OAM Proposal

IEEE 802.3bp - Interim Meeting - January 2015

William Lo, Marvell

OAM Proposal

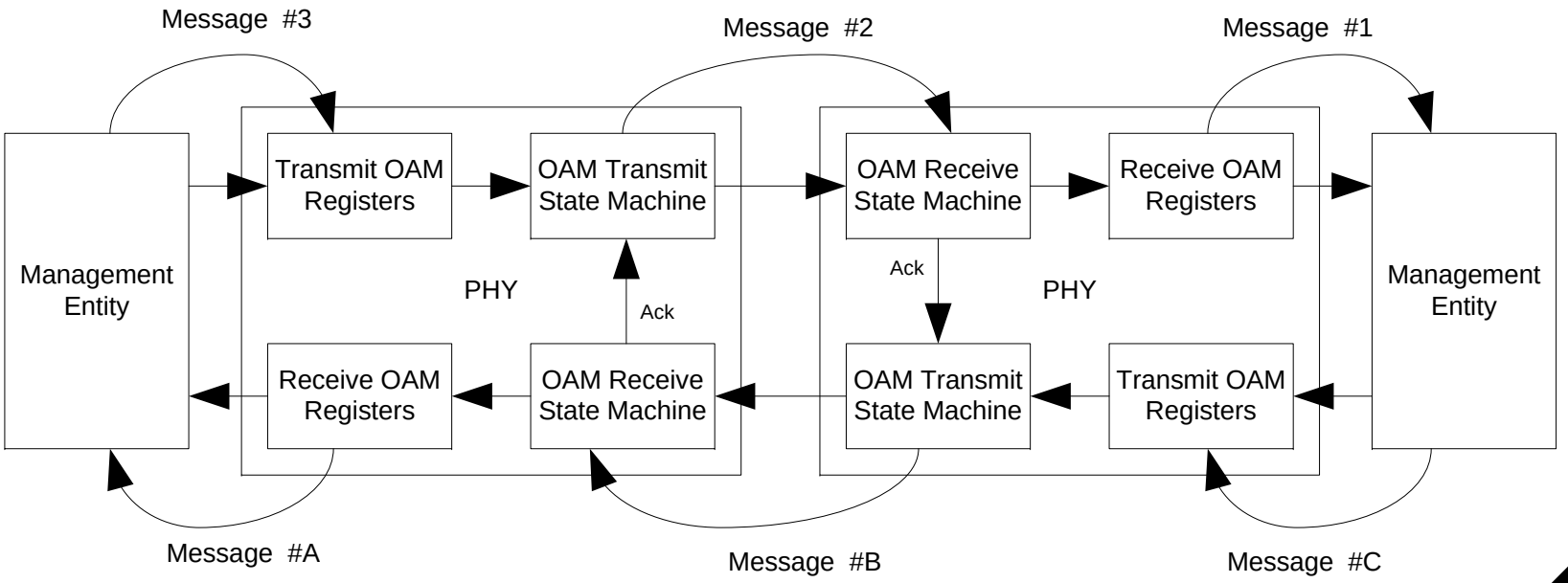
- OAM adopted as objective in Nov 2014 meeting without any detailed description on the mechanics (Matheus_3bp_01_1114.pdf)
- Current proposal reuses 12 byte info field mechanism that is used during training
  - Training and OAM are mutually exclusive so some hardware can be reused
  - Has 10 byte payload and same CRC16
- Balances hardware use in PHY vs. desire to exchange many bytes
  - Hardware to exchange 8 byte messages
  - Messages can be numbered to send 16 different 8 byte messages = 128 bytes
- Remains active during low power idle
- OAM message definition is beyond the scope of the task force
- OAM in this discussion occurs in the PCS layer
  - Normally when OAM is mentioned it occurs in the link layer or higher
  - We are defining OAM that only lives in the 1000BASE-T1 PCS layer

Definitions

- **OAM field**
  - 9-bit symbol that is inserted in the RS frame during normal operation, or XOR into the refresh cycle during LPI
- **OAM frame**
  - A collection of 12 consecutive 9-bit symbols
- **OAM message**
  - 8 byte message plus 4 bits that identify one of 16 possible message numbers
- **Management Entity**
  - Agent that exchange management information with its peer. The interaction occurs over the OAM transmit and receive registers.
- **Transmit OAM Registers**
  - Information is loaded into these registers to be transmitted to the link partner
- **Receive OAM Registers**
  - Information that is received is read from these registers

Model

- Management entity asynchronously interact with OAM registers
- All bytes of OAM Messages transferred between registers and state machines. Must make sure all bytes are transferred at the same time so that there is no split message.
- Up to 3 OAM messages can be in-flight in each direction
- Received OAM messages are acknowledged



OAM Frame

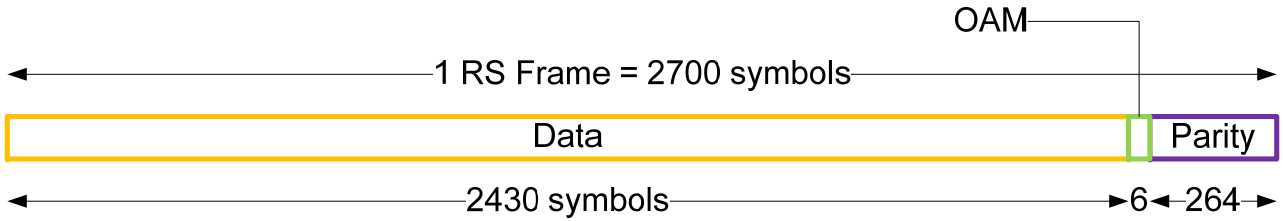
- 12 symbols of 9-bits each
- CRC16 $(x+1)(x^{15}+x+1)$ of the previous 10 bytes lower 8 bits only.
- 9th bit is parity. Even parity marks OAM frame boundary

	D8	D7	D6	D5	D4	D3	D2	D1	D0
Symbol 0	Even Parity	Reserved	Reserved	Reserved	Reserved	PingRx	PingTx	SNR<1>	SNR<0>
Symbol 1	Odd Parity	Valid	Toggle	Ack	TogAck	Message_Number<3:0>			
Symbol 2	Odd Parity	Message<0><7:0>							
Symbol 3	Odd Parity	Message<1><7:0>							
Symbol 4	Odd Parity	Message<2><7:0>							
Symbol 5	Odd Parity	Message<3><7:0>							
Symbol 6	Odd Parity	Message<4><7:0>							
Symbol 7	Odd Parity	Message<5><7:0>							
Symbol 8	Odd Parity	Message<6><7:0>							
Symbol 9	Odd Parity	Message<7><7:0>							
Symbol 10	Odd Parity	CRC16							First bit
Symbol 11	Odd Parity	Final bit	CRC16						

OAM Symbol Insertion

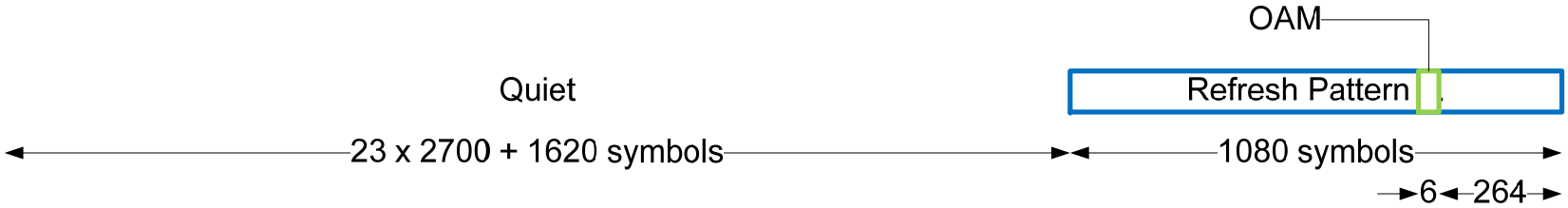
Normal operation

- One 9-bit OAM symbol inserted into OAM field in each RS Frame
- OAM field is scrambled and converted to PAM-3 in RS Frame (6 PAM3 symbols)
- 12 RS Frames per OAM Frame ➔ 12 x 3.6us = 43.2 us per OAM frame



Low Power Idle

- One 9-bit OAM symbol inserted into OAM field in each Refresh cycle
- OAM field is XOR into the scrambled sequence and converted to PAM-3 in refresh cycle
- 12 refresh cycle per OAM Frame ➔ 12 x (24 x 3.6 us) = 1036.8 us per OAM frame



Transition in and out of LPI

- OAM frame time between 43.2 us to 1036.8 us depending on number of RS frames vs. refresh cycles

OAM Frame Acceptance Criteria

- All symbol parity bits are as expected
- CRC16 is good
- RS frame is error free, or correctable
  - During LPI no such protection

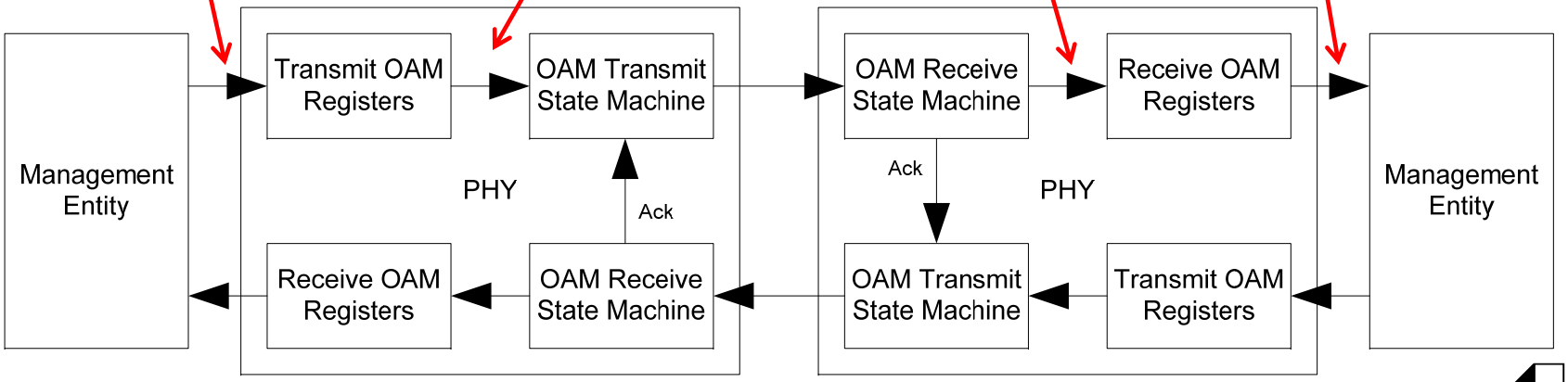
OAM Messages

- User defined Message_Number<3:0> - 16 possible messages
- User defined Message<7:0><7:0> - 8 byte message
- Valid bit – Indicates whether message is valid
- Toggle bit – Delineates between adjacent messages
- Ack bit – Acknowledgment that message was received
- AckTog bit – Toggle bit value of the message being acknowledged

- OAM message may be repeated over multiple OAM frames until acknowledge is received

Asynchronous Interaction

- Agent reads a register to see if Transmit OAM register available to load next OAM message. If so then load next message.
- Agent reads a register to see if Receive OAM register has any valid message pending. If so then read the message.
- Receive state machine will not transfer next message and will not acknowledge next message until agent reads Receive OAM register.
- Transmit state machine will not send next message until receive state machine is ready for it and only if a valid message is loaded into the Transmit OAM register



Acknowledge

- When message is received correctly by state machine and stored into the Receive OAM register, an acknowledge is sent back to let the other PHY know that message was received correctly.
- Transmit state machine knows it can send next OAM message if it receives an acknowledge.
- Pipe will eventually stall with 3 OAM messages if the agent does not read OAM message out since acknowledge cannot be sent
- No tight coupling between Management Entities. Simply poll to see whether OAM registers are available to write or read.

Other OAM signaling

- Exchanged on OAM frame level (not on the OAM message level)
- PHY Health (SNR)
  - SNR OK
  - SNR Marginal
  - Request link partner to exit LPI (used in EEE only. See Lo_3bp_03_0115.pdf)
  - Link is dying and will drop and re-link within 2 to 4ms
- Ping bits
  - Link partner reflect back PHY's Ping bit to test that link partner PHY is reacting independent of the management entity
- If OAM is not implemented then OAM field should always be set to all 0s

THANK YOU