802.3dj D3.1 Comment Resolution Common Topics

Matt Brown (Qualcomm), 802.3dj Chief Editor
Leon Bruckman, Nvidia

Introduction

- This slide package was assembled by the 802.3dj editorial team to provide background and detailed resolutions to aid in comment resolution.
- Specifically, these slides are for the various common-topic comments.

Comment #169
APSU wording changes

Cl **169** *SC* **169.2.10** *P* **210** *L* **40** # 169

Slavick, Jeff — Broadcom Inc

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

800GBASE-ER1 does not use the ILT and RTS functions. APSU is the result of establishing end to end communicatioin. ILT and RTS functions provide means to establish communication across ISLs, but are not required to do so (like in the case of ER1).

SuggestedRemedy

Update 116.2.9, 169.2.10 and 174.2.12
Autonomous path start-up (APSU) is the orderly startup of the path, which occurs when SIGNAL_OK reaches the OK status at every sublayer interface in both directions. This requires communication to be established across all ISLs within the path (see 178B.3) along with all sublayers transmit and receive paths operating as expected. PHYs and Physical layers capable of sending SIGNAL_OK in both the transmit and receive directions are able to make use of APSU.

Annex 178B defines the RTS and ILT functions that facilitate establishing the communication and the transfer of SIGNAL_OK across ISLs found within a path. These functions are supported by many PMD and AUI components that utilize various types of medium.

Details of how each PMD or AUI component supports APSU are found in the relevant clauses.

APSU is specified for use by the following PMD and AUI types:

<existing list of PMD and AUI types>

Update 178B.1 to be:
This annex defines the functions to support autonomous path startup (APSU) for Physical Layer implementations that include one or more inter-sublayer (ISLs) (see 178B.3) with data rate of 200 Gb/s.

Update 178B.2 first paragraph to be:
APSU is the coordinated, orderly initialization of all ISLs and sublayers 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.

Update 178B.4 to be:

APSU is not intrinsically a function, it is an externally observable behavior resulting from establishing communication through all sublayers in the path in both directions. 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 and sublayer within the PHY can be activated using APSU.
— An xMII Extender can be activated using APSU if every ISL and sublayer 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.

A path that does not meet the above criteria is not defined to support 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 SIGNAL_OK for every sublayer interface reaching the OK status. This occurs when:
— Each ISL establishes communication in both directions. For PMD and AUI components using the ILT and RTS functions local_rts and 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.

Proposed Response *Response Status* **O**

Comment #169
Changes to clauses 116, 169, and 174

116.2.9 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.

ILT and RTS functions are specified for use by the following PMD and AUI types:

— 200GAUI-1 C2C

— 200GAUI-1 C2M

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.

174.2.12 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.

ILT and RTS functions are specified for use by the following PMD and AUI types:

Illustration of proposed changes.

116.2.9 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.~~

Autonomous path start-up (APSU) is the orderly startup of the path, which occurs when SIGNAL_OK reaches the OK status at every sublayer interface in both directions. This requires communication to be established across all ISLs within the path (see 178B.3) along with all sublayers transmit and receive paths operating as expected. PHYs and Physical layers capable of sending SIGNAL_OK in both the transmit and receive directions are able to make use of APSU.

Annex 178B defines the RTS and ILT functions that facilitate establishing the communication and the transfer of SIGNAL_OK across ISLs found within a path. These functions are supported by many PMD and AUI components that utilize various types of medium.

Details of how each PMD or AUI component supports APSU are found in the relevant clauses.

~~ILT and RTS functions are~~ APSU is specified for use by the following PMD and AUI types:

<existing list of PMD and AUI types>

The proposed new initial paragraph is far too much detail for this summary subclause and, if needed, then should perhaps be added to Annex 178B.

Comment #169
Changes to 178B.1 and 178B.2

178B.1 Scope

This annex defines the 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

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

Illustration of proposed changes.

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~~.

The replacement with “functions to support” is unnecessary as it does not change anything. The other two changes are errors.

178B.2 Overview

APSU is the coordinated, orderly initialization of all ISLs and sublayers 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.

Comment #169
Changes to 178B.4

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.

A path that does not meet the above criteria is not defined to support 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 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.

Illustration of proposed changes.

178B.4 Autonomous path startup

APSU is not intrinsically a function, it is an externally observable behavior resulting from ~~the RTS and ILT functions~~ establishing communication through all sublayers in path in both directions. 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 and sublayer within the PHY can be activated using APSU..
— An xMII Extender can be activated using APSU if every ISL and sublayer 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..

A path that does not meet the above criteria is not defined to support 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 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.~~

APSU is the result of SIGNAL_OK for every sublayer interface reaching the OK status. This occurs when:
— Each ISL establishes communication in both directions. For PMD and AUI components using the ILT and RTS functions local_rts and 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.

Comment #188
State diagram

Cl **178B**	*SC* **178B.8.3.1**	*P* **898**	*L* **11**	# R1-188

Slavick, Jeff — Broadcom Inc

Comment Type **TR** *Comment Status* **D** *State diagrams (CA)*

Stating that variable takes on a particular value in a given state diagram state meas its value should be set by the state diagram.
In Figure 136-7 local_rx_ready is a variable set in the state diagram and depends upon a local_trained variable.
We should take on this same format for 178B-10

SuggestedRemedy

Rename local_rx_ready to local_trained and remove the last sentence of the current definition.
Create a new definition for local_rx_ready with a definition of "Boolean variable that indicates the state of the receiver. The value of local_rx_ready is set by the training control state diagram (Figure 178B-10a and Figure-10b).
In Figure 178B-10a and 10b replace all current instances of local_rx_ready with local_trained

In Figure 178B-10a add "local_rx_ready = false" to the INITIALIZE and RECOVERY states.
In Figure 178B-10b add "local_rx_ready = false" to the LOSS_OF_SIGNAL state.
In Figure 178B-10a add "local_rx_ready <= true" to TRAIN_REMOTE and RX_READY states.
In Figure 178B-10b add "local_rx_ready <= true" to PATH_UP state.

Proposed Response *Response Status* **W**

PROPOSED ACCEPT IN PRINCIPLE.
Implement with editorial license.

Comment #188 - Variables updates

local_trained

~~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 an input signal corresponding to tx_disable = true 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.~~

New

local_rx_ready

Boolean variable that indicates the state of the receiver. The value of local_rx_ready is set by the training control state diagram (Figure 178B-10a and Figure-10b).

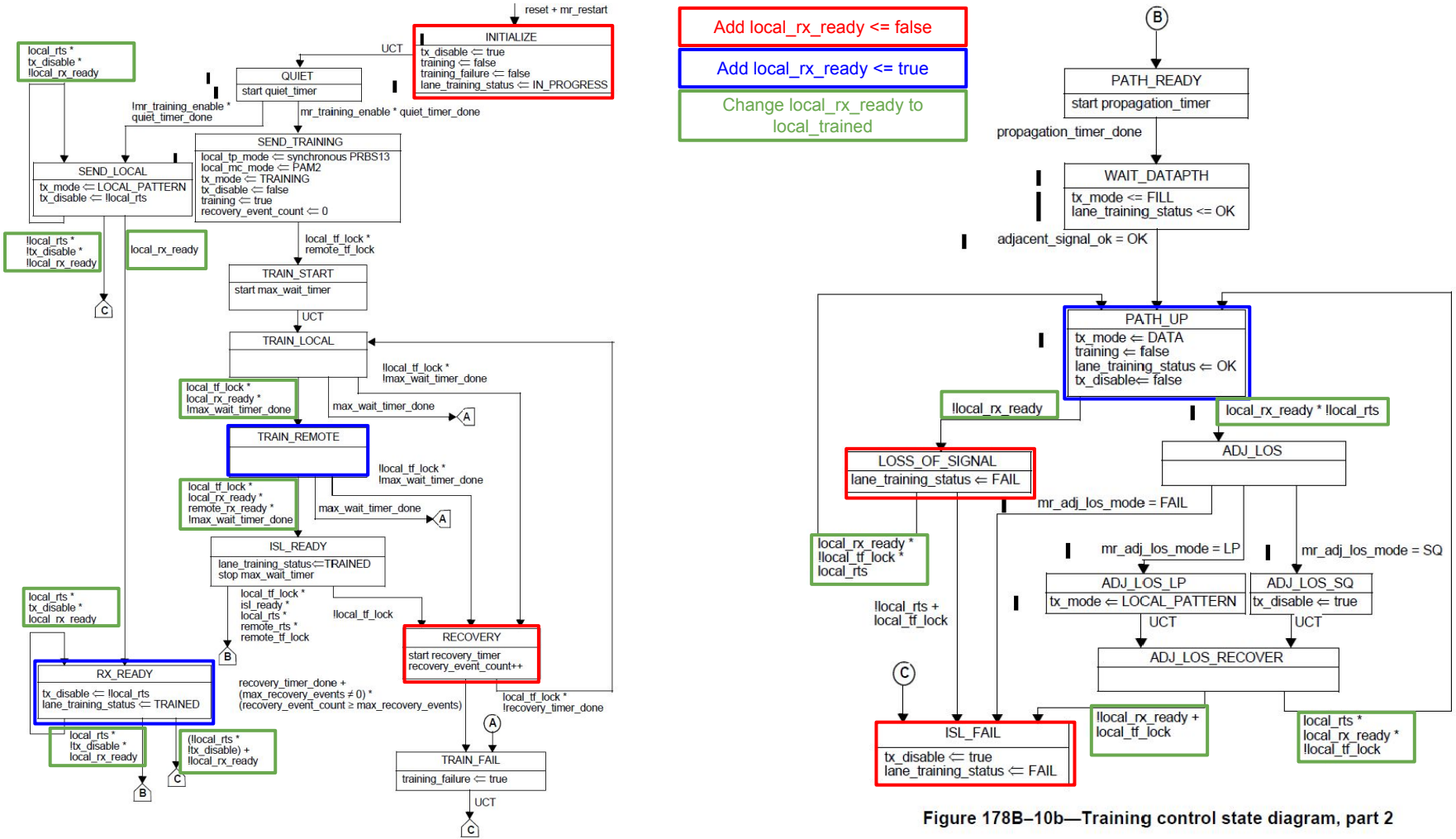

Figure 178B–10b—Training control state diagram, part 2

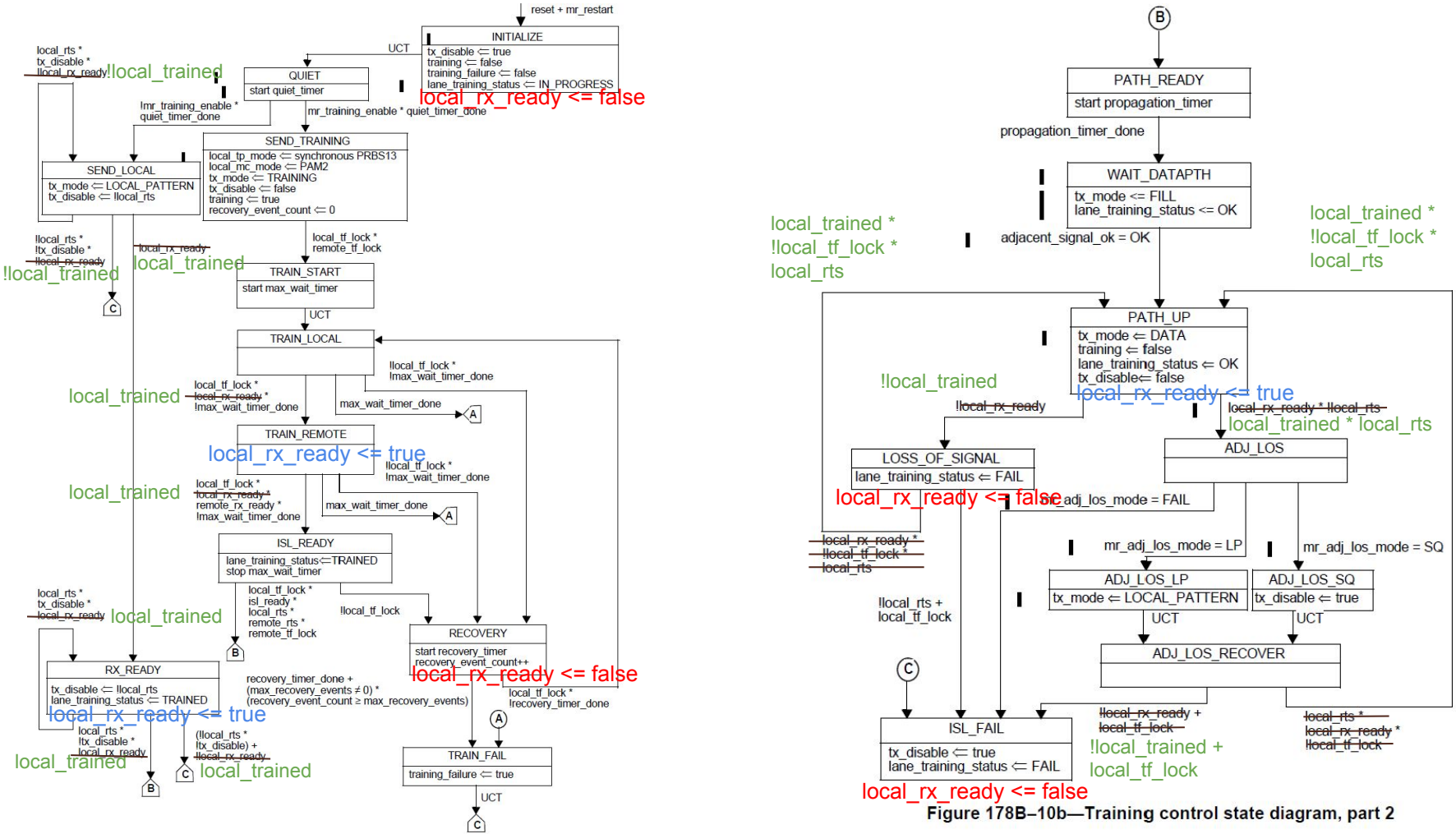

Figure 178B–10b—Training control state diagram, part 2