P802.1Qdt Text Contribution

Lily Lv (Huawei)
May-2022

Comments:

1. 6.7.3 When matching M_CONTROL.request to M_UNITDATA.request, is it ok to set Service_access_point_identifier and connection_identifier to be NULL? How does CB consider those parameters?
2. Reference plane should be considered and aligned in 802.1 Q
3. Does field ‘MBC’ in PFC configuration TLV format need update?
4. Position of field ‘HDRCap’ in PFC configuration TLV is not correct.

Contents

<<This amendment is based on IEEE Std 802.1Q-Rev-d1-02>>

1. Overview

1.3 Introduction

Insert the following items after item bh)

bi) Defines a means for two participating systems to automatically calculate the minimum buffer requirements to assure lossless operation.

bk) Defines a means for MACsec protection of PFC MAC control frames

2. Normative references

Insert the following reference in the appropriate collating sequence:

IEEE Std 1588: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

5. Conformance

5.4.1.7 DCBX Bridge requirements

Insert the following items after item h)

h) Support automatic PFC buffer requirement configuration (36.3)

5.11 System requirements for Priority-based Flow Control (PFC)

Insert the following items after item h)

i) Support automatic configuration of PFC buffer requirements for lossless operation.

6. Support of the MAC Service

Insert new subclause 6.7.3

6.7.3 Support of MACsec protection on PFC frames

PFC functionality generates and processes MAC control primitives. MACsec functionality generates and processes MAC service primitives. In order to protect PFC frames with MACsec, it is necessary to provide a shim layer that converts the PFC MAC control primitives to MAC service primitives.

As shown in figure aaa, after converting the control primitives from the upper layer PFC function to corresponding MAC service primitives, the MAC service primitives are sent to the MACsec function for encryption. Upon reception of an encrypted PFC frame, the reverse is performed. The decrypted MAC service primitives are converted to MAC control primitives and submit to upper layer PFC function.

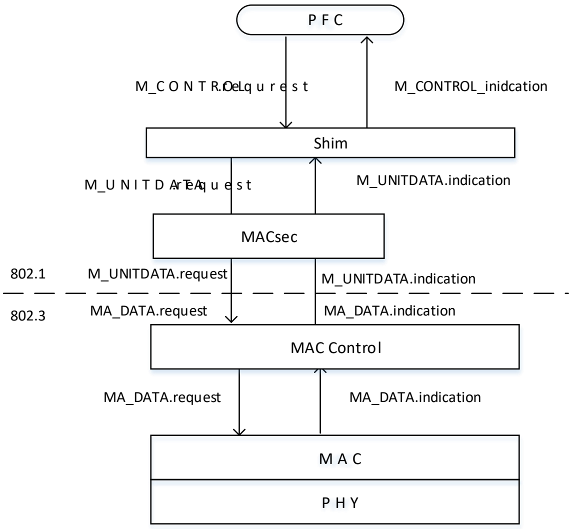


Figure aaa-- MACsec enable PFC frame transmission

When the shim function receives an M_CONTROL.request primitive and MACsec is enabled, the shim function generates a corresponding M_UNITDATA.request to the underlying MAC service. The parameters of M_CONROL.request primitive are destination_address, opcode, and request_operand_list. The parameters of M_UNITDATA.request primitive are destination_address, source_address, mac_service_data_unit, priority, drop_eligible, frame_check_sequence, service_access_point_identifier, connection_identifier.

— The destination_address parameter is passed unaltered

— The opcode (2 octets) and request_operand_list (18 octets) are combined as mac_service_data_unit

—Generate parameters of source_address, frame_check_sequence priority, drop_eligible, service_access_point_identifier, and connection_identifier.

- Priority parameter is set to 7, to give the MAC control frame highest priority for processing.
- Drop_eligible parameter is set to FAULSE.
- Service_access_point_identifier and connection_identifier paramters are set to NULL.

(

Comment:

Is it ok to set service_access_point_identifier and connection_identifier to be NULL? How does CB consider those parameters?

Response: Those parameters are OK to be NULL.

- service_access_point_identifier and connection_identifier are used locally.

In 802.1AC(2016) 11.1 Service primitives and parameters, it says “Unlike the other parameters of the service primitives, the service_access_point_identifier and connection_identifier are not parameters of the peer-to-peer service. The values of the service_access_point_identifier and connection_identifier are purely **local to the system** within which a given service request or service indication occurs, and are not conveyed to the communicating peer system.”

- The shim and .1CB do not interfere with each other.

802.1CB functions are located on top of MACsec, working in parallel with shim layer.

.1CB does not handle service_access_point_identifier, it only uses connection_identifer to carry stream identification information as a subparameter. Stream identification information is generated by specific function modules like stream identification function and sequencing function. Such information could be converted to explicit field in packet when transmitting to remote system.

- PFC is a peer-to-peer service with fixed DMAC. PFC M_CONTROL.request does not include service_access_point_identifier and connection_identifier information.
- Additionally, 802.11 parameter mapping shows us an example of setting the parameters to be NULL.

“13.2.4.2 IEEE 802.11 parameter mapping (indications)

When an IEEE 802.11 MA-UNITDATA.indication or DS-UNITDATA.indication primitive is received, the IEEE 802.11 convergence function (13.2.1 or 13.2.3) generates a corresponding M_UNITDATA.indication as follows:

……

d) The frame_check_sequence, service_access_point_identifier, and connection_identifier parameters are null.

…… ”

- But do we need to consider adding the 2 parameters to enhance flow control?

If the service_access_point_identifier and/or the connection_identifier can be used to distinguish the hidden-in-data-plane PFC control in addition to the priority parameter only, and thus to indicate to the MAC layer that an expedited treatment is required.

)

When the shim function receives an M_UNITDATA.indication primitive from the underlying MAC service, it reads the destination_address parameter. If destination_address parameter is 01-80-C2-00-00-01 which indicates it is a MAC control frame, the shim converts M_UNITDATA.indication to a corresponding M_CONTROL.indication. The parameters of M_UNITDATA.indication are destination_address, source_address, mac_service_data_unit, priority, drop_eligible, frame_check_sequence, service_access_point_identifier, connection_identifier. The parameters of M_CONROL.indication primitive are opcode, and indication_operand_list.

— The parameters of destination_address, source_address, priority, drop_eligible, frame_check_sequence, service_access_point_identifier, connection_identifier are dropped.

— The mac_service_data_unit parmeter is parsed to extract opcode parameter and indication_operand_list parameter.

- The first 2 octects is opcode parameter.
- The following 18 octets is operand_list parameter.

The shim function is transparent when MACsec is not enabled, as shown in figure bbb.

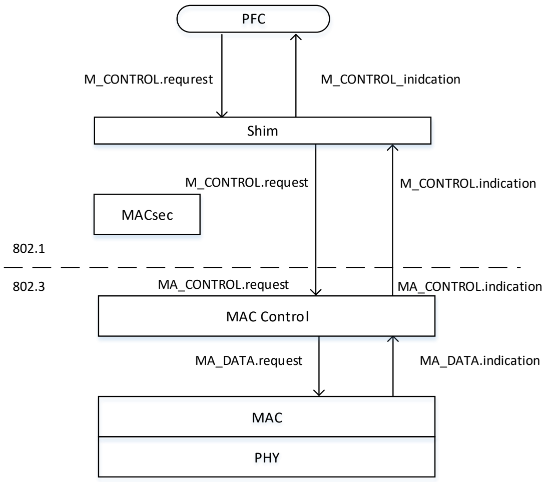


Figure bbb --- MACsec disabled PFC frame transmission

When the shim function receives an M_CONTROL.request primitive and MACsec is not enabled, it transparently passes the primitive to the underlying MAC control interface.

When the shim function receives an M_CONTROL.indication from underlying MAC service, it transparently passed the primitive to upper layer PFC function.

When the shim function receives an M_UNITDATA.indication primitive from the underlying MAC service and MACsec is not enabled, it transparently passes the primitive to the upper layer MAC service.

36. Priority-based Flow Control (PFC)

Modify the description as following.

This clause specifies the operation of PFC (see 36.1), ~~and~~ the architecture of Priority-based Flow Control in a PFC-aware system (see 36.2) and the automatic PFC headroom calculation (see 36.3).

36.1 PFC operation

36.1.1 Overview

Replace figure 36-1 with the following figure

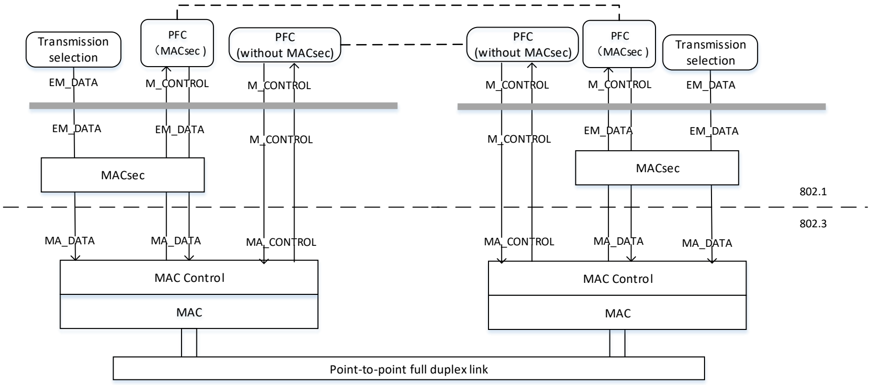


Figure 36-1—PFC peering

Insert new paragraph at the end of this subclause:

PFC is intended to be used on full-duplex links. When PFC is invoked, there is a time delay between the PFC invocation on the PFC initiator and the pause action on the PFC receiver. In order to guarantee no data frames are dropped by the PFC initiator, a certain amount of buffer needs to be available at the PFC initiator to absorb the data in flight after the PFC frame has been transmitted. The reserved buffer space is also known as PFC headroom. A method to automatically calculate the headroom is specified in subclause 36.3.

Insert new subclause 36.3 and its subclauses and tables, as shown, re-numbering as necessary.

36.3 Automatic PFC headroom calculation

Automatic PFC headroom calculation provides a method to configure the minimum amount of buffer space required on the PFC initiator to guarantee no packet loss when using PFC.

Automatic PFC headroom calculation follows a worst-case delay model to determine the headroom requirement (see figure N-3 of Annex N). The calculation considers the time between the PFC frame invocations by the PFC initiator, until the last bit of the PFC frame is received by the PFC receiver. The total delay value formula is specified in Annex as below.

$$\mathrm{DV} = 2 \times (\text{Max Frame}) + (\text{PFC Frame}) + 2 \times (\text{Cable Delay}) + \mathrm{TXd}_{s1} + \mathrm{RXd}_{s2} + \mathrm{HD}_{s2} + \mathrm{TXd}_{s2} + \mathrm{RXd}_{s1}$$

Cable delay is the propagation delay over the transmission medium.
TXds1 and RXds1 are the interface delay of PFC initiator. RXds2 and TXds2 are the interface delay of PFC receiver. Interface delay is specified N.3.

HDs2 is higher layer delay of PFC receiver. Higher layer delay is specified in N.4.

The total delay value can be divided into medium delay, internal processing delay and fixed delay, shown in figure xxx.

$$\text{Delay Value} = \underbrace{2 \text{ x (Cable Delay)}}_{\text{Medium delay}} + \underbrace{\text{TXds1} + \text{RXds2} + \text{HDs2} + \text{TXds2} + \text{RXds1}}_{\text{Internal processing delay}} + \underbrace{2 \text{ x (Max Frame)} + \text{(PFC Frame)}}_{\text{Fixed delay}}$$

Figure xxx --- Delay value formula

In the figure xxx, medium delay depends on the deployment environment. A measurement mechanism is described in clause 36.3.1. Internal processing delay is vendor specific, comprises interface delay and higher layer delay. The value of internal processing delay is calculated using the mechanism described in clause 36.3.2. Fixed delay equals to length of time to transmit 2 times maximum frame and PFC frame. With medium delay, internal processing delay and fixed delay, clause 36.3.3 describes the calculation of PFC headroom.

36.3.1 Medium delay measurement

Medium delay is the time of a full-duplex point-to-point round trip transmission. If it is symmetric point-to-point link, medium delay is 2 times cable delay. The measurement uses the peer-to-peer delay mechanism shown in figure yyy. It is the same as the mechanism specified in IEEE Std 1588-2019, supporting both one-step procedure and two-step procedure.

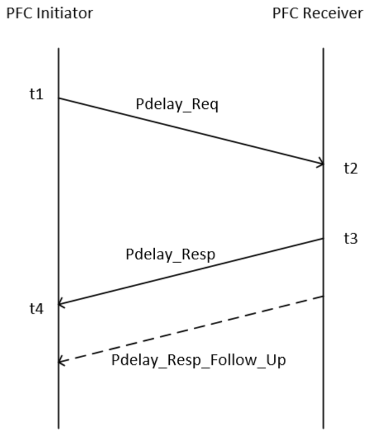


Figure yyy --- Medium delay measurement

For one-step procedure,

a) PFC initiator issues a Pdelay_Req message and generates a timestamp, t1.
b) PFC receiver generates a timestamp, t2, upon receipt of the Pdelay_Req message.
c) Upon receipt of Pdelay_Req message, PFC receiver issues a Pdelay_Resp message and generate a timestamp, t3.
   Pdelay_Resp message conveys the difference between the timestamp t2 and t3.
d) PFC initiator generates a timestamp, t4, upon receipt of the Pdelay_Resp

For two-step procedure,

a) PFC initiator issues a Pdelay_Req message and generates a timestamp, t1.

b) PFC receiver generates a timestamp, t2, upon receipt of the Pdelay_Req message.

c) Upon receipt of Pdelay_Req message, PFC receiver issues a Pdelay_Resp message and generate a timestamp, t3.
Pdelay_Resp message conveys the timestamp t2.

d) PFC receiver issues a Pdelay_Resp_Follow_Up message
Pdelay_Resp_Follow_Up message conveys the timestamp t3.

e) PFC initiator generates a timestamp, t4, upon receipt of the Pdelay_Resp

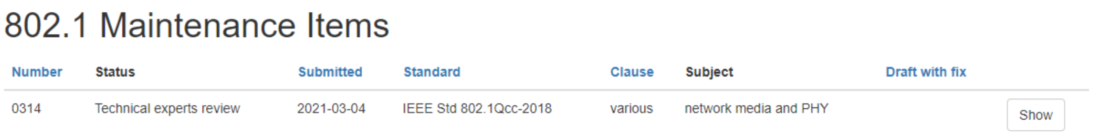
PFC initiator uses these 4 timestamps to compute medium delay.
Medium delay (MD) = t4 – t1 – (t3 – t2)

36.3.2. Internal processing delay calculation

Shown in figure xxx, the total value of internal processing delay equals to the sum of PFC initiator interface delay including TXds1, RXds1, and PFC receiver interface delay including TXds2, RXds2 and HDs2. The values are implementation specific. Although some MAC interfaces, such as IEEE Std 802.3 specify the maximum value of the interface delay, implementations always are much smaller. Both PFC initiator and PFC receiver obtain their own internal processing delays. PFC receiver conveys its internal processing delay to PFC initiator using DCBX (see clause 38). The calculation of total internal processing delay is done at PFC initiator. After receiving PFC receiver's internal processing delay, PFC initiator adds the received value and its internal processing delay, to get the total value of internal processing delay.

<<Note -- Consider to add 'reference plane' description.>>

(Comment: Reference plane should be considered and aligned in 802.1 Q
Response: Agree. Adopt the result of the maintenance group discussion.
There is a maintenance item 0314 discussing reference plane, reference point definition in 802.1AC and 802.1Q. The item points out current text describing reference plane and reference point in 802.1Q is problematic. Latest status is "It was agreed to review 802.11 timing reference points to see if these could be used in a common description of an 802.1 reference point."



802.1 Maintenance Items

Number	Status	Submitted	Standard	Clause	Subject	Draft with fix	
0314	Technical experts review	2021-03-04	IEEE Std 802.1Qcc-2018	various	network media and PHY		Show

)

36.3.3 PFC headroom calculation

The calculation of PFC headroom takes place at the PFC initiator. PFC initiator gets the delay value by adding medium delay (see subclause 36.3.1), internal processing delay (see subclause 36.3.2) and fixed delay.

Besides delay value, PFC headroom calculation needs a correction coefficient. That is to adjust the accuracy considering implementation specific impact, such as internal buffer fragmentation. So the headroom calculation formula is illustrated as below.

PFC headroom = Delay value * alpha

alpha is implementation specific coefficient.

38. Data Center Bridging eXchange protocol (DCBX)

38.2 Goals

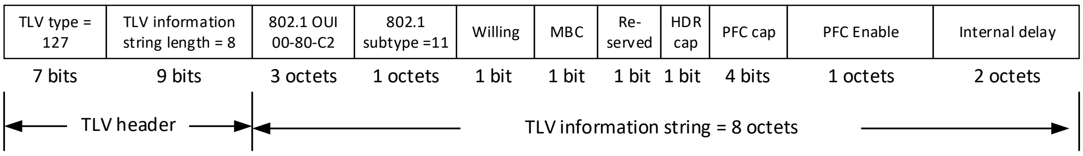
Modify item a) as following

The goals of DCBX are as follows:

a) Discovery of DCB capability in a peer port; for example, it can be used to determine if two link peer ports support PFC, and if automatic PFC headroom configuration is supported.

D.2.10 Priority-based Flow Control Configuration TLV

Replace figure D-10 with the following figure.



TLV type = 127	TLV information string length = 8	802.1 OUI 00-80-C2	802.1 subtype =11	Willing	MBC	Re-served	HDR cap	PFC cap	PFC Enable	Internal delay
7 bits	9 bits	3 octets	1 octets	1 bit	1 bit	1 bit	1 bit	4 bits	1 octets	2 octets

Figure D-10 Priority-based Flow Control Configuration TLV format

D.2.10.2 TLV information string length

Modify the description as following.

A 9-bit unsigned integer, occupying the LSB of the first octet of the TLV (the MSB of the TLV information string length) and the entire second octet of the TLV, containing the total number of octets in the TLV information string of the Priority-based Flow Control Configuration TLV. This does not count the TLV type and TLV information string length fields. It is equal to ~~6~~ 8.

(Comment:

Does field ‘MBC’ in PFC configuration TLV format need update?

Response: The definition of MBC is ambiguous. It needs update.

“36.1.3.3 Timing considerations

If MACsec is not supported, a queue shall go into paused state in no more than 614.4 ns since the reception of a PFC M_CONTROL.indication that paused that priority.

…….

If MACsec is used, a queue shall go into paused state in no more than 614.4 ns + ‘SecY transmit delay’ (see IEEE Std 802.1AE) since the reception of a PFC M_CONTROL.indication that paused that priority.

……

If MACsec is supported but not used, the delay computation has to take into account the MACsec Bypass Capability (MBC) bit in the PFC configuration TLV of DCBX (see IEEE Std 802.1Qaz subclause 38.5.4), that indicates if the link peer needs the extra time for MACsec. If the MBC bit is set to zero, the maximum PFC delay is 614.4 ns. If the MBC bit is set to one, the maximum PFC delay is 614.4 ns + ‘SecY transmit delay’. ”

MACsec can be supported but not used for PFC. MBC should be the indicator if MACsec is used for PFC.

“D.2.10.4 MBC

The MACsec Bypass Capability Bit. If set to zero, the sending station is capable of bypassing MACsec

processing when MACsec is disabled. If set to one, the sending station is not capable of bypassing MACsec processing when MACsec is disabled (see Clause 36). ”
)

Modify subclause D.2.10.4

D.2.10.4 MBC

The MACsec Bypass Capability Bit. If set to zero, the sending station is capable of bypassing MACsec processing for PFC when MACsec is ~~disabled~~ supported. If set to one, the sending station is not capable of bypassing MACsec processing for PFC when MACsec is ~~disabled~~ supported (see Clause 36).

Insert new subclause D.2.10.x

D.2.10.x HDR cap

A 1-bit unsigned integer that indicates the device support of automatic PFC headroom calculation. If the HDR cap bit is 1, and PFC is enabled on at least one traffic class, the automatic headroom calculation is enabled.

Insert new subclause D.2.10.x

D.2.10.y Internal delay

A 2-octet unsigned integer contains the length of time for which the device process received PFC pause frame. It includes TX interface delay, RX interface delay and higher layer delay. The value is measured in units of pause_quanta, equal to the time required to transmit 512 bits of a frame at the data rate of the MAC.

D.5 IEEE 802.1/LLDP extension MIB

TBD content

D.6 IEEE 802.1/LLDP extension YANG

TBD content

Annex M - Support for PFC in link layers without MAC Control

M.1 Overview

Modify the description as following.

Priority-based Flow Control (PFC) is a function defined for only point-to-point full-duplex links in terms of the M_CONTROL primitives (11.4 of IEEE Std 802.1AC-2016 [B9]). For IEEE 802.3 link layers the M_CONTROL primitives are mapped into the MAC Control MA_CONTROL primitives (IEEE Std 802.1AC), that use the PDU format defined in IEEE Std 802.3. Other link layers supporting point-to-point full-duplex operations need to define their mapping of the M_CONTROL primitives. Shim layer (see Clause 6.7.3) provides the way to map M_CONTROL primitives and M_UNIDATA primitives. This annex describes a PDU format suitable to support PFC.