Fault Handling in the refactored APSU FSMs (comments #109,140)

Luz Osorio, Nokia
Tom Huber, Nokia
Marco Mascitto, Nokia

Supporters

- Jeff Maki, HPE
- Mike Dudek, Marvell
- Sebastien Gareau, Ciena
- Mike Sluyski, Cisco

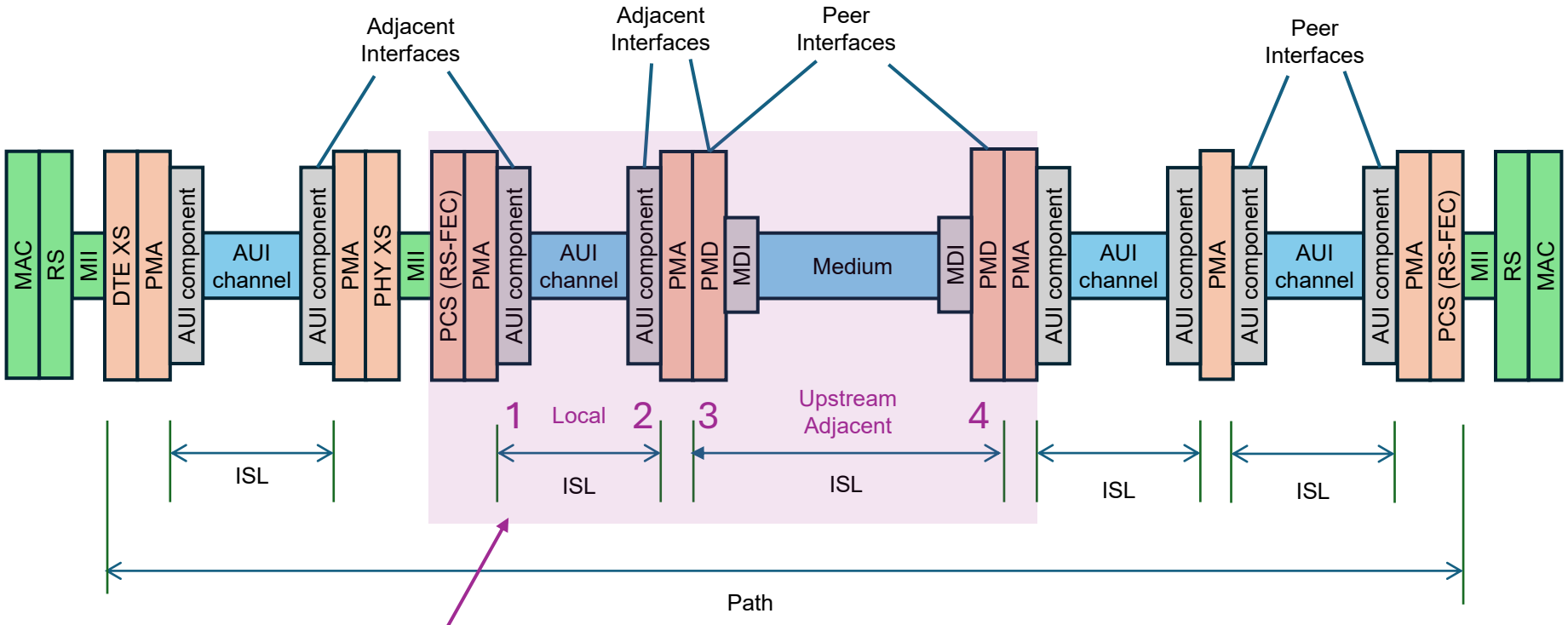
Introduction

- Current text does not have provisions to handle faults after the path switches to mission mode.
- This topic was address by ran_178b_01a_260422 in the context of existing FSMs.
- The handling of faults depends on the type of failure:
  - An ISL in a path is !RTS when either its interface or an upstream adjacent interface experiences a failure.
  - When the channel characteristics of an ISL do not change post recovery, the transmit EQs may not require changes, so training can complete quickly.
  - !RTS after mission mode data does not mean that training has failed. It is implied that training completed successfully to have reached mission mode data in the first place.
- The refactored FSMs simplify the specification of fault handling based on the type of failure as they separate the ILT and RTS FSM.
  - For example, failure of an upstream adjacent interface does not require retraining on the local ISL (if it is not disrupted) so the recovery for this fault is specified in the RTS FSM only.

Fault handling considerations

- The purpose of link training is to optimize an interface for performance (highest margin).
- When a signal disappears and reappears, link re-optimization is required to ensure best performance (highest margin)
  - The transition from LOS to !LOS is the same whether it occurs at initial setup or after the path has transitioned to mission mode data, so recovery should also be the same.
- When a signal experiences a fault, depending on its nature, it can cause changes in the signal response after recovery, or it can return with the exact characteristics prior to the failure. It is impossible to know this ahead of time.
- To minimize risk of link performance issues (possibly leading to long term link degradations or subsequent failures) care must be taken on how the recovery is completed.
- While fiber failures are more frequent, faults can also happen in internal ISLs.

APSU Path, ISLs and Interfaces

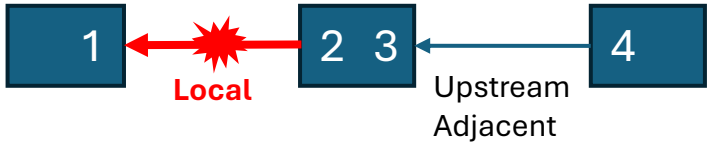


This presentation centers on a local and upstream adjacent interface for demonstration purposes.

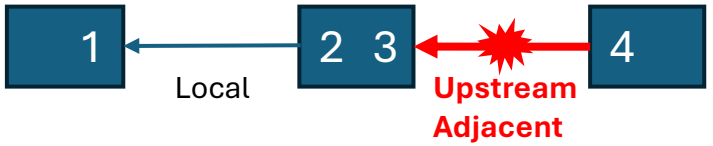
Failure types

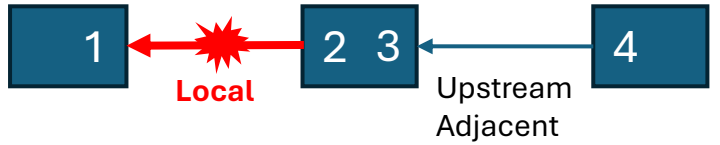
We can anticipate two failures types:

1. The local ISL has lost its signal (LOS) – Fault 2->1:
   - Local ISL (1) is !RTS because there is no established signal.
   - Local ISL (1) is !isl_ready because there is no signal.
   - Much like ISL initialization, the local ISL will require link training (if supported) to complete before going to RTS.



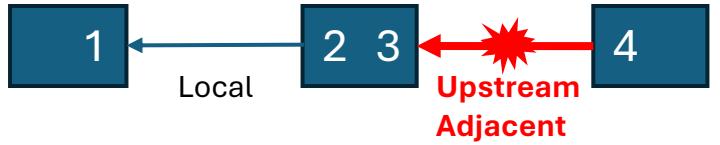
2. The adjacent ISL has lost its signal (!adjacent_intf_rx_ready) – Fault 4->3:
   - Local ISL (1) is !RTS because the adjacent ISL is not ready.
   - Local ISL (1) is isl_ready because its signal has not been disrupted.
   - Once the adjacent ISL is recovered and reaches RTS, the local ISL can transition back to RTS without training.





Local ISL signal failure recovery

- **Scenario 1**: The EQs on either side of the link are still optimal post recovery and do not require further adjustment. There is no way to know this until confirmed via the ILT function.
- **Scenario 2**: The EQs need to be adjusted post recovery to achieve best performance. There is no way to know this until confirmed via the ILT function.
- **Mandatory solution: retraining must be completed to ensure best link performance.**
- Restarting training from scratch penalizes both scenarios because in the first case no adjustments were necessary, and in the second case starting from scratch is also unnecessary.
- Not training will carry risk for scenario 2 – no optimal performance, link degradation over time (?), subsequent link failures (?)
- **Optional time-efficient solution**: when training has completed, retain the EQ values.
  - Scenario 1: Send ILT frames with saved values. Training will be completed “immediately”.
  - Scenario 2: Send ILT frames with saved values. Adjustments will be made and one can expect training will be completed faster.
  - **In both scenarios, an optimal link is guaranteed post recovery.**
- If the implementation does not save values, and starts training from scratch, recovery will take longer but the link will still reach the optimal performance



Adjacent ISL signal failure recovery

- Local ISL was not the source of the failure.
- **Scenario 1:** The components (2) have auto-squelch enabled which means the peer in the local ISL (1) experiences an LOS => Local ISL signal failure (LOS). Recovery per previous slide.
- **Scenario 2:** The components (2) generate a replacement signal. The peer in the local ISL will receive the replacement signal. Local ISL (1) does not experience a signal disruption (isl_ready remains true). Clock changes are experienced which means the local ISL is !RTS.
- Both scenarios must be handled in the standard because neither scenario (squelch/no-squelch) is mandatory for current AUIs or PMDs, and it is application or implementation dependent.
- **Recovery**:
  - Scenario 1 is handled per the previous slide.
  - Scenario 2 is handled via the refactored RTS FSM which controls the tx_mode function. Once the adjacent interface is recovered (adjacent_intf_rx_ready), the clock will switch back to mission mode, and the local ISL can go back to RTS.
    - For IMDD interfaces, it means the replacement signal must be ILT frames to transmit RTS.
    - For coherent PMDs, the replacement signal is either local pattern or OH bits (as specified in the applicable clause).
    - For future = which ever method is used for RTS signaling.

Proposal to save trained coefficients

- Currently annex 178B defines the initial condition settings to be the value of “initialize” per Table 179-8.
  - When “initialize” is the starting point, receivers will most likely request changes when the ILT function is executed, leading to iterative cycles to reach an optimal performance.
- Some applications prefer to reduce the recovery time in the event of a failure. Saving trained coefficient values can reduce recovery time.
- If previously trained values are saved in a preset, the receiver can request this preset as a first step.
  - This presentation proposes preset 7 for saved trained coefficients.
  - In the NEW_IC state of Figure 178B-12, ic_sel = preset 7.

Proposal to save trained coefficients

Proposed changes in red

ic_req

Enumerated variable that holds the initial condition setting to be applied to the local transmitter. It is assigned any of the values assigned to ic_sel, or the ~~value~~ initialize setting in Table 179-8. The value of ic_req is set by the coefficient update state diagram (Figure 178B–12).

ic_sel

Enumerated variable derived from the “initial condition request” bits from the control field of the received training frames (see 178B.7.4). This variable is assigned one of the following values (abbreviations used by the state diagram are included in parentheses): individual control (ind_ctl), preset 1, preset 2, preset 3, preset 4, preset 5, preset 6, preset 7.

Saving training values as preset

Proposed changes in red

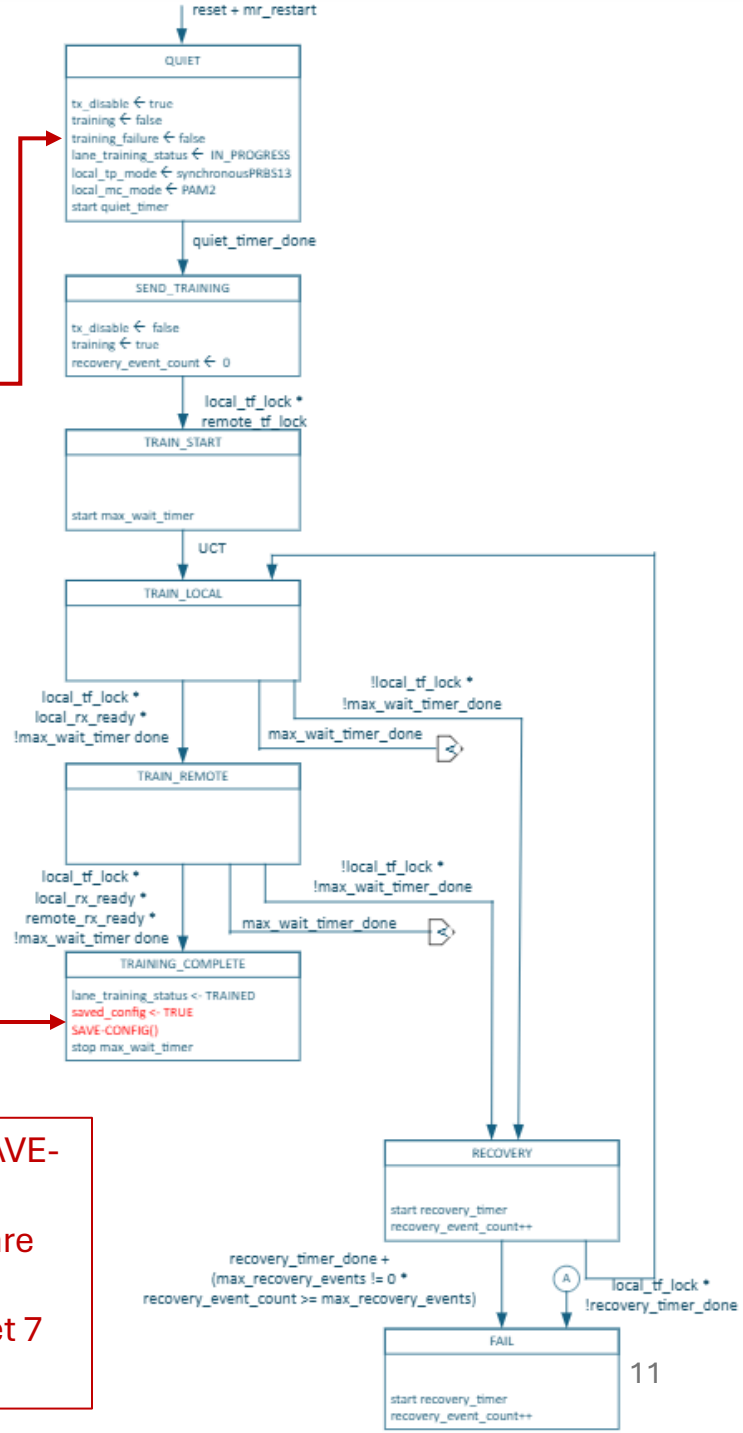
Note, the starting lp and mc variable values are the same as initialization. cycles to change these are negligible to the total recovery time

Table 178B–2—Control field structure for the E1 format

Bit(s)	Name	Description
15:14	Reserved	Transmit as 0, ignore on receipt
13:11	Initial condition request	13 12 11 1 1 1 = Reserved 1 0 1 = Preset 6 0 1 1 = Preset 5 0 0 1 = Preset 4 1 1 0 = Preset 3 1 0 0 = Preset 2 0 1 0 = Preset 1 0 0 0 = Individual coefficient control
10	Continue training	1 = Continue training 0 = Switch to data when training is completed
9:8	Modulation and precoding request	9 8 1 1 = PAM4 with precoding 1 0 = PAM4 without precoding 0 1 = Reserved 0 0 = PAM2
7	Reserved	Transmit as 0, ignore on receipt
6:5	Training pattern request	6 5 1 1 = free-running PRBS31 1 0 = Reserved 0 1 = free-running PRBS13 0 0 = synchronous PRBS13
4:2	Coefficient select	4 3 2 1 0 0 = Reserved 1 0 1 = $c(-3)$ 1 1 0 = $c(-2)$ 1 1 1 = $c(-1)$ 0 0 0 = $c(0)$ 0 0 1 = $c(1)$ 0 1 x = Reserved
1:0	Coefficient request	1 0 1 1 = No equalization 1 0 = Decrement 0 1 = Increment 0 0 = Hold

Preset 7

Proposed change to Table 178B-2: Change 13:11: 111b from Reserved to Preset 7 for saved trained coefficients



Proposed change to 178B-10 (ILT FSM): Add SAVE-CONFIG() function to TRAINING_COMPLETE.
Once training is complete, coefficient values are saved as preset 7.
Set saved_config = TRUE to indicate that Preset 7 has been updated from “initialize”.

Proposed new variables and functions

Preset 7

Predefined transmitter equalizer configuration. It is set to the initialize setting per Table 179-8 at the outset (before any training has been completed on the ISL). It is updated to the current coefficient values by the SAVED-CONFIG(k) function.

saved_config

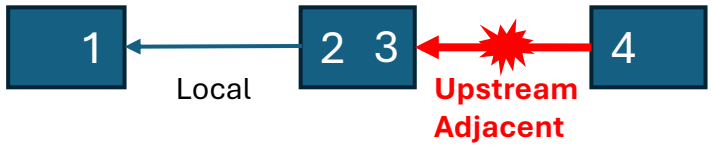
Boolean variable indicating whether previous trained values have been saved in preset 7. Indicates that preset 7 has coefficients values other than initialize settings per Table 179-8

SAVED-CONFIG(k)

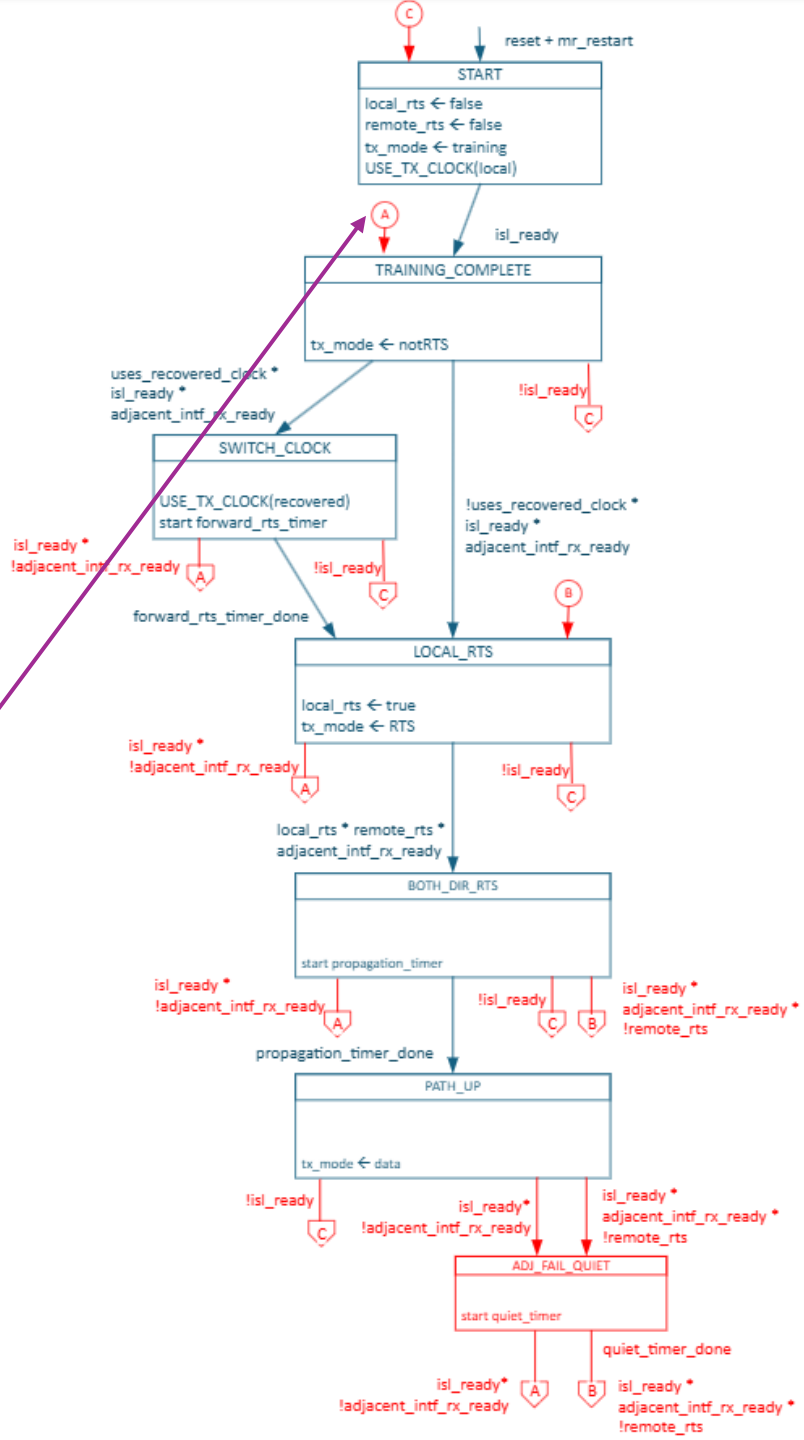
Function that saves the value of c(k) as preset 7 when entering the TRAINING_COMPLETE state. Preset 7 can be requested by the receiver using ic_sel to reduce training time during path recovery.

Failure handling in RTS FSM

Proposed changes in red

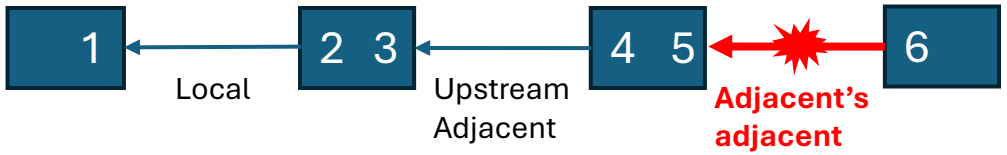


A. Adjacent ISL failure with replacement signal
   - Local signal does not need to be retrained because it was not disrupted (isl_ready)
   - Local signal is !RTS because !adjacent_intf_rx_ready.
   - Local interface detects replacement signal (1), after a quiet period the transmitter (2) switches to tx_mode = notRTS.
   - Wait until adjacent_intf_rx_ready to switch clock and go back to RTS.



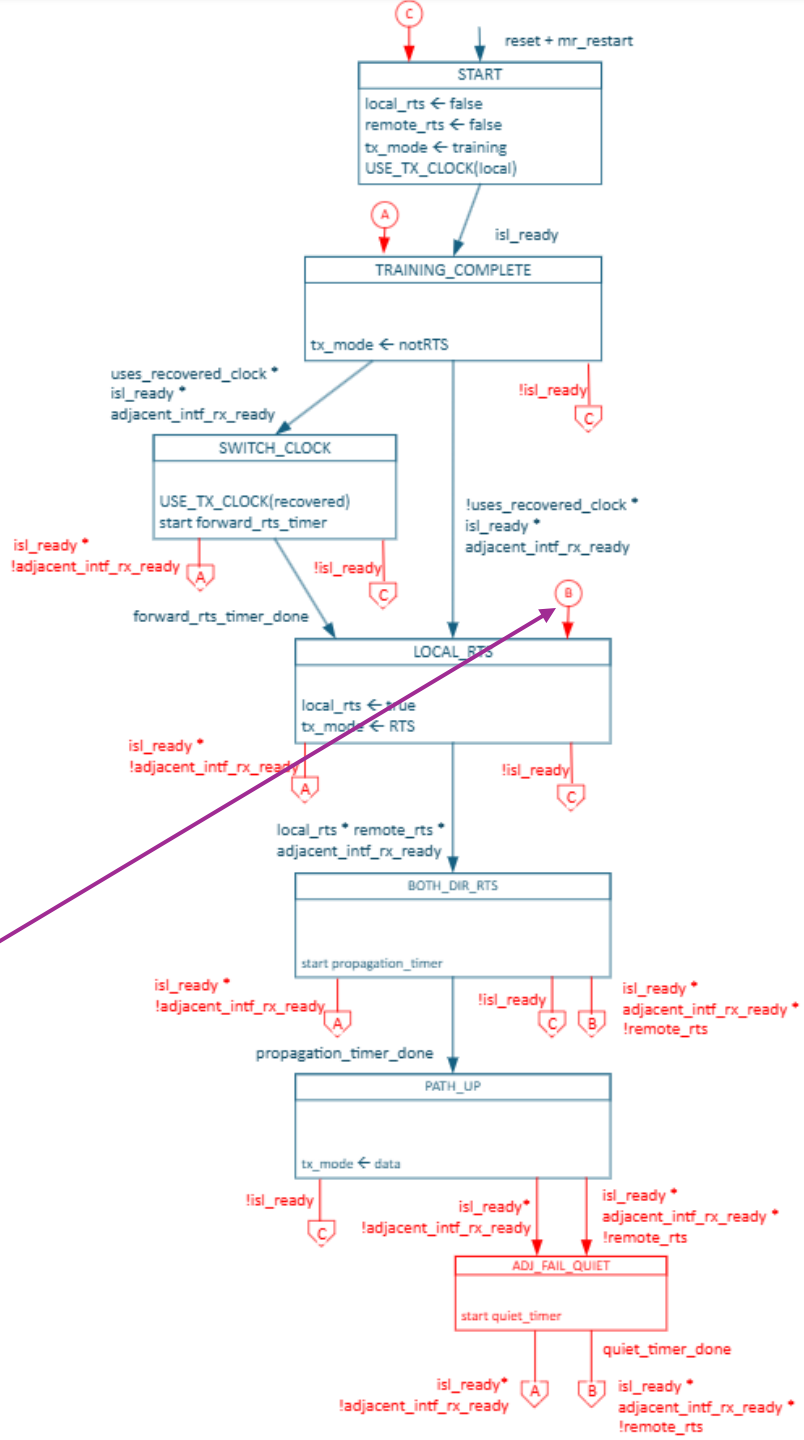
Failure handling in RTS FSM

Proposed changes in red



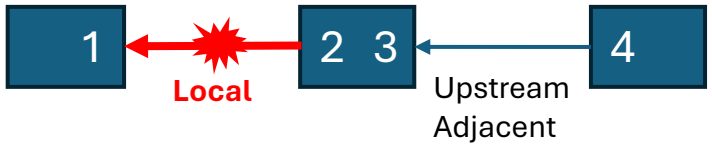
B. Adjacent's Adjacent ISL failure with replacement signal

- Local signal does not need to be retrained because it was not disrupted (isl_ready)
- Adjacent signal is ready because it was not disrupted (adjacent_intf_rx_ready)
- Local interface detects replacement signal (1), after a quiet period the transmitter (2) switches to tx_mode = RTS.
- Local signal is RTS but receives !remote_rts. Wait until remote_rts.



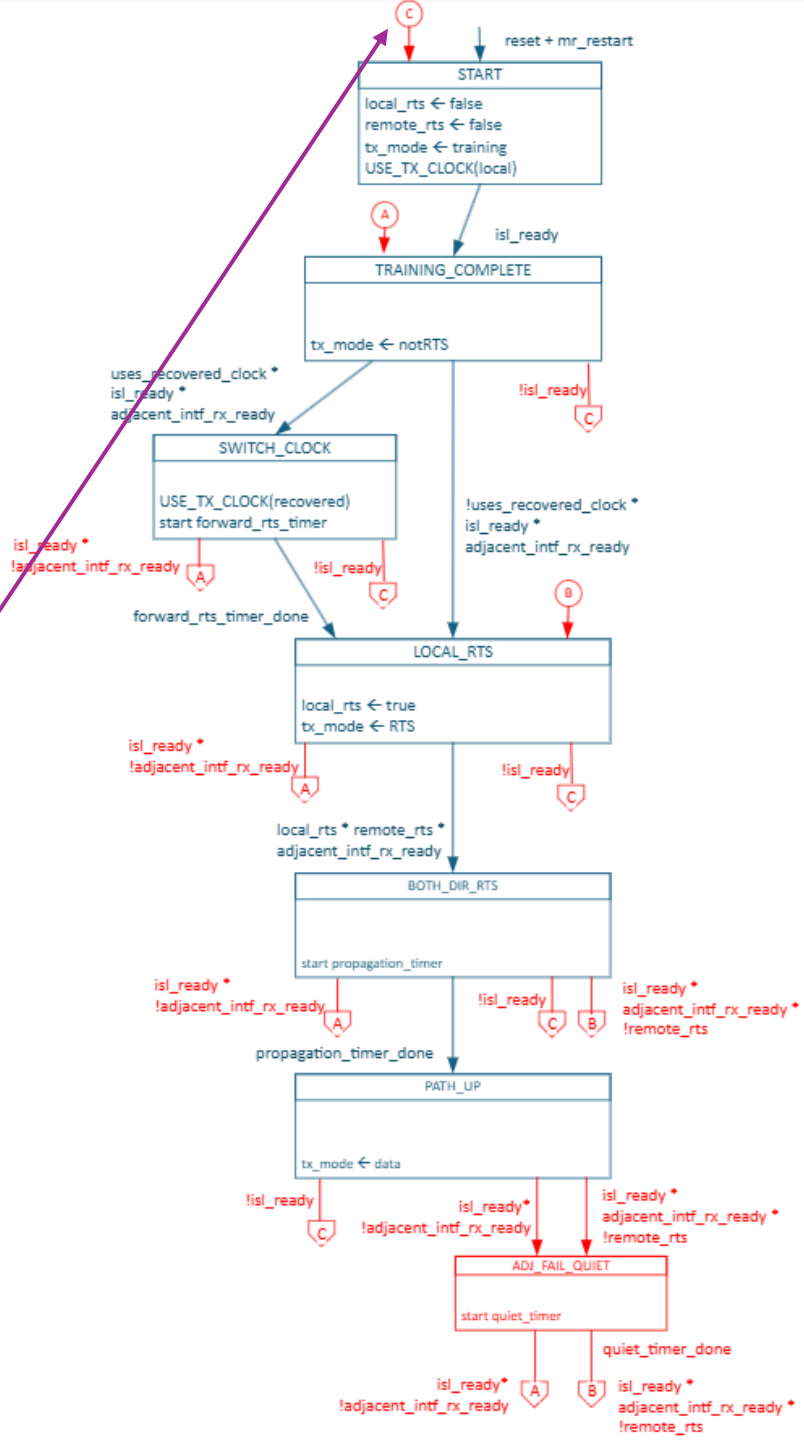
Failure handling in RTS FSM

Proposed changes in red



C. Local ISL signal failure

- Local signal is disrupted to !isl_ready
- Upon recovery, local signal must be trained to guarantee optimal performance -> tx_mode = Training



Differences from ran_178b_01a_260422

1. Save trained coefficient function added to speed up recovery while achieving optimal link performance.
2. An ISL shall go through training (link performance optimization) after it recovers from LOS but can start from saved values.
3. Each ISL autonomously decides whether to squelch or use replacement signal (implementation dependent) rather than through a management variable.
   - Annex 178B does not define the behavior during a failure (consequential actions) but rather specifies how to restore optimal performance once a failure has been cleared.

Summary

- The suggested changes will guarantee optimal performance after failure recovery, while reducing recovery time when saved_config is utilized.
- These modifications allow application-specific consequential actions, while specifying recovery methods to optimize link performance.
- The use of refactored FSMs clarify failure handling within the specification, and the separation of ILT and RTS functions facilitates distinguishing failure recovery based on type.

Thank you!