Towards a maintenance channel proposal for P802.3dv

Adee Ran, Cisco

Sridhar Ramesh, Maxlinear

Supporters

- Vasudevan Parthasarathy, Broadcom
- Leon Bruckman, NVIDIA

Introduction

- The study group presentation ran_400GPL_01_2609 introduced the motivation and general concept of having a maintenance channel (MC) alongside mission data in 400 GPL links.
- This presentation provides more details of a possible specification, to enable building consensus towards a baseline proposal.
- Additional presentations planned for this meeting provide more details on proposed uses of the MC.

Applications for maintenance channel

- Transmitter control
  - Transmitter settings configurable in initial training can also be changed in mission mode
  - May vary across link types (electrical, SMF, MMF, coherent)
  - “Small steps” assumed
- Remote CMIS access
  - Query the module remotely
  - Access any defined metric (SNR, FEC statistics…) as telemetry
  - Firmware update
- Time synchronization
  - Send timestamped messages
  - Measure delay close to the wire

MC principles of operation

- A PMA-level function, operating per-ISL [1]
- Repurpose existing alignment markers (AMs)
  - PMA assumed to be AM-aware (802.3dj clause 176)
  - Outgoing: replace original ”unique” AM content with MC payload
  - Incoming: decode MC payload, replace with original AM content
  - No line rate overhead
- Resilient to line errors without FEC protection
  - Message transmitted continuously until acknowledged (similar to ILT frames)
  - Protected by CRC16
  - MC content is scrambled; VCMT header is repetition-coded
- Enables precise timestamping of incoming messages
- Negotiation and startup procedure

[1] May be extended to multi-hop communication, relaying messages across a PMA to the next PMA

MC available bandwidth

- AM group is sent every 105 µs
- A minimum of 30 octets payload per AM group per lane
  - 1.6T PCS: 16 AMs over 4 physical lanes
- Per-PMA-lane theoretical max bandwidth is ~2 Mb/s
  - But per-message acknowledge requirement (“send and wait”) limits it to <1 Mb/s
  - Depending on FW response time, it can be much lower
  - 10-100 kb/s is a reasonable assumption
- Link-level messages can span multiple lanes

AM payloads (example from 800G PCS)

There are 15 octets per AM. The initial 3 octets CM0-CM2 are common, and their inverted versions CM3-CM5 are also common. The remaining 9 octets are unique per lane.

In the 800GBASE-R PCS, “flow 1” (additional 16 lanes) has the same content in the common octets CM0-CM5 and different content in the remaining 9 octets.

400G and 1.6T have a similar structure – 6 fixed octets and 9 unique octets per AM, both with 16 PCS lanes.

The common octets are used initially to obtain AM lock. They are retained to enable detecting loss of lock.

The unique octets are used initially for identifying the PCSL numbers for deskew/alignment. They can be repurposed for a maintenance channel.

Table 172–2—800GBASE-R flow 0 alignment marker encodings

PCS lane number	Encoding[a]														
	CM0	CM1	CM2	UP0	CM3	CM4	CM5	UP1	UM0	UM1	UM2	UP2	UM3	UM4	UM5
0	0x9A	0x4A	0x26	0xB6	0x65	0xB5	0xD9	0xD9	0xFE	0x71	0xF3	0x26	0x01	0x8E	0x0C
1				0x04				0x67	0xA5	0xDE	0x7E	0x98	0x5A	0x21	0x81
2				0x46				0xFE	0xC1	0xF3	0x56	0x01	0x3E	0x0C	0xA9
3				0x5A				0x84	0x79	0x80	0xD0	0x7B	0x86	0x7F	0x2F
4				0xE1				0x19	0xD5	0x51	0xF2	0xE6	0x2A	0xAE	0x0D
5				0xF2				0x4E	0xED	0x4F	0xD1	0xB1	0x12	0xB0	0x2E
6				0x3D				0xEE	0xBD	0x9C	0xA1	0x11	0x42	0x63	0x5E
7				0x22				0x32	0x29	0x76	0x5B	0xCD	0xD6	0x89	0xA4
8				0x60				0x9F	0x1E	0x73	0x75	0x60	0xE1	0x8C	0x8A
9				0x6B				0xA2	0x8E	0xC4	0x3C	0x5D	0x71	0x3B	0xC3
10				0xFA				0x04	0x6A	0xEB	0xD8	0xFB	0x95	0x14	0x27
11				0x6C				0x71	0xDD	0x66	0x38	0x8E	0x22	0x99	0xC7
12				0x18				0x5B	0x5D	0xF6	0x95	0xA4	0xA2	0x09	0x6A
13				0x14				0xCC	0xCE	0x97	0xC3	0x33	0x31	0x68	0x3C
14				0xD0				0xB1	0x35	0xFB	0xA6	0x4E	0xCA	0x04	0x59
15				0xB4				0x56	0x59	0xBA	0x79	0xA9	0xA6	0x45	0x86

[a] Each octet is transmitted LSB to MSB.

AM group in the data path (example – 400G)

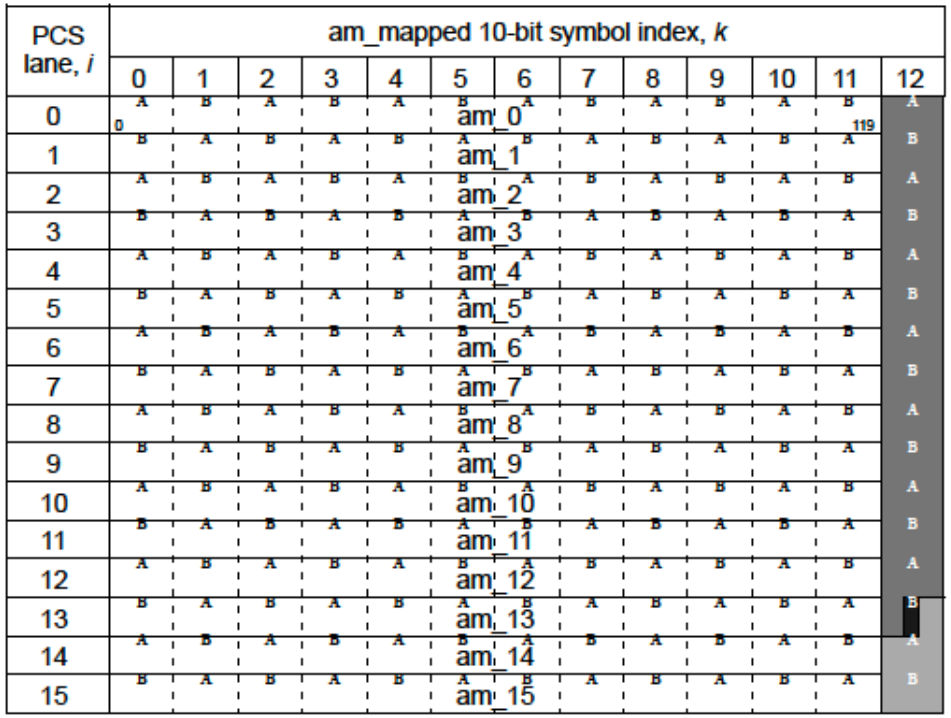


Figure 119–7—400GBASE-R alignment marker mapping to PCS lanes

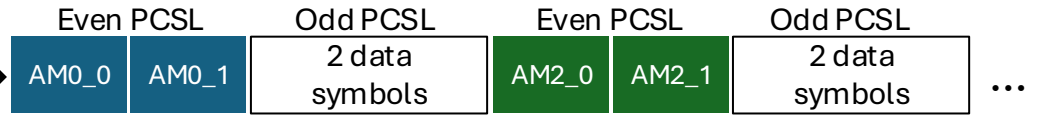
The AMs are sent at the same time on all PCS lanes.

(802.3dj)

An SM-PMA takes a symbol pair from each PCSL. Odd PCSLs are delayed by 136 symbols so their AMs do not appear together with the even lanes.

The result is, on each 200G lane:

16:2 SM-PMA



- A block of 32 symbols, where 16 are from the 8 even AMs and 16 are data
- At an offset of 136 symbols, another block with the odd AMs (not shown)

(802.3dv)

Another mux creates 400G per lane from two 200G per lane by symbol-quartet muxing (after deskewing). The result will be…



- A block of 64 symbols where 32 are from the even AMs and 32 are data
- At an offset of 136 symbols, another block of 64 symbols where 32 are from the odd AMs and 32 are data

AM group locations may vary between PMA lanes, but are fixed per PMA lane.

⇒ The MC functionality is easier to specify/implement on a per-PMA-lane basis.

Proposed message-block format

Octet index	Content	Details	Initial application
2:0	Virtual clock + Message type (VCMT) [1]	1 octet repeated 3 times Bit 7: outgoing message toggle Bit 6: incoming message toggle Bit 5: incoming message acknowledge Bits 4:0: outgoing message type	Type=0: MC internal messages Type=1: Transmitter adjustment Type=2: Remote CMIS request/response Type=3: Time synchronization Type=4-31: reserved (ignore on receipt)
3	Length	1 octet	For Type 0, 1,3 (single-block message) : 0 Can be used to enable multi-block messages (e.g. Type 2)
(9N-3):4	Message Body	9N-6 [2] message octets, interpreted per message type	For type 1: based on Annex 178B E1 format • Coefficient select • Coefficient request • Coefficient select echo • Coefficient status For other types: see other contributions
(9N-1):(9N-2)	CRC16[1]		

[1] See details on next slides

[2] N is the number of PCSLs in the PMA lane: 16 for 400G/800G, 4 for 1.6T

Content of modified AMs across PCS lanes

Octet index (label) in AM / AM index	0-2 (CM0-CM2)	3 (UP0)	4-6 (CM3-CM5)	7-14 (UP1-UM5)
i=0	0x9A, 0x4A, 0x26 (Original AM content)	VCMT1 (octet index 0)	0x65, 0xB5, 0xD9 (Original AM content)	VCMT2 (octet index 1), VCMT3 (octet index 2), length (octet index 3), Message Body (octet index 4-8)
i=1		Message Body (octet index 9)		Message Body (octet index 10-17)
i=2 to N-1		Message Body (continued)		Message Body (continued) + CRC

"Byte index" refers to the table in the previous slide.

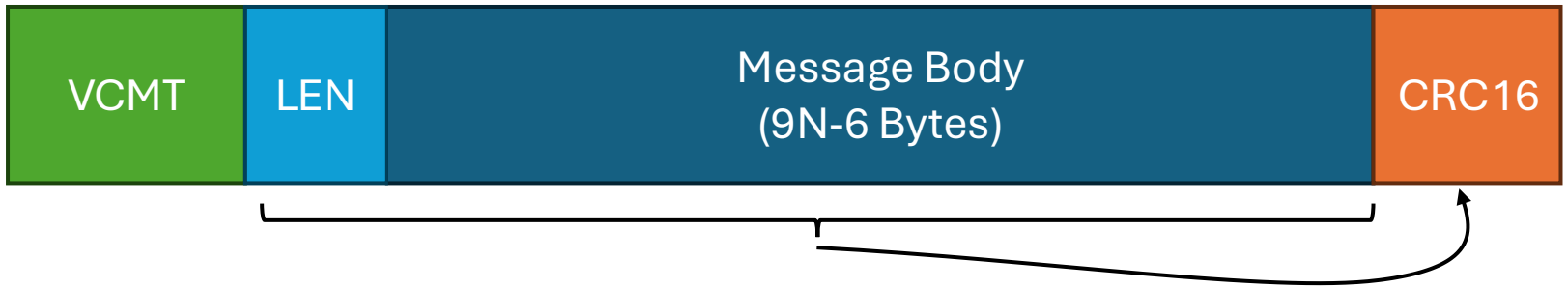
VCMT1, VCMT2, and VCMT3 are three copies of the octet "Virtual clock and message type" (VCMT).

The highlighted areas are scrambled (see next slides); the original AM content octets are not.

AM index is based on the PCSL lane number; or alternatively, on the order of the UP0 octet in the original AM (easier to identify).

N is the number of AMs per physical lane: 4 in 1.6TBASE-R, 16 in 800GBASE-R and 400GBASE-R.

Proposed message-block format (cont'd)



- CRC16 is calculated on the preceding 9N-5 (31 or 139) octets (LEN+Message body), then appended
  - Does not change between repetitions of the same message block
  - Serves as message integrity check, ensuring reliability of MC messages
- Uses the CRC-16-CCITT generator polynomial $x^{16} + x^{12} + x^5 + 1$ (0x1021)
- Receiver can use CRC16 to skip comparing multiple repetitions
  - If CRC matches, no need to wait
  - CRC failure indicates errors in reception (receiver should wait for the next repetition)

VCMT

Virtual clock and message type (VCMT) is an octet that is repeated 3 times.

It is used to handshake message reception and enable precise timestamping. It is repetition-encoded to enable decoding it using a bitwise majority vote.

- Bit 7 is an outgoing “virtual clock”. Toggling it indicates a new outgoing message block.
  - It should be toggled only when the message body/length changes, but not when other bits of the VCMT change
  - Enables “flushing” the receive logic and timestamping the first replica of the new message
- Bit 6 is a VCMT acknowledge. Toggling it indicates the detection of toggled bit 7 in the receiver, before processing the message.
  - Main purpose is to indicate reception of MC messages (regardless of their correctness) to the peer
  - Healthy channel indication, similar to “tf_lock” in ILT
  - A toggle of bit 6 is not considered a new message and does not allow the link partner to send a new message.
- Bit 5 is a message block acknowledge. Toggling it indicates successful receipt of incoming message block and readiness to receive a new message block.
  - A change of bit 5 is not considered a new message.
  - After bit 5 is toggled, the link partner is allowed to send a new message block (toggle bit 7).

The encoding and decoding of the VCMT octets should be done within a single message

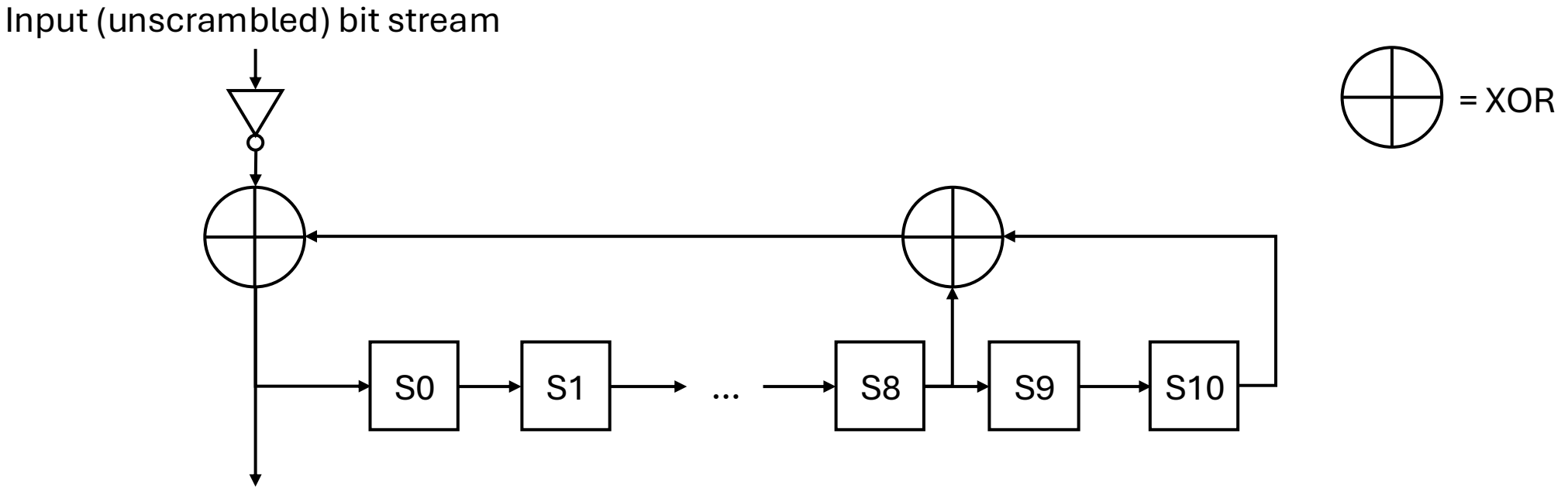
- Preferably in hardware, to enable indicating to software that a new message is received.

Message length and body

- The remainder of the message (length and body) is not repetition-encoded.
- **The same message is sent repeatedly until acknowledged by the receiver (through the VCMT in its outgoing message).**
  - This (along with the CRC16) enables mitigating bit errors
- The repeat+acknowledge mechanism gives control of the message rate to the consumer, enabling firmware handling of the MC messages.

Transmitter side scrambling

- The MC payload (9N octets per message block) is scrambled before transmission by a multiplicative binary scrambler, e.g. using a linear feedback shift register with polynomial $G(x)=1+x^9+x^{11}$ as shown below.
- Scrambling results in different content of each repeated message, preventing "killer packet" concerns. Should likely be implemented in hardware.

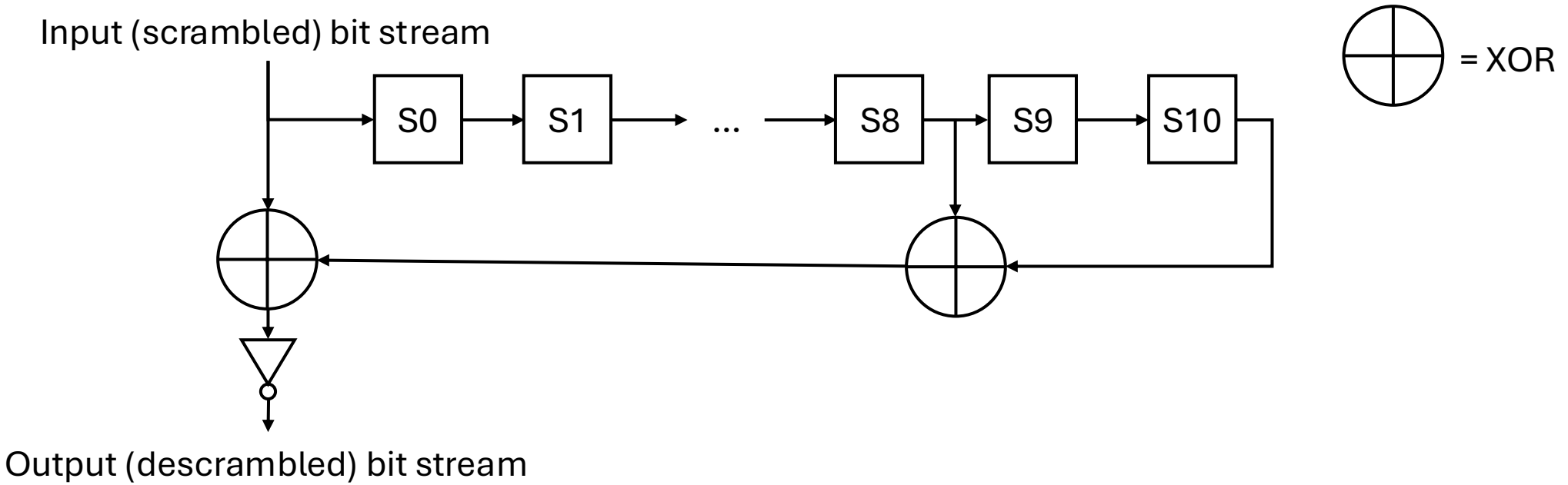


Note: the inverter enables sending all-zero messages

Receiver side descrambling

The MC payload is unscrambled using the corresponding self-synchronizing descrambler, e.g. using a linear feed-forward shift register as shown below.

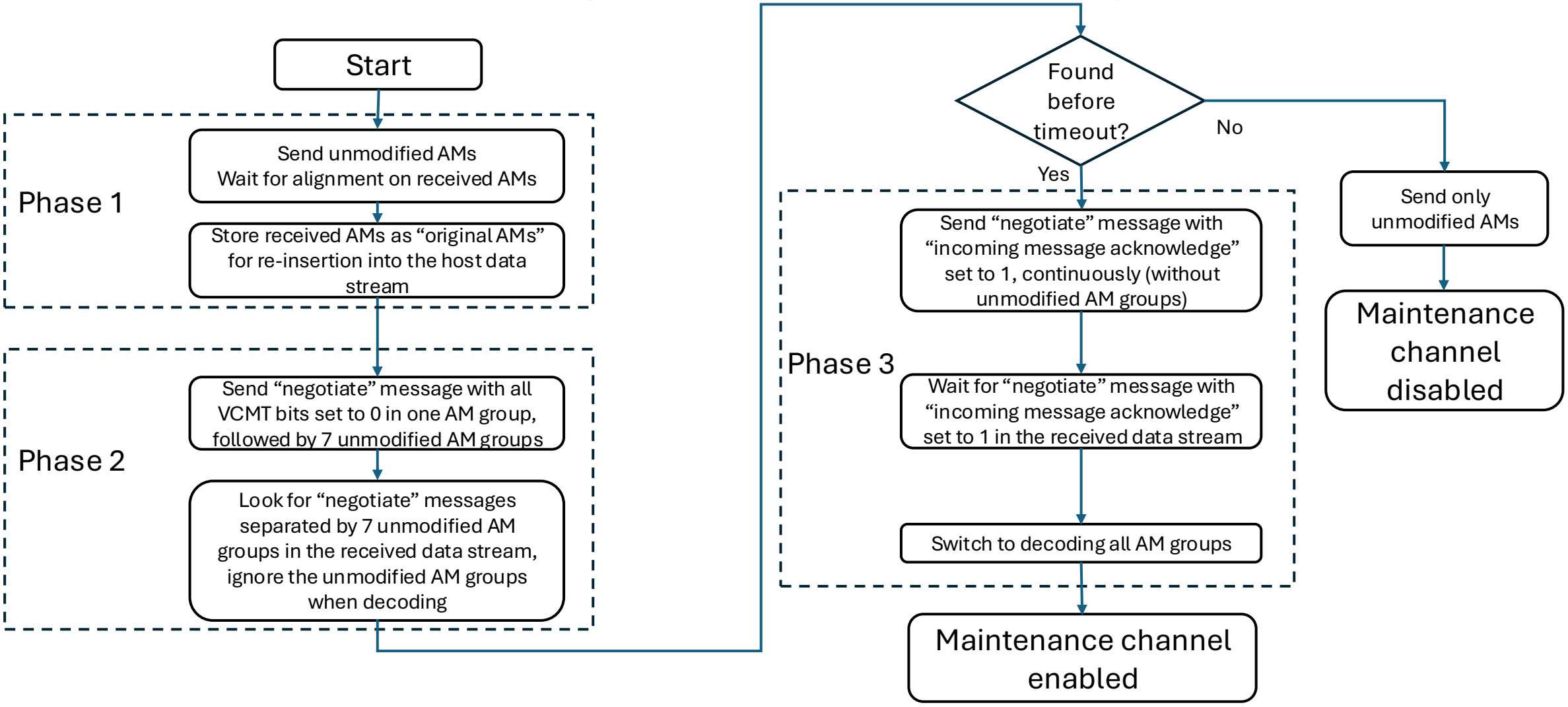
- Again, should be implemented in hardware



MC negotiation

- Usage of the MC requires both PMAs to first lock on the original (unmodified) AMs from each other.
  - Furthermore, the MC might be optional or disabled; its usage should be confirmed.
- Prior to using the MC, each PMA needs to verify that the other PMA can process it (and restore the original AMs such that the modified AMs will not propagate to the next ISL).
- Negotiation is based on the fact that, per the Ethernet PCS specification (IEEE 802.3 Figure 119–12) the process of re-aligning on AMs occurs only after receiving 5 consecutive “bad” AMs on a specific PCS lane. Therefore, a single modified AM every now and then should not trigger this process.
- A possible negotiation method is by sending “NEGOTIATE” messages separated by several unmodified AM groups.

Proposed MC negotiation flow diagram



MC negotiation procedure explained

A PMA that supports the MC performs the following negotiation process.

- Phase 1:
  - Send unmodified AM groups to the peer PMA.
  - Wait for successful alignment and identification of the AMs received from the peer PMA.
  - Store the original content of the received AMs for re-insertion into the data stream to the adjacent interface; proceed to phase 2.
- Phase 2:
  - In the outgoing data stream (to the peer PMA), send a “NEGOTIATE” message with VCMT value 0 (all toggle bits are 0, and message type NULL/IDLE). This message is sent once every 8 AM groups, followed by 7 unmodified AM groups. The scrambler state is updated as if the NEGOTIATE message was sent repeatedly, disregarding the 7 unmodified AM groups.
  - In the incoming data stream (from the peer PMA), look for AM groups that do not contain the original payloads separated by 7 AM groups with the original payloads. If such AM groups are found, feed them into the descrambler and decoding logic and extract the incoming VCMT. If the VCMT is consistently decoded as 0, it is a “NEGOTIATE” message, signaling that the other module is negotiating usage of the maintenance channel; proceed to phase 3.
- Phase 3
  - Change the outgoing VCMT bit 5 (message block acknowledge) to 1, and send the modified AM groups repeatedly (not separated by unmodified AM groups). In the incoming data stream, continue decoding by taking only one of every 8 AM groups into the decoder.
  - Look for an incoming message having 1 in VCMT bit 5 (incoming message acknowledge). When it is found, switch to decoding all AM groups in the incoming data stream; the negotiation process is completed successfully, and the maintenance channel can be used.

MC negotiation procedure explained (cont.)

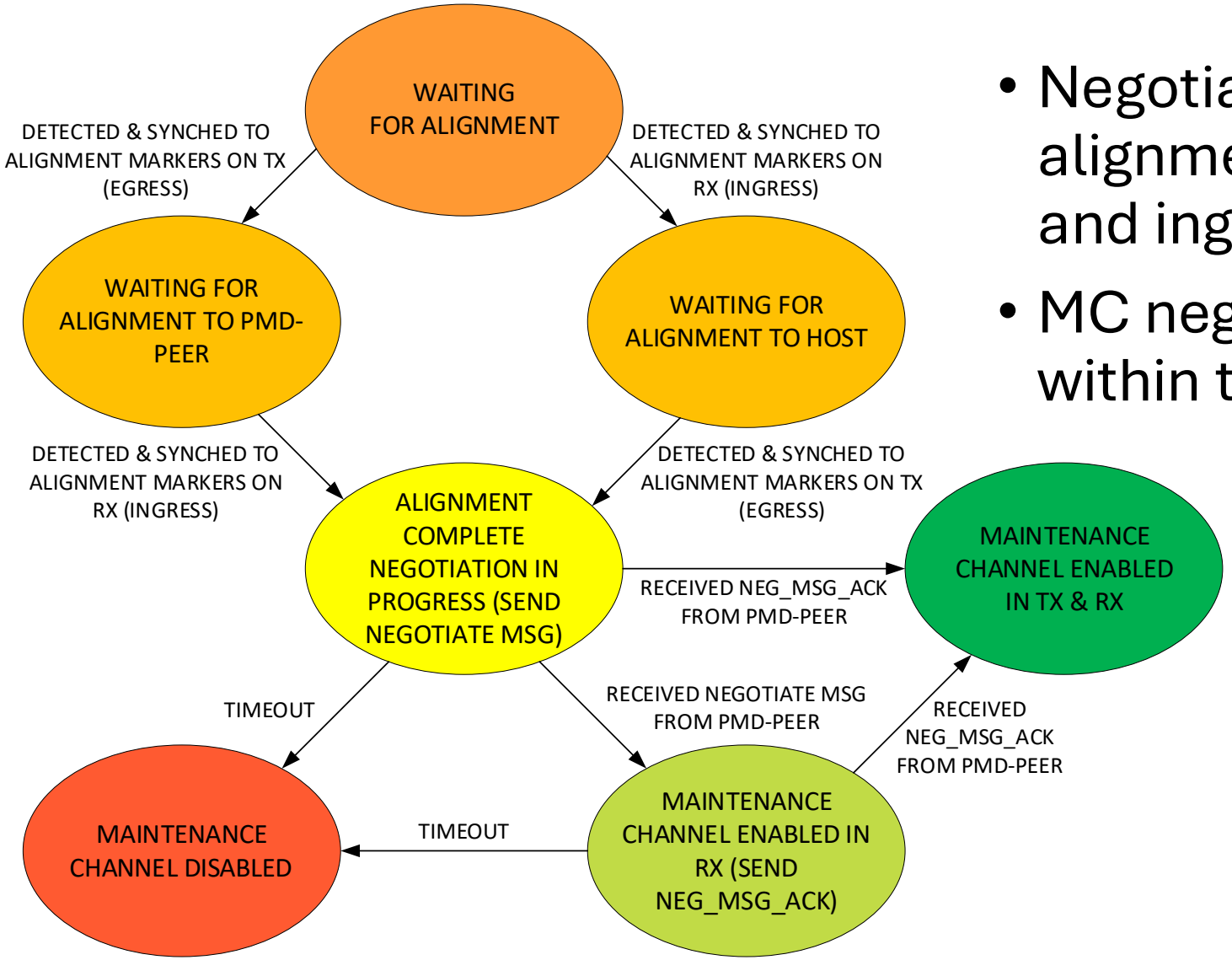
- If the transition from phase 2 to phase 3 does not occur after some time, e.g. 10 seconds, it means that the peer PMA does not support/enable the maintenance channel. In that case, the PMA stops sending the “negotiate” message (reverts to the unmodified AMs) and the MC is disabled.
- A PMA that loses AM lock from its peer (detected by the unmodified AM content) shall keep the MC towards the peer active (the signal might be temporarily disrupted and about to be recovered).
- PMA reset shall disable the MC and restart the negotiation (sending unmodified AM content)
- Detection of incoming unmodified AMs while the MC is active shall disable the MC and restart the negotiation.

Summary

- We presented a detailed proposal for a specification of a maintenance channel (MC) in “mission mode”
  - Connecting two AM-aware PMAs (or other entities) across an ISL
  - Point-to-point and per PMA lane
- Possible use cases include:
  - Transmitter adjustment
  - Time synchronization (separate presentation)
  - Remote CMIS access (separate presentation)
- Message content, scrambling, and negotiation method are proposed
  - May be modified or expanded after initial adoption
- Feedback towards consensus on a baseline proposal is welcome

Backup

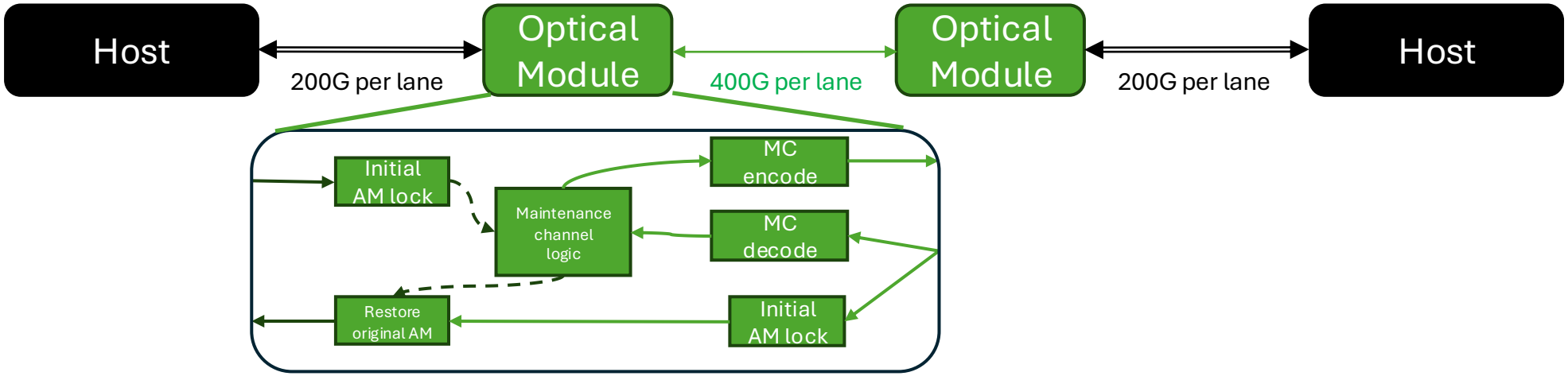
MC negotiation: State diagram view



- Negotiation process initiated after alignment completed in both egress and ingress paths
- MC negotiation to be completed within timeout, else MC is disabled
- Either side can disable MC by not sending or not replying to negotiation messages

Maintenance channel implementation example: communication between two modules

- In the communication between modules, the unique octets of the AMs are repurposed for use as a maintenance channel (exchanging message blocks).
  - Common octets are kept unchanged – to enable detection of loss of AM lock during mission mode.
- The original payloads of the AMs are stored during the initial AM lock.
- The original payloads are restored before sending them to the AUI host interface.
  - The MC is transparent to the host, preserving compatibility with existing hosts.
  - Future specifications may include a similar mechanism for host-to-module communication.



That’s all!

Questions?