# Ethernet Metadata Service interface (EMSI) baseline proposal

David Law, HPE
dlaw@hpe.com

# Supporters

Gary Nicholl, Cisco

Eugene Opsasnick, Broadcom

Kapil Shrikhande, Upscale AI

Viet Tran, Keysight

Ulf Parkholm, Ericsson

# Agenda

- General Approach
- Baseline proposal
  - Introduction, overview and Ethernet Metadata Service interface (EMSI)
  - Generic Reconciliation Sublayer (gRS)
- Open items

# General Approach

# General Approach

- Add a new Ethernet Metadata Services (EMS) Clause
  - Specifies two Ethernet metadata capability
    - A per-packet metadata capability using the preamble
    - A packet-independent metadata capability using Control Ordered Set metadata
  - Specifies constraints regarding the operation of these two capabilities
    - Data rates and PHY types that support each capability
    - Conditions that need to be met before a capability can be enabled
    - Service constraints (e.g., packet-independent metadata rate limit)
  - Specifies any constrain, and provide any consideration, regarding other capabilities
    - Interaction with Time Synchronisation, Energy-Efficient Ethernet, others?
  - Specifies a new Ethernet Metadata Service interface (EMSI)
    - Also know as a physical layer service access point (PSAP)
      - See IEEE Std 802-2024 Overview and Architecture
  - Specifies a set of EMS optional additions to the RS

# Introduction, overview and Ethernet Metadata Service interface (EMSI)

# Baseline text
# EMS : Introduction and overview

## XXX Ethernet Metadata Services

### XXX.1 Introduction

Metadata is information about the data contained within a packet, or about the Ethernet link and the data it is communicating. This clause specifies Ethernet metadata services that enable the transfer of metadata over a subset of full-duplex point-to-point Ethernet links through two optional metadata capabilities. One optional capability supports the transfer of per-packet metadata, while the other supports the transfer of packet-independent metadata.

### XXX.2 Overview

Ethernet metadata services are provided to the Ethernet metadata services client (EMS client) through the Ethernet Metadata Service Interface (EMSI) specified in XXX.3 by functions within the generic Reconciliation Sublayer (gRS) specified in XXX.4. Within the scope of this Clause, the term gRS is used to denote any Reconciliation Sublayer supporting the EMSI.

The optional per-packet metadata capability enables metadata to be transferred in the packet preamble [data rates and/or PHYs TBD]. The optional packet-independent metadata capability enables metadata to be transferred in physical-layer Control Ordered Set by a subset of PHYs (see XX.Z) operating at data rates of 50 Gb/s or greater, independent of packets.

# Baseline text
# EMS : Overview (continued) and EMSI

An optional metadata capability is enabled only if it is supported by the data rate and PHY type, and only after it has been established that the link partner supports that metadata capability and the associated syntax and semantics of the metadata to be transmitted. The mechanism by which support for a metadata capability, and the corresponding syntax and semantics, is established is beyond the scope of this Clause. All metadata capabilities are disabled on link failure; the definition of link failure is implementation-dependent.

## XXX.3 Ethernet Metadata Service interface (EMSI)

## XXX.3.1 Introduction

This subclause specifies services provided by an extension to the Reconciliation Sublayers specified elsewhere in this standard.

## XXX.3.1.1 Interlayer service interfaces

Figure XXX–1 depicts the relationship between the EMS Client, the EMSI, and the gRS.

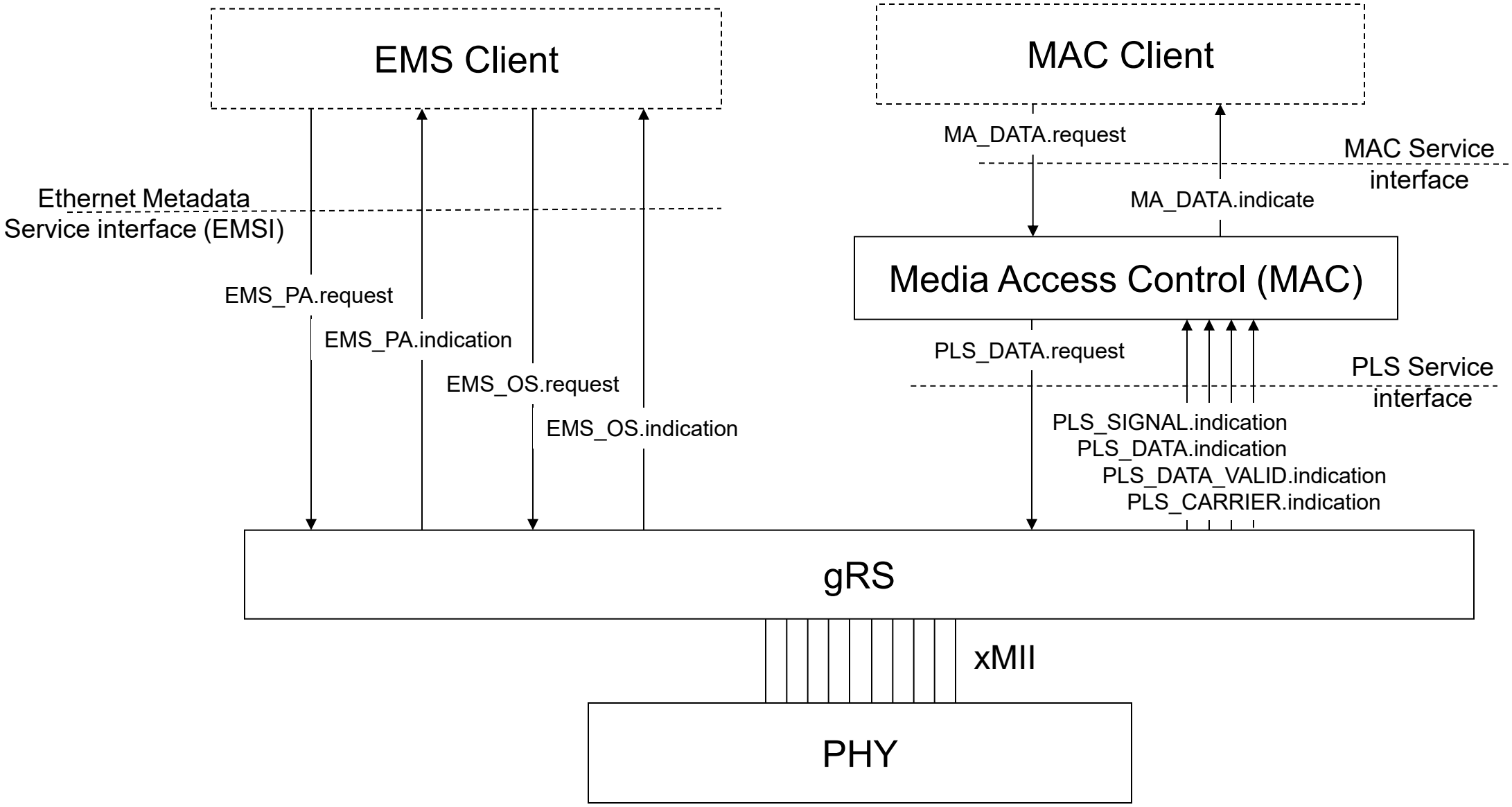



Figure XXX–1 Relationship of the EMS Client, EMSI, and gRS

# Baseline text
# Responsibilities of EMS client, EMSI

## XXX.3.1.2 Responsibilities of EMS client

When an EMS client decides to transmit metadata, it communicates it to the gRS using the EMSI. When the gRS receives metadata, it communicates it to the EMS client using the EMSI.

The conditions under which an EMS client decides to transmit metadata, the metadata syntax and semantics, and what action is taken by the EMS client when it receives metadata, are beyond the scope of this Clause.

## XXX.3.2 EMSI

The following specifies the service interface provided by the gRS to the EMS client. These services are described in an abstract manner and do not imply any particular implementation. The model used in this service interface specification is identical to that used in 1.2.2.

The following primitives are defined:

- Per-packet metadata primitives
  - EMS_PA.request
  - EMS_PA.indication
- Packet-independent metadata primitives
  - EMS_OS.request
  - EMS_OS.indication

# Baseline text
# Detailed service specification: EMS_PA.request primitive

## XXX.3.3 Detailed service specification

### XXX.3.3.1 Per-packet metadata primitives

The EMS_PA.request and EMS_PA.indication primitives are only provided by the gRS if it supports the optional per-packet metadata capability enabling metadata to be transferred in the packet preamble.

#### XXX.3.3.1.1 EMS_PA.request primitive

This primitive defines the transfer of metadata from the EMS client to the gRS for transmission in the next packet preamble.

##### XXX.3.3.1.1.1 Semantics

The semantics of the primitive are as follows:

EMS_PA.request(first_replace_octet, last_replace_octet, new_preamble_data)

The first_replace_octet parameter is an integer in the range 1 to 6 that defines the first preamble octet to be replaced by the data supplied in the new_preamble_data parameter. The last_replace_octet parameter is an integer in the range 1 to 6 that defines the last preamble octet to be replaced by the data supplied in the new_preamble_data parameter. Preamble octets are numbered using integer values 0 to 6, where preamble octet 0 is the first preamble octet transmitted by the MAC and preamble octet 6 is the last preamble octet transmitted by the MAC.

The new_preamble_data parameter is ((last_replace_octet - first_replace_octet ) + 1 octets long that defines the data used to replace the preamble field octets from first_replace_octet to last_replace_octet.

# Baseline text
# Detailed service specification: EMS_PA.request primitive

## XXX.3.3.1.1.2 When generated

The EMS client generates this primitive to transmit metadata in the preamble field of the next packet transmission.

## XXX.3.3.1.1.3 Effect of receipt

Upon receipt of this primitive, the gRS replaces the preamble field of the next packet transmission, from the octet defined by first_replace_octet parameter, to the octet defined by last_replace_octet parameter, with the data provided in new_preamble_data parameter (see XXX.4.1.1).

# Baseline text
# Detailed service specification: EMS_PA.indication primitive

## XXX.3.3.1.2 EMS_PA.indication primitive

This primitive defines the transfer of the preamble field from the gRS to the EMS client after reception of a packet.

## XXX.3.3.1.2.1 Semantics

The semantics of the primitive are as follows:

EMS_PA.indication (preamble_data)

The preamble_data parameter contains the first seven octets of a received packet.

## XXX.3.3.1.2.2 When generated

The gRS generates this primitive when it detects the eighth octet of a packet on the xMII receive signals (see XXX.4.2.1).

## XXX.3.3.1.2.3 Effect of receipt

Upon receipt of this primitive, the EMS client extracts metadata from the preamble_data parameter using knowledge of the syntax of the transmitted metadata. What actions it takes, if any, based on the extracted metadata is beyond the scope of this Clause.

# Baseline text

# Detailed service specification: EMS_OS.request primitive

## XXX.3.3.2 Packet-independent metadata primitives

The EMS_OS.request and EMS_OS.indication primitives are only provided by the gRS if it supports the optional packet-independent metadata capability enabling metadata to be transferred in Control Ordered Set.

## XXX.3.3.2.1 EMS_OS.request primitive

This primitive defines the transfer of metadata from the EMS client to the gRS for transmission in a Control Ordered Set.

## XXX. 3.3.2.1.1 Semantics

The semantics of the primitive are as follows:

EMS_OS.request(ordered_set_data)

The ordered_set_data parameter contains the seven data octets to be transmitted in the Control Ordered Set.

## XXX. 3.3.2.1.2 When generated

The EMS client generates this primitive to transmit metadata in a Control Ordered Set.

## XXX. 3.3.2.1.3 Effect of receipt

Upon receipt of this primitive, gRS inserts a Control Ordered Set using the seven data octets provided in the ordered_set_data parameter (see XXX.4.1.2) on the xMII transmit path.

# Baseline text
# Detailed service specification: EMS_OS.indication primitive

### XXX.3.3.2.2 EMS_OS.indication primitive

This primitive defines the transfer of the seven data octets of a Control Ordered Set from the gRS to the EMS client after reception of a Control Ordered Set.

### XXX.3.3.2.2.1 Semantics

The semantics of the primitive are as follows:

EMS_OS.indication(ordered_set_data)

The ordered_set_data parameter contains the seven octets received in the Control Ordered Set.

### XXX.3.3.2.2.2 When generated

The gRS generates this primitive when it detects a Control Ordered Set on the xMII receive signals (see XXX.4.2).

### XXX.3.3.2.2.3 Effect of receipt

Upon receipt of this primitive, the EMS Client extracts any metadata from the ordered_set_data parameter using knowledge of the syntax of the transmitted metadata. What actions it takes, if any, based on the extracted metadata is beyond the scope of this Clause.

# Generic Reconciliation Sublayer (gRS)

# Existing text
# Clause 35 : 1 Gb/s (GMII)

## 35.2.1.1.3 When generated

The GTX_CLK signal is generated by the Reconciliation sublayer. The TXD<7:0>, TX_EN and TX_ER signals are generated by the Reconciliation sublayer after every group of eight PLS_DATA.request transactions from the MAC sublayer to request the transmission of eight data bits on the physical medium, to extend the carrier event the equivalent of eight bits, or to stop transmission.

## 35.4 LPI Assertion and Detection

The definition of TX_EN, TX_ER and TXD<7:0> is derived from the state of PLS_DATA.request (35.2.1.1), except when it is overridden by an assertion of LP_IDLE.request.

# Existing text
# Clause 46 : 2.5 Gb/s, 5 Gb/s, and 10 Gb/s (XGMII)

## 46.1.6 XGMII structure

On transmit, each eight PLS_DATA.request transactions represent an octet transmitted by the MAC. The first octet is aligned to lane 0, the second to lane 1, the third to lane 2 the fourth to lane 3, then repeating with the fifth to lane 0, etc.

## 46.1.7.1.4 Effect of receipt

The OUTPUT_UNIT values are conveyed to the PHY by the signals TXD<31:0> and TXC<3:0> on each TX_CLK edge. Each PLS_DATA.request transaction shall be mapped to a TXD signal in sequence (TXD<0>, TXD<1>,... TXD<31>, TXD<0>) as described in 46.2. After 32 PLS_DATA.request transactions from the MAC sublayer (four octets of eight PLS_DATA.request transactions each), the RS requests transmission of 32 data bits by the PHY. The first octet of preamble shall be converted to a Start control character and aligned to lane 0. The TXD<31:0> and TXC<3:0> shall be generated by the RS for each 32 bit-times of the MAC sublayer.

# Existing text
# Clause 81: XLGMII/CGMII

## 81.1.6 XLGMII/CGMII structure

The 64 TXD and 8 TXC signals shall be organized into eight data lanes, as shall the 64 RXD and 8 RXC signals (see Table 81–2). The eight lanes in each direction share a common clock, TX_CLK for transmit and RX_CLK for receive. The eight lanes are used in round-robin sequence to carry an octet stream. On transmit, each eight PLS_DATA.request transactions represent an octet transmitted by the MAC. The first octet is aligned to lane 0, the second to lane 1, the third to lane 2, the fourth to lane 3, the fifth to lane 4, the sixth to lane 5, the seventh to lane 6 and the eighth to lane 7, then repeating with the ninth to lane 0, etc. Delimiters and interframe idle characters are encoded on the TXD and RXD signals with the control code indicated by assertion of TXC and RXC, respectively.

# Baseline text
# generic Reconciliation Sublayer (gRS)

## XXX.4 Generic Reconciliation Sublayer (gRS)

Two gRS functions are defined to support the EMSI. The Ethernet metadata transmit function (see XXX.4.1), which is responsible for servicing the EMS_PA.request and the EMS_OS.request primitives, and the Ethernet metadata receive function (see XXX.4.2), which is responsible for generating the EMS_PA.indication and the EMS_OS.indication primitives. The Ethernet metadata transmit and receive functions are contained in the RS as shown in Figure XXX–2 and supplement and amend the mapping of the xMII signals to/from the PLS service interface defined for the respective RS Clause.

There are two optional metadata capabilities. One optional capability supports the transfer of per-packet metadata, while the other supports the transfer of packet-independent metadata. All metadata capabilities shall be disabled on link failure. The definition of link failure is implementation-dependent.

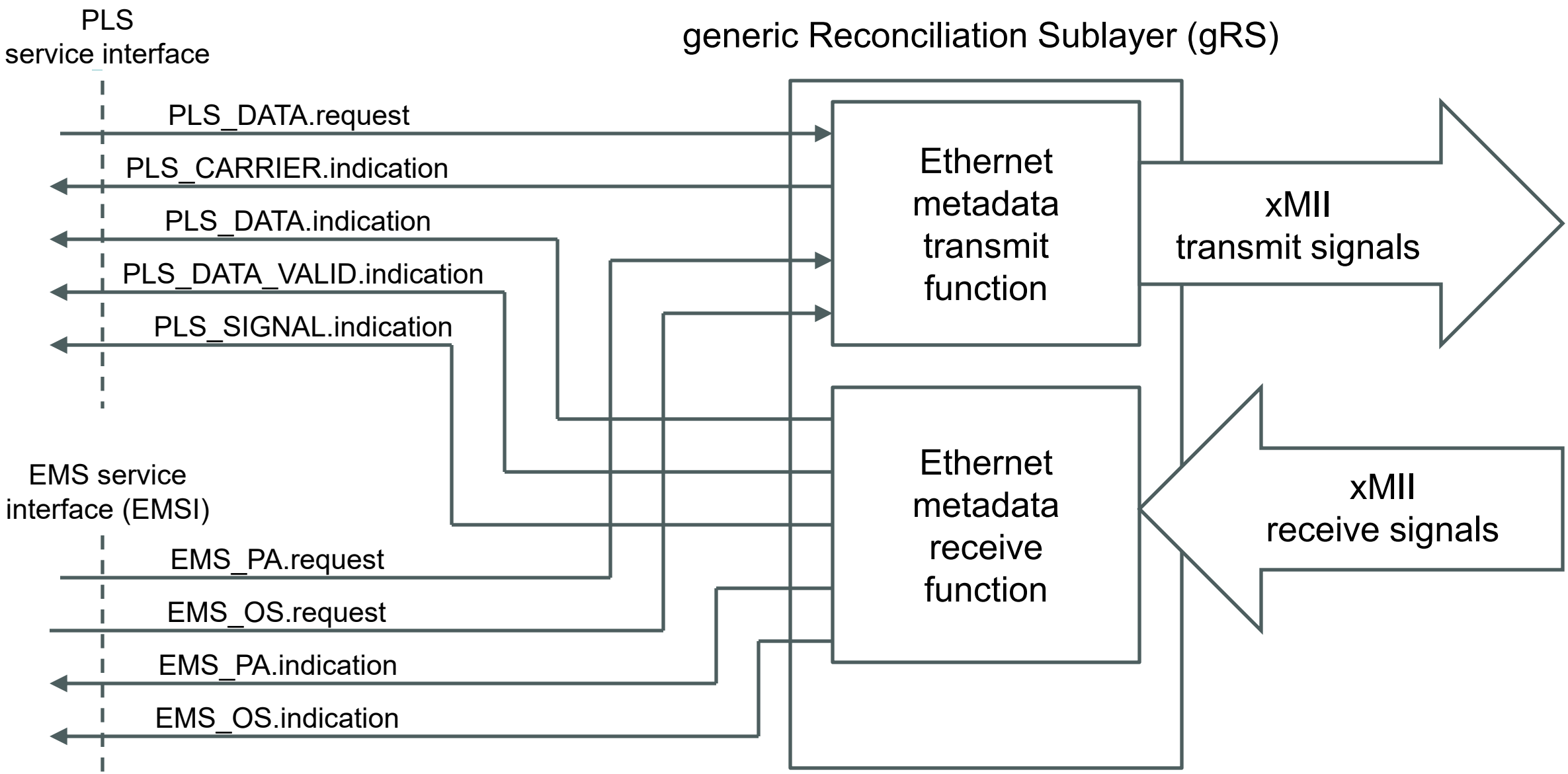



Figure XXX–2 gRS Ethernet metadata transmit and receive function

# Baseline text
# generic Reconciliation Sublayer (gRS)

A metadata capability, if implemented, shall only be enabled if it is supported by the data rate and PHY type, and only after it has been established that the link partner supports that metadata capability and the associated syntax and semantics of the metadata to be transmitted. The mechanism by which support for a metadata capability, and the corresponding syntax and semantics, is established is beyond the scope of this Clause.

The per-packet metadata capability is only supported by [TBD PHY list operating at data rates of TBD]. The packet-independent metadata capability is only supported by [PHY list] operating at data rates of 50 Gb/s or greater.

# Baseline text
# Metadata transmit function : per-packet metadata

## XXX.4.1 Metadata transmit function

### XXX.4.1.1 Per-packet metadata

If the per-packet metadata capability is enabled, upon receipt of an EMS_PA.request, the gRS shall, at the start of the next packet, discard the OUTPUT_UNIT values when servicing the PLS_DATA.request primitive associated with the first bit of the octet defined by first_replace_octet parameter, to the last bit of the octet defined by last_replace_octet parameter. The gRS shall replace the discarded bits with the data bits provided in new_preamble_data parameter on xMII transmit path.

The first_replace_octet and last_replace_octet parameters are integers in the range 1 to 6. Preamble octets are numbered using integer values 0 to 6, where preamble octet 0 is the first preamble octet transmitted by the MAC and preamble octet 6 is the last preamble octet transmitted by the MAC.

[Add any rules regarding use of non-FEC protected preambles, if supported]

An example of servicing an EMS_PA.request with the first_replace_octet parameter set to 1, the last_replace_octet parameter set to 4, and the new_preamble_data parameter set to a value D[31:0] is illustrated in figure XXX-3.

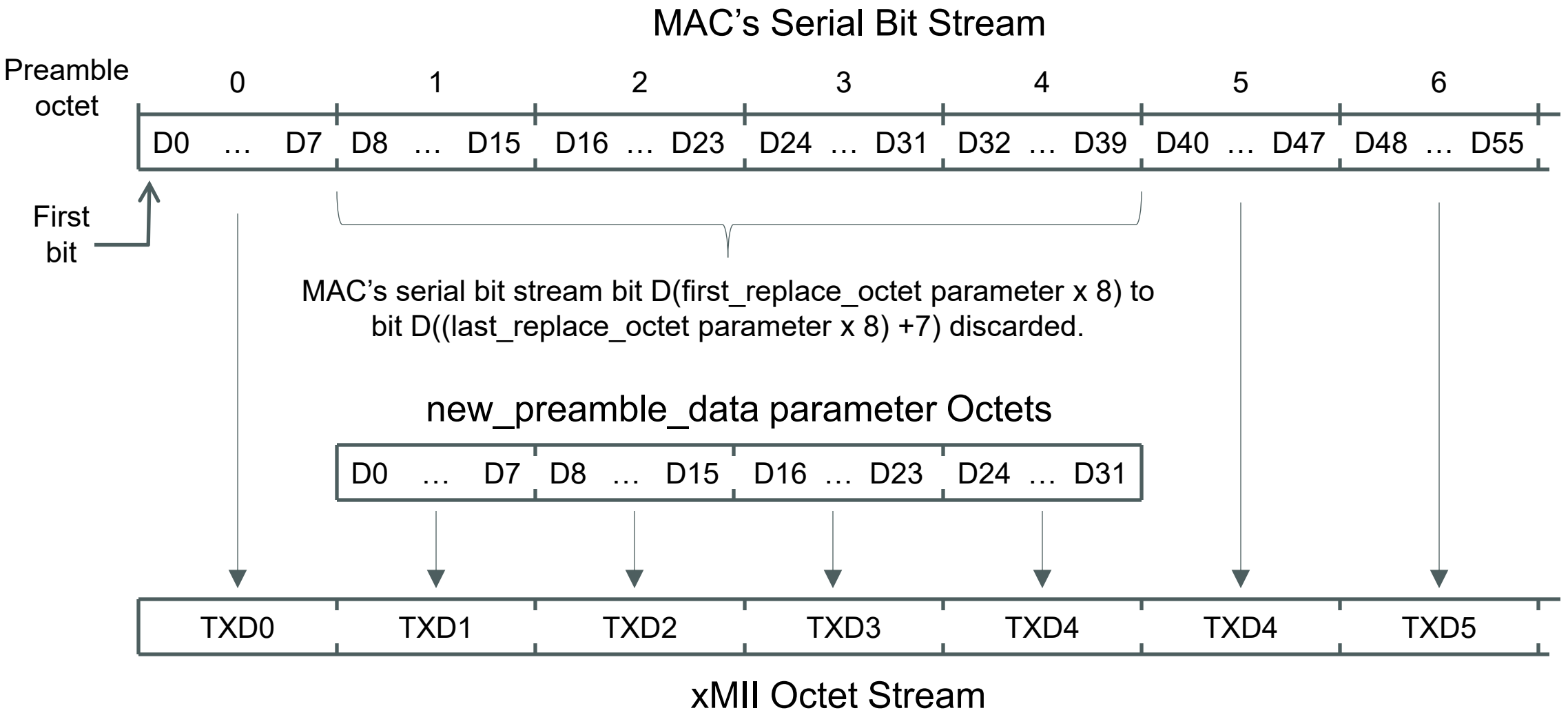



Figure XXX–3 Example of servicing an EMS_PA.request with first_replace_octet parameter set to 1 and last_replace_octet parameter set to 4

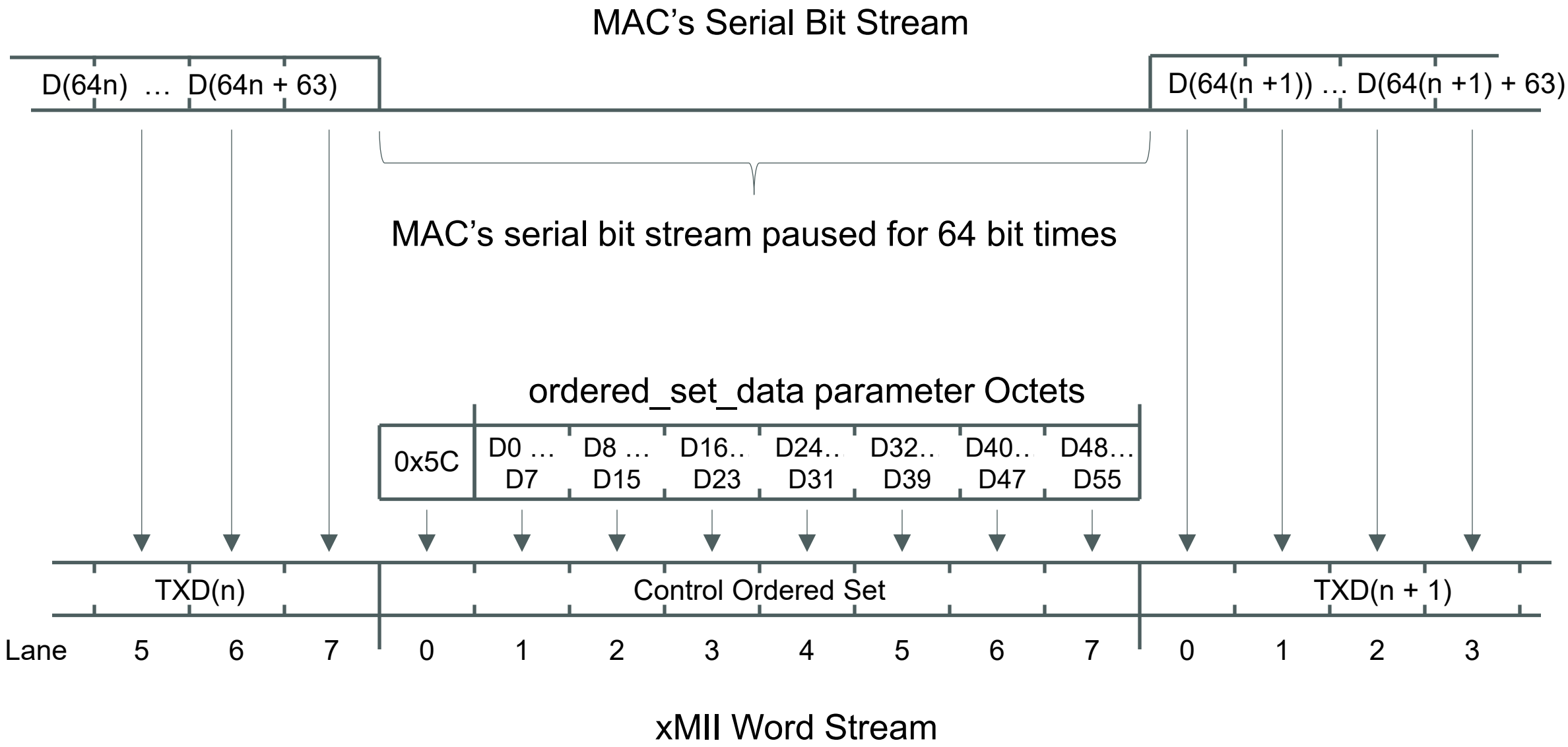
# Baseline text
# Metadata transmit function : packet-independent metadata

## XXX.4.1.2 Packet-independent metadata

If the packet independent metadata capability is enabled, the following actions will occur. Upon receipt of an EMS_OS.request during packet transmission, [add exceptions] the gRS shall interrupt packet transmission by halting servicing the MAC's PLS_DATA.request primitive for sixty-four bit times, inserting a control ordered set on the xMII transmit path during this time. Upon receipt of an EMS_OS.request during idle, [add exceptions] the gRS shall insert a control ordered set on the xMII transmit path.

The gRS shall use the ordered_set_data parameter as the source of the seven octets of the control ordered set transmitted on lanes 1 to 7 of the xMII.

Servicing an EMS_OS.request during packet transmission is illustrated in figure XXX-4.


MAC's Serial Bit Stream
D(64n) … D(64n + 63)
D(64(n +1)) … D(64(n +1) + 63)
MAC's serial bit stream paused for 64 bit times
ordered_set_data parameter Octets
0x5C
D0 … D7
D8 … D15
D16.. D23
D24.. D31
D32.. D39
D40.. D47
D48… D55
TXD(n)
Control Ordered Set
TXD(n + 1)
Lane 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3
xMII Word Stream

# Baseline text
# Metadata receive function

## XXX.4.2 Metadata receive function

### XXX.4.2.1 Per-packet metadata

If the per-packet metadata capability is enabled, when the gRS detects the eighth octet of a packet on the xMII receive signals, and there has been no per-packet metadata receive error, it shall generate an EMS_PA.indication primitive with the preamble_data parameter set to the value of the preceding seven octets received on the xMII receive signals. A per-packet metadata receive error is an error encoding on the xMII receive signals defined in the respective RS Clause during any of the first eight octet of a packet.

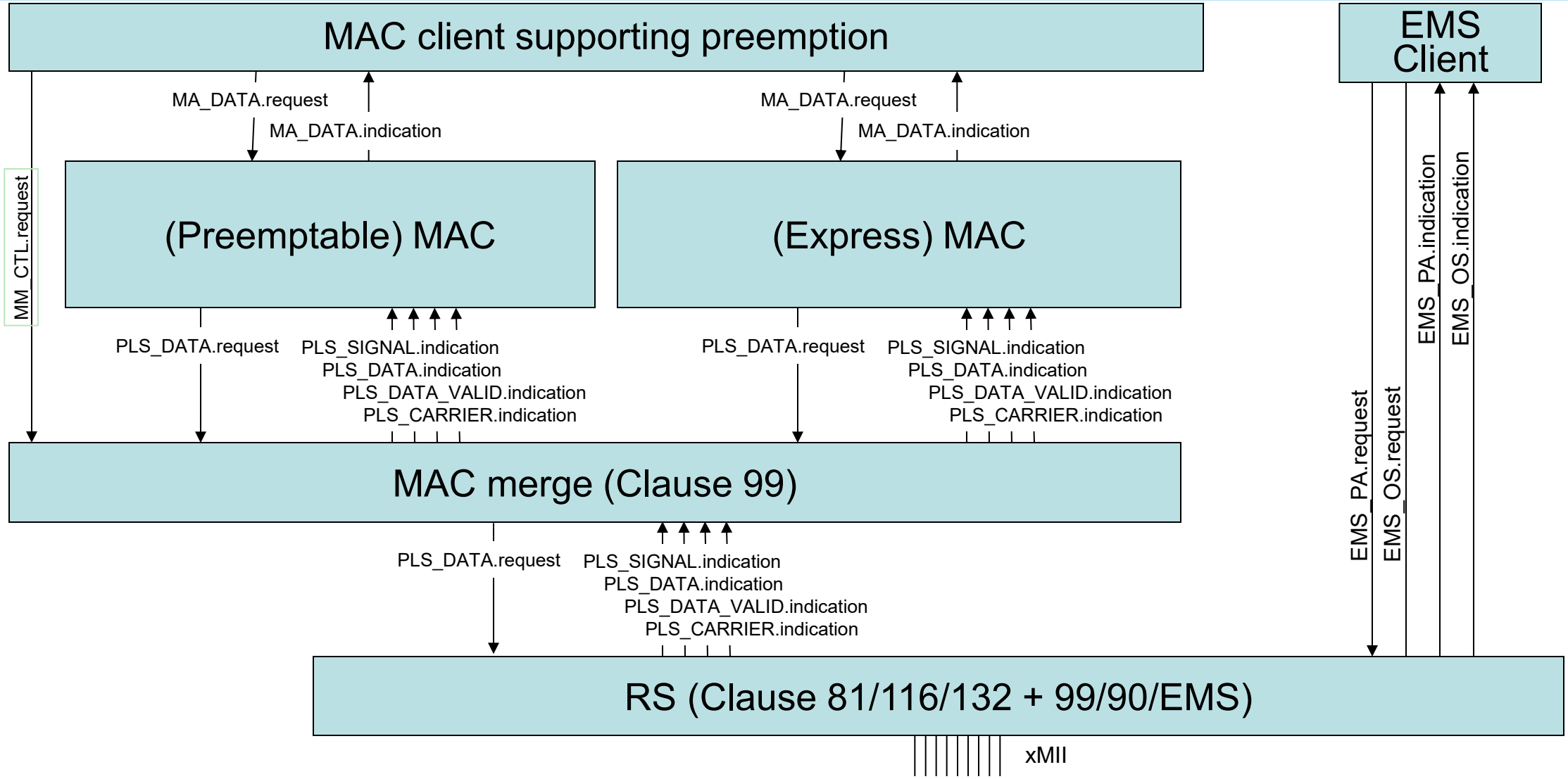
[Is the absence of a receive error during the preamble sufficient or should the EMS_PA.indication primitive only be generated absence of a receive error during the entire packet]

### XXX.4.2.2 Packet-independent metadata

If the packet-independent metadata capability is enabled, when the gRS detects a control ordered set on the xMII receive path, it shall generate an EMS_OS.indication primitive with the ordered_set_data primitive set to the value of the seven octets of the control ordered set received on lanes 1 to 7 of the xMII.

# Open items

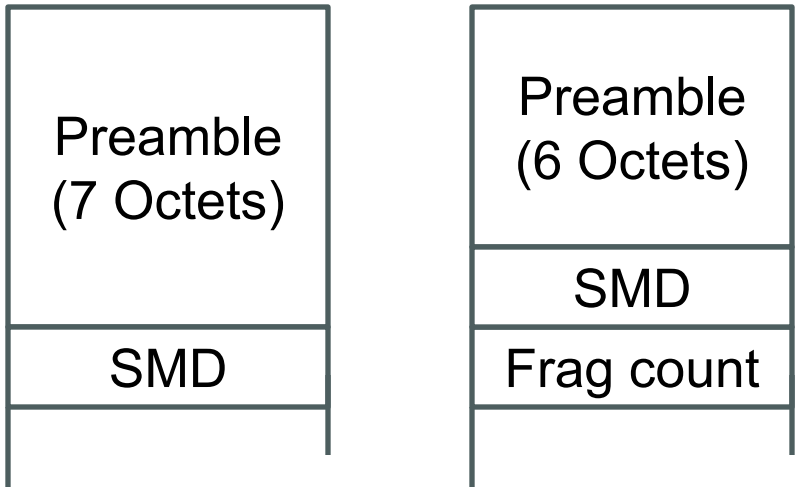
# MAC Merge Sublayer



# MAC Merge Sublayer

- MAC Merge sublayer sits above the RS
  - MAC Merge passes MAC Merge Packet (mPacket) to RS
- mPacket format
  - Six or seven preamble octets
  - Start mPacket Delimiter (SMD)
- Types of mPacket are indicated by the SMD value
  - Express MAC (eMAC) packet
  - A complete preemptable MAC (pMAC) packet
  - Initial fragment of a pMAC packet
  - Remaining fragment(s) of a pMAC packet
- How to associate per-packet metadata service
  - Could add additional primitives or parameters
  - Any approach would appear to require a seven-octet FIFO in the RS
    - RS doesn't know mPacket type until the SMD is received from the MAC Merge

See Figure 99-4 mPacket format




eMAC packet
complete pMAC packet,
initial pMAC fragment

Remaining
pMAC
fragment(s)

# Per-packet metadata

- MAUs/PHYs that use preamble and SFD
  - Example: 10BASE-T PHY with MII
  - Preamble used for RxCLK and SFD used for alignment (see 22.2.3.2.2)
- Continuous signalling PHYs
  - Examples:100BASE-X (4B/5B), 1000BASE-X (8B/10B), 200GBASE-R
  - No preamble loss, but not all preamble octets are transferred
    - 10BASE-T1L replaces the first two octets with a Start-of-Stream delimiter (SSD) symbol (146.3.3.1)
    - 100BASE-T1 replaces the first nine bits of preamble with SSD (96.3.3.2)
    - 100BASE-X replaces the first preamble octet with SSD (24.2.2.1.4)
    - 1000BASE-T replaces the first two preamble octets with SSD (40.3.1.3)
    - XGMII, XLGMII and CGMII replace the first preamble octet with start control character (see 46.2.2)
- Subset of continuous signalling PHYs with FEC-protected bit stream
  - Examples: 25GBASE-T1 and 50GBASE-R
  - FEC-protected bit stream provides error protection for the preamble data

# Questions

Thank you!