# 802.3dj D2.0 Comment Resolution Common Track

Matt Brown (Alphawave Semi), 802.3dj Chief Editor
Adee Ran (Cisco), 802.3dj Electrical Track Lead Editor
Leon Bruckman (Nvidia)

# Introduction

- This slide package was assembled by the 802.3dj editorial team to provide background and detailed resolutions to aid in comment resolution.
- Specifically, these slides are for the various common-track comments.

# Sub-clause Heirarchy

# 174A – Subclause hierarchy Comment #403

*Cl* **174A** *SC* **174A.8.1** *P* **679** *L* **38** # 403

Mi, Guangcan — Huawei Technologies Co., Ltd

*Comment Type* **ER** *Comment Status* **D** *subclause hierarchy (bucket)*

There is only one sub-clause under 174A.8, which is 174A.8.1, no need to have this level in the hierachy.

*SuggestedRemedy*

remove the hierachy of 174A.8.1, make its sub-clauses 174A.8.x

*Proposed Response* *Response Status* **W**

PROPOSED ACCEPT IN PRINCIPLE.
The subclause hierarchy could indeed be improved. See related slides in the following editorial contribution:
<URL>/brown_3dj_03_2507

Also the hierarchy 174A.9.1 and 174A.10.1 are unnecessary. These can be removed, the underlying headings promoted. Some rewording is necessary.

Current hierarchy:

174A.8 Error ratio tests for 200 Gb/s per lane ISLs <reword this one>
174A.8.1 Block error ratio test methods using PMA-based measurements <delete this one>
174A.8.1.1 PMA block error ratio test configurations <promote heading level>
174A.8.1.2 PMA block error counters <promote heading level>
174A.8.1.3 PMA error histogram measurement <promote heading level>
174A.8.1.4 Convolution of error histograms <promote heading level>
174A.8.1.5 Error mask test method using PMA-based measurements <promote heading level>
174A.8.1.6 Block error ratio method for all lanes using PMA-based measurements <promote heading level>
174A.8.1.7 Block error ratio method for a single lane using PMA-based measurements <promote heading level>
174A.9 Error ratio tests for 800GBASE-LR1 ISLs <reword this one>
174A.9.1 Block error ratio test methods using Inner FEC measurements <delete this one>
174A.10 Error ratio tests for a PHY <reword this one>
174A.10.1 Block error ratio method using PCS-based measurements <delete this one>

Proposed hierarchy:

174A.8 Error ratio tests for 200 Gb/s per lane ISLs using PMA measurements
174A.8.1 PMA block error ratio test configurations
174A.8.2 PMA block error counters
174A.8.3 PMA error histogram measurement
174A.8.4 Convolution of error histograms
174A.8.5 Error mask test method using PMA measurements
174A.8.6 Block error ratio method for all lanes using PMA measurements
174A.8.7 Block error ratio method for a single lane using PMA measurements
174A.9 Error ratio tests for 800GBASE-LR1 ISLs using Inner FEC measurements
174A.9.1 Block error ratio test methods using Inner FEC measurements
174A.10 Error ratio tests for a PHY
174A.10.1 Block error ratio method using PCS measurements

# Error Ratio Figures

# 174A — Error ratio figures Comment #106, 292

Cl **174A** SC **174A** P **677** L **21** # 292

Brown, Matt — Alphawave Semi

*Comment Type* **TR** *Comment Status* **D** *(Common) Error ratio figure*

Diagrams showing the various paths or domains described in 174A.3 through 174A.7 would be very helpful to the reader of the annex.

*SuggestedRemedy*

Add a diagrams illustrating the paths described in 174A.3 through 174A.7.

*Proposed Response* *Response Status* **W**

PROPOSED ACCEPT.

Bruckman, Leon — Nvidia

*Comment Type* **TR** *Comment Status* **D** *(Common) Error ratio figure*

A figure will make this much more clear

*SuggestedRemedy*

Add a figure to show the link in 174A.5, 174A.6 and 174A.7

*Proposed Response* *Response Status* **W**

PROPOSED ACCEPT IN PRINCIPLE.
Resolve using the repsonse to comment #292.

#292 and #106 propose adding figures to help understand the context for each of the error ratio allocations.
Note also, that #590 (in bucket #1) proposes to rename “network path” to “mac-to-mac path”.

Cl **174A** SC **174A.3** P **677** L **35** # 590

Shrikhande, Kapil — Marvell

*Comment Type* **T** *Comment Status* **D** *(Common) (bucket)*

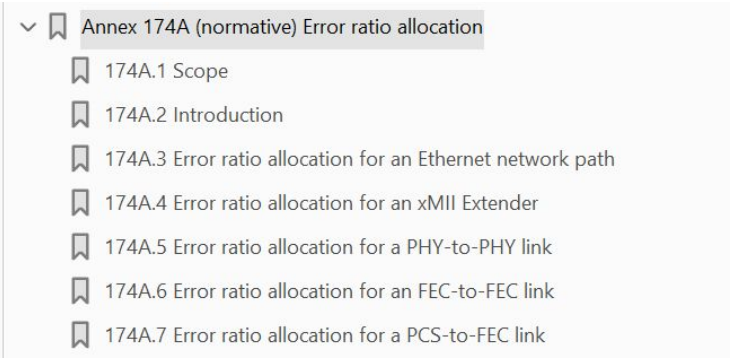
In the subclause title "Error ratio allocation for an Ethernet network path", the term "network path" is a bit vague. Network path may mean a multi-hop network path (e.g. End Host to Switch to End host). Should search for a more descriptive term to use instead of "network path". Since the error allocation is from the PLS service interface of one RS to the PLS service interface of the other RS, suggest using "RS-to-RS" ? or MAC-to-MAC ? This is similar to PHY-to-PHY, PCS-to-FEC, etc. terminology used in other sections of this annex.

*SuggestedRemedy*

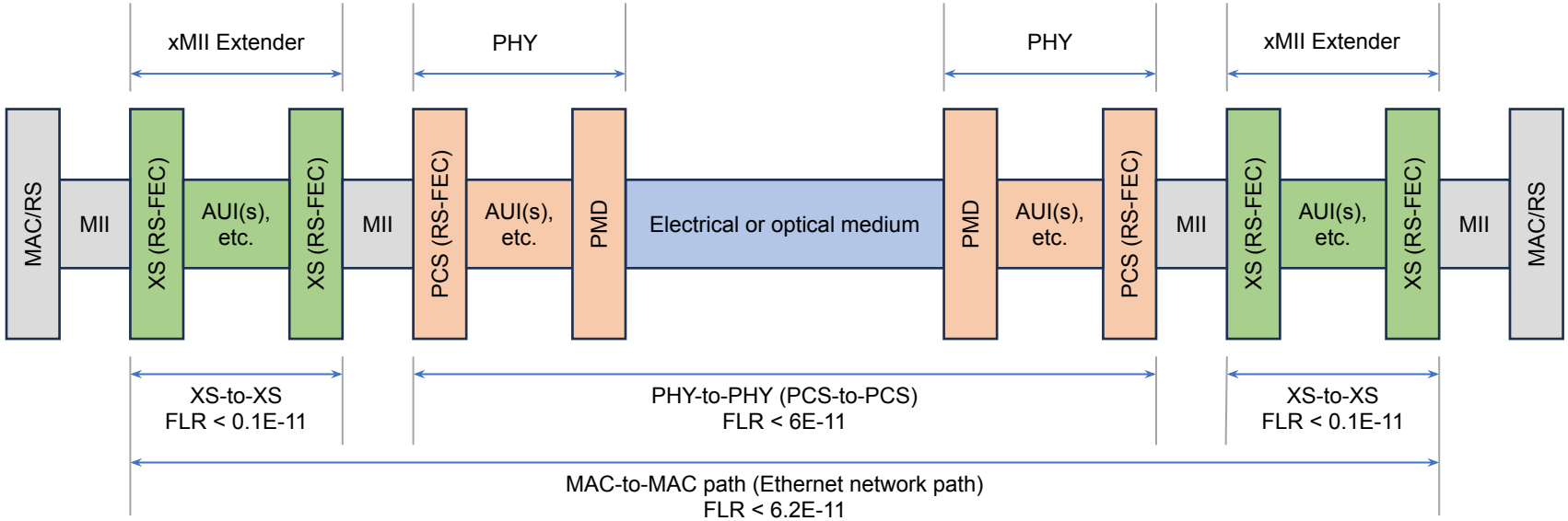
Replace "network path" in the subclause title with "RS-to-RS".

*Proposed Response* *Response Status* **W**

PROPOSED ACCEPT IN PRINCIPLE.
Ultimate the path is from MAC to MAC. Also, RS can easily be misinterpreted as meaning RS-FEC.
Change "network path" to MAC-to-MAC path.

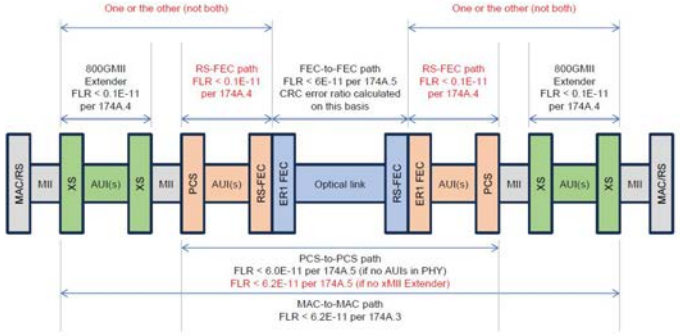


# 174A — Error ratio figures
# Proposed generalized figure for MAC-to-MAC path
# Relevant to 174A.3, 174A.4, 174A.5 (excluding 800GBASE-ER1/ER1-20)



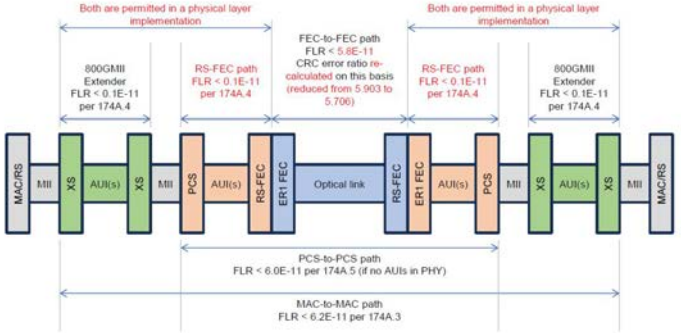
# 174A — Error ratio figures
# Proposed generalized figure for MAC-to-MAC path
# Relevant to 174A.3, 174A.4, 174A.5 (for 800GBASE-ER1/ER1-20)

Option #2: Allow either AUIs in the PHY or Extender, but not both




March 2025 IEEE 802.3dj Task Force

Option #3: Reduce FLR option for ER1 FEC path




March 2025 IEEE 802.3dj Task Force 19

For 800GBASE-ER1/ER1-20…
Option #2 (above) was adopted in Draft 1.5.
- Requires different figure

D2.0 comment #xxx proposes to adopt option # (right)
- Can use same figure as previous slide, if adopted

These options were proposed in the following contribution:
https://www.ieee802.org/3/dj/public/25_03/brown_3dj_04a_2503.pdf

# 174A — Error ratio figures
# Comment #585

Cl 174A SC 174A.6 P678 L28 # 585

Nicholl, Gary Cisco Systems

*Comment Type* TR *Comment Status* D *(Common) FLR allocation*

FLR allocation for 800GBASE-ER1/ER1-20.

During the March plenary the consensus was to adopt option# 2 of https://www.ieee802.org/3/dj/public/25_03/brown_3dj_04a_2503.pdf, for the FLR allocation for 800GBASE-ER1/ER1-20.

Also, see the final response to comment #16 in https://www.ieee802.org/3/dj/comments/D1p4/8023dj_D1p4_comments_final_clause.pdf.

An implication of this decision is that 800GBASE-ER1/ER1-20 PHYs are different from other 802.3dj PHYs, in that you are only allowed to have AUIs in the PHY or Extender, but not both (see slide 18 of brown_3dj_04a_2503). For other 802.3dj PHYs you are allowed to have AUIs in both the PHY and the Extender.

This means it is possible to have a host design that contains two AUIs (one in an Extender and one in the PHY) that would not support an 800GBASE-ER1/ER1-20 PHY, but would support all other 802.3dj PHYs.

I don't tihnk that an 800GBASE-ER1/ER1-20 PHY should be treated as a special case.

I propose changing the FLR allocation for the 800GBASE-ER1/ER1-20 PHY to be consistent with all other 802.3dj PHYs, such that there are no restriction on which hosts an 800GBASE-ER1/ER1-20 PHY can be deployed in.

This is essentially option #3 in brown_3dj_04a_2503, where the FLR of a 800GBASE-ER1/ER1-20 PHY, with or without an AUI, is defined as 6 x 10-11 (consistent with all other 802.3dj PHYs). This in turn means reducing the FLR for the ER1-to-ER1 FEC link from 6 x 10-11 to 5.8 x 10-11.

*SuggestedRemedy*

Change the FLR allocation for 800GBASE-ER1/ER1-20 to implement option #3 in https://www.ieee802.org/3/dj/public/25_03/brown_3dj_04a_2503.pdf.

Make the necessary changes in clauses 187 and 174A.

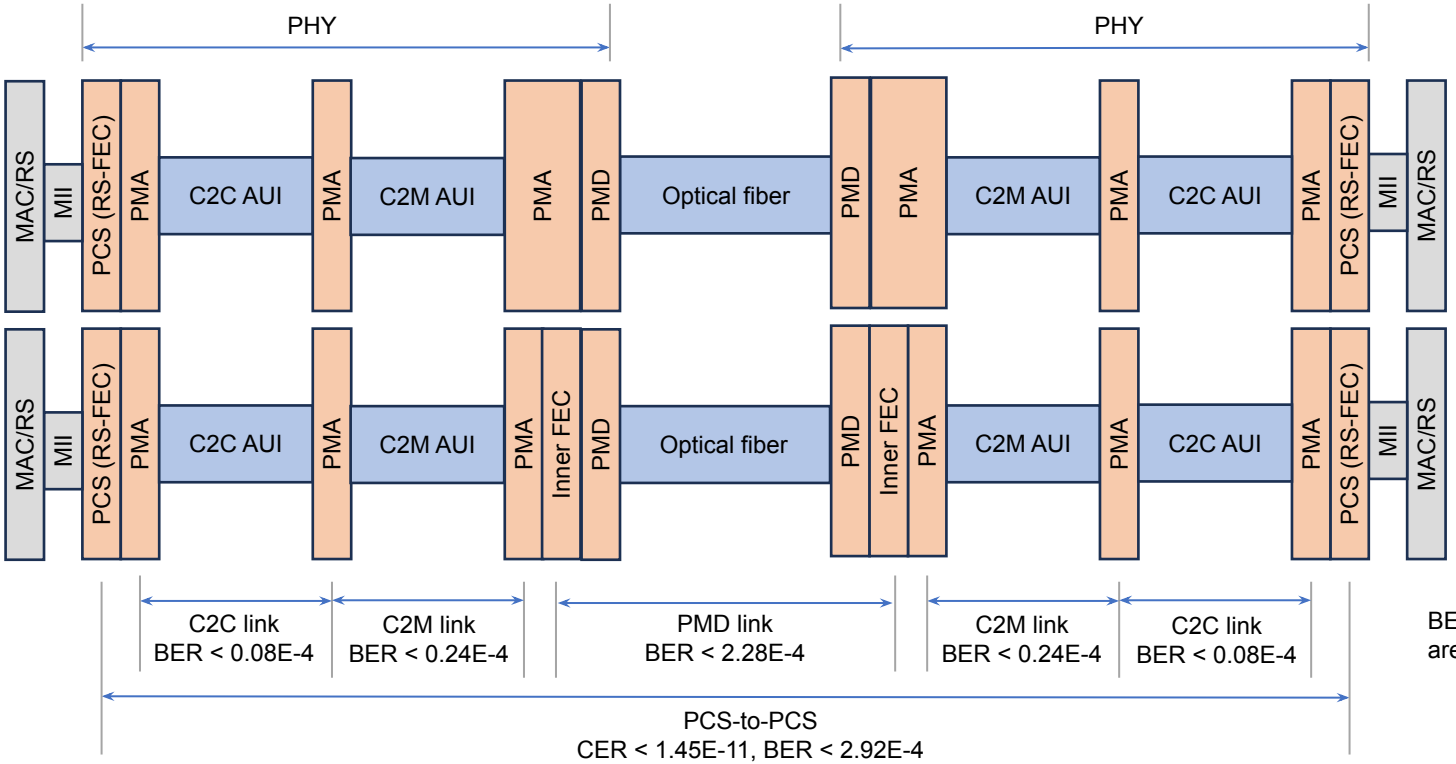
A suuporting presentation will be provided.

*Proposed Response* *Response Status* W

PROPOSED REJECT.
The comment proposes to change a decision made by the CRG as detailed in the comment. However, the comment makes a good case and a proposal is forthcoming. Pending task force review of the supporting contribution.

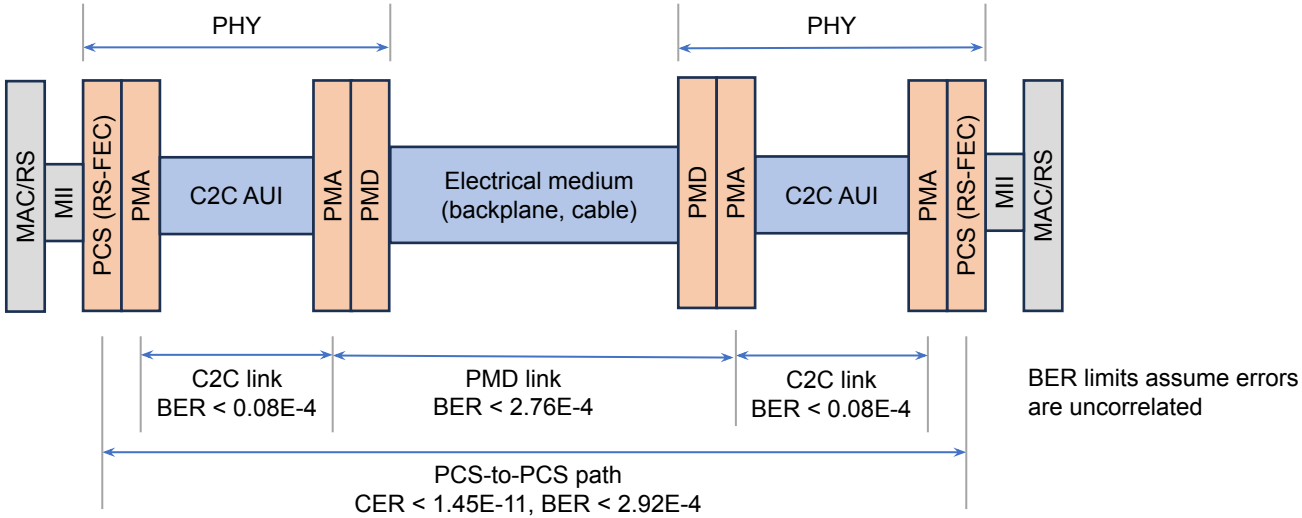
Comment #585 proposes to change the FLR budgeting for 800GBASE-LR1.
The appropriate diagram can be used once that decision is made.
Shown here for reference.

# 174A — Error ratio figures
# Proposed figure for optical PHY types




BER limits assume errors are uncorrelated

# 174A — Error ratio figures
# Proposed figure for electrical PHY types




BER limits assume errors are uncorrelated

# DATA/TRAINING mode
# Comments 191, 190, 192, 193, 195, 196, 198, 163, 166, 177

# DATA/TRAINING mode
# Comments 191, 190, 192, 193, 195, 196, 198, 163, 166, 177

Cl 179 SC 179.8.2 P 391 L 31 # 191

Huber, Thomas Nokia

*Comment Type* T *Comment Status* D *mon) DATA/TRAINING mode*

While it is clear what "DATA mode" is intended to mean here in the context of ILT, that term has specific meaning for 1000BASE-T PHYs that differs from what is intended here (see 1.4.278) Annex 178B.5 indicates that in the context of ILT, "data mode" means the variable tx_mode has the value 'data', which is associated with being in the PATH_UP state per figure 178B-8. As such, it would be more clear if the text in 179.8.2 referred to the PATH_UP state.

*SuggestedRemedy*

Change "When operating in DATA mode, ..." to "When operating in the PATH_UP state (see Figure 178B-8),..."

*Proposed Response* *Response Status* W

PROPOSED ACCEPT IN PRINCIPLE.
The two modes of the PMD transmit function are explicitly defined in the first paragraph of 179.8.2: "The PMD transmit function has two operating modes: DATA and TRAINING. The operating mode is controlled by the ILT function (see 179.8.9)". These modes are referenced in multiple places in the draft (although they are not currently defined by all PMDs).

The suggested remedy refers to a state of the training state diagram, but there is a variable, tx_mode, that explicitly controls the "DATA mode" behavior. This variable can be referenced to improve clarity.
Also, DATA and TRAINING modes of the transmit function should be defined for all PMDs that include an ILT function, and all references to these modes should be linked to the transmit function.

In the first pragraph of 179.8.2, change "The operating mode is controlled by the ILT function (see 179.8.9)" to "The operating mode is controlled by the tx_mode variable of the ILT function (see 179.8.9): it is DATA when tx_mode=data, and TRAINING otherwise".
Add similar paragraphs in 180.5.2, 181.5.2, 182.5.2, and 183.5.2 (possibly also 185.5.2 and 187.5.2 if ILT is added to these clauses).
Add an explicit reference to the transmit function in all instances of "DATA mode" and "TRAINING mode" across the draft, where appropriate.

Implement with editorial license.

**179.8.2 PMD transmit function**

The PMD transmit function has two operating modes: DATA and TRAINING. The operating mode is controlled by the ILT function (see 179.8.9).

When operating in DATA mode, the PMD transmit function shall convert the tx_symbol parameters of the PMD service interface message PMD:IS_UNITDATA_*i*.request on each lane into a stream of PAM4 symbols for transmission as electrical signals on the corresponding contacts of the MDI (see 179.12). The PAM4 symbol values 0, 1, 2, and 3 correspond to the tx_symbol values zero, one, two, and three, respectively. When operating in TRAINING mode, the PAM4 symbol stream on each lane is taken from the output of the training pattern generator in the PMD control function (see Figure 178B–4).

# DATA/TRAINING mode
# Comments 191, 190, 192, 193, 195, 196, 198, 163, 166, 177

Suggested change in 179.8.2

**179.8.2 PMD transmit function**

The PMD transmit function has two operating modes: DATA and TRAINING. The operating mode is controlled by the tx_mode variable of the ILT function (see 179.8.9): it is DATA when tx_mode = data, and TRAINING otherwise.

Similar paragraphs should be added in other "PMD transmit function" subclauses 180.5.2, 181.5.2, 182.5.2, and 183.5.2.
Example in 180.5.2:

**180.5.2 PMD transmit function**

The PMD transmit function has two operating modes: DATA and TRAINING. The operating mode is controlled by the tx_mode variable of the ILT function (see 180.5.12): it is DATA when tx_mode = data, and TRAINING otherwise.

~~The~~ When operating in DATA mode, the PMD Transmit function shall convert the *n* symbol streams requested by the PMD service interface messages PMD:IS_UNITDATA_0.request to PMD:IS_UNITDATA_*n*–1.request into *n* separate optical signals. The *n* optical signals shall then be delivered to the MDI, which contains *n* parallel light paths for transmit, according to the transmit optical specifications in this clause. The highest optical power level in each signal shall correspond to tx_symbol = three and the lowest shall correspond to tx_symbol = zero.

When operating in TRAINING mode, the PAM4 symbol stream on each lane is taken from the output of the training pattern generator in the PMD control function (see Figure 178B–4).

If ILT is added to coherent PMDs (175 and 187) then changes should be applied in 185.5.2 and 187.5.2 too.

Note that, in these PMDs, TRAINING mode does not use a PAM4 symbol stream; it has a different effect (send local_pattern). Appropriate modifications should be made.

Suggested change in 179.8.9

**179.8.9 Inter-sublayer link training (ILT) function**

A PMD shall provide the inter-sublayer link training (ILT) function for a Type E1 interface, specified in Annex 178B. When the variable mr_training_enable is true, the ILT function is used to request changes to the peer transmitter state (modulation, training pattern, and precoder state), control the transmitter output on each lane of the MDI, indicate the receiver state, and coordinate the transition of the PMD transmit function to DATA mode.

Similar changes should be made in all instances of "DATA mode".

# ILT state diagrams
# Comments 459, 626

# ILT state diagrams
# Comments 459, 626

*Cl* **178B** *SC* **178B.14.2.1** *P* **804** *L* **32** # 459

Slavick, Jeff — Broadcom

*Comment Type* **TR** *Comment Status* **D** *(Common) ILT state diagrams*

Training status can not be both a AUI component variable and a per-lane training variable. Local_rts is an equivalent status to it and is mapped to a MDIO register bit.

*SuggestedRemedy*

Move the definition of training_status to 178B14.3.1
Remove the enumeration of "READY" from its definition.
Delete training_status <= READY from Figyre 178B-7

*Proposed Response* *Response Status* **W**

PROPOSED ACCEPT IN PRINCIPLE.
training_status is used by the PMDs and AUIs (see 178.4, 179.4, 180.3, 181.3, 182.3, 183.3, 176C.6 and 176D.4) so it shall be assigned a value by ILT.
It is a per-interface variable that is assigned to all lanes of the interface.
Define a new variable in 178B.14.3.1: lane_training_status. Defined as: Enumerated variable that indicates the status of the per-lane ILT function. This variable may be assigned one of the following values: IN_PROGRESS, OK, FAIL.
Use this new variable in the per-lane state diagrams instead of training_status.
Change the definition of the variable training_status to: Enumerated variable that indicates the status of the per-interface ILT function. This variable may be assigned one of the following values: IN_PROGRESS, READY, OK, FAIL. The value READY is assigned by the RTS update state diagram (Figure 178B-8) and other values are assigned according to the lane_training_status variable (see 178B.14.3.1):
IN_PROGRESS - lane_training_status variable = IN_PROGRESS for any lane assigned to the interface
OK - lane_training_status variable = OK for all lanes assigned to the interface
FAIL - lane_training_status variable = FAIL for any lane assigned to the interface
Implement with editorial license.

*Cl* **178B** *SC* **178B.14.3.5** *P* **810** *L* **7** # 626

Law, David — HPE

*Comment Type* **TR** *Comment Status* **D** *(Common) ILT state diagrams*

The variable training_status is used by the 'Training control state diagram' in subclause 178B.14.3.5 'State diagram figures' but is not defined in the associated subclause 178B.14.3.1 'Variables'.

In addition, it appears that the training_status is a per-interface variable based on the definition found in 178B.14.2.1 'Variables', yet it appears to be driven by both the per-interface 'RTS update state diagram' (Figure 178B–7) and the per-lane 'Training control state diagram' (Figure 178B–8). I'm not sure how this would operate.

As an example, if the Training control state diagram on one lane in an interface enters the FAIL state, it would set training_status for the interface to FAIL. If, however, the Training control state diagram on another lane in the same interface enters the PATH_UP state immediately afterwards, training_status for the interface would then be set to OK. This doesn't seem to be correct.

*SuggestedRemedy*

Provide a definition for the training_status variable used in Figure 178B–8 'Training control state diagram' in its associated subclause 178B.14.3.1 'Variables'. In addition, clarify the operation of training_status regarding it being driven by both the per-interface 'RTS update state diagram' (Figure 178B–7) and the per-lane 'Training control state diagram'.

*Proposed Response* *Response Status* **W**

PROPOSED ACCEPT IN PRINCIPLE.
Resolve using the response to comment #459.

# ILT state diagrams
# Comments 459, 626

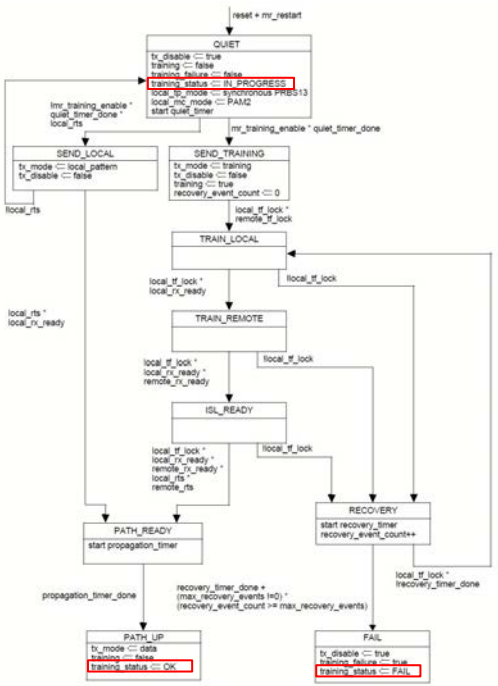



Figure 178B–8—Training control state diagram

Per-lane state diagram

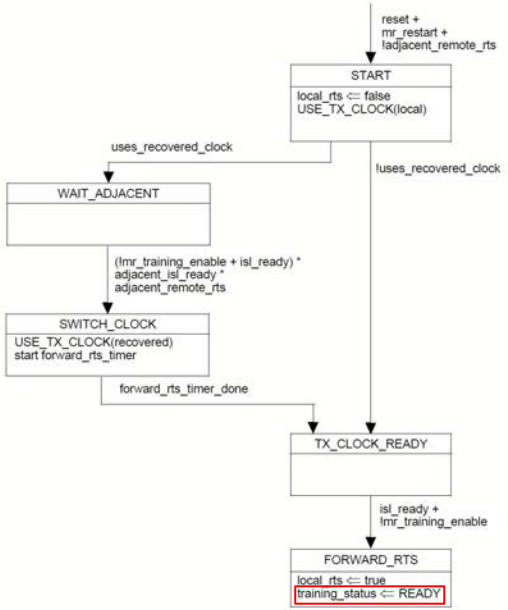



Figure 178B–7—RTS update state diagram

Per-interface state diagram

training_status

Enumerated variable that indicates the status of the ILT function. This variable may be assigned one of the following values: IN_PROGRESS, READY, OK, FAIL.

Problem:

training_status is defined as a per-interface variable (in 178B.14.2.1), but it appears in both per-lane and per-interface state diagrams. Formally there needs to be more than one variable.

Note that the value READY is assigned only be the per-interface state diagram; READY is equivalent to (all lanes are IN_PROGRESS) and (local_rts is true).

# ILT state diagrams
# Comments 459, 626

**Proposed changes:**

Define a new variable in 178B.14.3.1 (Per-lane variables): lane_training_status

lane_training_status

Enumerated variable that indicates the status of the training control state diagram. This variable may be assigned one of the following values: IN_PROGRESS, OK, FAIL.

Use this new variable in the per-lane state diagrams instead of training_status.

**For the per-interface variable training_status:**

**Option A**: keep it as a state diagram variable and change its definition as follows:

training_status

Enumerated variable that indicates the status of the ILT function. This variable may be assigned one of the following values: IN_PROGRESS, READY, OK, FAIL. Assignment to this variable occurs both by the RTS update state diagram (Figure 178B-8) and by changes to the lane_training_status variables of the lanes in the interface, as follows:

- IN_PROGRESS: assigned when lane_training_status = IN_PROGRESS on all lanes.
- READY: assigned by the RTS update state diagram.
- OK: assigned when lane_training_status variable = OK on all lanes
- FAIL: assigned with lane_training_status variable = FAIL on any lane

**Option B**: delete its assignment from the per-interface state diagram (Figure 178B-8), making it a non-state-diagram variable, and use the following definition:

training_status

Enumerated variable that indicates the status of the ILT function. This variable may be assigned one of the following values: IN_PROGRESS, READY, OK, FAIL. This variable is assigned as follows:

- IN_PROGRESS: if lane_training_status = IN_PROGRESS on all lanes and local_rts = false
- READY: if lane_training_status = IN_PROGRESS on all lanes and local_rts = true
- OK: if lane_training_status = OK on all lanes
- FAIL: if lane_training_status = FAIL on any lane

# ILT 178B — Adjacent service interface Comment #123, 448

# ILT 178B — Adjacent service interface Comment #123, 448

Cl 178B SC 178B.14.2.1 P 803 L 46 # 123

Mascitto, Marco Nokia

*Comment Type* E *Comment Status* D *(Common) ILT adjecency*

This is not very clear. I would suggest adding the definition of "adjacent service interface" in subclause 178B.3.

*SuggestedRemedy*

I would suggest adding the definition of "adjacent service interface" to subclause 178B.3 and referencing a diagram, like the one on Slide 3 of "Making Sense out of ILT" (J. D'Ambrosia, M. Brown, 802.3dj Joint Ad hoc Mtg - 05 Jun 2025).

Adjacent service interface
The service interface adjoining a PMD or AUI component to a PMA.

*Proposed Response* *Response Status* W

PROPOSED ACCEPT IN PRINCIPLE.

The term "adjacent service interface" is not clearly defined.

Editorial slides will be provided to address this.

Resolve along with comment #448.

Cl 178B SC 178B.14.2.1 P 803 L 47 # 448

Ran, Adee Cisco Systems

*Comment Type* T *Comment Status* D *(Common) ILT adjecency*

The second case in the NOTE says: "For ILT in an AUI component above a PMA, the adjacent service interface is the interface below the AUI component". That is the PMA's service interface. It may be easier to understand if it is stated.
Also, a figure illustrating the two cases would be helpful.

*SuggestedRemedy*

Change "the adjacent service interface is the interface below the AUI component" to "the adjacent service interface is the PMA service interface (below the AUI component)".
Add a figure, with editorial license.

*Proposed Response* *Response Status* W

PROPOSED ACCEPT IN PRINCIPLE.

Clarification of the term "adjacent service interface" is not clearly defined.

Editorial slides will be provided to address this.

Resolve along with comment #123.

**178B.14.2.1 Variables**

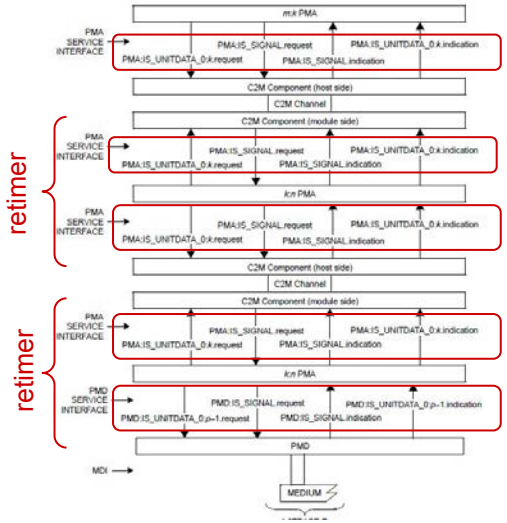
adjacent_signal_ok

Enumerated variable derived from the value of the SIGNAL_OK parameter on the adjacent service interface. It takes one of the following values: IN_PROGRESS, READY, OK, FAIL.

NOTE — For ILT in a PMD or an AUI component below a PMA, the adjacent service interface is the service interface of the PMD or AUI component, and SIGNAL_OK is received via the IS_SIGNAL.request primitive. For ILT in an AUI component above a PMA, the adjacent service interface is the interface below the AUI component, and SIGNAL_OK is received via the IS_SIGNAL.indication primitive.

adjacent_remote_rts

Boolean variable that indicates the value of remote_rts on the adjacent service interface. It is true if adjacent_signal_ok is OK and false otherwise.




There is only one service interface on an AUI component or PMD, so it is not necessary to qualify it with the word “adjacent”.

In fact, the word adjacent is used to reference the AUI component or PMD on the other side of a retimer.

Change the note in 178B.14.2.1 to:

NOTE — For a PMD or an AUI component below a PMA, SIGNAL_OK is received via the IS_SIGNAL.request primitive of the service interface above the PMD or AUI component. For an AUI component above a PMA, the SIGNAL_OK is received via the IS_SIGNAL.indication primitive of the service interface below the AUI component.

Change other references of “adjacent service interface” to “service interface”.

# 174A – terminology Comment #52

# 174A – terminology Comment #52

CI 178B SC 178B.3 P 786 L 33 # 52

D'Ambrosia, John Futurewei, U.S. Subsidiary of Huawei

*Comment Type* E *Comment Status* D *(Common) ILT scope*

Given the introduction of inter-sublayer link training to the Ethernet world, it would be helpful if the term inter-sublayer link (ISL) was displayed graphically for the reader.

*SuggestedRemedy*

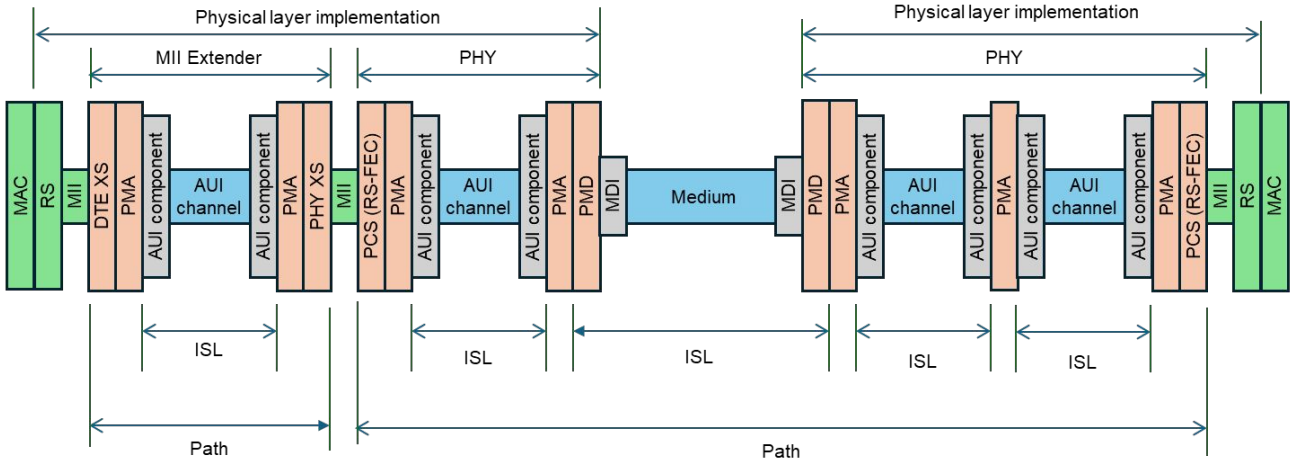
Implement figure on Page 3 of https://www.ieee802.org/3/dj/public/adhoc/electrical/25_0605/dambrosia_3dj_elec_02_250605.pdf with editorial license

*Proposed Response* *Response Status* W

PROPOSED ACCEPT IN PRINCIPLE.
Pending review of the following presentation and CRG discussion.
<URL of presentation>

A related presentation has not been requested (yet).
However, this slide provides an figure, update since presented to the ad hoc.
The diagram captures the various entities as defined for ILT in D2.0.
Provided to the editorial team by John D'Ambrosia.
This diagram is provided as a reference for discussion and as a proposed diagram to add into Annex 178B.

As shown, the diagram includes two path types: XS-to-XS in an xMII extender and PCS-to-PCS across a pair of PHYs and the medium between.



# Comments #421

Leon Bruckman, Nvidia

# Comment

*Cl* **178B** *SC* **178B.5.3** *P* **789** *L* **44** # 421

Ran, Adee — Cisco Systems

*Comment Type* **TR** *Comment Status* **D** *(Common) ILT extender*

The text about training xMII extenders does not address the communication of the status variables isl_ready and remote_rts between interfaces (PMD to AUI and vice versa) when there is a PHY XS and PCS between them.
Ideally, this communication should be the same as the one defined in 178B.14.2.1 using adjacent_signal_ok, but the case of an extender is not covered by NOTE that describes what "adjacent" is.

Since this behavior is specific to PHYs attached to extenders, it should be specified in this subclause, preferably with a diagram.

*SuggestedRemedy*

Add a NOTE in 178B.5.3 stating that, for the purpose of adjacent_signal_ok, the adjacent interface of a PMD in a PHY attached to an xMII extender is the service interface of the PHY XS; and the adjacent interface of the AUI component above the PHY XS is the service interface of the PMD.
Add a figure to illustrate the communication of adjacent_signal_ok between the PMD and the AUI (across the PCS and PHY XS, and possibly other sublayers).

*Proposed Response* *Response Status* **W**

PROPOSED ACCEPT IN PRINCIPLE.
Implement suggested remedy with editorial license.

# D2.0

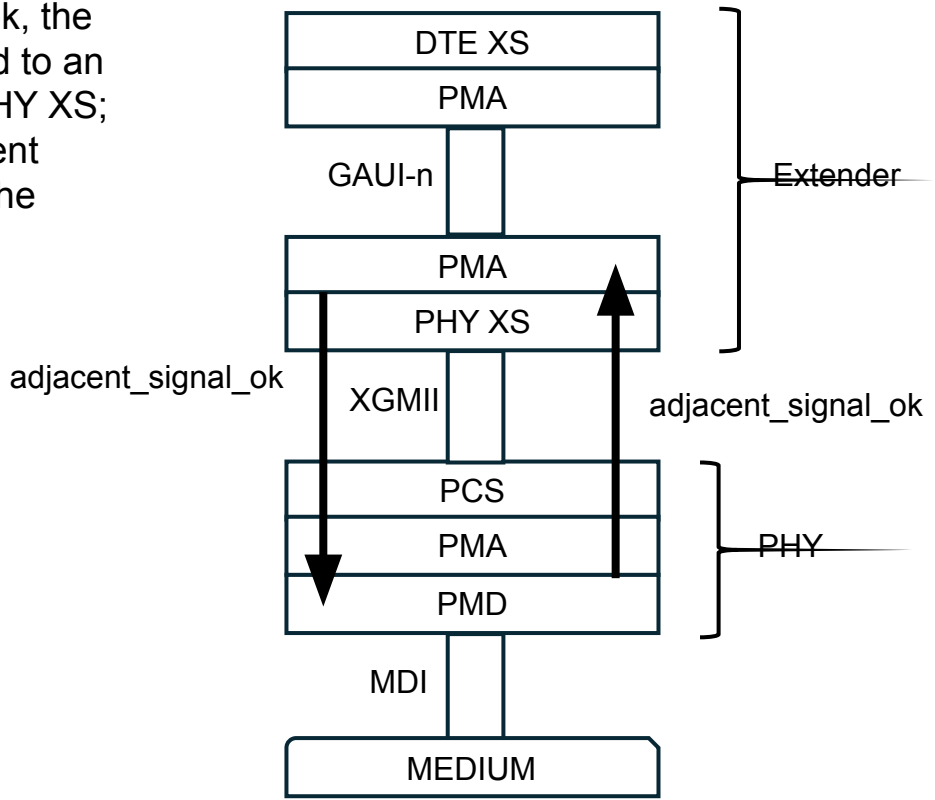
## 178B.5.3 Training xMII Extenders

Training on an ISL within an xMII Extender follows the same process, such that local_rts and remote_rts are communicated to the PHY XS using its IS_SIGNAL.indication and IS_SIGNAL.request primitives.

AUI components within an xMII Extender may train before or in parallel with the PCS to PCS path, and training signaling will continue until the main path is ready. This is the same behavior as AUI components within a PHY.

# Proposed response – Option 1

NOTE – For the purpose of adjacent_signal_ok, the adjacent interface of a PMD in a PHY attached to an xMII extender is the service interface of the PHY XS; and the adjacent interface of the AUI component above the PHY XS is the service interface of the PMD.



# Proposed response – Options 2 and 3

- Option 2
  - ILT is done separately on extenders, and RTS is not communicated to the RS-to-RS link – such that extenders create separate paths
  - As can be interpreted from D2.0 text
- Option 3
  - Either Option 1 or Option 2 is allowed (left open for implementation)

# ILT overview

Comments 220, 290, 374, 498, 553, 116

# ILT overview - text in question

### 178B.2 Overview

ILT for electrical and optical interfaces performs the path start-up protocol. ILT facilitates timing recovery and equalization in a ISL or multi-ISL paths while providing a mechanism through which the receiver may configure the transmitter to optimize performance. ILT supports these functions through the continuous exchange of fixed-length training frames between peer interfaces in an ISL and the coordination of the ISLs along a path by transporting end-to-end indications.

The overview is short and focuses on training frames. It does does not point to 178B.5 where the path start-up aspects are described in more detail.

### 178B.5 Path start-up protocol

ILT enables independent ISL training in a multi-ISL path that includes AUI components and PMDs. It also supports operation over paths that include ISLs that do not implement ILT.

Path start-up is achieved as follows:

— local_rts indicates that an AUI component or PMD is ready to send and receive normal data and propagates from the PCS at one end of the path towards the PCS at the other end of the path.

— remote_rts indicates that a remote AUI component or PMD is ready to send and receive normal data and propagates similarly and independently from PCS to PCS in both directions.

— local_rts and remote_rts are propagated only across an ISL that is ready to send data.

— When a device both sends local_rts and receives remote_rts in both directions, it means all the ISLs in the same path (see 178B.3) are ready and it switches to data mode (tx_mode = data, see 178B.14.3.1).

— When all devices are in data mode, communication on the path is established.

Ready to send (RTS) propagates over ISLs using one of the following methods:

— The Continue training bit in the control field of the training frames (see 178B.7.2) if training is enabled.

— The transmit disable and the AUI component or PMD signal detect function otherwise.

NOTE — Interactions with AUI components and PMDs that do not support ILT as specified in this annex (e.g. those defined in Clause 120 or Clause 173) use the second method.

More details in 178B.5.1 through 178B.5.3

# ILT overview
# Comments 220, 290, 374, 498, 553, 116

| | |
|---|---|
| 220 | The overview of ILT is confusing. ILT has two aspects - there is per-ISL training, and there is the end-to-end path startup behavior. These need to be more clearly separated in the overview text. The "continuous exchange of fixed-length training frames" is not entirely accurate - that may be what happens during the training phase, but is certainly not what happens once the training is completed. |
| 290 | The term inter-sublayer link training (or ILT) by name defines a protocol over an intersublayer link (or ISL). Each ISL is one of several possible physical links between a pair of MAC sublayers. It is possible only a subset of the ISLs supports ILT. Annex 178B also defines a path start-up protocol which uses the outcome of ILT on each of the physical links, where supported, to determine when the path between a pair of PCSs or between a pair of extender suppliers is ready, allowing for some ISLs that do not support ILT.<br>However, the combination of these two layers of functionality are references only as ILT. This is confusing! |
| 374 | 3 major functions are included in the ILT: Electrical LT, Optical LT, and inter-sublayer link signal or RTS. Designating everting as ILT is rather confusing throughout the draft. |
| 424 | There should be a distinction between "ILT", which is a protocol on a single ISL, and the end-to-end (RS-to-RS) path bring-up procedure. The latter is an ability that is enabled by the former, but is system-level result, while ILT is a local mechanism.<br>Additional terminology may be helpful, e.g. "Physical layer startup procedure". |
| 498 | Change "in a ISL or multi-ISL paths" to "in a ISL path or multi-ISL paths" |
| 553 | The description "ILT supports these functions through the continuous exchange of fixedlength training frames between peer interfaces in an ISL" indicates training frames are continuously exchanged. The presumed purpose to be contiuous would be for the AUI components to update their equalization coeficients yet there is no desription of returning to training such as with recovered clock while continuing to carry real traffic nor is there status indicators that updated training is occurring. |
| 116 | Replace: "ILT enables independent ISL training in a multi-ISL path that includes AUI components and PMDs. It also supports operation over paths that include ISLs that do not implement ILT".<br>With "ILT supports independent training of ISLs in a multi-ISL path. ILT also operates over paths that include ISLs that do not support ILT". |

# ILT overview
# Comments 220, 290, 374, 498, 553, 116

Suggested rewrite of 178B.2:

## 178B.2 Overview

ILT is an internal function of physically instantiated interfaces (PMDs and AUI components) that controls the establishment of communication with a peer interface within an inter-sublayer link (ISL).

ILT functions in each sublayer utilize service interface primitives to communicate the status of each ISL to adjacent sublayers. Through this communication, ILT creates a well-defined path start-up process for paths that include one or more ISLs. Initially all ISLs are in TRAINING mode, in which the data sent to the peer is generated locally by each interface. The ILT function provides coordinated transition of all ISLs to DATA mode, in which data is communicated across interfaces between the endpoints of the path.

For interfaces that use PAM4 signaling, ILT includes a training protocol, used in TRAINING mode, that facilitates clock and data recovery and enables control of peer transmitter settings by exchange of fixed-length training frames between peer interfaces. Two versions of the training protocol are defined, E1 and O1. ILT can also establish communication between interfaces that do not use a training protocol.

Terminology used by ILT is defined in 178B.3. The behavior of ILT within a specific interface is described in 178B.4. The path startup process is described in 178B.5. 178B.6 through 178B.13 specify the E1 and O1 training protocols for PAM4 interfaces. The state diagrams in Figure 178B–7 and Figure 178B–8, and their associated variables defined in 178B.14, apply for all interfaces that include an ILT function, with or without a training protocol.

# ILT overview
# Comments 220, 290, 374, 498, 553, 116

Suggested change in 178B.5:

### 178B.5 Path start-up process

ILT enables establishment of communication independently in each ISL within a path consisting of one of more ISLs, that can include AUI components and PMDs, and includes a training protocol that enables optimization of transmitter settings. The path can include ISLs that do not use a training protocol.

The status of each ISL is communicated to adjacent sublayers using service interface primitives. This enables start-up of the whole path by coordinated transition of all interfaces in the path from TRAINING mode to DATA mode.

Path start-up is achieved as follows:

< existing text >

Additional editorial changes in 178B.3 through 178B.5 (which are all introductory subclauses) may be beneficial.
Implement with editorial license.