

# Autonomous path startup with coherent PHYs (comment #219)

Adee Ran, Mike Sluyski, Doug Cattarusa (Cisco)  
Leon Bruckman (Nvidia)

# Supporters

# Overview

- Coherent PHYs currently do not have a training protocol over the medium and no ILT/RTS functions, thus no support of “path startup”.
- There seems to be consensus that these functions, as defined in Annex 178B, should be included in coherent PHYs too.
  - See straw polls in the [response to comment #418 against D2.0](#).
- As detailed in a previous presentation [ran\\_3dj\\_02\\_2507](#), the state diagrams in Annex 178B enable path startup either with or without a training protocol on each ISL.
  - If **mr\_training\_enable=false**, a training protocol is not used and local\_rts, the information required for path start-up, is indicated by squelching or transmission of a local pattern.
  - The proposal in [ran\\_3dj\\_03a\\_2507](#) was to use this method in coherent PHYs... but some drawbacks were identified.
  - An alternative is to **define a “training” protocol** suitable for coherent signaling.
- This presentation includes a proposal for **an ILT function with a new training frame format for coherent PHYs**, which will enable autonomous path startup (as defined in 178B.4) for paths that include these PHYs.

# Background: path startup without training

(not specific to coherent; not part of this proposal)

The following method can be used by a module to interoperate with link partners that do not implement a training protocol on the media side (“legacy”).

- On the media-side interface, set the ILT variable `mr_training_enable` to false; this causes the left-hand part of the diagram to be used.
- In this part of the diagram, QUIET state is maintained when `local_rts`=false (which means host interface training is not completed), and the transmitter is disabled.
  - In CMIS terms, the module delays activation of the transmitter (media side) until the host side is fully activated.
  - From the link partner’s point of view, there is no signal.
- When `local_rts` becomes true (host interface ready), the state is changed to SEND\_LOCAL and the transmitter is enabled, **with a locally generated pattern** instead of the training frames received from the host interface.
  - In CMIS terms, it is in the DPActivated state but with output from a pattern generator.
  - The link partner now sees a signal that it can lock on.
- When the receiver is locked (`local_rx_ready`), the state becomes PATH\_READY for a period of propagation\_timer (transient state).
  - The link partner signal must be valid because it is “legacy”.
- The final steady state is PATH\_UP where the transmitter changes to DATA mode (`tx_mode`=data), transmits the data from the host interface to the media and vice versa.
- This method is essentially the same as existing “legacy” behavior...
  - But it is a valid “ILT behavior” that enables path start-up.
  - The limitation is that host interface training must be completed before transmitter activation.

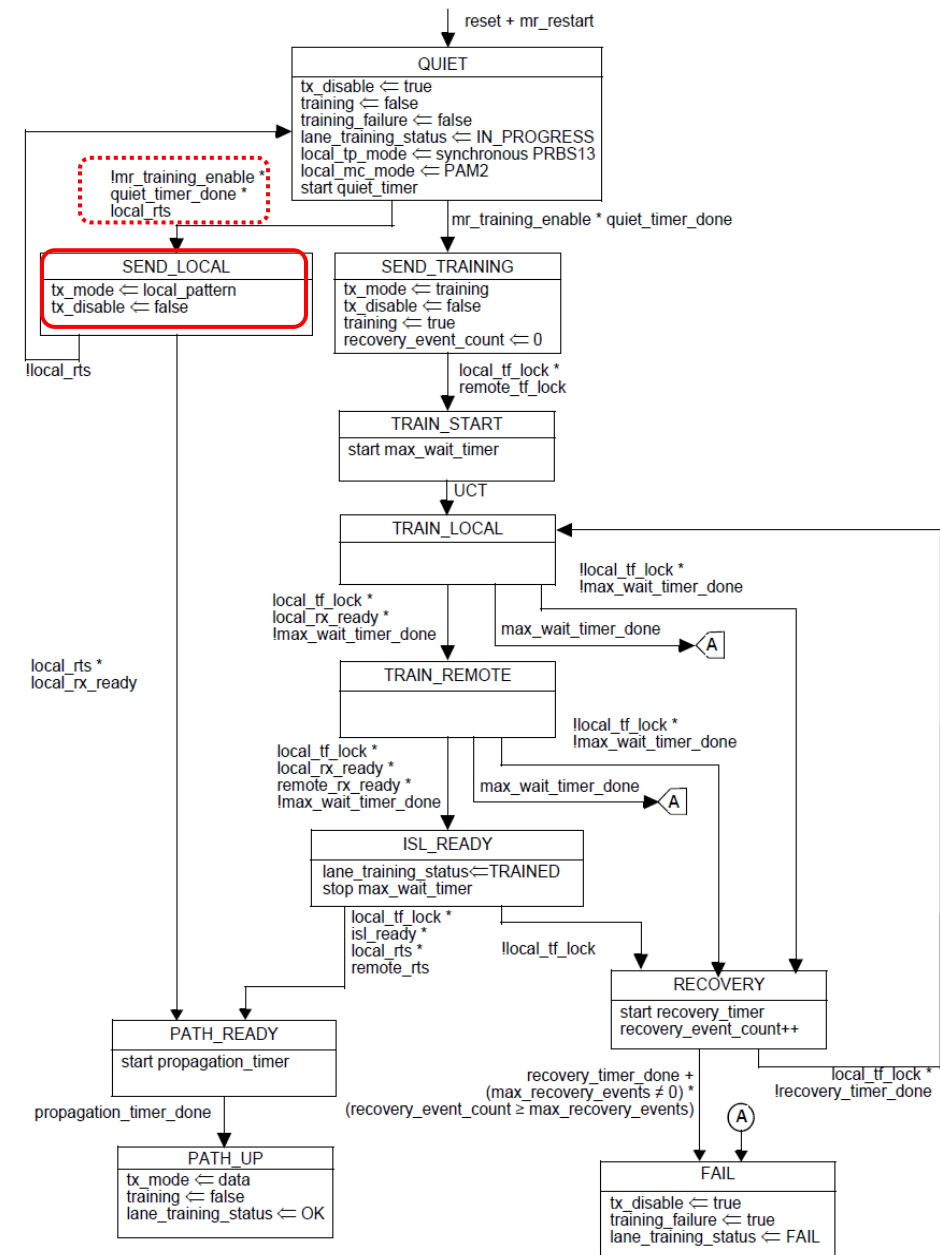


Figure 178B-10—Training control state diagram  
As of D2.2; Note that there are comments suggesting changes to this diagram

# Why should coherent PHYs use a training protocol?

- Autonomous path startup requires communicating RTS across each ISL.
- There are advantages to using a coherent-modulated signal instead of communicating RTS via squelch.
  - Enable training on the host interface independently of the state of the media interface (coherent links may be enabled prior to activation of their endpoints).
  - Support multiplexing applications (beyond the scope of 802.3dj).
  - Avoid squelching a coherent transmitter when the host interface aborts training (not good!)
- The training protocol needs to communicate at least the RTS indications (**local\_rts**/**remote\_rts** variables, one bit in each direction).
  - The additional **rx\_ready** and **tf\_lock** variables (2 bits in each direction) are used to coordinate the start and end of the “training” states where requests are exchanged (e.g. training pattern, modulation, and precoding).
  - If coherent PHYs don’t need to exchange any requests, then these variables are not required; but it is preferable to enable exchange of requests as a future extension.

# Architectural considerations for coherent PHYs

- The architecture of the coherent PHYs (clauses 185 and 187) is different from electrical and IM-DD PHYs
  - Encoding of the traffic, including pilot sequences and data striping across the coherent data streams, happens in a FEC sublayer above the PMD (clauses 184 and 186 respectively).
  - PMD:IS\_UNITDATA.indication are “analog” signals that include the pilot sequences and other framing fields created by the upper layer.
  - Pilot sequences must always be present for receiver operation. This includes TRAINING mode too.
  - The rx\_ready variable of Annex 178B (which are mapped to the SIGNAL\_OK service interface parameter) should be generated by the sublayer that includes the ILT function.
- Consequently, it might make sense that in these PHYs the ILT function reside in the FEC sublayer rather than in the PMD.
  - More on that later...
- We assume the “training” protocol is not used for actual receiver training.
  - Most of the ILT-controlled functions (equalization, precoding, PAM2/PAM4 selection, choice of test patterns, polarity correction) are not required.

# TRAINING mode signaling for coherent PHYs

The following options were considered:

- A. Map **local\_rts** to a bit in the DSP frame (clause 187) or inner FEC pilot sequence (clause 185) while the payload is an “encoded PRBS31” pattern. Recover **remote\_rts** from the received DSP frames.
  - B. XOR **local\_rts** with the PRBS31 before encoding. The receiver recovers **remote\_rts** by detecting the polarity of the PRBS31 in the payload.
  - C. Send Annex 176D training frames (or their bit stream equivalent) as the payload. Recover **remote\_rts** from the received payload.
- For 802.3dj purposes, any of the methods above would work, and the implementation cost is negligible.
  - **Option A** will likely have the minimal impact on design, since overhead bits may be controllable even in existing designs
    - However, it assumes the coherent link carries **only one RTS** in each direction.
    - This prevents use cases where a coherent link carries multiple logical links (aka “multiplexing”), defined outside of 802.3.
  - **Option B** can support multiple logical links, but is limited to one bit per logical link – would prevent future extension.
  - **Option C** looks like an overkill, but has some benefits:
    - “training frame lock” and “receiver ready” can be useful debugging information
    - Common specification and implementation in both coherent PHYs (LR and ER)
    - Re-use of most of the specifications in annex 176B
    - Beyond the scope of 802.3, links from multiple hosts can be multiplexed, and each one keeps its own status bits.
  - **Option C is described in detail in the subsequent slides.**

# Proposed technical changes in Annex 178B

- Define a new training format O2, which is based on the existing format O1 with the following exceptions:
    - The training frame is a PRBS31 pattern occasionally overridden by Marker+Control field+Status field (DME encoded).
    - Bits 6:5 of the control field (“Training pattern request”) are reserved, always sent/received as 11 (“free-running PRBS31”); this matches the training frame encoding.
    - Bits 9:8 (“Modulation and precoding request”) are reserved, always sent/received as 00 (“PAM2”). There is no requirement to change these bits for completing the protocol.
  - Make the necessary changes across the annex to include this format.
    - O2 is like O1, but in addition “Training pattern setting” (176B.7.9) does not apply.
    - The variables **local\_mc\_mode**, **remote\_mc\_mode**, **local\_tp\_mode**, and **remote\_tp\_mode** have fixed values and are not considered in the conditions for exiting the training protocol.
- The coherent FEC sublayer uses the training frames generated by the ILT logic (mapped to a bit stream) as a **payload into the FEC encoder**.
  - A receiver can feed the received payload (after FEC decoding) into the ILT logic, which will identify the marker and decode the DME content.
  - The Annex 178B state diagrams, variables, etc. stay mostly intact.



# Where should the ILT/RTS function reside in coherent PHYs?

- Option 1: In the FEC sublayer of 800GBASE-LR1 (C184) and 800GBASE-ER1 (C186)

Pros	<ul style="list-style-type: none"><li>• Annex 178B functionality requires digital implementation; the digital functionality in these PHYs is in the FEC sublayers</li></ul>
Cons	<ul style="list-style-type: none"><li>• Annex 178B is written for PMDs and AUIs; these PHYs will require several exceptions (next slide)</li></ul>

- Option 2: In the PMD sublayer of 800GBASE-LR1 (C185) and 800GBASE-ER1 (C187)

Pros	<ul style="list-style-type: none"><li>• It is consistent with the location of the ILT/RTS function in all other PHYs</li><li>• Fewer changes required in Annex 178B – more consistent specifications</li></ul>
Cons	<ul style="list-style-type: none"><li>• The PMD will need to include digital functions (generation and decoding of frames) which currently exist in upper sublayers – duplication</li><li>• Does not map nicely to implementations</li></ul>

- An important functional difference between the options: in Option 1 the LT frames are protected by the (Inner) FEC, while in Option 2 they are not.
- Option 1 is described in detail below. See backup for Option 2.

# Examples of exceptions needed in Annex 178B

- Change the definition of “interface”:
  - Define a new term: Coherent interface
  - A coherent interface is either 800GBASE-LR1 or 800GBASE-ER1
- Change “AUI component or PMD” to “AUI component, PMD or coherent interface” in 6 places
- Modify architectural figures 178B-1 and 178B-2 or add new figures that include examples of ISLs that include coherent interfaces
- Update Figure 178B-3 to accommodate a coherent interface

(Note: the editors may find other ways to achieve the same result)

# Proposed changes in clause 184 and 185

- Add Annex 178B to the Physical Layer clauses table (Table 185–1).
- In 184.3 (service interface), update the service interface to include a FEC:IS\_SIGNAL.request primitive to convey the RTS indication (based on the definition of this primitive in 176.3).
- In 184.4 (transmit function), add the two modes of the transmit function (TRAINING and DATA). In DATA mode the transmit function works as defined in D2.2. In TRAINING mode it uses the output of the ILT function instead of the tx\_symbol stream from the service interface, and operates as follows:
  - If mr\_training\_enable is true (default):
    - When tx\_disable is false, training frame format O2 is used. The training frames are converted to a bit stream by mapping the PAM2 symbols 0 and 3 (as defined in Annex 178B) to the binary values 0 and 1, and then encoded similar to the PRBS31 test pattern generator (184.6.1): 10-bit blocks from the training frames are round-robin distributed to the 32 pcsla Inner FEC flows.
    - When tx\_disable is true, instead of turning the transmitter off, the PRBS31 test pattern (as defined in 184.6.1) is transmitted without the marker+DME (to cause ILT in the partner to lose frame lock).
  - If mr\_training\_enable is false (optional, e.g. for “legacy” support):
    - When tx\_disable is false, the PRBS31 test pattern (as defined in 184.6.1) is transmitted.
    - tx\_disable=true is identical to the PMD global transmit disable.
- Add an ILT subclause in clause 184 (e.g. after 184.5, or under a new hierarchy “functional specifications” which would include 184.4 and 184.5).
  - Content based on 180.5.12 except that the training frames use O2 format and local\_pattern transmits the PRBS31 test pattern (as defined in 184.6.1).
  - Polarity detection and correction are not defined.

# Functional block diagram changes in Clause 184

Similar changes will be required in clause 186 (e.g. figure 186-3)

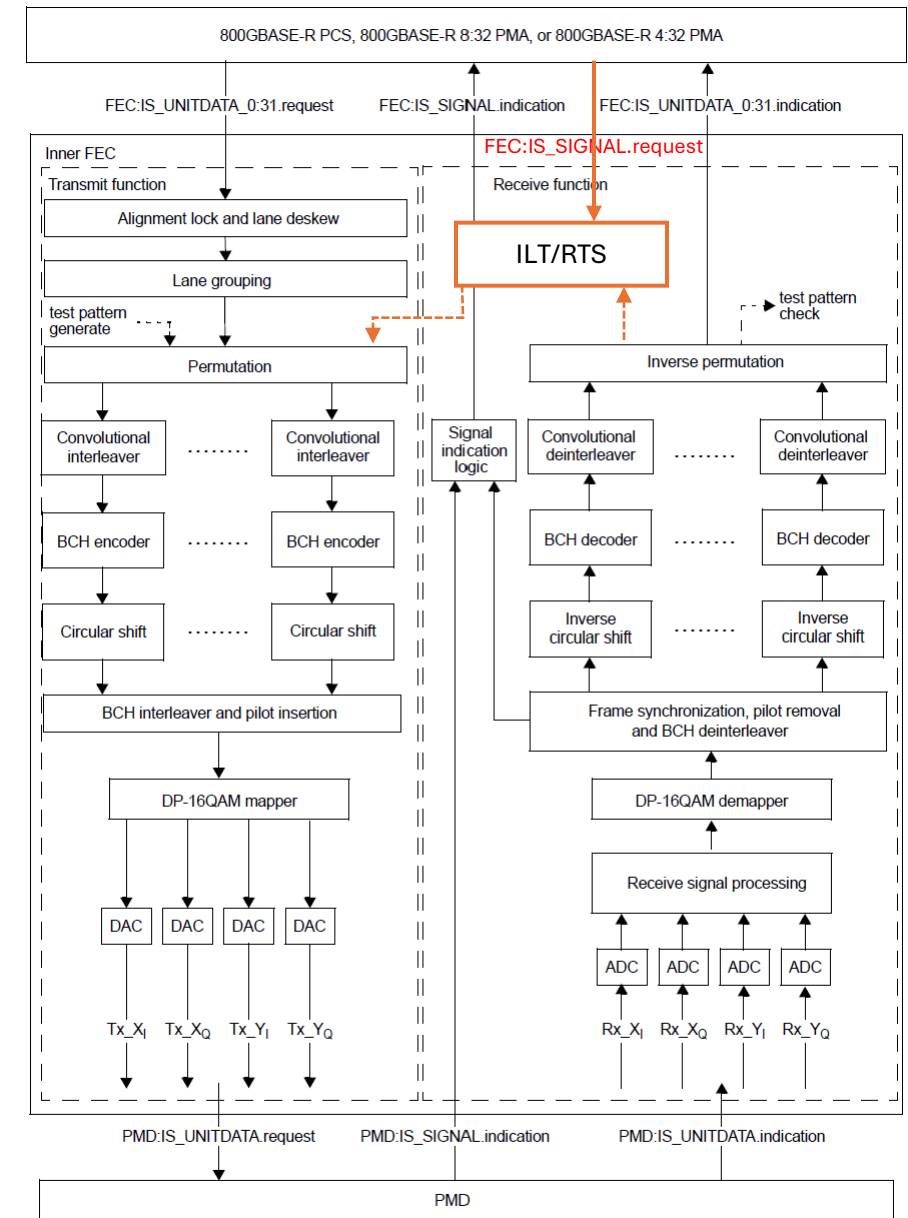


Figure 184-2—Inner FEC functional block diagram

# Proposed changes in clause 186 and 187

- Add Annex 178B to the Physical Layer clauses table (Table 187–1).
- In 186.2.2 (FEC service interface), update the service interface to include a FEC:IS\_SIGNAL.request primitive to convey the RTS indication (based on the definition of this primitive in 176.3).
- In 186.2.3 (FEC transmit function), add the two modes of the transmit function (TRAINING and DATA). DATA mode is as defined in D2.2. In TRAINING mode it uses the input and output of the ILT function instead of the client (PMA) data, and operates as follows :
  - If mr\_training\_enable is true (default):
    - When tx\_disable is false, training frame format O2 is used. The training frames are converted to a bit stream by mapping the PAM2 symbols 0 and 3 (as defined in Annex 178B) to the binary values 0 and 1, and then encoded similar to the PRBS31 test pattern generator in 186.2.3.12: mapped into each of the 800GBASE-ER1 tributary frames.
    - When tx\_disable is true, instead of turning off the transmitter, the PRBS31 test pattern (as defined in 186.2.3.12) is transmitted without the marker+DME (to cause ILT in the partner to lose frame lock).
  - If mr\_training\_enable is false (optional , e.g. for “legacy” support):
    - When tx\_disable is false, the PRBS31 test pattern (as defined in 186.2.3.12) is transmitted.
    - tx\_disable=true is identical to the PMD global transmit disable.
- Add an ILT subclause in clause 186 (e.g. as new subclause 186.2.5, or under a new hierarchy “functional specifications”).
  - Content based on 180.5.12 (with any modifications due to comments), except that the training frames use O2 format and local\_pattern transmits the PRBS31 test pattern (as defined in 186.2.3.12).
  - Polarity detection and correction are not defined.

# ILT training frame structure (Annex 178B)

- The payload consists of consecutively repeating training frames, as shown on the right.
- The control field and status field consist of cells that represent one bit each, using Differential Manchester Encoding (DME).
  - See next slides for bit assignments.
- The training pattern for coherent links is proposed to be free-running PRBS31.

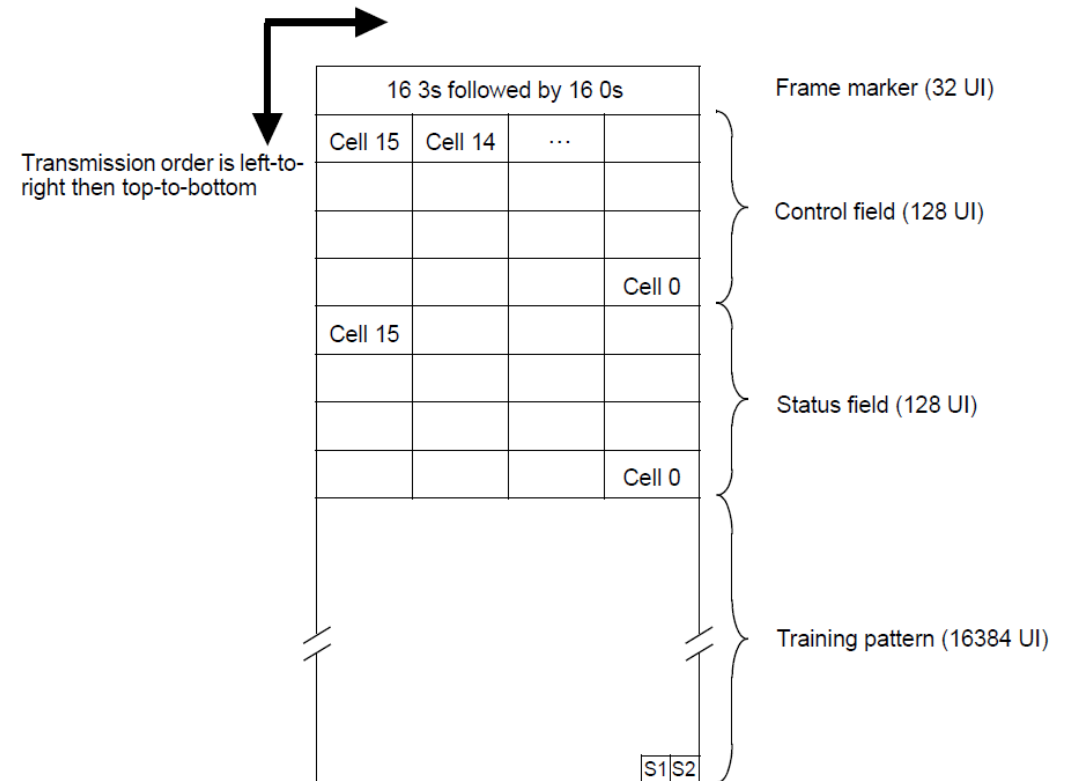


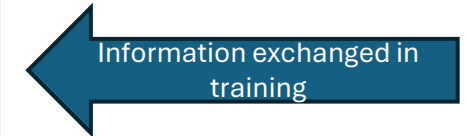
Figure 178B-5—Training frame structure

(No change proposed to this figure)

# Control field structure

Table 178B-3—Control field structure for the O2 format


Bit(s)	Name	Description
15:11	Reserved	Transmit as 0, ignore on receipt
10	Continue training	1 = Continue training 0 = Switch to data when training is completed
9:8	Reserved	Transmit as 00, ignore on receipt
7	Reserved	Transmit as 0, ignore on receipt
6:5	Reserved	Transmit as 11, ignore on receipt
4:0	Reserved	Transmit as 0, ignore on receipt




# Status field structure

Table 178B-5—Status field structure for the O2 format

Bit(s)	Name	Description
15	Receiver ready	1 = Training is complete and the receiver is ready for data 0 = Request for training to continue
14	Reserved	Transmit as 0, ignore on receipt
13:12	Reserved	Transmit as 11, ignore on receipt
11:10	Reserved	Transmit as 00, ignore on receipt
9	Receiver frame lock	1 = Frame boundaries identified 0 = Frame boundaries not identified
8	Reserved	Transmit as 0, ignore on receipt
7	Parity	Even parity bit
6:0	Reserved	Transmit as 0, ignore on receipt



Information exchanged in training



Information exchanged in training



# Editorial implementation (if adopted)

- The proposal requires several changes in coherent clauses.
  - With option 1, the additional content is mostly in clauses 184 and 186. Clauses 185 and 187 need minor changes (subclause tables).
- Changes in Annex 178B are mostly listed above.
  - Some additional exceptions may be required with option 1.
- The intent of this presentation is to present the proposed solution at a relatively high level, similar to a baseline proposal.
  - Listing the full edits would make this a much longer presentation.
  - It is suggested to **give the clause editors broad license, as usually done in baseline proposals**, to cover any implications not listed explicitly.

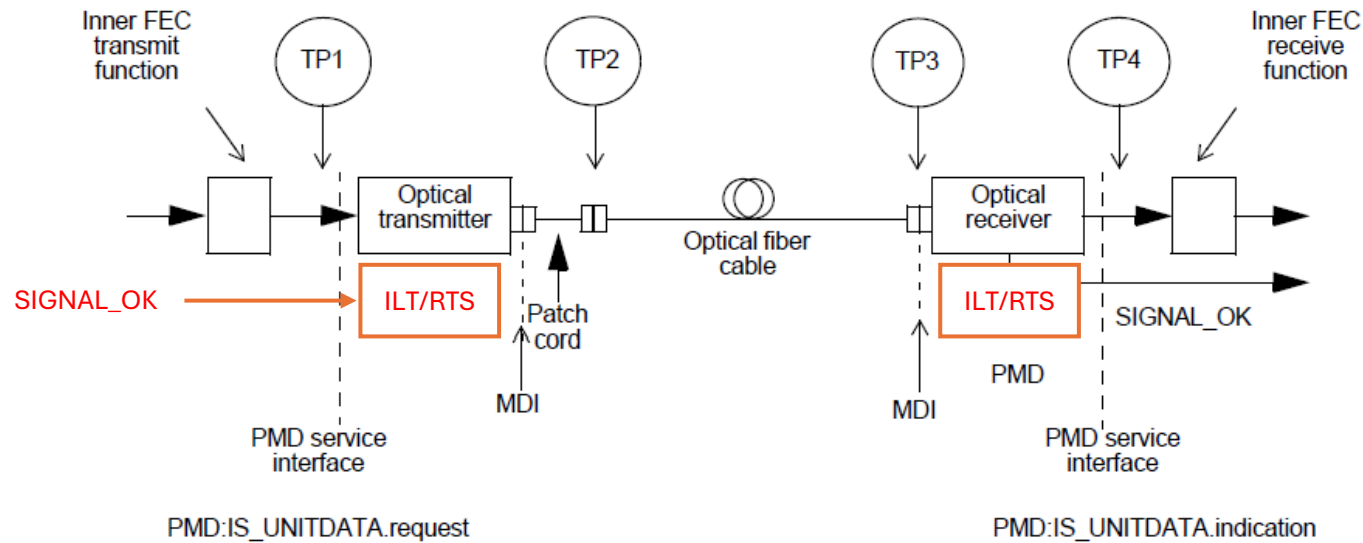
# Backup

# Editorial implications of the alternatives for communicating RTS – options A and B

(which seem less preferable)

- Option A is to use any reserved fields in the coherent DSP frame (or carrier frame) to communicate ILT variables.
  - The mapping between ILT variables and coherent frame bits should be defined.
  - The receiver and transmitter should use the same bits.
  - This method is straightforward in 800GBASE-ER1, but may be less feasible in 800GBASE-LR1, which uses a more compact DSP frame (see screenshot on the next slide). A different mapping should be used (e.g. to part of the pilot sequence).
- Option B uses the transmitted bit sequence (payload) instead, and communicates one bit (local\_RTS) by XORing it with the PRBS31 generator output. The result is transmitted as a payload.
  - The receiver detects whether its input matches PRBS31 or logically inverted PRBS31 and thus decodes the bit as remote\_RTS.
  - This method works equally in 800GBASE-ER1 and 800GBASE-LR1 but only communicates one bit.
  - Note that it is possible to expand this method to transmit multiple bits of information by using the PRBS31 generator as a scrambler. However, this will have comparable complexity to option C.
- These two alternatives do not use most of the definitions of Annex 178B, so will require many exceptions to be added.

# PMD clause changes assuming option 2 (ILT function resides in the PMD)



For clarity, only one direction of transmission is shown

**Figure 185-3—Block diagram for 800GBASE-LR1 transmit/receive paths**

Note: The ILT/RTS function includes also the generation/termination of the DSP frame (Pilots).

# PMD clause changes assuming option 2

- Add ILT to the Physical Layer clauses (Table 185–1).
- Add to the PMD service interface section:
  - The SIGNAL\_OK parameter of the PMD:IS\_SIGNAL.indication primitive corresponds to the variable training\_status of the ILT and RTS functions as defined in 178B.8.2.1. When SIGNAL\_OK is either IN\_PROGRESS or FAIL, the rx\_symbol parameters of PMD:IS\_UNITDATA\_i.indication on all lanes are unspecified.
- Add to the PMD transmit function section:
  - The PMD transmit function has two operating modes: DATA and TRAINING. The operating mode is controlled by the tx\_mode variable of the ILT function: it is DATA when tx\_mode = data, and TRAINING otherwise.
  - When operating in DATA mode, the PMD Transmit function shall convert the n symbol streams requested by the PMD service interface messages PMD:IS\_UNITDATA\_0.request to PMD:IS\_UNITDATA\_n–1.request into an optical signal. The optical signal shall then be delivered to the MDI, according to the transmit optical specifications in this clause.
  - When operating in TRAINING mode, the DP-16QAM symbol stream on each lane is taken from the output of the training pattern generator in the ISL training function (see 178B.7 and Figure 178B–6).

# PMD clause changes assuming option 2

- Add a new section: PSU functions
  - A PMD shall provide the ILT function for a Type O1 interface, specified in Annex 178B (see 178B.7.4 ) with the Modulation and precoding filed set to PAM2 (00) and Training pattern request set to free-running PRBS31 (11)
  - When the variable mr\_training\_enable is true, the ILT function is used to request changes to the peer transmitter state (modulation, training pattern, and precoder state), indicate the receiver state, and coordinate the transition to DATA mode.
  - When mr\_training\_enable is false and tx\_mode is local\_pattern (see 178B.8.3.1) the PMD transmits PRBS31Q (see 176.7.4.2).
  - In the transmit path the ILT/RTS functions shall implement the DSP frame generation (see 184.4.10), the DP-16QAM mapper and Symbol mapping to analog signals (see 184.4.11) functions. In the receive path the ILT/RTS function shall implement the DP-16QAM demapper (see 184.5.3) and DSP frame synchronization and pilot removal (see 184.5.4) functions. The ILT frames are mapped to the payload area of the DSP frames.
  - The default value of max\_wait\_timer\_duration is 60, corresponding to a duration of 60 seconds for max\_wait\_timer (see 178B.8.3.3).
- Add ILT/RTS variables to the MDIO