Cl **00** *SC* **0** *P* *L* # 3
Grow, Robert RMG Consulting

Comment Type **TR** *Comment Status* **X**

Concur with D2.2 ballot comment #13.

SuggestedRemedy

Per D2.2 ballot comment #13

Proposed Response *Response Status* **O**

Cl **00** *SC* **0** *P* *L* # 2
Grow, Robert RMG Consulting

Comment Type **TR** *Comment Status* **X**

Concur with D2.2 ballot comment #31 first comment paragraph, and recommendation to withdraw or hibernate the project. I also disagree with the rebuttal to that point. There has been insufficient participation from experts in IEEE Std 802.3 to assure specifications are correct, do not break other portions of the standard, and do not unacceptably restrict future PHY options. Participation promised in the PAR has not been met.

SuggestedRemedy

Withdraw or hibernate the project

Proposed Response *Response Status* **O**

Cl **30** *SC* **30.14.1.5** *P* **25** *L* **19** # 7
Scruton, Peter University of New Ham

Comment Type **E** *Comment Status* **X**

Excess period

(unimportant and additionally out of scope, reject at your own leisure should you choose)

SuggestedRemedy

remove the period

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.1** *P* **35** *L* **40** # 8
Scruton, Peter University of New Ham

Comment Type **T** *Comment Status* **X**

Calculation of duration of a 2000 byte frame should include the preamble and SFD (additional 64 bit times). So this would make the delay at 100 be 160.64 and for 1000 would be 16.064 uS.

SuggestedRemedy

"For example, the duration of a 2000 octet packet (including Preamble and SFD) on a 100 Mb/s link is 160.64 us and on a 1 Gb/s link is 16.064 us."

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.1** *P* **35** *L* **40** # 6
Scruton, Peter University of New Ham

Comment Type **T** *Comment Status* **X**

"This is an upperbound on the additional ..."

I don't see this as the upperbound as the upperbound should include the preamble and SFD and the time to transmit the IPG as that would be the extra delay if they were both presented at the same time. Which would be 16000 + 64 + 96 = 16160 or 161.6uS at 100.

SuggestedRemedy

Solution 1:
This, along with the time associated for an IPG, provides an... "

Solution 2:
or combine it with the delay calculation of the previous sentence.

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.3.2** *P* **40** *L* **33** # 11

Scruton, Peter University of New Ham

Comment Type **E** *Comment Status* **X**

Statement:
"0x55 (binary 10101010)."

The binary transmission order is 10101010 as IEEE 802.3-2012 subclause 3.1.1 states transmission order of a byte as LSB to MSB, but 0x55 in binary is not 10101010.

(I recognize this may be out of scope for this comment cycle, but it would appear to be an easy fix.)

SuggestedRemedy

change to something like:
0x55 (which would create a bit order transmission of 10101010, normal preamble).

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.3.6** *P* **42** *L* **23** # 12

Scruton, Peter University of New Ham

Comment Type **ER** *Comment Status* **X**

The text states: " The mCRC shall be calculated on the octets of the frame from the first octet of the frame (i.e., the octet following the SFD sent by the pMAC) to the last octet transmitted in that mPacket by:"

The last octet transmitted in the mPacket would be the last octet of the CRC. I believe the intention is to be the last byte of the MDATA field.

Further the calculation of mCRC is based on starting from the first byte of the frame sent by the pMAC. So it appears that the way it is written this would require the transmitter to remember all octets transmitted in all previous fragments of this frame and generate mCRC based on all transmitted MDATA fields sent for this frame.

(I understand this comment is out of scope, but the remedy should be easily implemented.)

SuggestedRemedy

Adjust text to: "The mCRC shall be calculated on the octets of the current MDATA field by: "

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.4.3** *P* **43** *L* **20** # 13

Scruton, Peter University of New Ham

Comment Type **E** *Comment Status* **X**

missing period

(Out of scope. Easy fix, but reject as out of scope at your liesure should you choose.)

SuggestedRemedy

add period
"Transmission of a verify packet is repeated if no response is received."

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.4.4** *P* **44** *L* **15** # 9

Scruton, Peter University of New Ham

Comment Type **TR** *Comment Status* **X**

According to IEEE 802.3-2012 Subclause 22.2.4.1.8:
"The behavior of the CRS signal is unspecified when the duplex mode bit 0.8 in the control register is set to a logic one, as described in 22.2.4.1.8, or when the Auto-Negotiation process selects a full duplex mode of operation."

This runs counter to the assertion on p44 L15 (99.4.4).
Which is: "In full duplex operation, the PLS_CARRIER.indication primitive is not produced unless EEE (Clause 78) or Link Interruption (46.3.4) is supported."

As a result there may be PHYs that do cause the CRS signal assertion on reception in Full Duplex.

(I recognize this may be out of scope)

SuggestedRemedy

Add text to disallow the use of PHYs that will assert this signal for reasons other than the transmit media is unavailable (EEE or other).

Here in 99.4.4 suggest wording change to:
"The use of preemption is only allowed in full duplex operation, and the PLS_CARRIER.indication primitive shall not be produced while preemption capability is enabled by a PHY conforming to this clause, unless EEE (Clause 78) or Link Interruption (46.3.4) is supported."

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.4.5** *P* **44** *L* **27** # 15

Scruton, Peter University of New Ham

Comment Type **T** *Comment Status* **X**

Ambiguous:
"If an mPacket containing an SMD-S is received when Receive processing was processing an incomplete preempted packet, Receive processing shall ensure that the MAC detects a FrameCheckError in that frame."

Which frame does "that" refer to. In the state diagram this would refer to the previous partial.

(I recognize this as out of scope)

SuggestedRemedy

Change to:

"... MAC detects a FrameCheckError in the partially received frame."

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.4.5** *P* **44** *L* **33** # 14

Scruton, Peter University of New Ham

Comment Type **T** *Comment Status* **X**

"Other techniques may be employed to respond to a received Error control character provided that the result is that the MAC sublayer behaves as though a FrameCheckError occurred in the received frame."

If this is referring to a PCS Coding error this layer should never see an Error control character. At least for 100BASE-TX for an error during frame reception the PCS should see a Code Group Error and flag RX_ER while RX_DV is still asserted and the RS underneath this layer should enforce this by handing something up that would ensure that the MAC would behave as though a FrameCheckError occurred.

(This may be out of scope, but an easy fix)

SuggestedRemedy

Option A:
Strike sentence.

Option B:
As it is talking about enforcing a sequencing order error, we could update sentence:
"Other techniques may be employed to respond to this error provided that the result is that the MAC sublayer behaves as though a FrameCheckError occurred in the received frame."

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.4.7.1** *P* **45** *L* **43** # 16

Scruton, Peter University of New Ham

Comment Type **E** *Comment Status* **X**

"PLS_DATA.requst" should be PLS_DATA.request

(out of scope, but easy fix)

SuggestedRemedy

change "PLS_DATA.requst" to be "PLS_DATA.request"

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.4.7.2** *P* **45** *L* **50** # 4

Scruton, Peter University of New Ham

Comment Type **T** *Comment Status* **X**

I understand what is intended, but I'm not a fan of the current way it is written as this is saying binary and then gives a hex.

"The binary value 0x55"
"The binary value 0xD5"

Also I'm not sure the order of the filling of the vectors for pTX_DATA and rRX_DATA.

(I recognize this as out of scope)

SuggestedRemedy

Change these to be 8-bit vector data <7:0> or <0:7> values. I can't seem to tell if this should be <0:7> or <7:0> it looks like rTX_DATA and pRX_DATA flips it so I think it would go in as <0:7> and then it will get flipped? I know the way it should be transmitted going down the stack it would go 1 then 0101010 and SFD as 10101011...

So the remedy if rTX_DATA flips would be along the line of:

the 8-bit vector <0:7> of 0x55
the 8-bit vector <0:7> of 0xD5

I would also suggest clarifying the bit ordering in rTX_DATA and pRX_DATA in the functions on 99.4.7.4.

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.4.7.3** *P* **46** *L* **45** # 1

Hajduczenia, Marek Bright House Network

Comment Type **TR** *Comment Status* **X**

A new variable pAllow was added in D2.4, which is set then to TRUE in Figure 99–5 in state SEND_SMD_S.
However, this variable is never assigned a default value, and it is never reset explicitly or implicitly to FALSE.

SuggestedRemedy

Given that there is no clear description of when this variable would be reset to FALSE, I'd suggest what follows:
- in Figure 99–5, set pAllow to FALSE in INIT_TX_PROC state
- in Figure 99–5, add "pAllow <== FALSE" assignment in TX_MCRC state (initial fragment has been sent by then, no need to indicate the intention to sent preemptable frame anymore)
Alternatively, default value could be added to the definition of variable, if needed.

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.4.7.7** *P* **50** *L* **1** # 10

Scruton, Peter University of New Ham

Comment Type **TR** *Comment Status* **X**

Figure 99-5 Transmit Processing State Diagram

As PLS_Carrier.indication could be produced in EEE or Link Interruption perhaps it may be advisable to have an additional entrance condition in the START_PREAMBLE, and the transition from RESUME_WAIT into RESUME_PREAMBLE to also be And-ed with PLS_Carrier.indication=CARRIER_OFF.

This way if a Preemptable packet arrives while the media is unavailable the decision as to whether to send this frame will not be made until after the media is available. This way if the media is unavailable the (an Express Frame may be available at that time).

Related to this if EEE is allowed (looks to currently be the case) then LP_IDLE.request shall not be set to ASSERT when frames need to be transmitted and also 802.3-2012 subclause 22.7.2:
"The operation of LPI in the PHY requires that the MAC does not send valid data for a time after LPI has been de-asserted as governed by resolved Transmit Tw_sys defined in 78.4.2.3.

This wake up time is enforced by the transmit LPI state diagram and the rules mapping CARRIER_SENSE.indication defined in 22.2.1.3. The implementation shall conform to the behavior described by the transmit LPI state diagram shown in Figure 22-23."

SuggestedRemedy

Solution A:
Specifically allow EEE:
Add signal LP_IDLE.request into Figures 99-2 and 99-3.
Add necessary states and transitions to Figure 99-5 to accomplish:
- Allow asserting LP_IDLE.request, but when traffic is to be sent deassert and timeout before transmit.

Solution B:
A statement requiring that if EEE is enabled ensure that LP_IDLE.request remains Deasserted.

Proposed Response *Response Status* **O**

Cl **99** *SC* **99.4.7.7** *P* **52** *L* **1** # 5

Scruton, Peter University of New Ham

Comment Type **T** *Comment Status* **X**

Figure 99-7

States: RCV_V and RCV_R
These both have pRX_DV(False) calls. This looks to be done with the intention that if there is a V or an R saying that any continuation of a preempted frame would be wrong. I don't think the R would imply that, as 99.4 would seem to indicate that it should always be ready to accept.

It is strange that these have the affect of altering the states typically used in figure 99-6.

If the intention is to discard in this case it could be done with an additional state transition in Figure 99-6.

SuggestedRemedy

Remove those pRX_DV(False) calls in Figure 99-7 states RCV_V and RCV_R.

If it is desired to discard when the remote side does V add a transition in Figure 99-6 from CHECK_FOR_RESUME to ASSEMBLY_ERROR on condition V (because entering ASSEMBLY_ERROR increments a statistic see 30.14.1.8 counter for Assembly errors it may or may not be desireable to count this as an assembly error, if not then this may be a new state with the DISCARD function inside and then a transition on !rRxDV to IDLE_RX_PROC).

Proposed Response *Response Status* **O**