APSU direction for 802.3dv

Leon Bruckman, Nvidia

Luz Osorio, Nokia

Contributors

- Kent Lusted, Synopsys
- Matt Brown, Qualcomm

Supporters

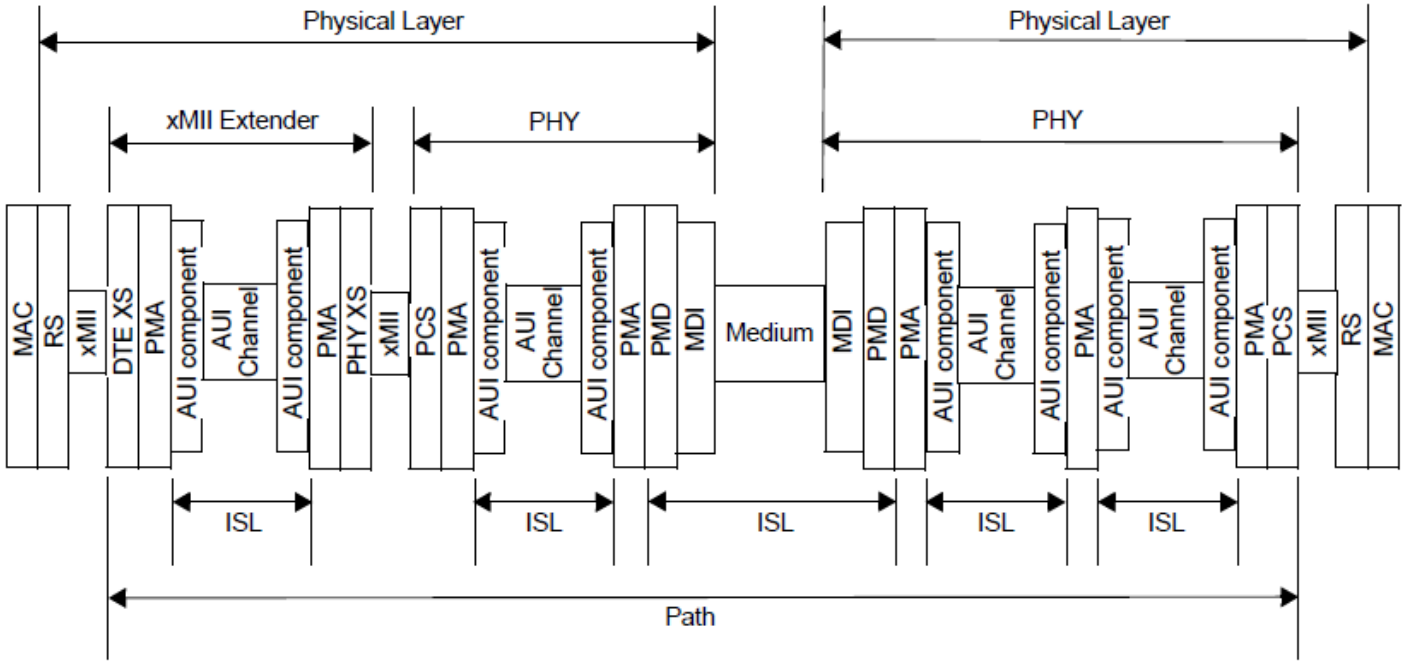
- Matt Brown, Qualcomm
- Marco Mascitto, Nokia
- Tom Huber, Nokia
- Adee Ran, Cisco

Introduction

- This presentation provides an overview of Autonomous Path StartUp (APSU) as defined in 802.3dj Annex 178B.
- The presentation covers the key definitions, an overview of the RTS and ILT functions, and the training frame formats (E1 electrical and O1 optical) from IEEE 802.3dj Annex 178B.
- The authors request that the task force consider defining APSU for 400GPL interfaces, starting with APSU as defined in IEEE 802.3dj Annex 178B, and study possible enhancements.

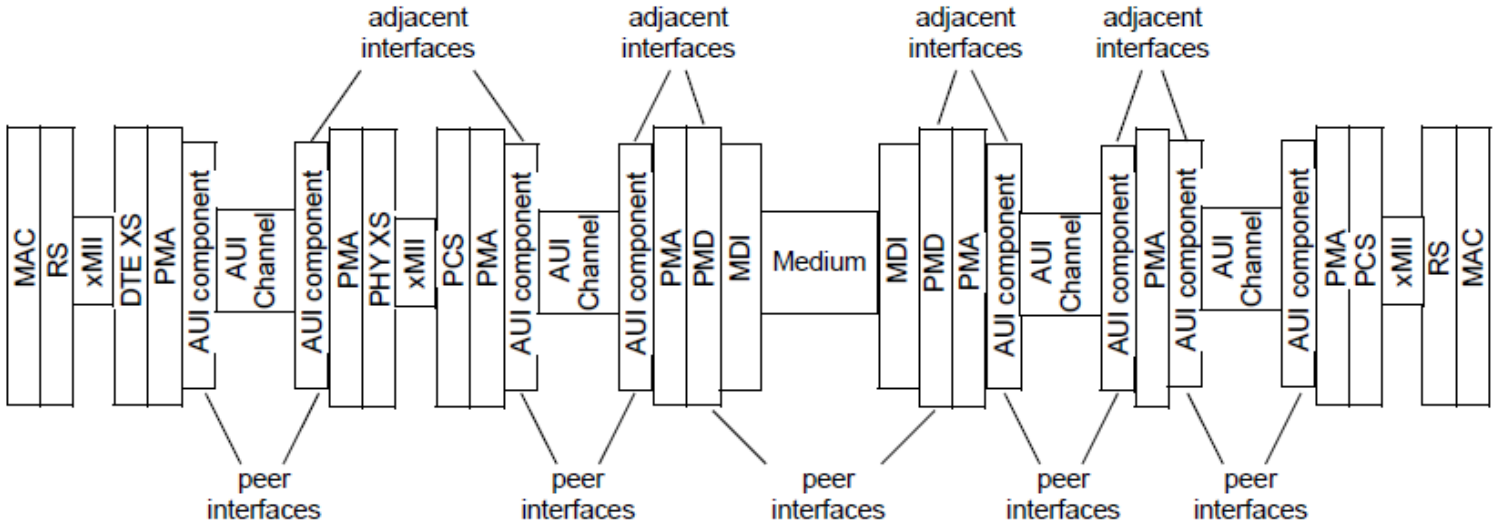
Definitions (1)

- **Path:** The sequence of ISLs and intermediate sublayers providing the connectivity between two Reconciliation Sublayers (RSs).
- **Inter-sublayer link (ISL):** Either a pair of AUI components and the AUI channel between, or a pair of PMDs (in different PHYs) and the medium between.
- **Interface:** Unless qualified otherwise, interface is either an AUI component or a PMD.
- **AUI component:** Either a C2C component or a C2M component.



Definitions (2)

- **Adjacent interface**: An interface whose service interface is connected with this interface via one or more intermediate sublayers.
- **Peer interface:** The interface at the other end of the ISL. In the case where the ISL is between a pair of PMDs, this term is equivalent to link partner.



What is Autonomous Path StartUp – APSU?

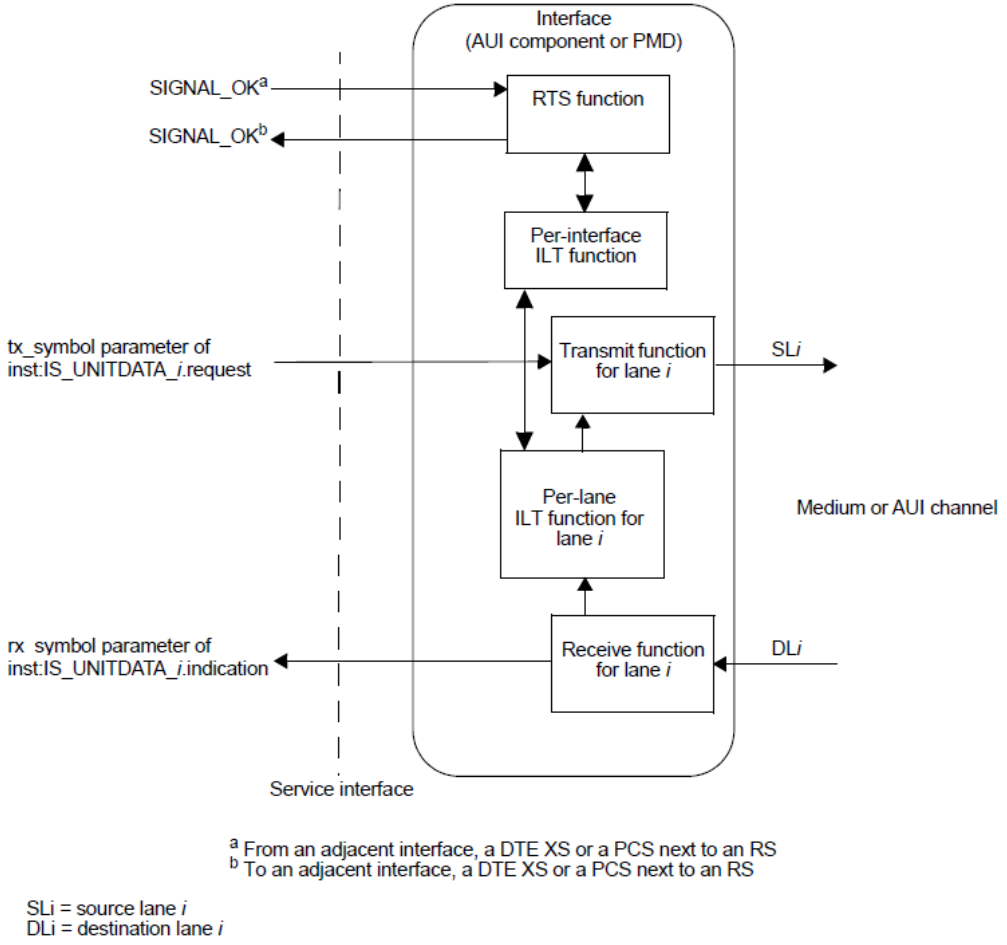
- APSU is the coordinated, orderly initialization of all ISLs and sublayers in a path.
  - APSU is not intrinsically a function, it is an externally observable behavior resulting from establishing communication through all sublayers in the path in both directions.
- APSU is facilitated by the ready-to-send (RTS) function and, where required, the inter–sublayer link training (ILT) function.
- The ILT function is only defined for PAM4 interfaces, while the RTS function is defined for PAM4 interfaces and coherent interfaces

Ready-to-send (RTS) function

- The RTS function provides the communication between end points that relays the status of the individual ISLs for the purpose of bringing the link up (ready-to-send status).
- Controls the switchover of the transmitter clock source to its DATA mode clock.
- Facilitates the indication of the local interface readiness to switch to DATA mode (local_rts) to other interfaces (peer and adjacent).
- There is an RTS function defined for PAM4 interfaces.
- There is an RTS function defined for coherent interfaces.

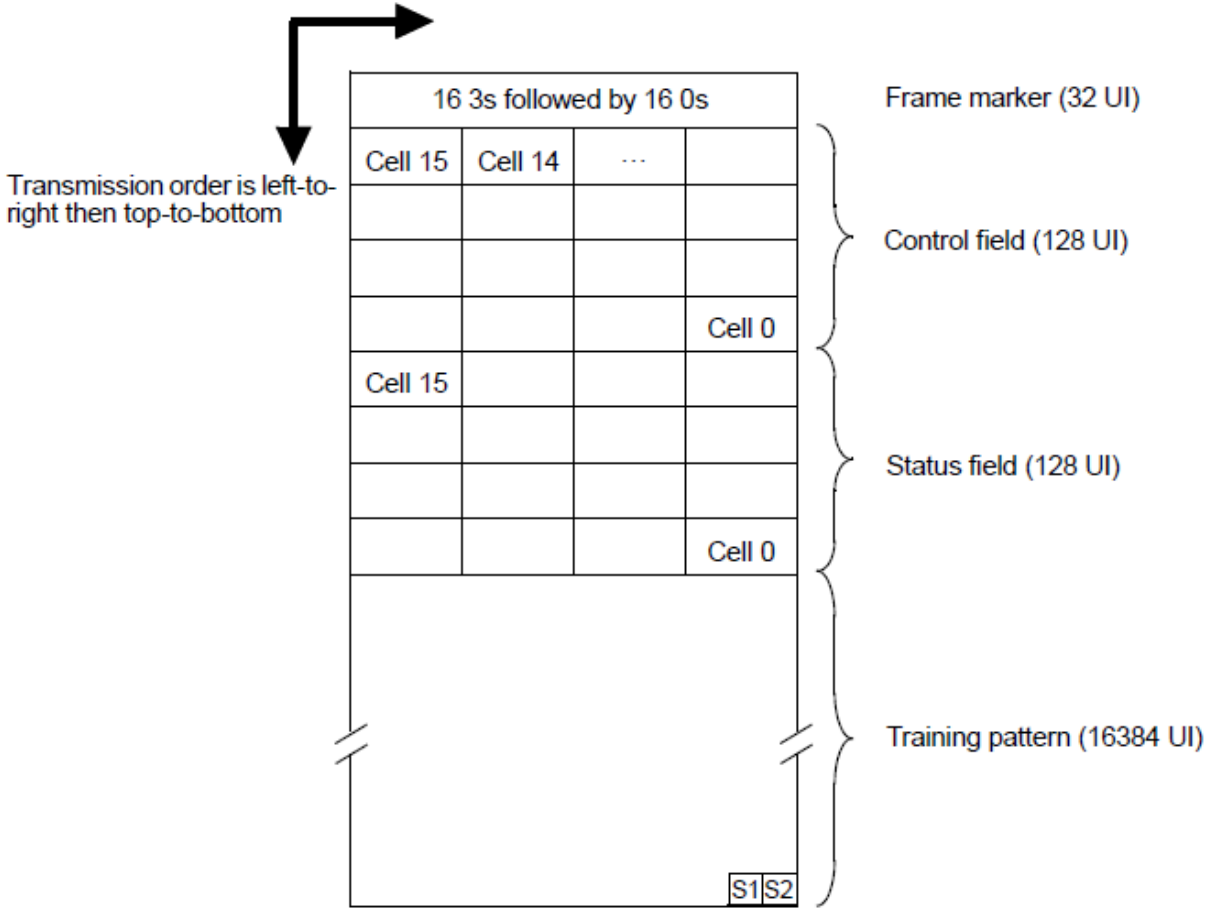
ILT function for PAM4 interfaces

- ILT optimizes PAM4 signaling.
- Establishes communication between peer PAM4 interfaces, including clock and data recovery.
- Can adjust transmitter parameters during training using ILT training frames.
- There is a separate training frame format for electrical interfaces (E1) vs. optical interfaces (O1)
- Facilitates the transfer of the RTS status of the transmitter
- There is a per-interface ILT function
- There is a per-lane ILT function



Training frame structure

- Frame marker
  - Training frames are delimited by a specific sequence of PAM4 symbols. The training frame marker is a run of 16 consecutive “3” symbols followed by a run of 16 consecutive “0” symbols.
- Control and status fields
  - The control field is composed of 16 bits.
  - Each bit of the control and status fields is sent as a differential Manchester encoded (DME) cell, where each cell is eight-unit intervals in length.
- Training pattern
  - Training pattern is either PAM2, PAM4 or PAM4 with precoding modulated.
  - There are three training pattern generator functions: synchronous PRBS13, free-running PRBS13, and freerunning PRBS31



Overhead E1 training frame format

Control field

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	Not ready to send	1 = Not ready to send 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

Status field

Bit(s)	Name	Description
15	Receiver ready	1 = Training is complete and the receiver is ready for data 0 = Request for training to continue
14	Reserved	Transmit as 0, ignore on receipt
13:12	Training pattern status	13 12 1 1 = free-running PRBS31 1 0 = Reserved 0 1 = free-running PRBS13 0 0 = synchronous PRBS13
11:10	Modulation and precoding status	11 10 1 1 = PAM4 with precoding 1 0 = PAM4 without precoding 0 1 = Reserved 0 0 = PAM2
9	Receiver frame lock	1 = Frame boundaries identified 0 = Frame boundaries not identified
8	Initial condition status	1 = Updated 0 = Not updated
7	Parity	Even parity bit
6	Reserved	Transmit as 0, ignore on receipt
5:3	Coefficient select echo	5 4 3 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
2:0	Coefficient status	2 1 0 1 1 1 = Reserved 1 1 0 = Coefficient at limit and equalization limit 1 0 1 = Reserved 1 0 0 = Equalization limit 0 1 1 = Coefficient not supported 0 1 0 = Coefficient at limit 0 0 1 = Updated 0 0 0 = Not updated

Overhead O1 training frame format

Control field

Bit(s)	Name	Description
15:11	Reserved	Transmit as 0, ignore on receipt
10	Not ready to send	1 = Not ready to send 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
4:0	Reserved	Transmit as 0, ignore on receipt

Status field

Bit(s)	Name	Description
15	Receiver ready	1 = Training is complete and the receiver is ready for data 0 = Request for training to continue
14	Reserved	Transmit as 0, ignore on receipt
13:12	Training pattern status	13 12 1 1 = free-running PRBS31 1 0 = Reserved 0 1 = free-running PRBS13 0 0 = synchronous PRBS13
11:10	Modulation and precoding status	11 10 1 1 = PAM4 with precoding 1 0 = PAM4 without precoding 0 1 = Reserved 0 0 = PAM2
9	Receiver frame lock	1 = Frame boundaries identified 0 = Frame boundaries not identified
8	Reserved	Transmit as 0, ignore on receipt
7	Parity	Even parity bit
6:0	Reserved	Transmit as 0, ignore on receipt

APSU examples

- Examples of APSU with training enable and disable can be found in:
- https://www.ieee802.org/3/dj/public/adhoc/178b/25_1029/slavick_178b_01a_251029.pdf
- https://www.ieee802.org/3/dj/public/adhoc/178b/25_1029/slavick_178b_02a_251029.pdf
- https://www.ieee802.org/3/dj/public/adhoc/178b/25_1029/slavick_178b_03a_251029.pdf

Considerations for future work

Considerations for APSU:

- Expand the optical training capabilities
- Support for linear and RTLR architectures
- Improve efficiency
- PAM6 adaptations

Summary

- 400GPL can benefit from APSU functionality (ILT and RTS).
- Consider starting with APSU as defined in IEEE 802.3dj Annex 178B
- Consider future enhancements
  - Reach out with ideas

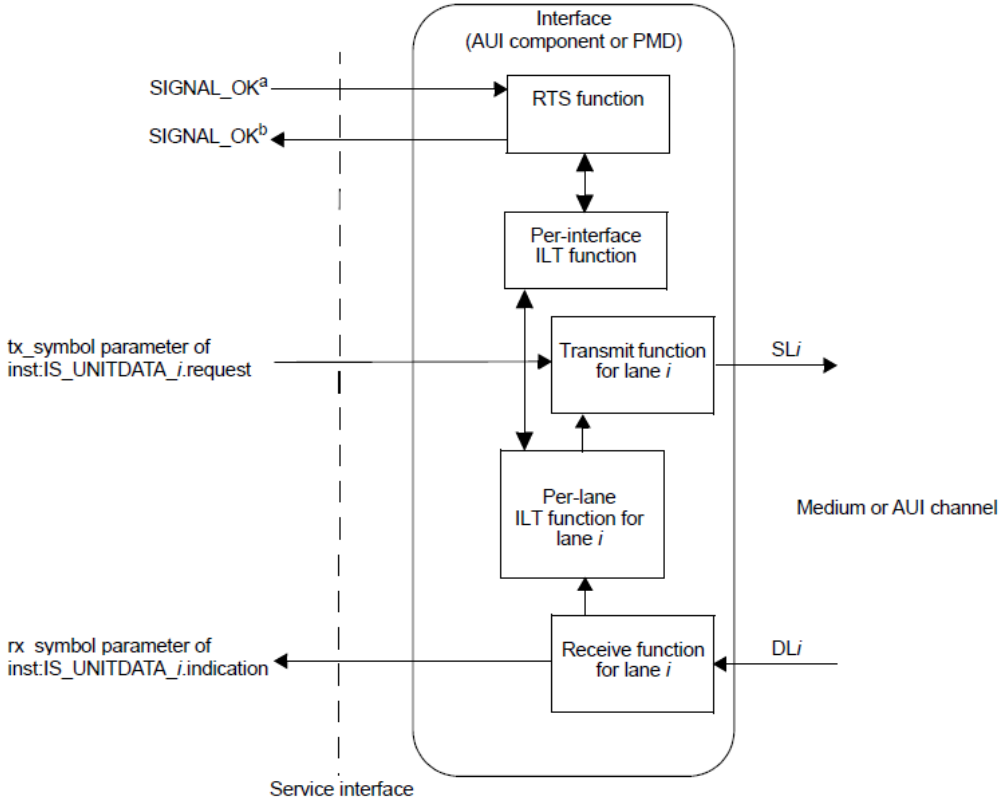
Thanks

RTS … in more detail

- The behavior of the RTS function is as follows:
  - Initially, the local_rts variable is set false.
  - Wait for the adjacent interface to be ready for DATA mode, as indicated by the SIGNAL_OK parameter from the adjacent interface being equal to READY or OK.
  - If the ILT function is operating in TRAINING mode (mr_training_enable = true), wait for local ISL to complete its adaptation, as indicated by training_status equal to READY or OK.
  - If the interface is implemented in a retimer that uses the recovered clock in DATA mode, initiate the swap to the recovered clock from the adjacent interface.
  - Set the local_rts variable to true, indicating readiness to move to DATA mode.
- Forwarding RTS
  - PAM4 interfaces indicate their RTS status using a bit in the ILT frame control overhead
  - Coherent interfaces indicate their status in two ways:
    - 800GBASE-LR1:
      - Not RTS: Squelch
      - RTS: Unsquelch. Transmit a FILL signal
    - 800GBASE-ER1: The RTS status is signaled to the peer node via the MNT overhead filed

ILT function for PAM4 interface … in more detail

- There is a per-interface ILT function
  - The per-interface ILT function monitors all of the independent per-lane ILT functions, provides summary status information to the RTS function, and relays RTS states to the per-lane ILT functions.
- There is a per-lane ILT function
  - The per-lane ILT function is responsible for the locally generated data that is transmitted while not in DATA mode and for the adaptation processes of the received signal.
  - Each lane of the interface runs an independent instance of the per-lane ILT function.



[a] From an adjacent interface, a DTE XS or a PCS next to an RS
[b] To an adjacent interface, a DTE XS or a PCS next to an RS

SLi = source lane *i*
DLi = destination lane *i*