OAM Tutorial

Current as of IEEE P802.3ah/D1.3™

- **Presenter:**
  - **Kevin Daines - EFM OAM Editor**
- **Contributor:**
  - **Jonathan Thatcher**

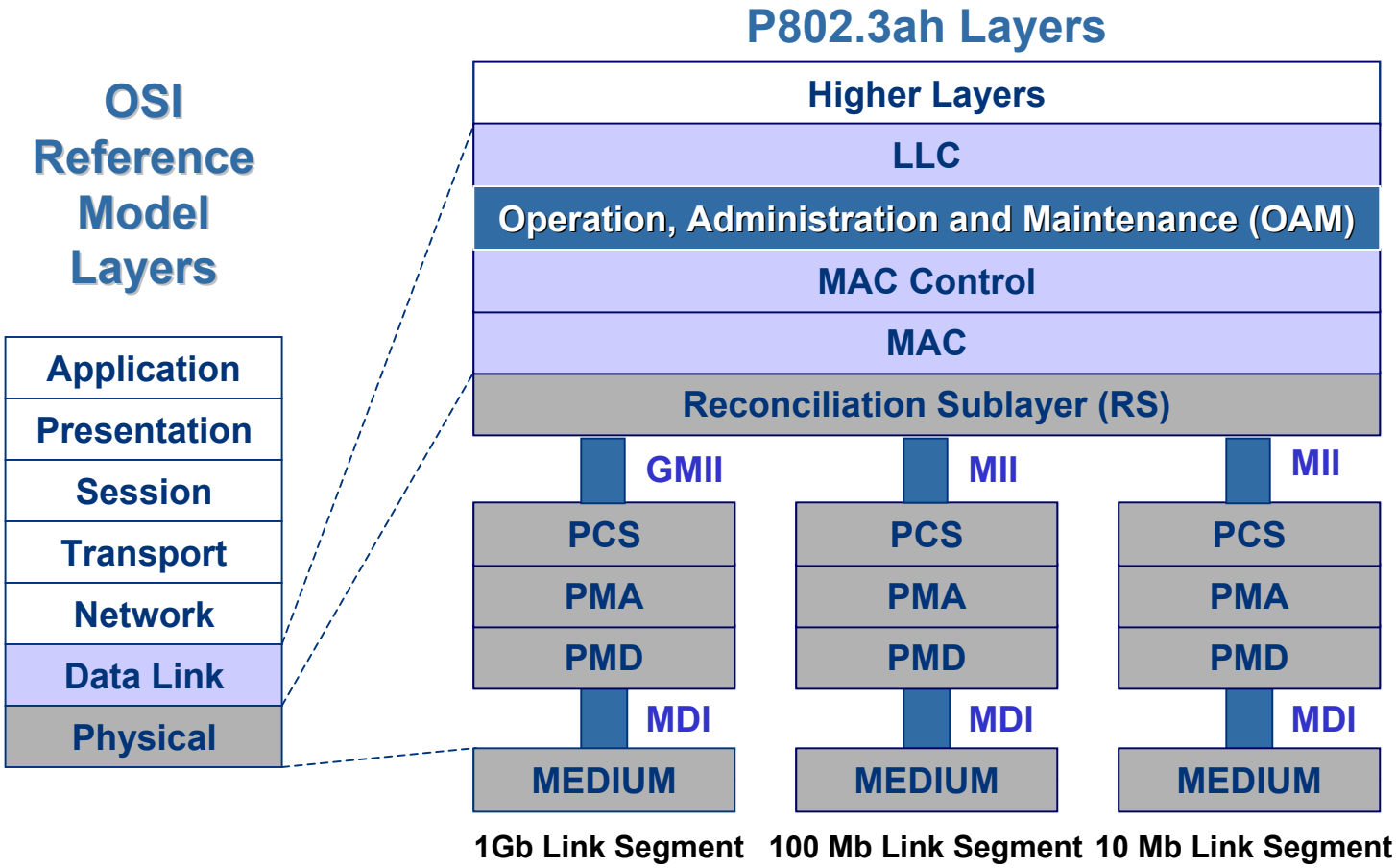
Agenda

- Overview
- OAM Protocol Data Units (OAMPDUs)
- Events
- Variable Retrieval
- Remote Loopback
- Vendor Specific Extensions
- Internal Block Diagram
- Modes

Overview: Parent Organizations

- IEEE 802 LMSC
  - *L*ocal Area Network/*M*etropolitan Area Network *S*tandards *C*ommittee
- IEEE 802.3 CSMA/CD
  - *C*arrier *S*ense *M*ultiple *A*ccess with *C*ollision *D*etect (CSMA/CD) Working Group
    - Commonly referred to as the Ethernet Working Group
- IEEE P802.3ah Ethernet in the First Mile Task Force (EFM)

Overview: OSI Layer Stack



OAM = Operations, Administration & Maintenance
MDI = Medium Dependent Interface
(G)MII = (Gigabit) Media Independent Interface

PCS = Physical Coding Sublayer
PMA = Physical Medium Attachment
PMD = Physical Medium Dependent

Overview: Objectives

- OAM provides mechanisms to:
  - Monitor link operation and health
  - Improve fault isolation
- Method: OAM data conveyed in basic (*untagged*) 802.3 Slow Protocol Frames
  - Sent between two ends of a single link
  - Applicable to all P2P and emulated P2P Ethernet links
  - Slow protocol allows S/W implementation
- Fills major requirement to reduce EFM OpEx

Overview: Non-objectives

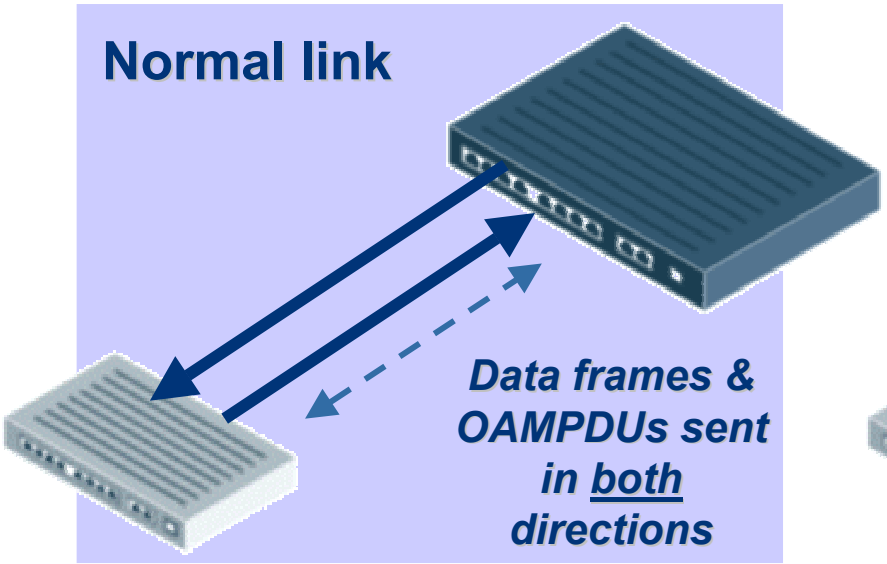
- Does not provide capabilities for:
  - Station management
  - Bandwidth allocation
  - Provisioning
    - No SET functions
  - End-to-end OAM communication

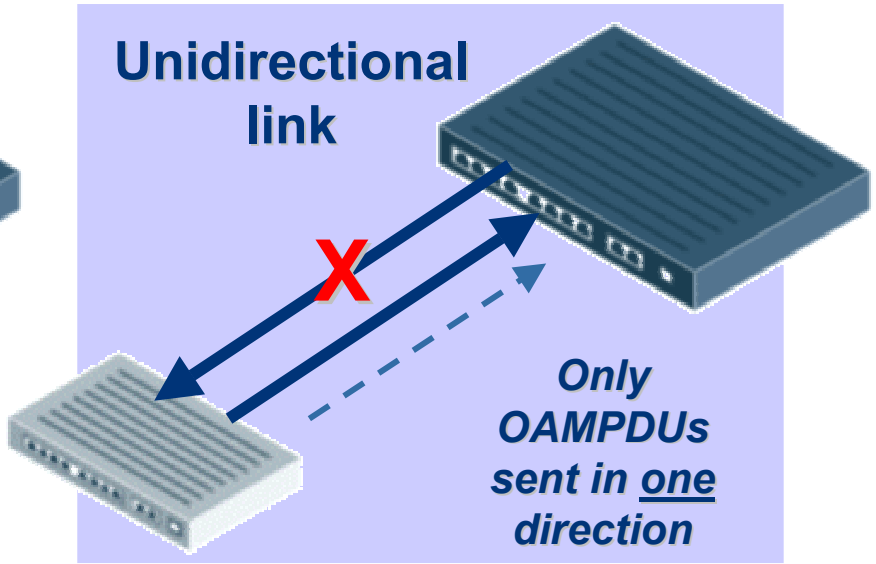
Overview: Compatibility

- OAM is optional and may be implemented in software or hardware
  - Also, may be implemented on one or more ports within a system
- 802.3x MAC Flow Control (PAUSE) inhibits all traffic *including* OAMPDUs
- Support for Unidirectional Operation is mutually exclusive with 802.3z Auto Neg
  - 802.3z Auto Neg must be disabled for OAMPDUs to be sent over unidirectional links
- All P2P and emulated P2P PHYs supported

OAMPDU: Unidirectional

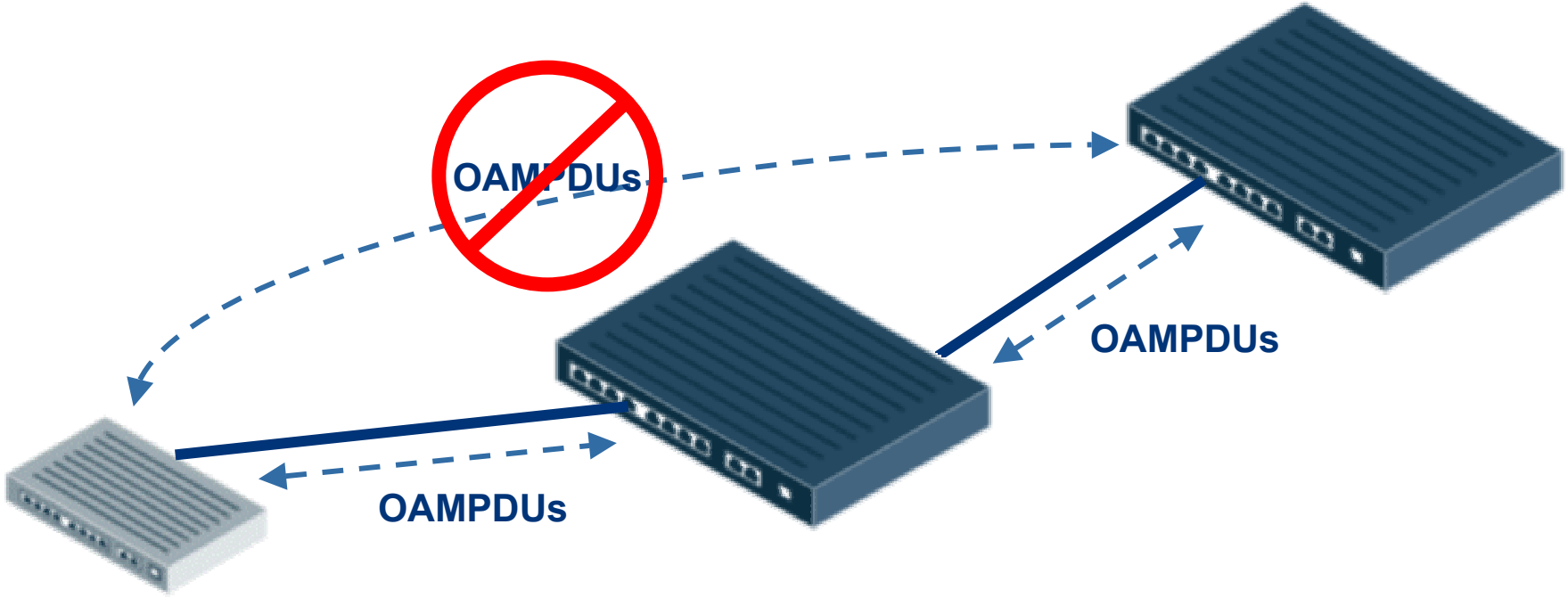
- OAM adding optional PCS feature to allow links to operate unidirectionally
  - *Legacy links become inoperable when one direction fails*
  - Newer links can send OAMPDUs unidirectionally to signal fault information





OAMPDU: Forwarding - NOT

- Only traverse a single link
  - Not forwarded by bridges
- Communication beyond a single link left to higher layers



OAMPDU: Size/Rate

- Must be standard frame length
  - (i.e. 64-1518 octets)
  - Maximum size determined during Discovery process
- Must be untagged

Octets	
6	01-80-c2-00-00-02 [*Slow Protocol*]
6	MAC Source Address
2	Type=88-09 [*Slow Protocols*]
1	Subtype = 0x03 [*OAM*]
2	Flags field
1	Code
42-1496	Data field
4	Frame Check Sequence
64-1518	

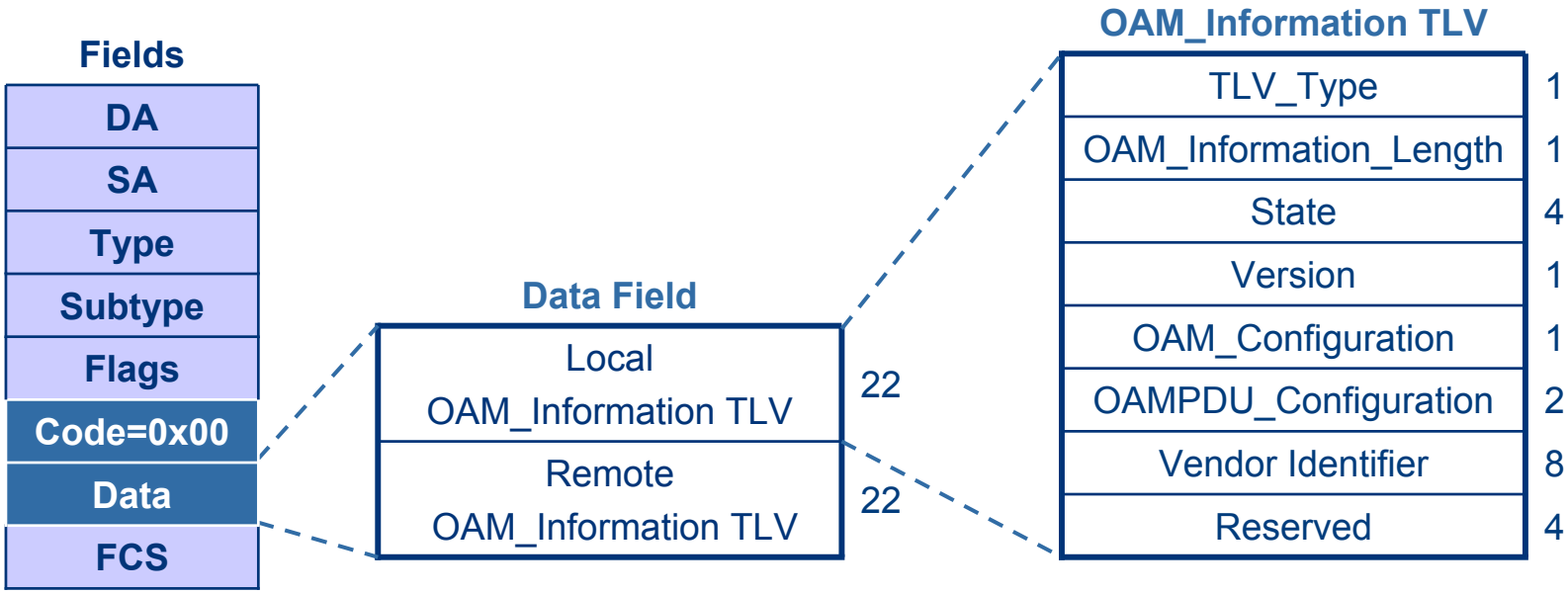
- Maximum of (10) OAMPDUs per second
  - Max rate defined in Annex 43B
  - May be sent multiple times to increase likelihood of reception by remote device (e.g. in the case of high bit errors)

OAM Critical Events

- Link Fault
  - Signal remote device that receive path is broken
- Dying Gasp
  - Signal remote device that unrecoverable local fault (e.g. power failure) has occurred
- Sent in Flags field of every OAMPDU
- May be sent immediately

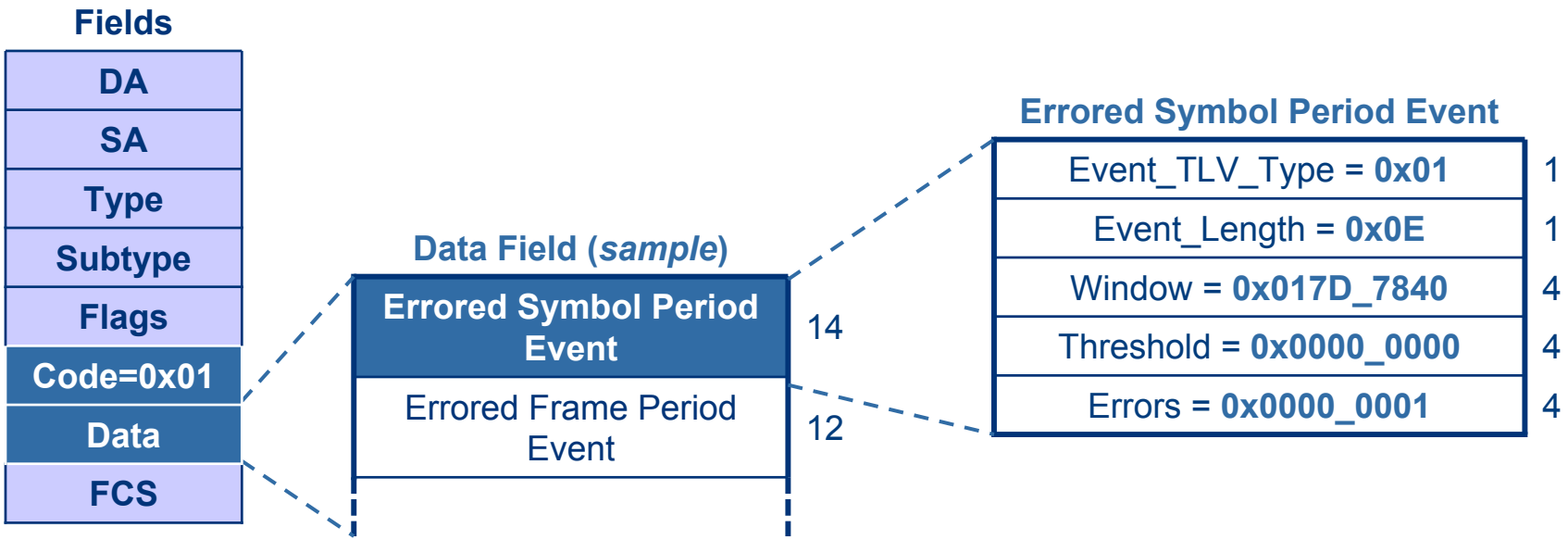
OAMPDU: Information

- Code: 0x00
- Data field: Local and Remote OAM_Information TLVs
- Length: 64 octets



OAMPDU: Event Notification

- Code: 0x01
- Data field: Event TLV(s) → *Descriptions follow*
- Length: *Variable*



OAM Non-critical Events

- Errored Symbol Period Event
- Errored Frame Seconds Event
- Errored Frame Period Event
- Errored Frame Seconds Summary Event

Descriptions follow

- Sent as Event TLVs within Event Notification OAMPDU
- May be sent multiple times to increase likelihood of reception by remote device (e.g. in the case of high bit errors)

Errored Symbol Period Event

- A window, measured in number of symbols, where number of errored symbols exceeded a threshold
- Type: 0x01
- Length: 0x0E (14 octets)
- Value:

Fields	Width	Description
Window	32-bits	Lower bound: Symbols in 1 second Upper bound: Symbols in 60 seconds
Threshold	32-bits	Lower bound: 0 Upper bound: unspecified
Errors	32-bits	# of symbols errors in *Window*

Likely change in D1.4

Errored Frame Seconds Event

- A window, measured in 100ms intervals, where number of errored frames exceeded a threshold
- Type: 0x02
- Length: 0x0C (12 octets)
- Value:

Fields	Width	Description
Window	16-bits	Lower bound: 1 second Upper bound: 60 seconds
Threshold	32-bits	Lower bound: 0 Upper bound: unspecified
Errors	32-bits	# of frame errors in *Window*

Errored Frame Period Event

- A window, measured in frames, where number of errored frames exceeded a threshold
- Type: 0x03
- Length: 0x0E (14 octets)
- Value:

Fields	Width	Description
Window	32-bits	Lower bound: # of 64B frames in 1 second Upper bound: # of 64B frames in 60 seconds
Threshold	32-bits	Lower bound: 0 Upper bound: unspecified
Errors	32-bits	# of frame errors in *Window*

Errored Frame Seconds Summary

- A window, measured in 100ms intervals, where number of errored frame seconds exceeded a threshold
- Type: 0x04
- Length: 0x08 (8 octets)
- Value:

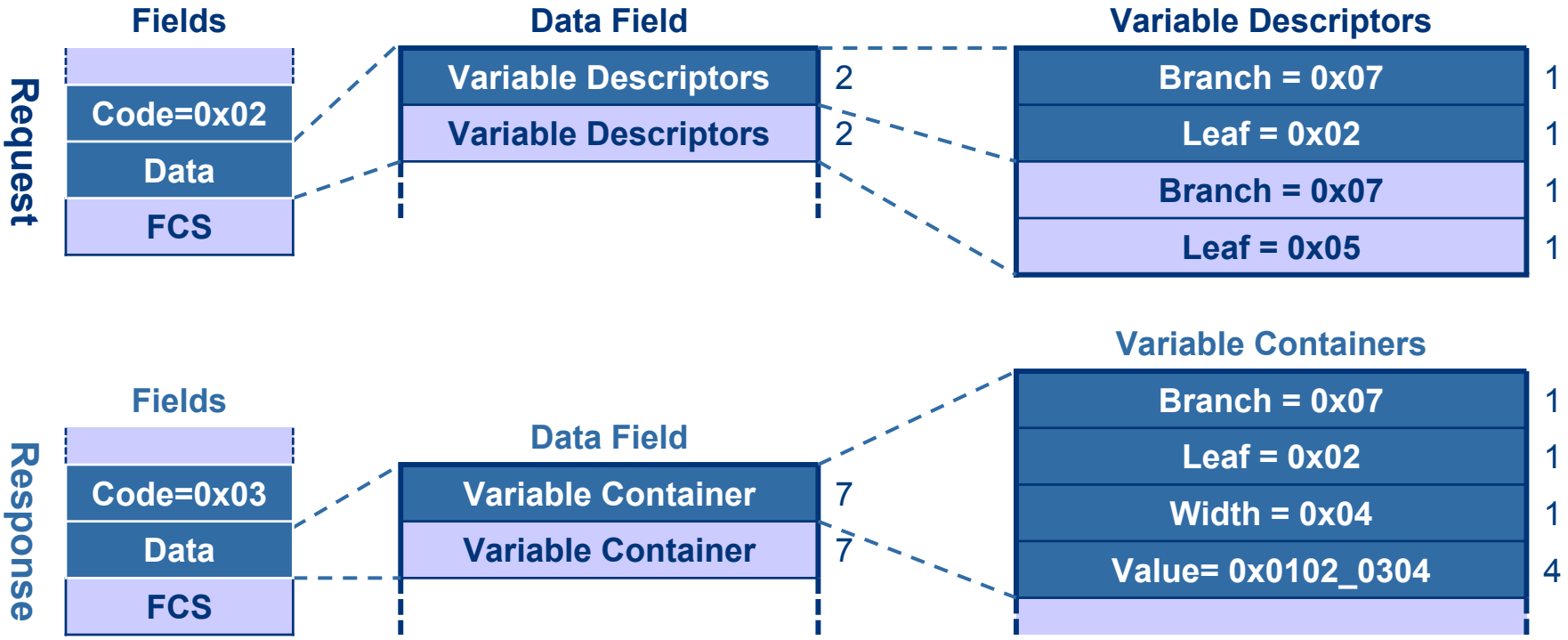
Fields	Width	Description
Window	16-bits	Lower bound: 10 seconds Upper bound: 600 seconds
Threshold	16-bits	Lower bound: 0 Upper bound: unspecified
Errors	16-bits	# of errored frame seconds in *Window*

Event: Extensions

- Vendor Specific Event TLVs
  - Type: 0x80-0xFF
  - Length: Vendor Specific
  - Value: Vendor Specific

OAMPDU: Variable Req/Resp

- **Variable Request**
  - Code: 0x02
  - Data: Variable *Descriptors*
  - Length: *Variable*
- **Variable Response**
  - Code: 0x03
  - Data: Variable *Containers*
  - Length: *Variable*



Variable Retrieval

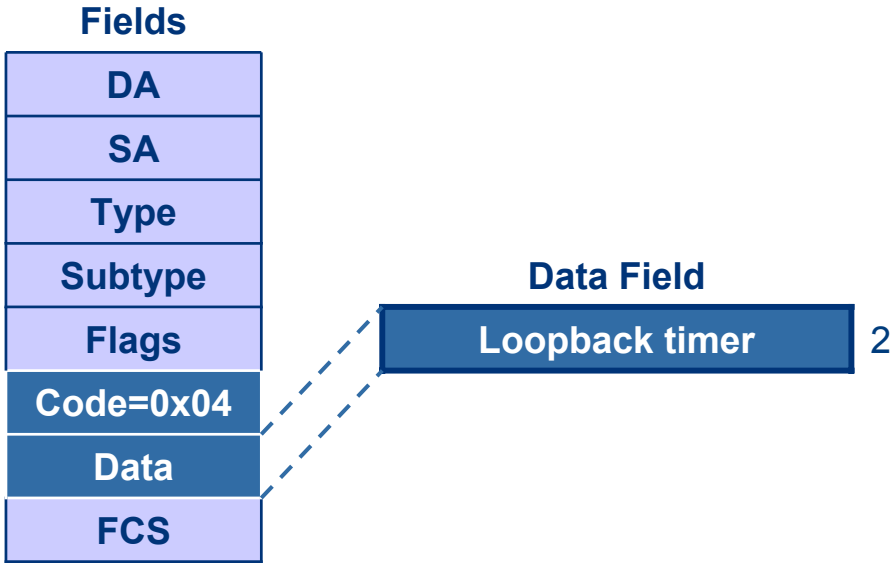
- Transfer Ethernet counters and statistics via Variable Containers/Descriptors
- Variables are referenced using Annex 30A CMIP registration arcs
- Can be used to emulate L2 Ping
  - (i.e. Tx Variable Request, Rx Variable Response)

- Examples:

Variable	CMIP Registration Arcs	
	Branch	Leaf
aFramesTransmittedOK	0x07	0x02
aFrameCheckSequenceErrors	0x07	0x06
aOctetsReceivedOK	0x07	0x0E

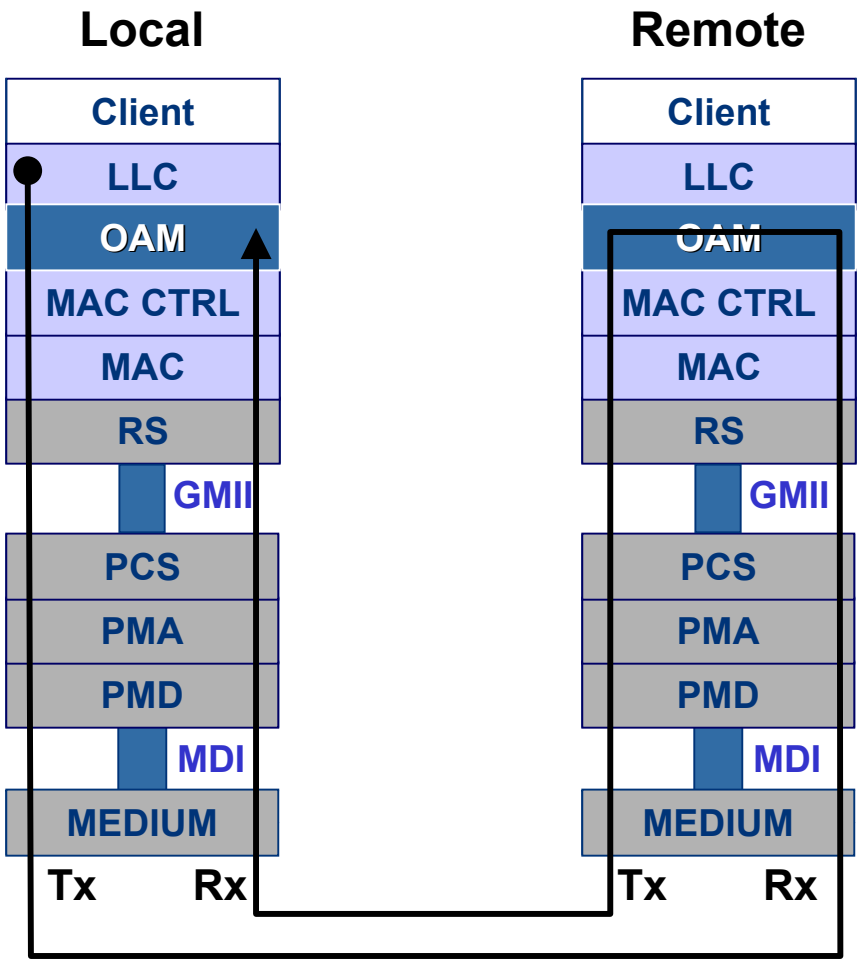
OAMPDU: Loopback Control

- Code: 0x04
- Data field: Loopback timer (Conveyed in 100ms increments)
- Length: 64 octets



Remote Loopback

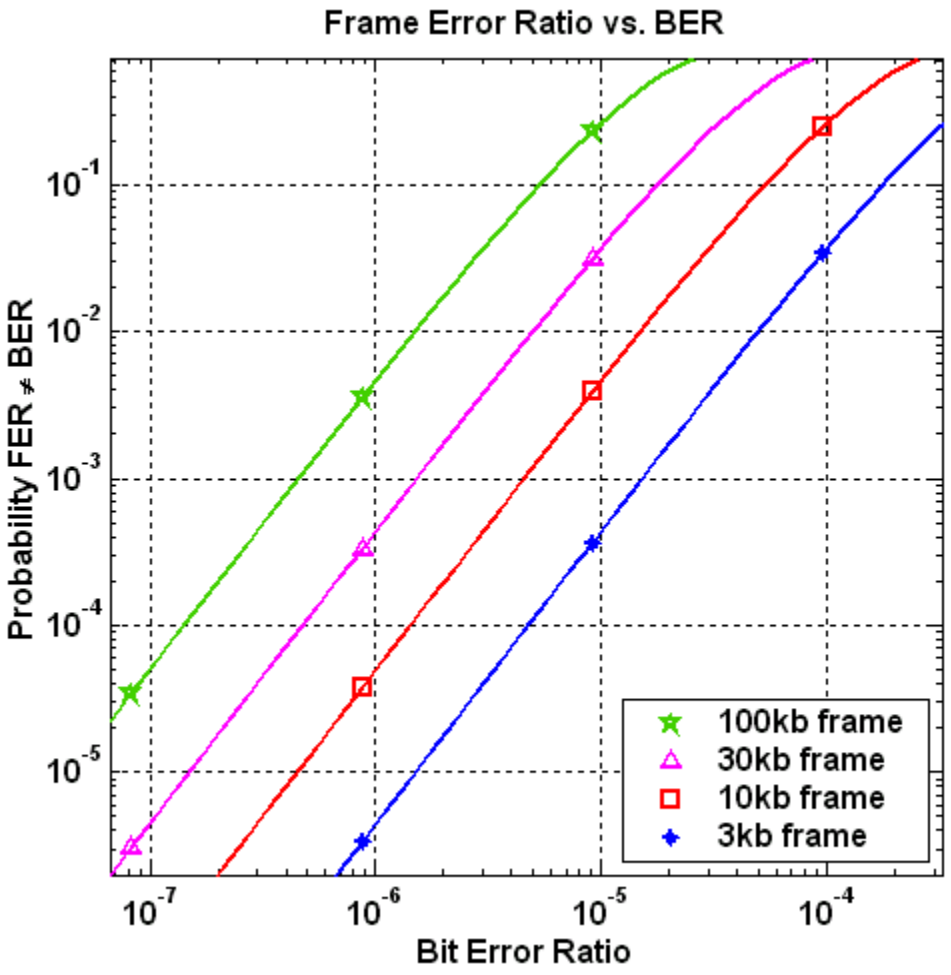
- Local end sends Loopback Control OAMPDU requesting remote end to go into loopback for a prescribed period of time
- Local end sends arbitrary data frames
- Remote end returns data frames
- Frame BER equals bit BER to high probability when bit BER is better than 10^{-6}



Can be implemented in H/W or S/W

Frame Errors vs. Bit Errors

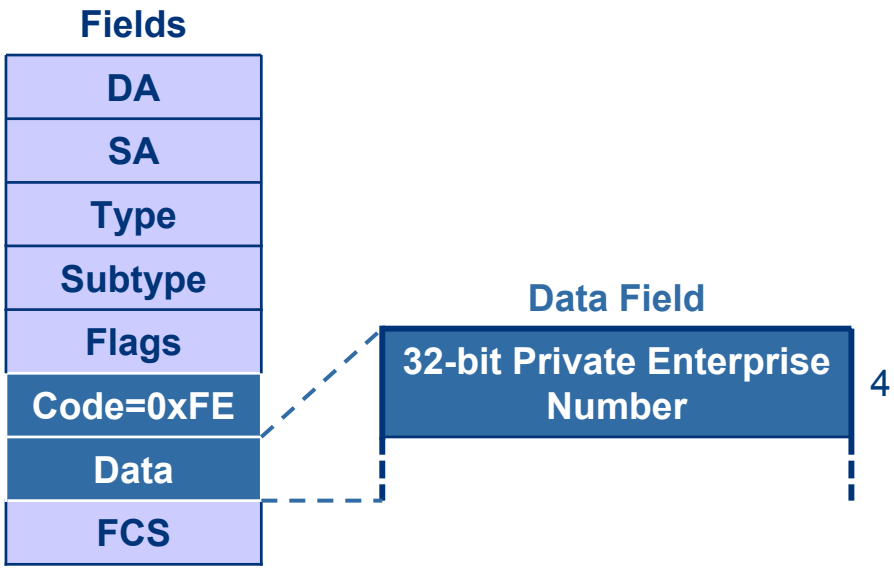
- Assume errors are Poisson distributed in time
  - e.g., system dominated by white, Gaussian noise
  - ignores burst noise
- FER = BER if probability of >1 bit errors over the length of the frame is small
  - depends on BER & frame length
  - depends on acceptable probability for FER ≠ BER
- Sample calculation:
  - 30kb frame
  - acceptable probability ≤ 1%
  - ⇒ BER ≤ 6 x 10^{-6}



Source: John Ewen, JDSU 2002

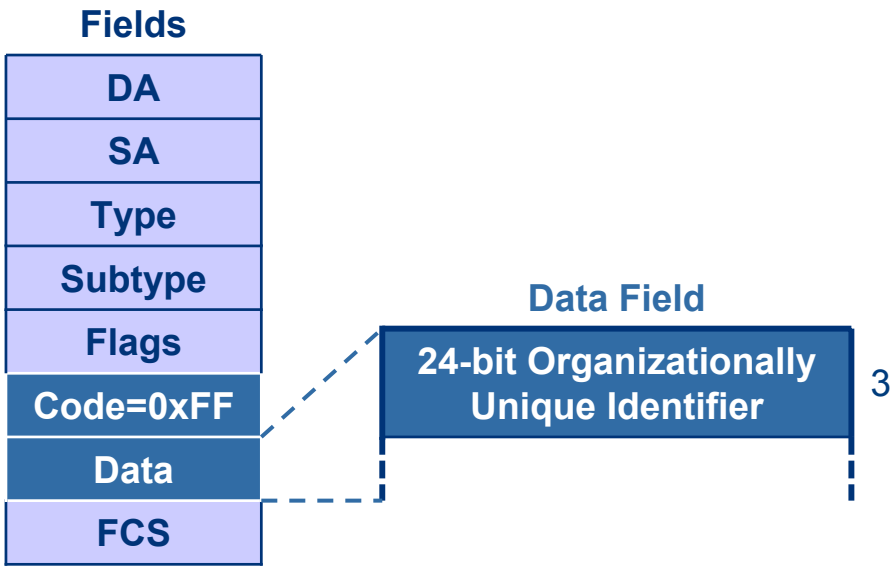
OAMPDU: Vendor Specific IANA

- Code: 0xFE
- Distinguisher: IANA 32-bit Private Enterprise Number
- Data field: Vendor Specific



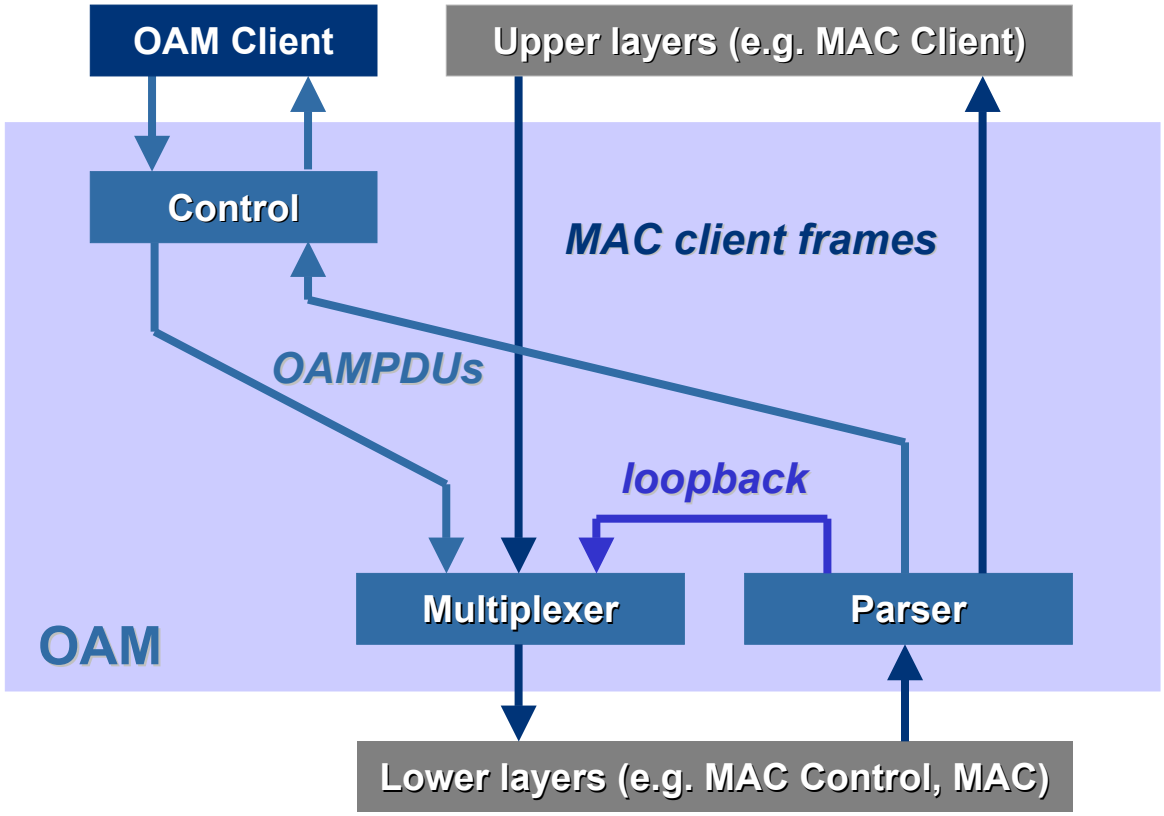
OAMPDU: Vendor Specific OUI

- Code: 0xFF
- Distinguisher: IEEE 24-bit Organizationally Unique Identifier
- Data field: Vendor Specific



OAM Sublayer Block Diagram

- **OAM Client**
  - Configures OAM sublayer
  - Processes received PDUs
  - Transmits PDUs
- **Parser**
  - Inspects received frames
  - In normal mode, sends non-PDUs to upper layer
  - In loopback mode, sends non-PDUs to Multiplexer
  - Sends PDUs to Control block
- **Multiplexer**
  - Multiplexes PDUs and non-PDUs



OAM Modes

- Two modes are defined: Active and Passive
  - A port in Active mode initiatives the OAM Discovery process and is allowed to send any OAMPDU
    - At least one port must be *Active mode*
  - A port in Passive mode waits for the remote device to initiate the Discovery process, responds to requests for Variables

Thank You!