Consensus proposal for RTS FSM for coherent PMDs (comments 140-143, 165-166)

Tom Huber, Nokia

Jeff Slavick, Broadcom

Supporters

- Marco Mascitto, Nokia
- Luz Osorio, Nokia
- Mike Dudek, Marvell
- Matt Brown, Qualcomm
- Leon Bruckman, Nvidia
- Kent Lusted, Synopsys
- Gary Nicholl, Cisco

Introduction

- This is a compromise proposal to clarify RTS for coherent PMDs
  - No changes to IMDD/PAM4 “Link training control” and “RTS Update” FSMs
  - Minimal changes to existing text in the introductory subclauses
  - Clarify 178B.5 through 178B.8 apply only to PAM4 ISLs
  - Introduce a new level 2 subclause to describe RTS for coherent ISLs
- The proposal is not changing the behavior of LR1 or ER1
  - For ER1, it is adding an FSM to describe the RTS function
  - For LR1, it is refactoring the existing FSMs into a single RTS state diagram to remove the ILT aspects that are not used by LR1
  - Provides a consistent methodology for describing RTS for coherent ISLs

Overview of proposal

- In Annex 178B
  - Update headings of 178B.5 to 178B.8 to clarify that they apply only to PAM4 ISLs
  - Minor update to 178B.2 Overview to remove PAM4-specific text
  - Minor update to 178B.4 APSU to remove PAM4-specific text
  - Add new subclause 178B.9 to specify RTS for coherent ISLs
  - Update Management and PICS subclauses to reflect the new RTS for coherent ISLs subclause
- Editorial updates to 185, 186, and 187 to align to the above

178B.5 – 178B.8 are specific to PAM4 ISLs

The minimal set of changes is to clarify 178B.5 through 178B.8 as applying to PAM4 and add a new 178B.9 for coherent ISLs, renumbering the last 3 subclauses and making some updates to the management and PICS clauses:

- 178B.5 – RTS and ILT functions within a PAM4 interface
- 178B.6 – RTS function for PAM4 interfaces
- 178B.7 – ILT function for PAM4 interfaces
- 178B.8 – State diagrams for PAM4 interfaces
- 178B.9 – RTS function for coherent ISLs
    - 178B.9.1 – RTS function for coherent interfaces
    - 178B.9.2 – State diagrams
- 178B.10 Training retimers (current 178B.9; no changes)
- 178B.11 Management variables (current 178B.10; add references for variables)
- 178B.12 PICS (current 178B.11; updates needed)

116.2.9, 169.2.10, 174.2.12

169.2.10 Autonomous path startup (APSU)

The autonomous path startup (APSU) functions are defined in Annex 178B. The inter-sublayer link training (ILT) function facilitates the orderly startup of an inter-sublayer link (ISL). The ready-to-send (RTS) function coordinates the startup of a series of ISLs along a path. Path and ISL are defined in 178B.3.

Change text to:
Autonomous path startup (APSU) is the coordinated, orderly initialization of all inter-sublayer links (ISLs) and sublayers in a path. Functions to support APSU are defined in Annex 178B.

178B.1 Scope

178B.1 Scope

This annex defines ~~the~~ functions to support autonomous path startup (APSU) ~~functions~~ for Physical Layer implementations that include one or more inter-sublayer links (ISLs) (see 178B.3) with data rate of 200 Gb/s or higher per lane.

178B.2 Overview

Yellow highlighted text is specific to PAM4 ISLs, since ILT is only used by those ISLs. Since APSU includes coherent ISLs, the text should be generalized.
Green highlighted text is missing detail on sublayers

APSU is the coordinated, orderly initialization of all ISLs in a path (Figure 178B–1). APSU is facilitated by the combination of the ready-to-send (RTS) function and the inter–sublayer link training (ILT) function. APSU is described in 178B.4.

The RTS function is an internal function of an AUI component or a PMD, collectively referred to as interfaces (see 178B.3), that propagates the readiness of each ISL to transition to DATA mode. The RTS function propagates the RTS status between peer and adjacent interfaces. RTS status indicates whether an ISL is ready to transition to DATA mode, or not. The RTS status is transferred between adjacent interfaces via the service interface primitives and between peer interfaces via the ILT function. The RTS function is described in 178B.6.

The ILT function is an internal function of an AUI component or a PMD that controls the establishment of communication with a peer interface within an inter-sublayer link (ISL). The ILT function provides optimization of data signaling and coordinates the transition to DATA mode on an ISL. The ILT function is defined in 178B.7.

NOTE—AUI components and PMDs that are not specified to support the ILT function or the RTS function as defined in this annex may include equivalent capabilities, however this is beyond the scope of this standard.

Terminology used in this annex is provided in 178B.3.

Add "and sublayers" after ISLs.

In this text, change "via the ILT function" to "the ILT function or another PMD-specific function."

Change the start of the paragraph to "For PAM4 ISLs, the ILT function is an internal function…". With that change, the highlighted text is accurate.

178B.4 APSU (first half)

Highlighted text is specific to PAM4 ISLs and not correct for coherent ISLs

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.

Change “the RTS and ILT functions” to “establishing communication through all sublayers in the path in both directions”

Change to “(see 178B.6 and 178B.9)”

Add “and sublayer” after ISL

178B.4 APSU (second half)

APSU is the result of each ISL in the path transitioning to DATA mode as follows:

— The local_rts variable at an interface indicates that the interface is ready to send data. An interface that uses this annex signals that it is ready to send to a peer interface via the transmitted signal, as specified in 178B.6. An interface signals that it is ready to send data to an adjacent sublayer via the inter-sublayer service interface.

— The remote_rts variable indicates that the peer interface is ready to send. An interface asserts remote_rts when the receive function detects RTS signaling; this indicates that local_rts is true in the peer interface.

— When both local_rts and remote_rts are true for an interface, it switches to DATA mode (tx_mode = DATA, see 178B.8.3.1).

— End-to-end communication is established when all interfaces in the path are in DATA mode.

Replace second paragraph and bullet list with:

The local_rts and c_local_rts variables indicate that an interface is ready to send data. The remote_rts and c_remote_rts variabels indicate that the peer interface is ready to send data. These variables are set by the RTS functions.

APSU is the result of SIGNAL_OK for every sublayer interface reaching the OK status. This occurs when:

— Each sublayer is receiving and transmitting data of the expected format.

— Communication is established in both directions of each ISL. For PMDs using the ILT and RTS functions in 178B.8, the variables local_rts and remote_rts are both true. For PMDs using the RTS function in 178B.9, the variables c_local_rts and c_remote_rts are both true.

— Each sublayer is receiving and transmitting data of the expected format.

— End-to-end communication between the RS sublayers is established

Text for new subclause 178B.9

178B.9 RTS function for coherent PMDs

This subclause describes the RTS function for use with PMDs using coherent detection.

Text for new subclause 178B.9.1

178B.9.1 RTS function within a coherent interface

The RTS function in a coherent interface controls the behavior of the transmitter during startup, including the switch over of the transmitter clock source to its DATA mode clock (when necessary). It indicates to peer and adjacent interfaces whether the interface is ready to switch to DATA mode. The RTS function state diagrams (Figure 178B-y and Figure 178B-z) and their associated variables define the process used to communicate the interface status and control the clock source and data source transitions.

An overview of the behavior of the RTS function for coherent interfaces is as follows:

- Initially, local_rts and remote_rts variables are false
- Wait for training of the interface (details are PMD-specific) to complete and for the upstream adjacent interface to be ready for DATA mode (as indicated by the SIGNAL_OK parameter from the adjacent interface being READY or OK).
- If the interface is implemented in a retimer that uses the recovered clock in DATA mode, change to the recovered clock
- local_rts becomes true, indicating readiness to send data (SIGNAL_OK=READY)
- Wait for the peer to be ready, as indicated by remote_rts
- Wait for SIGNAL_OK to be OK, and switch to DATA mode

Text for new subclause 178B.9.2

178B.9.2 State diagrams

This subclause is composed of detailed state diagram processes, including the associated definition of variables, functions, counters, and timers.

These state diagrams are specific to coherent PMDs. Some coherent PMDs squelch the output when the PMD is not ready to send, while others do not. Separate state diagrams are used for the different types of coherent PMDs.

178B.9.2.1 Variables (1)

179B.9.2.1 Variables

c_adjacent_signal_ok: Enumerated variable derived from the value of the SIGNAL_OK parameter on the adjacent service interface. It takes one of the following values: IN_PROGRESS, READY, OK, FAIL

c_adjacent_intf_rx_ready: Boolean variable that is true when c_adjacent_signal_OK is either READY or OK and false otherwise. If there is no adjacent interface this variable is always true.

c_isl_ready: Boolean variable that is set to true when c_local_rx_ready ~~and remote_rx_ready are true for all lanes of the interface~~ is true and the ISL has completed any other training processes, and to false otherwise. The criteria for completing training are specified by the clause or annex that defines the interface.

c_local_rts: Boolean variable that indicates that an interface is ready to send data. ~~It is set by the RTS update state diagram (Figure 178B-9). The logical-NOT of this variable is encoded as the not ready to send bit in the control field of transmitted training frames.~~ It is true if c_isl_ready and c_adjacent_intf_rx_ready are true and is otherwise false.

178B.9.2.1 Variables (2)

c_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 the expected signal format, ~~that the remote transmit has been optimized if mr_training_enable is true,~~ that the local receive equalizers have been optimized, and that no further adjustments are required for normal data transmission. The specific conditions for setting this variable to true are implementation dependent. It shall be set to false when no ~~an~~ input signal (corresponding to ~~tx_disable = true~~ tx_mode = QUIET in the peer) is received for 100 ms (the minimum duration of the quiet_timer) or longer. ~~When in the RECOVERY state (Figure 178B–10a) this variable is false.~~

c_mr_restart: Boolean variable used by management to restart the ~~RTS and ILT functions~~ RTS function.

c_remote_rts: Boolean variable that indicates the value of local_rts variable in the peer interface. The mechanism for signaling RTS status is specified by the clause or annex that defines the interface. ~~If mr_training_enable is true then remote_rts is set to true when the received training frames on all lanes of the interface have the not ready to send bit (see 178B.7.4.2) set to zero, otherwise it is set to false. If mr_training_enable is false then~~ For a coherent interface with squelching, remote_rts is always true.

178B.9.2.1 – Variables (3)

c_reset: Boolean variable that controls ~~the~~ global resetting of the ~~RTS and ILT~~ RTS state machine~~s~~. It is true whenever a reset is necessary including when initiated by ~~PMA_reset for AUI components, PMD_reset for PMDs~~ reset of the FEC, PMA, or PMD sublayer and during power on, and it is false otherwise.

c_uses_recovered_clock: Boolean variable set to true to indicate that the interface uses a clock recovered by an adjacent interface (see 178B.10~~9~~) to drive its output in DATA mode ~~(tx_mode = DATA, see 178B.8.3.1). Otherwise it is~~ and set to false otherwise.

178B.9.2.1 Variables (4)

c_tx_mode: Enumerated variable that controls the content of the transmitter output of the lane ~~when tx_disable is false~~. It is set by the RTS ~~training control~~ state diagrams (Figure 178B–y~~10a~~ and Figure 178B–z~~10b~~) to one of the values:

- QUIET: transmitter output squelched
- TRAINING: transmit the training signal specified by the clause or annex that defines the interface ~~training frames (see 178B.7.3).~~
- ~~LOCAL_PATTERN: transmits a pattern specified by the clause or annex that defines the interface.~~
- NOT-RTS: transmit the not-ready-to-send signal specified by the clause or annex that defines the interface
- RTS: transmit the ready-to-send signal specified by the clause or annex that defines the interface
- FILL: transmit the fill signal specified by the clause or annex that defines the interface ~~either training frames or LOCAL_PATTERN~~
- DATA: transmit data from the client sublayer ~~PMA or Inner FEC~~.

178B.9.2.2 - Functions

179B.9.2.2 Functions

C_USE_TX_CLOCK(source): selects the clock source for the transmitter. The source parameter takes one of the values:

- local: use a local clock to drive the transmitter output
- recovered: use a clock recovered by the adjacent interface receiver to drive the transmitter output
- NOTE – the details of the clock recovery and forwarding architecture are beyond the scope of this standard

178B.9.2.3 – Timers

179B.9.2.3 Timers

c_forward_~~rts~~clock_timer: This timer is started when the ~~RTS Update~~ RTS state diagram (~~Figure 178B-9~~ Figure 178B-y or Figure 178B-z) enters the SWITCH_CLOCK state. It provides time for the clock switch to propagate to the peer interface. The terminal count of forward_ ~~rts~~clock_timer is between 10 ms and 90 ms.

c_propagation_timer: This timer is started when the ~~training control state diagram (Figure 178B–10b) on a lane~~ RTS state diagram (Figure 178B-y or Figure 178B-z) enters the PATH_READY state. It provides time for the RTS status to propagate before switching to mission data. The terminal count of this timer is between 100 ms and 200 ms.

c_quiet_timer: This timer is started when the ~~training control state diagram (Figure 178B–10a) on a lane enters~~ RTS state diagram is in the QUIET state. It provides a period of time when the transmitter output is squelched. The terminal count of this timer is between 100 ms and 200 ms.

178B.9.2.4 – State diagram figures

Text proposal:

178B.9.2.4 State diagram figures

Coherent interfaces that squelch when they are not ready to send shall implement the state diagram shown in figure 178B-y.

Coherent interfaces that do not squelch when they are not ready to send shall implement the state diagram shown in figure 178B-z.

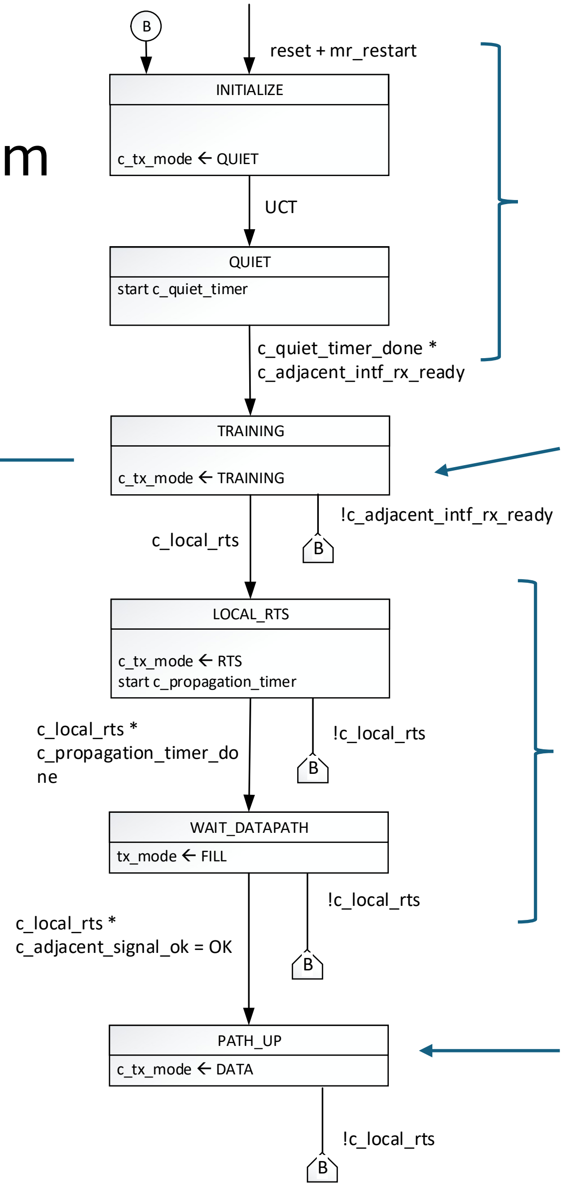
In the two slides that follow, the proposal is to include the state diagram figures only. The other text and tables (including the blue arrows pointing from text to specific states and braces) are intended to explain the diagrams in the context of this presentation.

Figure 178B-y
RTS with squelch state diagram

This is used with LR1

“training” means transmitting the test pattern and tuning the rx based on the received test pattern (which may already have been done, or may happen later, since the two directions are independent)

tx_mode	Output
QUIET	squelch
TRAINING	PRBS31
RTS	PRBS31
FILL	PRBS31
DATA	data



Squelch until quiet_timer is done and upstream intf is ready

Wait for the ISL to be ready

Wait for the status to propagate to the ends of the path, then for SIGNAL_OK to become OK

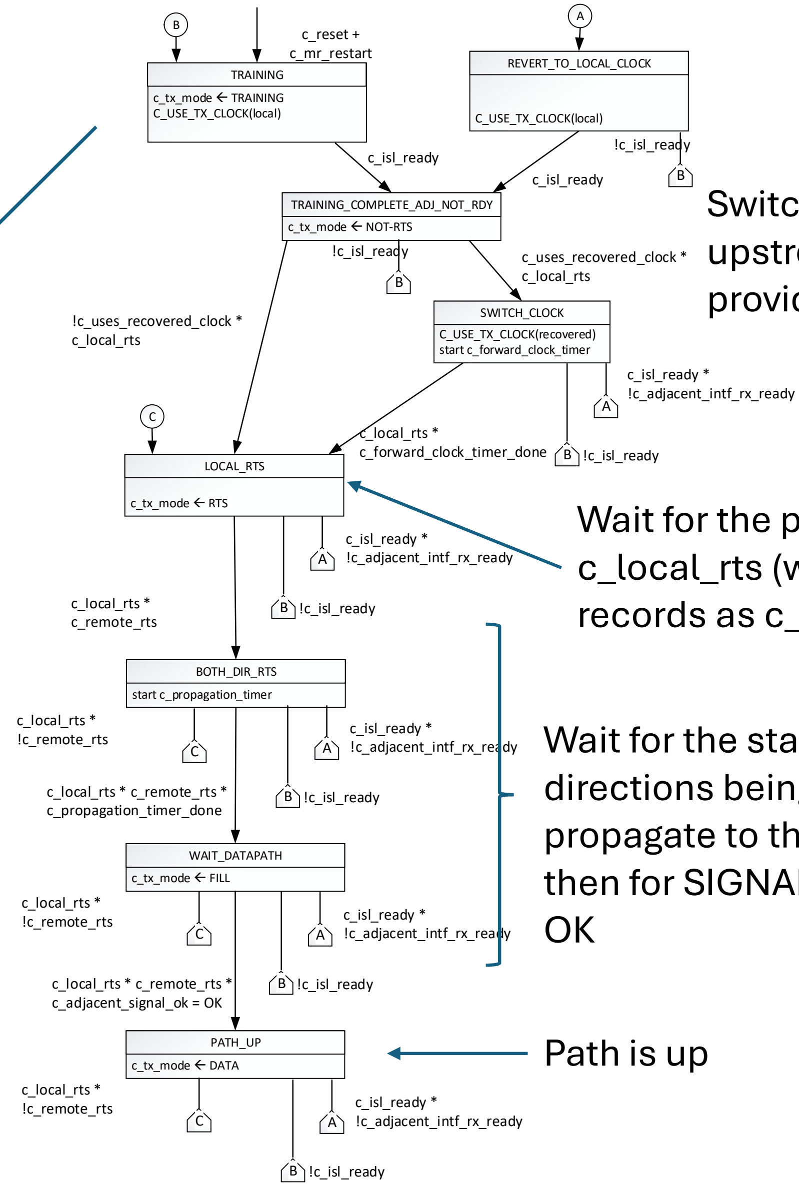
Path is up

Figure 178B-z RTS without squelch state diagram

Used with ER1

ER1 always transmits on its own clock, so no quiet period is needed, so the FSM starts at TRAINING. Training is complete when frame alignment is complete and the receiver is adjusted

C_tx_mode	Client signal	MNT value
TRAINING	Idle pattern	IN_PROGRESS
NOT-RTS	Idle pattern	IN_PROGRESS
RTS	Idle pattern	READY
FILL	Idle pattern	READY
DATA	Mission data	OK



Switch clock when upstream indicates it is providing mission clock

Wait for the peer to reach c_local_rts (which this node records as c_remote_rts)

Wait for the status of both directions being ready to propagate to the ends of the path, then for SIGNAL_OK to become OK

Path is up

Management variable table updates

- Update the management variable tables to include the variables that are used in the new subclause 178B.9.2.1 are included as references in the table
  - e.g., isl_ready currently references 178B.8.2.1; c_isl_ready and a reference to 178B.9.2.1 would be added, as shown below
  - Similar changes need to be made for the other state variables from 178B.9.2.1 that appear in these tables

Status variable	Variable reference	MDIO register/bit number	MDIO register/bit reference
isl_read, c_isl_ready	178B.8.2.1, 178B.9.2.1	1.1476.8	45.2.1.168c

PICS updates

- Add a table defining three ‘major capabilities’
  - APSU for PAM4 ISLs
  - APSU for coherent ISLs that squelch
  - APSU for coherent ISLs that do not squelch
- 178B.11.3
  - Update the title of the clause to indicate it applies only to PAM4 ISLs
  - Make all rows conditional based on the major capability for PAM4 ISLs
- 178B.11.5
  - Add two new rows, one for each of the coherent RTS FSMs
  - Make all rows conditional based on the appropriate major capability

Subclauses 185.5.6 and 185.5.7

185.5.6 PMD reset function

If the variable PMD_reset is asserted, the PMD shall be reset as defined in 45.2.1.1.1.

PMD reset shall reset the ILT and RTS functions associated with the PMD (see 185.5.11).

185.5.7 Global PMD transmit disable function (optional)

The global PMD transmit disable function enables the optical transmitter to be disabled. This function is used by the ILT function. It may optionally be provided through the control variable Global_PMD_transmit_disable.

If either Global_PMD_transmit_disable or the variable tx_disable of the ILT function (see 185.5.11) is set to one, this function shall turn off the optical transmitter so that the transmitter meets the requirements of the average launch power of the OFF transmitter in Table 185–5.

If a PMD fault is detected (see 185.5.8), then the PMD may set the Global_PMD_transmit_disable variable to one, turning off the optical transmitter.

Text is specific to using the same FSMs as PAM4 ISLs

Change to “the RTS function”

Delete this.

Change to “the variable c_tx_mode = QUIET (see 185.5.11)”

Subclause 185.5.11

A PMD shall provide the ready-to-send (RTS) function using the RTS with squelch state diagram (see 178B.9.2) ~~and the ILT function~~ as specified in Annex 178B.

~~The variable mr_training_enable is always set to false.~~

When c_tx_mode = QUIET, the PMD squelches the output.

When c_tx_mode = TRAINING, RTS, or FILL~~local_pattern (see 178B.8.3.1)~~, the PMD transmits PRBS31 encoded by the 800GBASE-LR1 Inner FEC as defined for the test pattern generator in 184.7.1.

The variable c_isl_ready is true when c_local_rx_ready is true.

New clause 186.2.6

186.2.6 Autonomous path startup (APSU) functions

The ER1 FEC shall provide the ready-to-send (RTS) function using the RTS without squelch state diagram (see 178B.9.2) as specified in Annex 178B.

The RTS status is signaled via the MNT bits using the mapping defined in 186.2.2.

The transmit function transmits ER1 FEC frames regardless of the value of c_tx_mode. When c_tx_mode = TRAINING, NOT-RTS, RTS, or FILL, the payload of the ER1 FEC frame is filled with the scrambled idle test pattern defined in 186.2.3.12.

The variable c_isl_ready becomes true when the FEC sublayer has aligned to the ER1 FEC frame and the ER1 tributary frame and c_local_rx_ready is true. Note that this requires that the PMA be aligned to the DSP frame also.

The variable c_remote_rts is set to true if the received SIGNAL_OK status in the MNT bits from the peer interface indicate a status of READY or OK (see Table 186-2), and is set to false otherwise.

Changes to 187.5.11

APSU functions are supported by the 800GBASE-ER1 FEC sublayer (see 186.2.6).

Summary

- This proposal adds RTS FSMs to describe the RTS process for coherent ISLs
  - Provides a common descriptive methodology for APSU support for all ISLs
  - Simplifies understanding of APSU for readers who are only interested in coherent ISLs
  - Makes a minimal set of editorial changes to existing text to clarify aspects that apply only for PAM4 ISLs
  - Provides a solid foundation for APSU for any future coherent ISLs to use, making the APSU material in 802.3dj a complete solution

Thank you!

Backup

History

- APSU FSMs were substantially revised in September 2025 interim meeting
- 14 April 2026 ad hoc straw poll #1 showed interest in pursuing full refactoring of APSU FSMs (Y:14 N:4 NMI:2 A:3)
- A more complete proposal was provided in May; concerns were raised and discussion shifted toward making no changes to existing FSMs, only introducing new FSMs for coherent ISLs
- 3 June ad hoc introduced proposed RTS for coherent PMDs FSM
- 25 June ad hoc provided an update addressing comments
- Straw poll #2 showed interest in this direction (Y:12 N:5, A:3), straw poll #3 showed support for the specific changes being proposed (Y:13 N:4, A:4)
- This presentation provides further updates/simplifications to address concerns about complexity

Why does ER1 need states related to a clock transition? Doesn't ER1 use its own clock?

- While the ER1 FEC/PMA/PMD sublayers always uses a local clock and map the 'client' (i.e., the signal received from the AUI or PCS) asynchronously into the ER1 frame structure, the client signal still has a clock transition
  - While the AUI is training, mission clock is not available; the ER1 FEC sublayer needs to map something into the payload of the ER1 tributary frame structure, and that signal is generated with a local clock
  - When the AUI reaches ISL_READY, mission clock is available and the client of the ER1 FEC needs to change, even though the ER1 FEC/PMA/PMD clock itself does not change
  - The client clock is conveyed across the ER1 FEC/PMA/PMD, so the concept of forwarding mission clock once the AUI starts providing mission clock applies to ER1 PMDs also
- The RTS function applies to all ISLs
  - The fact that an ER1 PHY signals RTS information differently from PAM4 PHYs or LR1 PHYs doesn't mean ER1 PHYs don't need an RTS function
  - Describing the behavior of an ER1 PHY as tunneling the RTS status of the upstream AUI rather than implementing an RTS function is an incomplete description; when the PMD is failed or not yet trained, it clearly is not tunneling the status of the upstream AUI, but instead providing its own RTS status information

Overview of 178B.9.2.x subclause proposals

- Subclauses for variables, functions, timers, and state diagrams, following the usual format
- The text for the variables, functions, and timers is shown with revision marks compared to the definitions that are in 178B.8.2.x or 178B.8.3.x (changes are to remove aspects that are PAM4-specific or introduce material that is needed for coherent) to make it easier to see what is different for coherent vs PAM4 ISLs
- However, this will all be new text in a new subclause; no changes to 178B.8.x are being proposed