

# APSU behavior upon loss of signal

Adee Ran, Cisco

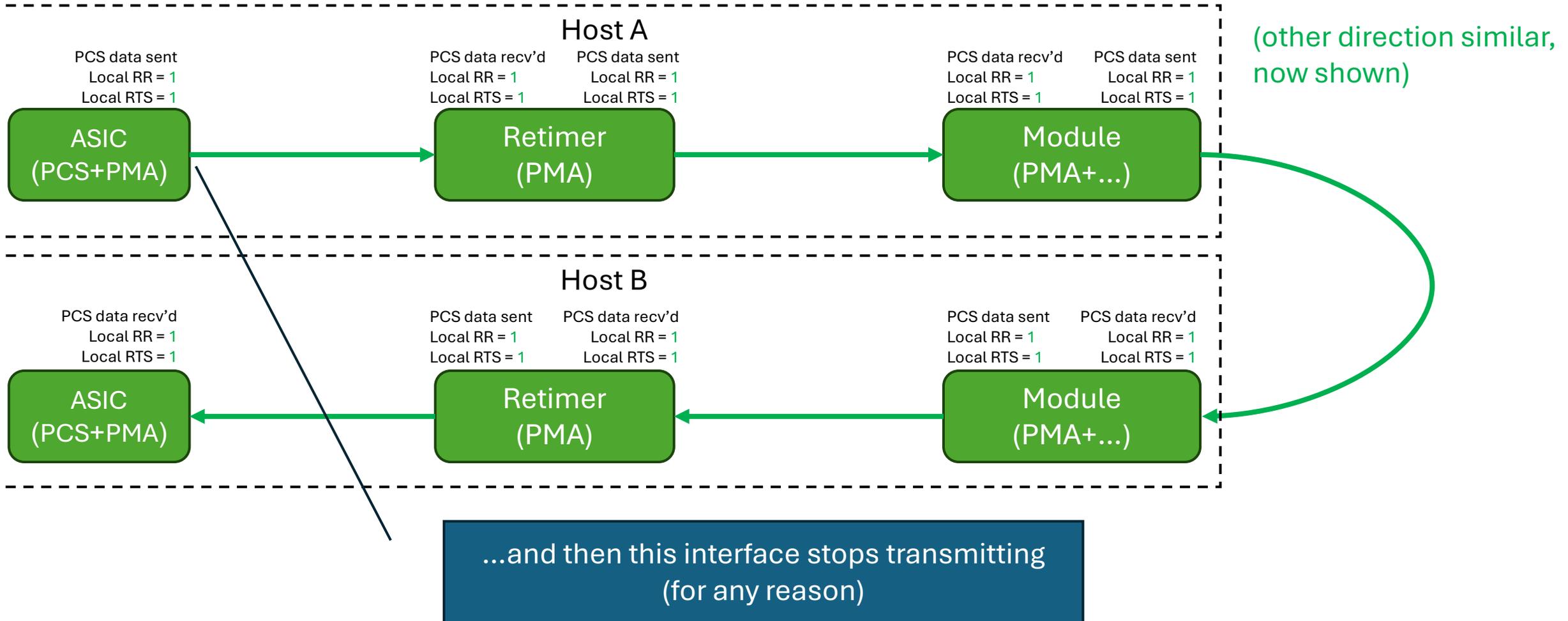
# Supporters

# Background and goal

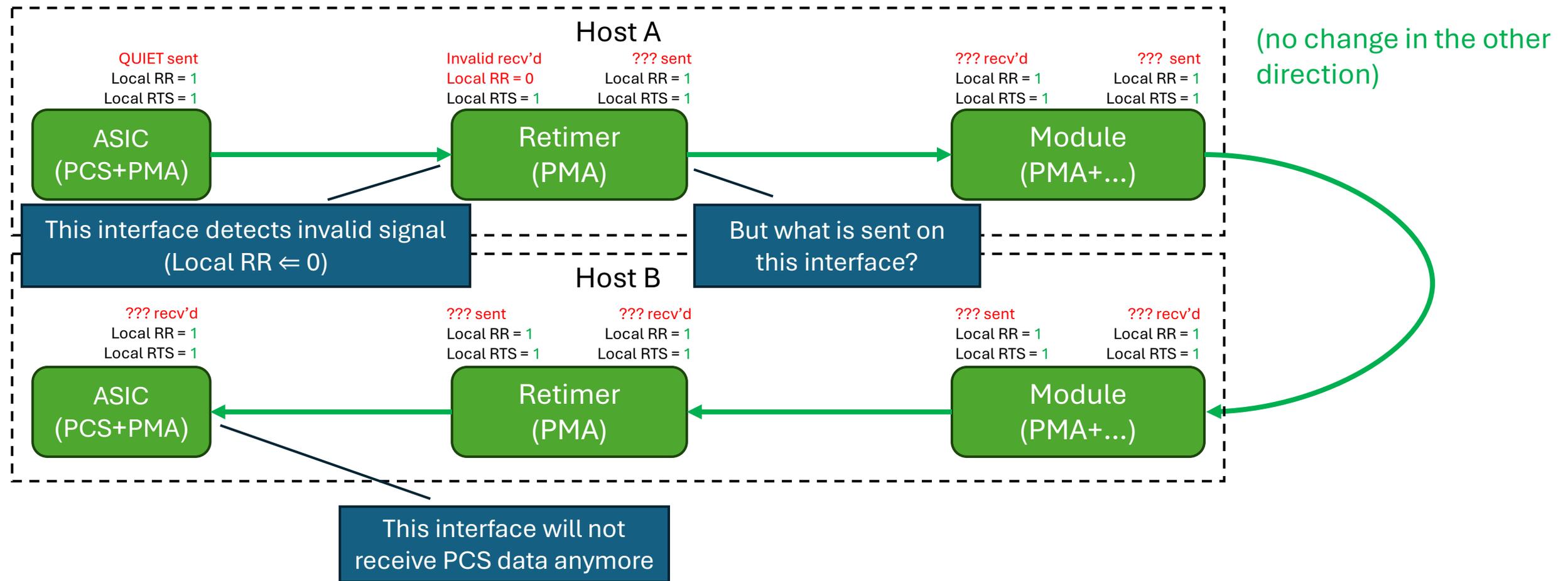
- Annex 178B defines requirements from PMDs and AUIs to enable autonomous path startup (APSU)
  - ILT function
  - RTS function
- The status of the RTS function is conveyed through the service interface parameter SIGNAL\_OK (IS\_SIGNAL.request/indication)
  - SIGNAL\_OK also indicates validity of the IS\_UNITDATA parameters (rx\_symbol or tx\_symbol)
- In recent discussions, the path behavior when APSU is restarted on one device was questioned
  - Also, what happens when an interface loses its input signal, for any reason
  - Should APSU be restarted? When?
- This presentation explores these questions and suggests some changes



# Consider a path that is fully “up”



# Chain of events with current specifications



# Open questions

- What do we expect a retimer to send in one interface with invalid input on the adjacent interface?
- Do we expect re-training? On which ISLs?
- Do we want re-training to be started automatically?

Not answered in D3.0!

# D3.0 content

## PMDs

### 179.4 Service interfaces (other PMDs are similar)

The SIGNAL\_OK parameter of the PMD:IS\_SIGNAL.indication primitive corresponds to the variable rts\_status of the ready-to-send (RTS) function, as defined in 178B.8.2.1. When SIGNAL\_OK is either IN\_PROGRESS or FAIL, the rx\_symbol parameters of PMD:IS\_UNITDATA\_i.indication on all lanes are unspecified.

## AUIs

### 176D.4 Service interfaces (176C.4 is similar)

The SIGNAL\_OK parameter of the PMA:IS\_SIGNAL.indication (for an AUI component above the AUI channel) or PMA:IS\_SIGNAL.request (for an AUI component below the AUI channel) corresponds to the variable rts\_status of the RTS function, as defined in 178B.8.2.1. When SIGNAL\_OK is either IN\_PROGRESS or FAIL, the corresponding tx\_symbol parameters on all lanes are unspecified.

# Expected behavior with loss of signal (LOS)

- Assuming loss of input signal on an interface can be detected – two options seem possible for the adjacent interface:
  - A. Squelch the output signal (CMIS “auto squelch”)
  - B. Send a locally generated replacement signal (assuming there is still a local clock with the same frequency as the data-mode clock)
- Note that restarting the ILT state diagram results in “A then B”
- Both options require informing the adjacent interface that there is no valid data
  - This status is communicated by SIGNAL\_OK which corresponds to rts\_status
  - Therefore rts\_status should change when the signal becomes invalid

# How rts\_status is derived (D3.0)

## 178B.8.2.1 Variables

### rts\_status

Enumerated variable that indicates the status of the RTS function. This variable is assigned one of the following values as follows:

FAIL: training\_status is FAIL

OK: training\_status is OK

READY: remote\_rts is true and training\_status is READY

IN\_PROGRESS: training\_status is IN\_PROGRESS or remote\_rts is false and training\_status is READY

### training\_status

Enumerated variable that indicates the status of the ILT function. This variable is assigned one of the following values as follows:

FAIL: Any lane\_training\_status is FAIL

OK: All lane\_training\_status are OK

READY: All lanes have lane\_training\_status of TRAINED or OK with at least one lane being TRAINED

IN\_PROGRESS: At least one lane has lane\_training\_status of IN\_PROGRESS and no lanes have lane\_training\_status of FAIL

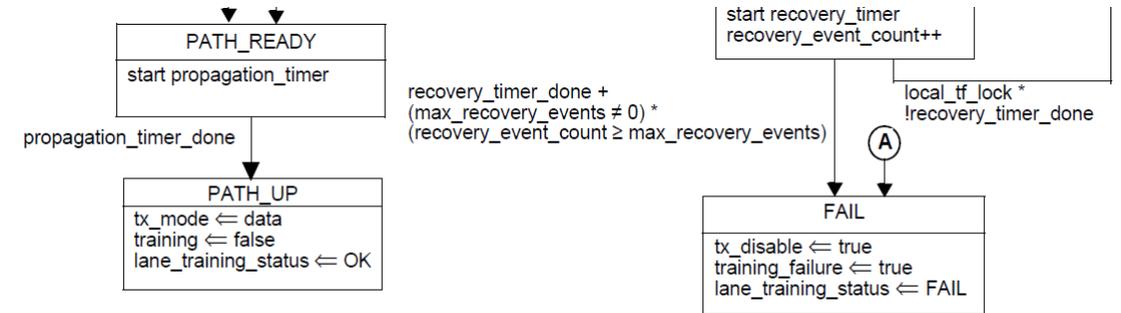


Figure 178B-10—Training control state diagram

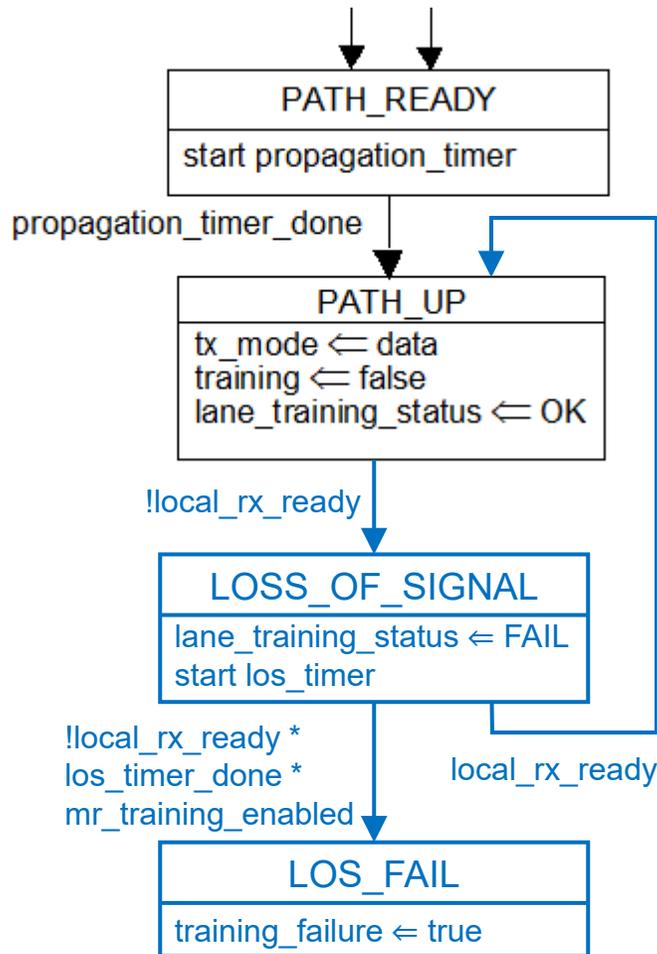
Lane\_training\_status is set in PATH\_UP  
This is a terminal state!  
status will stay OK until mr\_restart

➔ We need to add some way to change lane\_training\_status to FAIL on an active link

# Do we expect re-training? Automatically?

- Restarting training is “expensive” in terms of downtime
  - If there is a short disruption – the devices may be able to recover
  - But if an ISL is down for a long time (media disconnect, module replacement, etc.) then it’s a clean way to restart
- Note that if training is disabled, `mr_restart_training` still has an effect
  - If the trained parameters are retained – recovery may be faster than with training enabled
- Preferably, the timing should be left open for implementation choice
  - That is, management rather than by state diagram, timers, etc.
- The behavior needs to be defined:
  - During a temporary disruption event
  - During a training-restart event

# Missing piece: handling signal disruptions

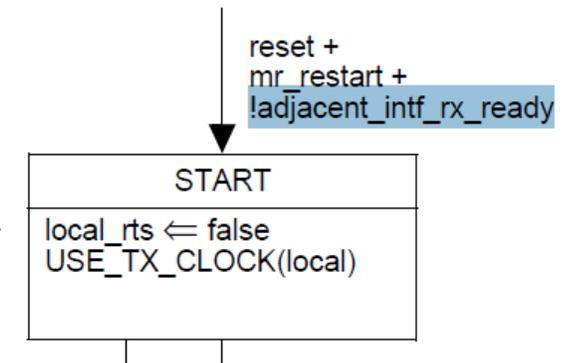


To handle disruptions, a new state (LOSS\_OF\_SIGNAL) can be added to the training control state diagram.

Entering this state (in any of the lanes) will cause:

1. **lane\_training\_status** ← FAIL
2. **training\_status** (per-interface aggregate of lane\_training\_status) ← FAIL.
3. **rts\_status** ← FAIL (per definition).
4. **SIGNAL\_OK** parameter of **inst:IS\_SIGNAL.(indication/request)** is FAIL (reflects rts\_status).
5. This gets communicated to the adjacent interface via **inst:IS\_SIGNAL.(request/indication)**.
6. **adjacent\_signal\_ok** ← FAIL of the adjacent interface (per definition).
7. RTS update state diagram of the adjacent interface transitions to START, sets **local\_rts** ← false and uses local clock (see below).

Top of Figure 178B-9  
(no change needed)



Proposed addition to Figure 178B-10

# Missing piece: handling signal disruptions (2)

- What happens to the path during disruption?
  - The output of the adjacent interface is undefined. This may mean:
    - Squelched output, which propagates the disruption to the next ISL.
    - Locally generated pattern with local clock.
  - Even with a locally generated pattern, the receiving PCS (and possibly other sublayers in the path) may lose AM lock after some time.
    - This should not automatically cause restart of training!
  - If the disruption is temporary, when it ends (`local_rx_ready`) a similar chain of events will set **training\_status**, **rts\_status**, and **adjacent\_signal\_ok** back to OK, and the adjacent interface can regain **local\_rts**  $\leftarrow$  **true**.
- When should training be restarted after disruption?
  - If the disruption persists long enough, the state will transition to `LOS_FAIL` and assert `training_failure`. This ensures that a “peer restart” event will not be missed.
  - Management is expected to issue `mr_restart` from there (timing not specified).

# Missing piece: handling signal disruptions (3)

- Signal disruption (detection+recovery) longer than quiet\_timer can be a result of the peer having mr\_restart asserted.
  - If mr\_training\_enable is true, this condition should cause a restart of the local device too (training is symmetric).
  - It may be possible to detect restart of training by other means, but it's hard to cover all implementations.
- **If mr\_training\_enable is true**, we should enable distinguishing a temporary disruption from a peer restart.
  - This can be achieved by having:  
$$(\text{max time to deassert local\_rx\_ready}) + (\text{max los\_timer}) < (\text{min quiet\_timer})$$
  - Proposed values: **(40 ms local\_rx\_ready) + (50 ms LOS) < (100 ms quiet)**
  - Alternatively, we can increase the minimum quiet\_timer to allow a longer los\_timer
  - An alternative approach is shown on [slide 17](#)
- **If mr\_training\_enable is false**, there is no need to restart training
  - “training” is asymmetric in this case, and recovery from LOS is fast
  - The interface output is not squelched upon LOS at the interface input

# Implicit assumptions

1. An interface can detect that its peer entered the QUIET state, and it will cause `local_rx_ready` to be false
  - `local_rx_ready` definition includes “the peer interface transmitter is transmitting a PAM4 signal” but **there are no timing requirements or other guarantee it will be detected within `quiet_timer` (100-200 ms)**
2. (based on previous generations) having an invalid input signal (e.g. QUIET) will cause `SIGNAL_OK` to become FAIL
  - But **in this project `SIGNAL_OK` is derived only from `rts_status` (see previous slides)**
3. `SIGNAL_OK=FAIL` at the service interface input will propagate to subsequent sublayers/ISLs, e.g. through squelch
  - Squelching was allowed as an option for PMAs in 802.3df
  - But **in this project, squelch it is not mentioned**
  - As an example, for a PMD, `SIGNAL_OK=FAIL` in `IS_SIGNAL.request` means the adjacent interface (e.g. AUI-C2M) has no incoming signal – but there is no mention of the effect in the PMDs
4. Management will monitor some internal state (in any device) to decide whether to restart training (`mr_restart`)
  - Makes sense. But **currently not required**

# Output when SIGNAL\_OK=FAIL

- But the PMDs (neither in clauses 182/183 nor others) do not have a specified behavior when PMD:IS\_SIGNAL.request has the value FAIL
  - AUIs are functionally equivalent to PMDs so have no such specifications either
  - Neither does clause 184 (inner FEC)
- The exception is 177.3, which has the following text:

The Inner FEC receives signal status information from the client sublayer (PMA) using the FEC:IS\_SIGNAL.request(SIGNAL\_OK) primitive. The SIGNAL\_OK parameter takes one of four values: OK, READY, IN\_PROGRESS, or FAIL. When SIGNAL\_OK is IN\_PROGRESS or FAIL, the corresponding tx\_symbol parameters on all lanes are unspecified.

  - It addresses the case of FAIL in **FEC:IS\_SIGNAL.request**
  - Table 177-2 maps this to FAIL in **PMD:IS\_SIGNAL.request**
- No change required in clause 177

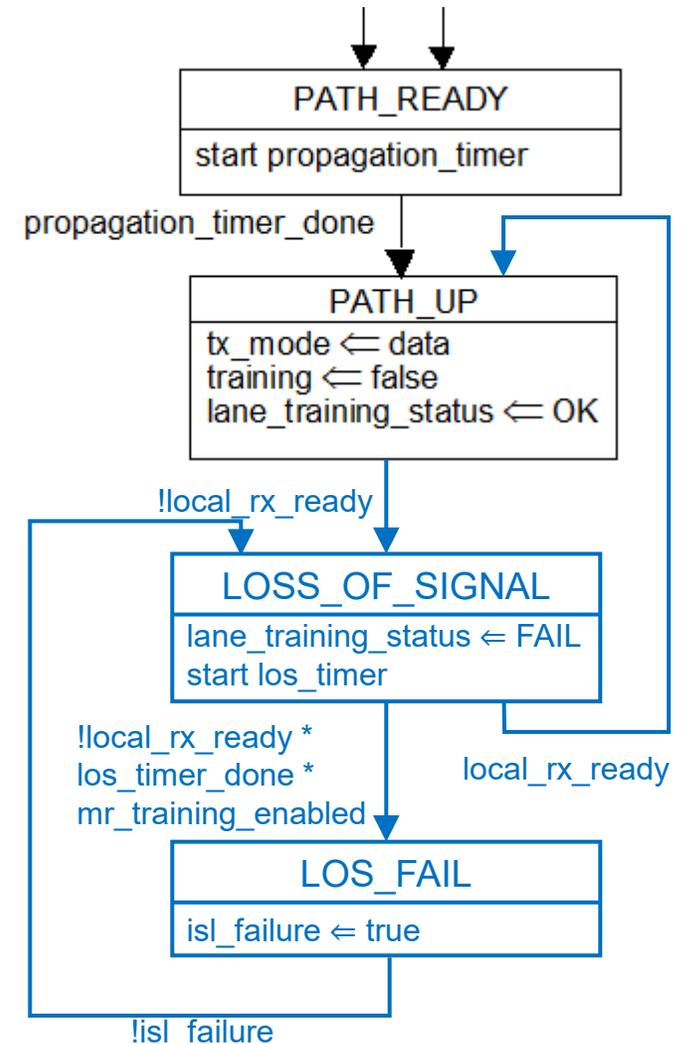
# Possible addition: recovery from extended LOS

The suggested additional states shown on [slide 12](#) result in LOS\_FAIL being a terminal state. Recovery from LOS\_FAIL requires restart of training (by management).

It may be desirable to enable recovery from this state without re-training – **let management decide if it is a failure or not.**

One way to achieve that is by adding a transition as shown on the right, with a new indication variable “isl\_failure” that is R/W, such that management can reset it.

- This would enable recovery from LOS\_FAIL by high-level management (e.g. CMIS host) without restarting training – for example, if the PCS has re-acquired AM lock.
- High-level management is also able to restart training on an ISL using mr\_restart, for example, if the PCS has lost AM lock for an extended time.



Optional addition to Figure 178B–10

# Proposed changes – 1 (Annex 178B)

- Add the proposed changes to the training control state diagram (either [Slide 12](#) or [Slide 17](#)).
- Add a new timer in 178B.8.3.3:

## los\_timer

This timer is started when the training control state diagram (Figure 178B–10) on a lane enters the LOSS\_OF\_SIGNAL state. The terminal count of this timer is between 40 ms and 50 ms.

- Change the definition of local\_rx\_ready in 178B.8.3.1:

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 a PAM4 signal, that the remote transmit has been optimized if mr\_training\_enable is true, the local receive equalizers have been optimized, and that no further adjustments are required for normal data transmission. [The specific conditions and timing for setting this variable to true are implementation dependent. When this variable is true, application of an input signal corresponding to tx\\_disable=true in the peer shall set it to false within 40 ms.](#)

- After the definition of training\_failure in 178B.8.3.1, add the following NOTE

[NOTE—a value of true in training\\_failure indicates that training needs to be restarted by asserting reset or mr\\_restart. The timing for this operation is not specified.](#)

- Change the NOTE following the definition of mr\_restart in 178B.8.2.1 as follows:

NOTE — There is no specified time limit for the APSU. Restarting APSU might result in live-lock situation, thus APSU should only be restarted if there is an indication of [training\\_failure \(see 178B.8.3.1\)](#) or an unrecoverable fault. The definition of unrecoverable fault is beyond the scope of this standard.

# Proposed changes – 2 (service interfaces)

- In the service interface subclauses, add change the existing paragraphs and add an informative note as follows:
  - The SIGNAL\_OK parameter of the PMD:IS\_SIGNAL.indication primitive corresponds to the variable rts\_status of the ready-to-send (RTS) function (see <reference>), as defined in 178B.8.2.1. When SIGNAL\_OK is either IN\_PROGRESS or FAIL, the rx\_symbol parameters of PMD:IS\_UNITDATA\_i.indication on all lanes are unspecified.
  - The SIGNAL\_OK parameter of the PMD:IS\_SIGNAL.request provides the status from ISLs above the PMD. [When this parameter has the value FAIL, the tx\\_symbol parameters of the PMD:IS\\_UNITDATA\\_i.request are unspecified, and if the PMD transmit function is in DATA mode \(see <reference>\), the transmitter output is unspecified.](#)
  - [NOTE—Examples of possible behavior when SIGNAL\\_OK is FAIL are transmitter disabled \(e.g., using the PMD global transmit disable function, see <reference>\) or transmitting an internally-generated PAM4 signal using a local clock.](#)
- This should be applied in 178.4, 179.4, 180.3, 181.3, 182.3, and 183.3
- In 176C.4 and 176D.4 the text is different, but should be edited to have the same effect
- 184.3 and 186.2.2 should be edited include FEC:IS\_SIGNAL.request (similar to 177.3).

# Summary

- The D3.0 specifications of the ILT function and PMD/AUI specifications do not adequately address signal disruption events.
- The changes proposed on slides 12, 16, and 17 define the behavior when an invalid signal is detected on an interface.
  - A new LOSS\_OF\_SIGNAL state is defined for temporary disruption.
  - Disruption longer than a specified time (suggested 40-50 ms) will cause a transition to FAIL state. Recovery from FAIL should be handled by management.
- The proposed changes allow a retimer to either squelch or send a locally-generated pattern in DATA mode during LOS in the adjacent interface.

# That's all!

Questions?