

Local Pattern in ILT (in support comment #463)

Jeff Slavick (Broadcom)

CI 178B	SC 178B.7.3.5	P 860	L 16	# 463
---------	---------------	-------	------	-------

Slavick, Jeff	Broadcom
---------------	----------

Comment Type	TR	Comment Status	D	State diagrams (CI)
--------------	----	----------------	---	---------------------

local_rts can't be a requirement to transition into SEND_LOCAL as it requires you to have local_rx_ready in order to assert local_rts which you can't have if you haven't started to send data, which you don't do until you enter SEND_LOCAL

SuggestedRemedy

Presentation to be supplied.

The passing of the state of “isl_ready & remote_rts” is lost

With the update to using a formal service interface of the state of the adjacent interface using signal_ok we've lost the details of the state of the partner interface. Which affects the setting of local_rts to true.

In the original implementation (D1.1) adjacent_remote_rts and adjacent_isl_ready were copies of remote_rts and isl_ready from the partner interface. So when the “Continue Training” bit went low and the local interface was done adapting it'd swap over to forwarded clock mode. Which requires 3 states of information (we only have 2):

IN_PROGRESS: Adapting the link on local clock

TRAINED: Done Adapting the link, can swap to mission clock

READY: Done Adapting link, on mission clock

Issue with RTS FSM

We can't identify when to swap to mission clock and has caused some dead-locks in the path-startup process.

RTS FSM controls of the local_rts variable which prevents the transition from the QUIET state to SEND_LOCAL in Figure 178B-10

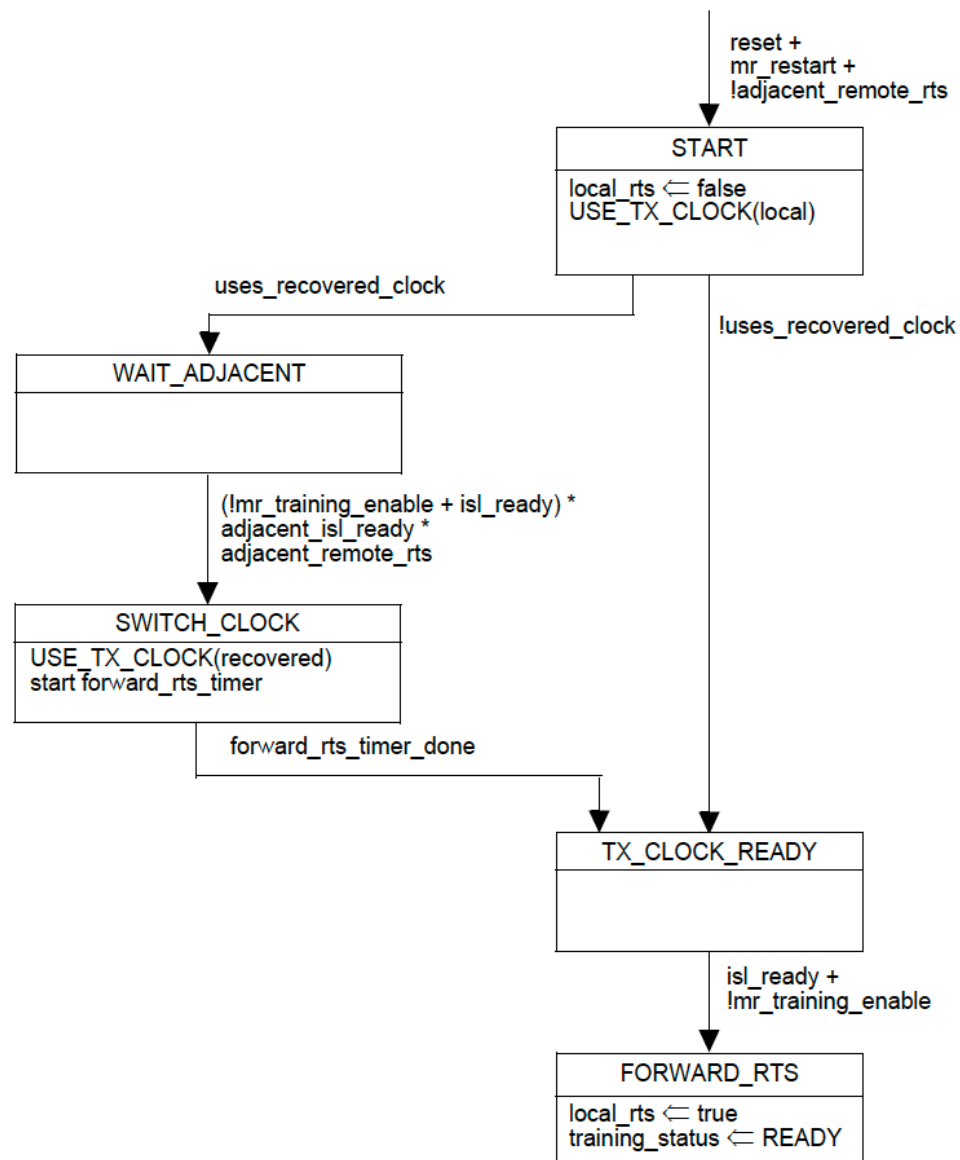


Figure 178B-9 – RTS update state diagram

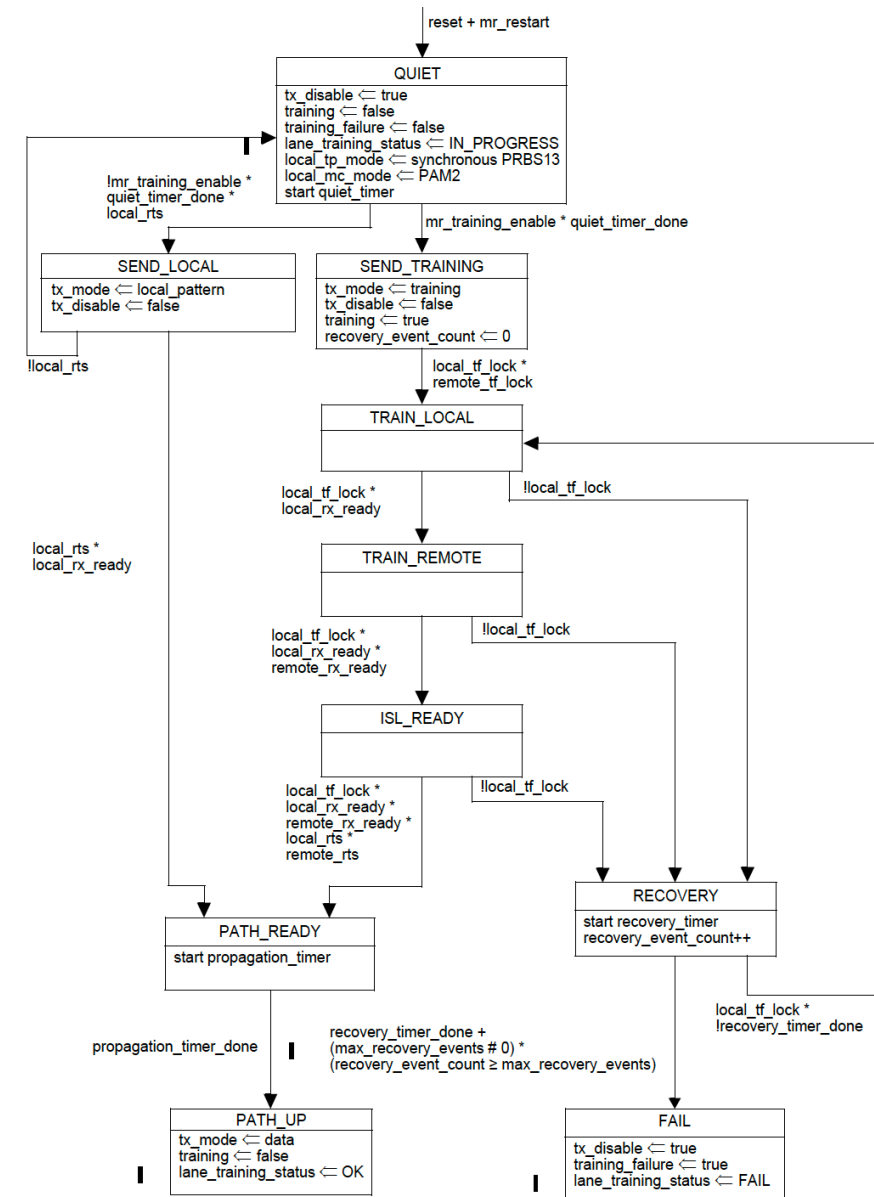
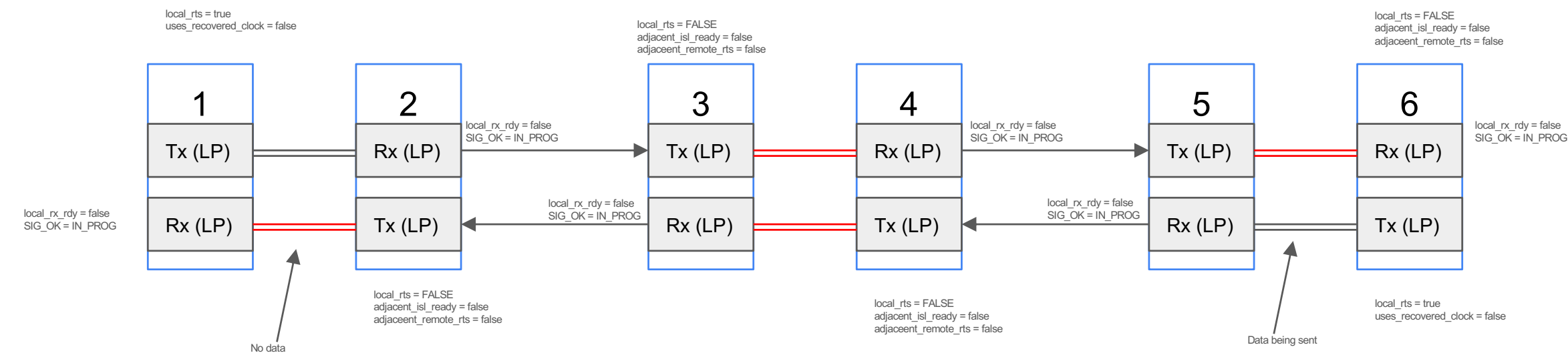


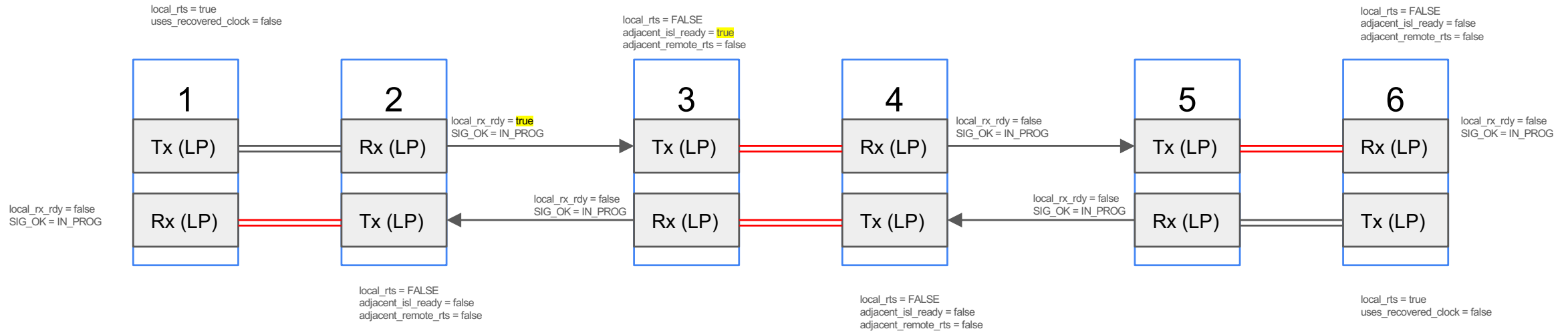
Figure 178B-10—Training control state diagram

ILT started on all ISLs with mr_training_enable = false



Interface 1 turns on Tx with local pattern.
Interface 6 turns on Tx with local pattern.
uses_recovered_clock = true for ALL interfaces except 1 & 6

ILT started on all ISLs with mr_training_enable = false



Interface 2 Rx is now happy

Interface 3 adjacent_isl_ready is now true

Interface 3 adjacent_remote_rts is false since SIG_OK != OK

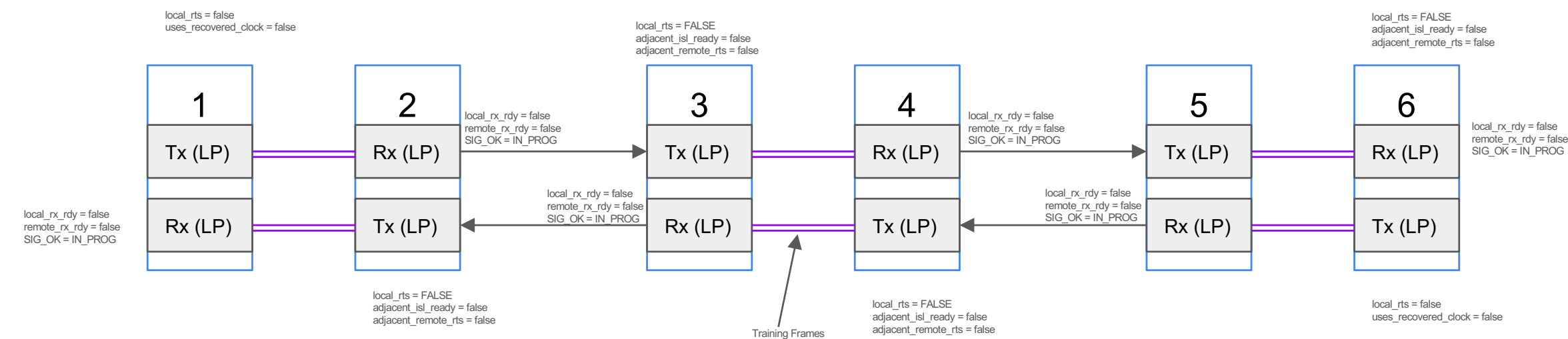
Interface 2 SIG_OK towards Interface 3 is still IN_PROG (local_rts is false)

Interface 3 Tx still disabled.

What about in Training mode

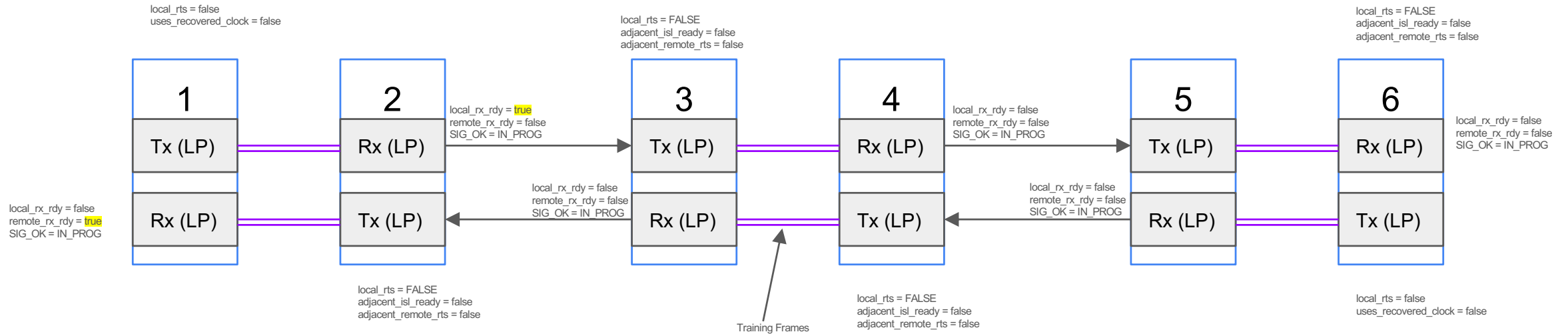
Similar issue exists for training frame mode

ILT: ALL interfaces with mr_training_enable = true



All begin to send frames upon reset removal

ILT: ALL interfaces with mr_training_enable = true



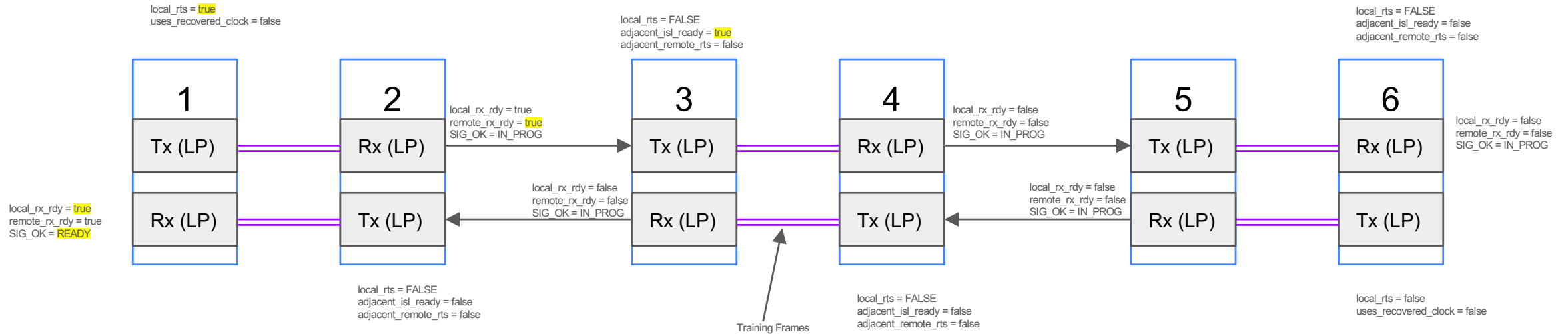
Rx2 finishes up

`local_rx_rdy = true` for Intf2

`remote_rx_rdy = true` for Intf1

`adjacent_isl_ready = false` for Intf3 since Intf2 `remote_rx_rdy` is false

ILT: ALL interfaces with mr_training_enable = true



Rx1 finishes up

local_rx_rdy = true for Intf1

remote_rx_rdy = true for Intf1

local_rts = true for Intf1

adjacent_isl_ready = false for Intf3 since SIG_OK is still IN_PROG

A band-aid fix to address local pattern issue.

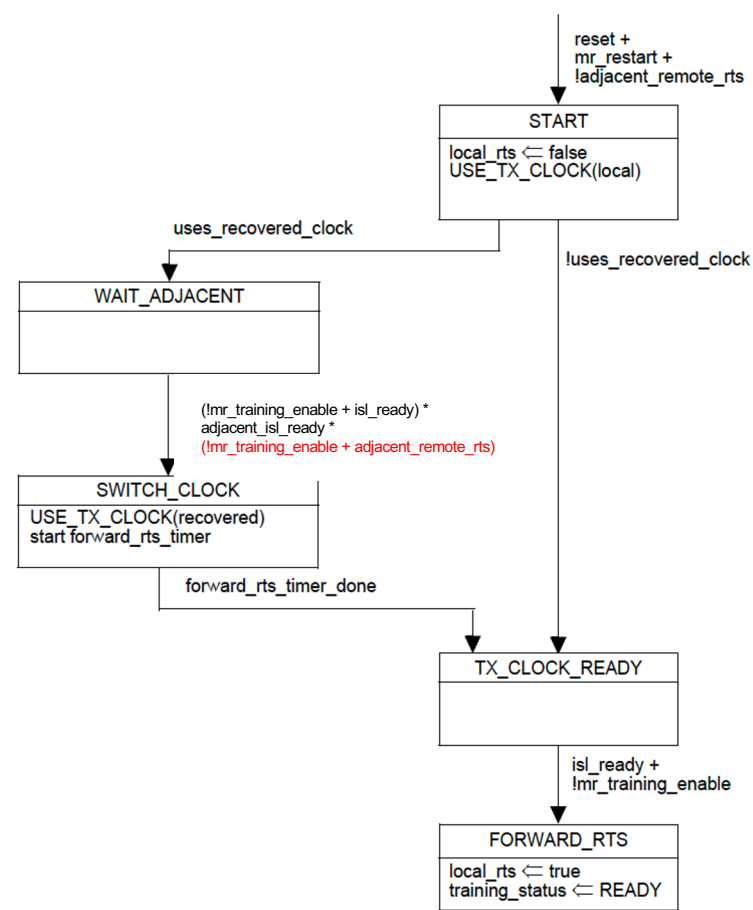
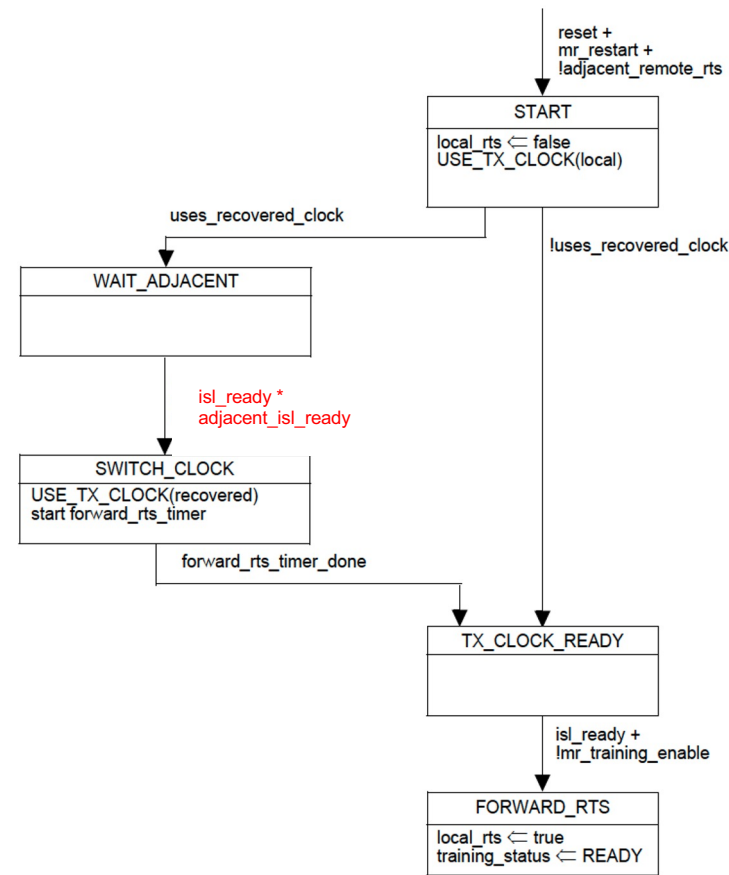


Figure 178B-9—RTS update state diagram

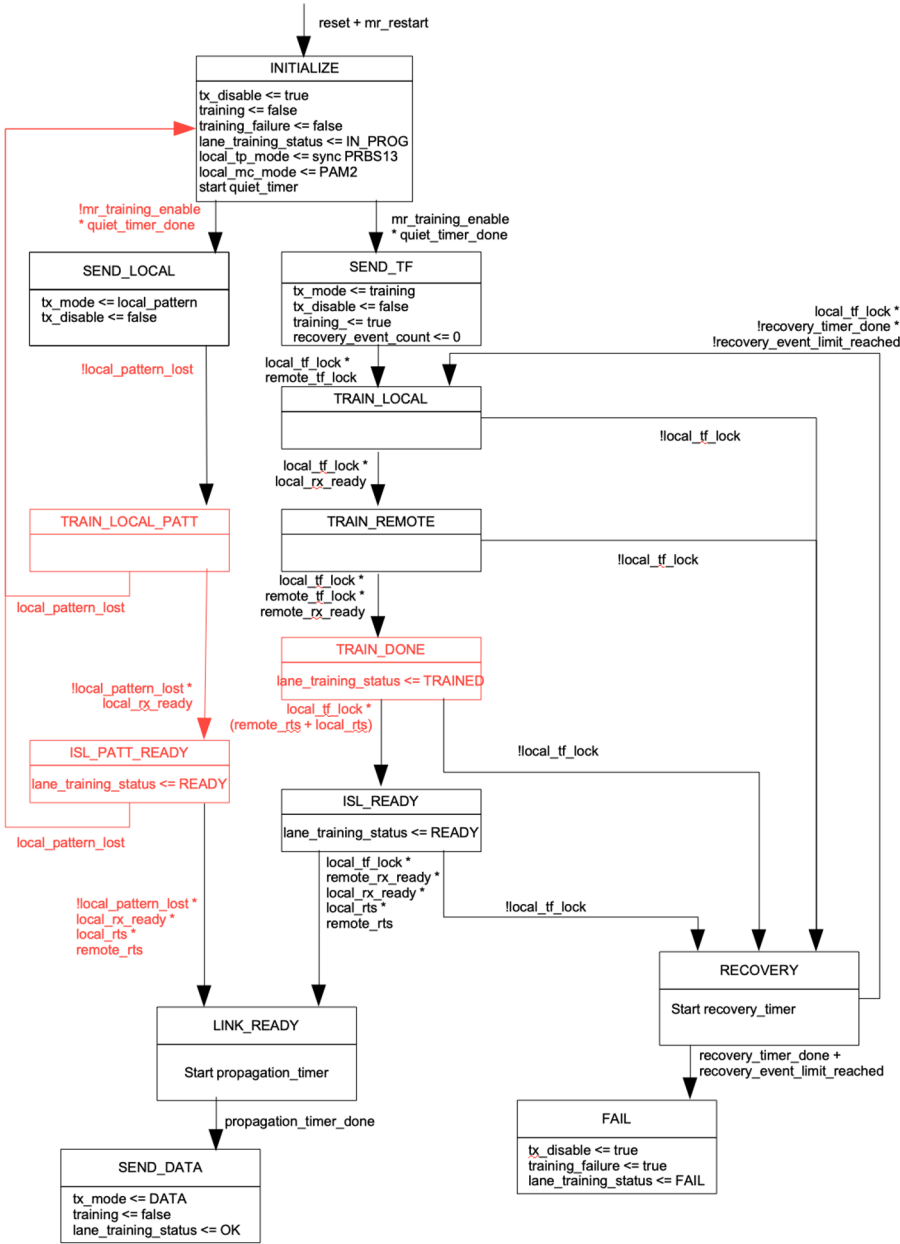
Larger update; add TRAINED state



Adjacent_isl_ready is signal_ok
= (TRAINED | READY | OK)

local_pattern_lost is a loose
pattern/signal_detect

Figure 178B-9—RTS update state diagram



Conclusion

Band-aid will cover the local pattern issue.

Still an issue with training frame format bring-up getting stuck.

The Larger update attempts to align the local pattern mode of bring-up with training frame mode. Along with providing method to distinguish between the various states of the link state. Which would break the training frame bring up issue.

Since the start-up process will get dead-locked we need to address it.