

Partitioning RS functions with POPI

G. ZIMMERMAN/CME CONSULTING

7/12/2026

Overview

Problem statement

RS-stack architecture

How POPI might address this

Questions to consider for discussion

Problem statement

IEEE Std 802.3 partitions a number of functions into the RS which are tightly coupled to the PHY

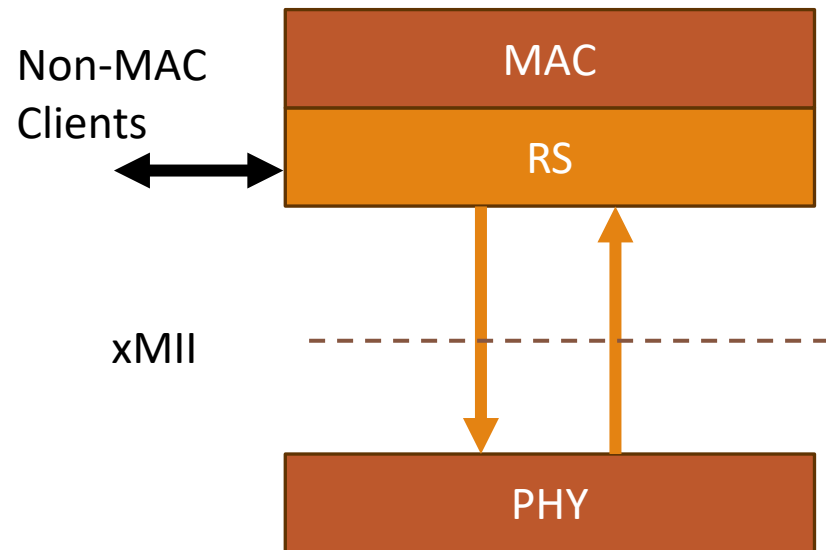
The normal model for an RS is that it interfaces with signals to the PHY (usually to the PCS) and with primitives to the MAC or non-MAC clients

This results in PHY implementations which incorporate RS functions on the PHY side of the xMII boundary – contrary to the standard architecture

Notable examples: PLCA, Timesync (TSSI), and generation of LPI in EEE

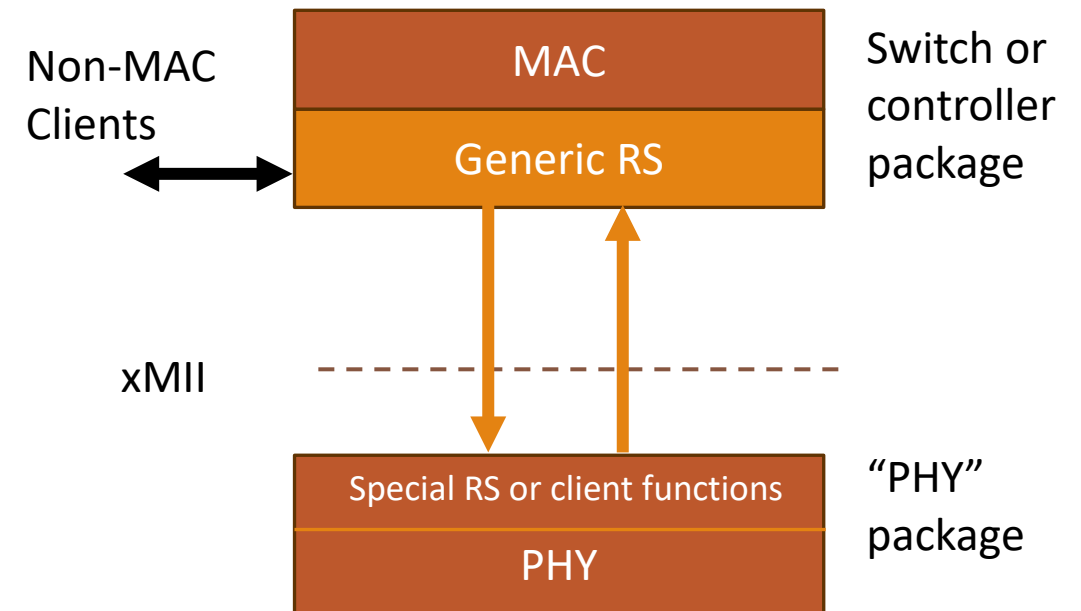
Usual architectural model vs. PHY implementation

Model in IEEE Std 802.3-2022



See, e.g., Figure 78-1, 90-1, or 148-1 & 148-2

Example implementation (EEE)



Special RS or client functions may include, for example, assertion of LPI, inserting timesync information, or handling PLCA state diagrams

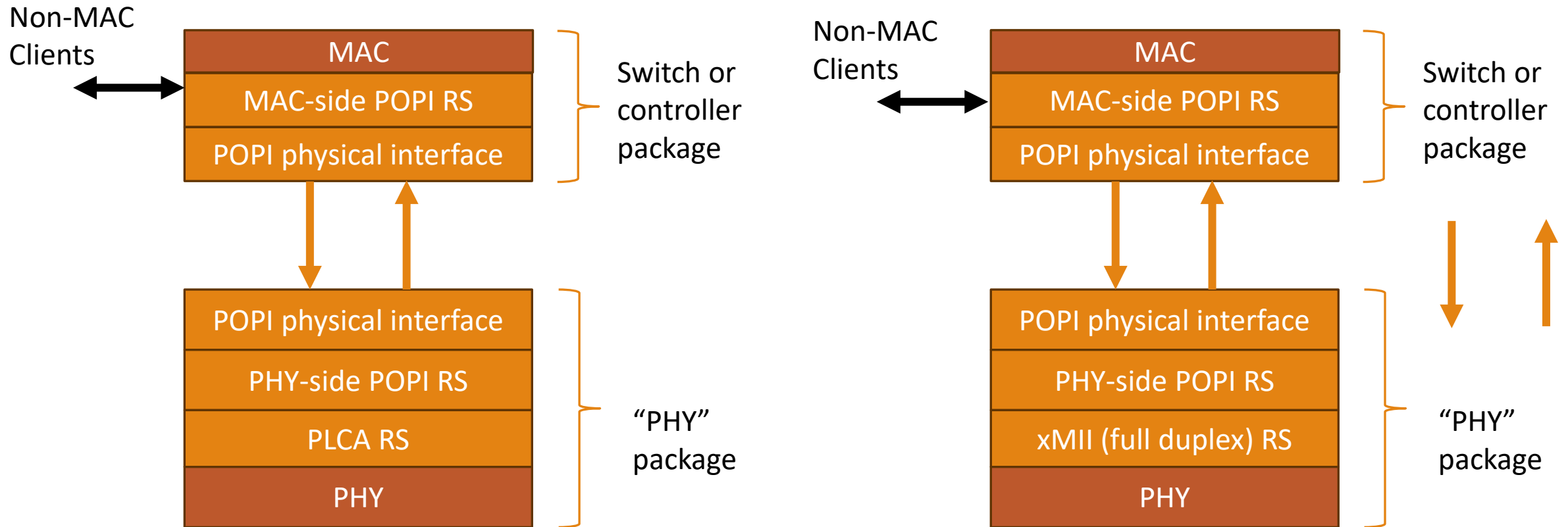
We have a chance to fix this

By specifying POPI as a physical interface and as an RS we can say architecturally how it fits

POPI necessarily will NOT contain the specialized functions of, for example, a clause 90 or 148 RS, or the LPI client

BUT, if we specify POPI right, we can layer these RS's and carry the necessary information

Potential POPI “RS stack” architecture



How might we accomplish this?

Primitives would need to be encoded/decoded by the POPI RS on either side

- Asymmetry in primitives creates a 'phy-side' and 'mac-side' POPI RS

Would need a general facility for encoding/decoding new primitives to allow for expandability

- Macroframe structure could provide this

Questions

Can the MAC-side POPI RS be a generic PCS abstraction layer?

Is the PHY-side RS an application-specific shim layer to map to existing MII, GMII, XGMII and handle specialized functions?

- If so, primitives would be encoded in the POPI data stream to be decoded by a complementary POPI RS on the other side

How much generality can we get?

- What primitives are generic?
 - Media not available (pause transmission – PHY RS to POPI RS signaling)
 - Periodic control (allows phy or system to provide timing)
- What primitives are specific?
 - PLCA: specific PLCA control – skip TO, grab TO, packet pending,...
 - Timesync
 - Does FMPS want something specific?
 - Other traffic prioritization?

Allowing a priority 8-byte macroframe that gets transported in place of a data timeslot. (pausing data transfer on any frame in process from either direction) might accomplish that

- Can the PHY -> RS primitive interfaces do this naturally? (the MAC->RS does...)
- If not, the 'shim RS' could buffer...

Discussion?

Thank you!
