squelched or transmitting a local pattern and may or may not be on mission clock
  - data means the interface is on mission clock and data
- The proposed refactoring uses four values with more clearly defined meanings:
  - training means training information (whatever that means for the ISL) is being sent
  - notRTS means training is completed but the interface is not yet on mission clock because the upstream adjacent interface is not yet ready; not-RTS status is being sent
  - RTS means training is complete and the interface is on mission clock; RTS status is being sent
  - data means the interface is on mission data

tx_mode summary

Current definitions

tx_mode value	Training complete?	Mission clock?	Mission data?
training	Maybe[1]	Maybe[2]	N
local_pattern[3]	N/A	Maybe[2]	N
data	Y	Y	Y

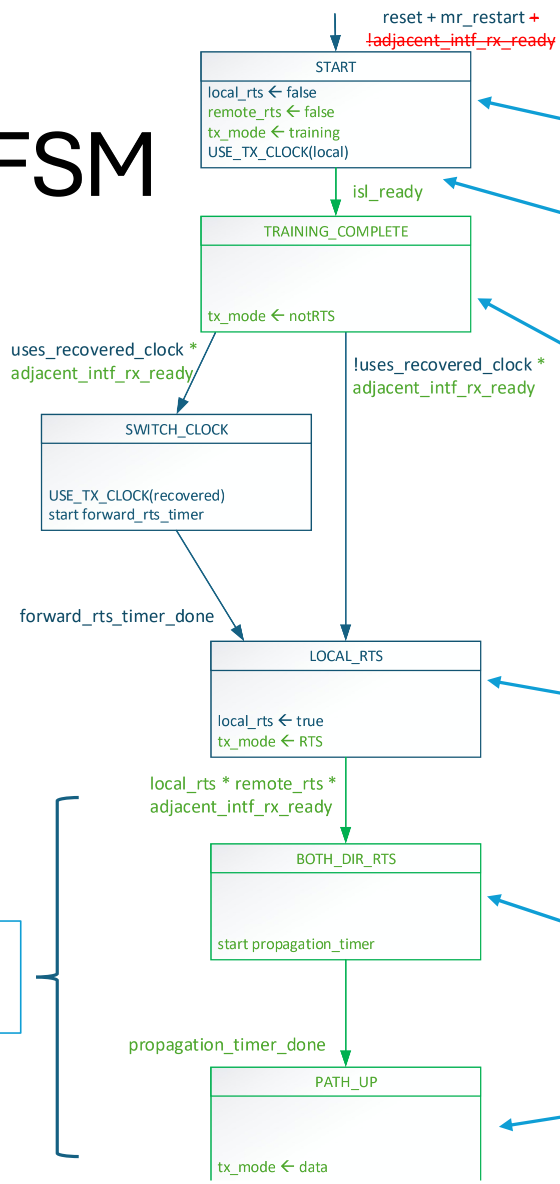
[1] training being complete is determined by the isl_ready variable, or by the lane_training_status for each lane being TRAINED
[2] mission clock is determined by the local_rts variable
[3] this mode is used by (some) ISLs when training is disabled or not supported

Proposed definitions

tx_mode value	Training complete?	Mission clock?	Mission data?
training	N	N	N
notRTS	Y	N	N
RTS	Y	Y	N
data	Y	Y	Y

The local pattern transmitted in notRTS and RTS modes is PMD/AUI-specific; some use training frames, some use other patterns

RTS FSM



Removed !adjacent_intf_rx_ready, as that should not cause a restart of the entire RTS process. Fault handling is in the context of this FSM is addressed in osorio_3dj_01_2605

Training (whatever that means for the ISL) happens in this state; when training is complete, isl_ready becomes true

isl_ready is set to true based on criteria that are specific to the PMD/AUI and what sort of training it does (this definition of isl_ready preserves the behavior of the blue states)

This new state means training is completed but the upstream adjacent interface is not yet RTS.

- new tx_mode = 'notRTS' means the interface sends whatever it sends when training is completed but it is not yet on mission clock or mission data. This can be specified in each PMD/AUI or in a table in Annex 178B. If the upstream adjacent interface is ready before training completes, one of the exit transitions is taken immediately after entering this state

This state means training is completed and the ISL has switched to mission clock

- new tx_mode = 'RTS' means the interface sends whatever it sends when training is completed and is on mission clock, but not mission data. This can be specified in each PMD/AUI or in a table in Annex 178B.
- If remote_rts becomes true before getting to this state, the exit transition will be taken immediately upon entering

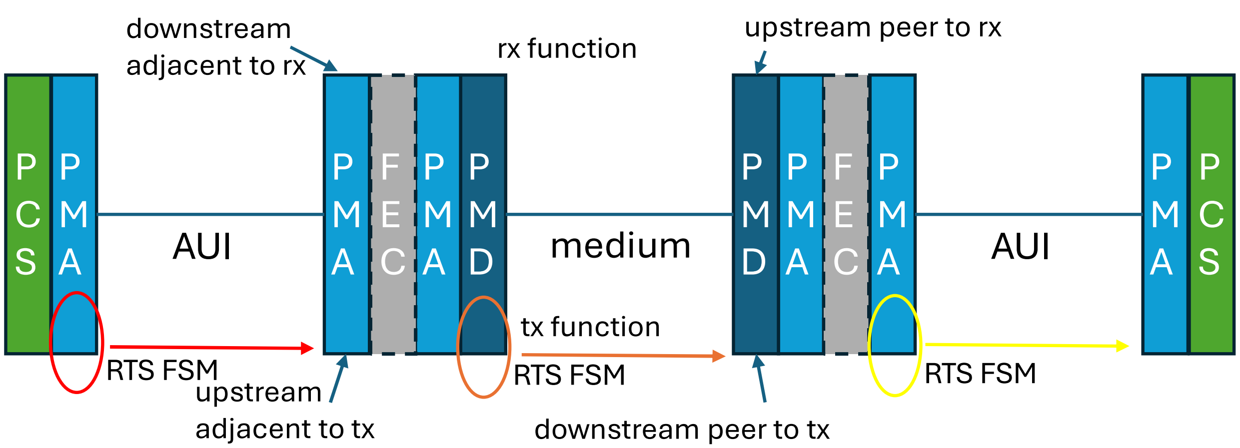
Moved from figure 178B-10 because these states are about RTS

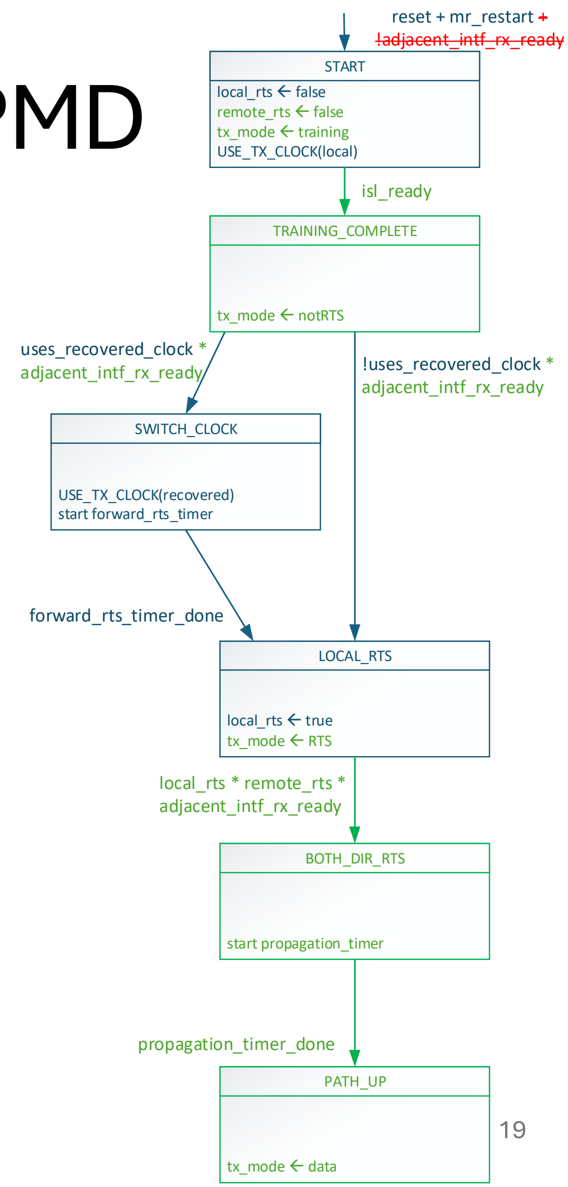
This is the "PATH_READY" state in current figure 178B-10

This is the "PATH_UP" state in current figure 178B-10

Worked example with an IMDD PMD Left-to-right

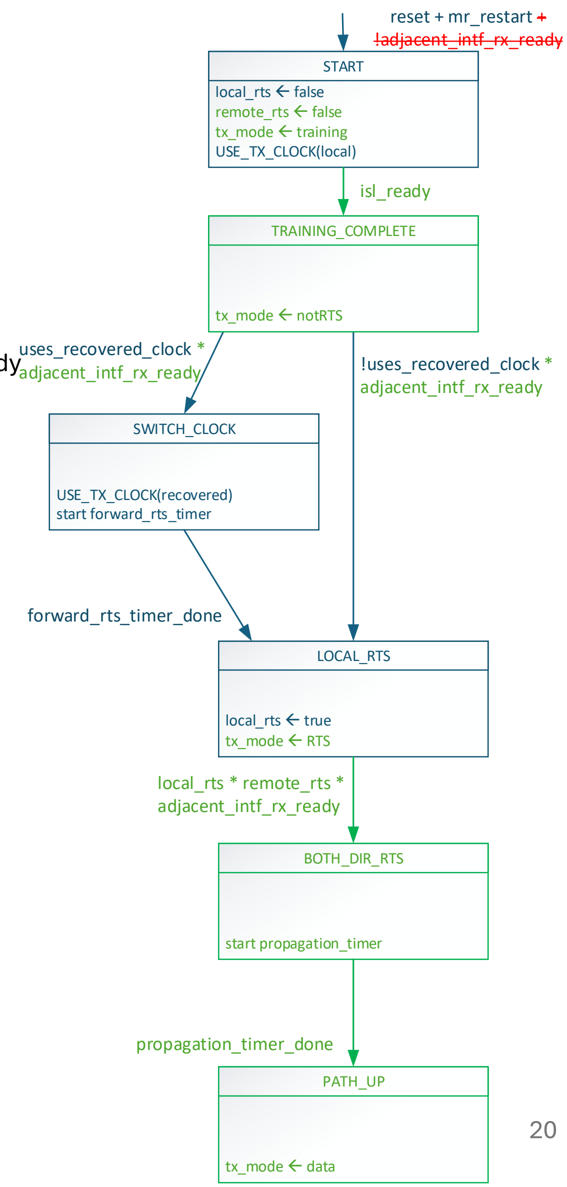
1. RTS FSMs begin in START. ILT runs independently on the AUIs and PMD. As each finishes training, it sets isl_ready to true, and the ILT FSM remains in the TRAINING_COMPLETE state
2. For the red FSM, adjacent_intf_rx_ready is always true. When training finishes, it moves through TRAINING_COMPLETE to LOCAL_RTS (it is always on mission clock) and signals RTS (Signal_OK = READY) across the AUI
3. For the orange FSM, adjacent_intf_rx_ready is false until the red FSM has signaled RTS, so it may wait in TRAINING_COMPLETE for that to happen and then progress through SWITCH_CLOCK to LOCAL_RTS, at which point it signals RTS across the medium.
4. For the yellow FSM, adjacent_intf_rx_ready is false until the orange FSM has signaled RTS, so it may wait in TRAINING_COMPLETE for that to happen and then progress through SWITCH_CLOCK to LOCAL_RTS.
5. As remote_rts becomes true due to events in the right-to-left direction, the RTS FSMs move to BOTH_DIR_RTS, start the propagation timer, and move to PATH_UP and mission data when the timer expires

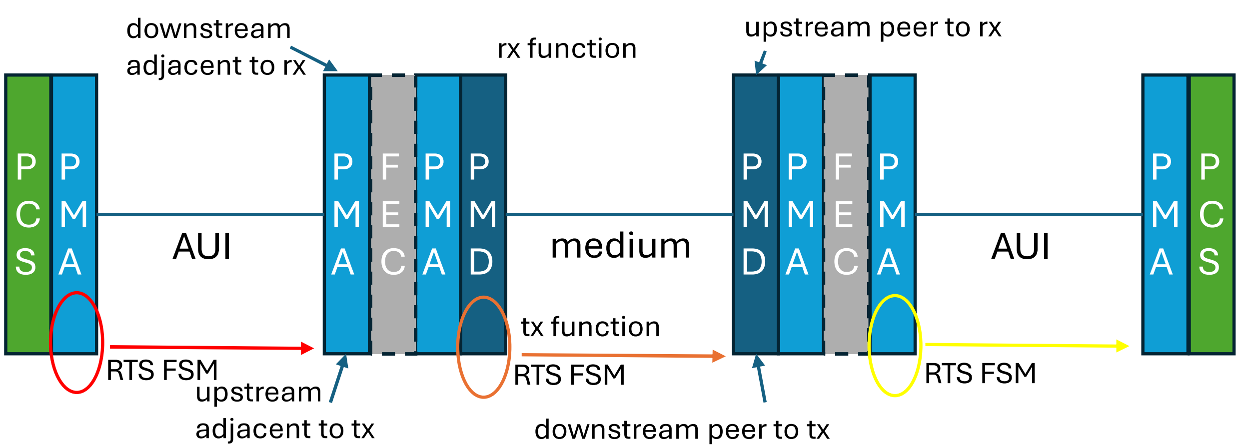




Worked example with PMD that doesn’t use ILT

1. RTS FSMs begin in START. ILT runs on the AUIs. As each finishes it sets isl_ready to true, and the ILT FSM remains in the TRAINING_COMPLETE state. Meanwhile, the PMD does whatever it does and sets isl_ready to true (e.g, for IMDD PMDD with training disabled, isl_ready is always true; for an ER1 PMD, isl_ready is true when the DSP frame, FEC frame, and ER1 tributary frame have been acquired).
2. Steps 2-5 are exactly the same as for the prior case – the RTS function is the same whether or not ISLs support ILT.





Key variable definitions (no change from current text)

- adjacent_intf_rx_ready: Boolean variable that is true when adjacent_signal_ok is either READY or OK and false otherwise. If there is no adjacent interface this variable is always true.
- adjacent_signal_ok: Enumerated variable derived from the value of the SIGNAL_OK parameter on the service interface. It takes one of the following values: IN_PROGRESS, READY, OK, FAIL.
- local_rx_ready: Boolean variable that is set to true when the receiver on a lane of the interface has determined that the peer interface transmitter is transmitting a PAM4 signal, that the remote transmit has been optimized if mr_training_enable is true, the local receive equalizers have been optimized, and that no further adjustments are required for normal data transmission.
- remote_rx_ready: Boolean variable that indicates the value of local_rx_ready on a lane of the peer interface. If mr_training_enable is true, it is derived from the receiver ready bit of the status field of received training frames on the corresponding lane of the interface. Otherwise it is set to true.

Key variable definitions that will need to change (1)

- isl_ready: Boolean variable that is set to true when training is complete for an interface. The details of what it means to train an interface are explained in the appropriate sublayer clause and/or summarized in table X in annex 178B ~~local_rx_ready and remote_rx_ready are true for all lanes of the interface, and to false otherwise~~.

PMD/AUI	When is isl_ready set
200G/lane IMDD PMD/AUIs with ILT	When every lane has completed ILT (set local_rx_ready and remote_rx_ready) per Annex 178B.7
200G/lane IMDD PMD/AUIs without ILT	Always true (enables the behavior of the blue states in current figure 178B-10)
800GBASE-LR1	Always true
800GBASE-ER1	When the PMA has finished frame alignment to the DSP frame and the FEC sublayer has aligned to the FEC frame and the ER1 tributary frame

Key variable definitions that will need to change (2)

- tx_mode will need to be revised to distinguish the different signals that are transmitted during the RTS process; it should support these values:
  - training (whatever that means for the ISL in question)
    - This could be training frames, or a local pattern, or something else entirely, but RTS doesn't need to know this detail; that belongs in an ILT FSM or PMD/AUI definition
  - not ready to send
    - This value is sent when training is complete, but adjacent intf is not RTS
  - ready to send
    - This value is sent when training is complete, adjacent intf is RTS, the interface has switched to mission clock, but peer is not yet RTS
  - data
- Definitions for local_rts and remote_rts will need to be revised to remove text related to the use of E1/O1 training frames since not all ISLs use those, but all ISLs can participate in RTS:
  - local_rts: Boolean variable that indicates that an interface is ready to send data. Its value is set by the RTS ~~update~~ state diagram (Figure 178B–9). ~~The logical-NOT of this variable is encoded as the continue training bit in the control field of transmitted training frames.~~
  - remote_rts: Boolean variable that indicates the value of local_rts variable in the peer interface. The details of how the local_rts status is signaled are explained in the relevant sublayer clause and summarized in table X.~~If mr_training_enable is true and the "continue training" bit of the control field of received training frames on all lanes of the interface is zero then remote_rts is set to true, otherwise it is set to false. If mr_training_enable is false then remote_rts is always true.~~

Comment #132

Cl **178B** *SC* **178B.5** *P* **868** *L* **8** # I-132

Huber, Thomas Nokia

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

Figure 178B-3 shows bidirectional communication between the RTS and ILT functions. This is presumably the result of the chocie to use the ILT training frames to signal RTS information on 200G IMDD PHYs - but that choice doesn't make the RTS signaling part of the ILT function. There is no need for the RTS function to communicate to the ILT function.

SuggestedRemedy

Change the arrow between ILT and RTS to be one-way from ILT to RTS.Further detail will be provided in a presentation

Proposed Response *Response Status* **O**

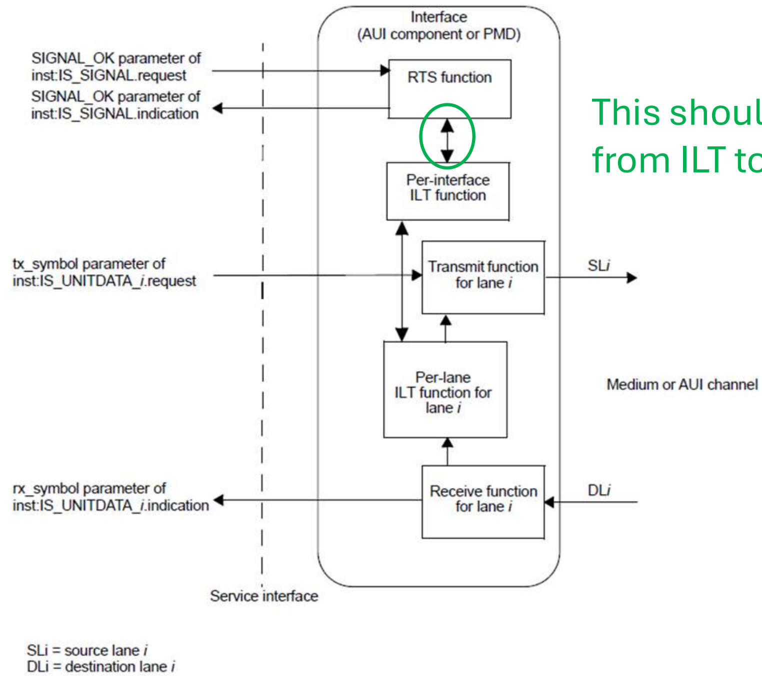


Figure 178B–3—RTS and ILT functions within interfaces

This should be one-way from ILT to RTS

With the refactored functions/FSMs:

- Per-interface ILT function is simply the aggregation of local_rx_ready and remote_rx_ready for each lane into isl_ready
- All other existing per-interface behavior is part of RTS

Comment #133

Cl **178B** *SC* **178B.5** *P* **869** *L* **14** # I-133

Huber, Thomas Nokia

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

Figure 178B-4 shows local_rts as a signal communicated from the RTS function to the ILT function and remote_rts as a signal communicated from the ILT function to the RTS function. Since local_rts cannot be true unless the ILT function is complete on every lane, the ILT function cannot possibly depend on local_rts. Remote_rts is unrelated to training the link, so it should not be coming through the ILT function, it should be a driect input to the RTS function. The confusion seems to be that the sa,e frame structure used for ILT is used to signal RTS for 200G IMDD AUIs/PMDs - but that doesn't make the RTS signaling part of the ILT function.

SuggestedRemedy

Revise figure 178B-4 to not show local_rts and remote_rts as signals between the two functions, but rather to show a 'training complete' signal from ILT to RTS. Further details will be provided in a presentation.

Proposed Response *Response Status* **O**

- The suggested 'training complete' is already shown as training_status; this could be changed to isl_ready
- Delete local_rts and remote_rts signals, as those are entirely contained in the RTS function

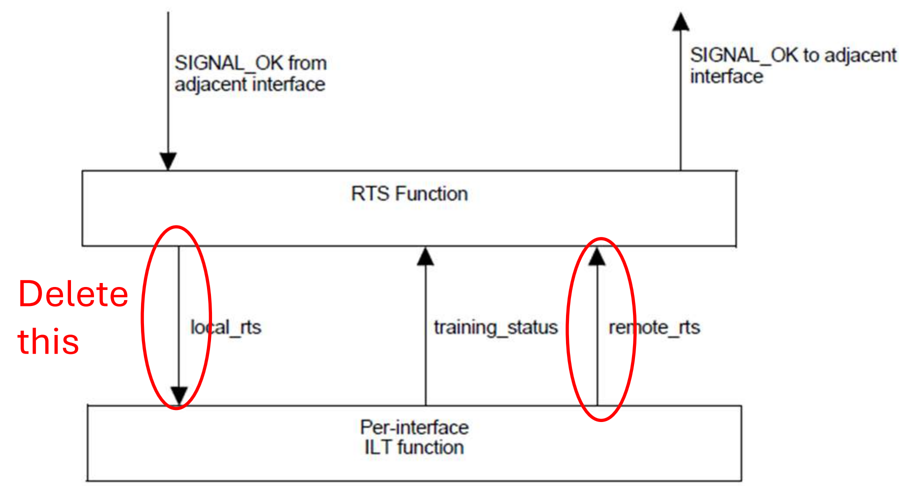


Figure 178B–4—Signaling between RTS and ILT functions

Comments #134-136

CI **178B** *SC* **178B.6** *P* **869** *L* **31** # I-134

Huber, Thomas Nokia

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

The operation of the RTS function should have no dependency on whether training is enabled (or even defined) for an ISL. Having such a dependency complicates extending APSU to PMDs that don't use ILT. The ISL needs to indicate to the RTS function that "training is complete" (whatever that means for the ISL in question).

SuggestedRemedy

Change "The operation of the RTS function is different based upon the state of mr_training_enable and whether the interface is implemented in a retimer (see 178B.9)."
to
"The operation of the RTS function is different based upon whether the interface is implemented in a retimer (see 178B.9)."
Further details will be provided in a presentation.

Proposed Response *Response Status* **O**

- Text proposals in these comments are consistent with the separation of ILT and RTS FSMs proposed in this presentation

CI **178B** *SC* **178B.6** *P* **869** *L* **46** # I-135

Huber, Thomas Nokia

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

RTS status signalling cannot provided by the ILT function, since some PMDs don't have that function. RTS status is signaled by a PMD- or AUI-specific mechanism (which in some cases is the same training frame that ILT uses, but that does not make this signaling part of the ILT function).

SuggestedRemedy

Change "The state of the local_rts variable is provided to the peer interface via the ILT function (Figure 178B–4)."
to
"The state of the local_rts variable is provided to the peer interface via a PMD- or AUI-specific mechanism."
Further details on the implications of this will be provided in a presentation.

Proposed Response *Response Status* **O**

CI **178B** *SC* **178B.7** *P* **870** *L* **4** # I-136

Huber, Thomas Nokia

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

RTS status signaling cannot be transferred by the ILT function, since some PMDs don't have that function. RTS status is signaled by a PMD- or AUI-specific mechanism (which in some cases is the same training frame that ILT uses, but that does not make this signaling part of the ILT function).

SuggestedRemedy

Change "The ILT function facilitates the establishment of communication between the peer interfaces of an ISL, including clock and data recovery and the transfer of the RTS status of the transmitter."
to
"The ILT function facilitates the establishment of communication between the peer interfaces of an ISL."
Further details on the implications of this will be provided in a presentation.

Proposed Response *Response Status* **O**

Comment #137

Cl **178B** *SC* **178B** *P* **870** *L* **9** *#* I-137

Huber, Thomas Nokia Minimize

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

The ILT function shouldn't have a DATA or a LOCAL PATTERN mode. This function is providing training on all the lanes of the PMD/AUI, not RTS signaling.

SuggestedRemedy

Remove DATA and LOCAL PATTERN modes from 178B.7. Further details will be provided in a presentation.

Proposed Response *Response Status* **O**

- The scope of the ILT function is training lanes of an ISL
  - It should reach a stable state when training is completed, and remain there unless training is re-initiated
  - As described in the presentation, tx_mode is a property of the RTS function, not the ILT function
- The proposed changes to the FSMs make the RTS FSM in charge of tx_mode
  - For an ISL that uses ILT, the ILT FSM would run on each lane when tx_mode = training
  - When training is completed for all lanes (indicated by isl_ready = true), the RTS FSM would set tx_mode to not-RTS or RTS based on the value of adjacent_intf_rx_ready at the time isl_ready becomes true

Comment #134-135 detailed text changes

178B.6 RTS function

The RTS function facilitates the indication of the local interface readiness to switch to DATA mode (local_rts) to other interfaces (peer and adjacent) and controls the switch over of the transmitter clock source to its DATA mode clock (when necessary). The RTS update state diagram (Figure 178B–9) and its associated variables define the process used for ~~the clock source transition~~ these transitions. The operation of the RTS function is different based upon ~~the state of mr_training_enable and~~ whether the interface is implemented in a retimer (see 178B.9).

An overview of the behavior of the RTS function is as follows:

— Initially, the local_rts variable is set false.

— Wait for the adjacent interface to be ready for DATA mode, as indicated by the SIGNAL_OK parameter from the adjacent interface being equal to READY or OK

— ~~If the ILT function is operating in TRAINING mode (mr_training_enable = true), w~~ Wait for local ISL to complete its adaptation, as indicated by ~~training_status equal to READY or OK~~ isl_ready equal to true

— If the interface is implemented in a retimer that uses the recovered clock in DATA mode, initiate the swap to the recovered clock from the adjacent interface

— Set the local_rts variable to true, indicating readiness to move to DATA mode The state of the local_rts variable is provided to the peer interface via ~~the ILT function (Figure 178B–4).~~ a PMD- or AUI-specific mechanism.

The state of the rts_status variable is provided by the interface to the adjacent interface via the service interface SIGNAL_OK parameter.

There is no specified timeout when waiting for either the local_rts or the remote_rts variables to change.

Comment #136-137 detailed text changes

178B.7 ILT function

The ILT function facilitates the establishment of communication between the peer interfaces of an ISL~~, including clock and data recovery and the transfer of the RTS status of the transmitter~~. The per-interface ILT function (see 178B.7.1) provides an aggregate status of all the per-lane ILT functions (see 178B.7.1) to the RTS function and indicates the readiness of the transmitter to send data (local_rts) ~~to the per-lane ILT functions~~.

~~When the ILT function is not in DATA mode, the transmitter ignores the input service interface IS_UNITDATA primitive and the ILT transmit function locally generates the data that is sent to its peer interface.~~

~~The ILT function has three modes of operation based on the tx_mode variable (see 178B.8.3.1):~~

~~— TRAINING mode (tx_mode = training): training frames are sent between peer interfaces~~

~~— LOCAL_PATTERN mode (tx_mode = local_pattern): a pattern is transmitted to the peer interface~~

~~— DATA mode (tx_mode = data): data is passed from the channel to the service interface and vice versa~~

~~TRAINING mode is used when training is enabled (mr_training_enable = true). This mode~~ The ILT function utilizes a training protocol, which enables the two peer interfaces to communicate with each other through transmission of fixed-length training frames (see 178B.7.3). The training protocol provides methods for each interface of an ISL to request changes of the peer interface transmitter state~~,~~ and provides status in response to requests received~~, and provides status of the local receiver state, and coordinates the transition to DATA mode~~.

~~LOCAL_PATTERN mode is used when training is disabled (mr_training_enable = false). This mode provides a method to indicate the readiness of the transmitter to transition to DATA mode to the peer interface using a locally generated pattern, as specified in the clause or annex that defines the interface.~~

~~DATA mode is entered when the ILT function has successfully completed its operations. If training is enabled (mr_training_enable = true), when transitioning from TRAINING mode to DATA mode the state of certain transmitter parameters that were derived by the training protocol are retained (e.g. precoder on/off).~~

The training control state diagram (Figure 178B–10) and its associated variables ~~are used by both TRAINING mode and LOCAL_PATTERN mode of operation. The~~ as well we thestate diagrams in Figure 178B–11 and Figure 178B–12 and their associated variables, apply only to PMDs/AUIs that support ILT when training is enabled (mr_training_enable = true).

Comment #137 related text changes 178B.7.2, page 870 line 50

The per-lane ILT function is responsible for the locally generated data that is transmitted while tx_mode = training ~~not in DATA mode~~ and for the adaptation processes of the received signal. Each lane of the interface runs an independent instance of the per-lane ILT function.

~~When in TRAINING mode the frame format specified in 178B.7.3 is used by the receiver to configure its peer interface transmitter (see 178B.7.4).~~ There are two formats of the training protocol frames defined, E1 and O1 (see 178B.7.4 and 178B.7.5). The clause or annex that defines the interface specifies which format is used. The transmitter parameters available for modification by the training protocol depends upon the interface type.

~~When in LOCAL_PATTERN mode the pattern transmitted is specified in the clause or annex that defines the interface. An interface indicates RTS to its peer interface by exiting the transmit disable state. The transmitter and receiver operate independently of each other. The transmitter parameters are configured through management control variables. A receiver adapts to the incoming signal regardless of the local transmitter state.~~

The per-lane ILT function is defined by 178B.7.3 through 178B.7.10 and the state diagrams in Figure 178B–10, Figure 178B–11, and Figure 178B–12.

Comment #140

Cl **178B** *SC* **178B.8** *P* **884** *L* **17** # I-140

Huber, Thomas Nokia

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

The state diagrams in figures 178B-9 and 178B-10 need to be refactored to separate states related to per-lane link training from states related to per-interface ready-to-send functions. This will simplify extension of APSU to PMD types other than 200G/lane IMDD.

SuggestedRemedy

A presentation with a complete proposal to revise the diagrams and supporting text (variable definitions, etc.) will be provided.

Proposed Response *Response Status* **O**

- The presentation has proposed new FSMs to replace the existing “RTS update” and “training control” FSMs
- Text related to the functional partitioning and the FSMs will need to be updated to reflect the modified FSMs

Summary

- The proposed modifications will separate ILT and RTS functions in a way that enables understanding one of them without having to understand the other
- This significantly simplifies extension of the RTS function to PMDs that do not support the ILT function
  - This includes not only the coherent PMDs in P802.3dj, but also future PMDs that may be defined by other task forces that may need different types of link training than what is provided by the E1/O1 frames defined in Annex 178B
  - It could also extend to legacy PMDs if a mechanism to set isl_ready can be implemented (e.g., using SIGNAL_OK to set isl_ready)
- This simplifies reuse of RTS by other SDOs for other applications
- While the proposed modifications may appear to be large in terms of the amount of text that would be affected, they do not change the behavior

Thank you!