# Inter-sublayer service interface for training

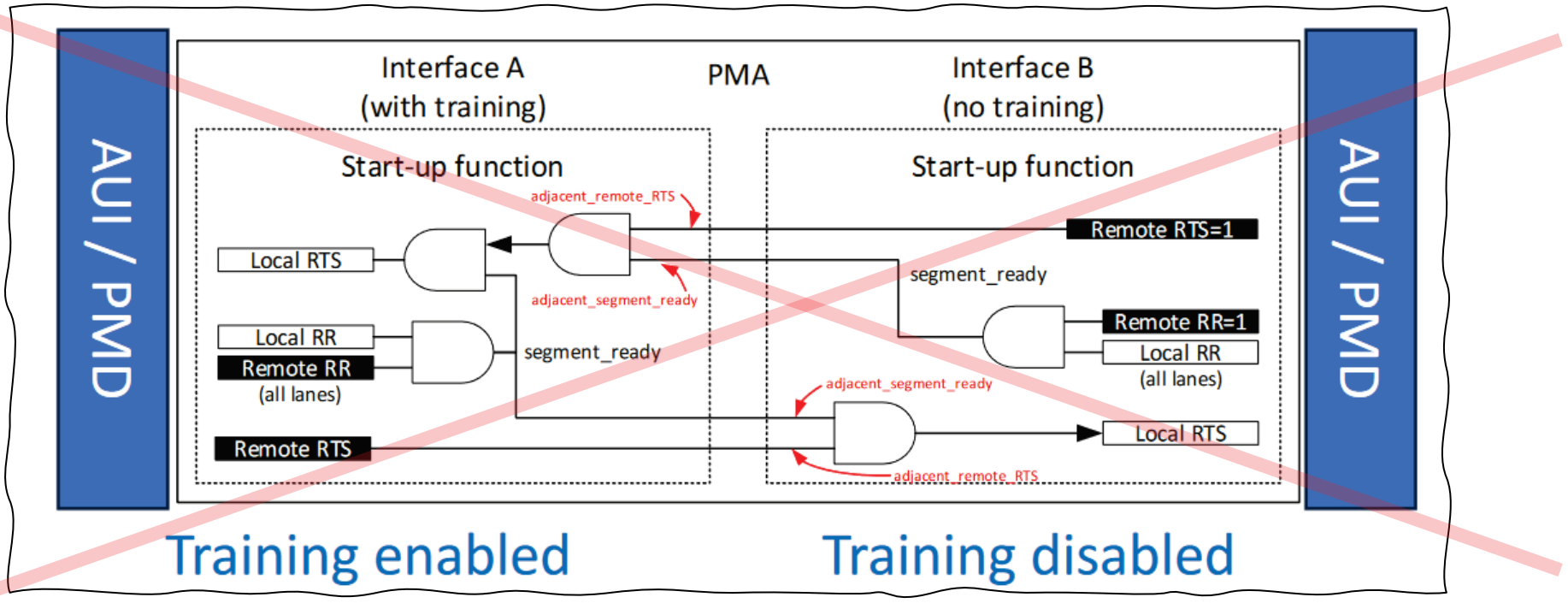
(comments #194, #195 against D1.0)

Adee Ran, Cisco

# Introduction

- The adopted segment-by-segment training concept relies on passing indication about the status of training between sublayers.
- When there is a physical interface with a training protocol, RTS is communicated using the protocol. But when two sublayers are attached, e.g. PMD and PMA, the status needs to be communicated through the service interface connecting these sublayers.
- This is not fully covered in D1.0.
- In addition, the effect of segment-by-segment training on the Auto-Negotiation (AN) function needs to be addressed.

# Information flow between PMA interfaces

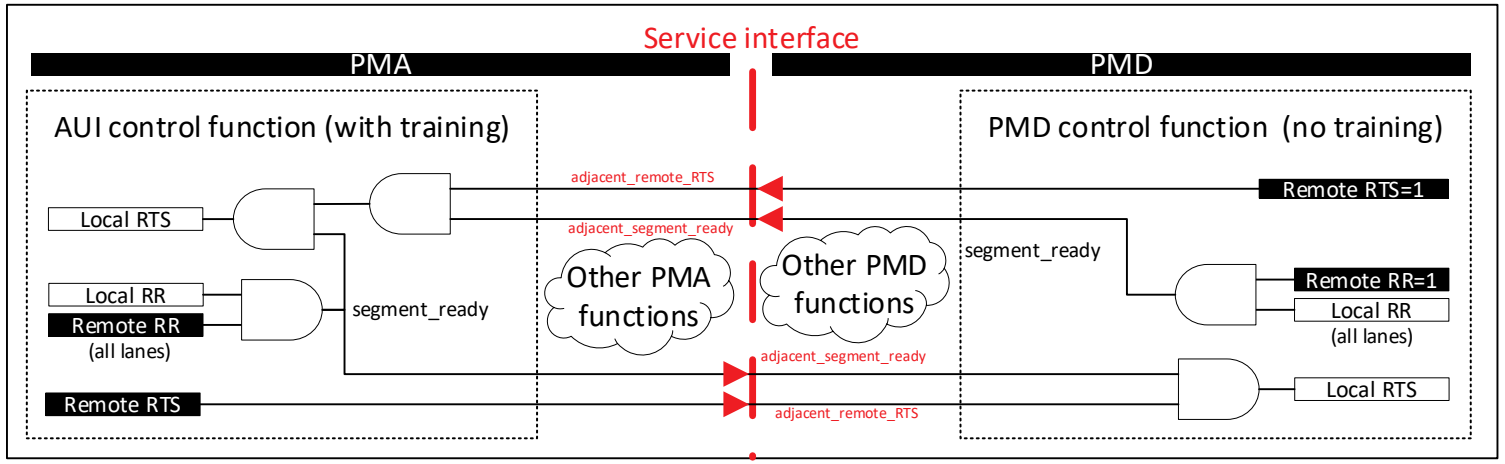


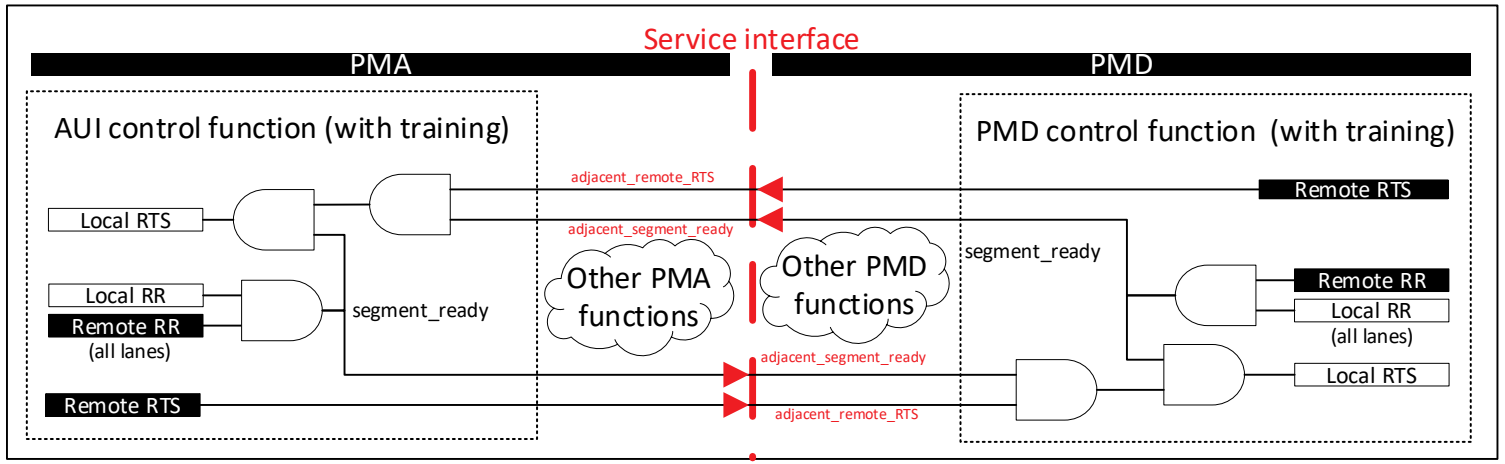

The diagram presented in the training proposal assumed that the training is a function of the PMA (assuming the PMA includes the SerDes), and thus the signals shown by the arrows are internal variables.

However, in implementation of the training proposal in D1.0 it was decided that the training belongs in the PMD or the AUI transceiver.

**Thus, the training information that was supposed to be internal to the PMA needs to be passed between sublayers over the inter-sublayer service interface…**

# Information flow across service interfaces




Two bits of information are passed between sublayers in each direction:

- segment_ready – essentially "segment-level signal detect"
- remote_RTS – essentially "link-level signal detect"

In addition, indication of training failure should be communicated.

Can this be done with the existing service interface?

# IS_SIGNAL in the existing inter-sublayer service interface

**174.3.1 Inter-sublayer service interface**

The inter-sublayer service interface is described in an abstract manner and does not imply any particular implementation. The inter-sublayer service interface primitives are defined as follows:

IS_UNITDATA_*i*.request
IS_UNITDATA_*i*.indication
IS_SIGNAL.indication
IS_SIGNAL.request

<...>

The IS_SIGNAL.indication primitive is used to define the transfer of signal status from a sublayer to the next higher sublayer.

The IS_SIGNAL.request primitive is used to define the transfer of signal status to a sublayer from the next higher sublayer.

<...> (174.3.2)

As an example, the primitives for the PMD service interface are identified as follows:

PMD:IS_UNITDATA_*i*.request
PMD:IS_UNITDATA_*i*.indication
PMD:IS_SIGNAL.indication

**116.3.3.3.1 Semantics of the service primitive**

IS_SIGNAL.indication(SIGNAL_OK)

The SIGNAL_OK parameter can take on one of two values: OK or FAIL. A value of FAIL denotes that invalid data is being presented (rx_symbol parameters undefined) by the sublayer to the next higher sublayer. A value of OK does not guarantee valid data is being presented by the sublayer to the next higher sublayer.

There is one primitive in each direction
in clauses 174 and 169
But clause 120 defines only IS_SIGNAL.indication

The “request” primitive (egress signal indication) is not mentioned in many places (e.g. all diagrams)

The semantics of the primitive only allows one bit to be communicated in each direction…
And “OK” does not guarantee valid data.

# What does IS_SIGNAL mean?

- The IS_SIGNAL service interface primitives stem from the notion of optical power detection in a PMD ingress – which is a binary low-quality indication.
  - The training function requires detecting and decoding a specific signal (training frames) – not just a power detector.
  - Even if training is not enabled, devices for 200 Gb/s per lane (and even lower rates) include advanced DSPs that are capable of reliable indication of signal quality, based on CDR locking, SNR measurement, etc.
- Electrical PMDs use IS_SIGNAL.indication to pass the status of training in the ingress direction.
- Previously there was no need to pass anything in the egress direction.
  - But with training, this information is required.
- If we want to use IS_SIGNAL, the existing semantics is not adequate for new devices, and should be changed.
  - However, the service interface is used by multiple existing sublayers (starting rates from 25 Gb/s NRZ) – we cannot change it for deployed devices – so it needs to be defined carefully…
  - The next slides show possible backward-compatible changes. (there may be other ways to do it!)

# Proposed semantics – ingress direction

IS_SIGNAL.indication(SIGNAL_OK)

The SIGNAL_OK parameter can take on one of four values: OK, FAIL, IN_PROGRESS, or READY. The values IN_PROGRESS and READY are defined only for specific instances of the service interface that use the control function defined in Annex 176A.

A value of OK indicates that communication with the next lower sublayer is established. If the service interface supports the values IN_PROGRESS and READY, then a value of OK indicates that valid data is being presented by the sublayer to the next higher sublayer in the rx_symbol parameters.

A value of FAIL indicates the sublayer has not established communication with the next lower sublayer. Data is not being presented by the sublayer to the next higher sublayer (the rx_symbol parameters are unspecified). If the service interface supports the values IN_PROGRESS and READY, then a value of FAIL indicates that an attempt to communicate with the next lower sublayer has failed.

A value of IN_PROGRESS indicates that the sublayer is establishing communication with the next lower sublayer. Data is not being presented by the sublayer to the next higher sublayer (the rx_symbol parameters are unspecified), but it is considered a temporary state, and the sublayer does not require management intervention.

A value of READY indicates that communication with the next lower sublayer is established, but communication with the link partner is not fully established yet. The rx_symbol parameters presented to the next higher sublayer are valid but do not represent traffic data. It is considered a temporary state, and the sublayer does not require management intervention.

# Proposed semantics – egress direction

IS_SIGNAL.request(SIGNAL_OK)

The SIGNAL_OK parameter can take on one of four values: OK, FAIL, IN_PROGRESS, or READY. The values IN_PROGRESS and READY are defined only for specific instances of the service interface that use the control function defined in Annex 176A.

A value of OK indicates that communication with the next higher sublayer is established. If the service interface supports the values IN_PROGRESS and READY, then a value of OK indicates that valid data is being presented by the sublayer to the next lower sublayer in the tx_symbol parameters.

A value of FAIL indicates the sublayer has not established communication with the next higher sublayer. Data is not being presented by the sublayer to the next lower sublayer (the tx_symbol parameters are unspecified). If the service interface supports the values IN_PROGRESS and READY, then a value of FAIL indicates that an attempt to communicate with the next higher sublayer has failed.

A value of IN_PROGRESS indicates that the sublayer is establishing communication with the next higher sublayer. Data is not being presented by the sublayer to the lower higher sublayer (the tx_symbol parameters are unspecified), but it is considered a temporary state, and the sublayer does not require management intervention.

A value of READY indicates that communication with the next higher sublayer is established, but communication with some upper sublayer is not fully established yet. The tx_symbol parameters presented to the next lower sublayer are valid but do not represent traffic data. It is considered a temporary state, and the sublayer does not require management intervention.

# Communicating training status with the new service interface

- **FAIL** indicates that training has failed (FAIL state) in one or more lanes.
- **IN_PROGRESS** indicates that training is ongoing, but segment_ready is 0.
- **READY** indicates that training is completed, segment_ready is 1 but remote_RTS is 0, so the PMD is still in TRAINING mode.
- **OK** indicates that training is completed, segment_ready is 1, remote_RTS is 1, and the PMD is in DATA mode.

The receiving sublayer sets its adjacent_segment_ready and adjacent_remote_RTS variables accordingly.

If a value of FAIL is received it is propagated to the next service interface.

# What should be changed to enable the new service interface

- 116.3.1 Inter-sublayer service interface
  - Add IS_SIGNAL.request
- 116.3.2 Instances of the Inter-sublayer service interface
  - Add IS_SIGNAL.request, update Figure 116–2 and Figure 116–3
- 116.3.3.3.1 Semantics of the service primitive
  - Update the semantics of IS_SIGNAL.indication as in previous slides
- 116.3.3.4 IS_SIGNAL.request (new subclause)
  - Define the primitive with semantics as in previous slides

- 169.3.2 Instances of the inter-sublayer service interface
  - Add IS_SIGNAL.request, update Figure 169–2 and Figure 169–3
- 174.3.2 Instances of the inter-sublayer service interface
  - Add IS_SIGNAL.request, update Figure 174–2, Figure 174–3, and Figure 174–4

**(169 and 174 point to 116 for semantics)**

- **176A.10.2 Per-interface variables, functions and timers**
  - Add a variable (training_status?) to explicitly store the value mapped to SIGNAL_OK.
  - Set this variable in the state diagram in Figure 176A–6.

# Combining training and AN

# Interaction of training with the AN sublayer

- AN can be used in the segment that is attached to the media (PMD to PMD).
  - The result of AN can affect the AUIs – widths, whether they are 200G or 100G per lane (or other rates).
  - AUIs can take time to configure and train.
- The PCS associated with the PMD is required to support the AN service interface primitive AN_LINK.indication(link_status)
- Once AN selected a PHY (HCD) and enabled it, if **link_status** is **FAIL** for long enough time (**link_fail_inhibit_timer**), then AN will time out, disable the PHY, and restart…
  - But configuration and training on multiple segments can take an arbitrarily long time!
  - Having a long timeout has implications on recovery time
- Currently, the only way to prevent this restart is that the AN receives **link_status=OK** from the PCS before the timer expires.
- It is preferable not to have a mandatory timeout/restart when training is running on any of the AUIs.
  - Management can always restart AN or training, but this should be done only when a failure is indicated.

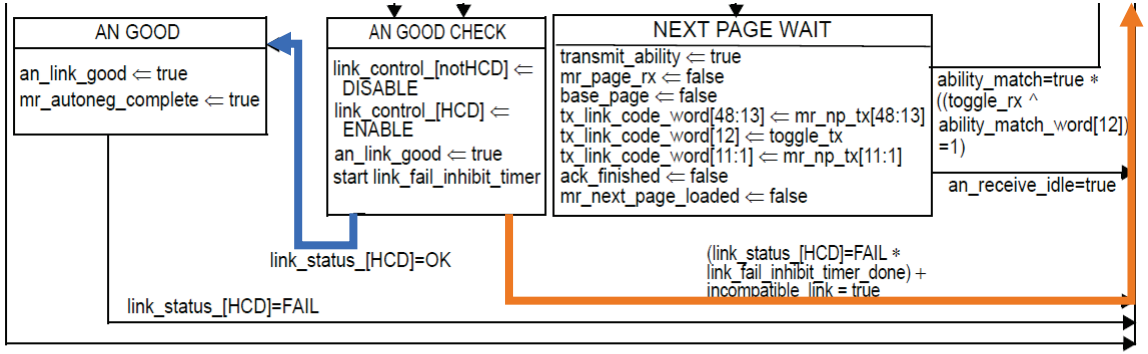



**Figure 73–11—Arbitration state diagram**

- As an alternative we could specify that PMD training starts only after AUIs have completed training.
  - While AUI are trained, AN signaling (e.g. null pages) should be continued on the media.
  - This would prevent **link_fail_inhibit_timer** from starting…
  - But also prevent concurrent training of multiple segments

**It is preferable to allow concurrent training when possible. This method should be considered as a last resort.**

# What can be done about AN

- A possible way is to define a third possible value for link_status, **IN_PROGRESS** (in addition to “FAIL” and “OK”).
- link_status=IN_PROGRESS when any PMA inside the PHY is sending training frames (which can happen for an arbitrarily long time).
  - This is indicated by having either IN_PROGRESS or READY in the PMA:IS_SIGNAL.indication.
  - link_status=IN_PROGRESS will keep the state diagram above in “AN GOOD CHECK” state, preventing AN from restarting.
  - When a PMA switches to data mode, the value of IS_SIGNAL.indication will become OK (RTS propagation). Eventually the adjacent PMA will change to OK when training is completed.
  - If the adjacent PMA reports OK, the PCS can start its own locking procedure. link_status of the PCS should be IN_PROGRESS until locking succeeds. It changes to FAIL if there is a timeout.
- If any PMD (or AUI device) loses frame lock for a long enough time, it will go to QUIET, and will set SIGNAL_OK=FAIL.
  - This will propagate towards the PCS which will set link_status=FAIL.
  - After link_fail_inhibit_timer, it will restart AN on one side of the link (starting with a quiet period).
  - On the other side of the medium, this will be detected by losing TFL, and then similarly SIGNAL_OK=FAIL will cause AN restart.

# Proposal

- Add a third possible value to the link_status parameter, IN_PROGRESS, as described in the previous slide.
- Support of the new value is mandatory for all 1.6 Tb/s PCS, and for any PCS connected to a 200 Gb/s per lane PMA. It is optional for other PCSs.
- The value IN_PROGRESS is assigned based on the adjacent PMA's signal_ok parameter as described in the previous slide.

# That’s all

Questions?

(See also backup slide)

# Backup

Service interfaces? I’m confused!

What does it all mean in practice?

# Implication for modules and retimers

- Modules contain more than one sublayer, e.g. PMA+PMD, and two interfaces
  - Line side, host side
- The presence of a signal, the completion of training, and the “connection to the PCS” (RTS) need to be communicated from one sublayer to another inside the same device.
  - These can be two separate Ips.
  - The communication between these IPs is modeled by the service interface but can be implemented in various ways.
- Similar considerations for on-board retimers (which are essentially two PMAs back-to-back).

# Implication for endpoints

- If AN is used – then right after the HCB is decided, the devices can be configured to the correct rates, widths etc.
  - The AN resolution can be read through management if it is in a separate device.
  - AN will not time out because training has not started yet, so the status is not FAIL anywhere.
- After this possible configuration, the PMA below the PCS starts communicating with its “AUI partner”.
- The PCS gets an indication from its PMA about the status of training across the link: FAIL, OK, or IN_PROGRESS (with the semantics defined in previous slides).
- The indication and its transitions can be used by “management” to control the PCS, upper layers, and/or the AN.
  - An initial value of IN_PROGRESS should be interpreted as link_status=IN_PROGRESS for AN_LINK.indication.
  - A transition from IN_PROGRESS to READY means the adjacent PMA is ready but some of the segments are still training. AN should not time out.
  - A transition from IN_PROGRESS or READY to FAIL means one of the segments has failed, and AN should restart.
  - A transition from IN_PROGRESS or READY to OK means all segments completed training, and the PCS can start its own alignment process.