DCBX PFC Configuration

Issues and proposals to clarify use of DCBX for PFC Configuration.[1],[2]

PFC Config—current state & goals

LLDP MIB and YANG support

DCBX spec + LLDP MIB vs YANG model

Mick Seaman
mickseaman@gmail.com

1. The whole point of P802.1Qdt is to make management of PFC easier. Automating buffer allocation calculation, whether by delay measurement or by using PTP together with realistic estimates of local delays in the possible configuration scenarios (unprotected data and PFC, MACsec protected data, MACsec protected data and PFC, Privacy protection) is just part of that.
2. This presentation follows up on some (but not all) of the issues described in https://www.ieee802.org/1/files/public/docs2024/dt-seaman-pfc-management-0924-v02.pdf

PFC Config—current state (1)[1]

A conformant PFC implementation only has to be able to enable PFC on one priority, but may be enabled on up to eight [5.11:a), g)].

If a traffic class encompasses both PFC and non-PFC priorities, the latter will be paused which is not recommended.[2,3]

If PFC is enabled on a port, 802.3 PAUSE is not used for that port.[4]

DCBX controls ‘the application of ETS and PFC’.[5,6]

But Clause 38 ‘DCBX’ only mentions PFC once. The PFC Configuration TLV is the only TLV passed by ‘Symmetric attribute passing’, but the state machine does not mention it explicitly and only considers ‘in progress’ configuration as a note.[7]

---

1. P802.1Q-Rev/D1.4 (unchanged, I believe w.r.t., the above points from IEEE Std 802.1Q-2022).
2. This recommendation is only in a Note (8.6.8 NOTE 2) which is problematic from a conformamnce language perspective.
3. See also 8.6.8.4 NOTE 3 (mis-numbered) “The use of PFC is likely to interfere with a traffic schedule …”
4. Assuming this means PFC configuration overrides PAUSE mgmt settings. Decision: inhibit PAUSE if PFC is not currently enabled because DCBX has not yet decided which priorities to PFC enable, but is likely to some (or perhaps DCBX local default is to enable some). See later.
5. IEEE8021-PFC-MIB only provides per port PFCLinkDelayAllowance, and counts of PFCRequests and PFCIndications, no configuration of which priorities are protected by PFC.
6. 8.6.8.2 ‘Credit-based Shaper Algorithm’: “Traffic classes using the credit-based shaper algorithm shall not use PFC and shall ignore the setting of the bits related to such classes in the PFC Enable bit vector.”
7. Which is not normative.

PFC Config—current state (2)

The ‘Symmetric state machine’ initializes an OperParam attribute value with a LocalAdminParam, and updates it (conditionally)[1] to the value in the last received TLV. The attribute value transmitted in the relevant TLV is always OperParam [38.4.2.1 a)]. LLDP is thus used to acknowlege receipt and update (if appropriate) of the attribute value sent by the peer.[2]

The PFC Configuration TLV attribute includes a PFC Enable bit map (D.2.10). There is no precision in the standard, as to what the term “enable PFC” and its reflection in each priority bit in a transmitted TLV means:

a). I will pause transmission for this priority, if I receive a PFC for it.

b). I will transmit PFCs for this priority.

c). I think traffic of this priority should be subject to PFC.

DCBX goals/benefits for PFC Config TLV use need to be made explicit.

---

1. Provided that the local system’s Willing (to be configured) bit is set, with a MAC Address comparison tie-breaker that ensures that only one of the two peers updates its OperParam value to the value received from the other.

2. 38.4 says “ LLDP is a unidirectional protocol.” It is unclear, given symmetric attribute passing of operational parameters with changes to those parameters as information is received, what that statement means in the context of DCBX. Depending on the rules for changes in the presence of constraints that could result in a chatty protocol, and, if any additional attribute is to be passed using symmetric attribute passing is not feasibe given that LLDP passes all the information that is not carried in an extension on each transmission.

DCBX PFC Config TLV goals

General DCBX goals (38.2) are peer capability detection, misconfiguration detection, and peer configuration (if peer allows).

For PFC Config specifically, **assume the major benefit** is ‘PFC storm’ avoidance—do not transmit PFCs that will have no effect.[1,2]

Additional benefit: turn off PFC for selected priorities to avoid congestion spreading or excess local buffering, without direct management of peer.[3]

802.1Q provides no guidance as to:

— what type of system[4] should be ‘Willing’, accepting its peer’s configuration.

— if not ‘Willing’ should it transmit PFCs without receiving its Peer’s TLV, or possibly without adapting to its Peer’s TLV.

— PFC priorities to enable if number of traffic classes supported is restrictive.

---

1. This is consistent with, though not explicitly stated by, use of PFC ‘only in a domain controlled by DCBX’—if ‘domain’ is simply a link.
2. `lldpXdot1dcbxConfigPFCTxEnable` enables TLV transmission, protocol operation decides whether that results in PFC transmission.
3. Clause 38, by using symmetric attribute passing to agree common values, assumes a symmetric relationship between the peers, but this is not necessarily the case at the edge of network where pausing end station transmission can cause it to pace its creation of frames to be transmitted, but pausing edge bridge transmission causes buffer accumulation in the bridge and possible congestion spreading
4. For example, edge bridge or end station, or level within the network hierarchy to prefer core or edge configured values. If both peers are ‘Willing’ then the choice is essentially random or vendor dependent (compares MAC Address values).

LLDP-EXT-DOT1-V2-MIB[1]

The management configured per-port boolean `lldpXdot1dcbxConfigPFCTxEnable` enables/disables PFC TLV transmission.[2]

Read-only `…dcbxLocPFCBasicEntry` and `…dcbxLocPFCEnableTable` describe the local system's settings[3] for the TLV and whether PFC is enabled for each priority. Similar read-only information is recorded for the remote system in `…dcbxRemPFC…` objects.[4]

Read-write entries in the `…dcbxAdminPFCTable` and `…dcbxAdminPFCEnableTable` control initial TLV settings.[5]

No objects for remote peer defaults,[6] before PFC Config TLV received could:

a). Not transmit PFC on any priority. Consistent with prior assumptions.

b). Assume PFC configurations match (not recommended).

---

1. D.5.5, page:line references below are to P802.1Q–2022–Rev/D1.4.
2. It takes 55 lines in the MIB to say just that. 2226:41 to page 2227:24.
3. Not described as 'operational' values, so interpretation/guesswork required to match the Clause 38 description. 2233:45 to 2235:8.
4. 2242:26 to 2243:70
5. Matching Clause 38. 2251:30 to 2253:2
6. Unlike, for example, LACP.

ieee802-dot1q-lldp-dcbx.yang[1]

DCBX TLV transmission is controlled by `tlvs-tx-org-dcbx-enable`.

The grouping `pfc-tlv` describes PFC Config TLV fields.[2] A single local system read-write copy, container `pfc-tlv-extension` augments `/lldp:lldp/lldp:port`, and a remote system read-only copy augments `/lldp:lldp/lldp:port/lldp:remote-systems-data`.

If the single local system `pfc-tlv` is the OperParam of 38.4.2.2 (current operational value of the attribute fields) then there is no way to see what administratively configured values led to that current state. If it is the LocalAdminParam value (which seems more likely, given read-write of relevant fields), then the YANG module does not provide the current state.[3]

1. D.6.6.3.
2. Page:line 2304:8 to 2305:13
3. The astute manager should be able to work out the current state, given the rules in this presentation (which are not in the standard) except for cases where the number of traffic classes supported by PFC is a constraint.

DCBX Symmetric TLV state machine

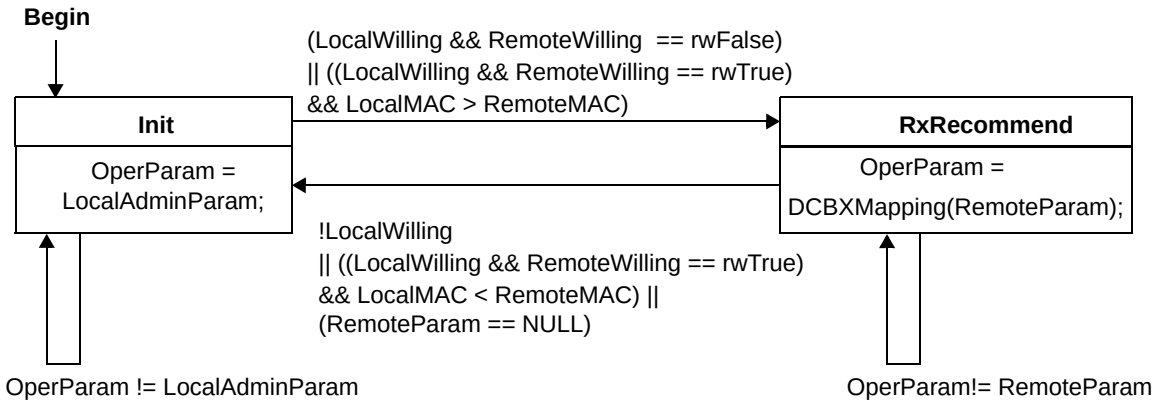

Figure 1-1—Symmetric state machine

NOTE—Through observation of the states and state variables it is possible to determine that the state machine is in the process of passing attributes. This knowledge may be useful for clients of DCBX. For example, a client may wish to delay use of the link while the DCBX state machine is in the process of passing and possibly setting attributes. A Pending indication indicating that the state machine is in this process may be created by the following equations:

Pending = RemoteParam == NULL || !LocalWilling && RemoteWilling == rwTrue && OperParam != RemoteParam;

- Start-up behavior unspecified. One peer can decide to delay use of the link, while the other believes it to be operational immediately. Can PFCs be sent immediately (potential storm) or not If? Pending as defined can be true for ever.
- If !LocalWilling and Remote has PFC capability constraints, Pending can be permanently True.
- State machine transition choice unspecified if two are possible, so RxRecommend can spin lock if DCBXMapping is not trivial identity function. Similarly Init can spin lock, defeating parameter updates.
- Defined value ‘rwNull’ not used, ‘RemoteParam == NULL’ instead, does not match prior description.
- No behavior defined for ‘!LocalWilling && !RemoteWilling’ - agree to disagree.

Proposed state machine