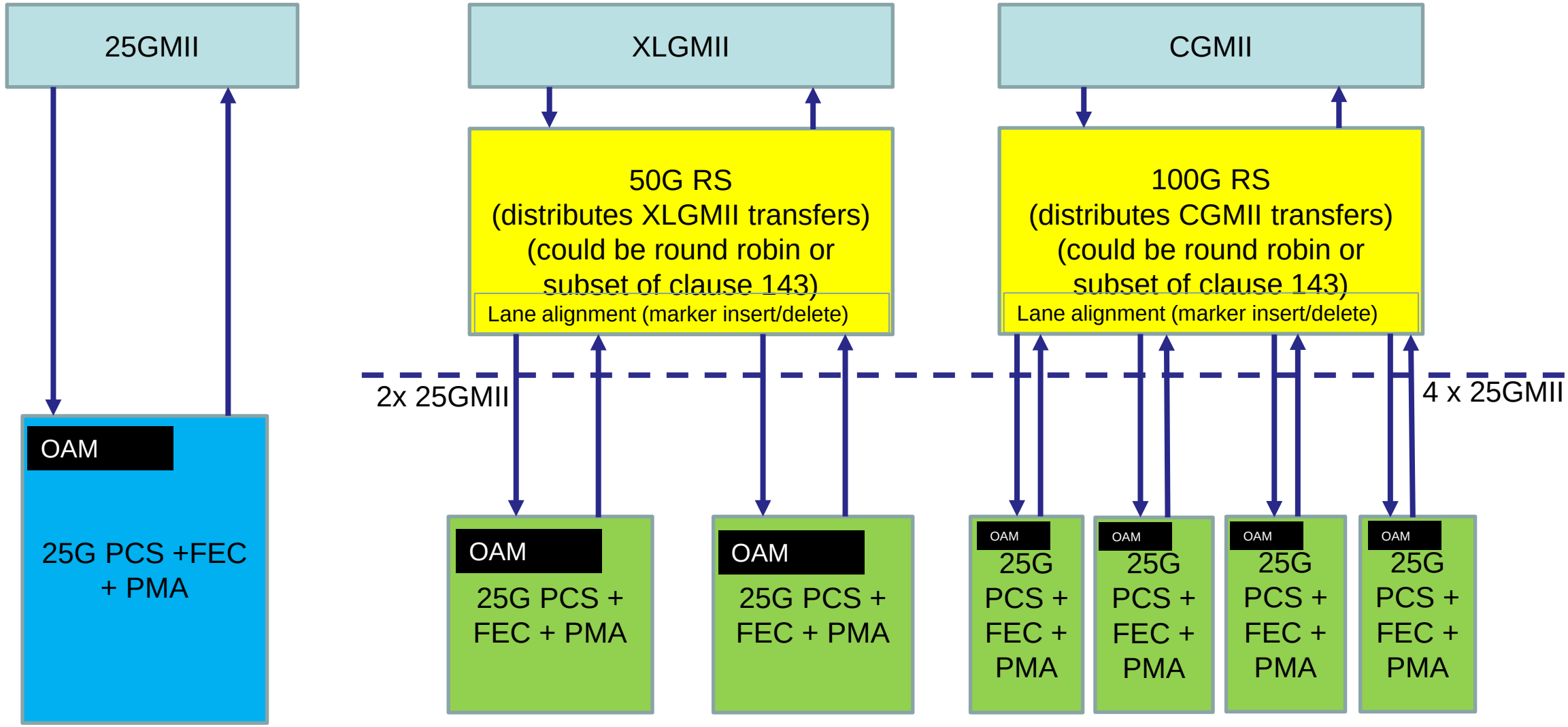
Update on Laning for 802.3cy

George Zimmerman

CME Consulting, Inc / Marvell

Lane PMA + FEC + PCS at the MII



Key Feature Decisions

- Architecture: PHYs joined at RS layer, PCS/PMA separate at 25GBASE-T1
  - ***NO CHANGES BELOW THE MII – KEY PREMISE***
- PHY-level Interface: 2x or 4x 25GMII
- Functions in the RS:
  - ☺ Lane swap correction – straightforward – identify lane at startup or with markers
  - ? Skew correction, lane alignment
    - Might be done at startup
    - All existing high-speed laning has opted for inserting alignment markers in data stream (more robust)
      - Inserting alignment markers steals bandwidth (use interpacket gaps)
    - IF WE ARE WRONG ABOUT FIXING ALIGNMENT AT STARTUP, THE PHY WILL NEED TO DROP LINK AND RETRAIN
  - X Coordinated control of PHY Resets
    - X **No standardized way to pass PHY resets up to the RS & coordinate multiple lanes**
    - ➢ ***No need to coordinate EEE – LPIs will be passed normally***

Alternative

- Define a completely new laning
  - High speed laned PHY alignments are generally in the PCS, not the MII
  - Define alignment markers in the PHY frames
    - Unnecessary overhead in every 25Gb/s Ethernet link
  - Define new interfaces for the 25G PHYs when laned
    - Possibly different than 25GMII

Clause 143 architecture

- Designed to work controlled by the MPMC
  - EPON MAC control sublayer provides MCRS_CTRL primitives not in Clause 4 MAC
- MCRS binds a MAC to multiple xMIIs
  - Each MAC is an “LLID” talking to multiple PCS/PMAs
  - Directions are independent
- MCRS converts serial MAC transmit data into parallel PHY streams
- MCRS maps xMII signals into individual MAC primitives
- Generates & expects continuous data & control characters

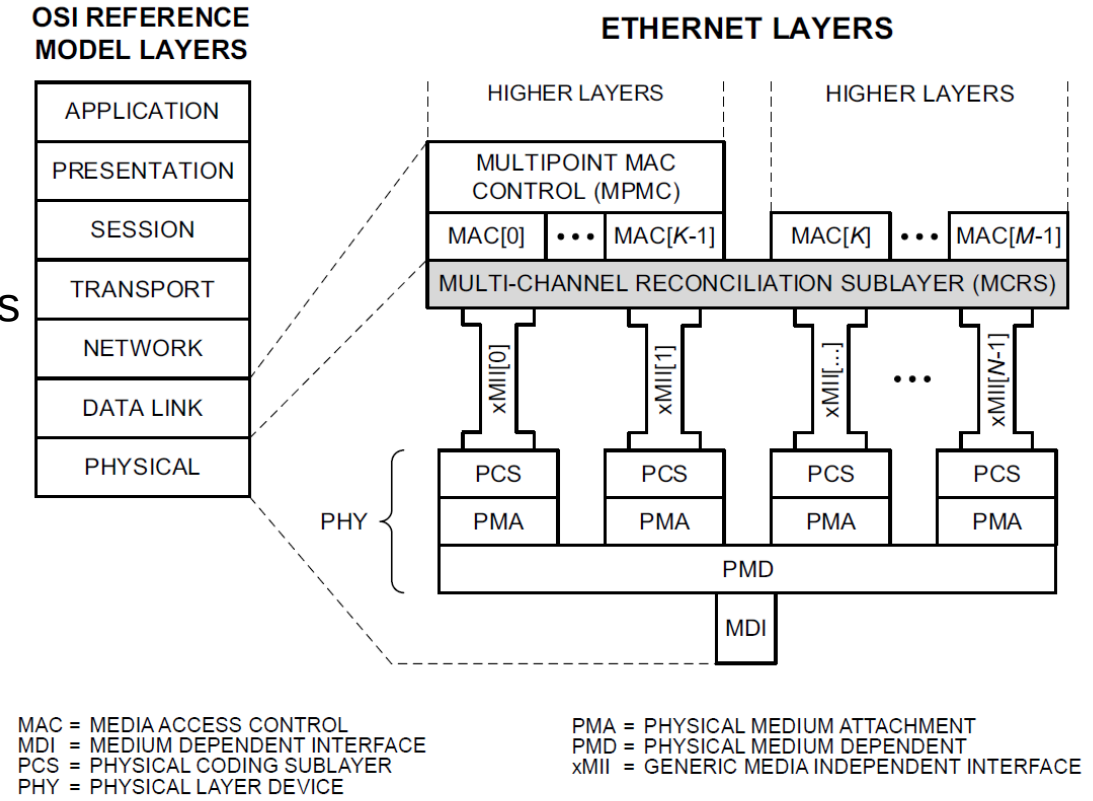


Figure 143–1—Relationship of MCRS to the OSI reference model

Source: IEEE P802.3dc D3.0, 143.1, 143.2

Clause 143 Envelope Quantum

- Clause 143 divides data into ‘Envelopes’
  - **Envelope** is a continuous transmission by a specific LLID (MAC iD) on a specific downstream / upstream channel
  - One EQ is 72 bits – 64 data and 8 control
  - Each EQ goes on an MII, (2 25GMII transfers to a PHY)

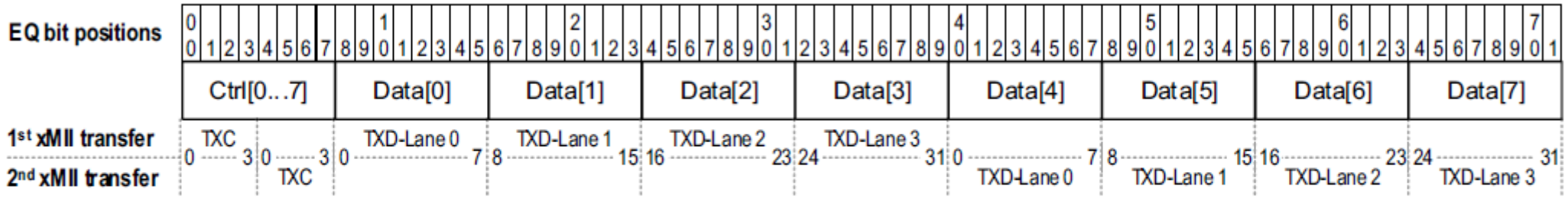


Figure 143–2—Envelope quantum (EQ) format

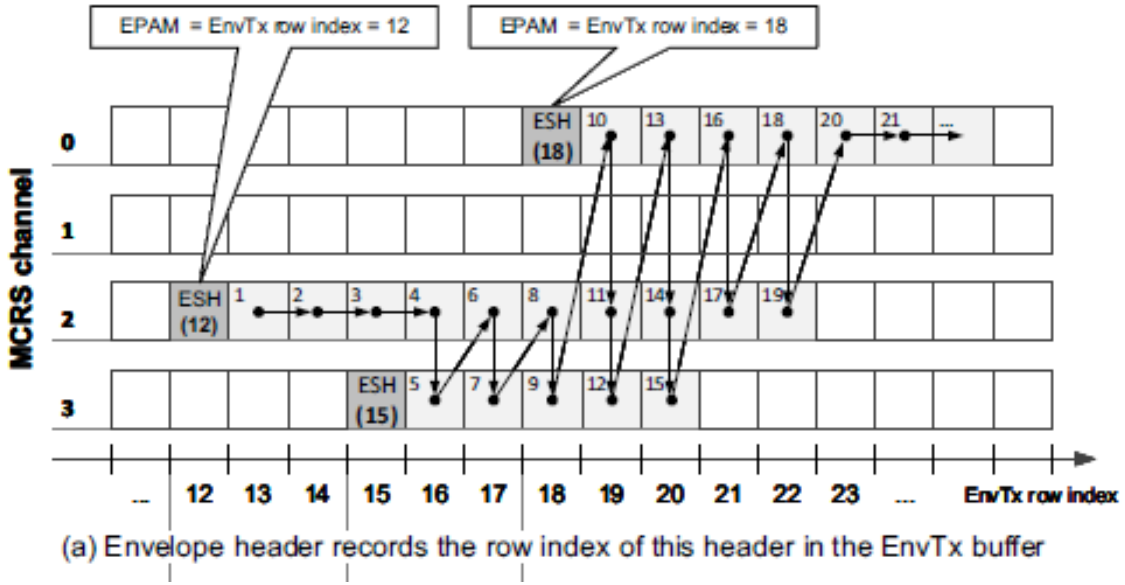
Source: IEEE P802.3dc D3.0

Envelopes
(see 143.2.4.2 Transmission Envelopes)

- An envelope is a continuous transmission by a specific MAC instance (LLID) on a specific PHY lane (MCRS channel)
  - EPON dynamically assigns (frequency) channels to multiple MACs
  - We would have one MAC and (1), 2 or 4 lanes – therefore (1), 2 or 4 envelopes
  - Each lane would have its own envelope
  - A frame may be (and in our case seems that it IS generally) striped over multiple lanes.
- Just need to pick the striping size
  - Determines buffering and (possible) latency in the RS
  - Suggest striping on either octets or on 8 octet groups (2 25GMII transfers, one 64B/65B PCS block)
    - Upside – minimizes buffering and additional delay
    - Downside – uncorrectable errors in one PHY/RS frame (130 transfers) even noninterleaved span 2080 (50G) or 4160 (100G) octets, which would corrupt multiple Ethernet frames, degrading frame error rate
- Questions –
  - Do we envision long links for 50G/100G laned connections? (do we really need 11m?)
  - Could we bypass RS-FEC entirely for shorter 50G/100G links?

Envelope Headers Mitigate Skew

- Envelope Start Header records order of envelope at the transmitter
  - "EPAM" field stores index
- Controls position stored in receive buffer
  - Envelopes read out in order
  - (buffer depth dependent on skew to tolerate)



Source: IEEE P802.3dc D3.0

Envelope Header - detail

- Contains:
  - Start control code (0xFB)
  - EnvType flag bit (ESH/ECH)
  - EPAM (position alignment)
  - CRC8 (header integrity)
  - Rest are not be needed by .3cy:
    - LLID (we only have 1 MAC)
    - Encryption bits (E and K) (not used)
    - Envelope Length (see next slide)
- 23 bits (3 octets) needed
  - Could use 3 octets for OAM or vendor-specific purposes

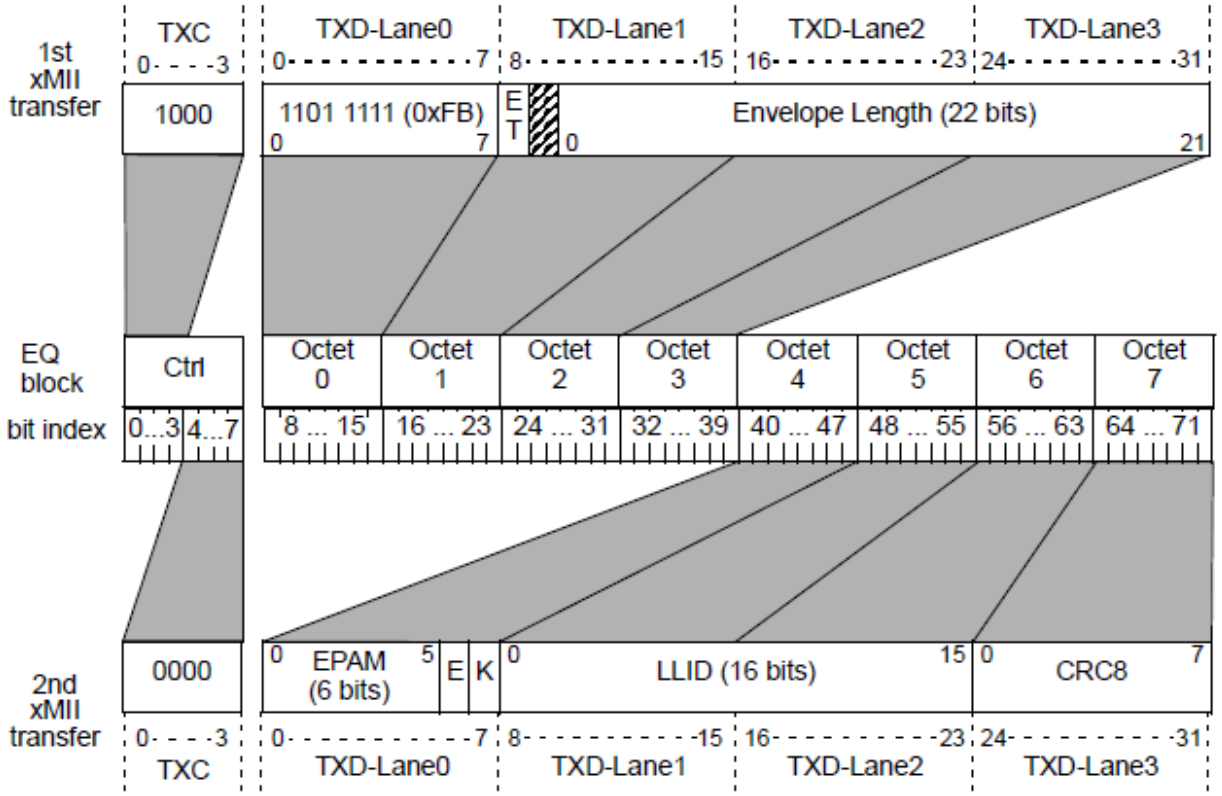


Figure 143–10—Mapping of envelope header fields into two xMII transfers

Source: IEEE P802.3dc D3.0

Envelope Headers
See 143.2.4.3 Envelope Headers

- Each Envelope starts with an Envelope Start Header
  - Sent when no data is being transmitted
    - This is hard to do with a continuous system and requires pausing the link availability for data (EPON has this feature)
    - BUT – we have a fixed number of envelopes – physically configured by a number of cables attached
    - AND – there is no ‘Envelope terminator’ so no need to limit envelope size
- Suggest:
  - send ESH (or equivalent) at startup only
  - with unlimited envelope length (since our system is static and envelopes are fixed)
  - Do initial lane alignment at startup with ESH
- The Envelope gets Envelope Continuation Headers sent in place of data frame preamble
  - Replacing all 8 octets of the preamble would be incompatible with Clause 99 MAC Merge

ECH Compatibility with Clause 99

Clause 143 ECH replaces first 8 octets – ECH cannot simply replace 8 octets

Normally preamble & SFD

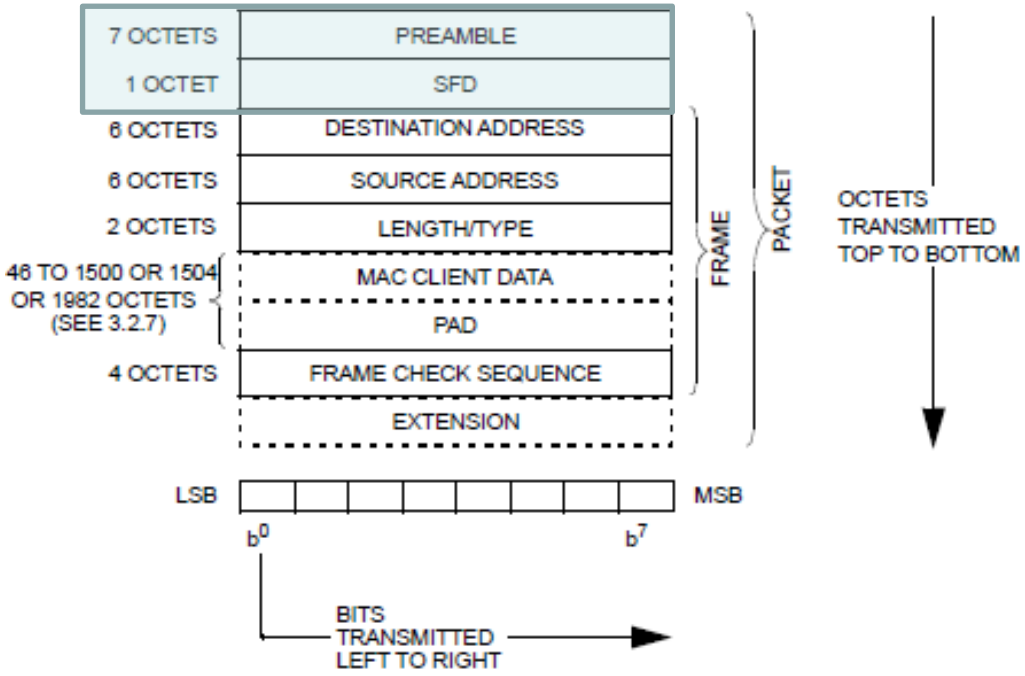


Figure 3–1—Packet format

Clause 99 uses 7th & 8th octets to convey information

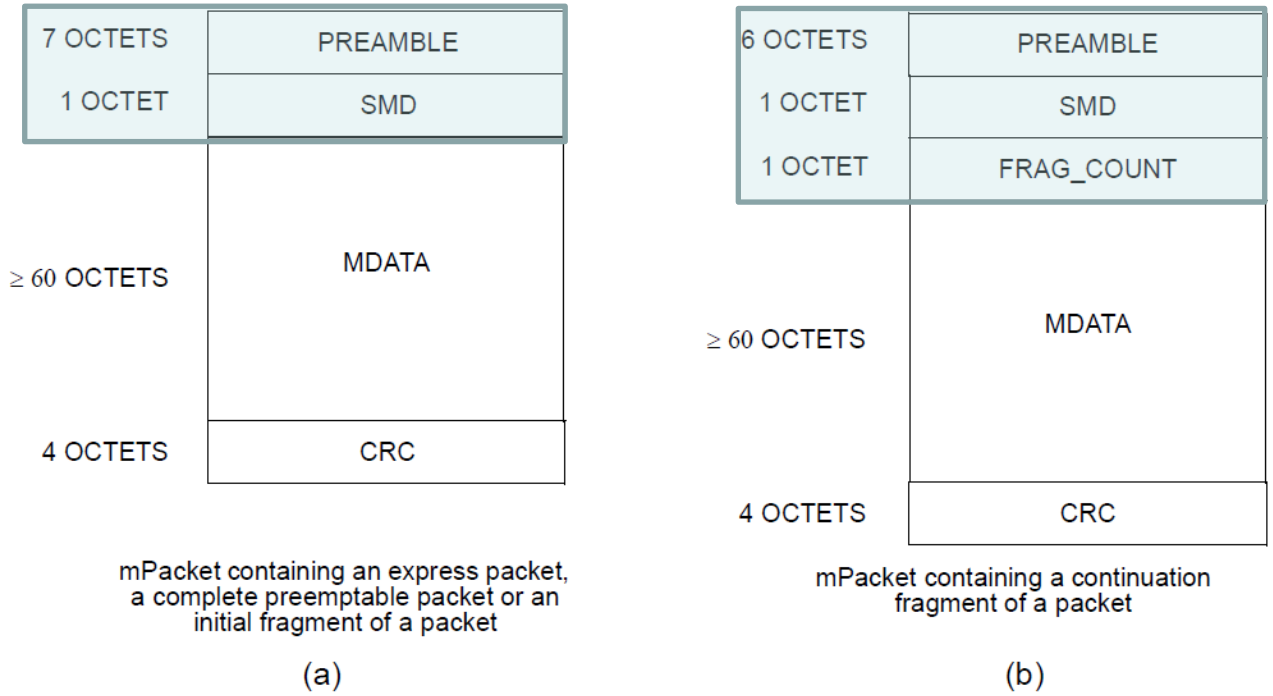


Figure 99–4—mPacket format

Source: IEEE Std 802.3-2018

New RS must encode EPAM (and any other information) in the first 6 octets and leave octets 7 & 8 intact

Key parameters simplify

- EPAM records the relative position of an envelope
  - 6 bit field which controls the receive buffer positioning
  - EPON lanes could have a lot more skew than .3cy
    - But frequency lanes can't cross… so it is 'row' (skew) only
  - 802.3cy can divide the EPAM as row and column (lane)
    - Determine number of bits as needed
- ADJ_BLOCK_SIZE, RATE_ADJ_SIZE = 0
- Need to redraw transmit state diagram process

New Envelope Headers

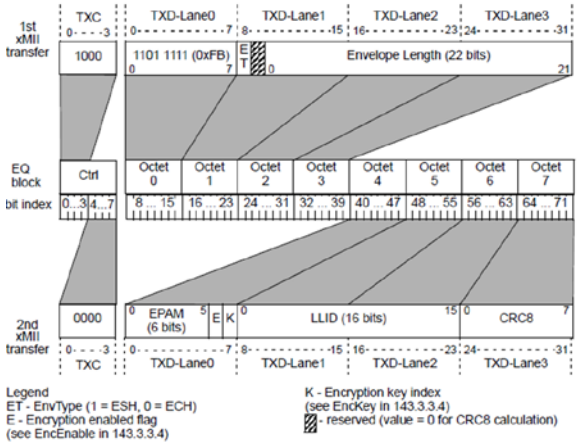


Figure 143–10—Mapping of envelope header fields into two xMII transfers

- Use first transfer only
  - Leave 2nd transfer (last 4 octets of preamble on ECH) intact
- Collapse EPAM field to 6 bits, specify row and column fields
- Reserve Octet 2 for OAM or vendor information (TBD)
- Consider whether CRC is needed (but for now retain it)

1st 25GMII Transfer

TXC	TXD Lane 0/Octet 0	TXD Lane 1/Octet 1				TXD Lane 2/Octet 2	TXD Lane 3/Octet 3
0 - 3	0 - - - - 7	8	9 - - - - 10	11 - - - 14	15	16 - - - - - - - - - - 23	24 - - - - - - - - - - 31
1000	1101 1111 (0xFB)	Env Type	EPAM col	EPAM row	Reserved	<OAM or Vendor>	CRC

2nd 25GMII Transfer <for ECH only, TBD for ESH>

TXC	TXD Lane 0/Octet 4	TXD Lane 1/Octet 5		TXD Lane 2/Octet 6	TXD Lane 3/Octet 7
0 - 3	0 - - - - 7	8 -	15	16 - - - - - - - - - - 23	24 - - - - - - - - - - 31
0000	Preamble Octet 4	Preamble Octet 5		Preamble Octet 6	Preamble Octet 7

Note 1 : TXC are Ctrl bits 0..3 and 4..7 of the EQ on first and 2nd transfers respectively

Note 2: Env Type is (1 = ESH, 0 = ECH)

Note 3: 2nd 25GMII is free to be used as needed for startup purposes for ESH

Modified Input Process diagram

- Startup process:
  - INIT is called at startup, sets EnvLeft to zero
  - EnvLeft is set to a nonzero length when start header is sent
  - EnvLeft is never decremented
- NEXT_ROW starts new stripe across lanes
- CHECK_NEW_ENV checks EnvLeft (startup) for each column (PHY lane),
  - if in startup, send ESH (start new env., once per col.)
  - Otherwise, GetMacBlock loads data from MAC
    - If the MAC data is a preamble, assemble and send ECH
    - Otherwise, send the data
- NEXT_COL goes to the next lane, or next row, depending on whether row is done

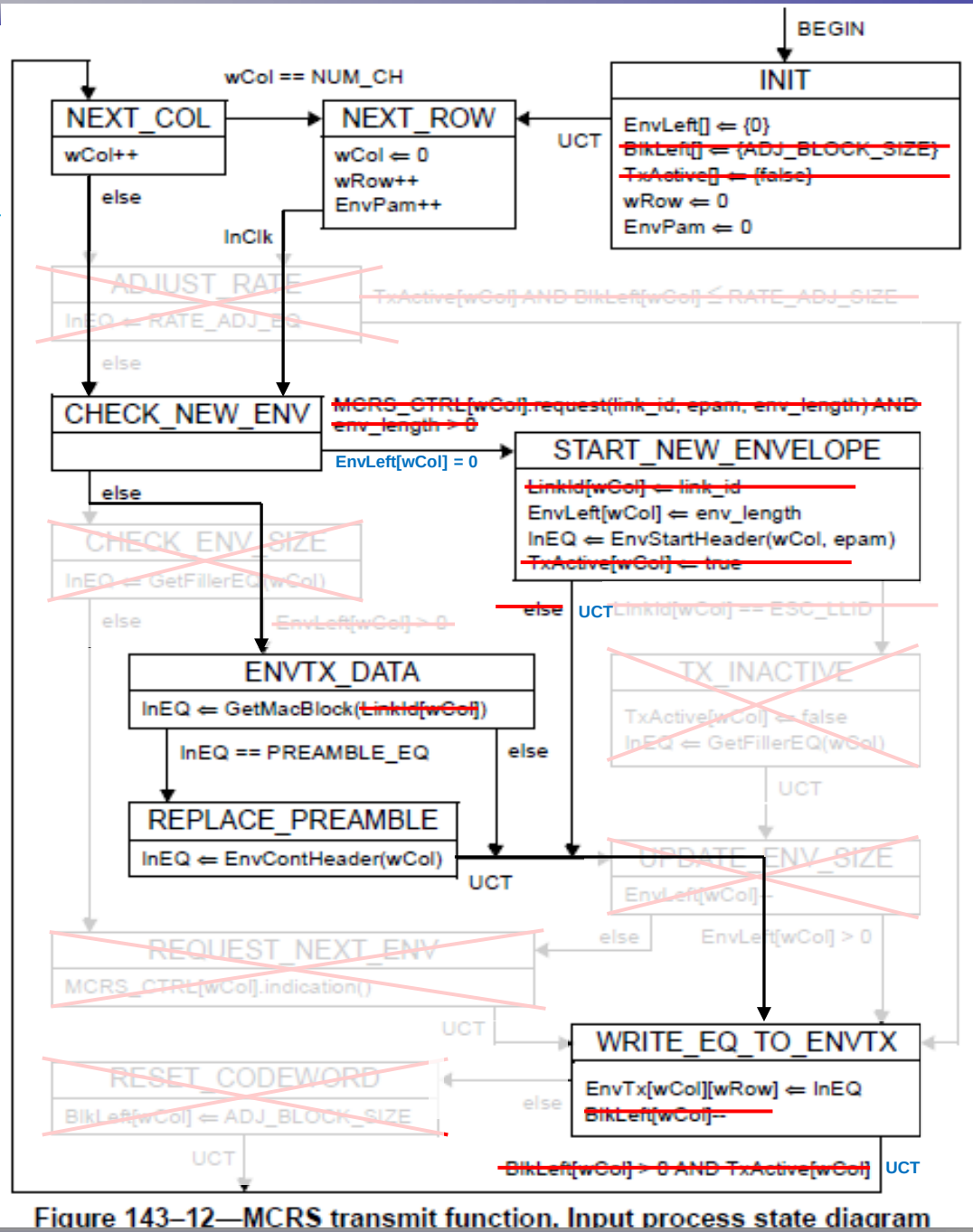


Figure 143–12—MCRS transmit function, Input process state diagram

PHY resets:

Basis to go forward (less mature, for discussion)

- Propagating a single reset to all 4 PHYs is unsolved
- Each PHY may be reset individually.
  - It is desirable that any lane being in reset stops the MAC from sending data
  - PCS_reset causes the PHY to present local faults at the 25GMII (see PCS receive state diagram) – and the RS sees these from each PHY lane
    - Desirable to OR the fault status of the 4 lanes
    - Do we want to report via management the lane that has a fault?
  - From Clause 46: “When this Local Fault status reaches an RS, the RS stops sending MAC data or LPI, and continuously generates a Remote Fault status on the transmit data path (possibly truncating a MAC frame being transmitted).”
- To discuss: (remember, RS’s don’t have MDIO, so this is clause 30 or other)
  - Do we want the reset to cause a new startup of the RS (resetting alignment), or do we want the RS to be separately reset?
  - Do we want to report which lane has a fault?

Still work to do…

How to pass (and synchronize) PHY reset information

- And any autonegotiation information…

Review/modify: (looks straightforward)

- GetMacBlock process code to use single MAC, remove any EPON-specific stuff
- ESH and ECH process code to reflect new ESH & ECH proposed here

Error checking – did I miss something?

- Compatibility with 25GMII implementations & with 50/100G MAC constraints

Major concern: All of this takes time and needs multiple skilled eyes

- May be better as a new/separate project with MAC/MII experts involved

Key premise of “no changes below MII” makes this work separable (and a separable component) from 25GBASE-T1

- Plan to present to NEA Ad hoc to gauge interest

Discussion?

THANK YOU!