Item *Cl* **00** *SC* *P* *L* **#** **18**

Geoff Thompson

Comment Type **TR** *Comment Status* **A**

I don't think the response on Discard Trash answers the question:
Is this 15 bits corresponds to Preamble bits??
It is about 15 of the first of the preamble bits How MAC will interprete about this discarding.
Won't bother it a bit, it isn't counting.
Please clarify why it is so?
The PLS throws away some of the bits to allow the analog receiver circuits to stabilize so that only well formed bits are presented to the MAC.

SuggestedRemedy

Rework during Plenary week.

Response *Response Status* **W**

ACCEPT IN PRINCIPLE.

See comment 12.

Item *Cl* **00** *SC* *P* **1** *L* **34** **#** **13**

Hugh Barrass

Comment Type **T** *Comment Status* **A**

In Section 1. DATA_VALID_STATUS, the last sentence is incorrect.

SuggestedRemedy

Change sentence:

This text provides the description of where DATA_NOT_VALID takes the DATA_VALID value.

to:

This text provides the description of where DATA_VALID_STATUS takes the DATA_VALID value.

Response *Response Status* **C**

ACCEPT.

Item *Cl* **00** *SC* *P* **1** *L* **38** **#** **12**

Hugh Barrass

Comment Type **T** *Comment Status* **A**

In Section 2. DISCARD_TRASH, the response is incomplete and will (probably) not satisfy.

SuggestedRemedy

Add new text before the first sentence:

The standard states in subclause 3.2.1 Preamble field... etc.

add:

The first 15 data bits received after CARRIER_ON are discarded in order to protect the DTE device from the effect of unreliable symbols immediately following the detection of the carrier. In normal operation these 15 bits will be part of the preamble which may be ignored and discarded.

Response *Response Status* **C**

ACCEPT.

Item **1** *Cl* **00** *SC* *P* **1** *L* **#** **17**

Lee Sendelbach

Comment Type **T** *Comment Status* **R**

The inquiry was right that the comment about "discarding 15 bits" is confusing in the DISCARD TRASH portion of figure 7-6. I can't find any text which even refers to figure 7-6 in the section. I agree that your text clarifies the situation well, but figure 7-6 is confusing. I would appreciate some clarification on the 15 bits.

SuggestedRemedy

Can we admit in the interpretation request that the DISCARD TRASH box in figure 7-6 is confusing and we will fix it in a future maintenance release?

Response *Response Status* **C**

REJECT.

While it is obscure, it is not incorrect. In addition see response to comment 12.

Item **1** *Cl* **00** *SC* *P* **1** *L* **#** **16**

Benjamin Brown

Comment Type **E** *Comment Status* **A**

You used the wrong word in the last sentence of the DATA_VALID_STATUS interpretation for 10Mb/s Carrier Detect

SuggestedRemedy

In this last sentence, replace "DATA_NOT_VALID" with "DATA_VALID_STATUS"

Response *Response Status* **C**

ACCEPT.

Item **1** *Cl* **00** *SC* *P* **1** *L* **#** **15**

Benjamin Brown

Comment Type **T** *Comment Status* **R**

I don't think you've answered the question.

SuggestedRemedy

Add in the response that the assignment

DATA_VALID_STATUS <= DATA_VALID

should be in the INITIALIZE state.

Response *Response Status* **C**

REJECT.

In the INITIALIZE state DATA_VALID_STATUS should to be set to DATA_NOT_VALID which it is. You would not want to power up RXing data.

Item **1** *Cl* **00** *SC* *P* **1** *L* **#** **14**

Brad Booth

Comment Type **E** *Comment Status* **A**

The last sentence to response #1 incorrectly states the variable name.

SuggestedRemedy

Change "DATA_NOT_VALID" to be "DATA_VALID_STATUS".

Response *Response Status* **C**

ACCEPT.

Item **1** *Cl* **00** *SC* *P* **1** *L* **#** **20**

Wael William Diab

Comment Type **T** *Comment Status* **R**

Discard Trash "Discard the first 15 bits received" the description given by the interpretation seems ok, but please make sure that it is clear that after receiveDataValid is asserted the MACs first look for 55 and the D5 before starting a frame. If the fits data is neither 55 nor D5 then some (specially older one's) do not start the frame and this results in frame drop.

SuggestedRemedy

see comment.

Response *Response Status* **C**

REJECT.

What you described may be in some implementations but is not what the standard states. The Pascal in Clause 4 requires that after the assertion of receiveDataValid the MAC await a valid SFD in procedure physicalSignalDecap.

Item **1** *Cl* **00** *SC* **Figure 7-6** *P* **1** *L* **#** **19**

Wael William Diab

Comment Type **T** *Comment Status* **R**

State-machines precedes text.

SuggestedRemedy

Although there is a description in section 7.2.2.1.6 it is prefered not go by this interpretation and just have thie state-machine, figure 7-6, description.

Response *Response Status* **C**

REJECT.

State Diagrams only override the text in the case of conflict. In this case there is no conflict and both text and diagram are necessary for a complete description.

Item **2** *Cl* **00** *SC* *P* **2** *L* **#** **21**

Wael William Diab

Comment Type **T** *Comment Status* **R**

The interpretation and the reference are appropriate for allowing entry into CARRIER DETECT state. However, the interpretation would also permit (carrier_status == OFF) * (r_bits[0] = 0) * (r_bits[9:2] = 00000000) as a valid entry into CARRIER DETECT. Care should be taken here.

SuggestedRemedy

CARRIER DETECT state entry condition of (carrier_status == OFF) * (r_bits[0] = 0) * (r_bits[9:2] != 11111111) and the interpretation (referencing 24.3.4.3 which states "The Carrier Detect process monitors the r_bits vector until it detects two non_contigous ZEROS in the incoming code-bit sequence.") contradicts the "it detects two non_contiguous ZEROS in the incoming code-bit" text of section, because as you see it has no two non_contiguous ZEROS and all ZEROS are contiguous and still allows entry.

Thus it is recommended that the state-machine takes precedence over text

Response *Response Status* **C**

REJECT.

As stated in the interpretation response r_bits[9:0] is a sliding window of the receive code bits with the newest bit placed in r_bit[0]. After every shift r_bits[9:0] is examined to see if it meets the condition r_bits[9:2] != 11111111 and r_bit[0] = 0. It would therefore be impossible to get to the condition where r_bits[9:2] = 00000000 without have gone through a previous condition that meets the condition r_bits[9:2] != 11111111 and r_bit[0] = 0. This would occur at the point where the stream of all zeros had been present for three bits and r_bits[9:0] = 1111111000, that is r_bits[9:2] = 11111110 and r_bit[0] = 0.