Enabling coherent PMDs to participate in 802.3dj APSU

Gary Nicholl, Cisco
Matt Brown, Qualcomm
Jeff Slavick, Broadcom
Luz Osorio, Nokia
Tom Huber, Nokia
Marco Mascitto, Nokia
Sebastien Gareau, Ciena
Guangcan Mi, Huawei
Xiang He, Huawei
Leon Bruckman, NVIDIA
Adee Ran, Cisco
Jeff Maki, HPE

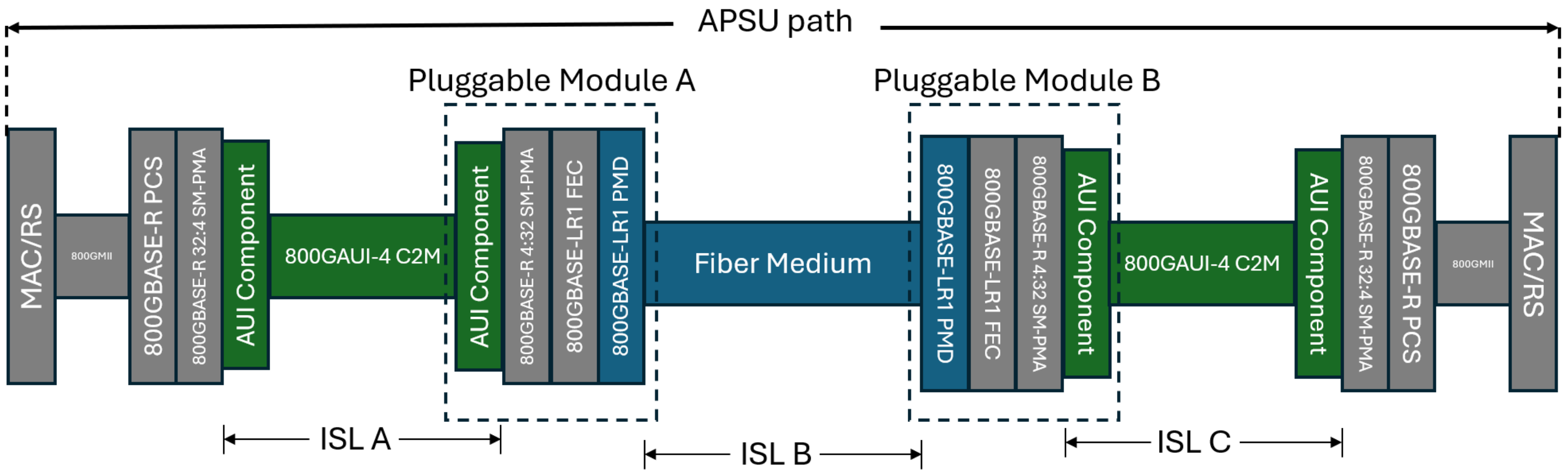
IEEE 802.3dj Plenary, March 2026

Background

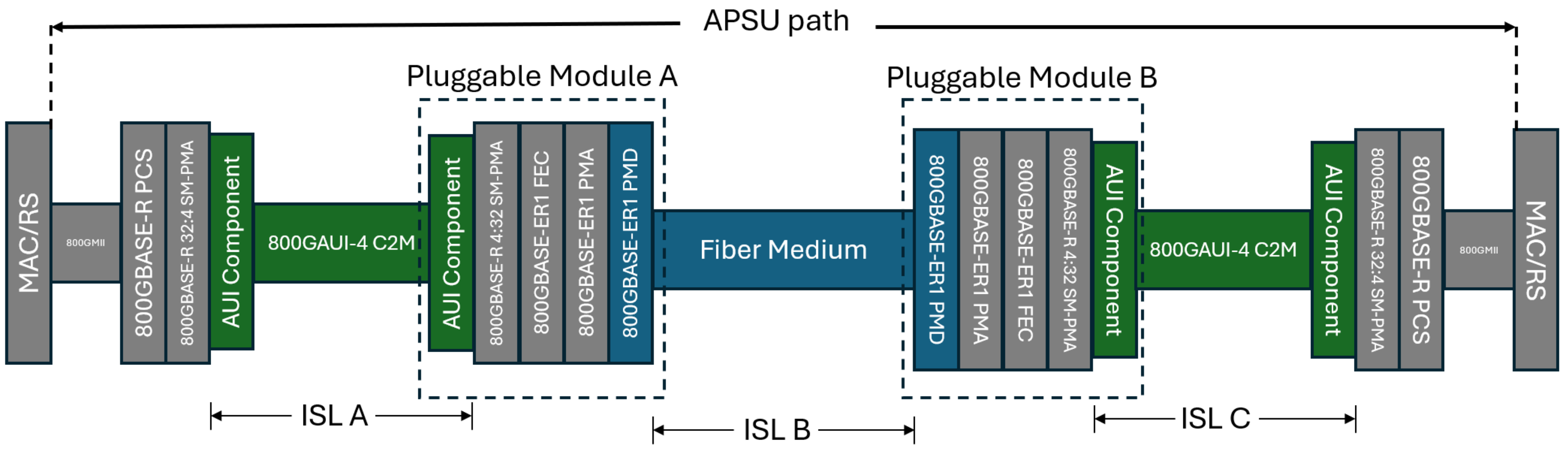
- There is a requirement to create paths that contain both 200G/lane AUIs and coherent interfaces.
- Annex 178B APSU is designed to minimize the amount of human intervention needed to initialize a path, which preserves the plug-and-play nature of Ethernet.
- The 800GBASE-LR1 and 800GBASE-ER/ER1-20 PMDs currently do not participate in APSU.
  - These coherent PMDs do not “train” in the classical way IMDD optical interfaces perform training using O1 training frames.
  - Sending O1 frames on the line interface without pilot sequences is not a valid coherent signal.
- There are currently no provisions for these coherent PMDs to support the RTS function in P802.3dj D3.0.
- OIF 1.6T interfaces could also benefit from a solution to support the RTS function, as they make use of 200G/lane AUIs on the client interface.
  - RTS is slightly more complicated when you consider 1600ZR+ interface, as it has multiple multiplexed clients
    - The optical interface also needs to be in mission mode prior to the training of any client interface, as clients may be activated at any time throughout the life of the path

The issue ....

800GBASE-LR1



800GBASE-ER1/ER1-20



ISL B currently presents an issue because there is no way to propagate RTS from ISL A to ISL C (and vice versa)

Goals and assumptions

- Select mechanisms for coherent interfaces to signal RTS.
- The goal is to find solutions for the following interfaces as part of this effort:
  - IEEE 802.3: 800GBASE-LR1 and 800GBASE-ER1/ER1-20
  - OIF IAs: 1600ZR, 1600ZR+ and 1600CL

  as LR1 and CL share common attributes, and ZR, ZR+ and ER1 share common attributes.
- Do not change the model for RTS propagation in 178B.
  - Keep the training independent from RTS propagation.
- As much as possible, leverage the mechanisms currently found in P802.3dj D3.0 to perform the signaling of RTS.
  - We cannot use squelch to signal RTS for coherent interfaces
  - We cannot use O1 because it is not a valid coherent signal
- A common solution for LR1 and ER1 is not required but can be considered depending on the trade-offs.

Options for IEEE 800GBASE-LR1 (and OIF 1600CL)

The options presented in this section apply to both LR1 and CL

OPTION LR1-A

Inner FEC encoded PRBS31

- Indicate local_rts through selection of a locally generated pattern that is mapped into the LR1 inner FEC:
  - Transmit inverse-PRBS31 local pattern when !local_rts
  - Transmit PRBS31 local pattern when local_rts
- It may be possible to use the LOCAL_PATTERN mode of the ILT function to implement this option.

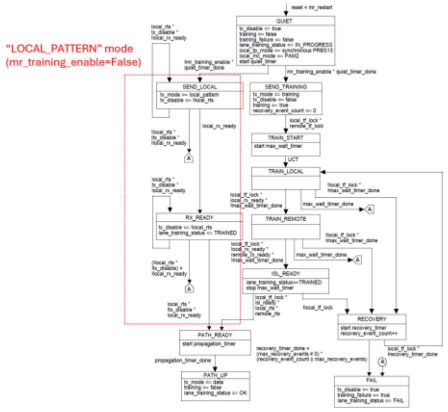


Figure 178B–10—Training control state diagram

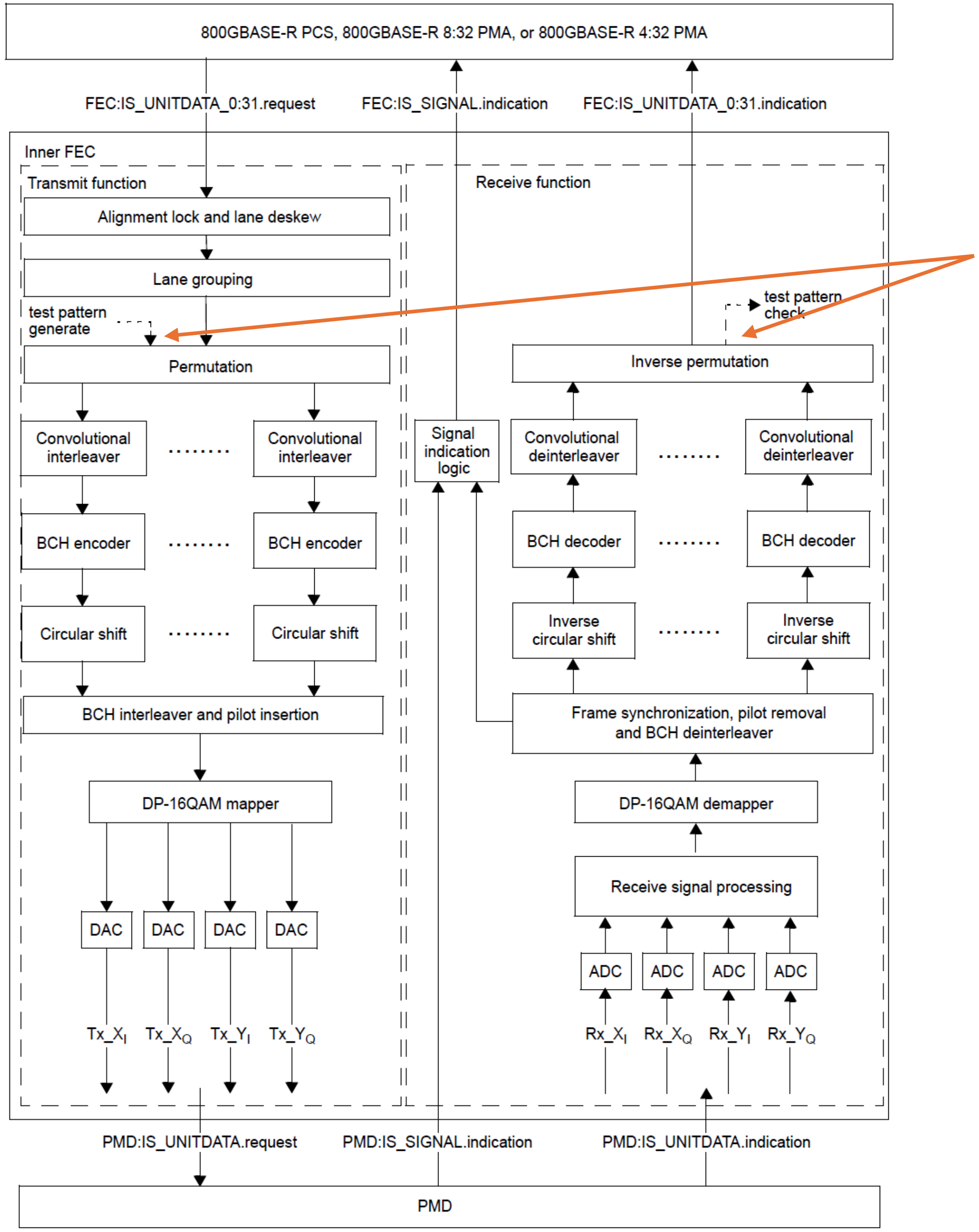


Figure 184–2—Inner FEC functional block diagram

PRBS31 pattern insertion point and detection point Cl184 Inner FEC

- Could be based on the test pattern generation and check points currently in P802.3dj D3.0 Cl184.
- Change would mean the ability to generate and detect an inverse-PRBS31 pattern, in addition to the currently defined PRBS31 pattern.

Define a new O2 training frame

- Presented on Nov 2025: https://www.ieee802.org/3/dj/public/25_11/ran_3dj_03_2511.pdf
- Define a new O2 training frame format. The O2 frame format provides only status signals, with no ability to change the transmitter pattern, modulation or equalization.
  - For example, O2 training frame structure contains only the control and status fields Parity, Receiver Ready (local_rx_ready), Receiver frame lock (local_tf_lock) and Continue Training (local_rts) with a training pattern of free-running PRBS31.
- The O2 training frames are mapped into the LR1 Inner FEC, similar to the PRBS31 generator and checker in option LR1-A.
- TRAINING mode of the ILT function is used to implement this option.

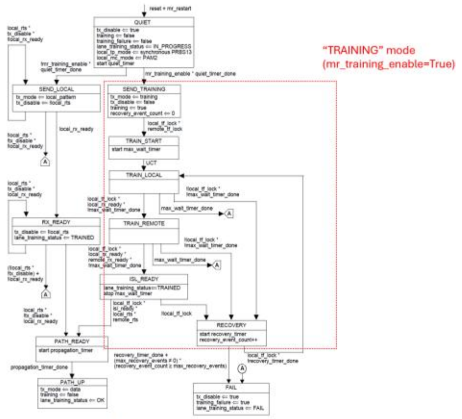
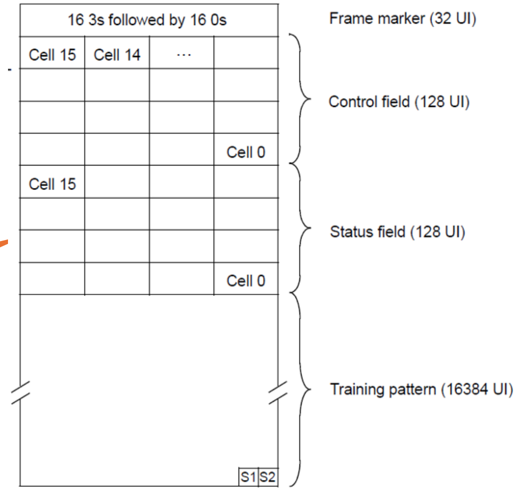


Figure 178B–10—Training control state diagram

O2 training frame insertion point and detection point (same as option LR1-A)

OPTION LR1-B

O2 training frame



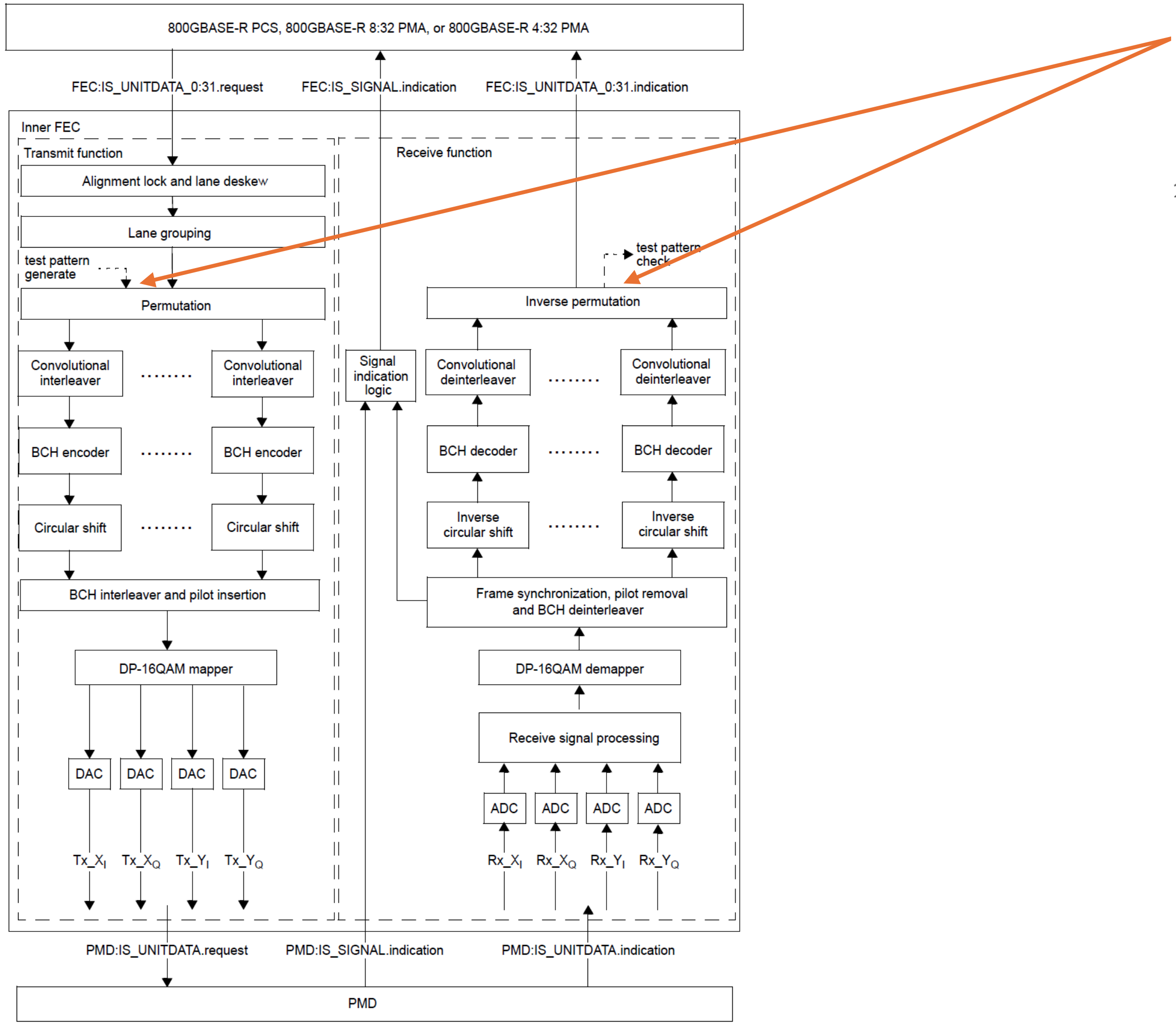


Figure 184–2—Inner FEC functional block diagram

Using pilot symbol sequences

OPTION LR1-C

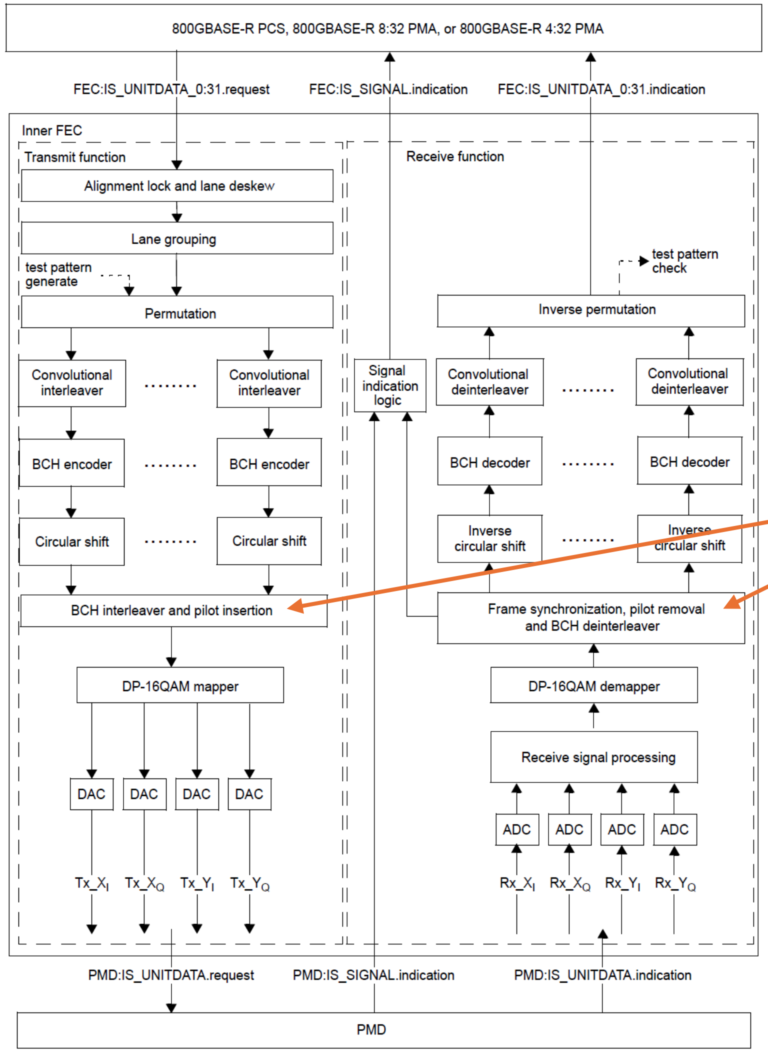


Figure 184–2—Inner FEC functional block diagram

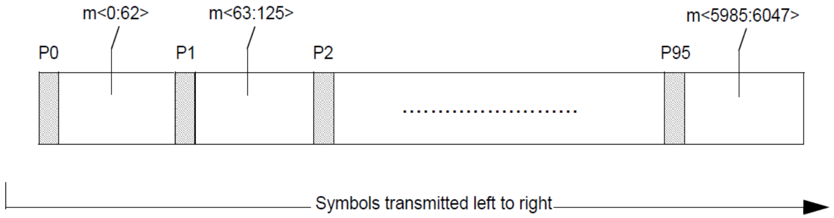
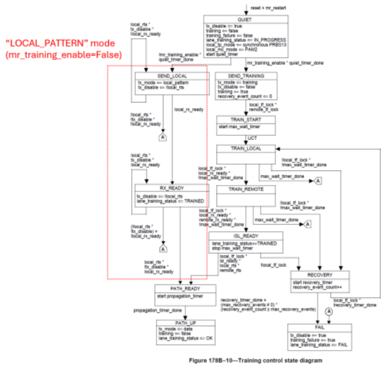


Figure 184–5—DSP frame

- Originally proposed in ILT for Coherent PMDs*
- The DSP frame contains a pilot sequence of 96 x 4-bits (P0 to P95) as described in Cl184.4.9
- Proposal is to use alternate pilot sequences to represent RTS and !RTS
  - Must be DC balanced
  - Good auto-correlation for sync and channel estimation
  - Low cross-correlation to minimize interference
- It may be possible to use the LOCAL_PATTERN mode of the ILT function to implement this option.



*Guangcan Mi, Xiang He (IEEE plenary meeting, July 2025)
https://ieee802.org/3/dj/public/25_07/mi_3dj_01a_2507.pdf

Options for 800GBASE-ER1/ER1-20 (OIF 1600ZR, OIF 1600ZR+)

The options presented in this section apply to ER1, ER1-20, ZR and ZR+

GMP mapped PRBS31

- Same as Option LR1-A, using Cl186 PRBS31 generator/checker point at the GMP mapper.
- Note: in OIF applications, this can be done on a per client basis, which would facilitate client multiplexing.

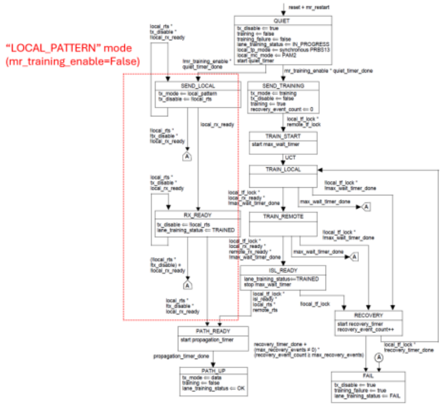


Figure 178B–10—Training control state diagram

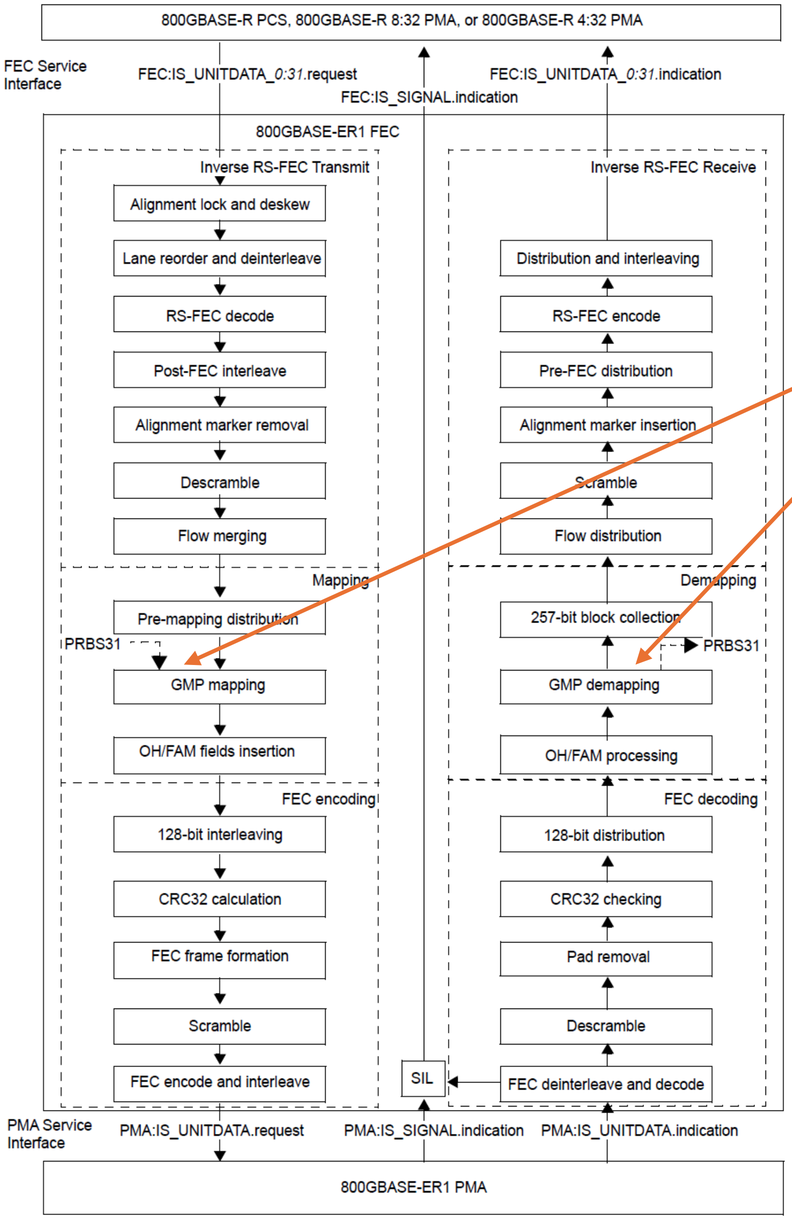


Figure 186–3—800GBASE-ER1 FEC functional block diagram

PRBS31 pattern insertion point and detection point Cl186 (GMP mapped)

Define a new O2 training frame

- Same as Option LR1-B but inserting the O2 training frames at the input of the GMP mapper.
- Note: in OIF applications, this can be done on a per client basis, which would facilitate client multiplexing.

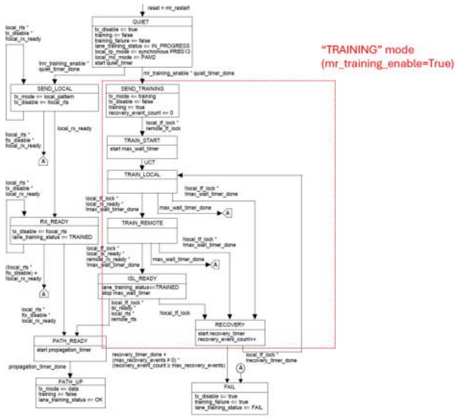


Figure 178B–10—Training control state diagram

O2 training frame insertion point and detection point (same as option ER1-A)

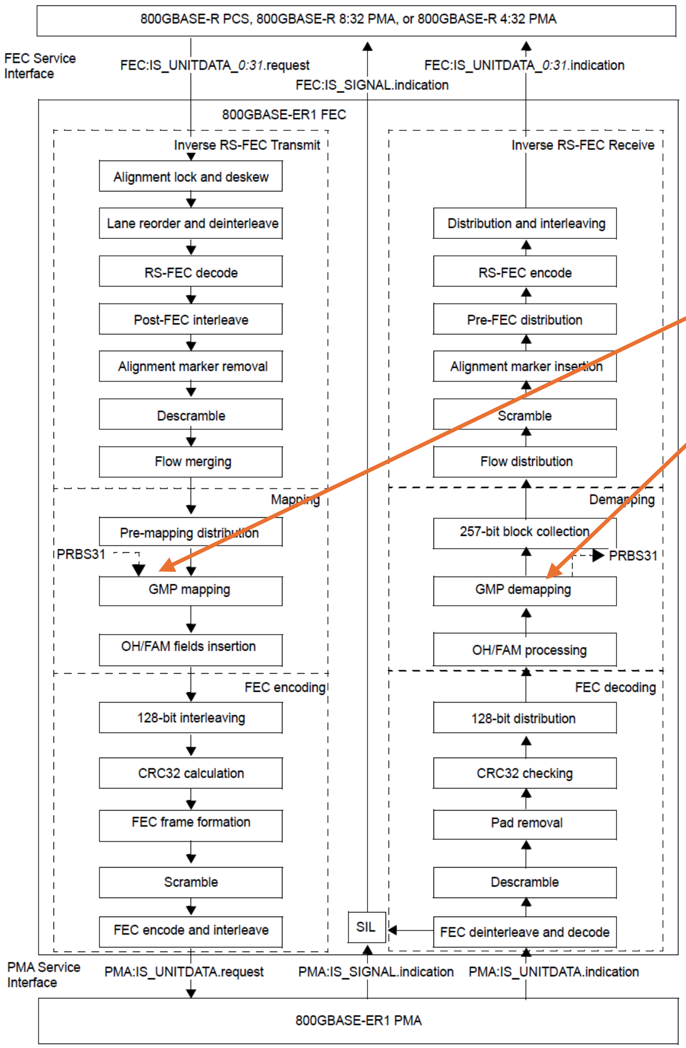
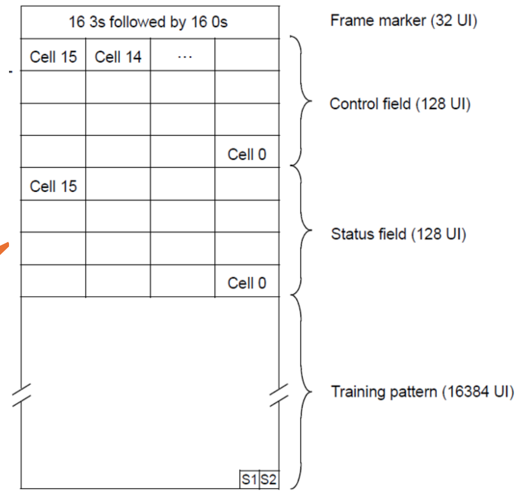


Figure 186–3—800GBASE-ER1 FEC functional block diagram

O2 training frame



Using OH bits in ER1 frame

- Unlike LR1, there are overhead bits available in the ER1 tributary frame (Cl186).
- Define bit values to indicate RTS status.
- Note: in OIF applications, this can be done on a per client basis, which would facilitate client multiplexing.

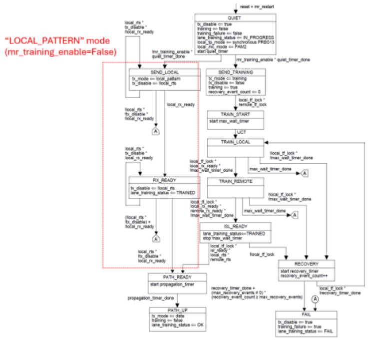


Figure 178B–10—Training control state diagram

OPTION ER1-C

800GBASE-ER1 tributary frame overhead

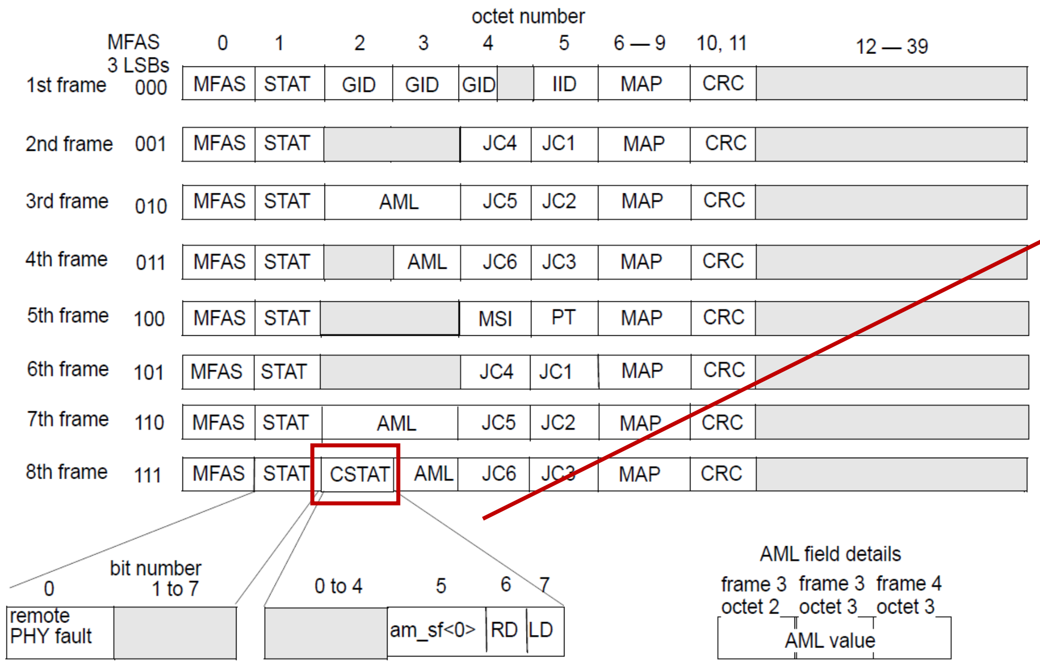
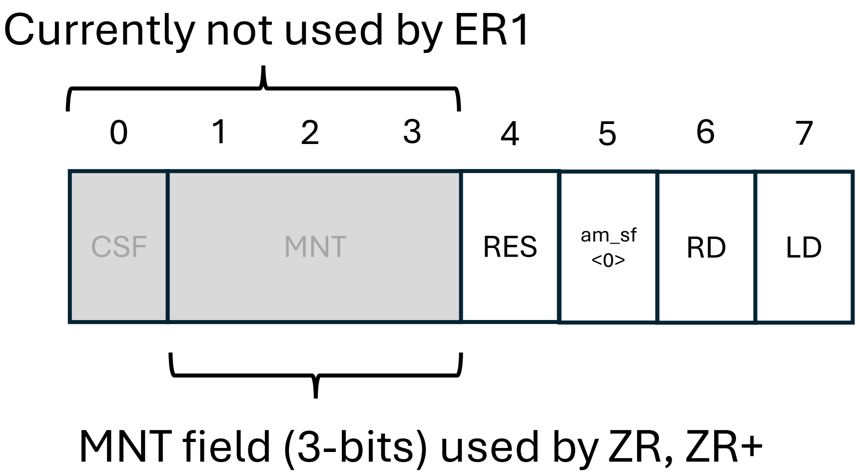


Figure 186–6—Contents of the OH field in eight frame multi-frame



Proposed modification

Value	ITU G709.1	OIF ZR, ZR+	IEEE ER1
000	Normal Operation	Normal Operation	Normal Operation/RTS
001	Not used	Not used	!RTS
010	Not used	Not used	Not used
011	Not used	Not used	Not used
100	Reserved (SquelchText)	Reserved	Reserved
101	LCK	LCK	Reserved
110	OCI	Reserved	Reserved
111	AIS	Reserved	Reserved

Options Matrix

Interface	Options	Design Changes	Changes to D3.0
800GBASE-LR1 (Compatible with OIF 1600 CL)	**LR1-A)** Inner FEC encoded PRBS31	Low	Enhance the Inner FEC PRBS31 test pattern logic to support inverted PRBS31
	LR1-B) Define a new O2 training frame	High	The addition of training frame generation and detection logic in the Inner FEC sublayer
	LR1-C) Using pilot symbol sequences	Medium*/High	Need to specify a new pilot sequence to indicate !RTS, with associated detection logic
800GBASE-ER1/ER1-20 (Compatible with OIF 1600ZR and 1600ZR+)	**ER1-A)** GMP mapped PRBS31	Low	Enhance the Inner FEC PRBS31 test pattern logic to support inverted PRBS31
	ER1-B) Define a new O2 training frame	High	The addition of training frame generation and detection logic in the Inner FEC sublayer
	ER1-C) Using OH bits in ER1 frame	None*/Low	Specify the values of the MNT bits to indicate RTS and !RTS

*assuming programmability already exists

Summary

- All the options are technically feasible.
- The options have different degrees of software and hardware implications on real world implementations.

Backup

Using an O2 training frame

O2 pattern format (identical to O1)

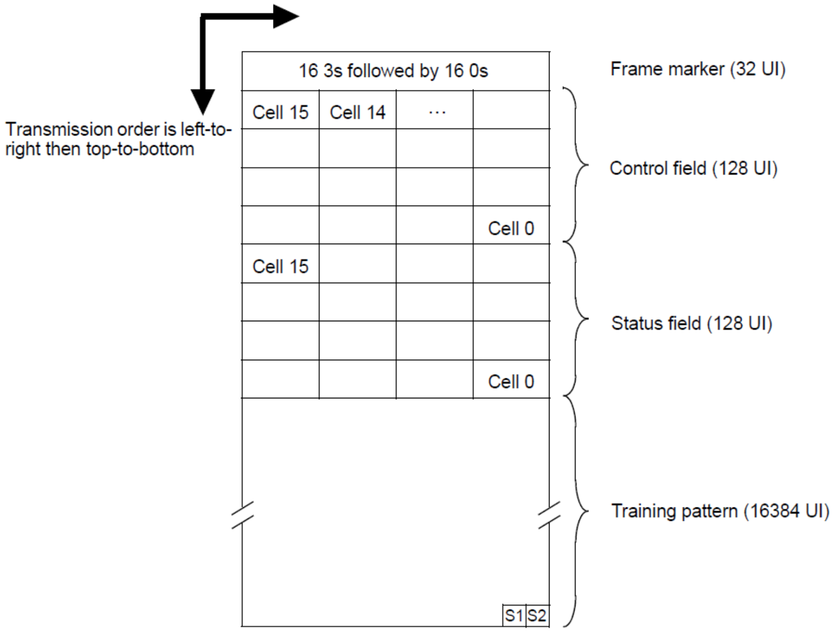


Figure 178B–5—Training frame structure

O2 pattern field structure (red indicates differences to O1, i.e. fields not necessary for this application)

Table 178B–3—Control field structure for the O1 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 00, ignore on receipt
4:0	Reserved	Transmit as 0, ignore on receipt

Table 178B–5—Status field structure for the O1 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 11, 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

ILT function

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 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, provides status in response to requests received, 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 state diagrams in Figure 178B–11 and Figure 178B–12 and their associated variables, apply only when training is enabled (mr_training_enable = true).

RTS function

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. 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), wait for local ISL to complete its adaptation, as indicated by training_status equal to READY or OK
- 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). 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.

APSU

178B.4 Autonomous path startup (APSU)

APSU is not intrinsically a function, it is an externally observable behavior resulting from the RTS and ILT functions. Support for APSU is defined as follows:

— An ISL can be activated using APSU if the two interfaces and the associated sublayers (e.g. PMA, Inner FEC), implement the RTS function (see 178B.6) and the ILT function (see 178B.7), or have equivalent functions.

— A PHY can be activated using APSU if every ISL within the PHY can be activated using APSU.

— An xMII Extender can be activated using APSU if every ISL within it can be activated using APSU.

— A Physical Layer can be activated using APSU if the PHY and xMII Extender (if implemented) can be activated using APSU.

— A path can be activated using APSU if the Physical Layer at each end can be activated using APSU.

An ISL, PHY, Physical Layer, or path that cannot be activated using APSU may be activated using management or other means beyond the scope of this annex.

APSU is the result of each ISL in the path transitioning to DATA mode using the RTS and ILT functions as follows:

— The local_rts variable indicates that an interface is ready-to-send data and propagates from the RS at one end of the path towards the RS at the other end of the path.

— The remote_rts variable indicates that the peer interface is ready to send and receive normal data. It propagates from RS to RS in both directions independently of each other.

— The local_rts and remote_rts variables are propagated only across an ISL that is ready to send data.

— When an interface sends local_rts and receives remote_rts, it means all the ISLs in the same path (see 178B.3) are ready and it switches to DATA mode (tx_mode = data, see 178B.8.3.1).

— When all interfaces in the path are in DATA mode, communication on the path is established.

mr_training_enable

Boolean variable that is set by management. When it is true, training is enabled on the interface. When it is false, training is not enabled on the interface.

802.3dj ILT state diagrams

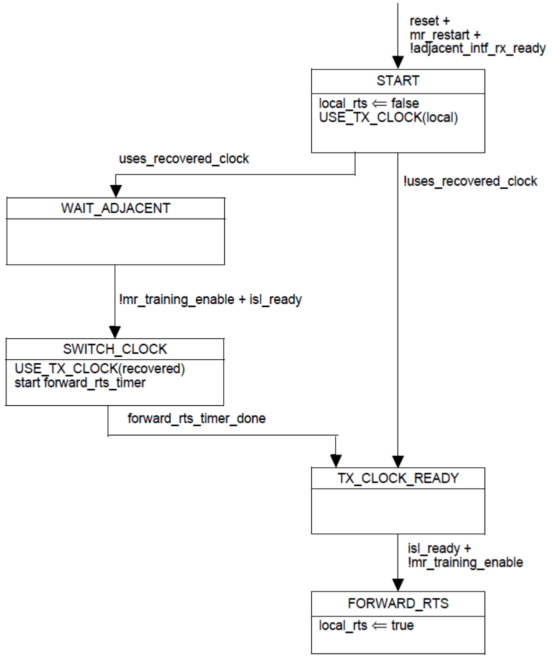


Figure 178B–9—RTS update state diagram

“LOCAL_PATTERN” mode (mr_training_enable=False)

“TRAINING” mode (mr_training_enable=True)

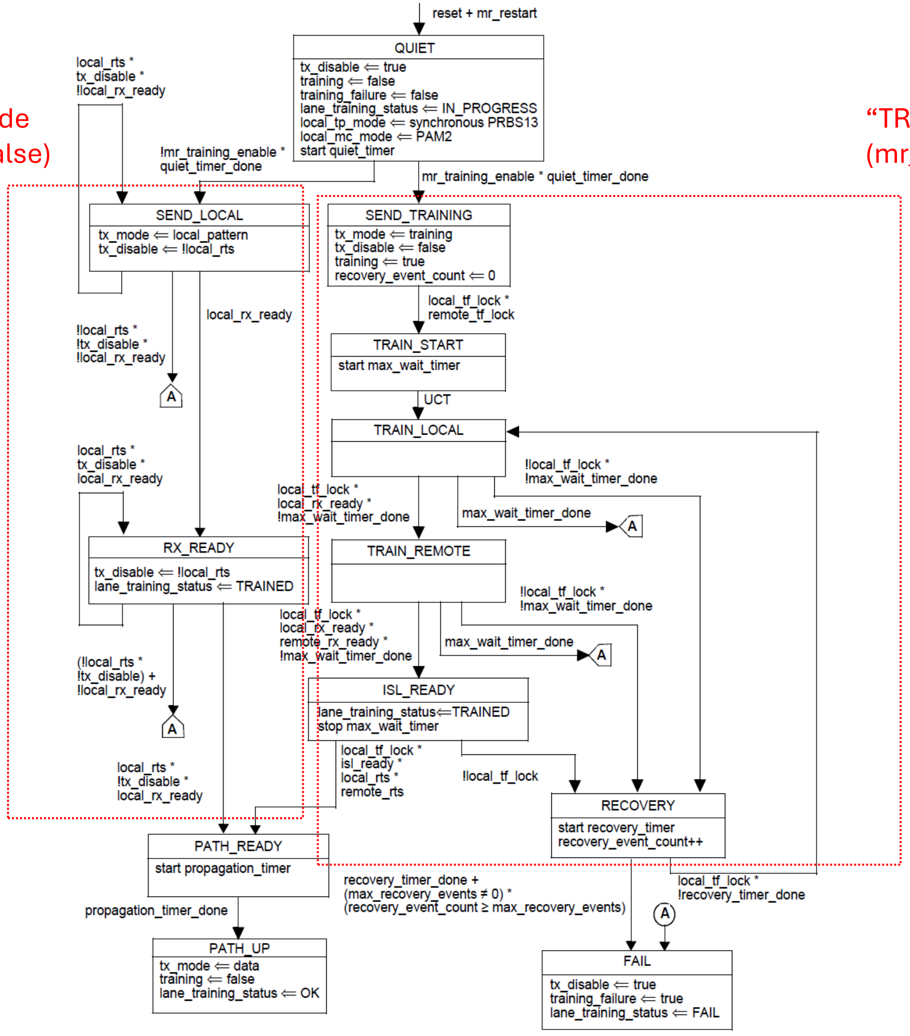


Figure 178B–10—Training control state diagram

RTS and ILT Functions

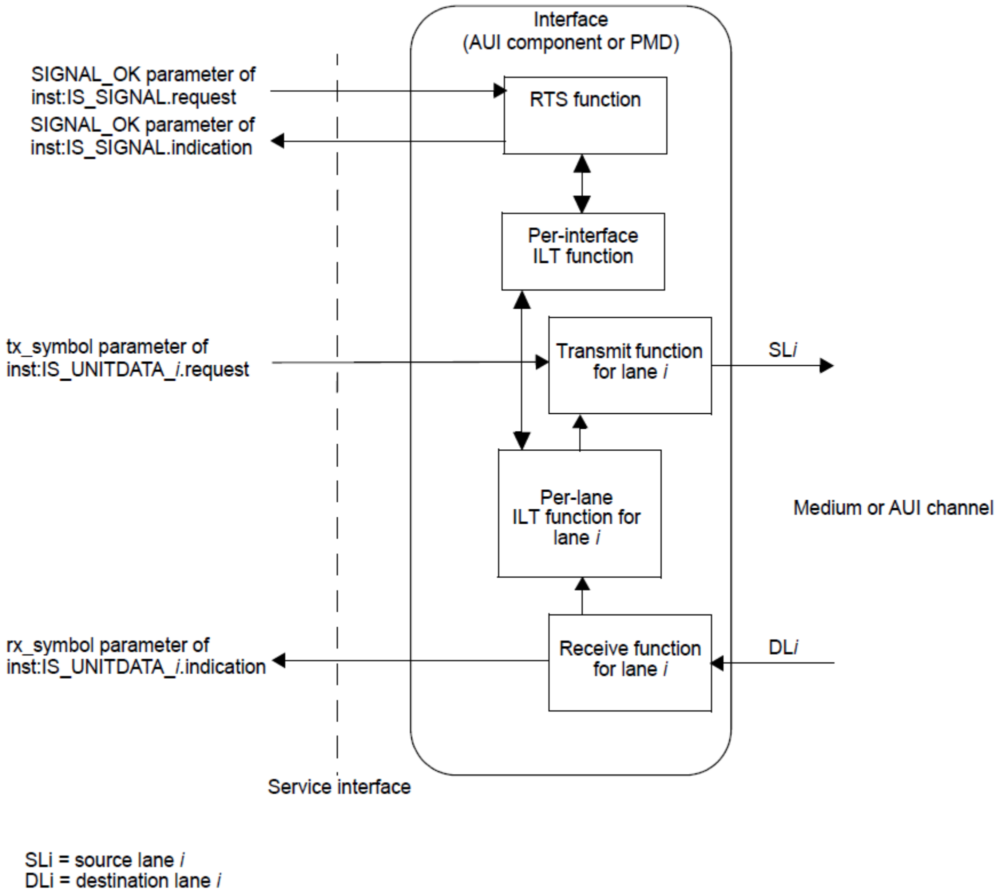
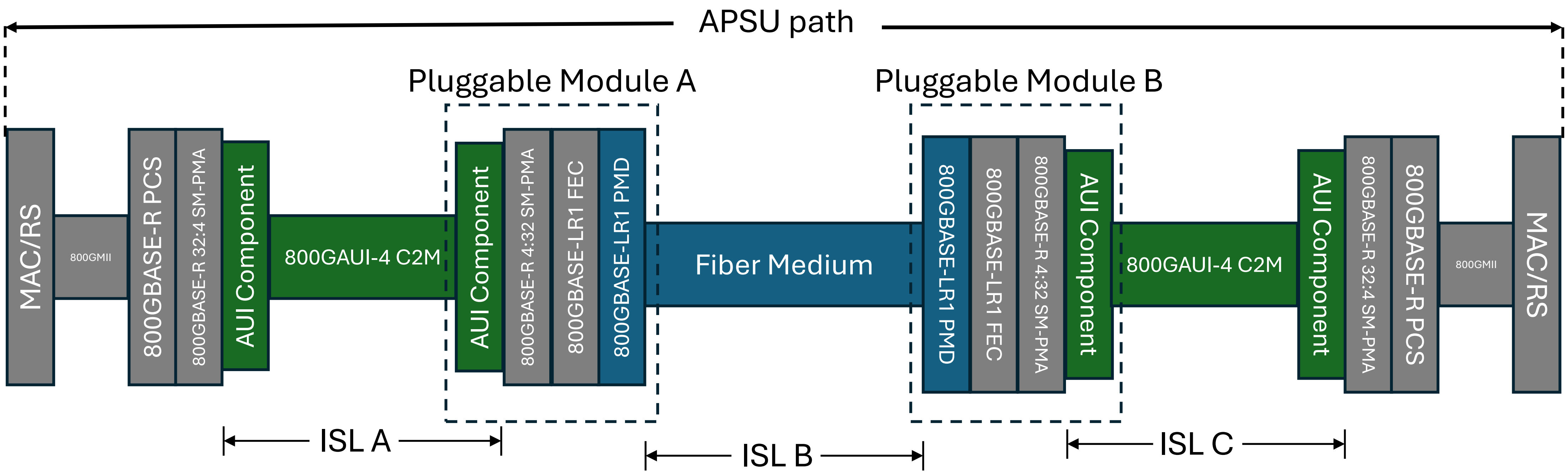


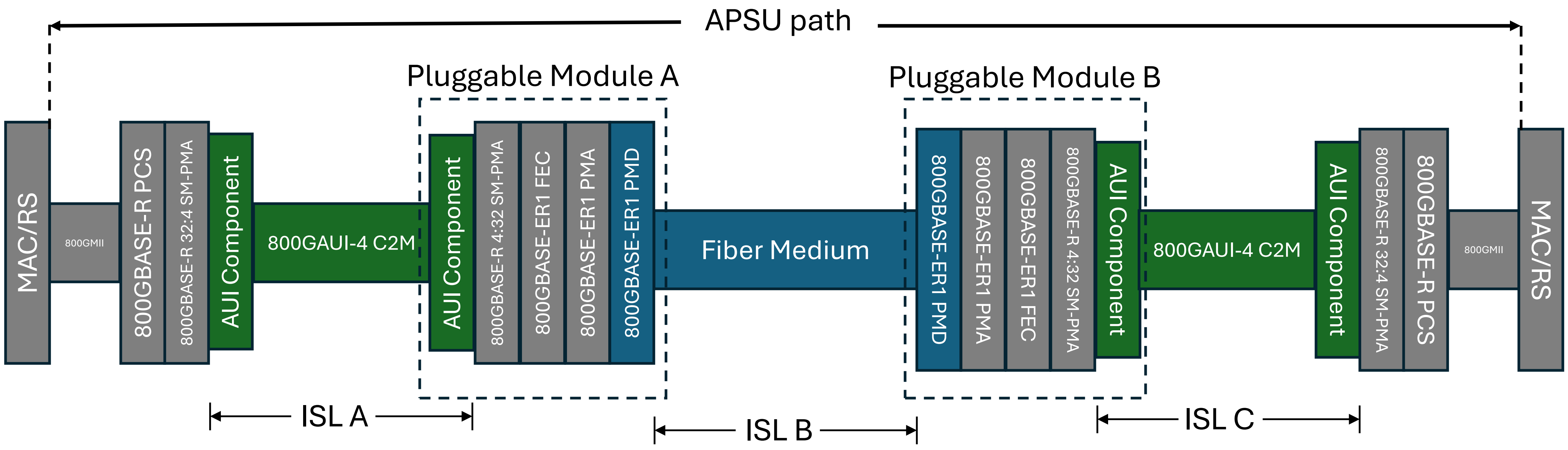
Figure 178B–3—RTS and ILT functions within interfaces

Example 1: 800GBASE-LR1



ISL B currently presents an issue because there is no way to propagate RTS from ISL A to ISL C.

Example 2: 800GBASE-ER1/ER1-20



ISL B currently presents an issue because there is no way to propagate RTS from ISL A to ISL C.