Segment by Segment AUI Training Thoughts

Adee Ran, Cisco

Kent Lusted, Intel

Intro

- In-band training for AUIs (segments of the link) was adopted in January 2024.
- The training method is based on the Clause 136 PMD control, which has been available for two generations of PAM4 transceivers.
- The PMD control function of Clause 136 is based on functions defined in the earlier Clause 92 and Clause 72, which are suitable for a single end-to-end electrical channel (e.g. passive copper cable or backplane), with no retimers.
- This presentation explores the challenges of training with a segmented link, with one or more retimers, and suggests a path forward.

Transition from training to data

- According to the PMD control state diagram, when training is completed, the PMD switches to DATA mode:
  1. Sends its input in the transmit direction to the media
  2. Sends its input in the receive direction (from the media) to its client (toward the MAC).
- The PMD clients are the PMA and PCS
- The PCS and PMA on both partners are assumed to be operational when the PMD is training
  - when the transition occurs there is already data to send in the transmit direction
  - Similarly, the link partner's data is received
  - The PCSs need to align on each other's data – and the link is up

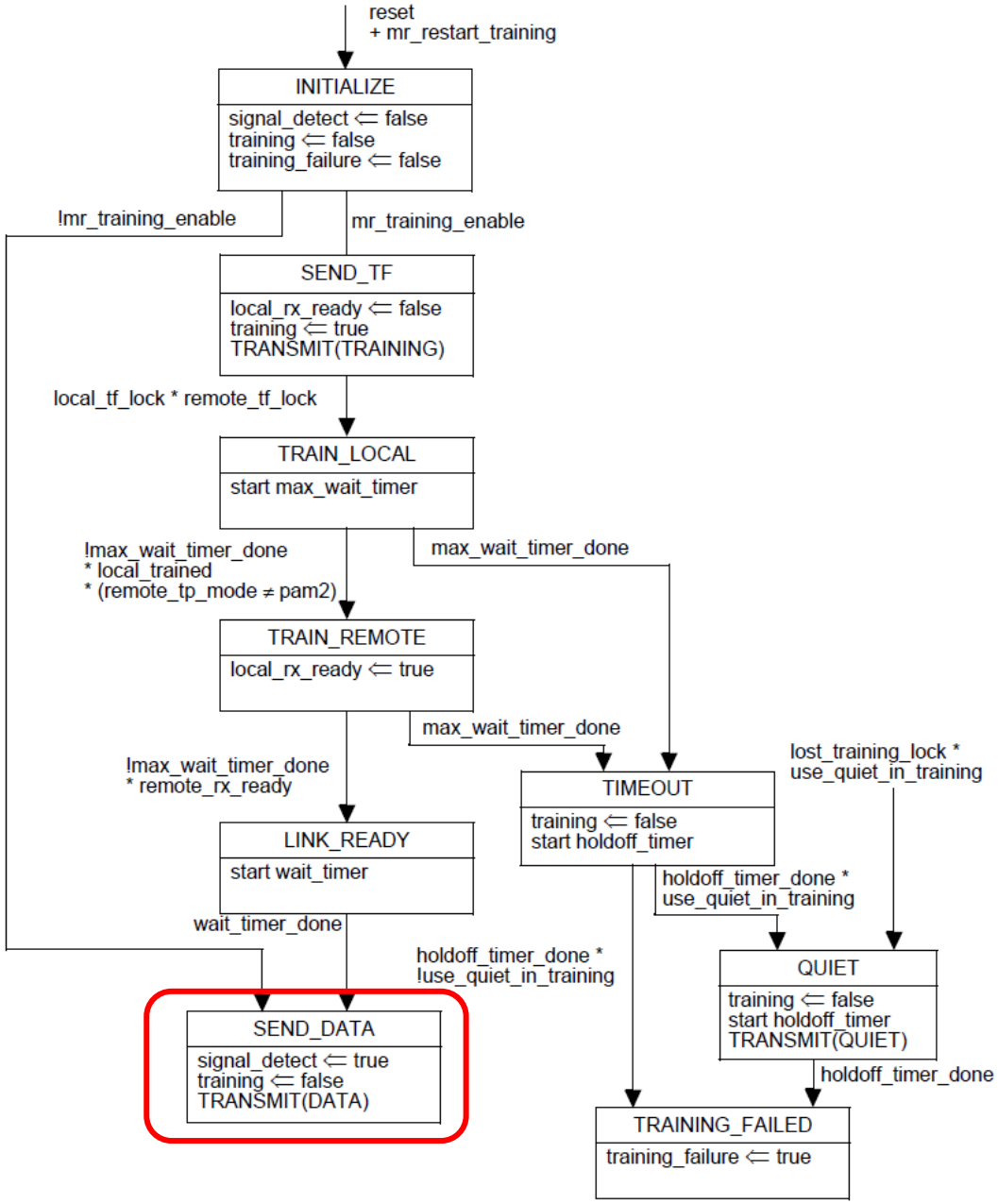
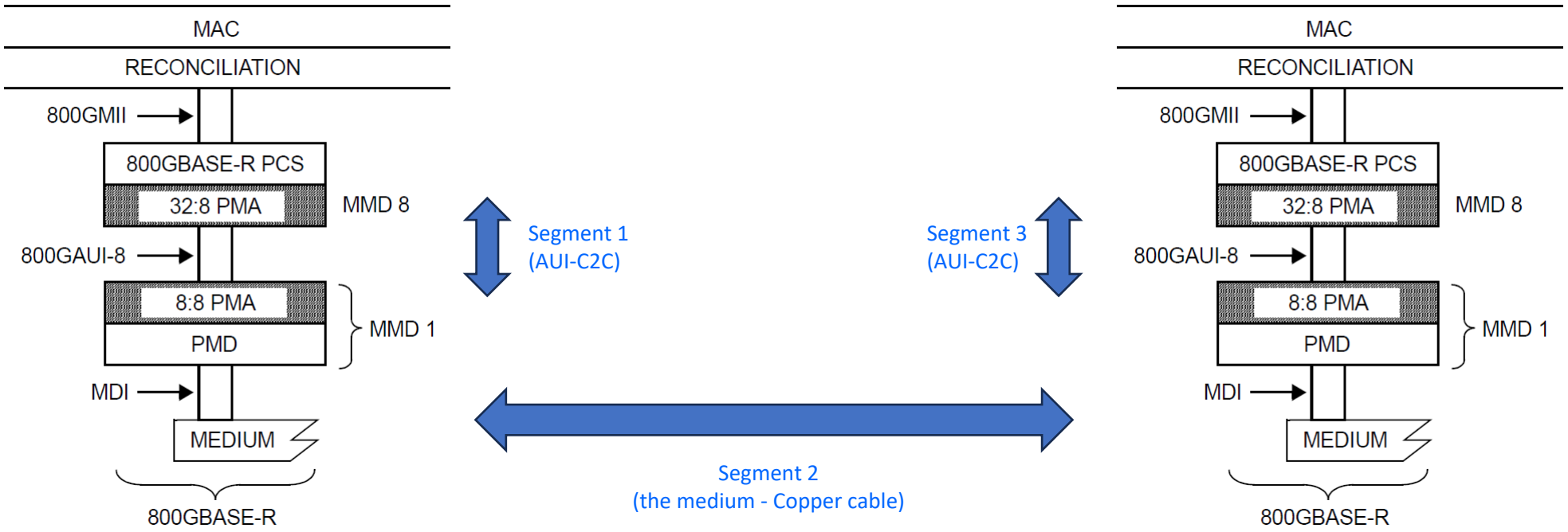


Figure 136–7—PMD control state diagram

Consider training over multiple segments (medium and any number of AUIs)

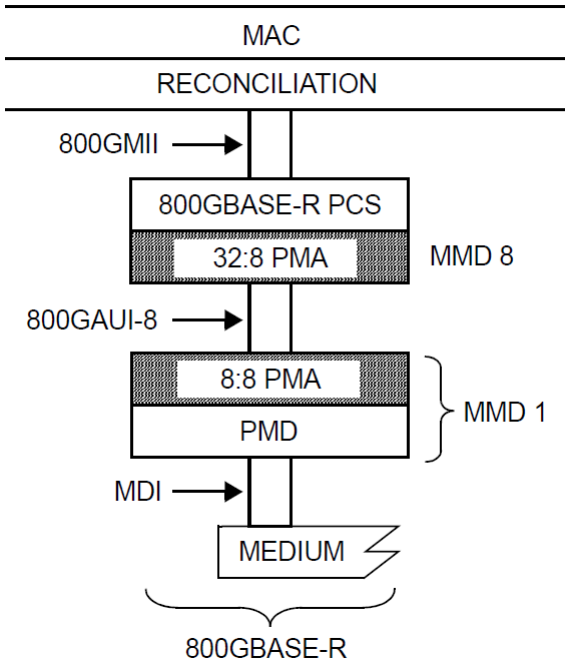
Example: 800GBASE-CR8 link (Clause 162) with 800GAUI-8 C2C on both sides



Can training work on segments?

Problem statement

- If the existing “PMD training” is attempted between the PMAs across an AUI:
  - When PMA training is done, it is possible that there is no data from the media to send “upward” yet
    - For example, the PMD may still be training with the link partner, or running AN, or there may be no partner
    - It is not defined what the bottom PMA should send upward in this case
  - If the PCS gets a signal_detect=OK indication, it will fail to lock
  - Restarting the process causes link flaps, possibly repeated on the next attempt
  - The existing state diagram does not allow any way out of this scenario
- If the PMD training is completed first, a similar problem occurs – the bottom PMA has no PCS data to send to the link partner
  - The link partner expects PCS data immediately
- Serializing the training on the two segments does not help – one will always switch first
- Unless the switch to data is synchronized somehow…



Enabling segment-by-segment training

- Ideally, we would like to transition to data mode in synchrony across multiple retimers and segments, when the whole link is ready.
  - This requires changes in the training state diagram.
- Additional goals:
  - Re-use the existing training protocol (“PMD control function” of clause 136/162) with as few changes as possible
  - Have common functionality for all segments (AUIs and media) and devices
  - Support a link with segments (AUIs or media) that use in-band training, and other segments that do not, with various combinations of AUI widths and PMD types
  - Have each device operate independently, with only local knowledge, and minimum management intervention
  - Enable training starting and completion in different segments in any order
  - Enable non-real-time implementations (e.g. using firmware and/or CMIS management)

Elements of the proposed solution

- When training is used on a segment, extend the training frame exchange duration (to keep the segment “alive”) until the whole link is ready.
- Indicate the availability of PCS data, and forward this indication across “ready” segments.
  - When the indication has propagated across the whole link, all devices can switch to data mode.
- When training is not used, use the “signal detect” and “transmit disable” functions instead to signal the presence of PCS data.
  - Leverage the functionality that was defined for the PMAs in 802.3df.
  - We assume existing devices behave this way.
- Timeouts are not required for training
  - Relaxes implementations and can help debugging
  - Failures are still detected and handled

Overview of the proposed training scheme

- A new handshake state diagram, based on the PMD state diagram (Figure 136-7)
  - The handshake function includes both “training” and “no training” modes.
- A new variable is introduced: **Ready To Send** (RTS)
  - RTS is exchanged over training frames between segment partners, similar to the existing **Receiver Ready** and **Training Frame Lock** variables.
  - All these variables have two versions, “local” (transmitted to the segment partner) and “remote” (received from the segment partner).
  - RTS is propagated across the segments, as will be shown next.
- Each PMA/PMD, on each physical interface, has a separate handshake function (and the associated variables).
  - Retimers, modules, etc. have two such functions, with specified information exchange between them.

Handshake state diagram

(instance on each lane of each interface)

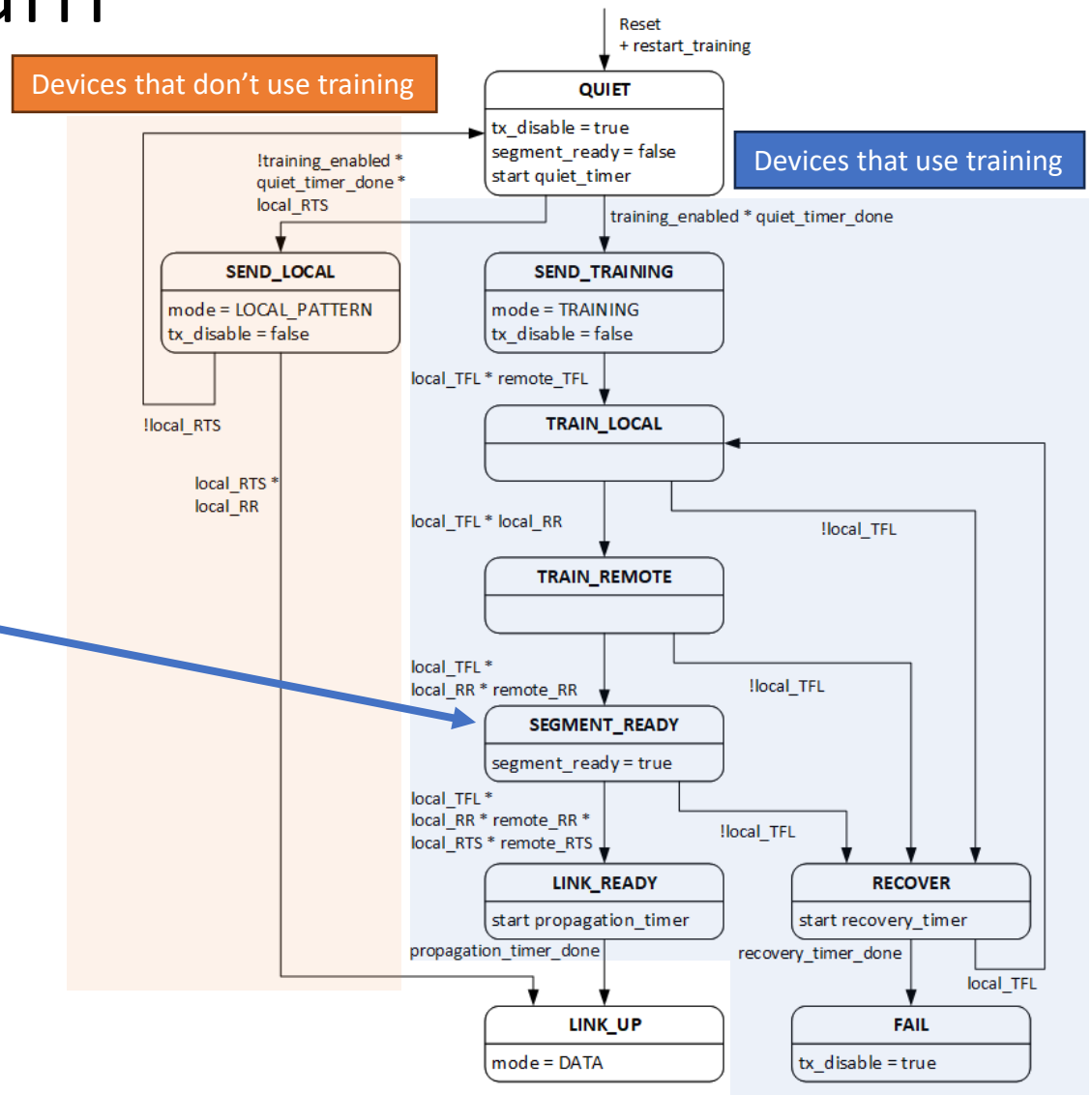
This diagram is based on the PMD control state diagram in Figure 136–7, with some state re-ordering (start with QUIET).

The new state SEGMENT_READY enables extending the exchange of training frames until the whole link can be brought up.

The condition for switching to DATA mode is that RTS is both **sent (local)** and **received (remote**). This indicates that the PCSs on both ends are “seen” on this interface.

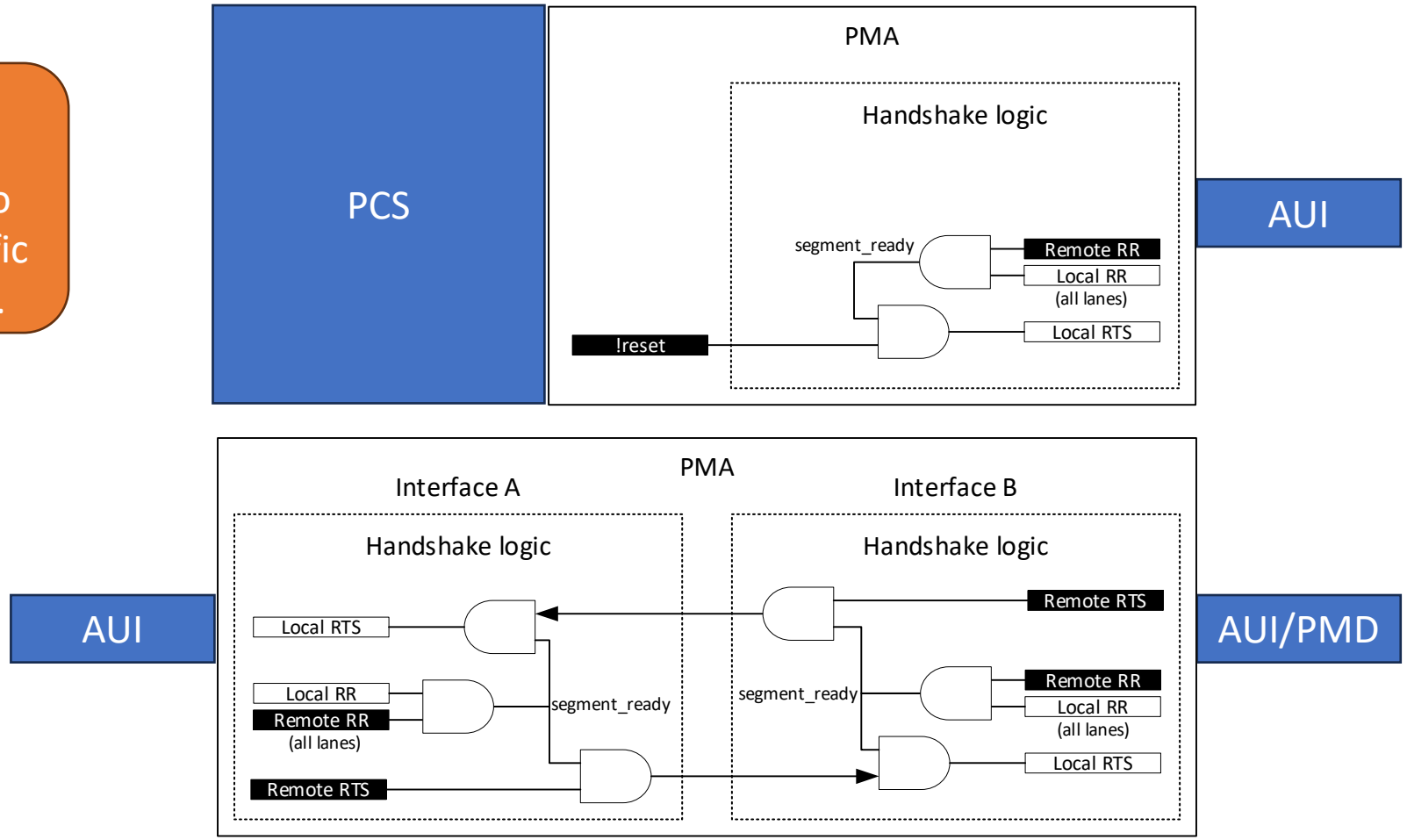
Since RTS is propagated, all PMAs will switch to DATA mode at about the same time.

Suggested timer periods:
- quiet_timer: ≥ 100 ms
- recovery_timer: 20 to 30 ms
- propagation_timer: 100 to 200 ms



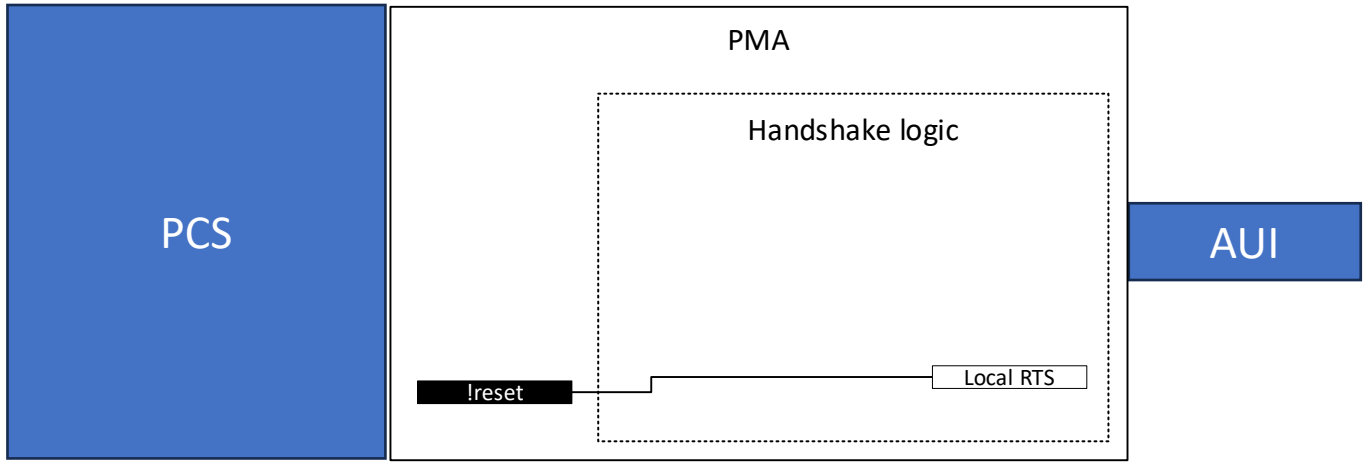
local_RTS generation and propagation when a training protocol is used

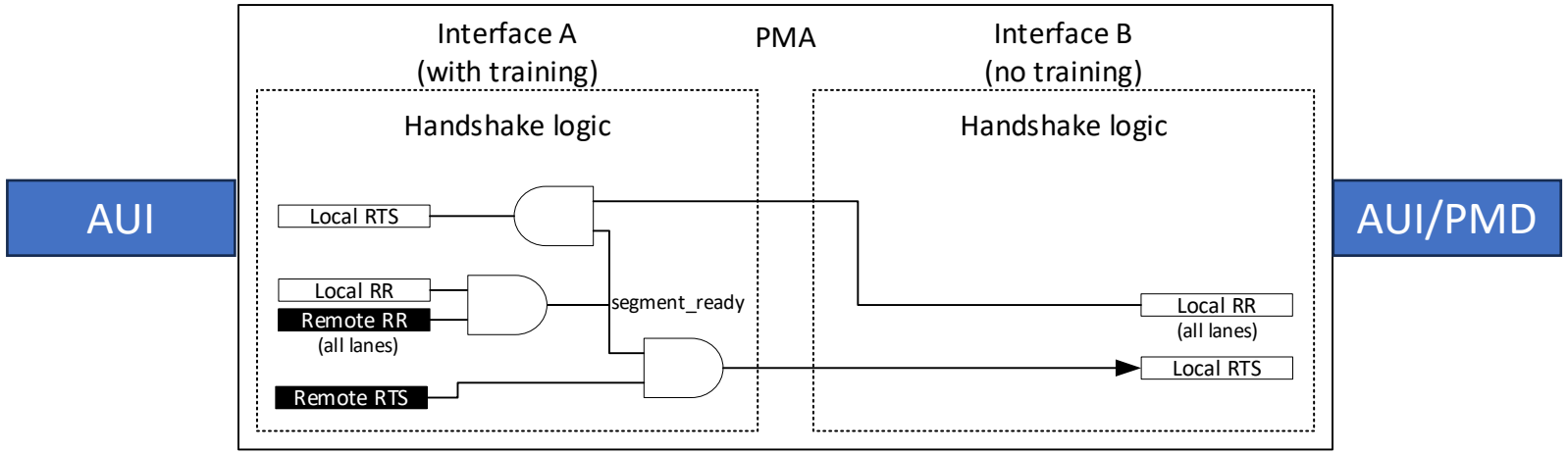
These are conceptual diagrams, and do not imply a specific implementation.



local_RTS generation and propagation when a training protocol is not used

These are conceptual diagrams, and do not imply a specific implementation.





Additional details to be discussed

Future work needed on:

- Tying it to the service interface
- Addressing clock source switching in retimers
- How are failures handled?
- How will it work with auto-negotiation?
- Management and observability

- This presentation only touches on the definitions of variables and state machines…

Summary

- Training a link composed of multiple segments requires synchronization
- There is an approach for segment-by-segment training that reuses the existing PMD control function with a few minor changes
  - New handshake state diagram with new variable “Ready to Send (RTS)”
- Detailed definitions of variables and state machines were provided
  - More details coming soon
- Intent is to bring a consensus proposal with sufficient details to implement in D1.x
  - It will be quite long…
  - Help would be most welcome!
- Anyone who is interested in more details is encouraged to contact the authors.

Thanks!