

802.3dj D3.0 Comment Resolution Common Topics

Matt Brown (Qualcomm), 802.3dj Chief Editor
Leon Bruckman (nVidia)
Jeff Slavick (Broadcom)

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 #29

CI 178B SC 178B.8.3.3 P 890 L 16 # [REDACTED]

Brown, Matthew Qualcomm

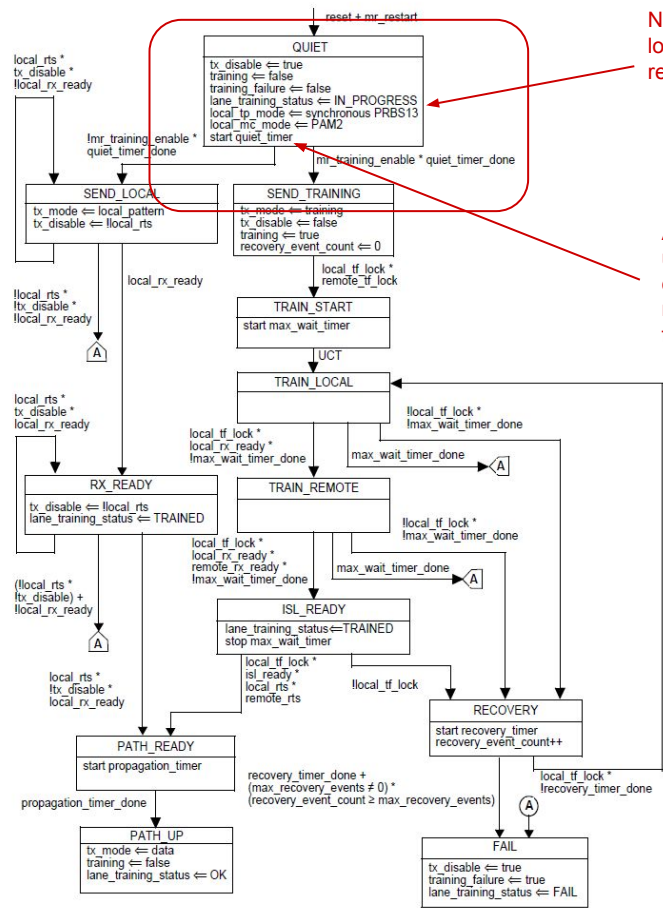
Comment Type T Comment Status X

The behavior of the quiet_timer is somewhat ambiguous. The state diagram in Figure xxx shows the quiet timer being started in "QUIET" state. If reset or mr_restart are true then this state is "entered" continuous until both are false. So does this mean the quiet_timer is constantly reset until both reset and mr_restart are false? In this case, the quiet_timer would only start counter after exiting the quiet state. Or is the intent that the timer starts counting right away and if it has expired the transition out of QUIET state can immediately occur.

Suggested Remedy

Add some clarity as to the intent. One option is to add an intermediate state that starts the quiet_timer when exiting the QUIET state and before entering the SEND_LOCAL or SEND_TRAINING states.

Proposed Response Response Status O



NOTE: local_mc_mode and local_tp_mode are not relevant to left-hand path.

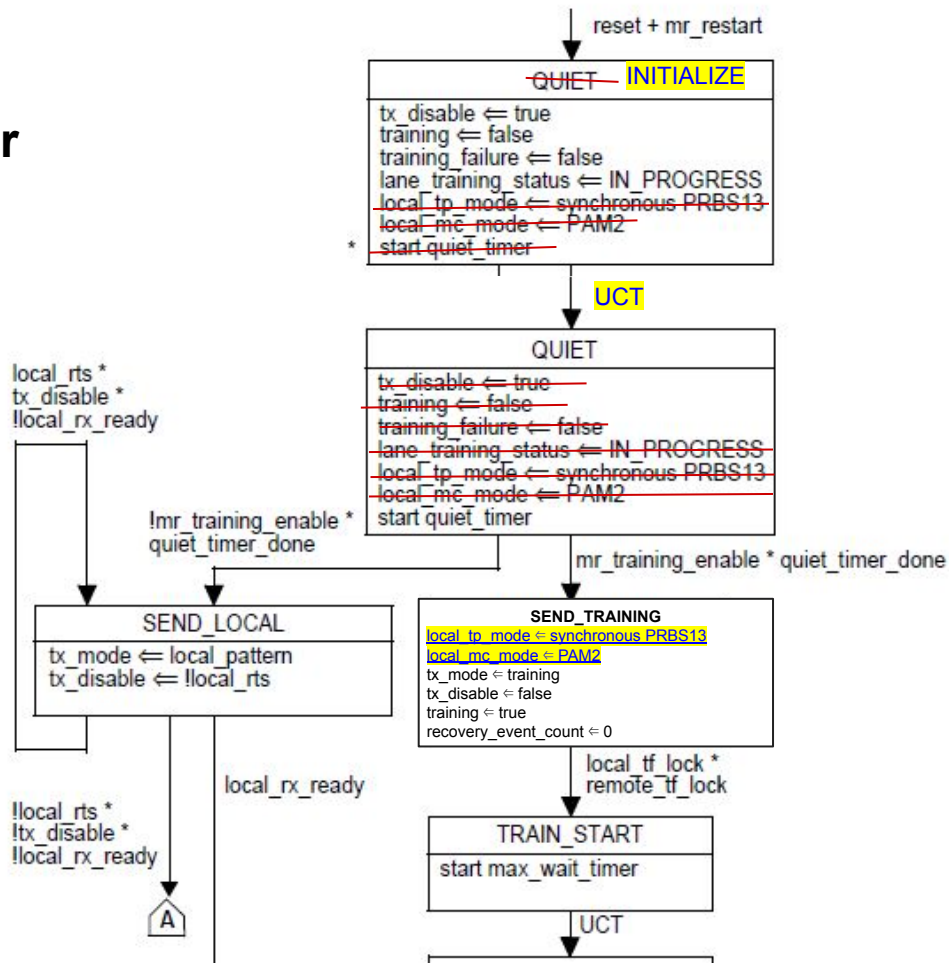
Ambiguous if timer starts upon entry to the state, or upon "reset + mr_restart" becoming false

Figure 178B-10—Training control state diagram

Comment #29

Training control quiet timer

Update Figure 178B-10 as shown to the right.



Comment #131

APSU description

CI 178B SC 178B.4 P 867 L 18 # I-131

Huber, Thomas

Nokia

Comment Type TR Comment Status X

The description of APSU in this bullet list needs to be rewritten. The variables local_rts and remote_rts do not propagate anywhere; they are state variables at an interface. RTS or IRTS is signaled between peer interfaces via a PMD-specific mechanism and signaled between adjacent sublayers via the service interface. An interface uses that signaling (plus other criteria) to determine the values of the local_rts and remote_rts state variables. Also, since we are extending APSU to coherent PMDs, the ILT function is not a requirement for APSU.

Suggested Remedy

Replace "APSU is the result of each ISL in the path transitioning to DATA mode using the RTS and ILT functions as follows:" and the subsequent bullet list with this:

"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 signals that it is ready to send to a peer interface via a PMD-specific or AUI-specific mechanism. 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"

Proposed Response

Response Status

Comment #131

APSU description – details of comment

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

Text in D3.0

- 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.

New text per suggested remedy of comment #131

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 signals that it is ready to send to a peer interface via a PMD-specific or AUI-specific mechanism. 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

Comment #131

APSU description –Proposed response to comment #131

Difference from suggested text in red

Change text as follows:

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 data, 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

Implement with editorial license

Comment #62

CI 178B SC 178B.5 P 869 L 5 # 62

WANG, Xuebo Huawei Technologies Co., Ltd

Comment Type **TR** Comment Status **D** Functions (CA)

In Figure 178B-4, RTS function provide the state of the `rts_status` variable from a reference interface to its adjacent interface via `SIGNAL_OK`. However, if we are talking about host devices, the reference interface has no adjacent interface. Under this case, in the receive direction, `SIGNAL_OK` will be sent to PCS or DTE xXS by `IS_SIGNAL.indication` primitive.

Suggested Remedy

Add a note to Figure 178B-4: If there is no adjacent interface, in the receive direction, `SIGNAL_OK` will be sent to PCS or DTE xXS via the `IS_SIGNAL.indication` primitive of PMA service interface.

Proposed Response Response Status **W**

PROPOSED REJECT.

The figure does show that `SIGNAL_OK` is a parameter of `IS_SIGNAL.indication`. See also section 178B.8.2.1 Variables `adjacent_signal_ok` and `adjacent_intf_rx_ready` and the note between them.

However, as noted by the comment in some cases there is no adjacent interface. In Figure 178B-4...

Change "SIGNAL_OK from adjacent interface"

to "SIGNAL_OK from the adjacent interface or from the sublayer above"

Change: "SIGNAL_OK to adjacent interface"

To: "SIGNAL_OK to the adjacent interface or to the sublayer above"

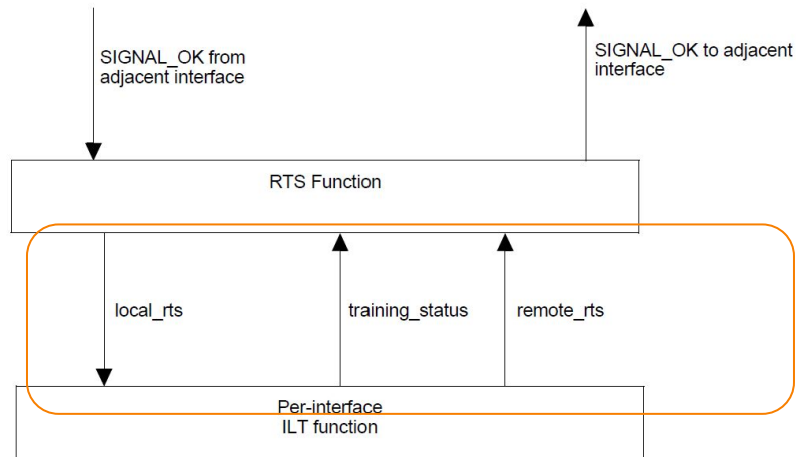


Figure 178B-4—Signaling between RTS and ILT functions

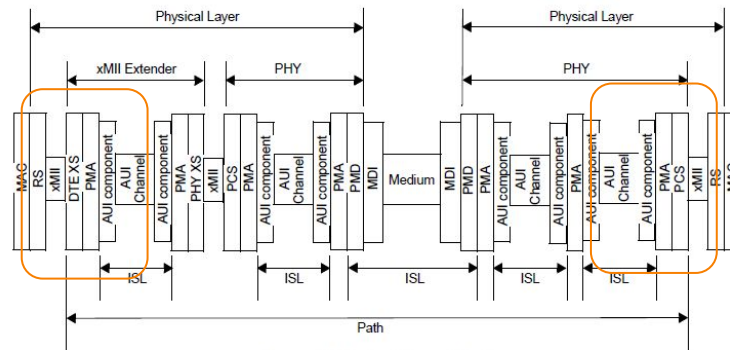


Figure 178B-1—ISL and Path