A 400GbE Architectural Option

IEEE P802.3bs 400 Gb/s Ethernet Task Force

May 2014 Norfolk

Pete Anslow - Ciena
Hugh Barrass – Cisco
John D'Ambrosia – Dell
Mark Gustlin – Xilinx
Adam Healey – Avago
David Law – HP
Gary Nicholl - Cisco
Dave Ofelt – Juniper

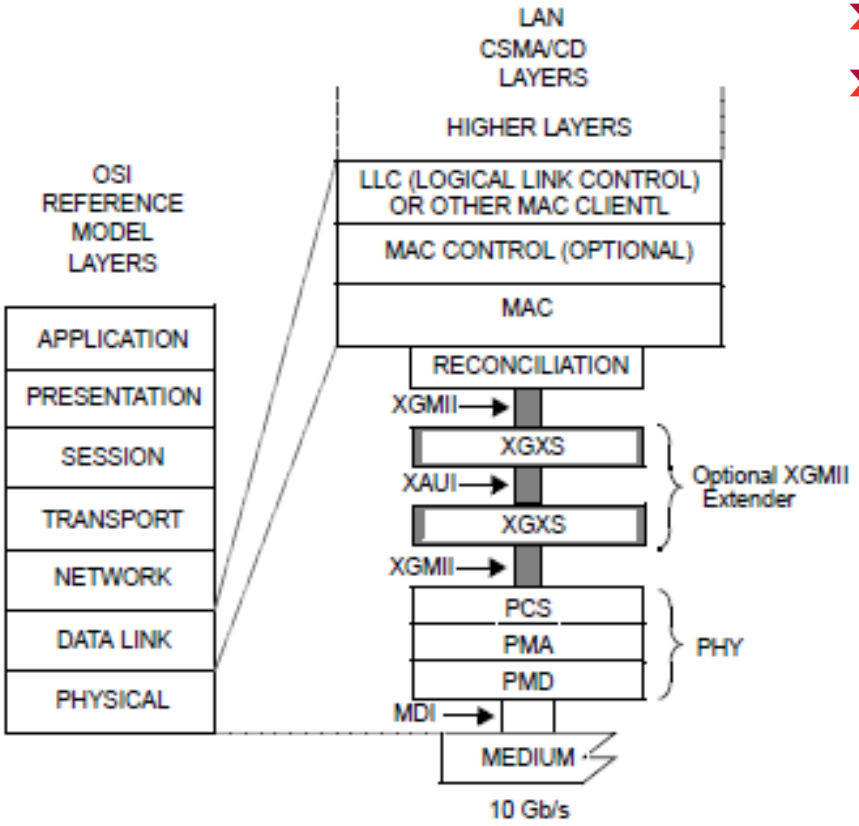
Agenda

- Requirements of the Architecture
- Review of the 10GbE and 100GbE architectures
- A possible 400GbE architecture
- Possible FEC strategies
- Possible example implementations

What Needs to be Supported in the Architecture?

- The coding needs of the electrical interface will vary independently from the PMD interface
- The requirements for each interface can be different, both the FEC, modulation and number of lanes can change over time for each interface
- We need a single high level architecture which can support the evolving requirements of the interfaces over time
  - This does not mean it is requires a complicated implementation

10GbE Architecture

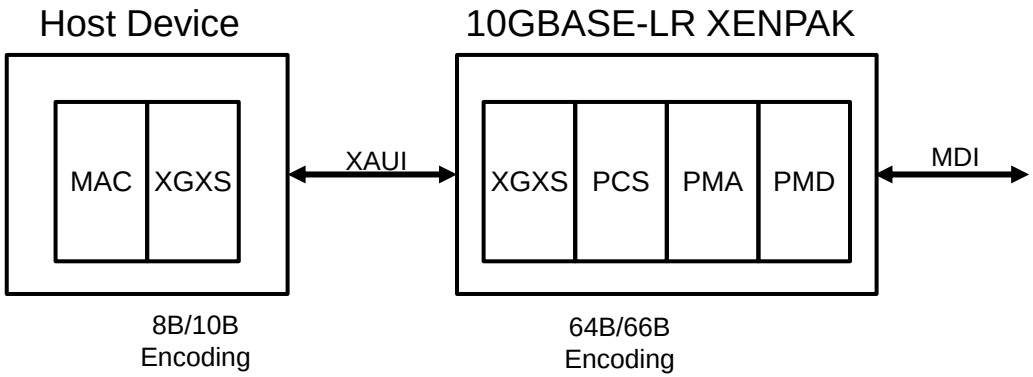


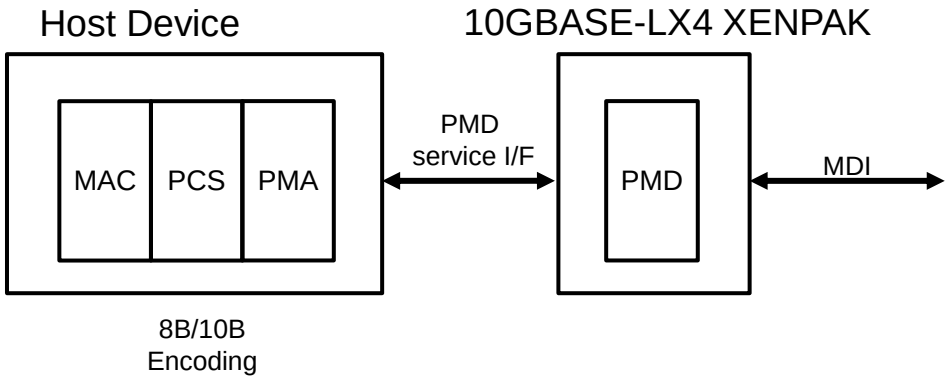
MAC = MEDIA ACCESS CONTROL
MDI = MEDIUM DEPENDENT INTERFACE
PCS = PHYSICAL CODING SUBLAYER
PHY = PHYSICAL LAYER DEVICE

PMA = PHYSICAL MEDIUM ATTACHMENT
PMD = PHYSICAL MEDIUM DEPENDENT
XAUI = 10 GIGABIT ATTACHMENT UNIT INTERFACE
XGMII = 10 GIGABIT MEDIA INDEPENDENT INTERFACE
XGXS = XGMII EXTENDER SUBLAYER

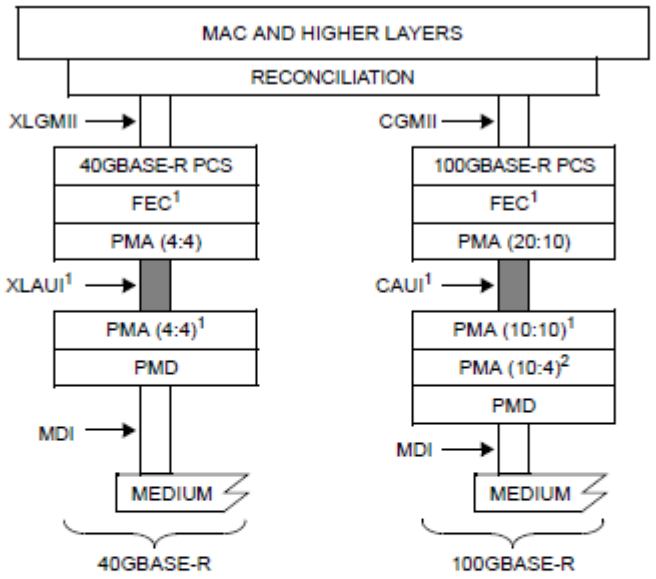
- Single PCS always next to the PMD/PMA
- Optional extender sublayer to extend the MII

10GbE Example implementations





100GbE Architecture



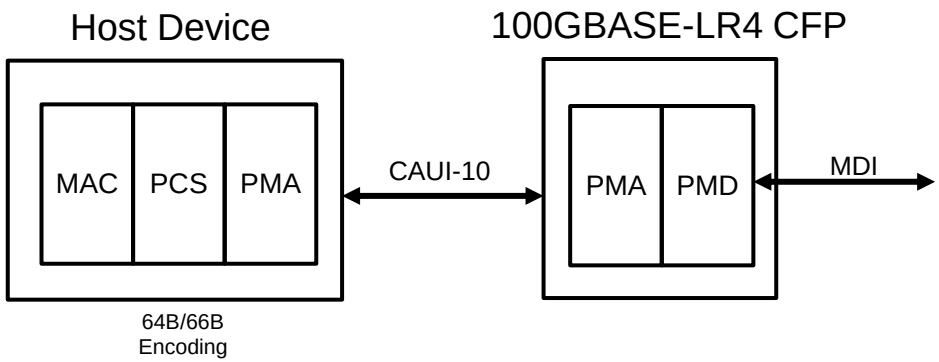
CAUI = 100 Gb/s ATTACHMENT UNIT INTERFACE
CGMII = 100 Gb/s MEDIA INDEPENDENT INTERFACE
FEC = FORWARD ERROR CORRECTION
MAC = MEDIA ACCESS CONTROL
MDI = MEDIUM DEPENDENT INTERFACE
PCS = PHYSICAL CODING SUBLAYER
PMA = PHYSICAL MEDIUM ATTACHMENT
PMD = PHYSICAL MEDIUM DEPENDENT
XLAUI = 40 Gb/s ATTACHMENT UNIT INTERFACE
XLGMII = 40 Gb/s MEDIA INDEPENDENT INTERFACE

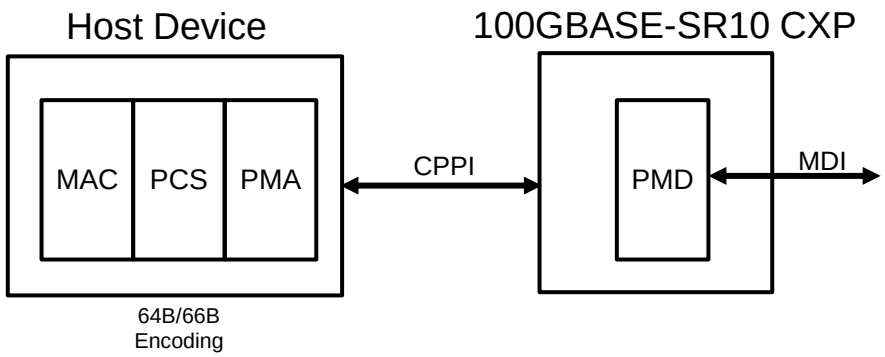
NOTE 1—OPTIONAL OR OMITTED DEPENDING ON PHY TYPE
NOTE 2—CONDITIONAL BASED ON PMD TYPE

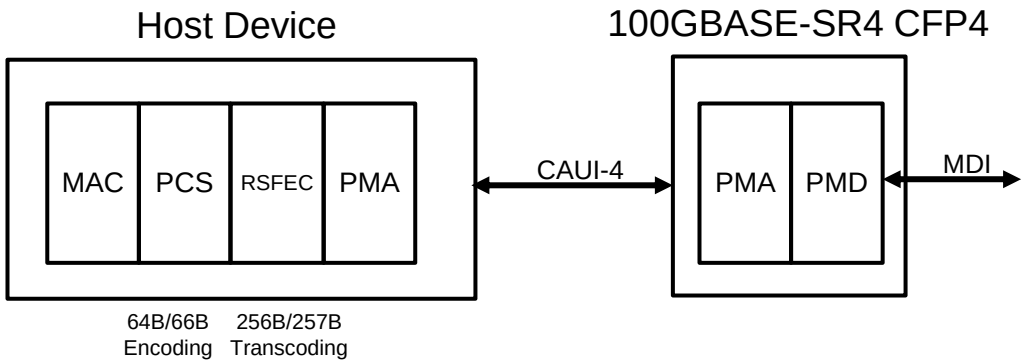
Figure 83A–1—Example relationship of XLAUI and CAUI to IEEE 802.3 CSMA/CD LAN model

- Single PCS always next to the MAC
- Vision was for bit muxing in the PMA to adapt to any PMD width
  - 802.3bj added RS-FEC which limits how that can be done
- No notion of a lower PCS or extender sublayer for unique PMD requirements (HGFEC or HOM)
- Multiple PMAs defined and each can be uniquely addressed
- Not everyone is happy with the RS-FEC sublayer, it has attributes of a PCS

100GbE Example implementations





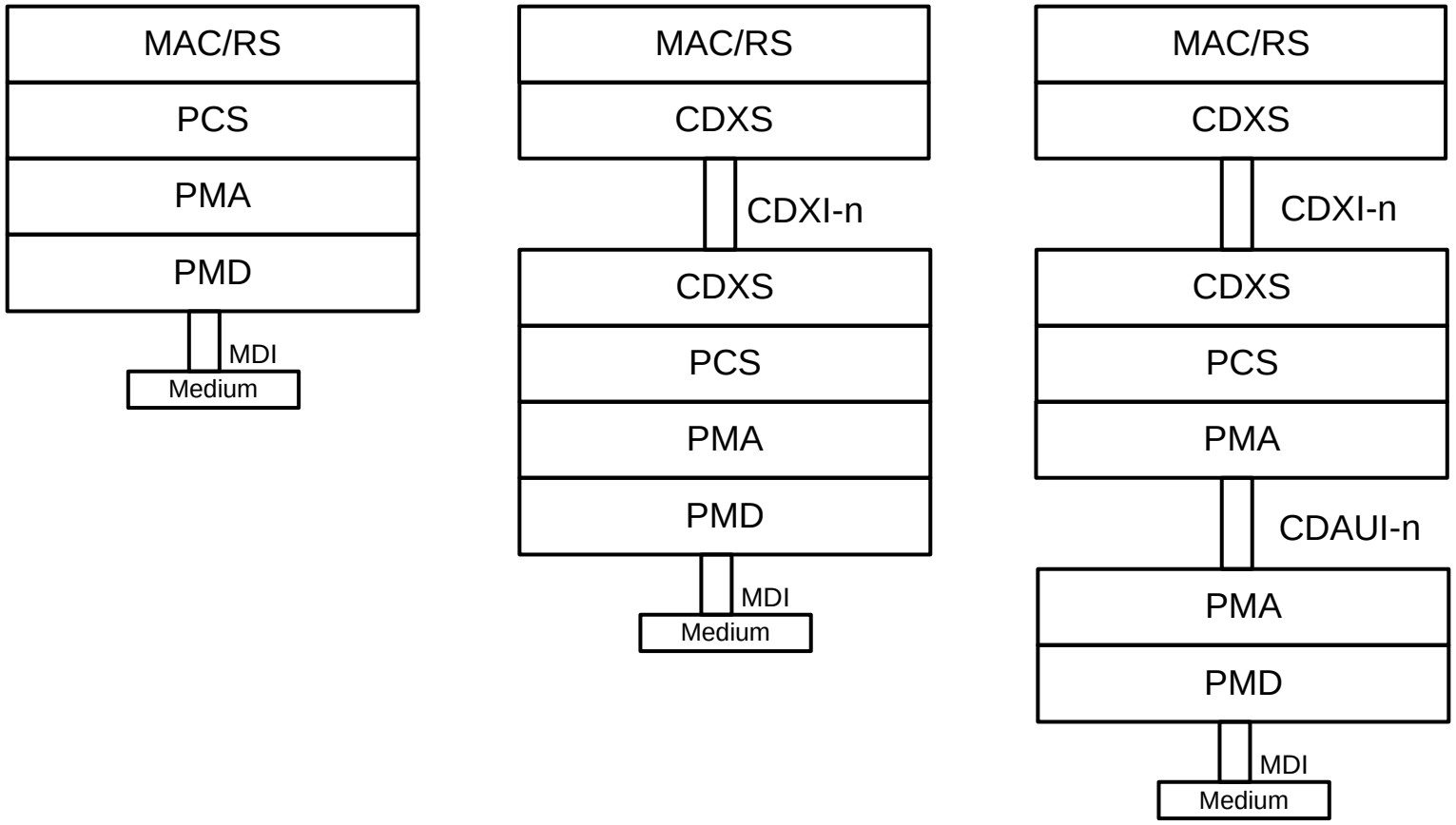


Names & definitions

- … the naming of things

Item	Name Used Temporarily	Function/definition
Extender sublayer	CDXS	Extends xMII (recovers raw 400G datastream) – used whenever a different coding or FEC is required further out in the PHY. Includes line code, FEC & timing required for extender interface.
Extender interface	CDXI-n	Interface between two CDXS, may be various widths
PMA interface	CDAUI-n	Physical instantiation of PMA service interface (similar to CAUI)

A Possible 400G Architecture



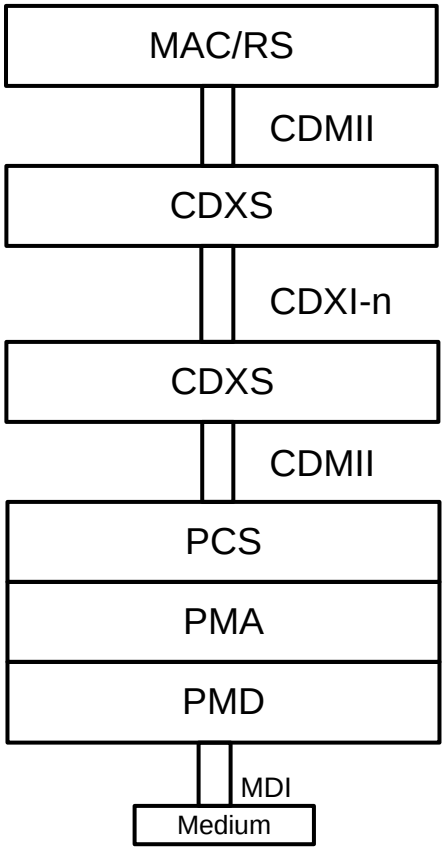
Sublayer Functions (at a high level)

Sublayer	10GbE	100GbE	400GbE (proposed)
MAC	Framing, addressing, error detection	Framing, addressing, error detection	Framing, addressing, error detection
PCS	Coding (8B/10B, 64B/66B), lane distribution, EEE	Coding (64B/66B), lane distribution, EEE	Coding, lane distribution, EEE, FEC
Extender	PCS + PMA	N/A	PCS + PMA + FEC
FEC	FEC, transcoding	FEC, transcoding, align and deskew	N/A?
PMA	Serialization, clock and data recovery	Muxing, clock and data recovery, HOM	Muxing, clock and data recovery, HOM??
PMD	Physical interface driver	Physical interface driver	Physical interface driver

Note that there are variations with a single speed, not all are captured in this table

A Possible 400G Architecture

- The interface between the CDXS and the MAC or PCS sublayer is always a CDMII



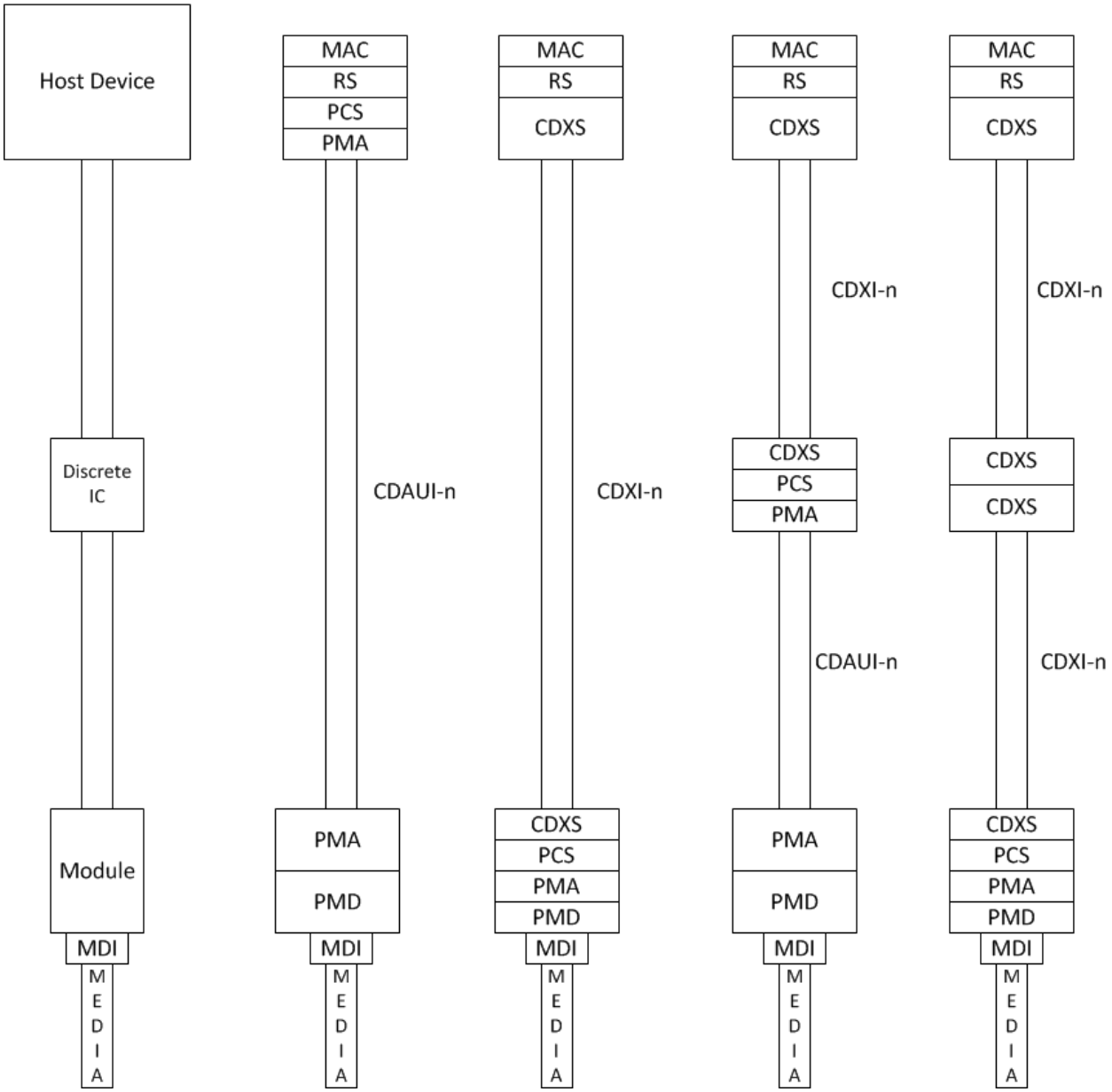
Line Encoding/FEC + PMA

Line Encoding/FEC + PMA

Line Encoding/FEC

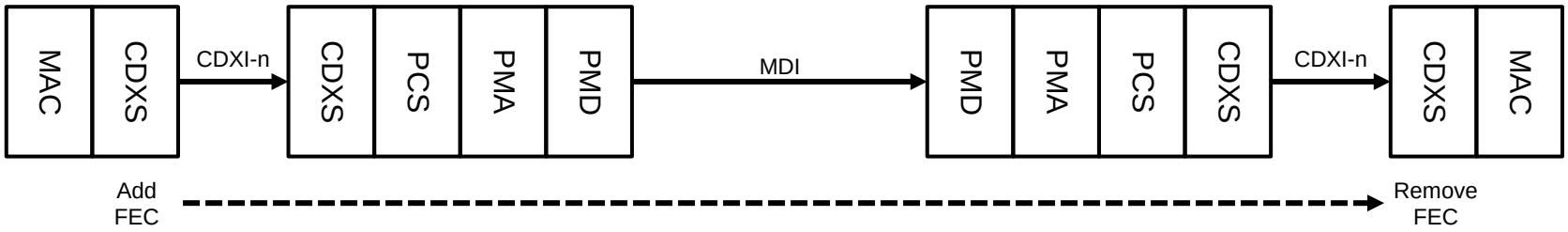
Muxing, CDR, HOM

400GbE Example Implementations



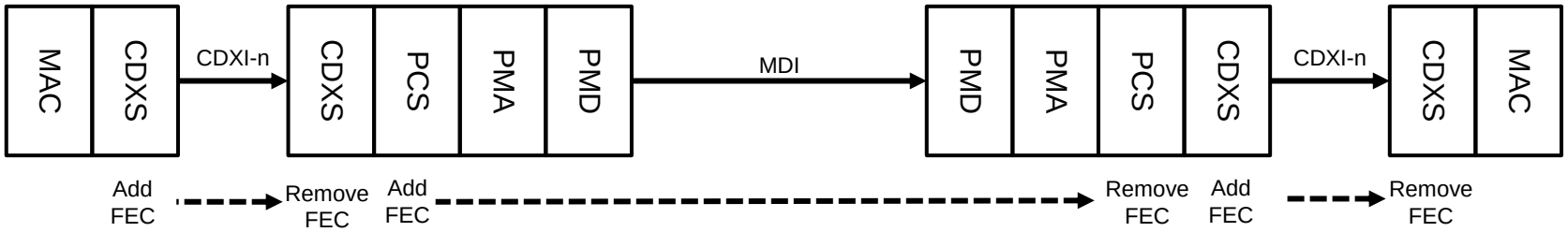
FEC Strategies: End to End

- End to End FEC pros and cons
  - \+ Simple, lowest overall complexity, latency and power
  - \- How to handle differentiation by application?
    - Short reach might require low latency, long reach can tolerate higher latency
  - \- How to handle the evolution of an electrical interface, legacy hosts etc.
    - Will mean in reality not having end to end FEC in some cases
  - \- How to allocate FEC error budget across multiple interfaces?
    - Works well if the BER contributed by the electrical interfaces is 0.1 x the BER from the PMD for instance



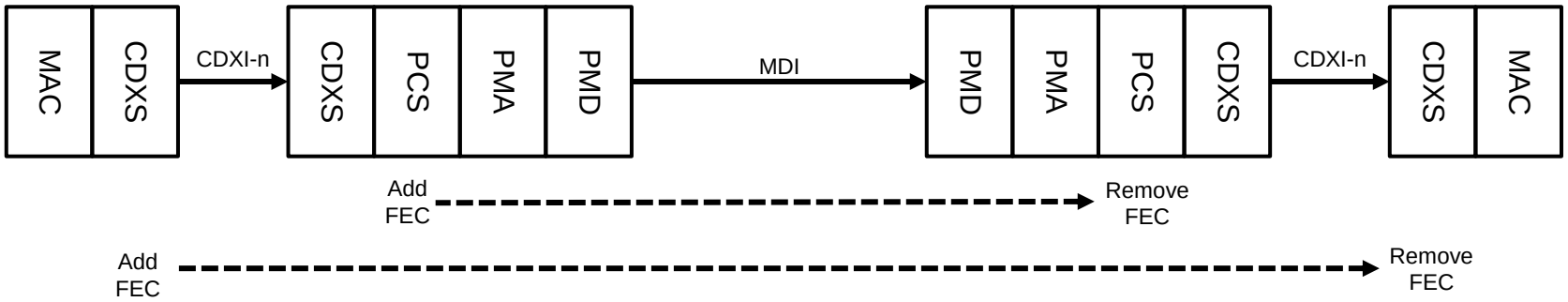
FEC Strategies: Segment by Segment

- Segment by Segment FEC pros and cons
  - \+ Most flexible, FEC is optimized for each application
  - \+ Easy to handle evolution of interfaces, legacy hosts etc.
  - \+ No issues with parsing BERs of multiple interfaces
  - \- Highest complexity, power, latency etc.



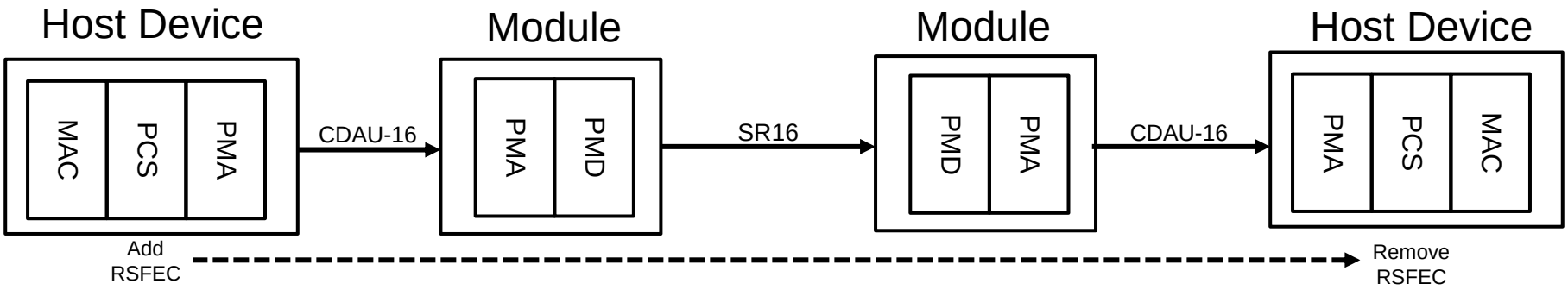
FEC Strategies: Encapsulated FECs

- Encapsulated FEC pros and cons
  - \+ Moderate complexity, latency and power
  - \+ Easier to handle evolution of interfaces, legacy hosts etc.
  - \- How to handle differentiation by application?
  - \- How to allocate FEC budget across multiple interfaces?
    - Up to 5 interfaces?
  - \- Bit rate might be higher than other options



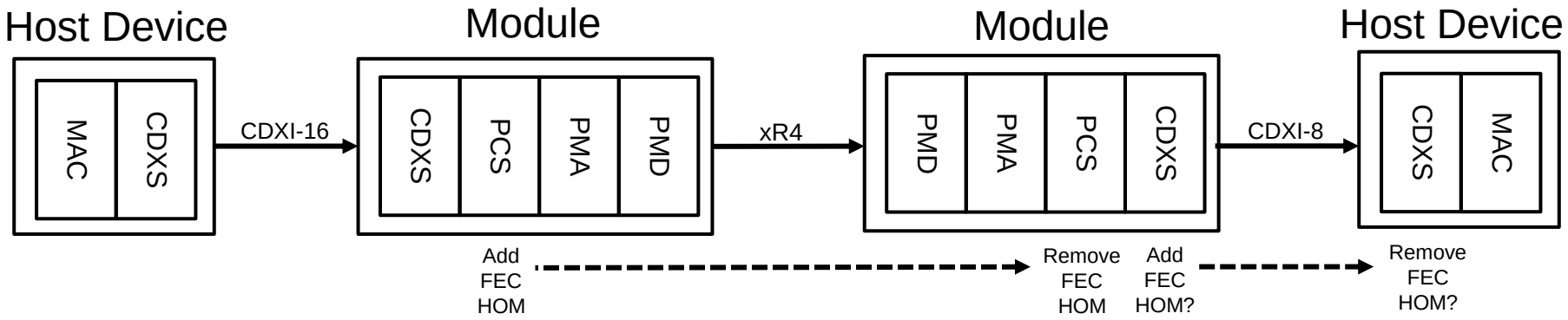
Possible SR16 Example

- One possibility is that RS-FEC and MLD is used for the PCS, so this enables simple implementations
- Here is what a real implementation might look like (note that CDAUI-16 might not require the RS-FEC):



Possible xR4 Example

- One possibility is that RS-FEC and MLD is used for the PCS, so this enables simple implementations
- Here is what a real implementation might look like:



- Assuming xR4 likely will require a high gain FEC
- Another variation is to use an encapsulated FEC

More Work to Do:

- FEC strategy
  - We need the FEC requirements of the PMDs
- How to handle end to end information?
  - Error Monitoring (BIP, FEC stats etc)
- How does EEE work in this architecture?

Thanks!