IEEE P802.3az Energy-efficient Ethernet architecture

David Law
3Com

David_Law@3Com.com

Contributor

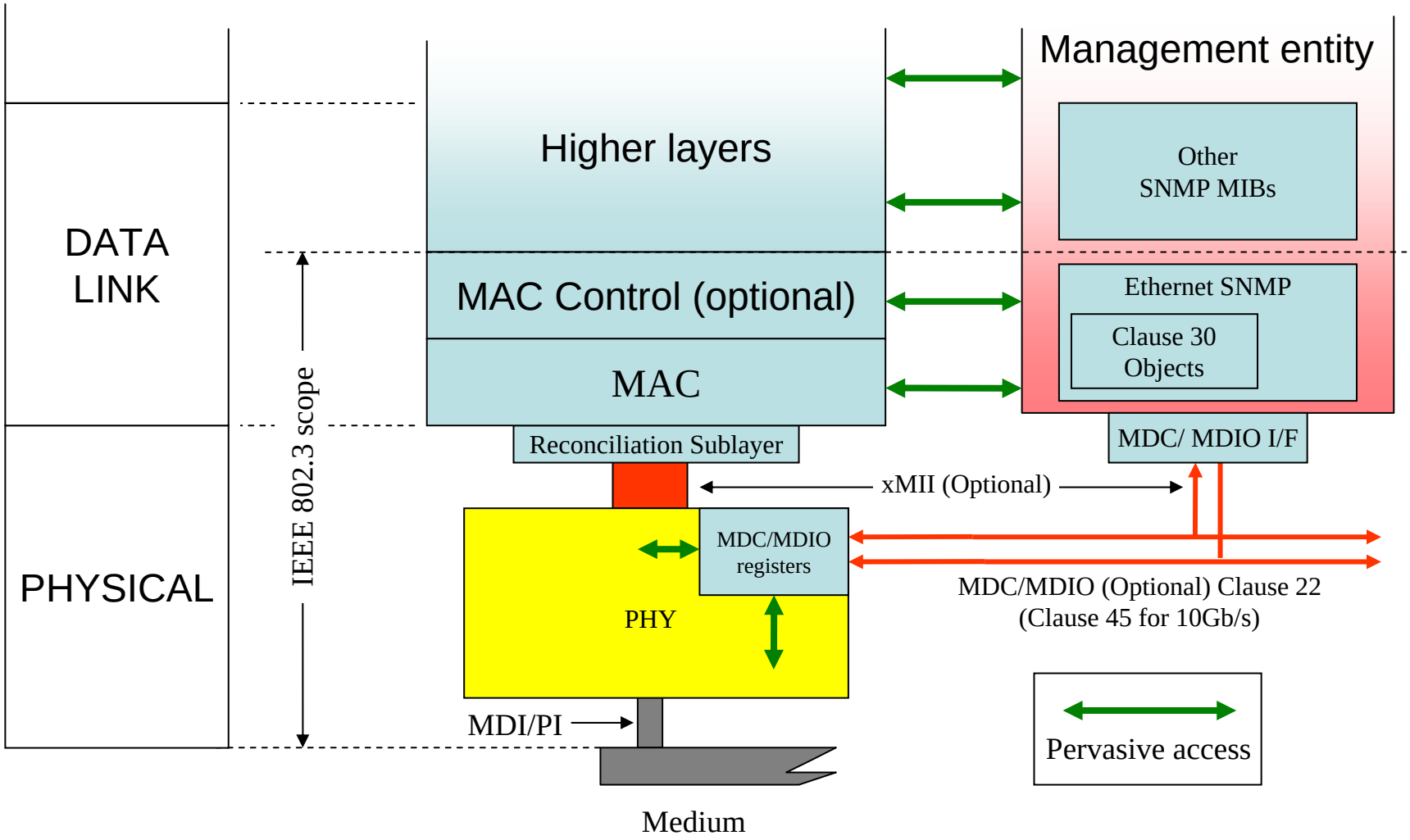
- Bob Grow Intel

Introduction

- Support LPI baseline proposals
- Not saying anything can't be implemented
- Need to specify externally observable behaviour
  - Any implementation that meets this conforms
  - Not sure we are there yet

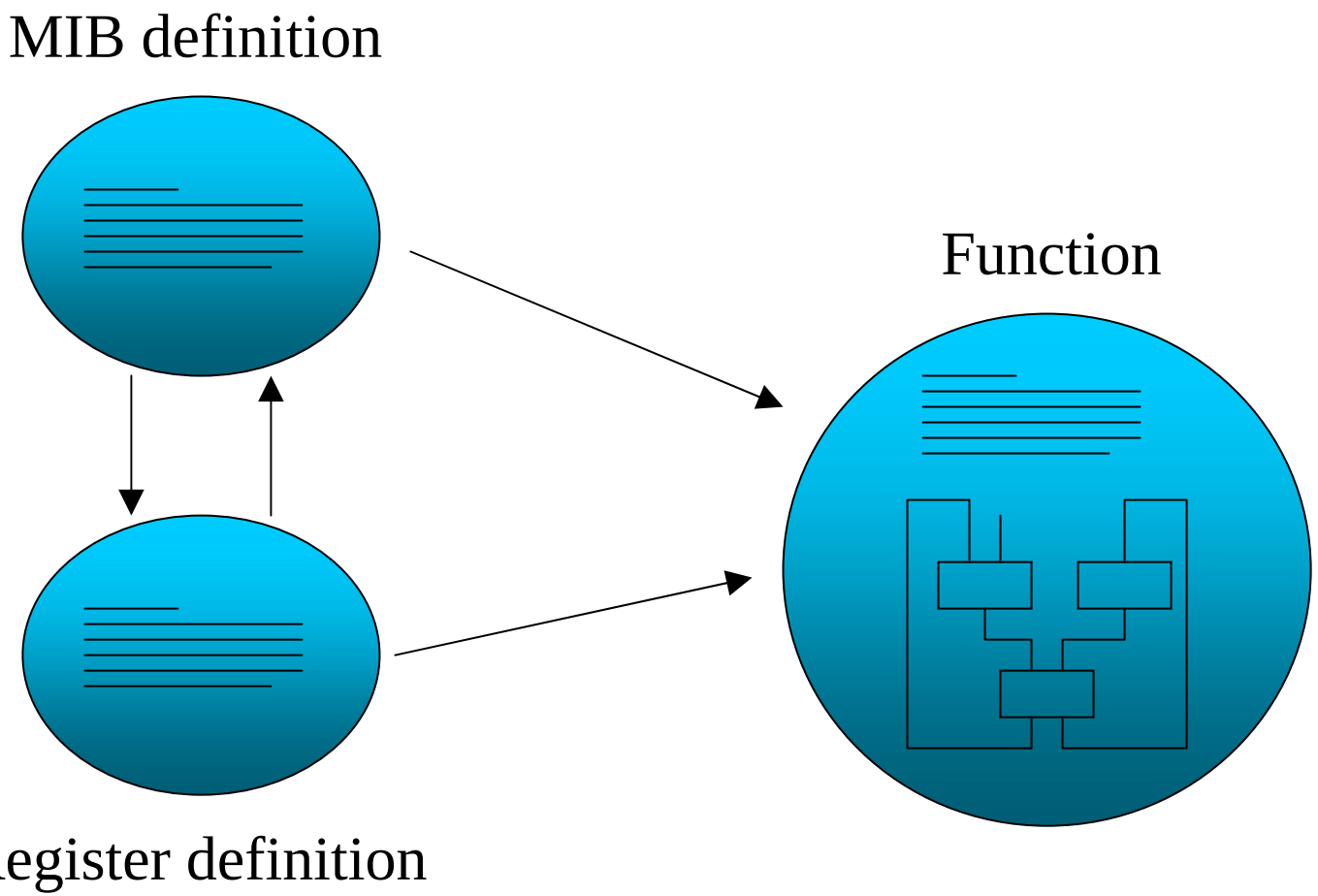
    (not unreasonable since only in Task Force review)
- Need architecture that supports all speeds
- Need architecture that supports future projects

Review of management architecture

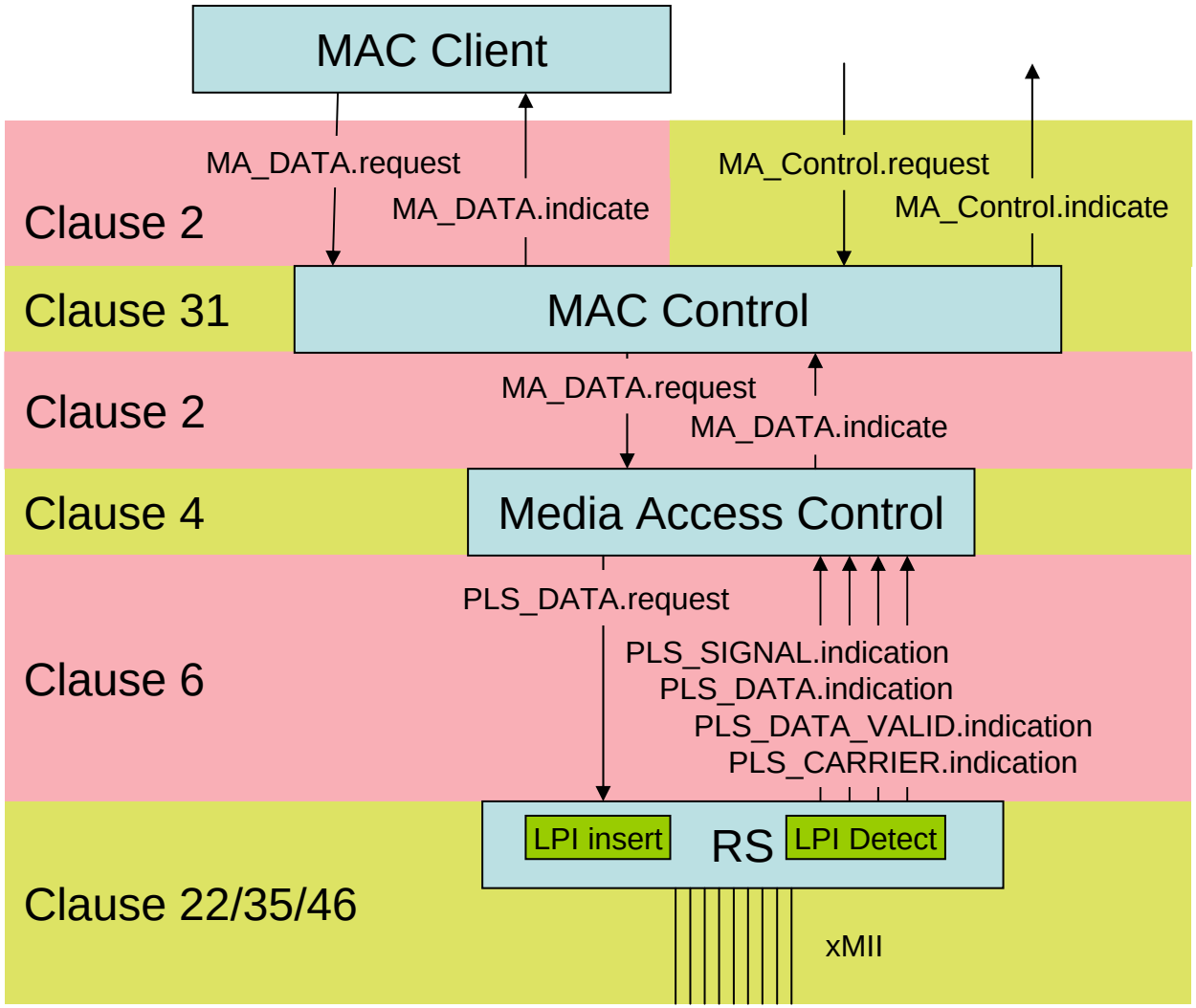


MIB, Registers and Function

Function in PHY needs register access to make it manageable



Review of sublayers and interfaces



Abstract Service Interface

- Clause 2 MAC Service Interface example

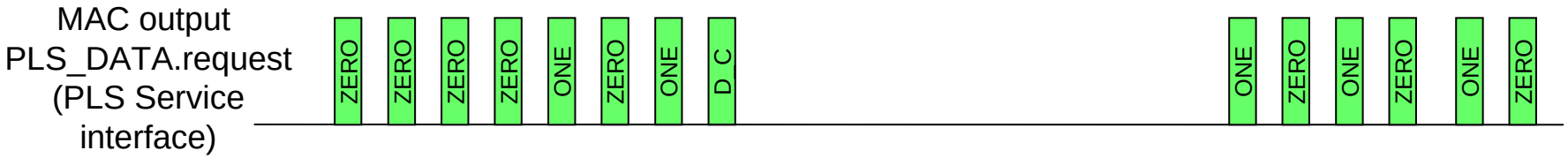
  MA_DATA.request (destination_address, source_address, mac_service_data_unit, frame_check_sequence)

  This primitive defines the transfer of data from a MAC client entity to a single peer entity or multiple peer entities in the case of group addresses.

- Clause 6 Physical Signaling (PLS) service Interface example

  PLS_DATA.request (OUTPUT_UNIT)

  The OUTPUT_UNIT parameter can take on one of three values: ONE, ZERO, or DATA_COMPLETE and represent a single data bit. The DATA_COMPLETE value signifies that the Media Access Control sublayer has no more data to output.



Low Power Idle models in draft

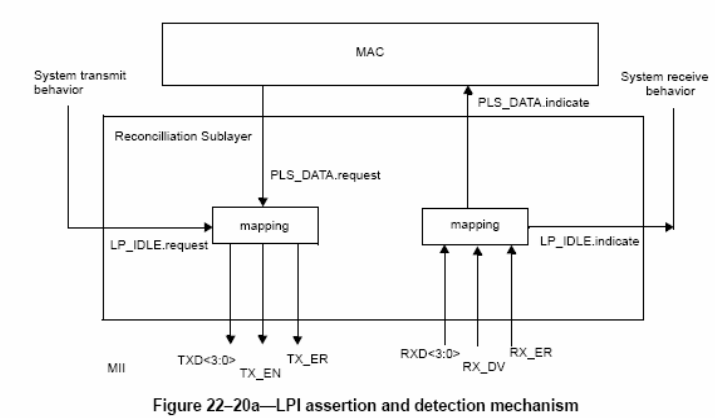


Figure 22–20a—LPI assertion and detection mechanism

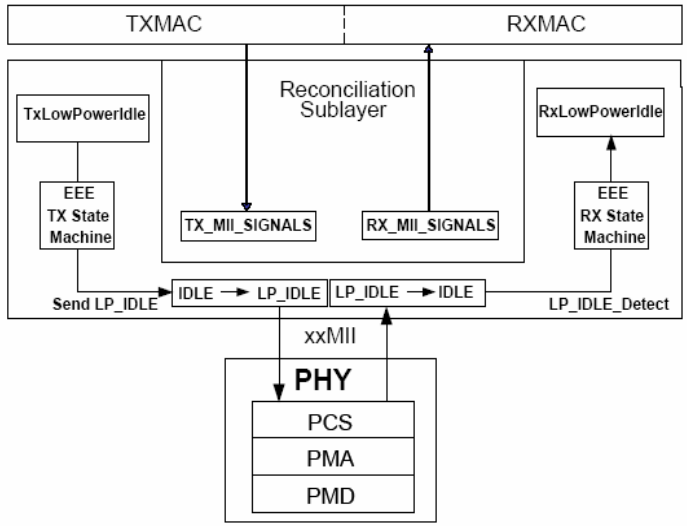


Figure 78–1—EEE Low Power Idle Generation and Detection Functionality

- Models aren't consistent
  - LPI controlled by primitive in one, a RS state machine in other
- Textual description is also different

  78.2.1 Low Power mode states description

  LP_IDLE: This codeword is used by MAC to force Transmit path of the local PHY to enter Low Power Mode
  - Yet no change proposed to MAC
- Will assume intent was primitive control of Assert Low Power Idle

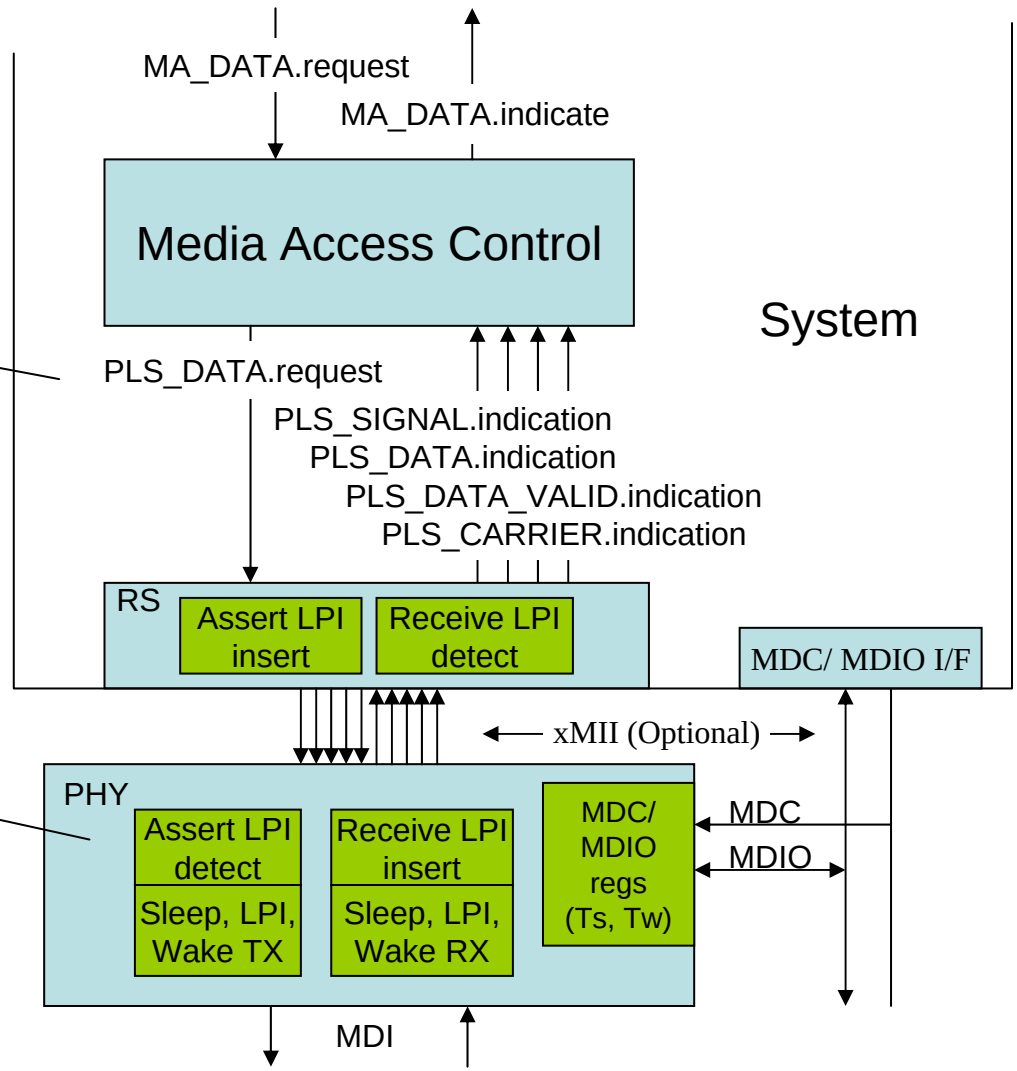
LPI basic model

PLS_DATA.request (OUTPUT_UNIT)

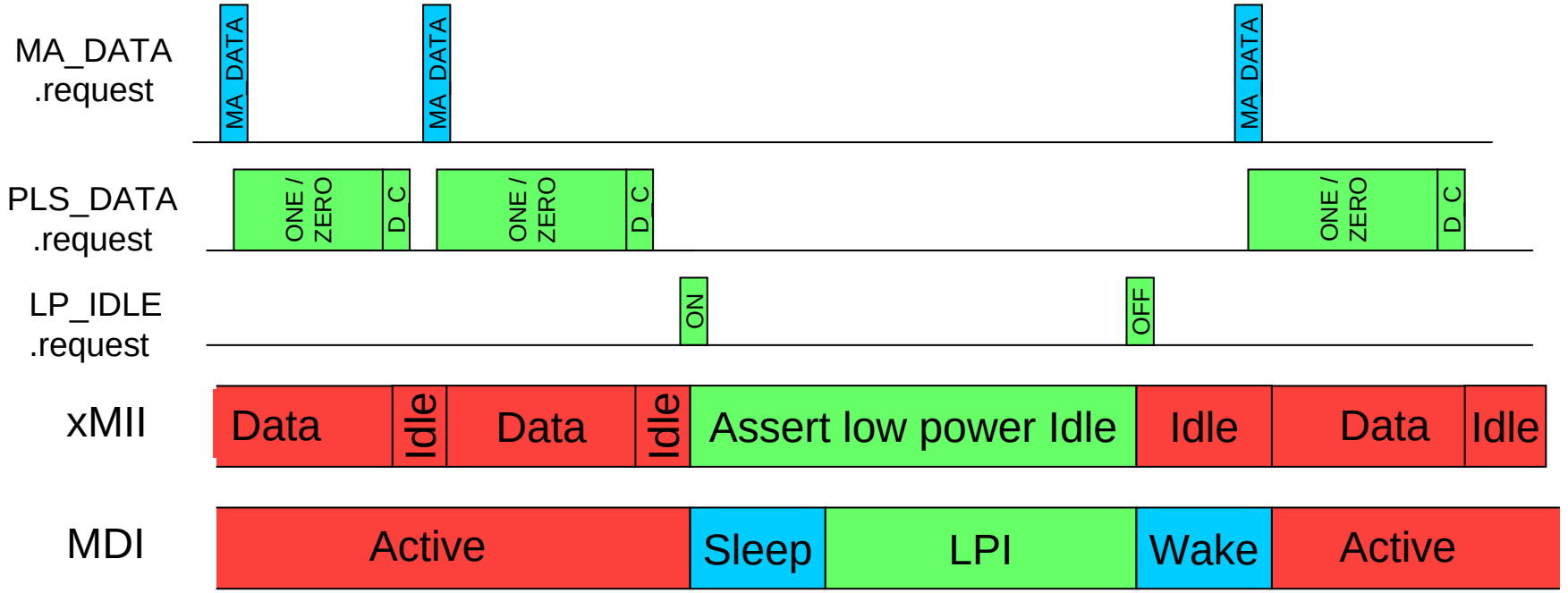
The OUTPUT_UNIT parameter can take on one of three values: ONE, ZERO, or DATA_COMPLETE and represent a single data bit. The DATA_COMPLETE value signifies that the Media Access Control sublayer has no more data to output.

(IEEE Std 802.3 Subclause 6.3.1.1)

Sleep, LPI and Wake have minimum timing requirements (Ts and Tw) but since there are no changes to the MII the 'Assert LPI insert' function in the RS has to meet these timing requirements <u>open loop</u>.



MII and MDI operation



Assert Low Power Idle (ALPI) constraints:

[1] Assert Low Power Idle cannot occur until min time after DATA_COMPLETE

[2] Assert Low Power Idle minimum duration > Ts + any minimum LPI duration

[3] Assert Low Power Idle must be negated Tw before next Packet (ONE/ZERO)

These apply to LP_IDLE.request if it maps directly to ALPI

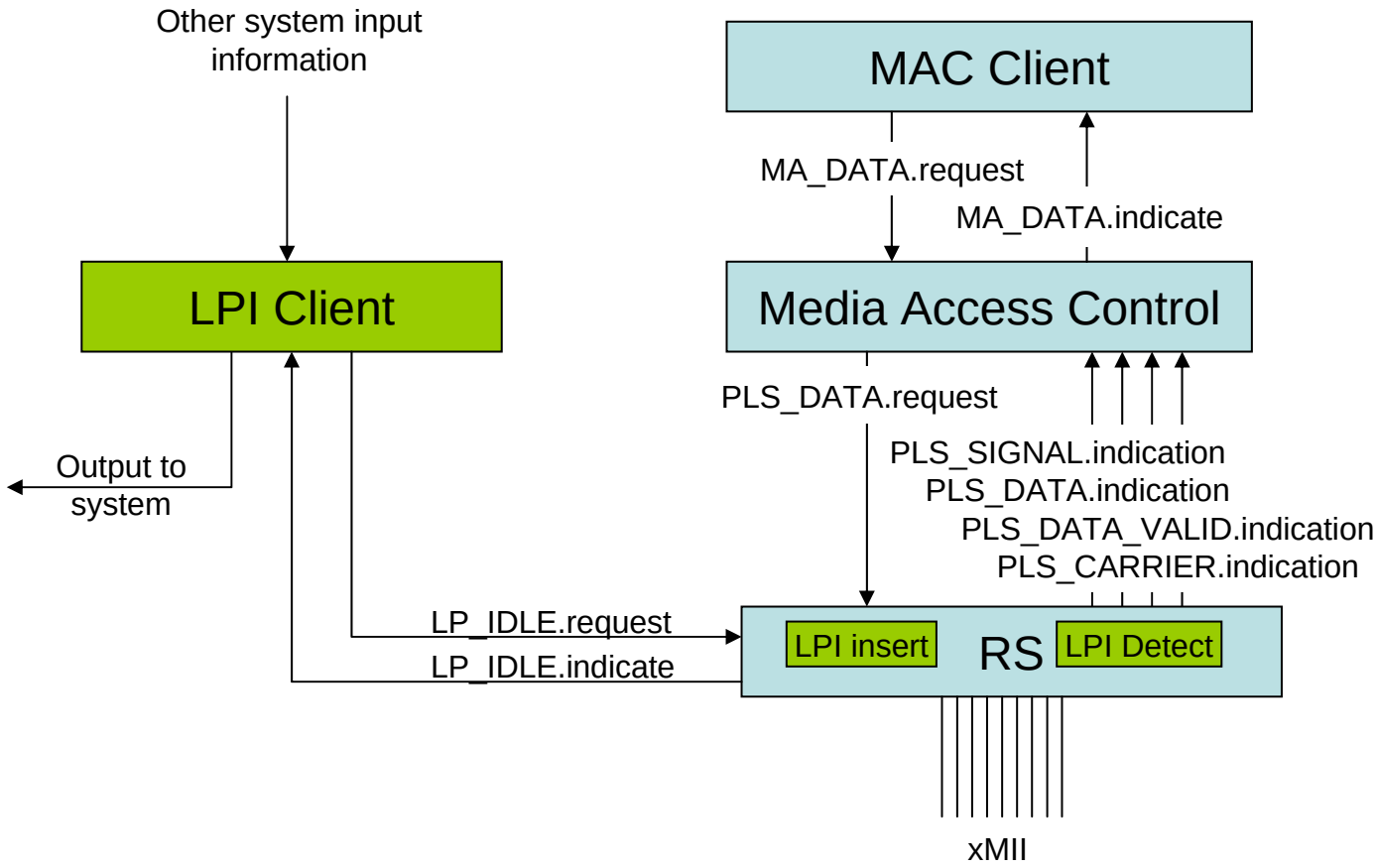
MAC Client constraints:

[1] Time to next MA_DATA.request > Ts + any minimum LPI duration + Tw

Options for specifying

- [1] Provide set of rules for Clients
  - Require LPI Client enforce ALPI timing
  - Require MAC Client enforce MA_DATA timing
    - Places requirements on specifications outwith 802.3
- [2] Provide a TX LPI MII State Diagram
  - State Diagram controls timing
  - Use PLS_CARRIER.indication to enforce MA_DATA timing
    - Only supports full-duplex mode
- [3] Provide a LPI Control sublayer
  - LPI Control State Diagram controls all timing
- [4] Close the loop
  - Don’t support xMII
    - Assume pervasive access
  - Don’t support physical implementation of xMII
    - Add a new signal from the PHY to the RS across the xMII

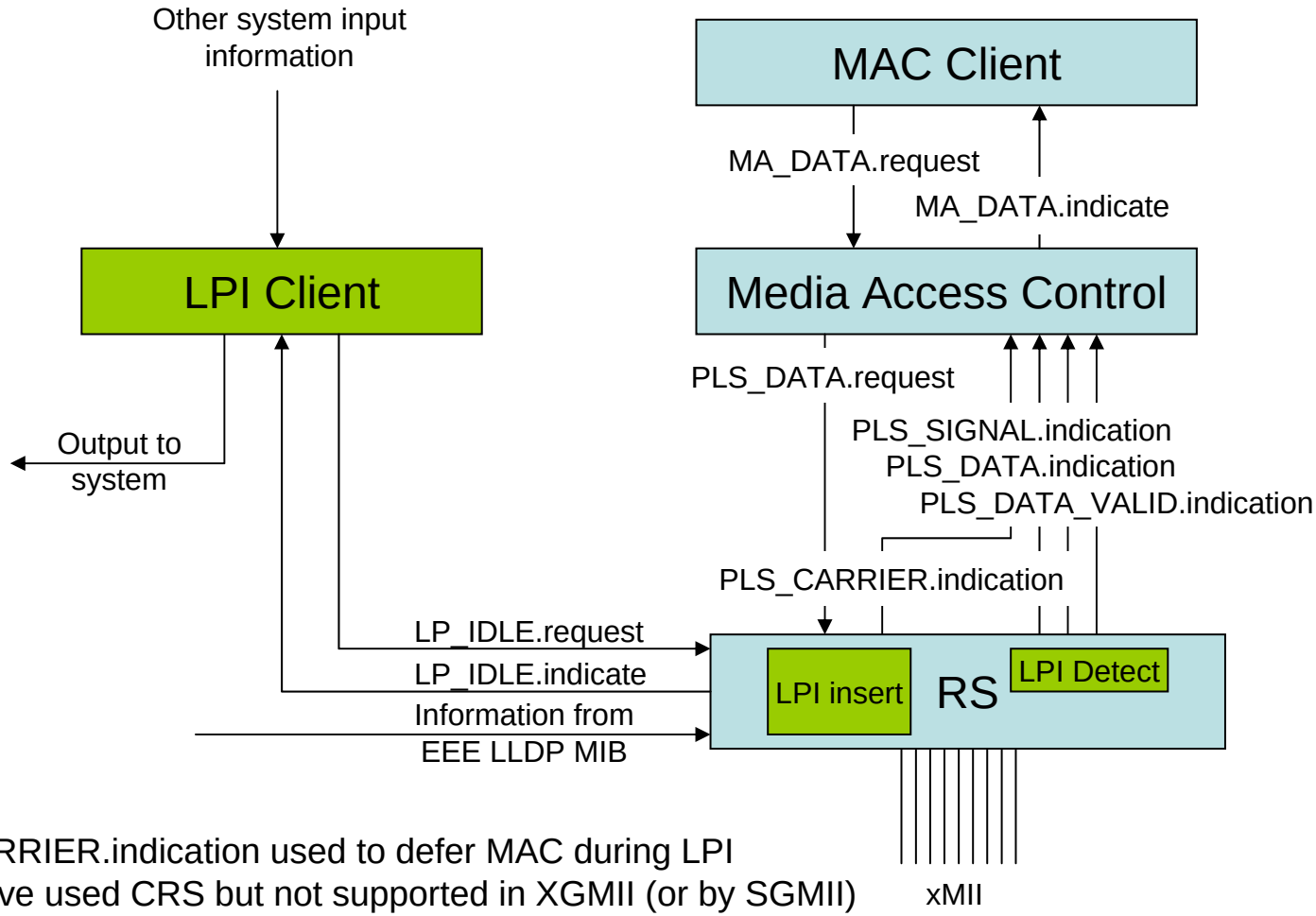
Option 1 – Client rules model



Option 1 – Client rules

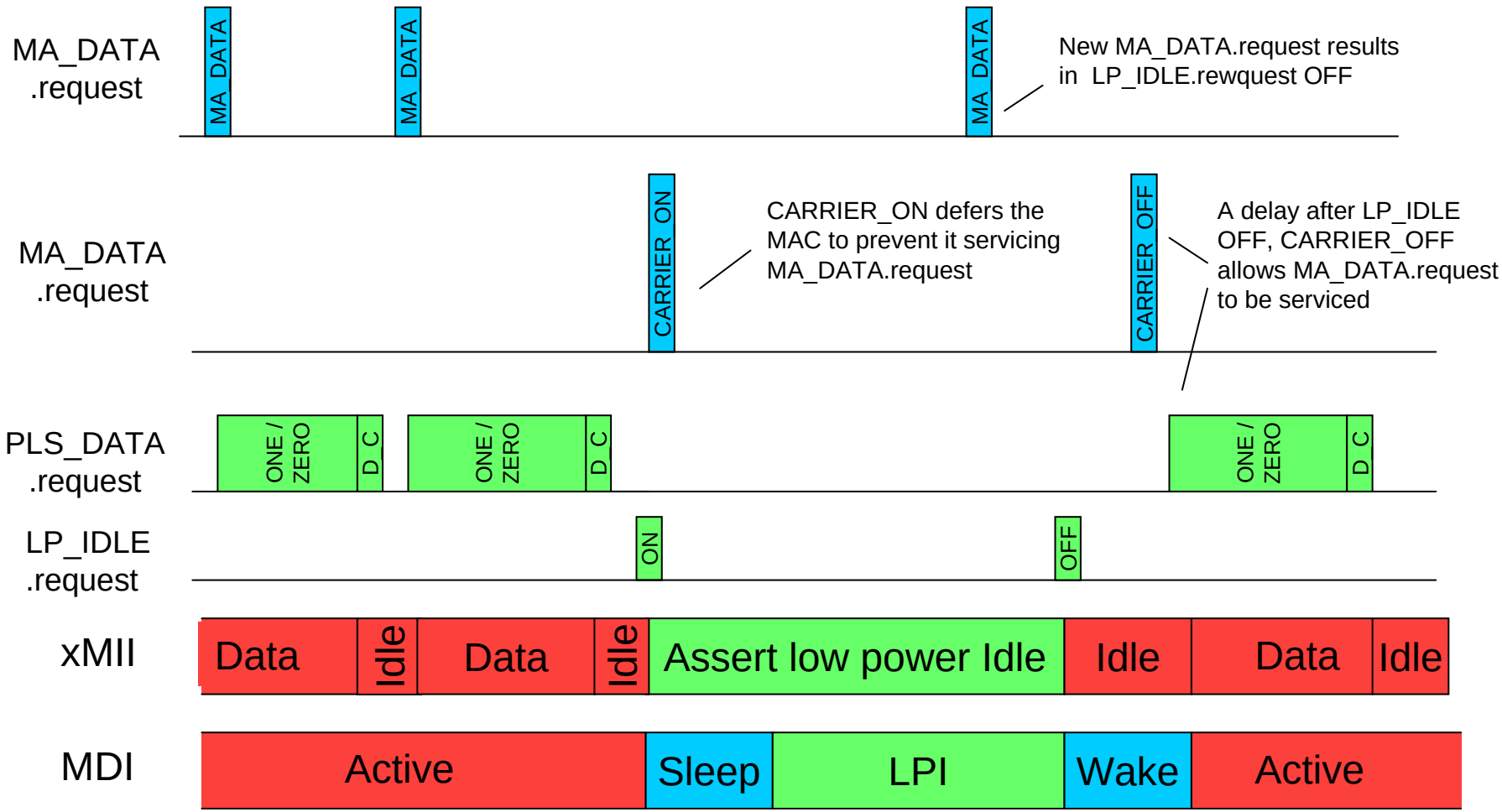
- Assume LPI Client has pervasive access
  - LPI Client Rules
    - Assert Low Power Idle cannot occur until a min delay after DATA_COMPLETE
    - Assert Low Power Idle minimum duration > Ts + any minimum LPI duration
    - Assert Low Power Idle must be negated Tw before end of DATA_COMPLETE
  - MAC Client Rules
    - Time to next MA_DATA.request > Ts + any minimum LPI duration + Tw
    - Must be met by all MAC Clients
      - MAC Control Client, OAM Client
- Don’t normally place constraints on clients
  - Control through servicing requests

Option 2 – RX TX LPI state machine



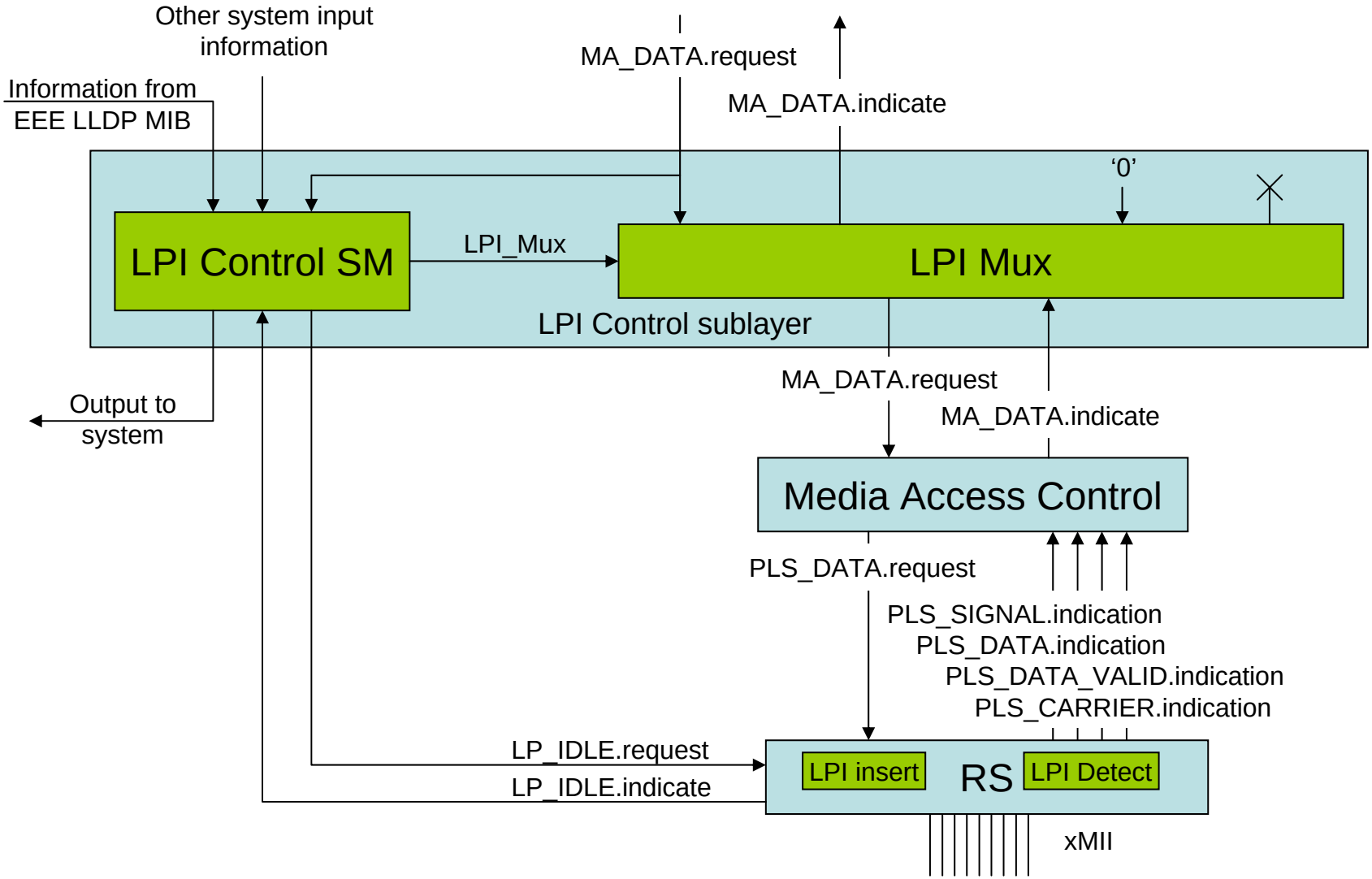
PLS_CARRIER.indication used to defer MAC during LPI
Could have used CRS but not supported in XGMII (or by SGMII)

Option 2 – MII and MDI operation

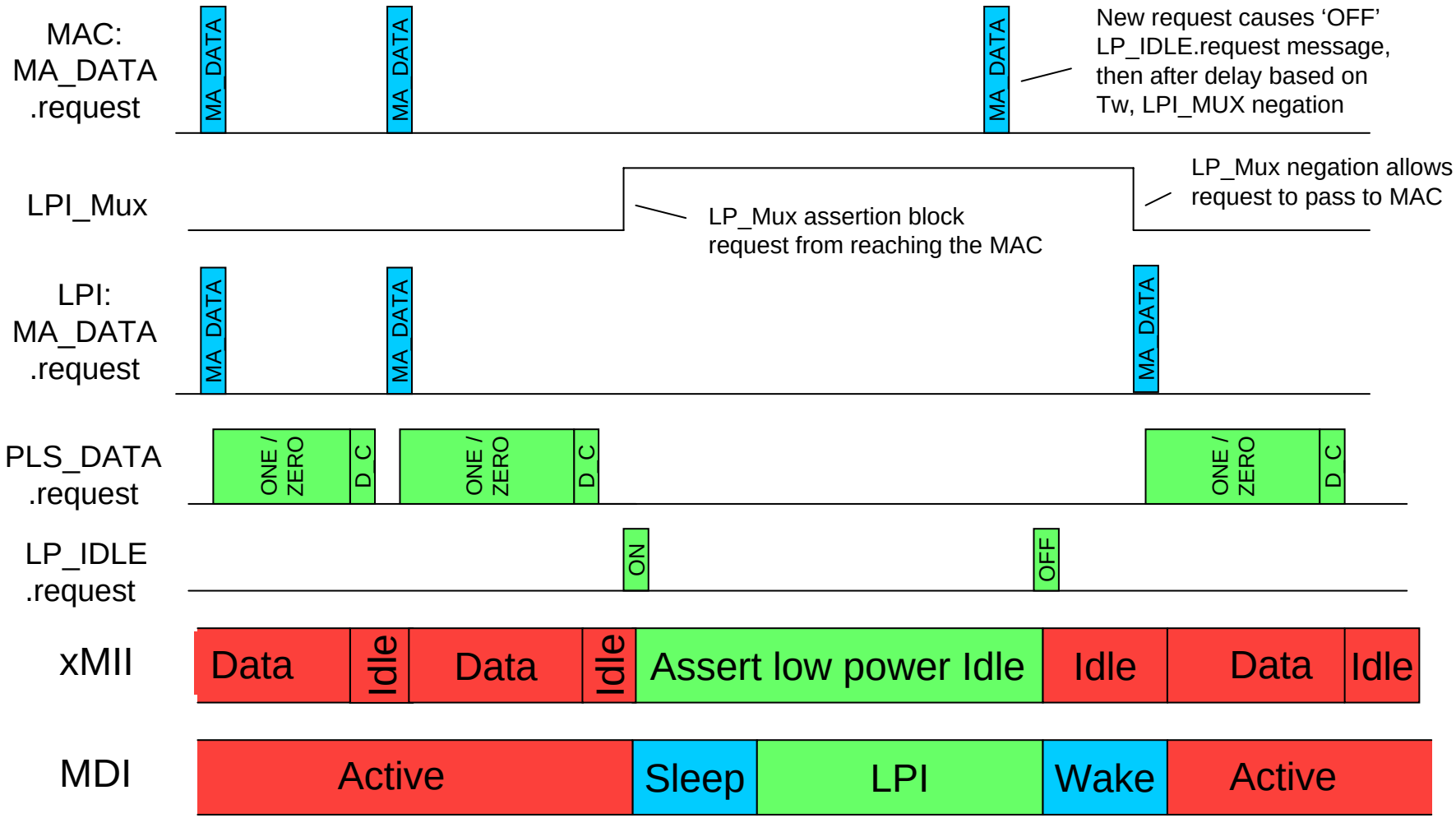


LPI and MAC Client rules enforced by LPI Control SM

Option 3 – LPI Control sublayer

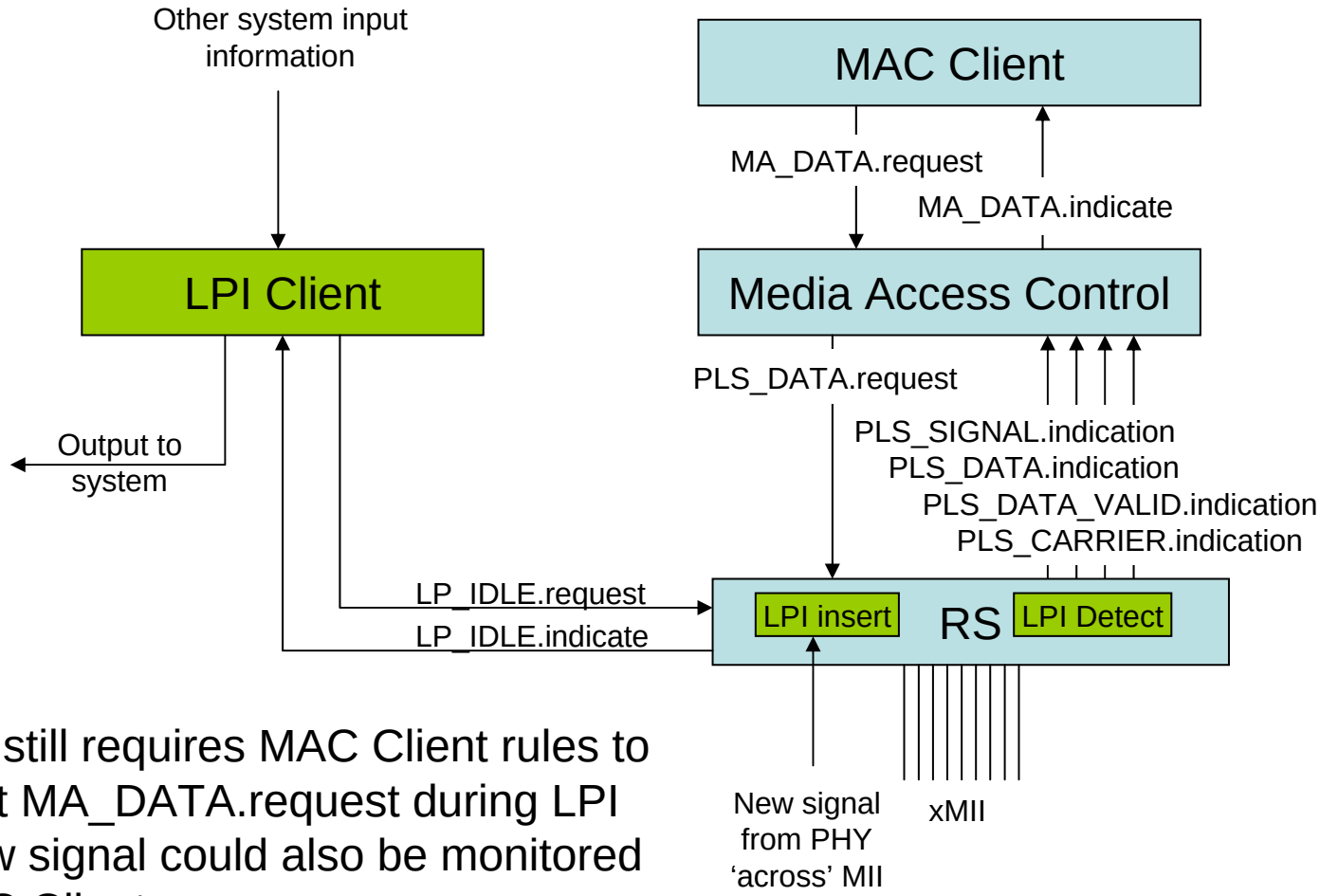


Option 3 – MII and MDI operation



LPI and MAC Client rules enforced by LPI Control SM

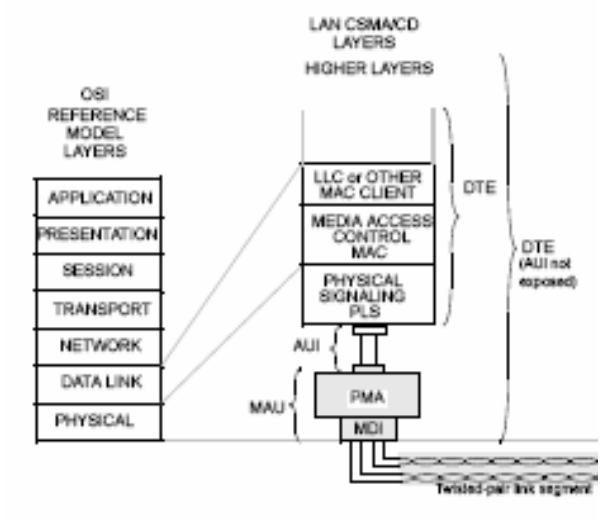
Option 4 – Don’t support xMII



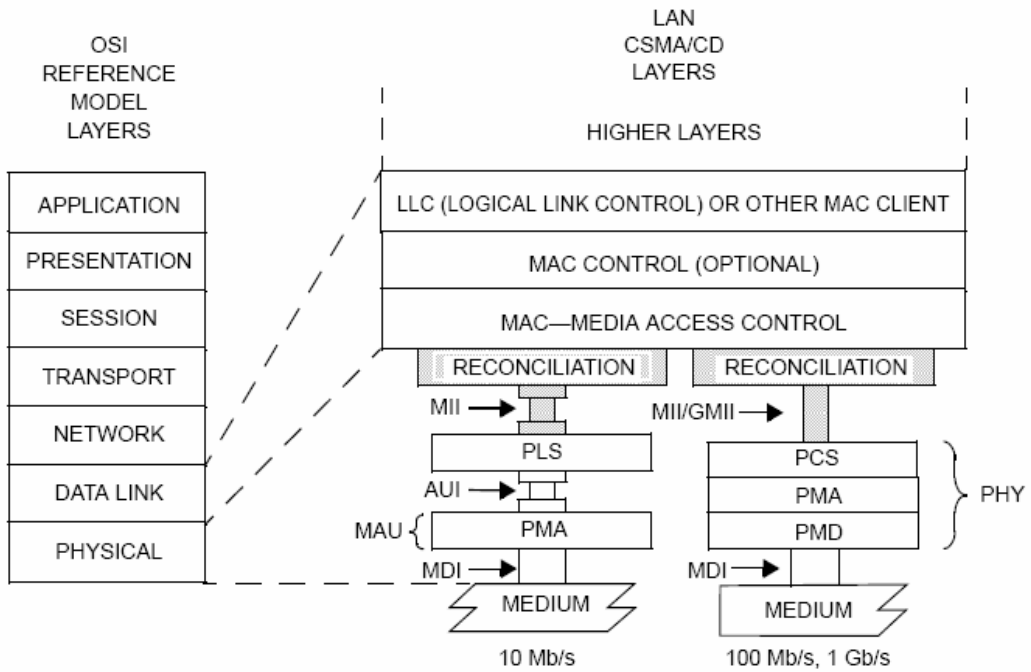
Note – still requires MAC Client rules to prevent MA_DATA.request during LPI but new signal could also be monitored by MAC Client.

10BASE-T

Clause 14 – 10BASE-T



Clause 22 - MII



- 10BASE-T MAU is specified as being below AUI
- 10BASE-T behaviour specified in terms of AUI signalling
  - AUI doesn't support Assert Low Power Idle (ALPI)
- What happens in case of PLS and AUI below MII
  - Option: specify ALPI never sent for 10Mb/s operation

10BASE-T

- Do we want to support LPI at 10BASE-T
  - 10BASE-T idle already lowest power
    - 10BASE-Te will lower it further
    - But significant savings based on LPI based system sleep
  - No sleep and wake defined
    - How will far end system enter and exit sleep
- Can packets be used at 10Mb/s
  - Sleep is start of IPG
  - Wake is start of Preamble

☺ Works across AUI

☹ Different architecture from other speeds