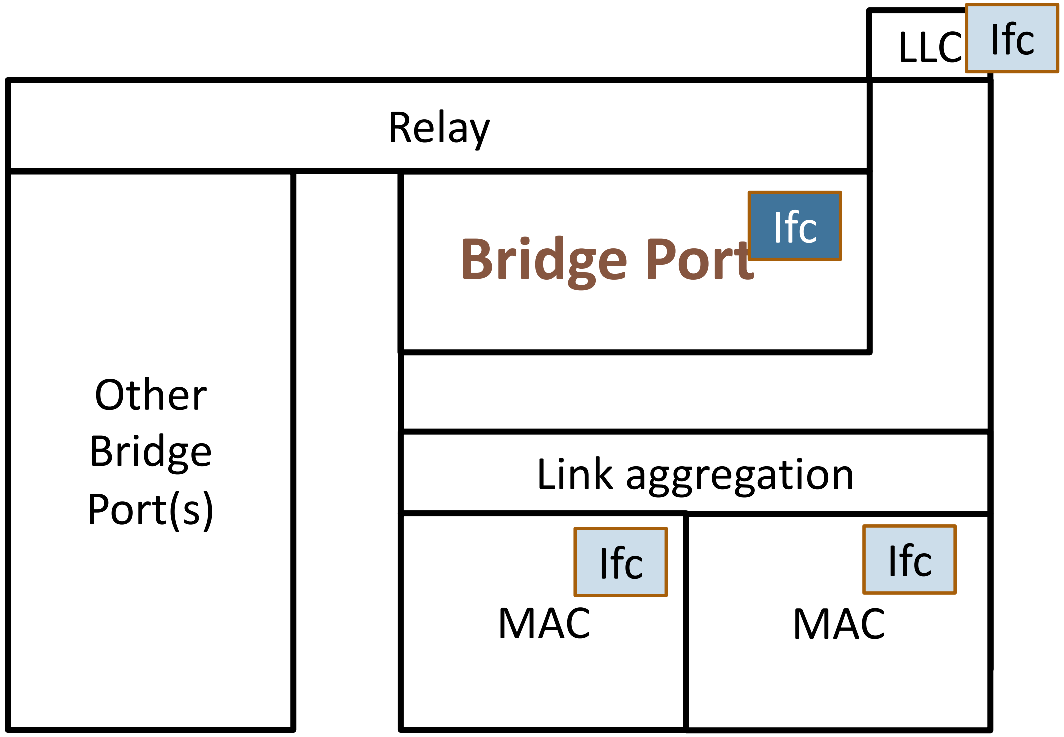
Tying YANG QoS Modules to Interfaces, Not to Bridge Ports

Norman Finn
Huawei Technologies Co. Ltd
cw-finn-interfaces-not-ports-0920-v04
1` November, 2020

Bridge Ports vs. Interfaces

- An interface is a set of managed objects (RFC2863, RFC8343) that can manage real things, e.g. IEEE Std 802.3 ports, or virtual things, e.g.:
  - Virtual interfaces between bridge components within a system.
  - Protocol shims
  - Aggregations of physical links.
  - One end of an RFC4667 Layer 2 Virtual Private Network (L2VPN) link.
- Interfaces can have 1:*n* or *n*:1 vertical relationships in an "interface stack".
- How many layers in a stack should an implementation expose to management as interfaces? This is a fun engineering question.
- **A Bridge Port is an interface**. (Not in 1999, but bridges with multiple components, connectivity fault management, and link aggregation made interfaces necessary.)
- An interface in a Bridge is **not necessarily** a Bridge Port, e.g.:
  - An aggregated link that is part of an aggregation of links.
  - An LLC "pocket" in the baggy pants can be instantiated in the interface stack.

Bridge Ports vs. Interfaces

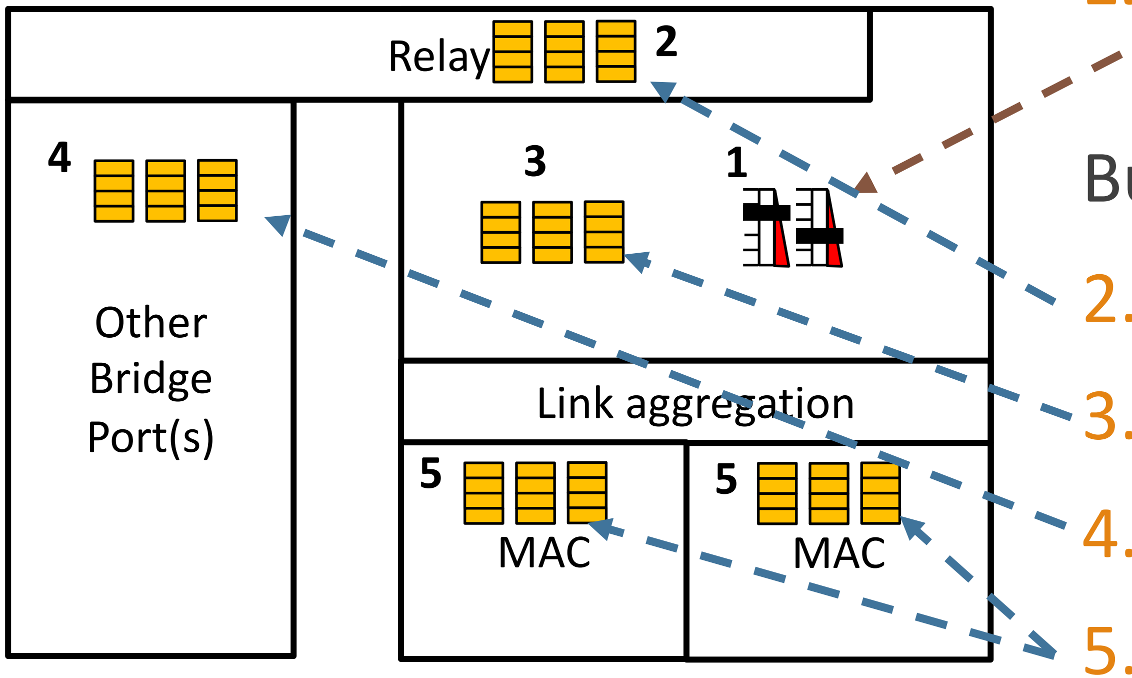


- Bridge Port is definitely an Interface
- LLC pants pocket can be exposed as an interface, but it is not a Bridge Port.
- Physical ports can be exposed as interfaces, and must be if Link Aggregation is to be managed. They are not Bridge Ports.

Where are queues and their controls in IEEE Std 802.1Q?

- In the bridge architecture, queues are associated with bridge ports.
- Depending on the implementation, the actual queuing structures may be in a central brain, tied to the bridge port, they may be on a line card, tied to a few ports, or they may on the individual ports. (Or they may be virtual, or distributed, or any anything else.)
- Originally, there was no difference; 1 bridge port = 1 physical port. The implementer simply had to ensure that the end result was equivalent to the simple model given in IEEE Std 802.1Q.

IEEE Std 802.1Q queuing model
Where are the queues? The controls?



1. **One set of controls** for transmitting per Bridge Port

But, **queues can be anywhere**:

2. Centralized queues
3. Bridge Port queues
4. Virtual out queues at input
5. Physical port queues

The implementer must ensure that the observable behavior is consistent with choice 3.

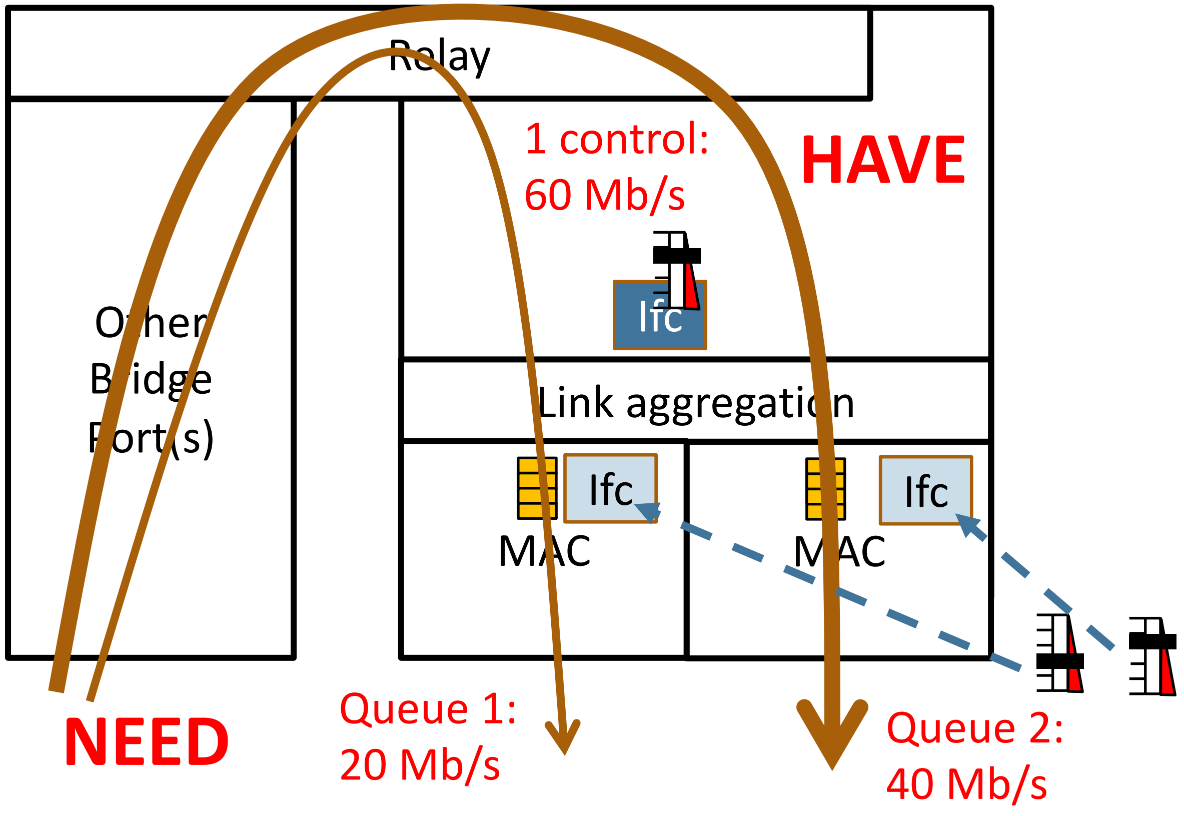
Where are QoS controls in IEEE Std 802.1Q?

- When Link Aggregation (IEEE Std 802.3ad, later IEEE Std 802.1AX) was introduced, it still didn’t matter whether one put the queues in the center or in a physical port, because strict priority works just the same if the bridge port parameters are applied to individual physical ports’ queues.
- The addition of Enhanced Transmission Selection (ETS, aka Weighted Fair Queuing) did not change this indifference; the bridge port parameters can be replicated to the physical ports, and everything works as expected.

Credit Based Shaper and Link Aggregation

- The Credit-Based Shaper introduced a problem, if one implements CBS in the physical ports of an aggregation.
  - When a bridge port’s CBS is split among several CBSs on several physical ports, then bandwidth of each Stream has to be added to the particular CBS on the physical port through which it passes; each physical ports’ CBS can require a different idle slope.
- So, if managing CBS using a centralized server (as defined in IEEE Std 802.1Qcc), the network controller needs to manage physical ports’ CBS parameters, not bridge ports’ CBS parameters.
- **This is not possible in IEEE P802.1Qcw D1.2.**

802.1Q model fails for CBS implemented per-physical port with LinkAg and CNC



Two 100 Mb/s links.

Two flows, 20 Mb/s, 40 Mb/s.

With **MSRP, no problem**. Implementation is hidden, and Bridge adjusts ports' shapers.

With **CNC** (network controller), **big problem**. The two shapers cannot be controlled with one control; two controls are necessary.

Other shapers

- The same considerations apply to Scheduled Traffic.
  - If a small periodic window is used for a critical Stream, it is wasteful to put that window on every port of an aggregation; only the physical port that carries the Stream needs the window.
- The same considerations apply to ATS.
  - Streams following the same path can share a queue, but if ATS is distributed over aggregated links, the sharing is altered.
- That is, a TSN network controller needs to manage physical ports for QoS, if that's where the shapers and queues are best modeled as residing for a particular implementation.
- *This is not about where the queues are implemented, but about how many sets of controls are needed to make TSN work.*

Non-bridge uses of QoS management

- End stations need to use the QoS mechanisms defined in IEEE Std 802.1Q and its amendments.
  - Scheduled transmission
  - Preemption
  - Priority Flow Control
  - Credit Based Shaper
- This is clearly intended by the existing text in IEEE Std 802.1Q, and is why that standard is so careful about using "Bridge Port" vs. "Port".
- But, the YANG modules in P802.1Qcw D1.2 are tied to the bridge port, and bridge component, and an end station has no bridge port and no bridge component.

Proposal: Tie QoS controls to Interfaces

- If we tie the QoS controls to interfaces, instead of bridge components and bridge ports, we can handle both cases:
  - Link Aggregation in a bridge.
  - End stations.
- We lose nothing by making this change; the interface can still be a bridge port.
- We do ***not*** need to change things like priority mapping, VLAN port table, and such; these are bridge port functions, and make no sense being applied to anything else. (Even the traffic-class-table is an L2-priority-to-queue assignment that is a bridge port function.) AFAIK, the current split between P802.1Qcw and IEEE Std 802.1Qcp is just fine.

Other considerations

- In my opinion, we do not need to worry about the MIBs; others may disagree.
- The implementation decides to what interfaces the QOS controls are attached, and whether that is a bridge port, a physical port, or some internal point in the interface stack (e.g. a line card).
- This change needs to be applied to **IEEE P802.1Qcr**, as well.
- The justification for moving the control of **input** features such as timed gates to the interface module is less strong, but is still present, because IEEE Std 802.1AX Distributed Resilient Network Interconnect (DRNI) can vary the reception timing greatly on different links of an aggregation.
- It is also important to IEEE P802.1DC that this change be made.
- There are lots of details to work out, but this is a start.

UPDATE: Nov. 1, 2020

- Applying controls for the Credit-Based Shaper on a per-physical Port basis by a Central Network Controller is **not** a good idea.
  - Link Aggregation is dynamic – links come and go.
  - Waiting for the CNC to reprogram the CBS parameters defeats the purpose of Link Aggregation’s quick response capabilities.
- **The need for end stations to have managed controls for QoS remains.**

UPDATE: Nov. 1, 2020

- When this presentation was made to the weekly dial-in, it was pointed out that changing the root of the P802.1Qcw YANG queuing controls from Bridge Port to Interface might conflict with Clause 12 of IEEE Std 802.1Q.
- So, I searched P802.1Q-rev D0.2 for "**Bridge Port**" and for "**queue**".

Conflicts between 802.1Q and rooting P802.1Qcw on interfaces:

UPDATE: Nov. 1, 2020

The whole document has been very careful to distinguish between "Port" (often, "transmission Port") and "Bridge Port".

In particular, Clause 12 (management) is very clean.

In the search process, I also found some other issues that can be raised to the Maintenance TG.

Conclusion

Rooting the IEEE 802.1Qcw (and, unfortunately, IEEE 802.1Qcr) QoS controls to the Interface, instead of the Bridge Port, is:

1. Necessary, because we need those controls to manage QoS in end stations; and
2. Needs no changes to the text of the document other than, perhaps, a mention in the YANG section about the fact that the root for QoS is not the same as the root for the other Bridge Port functions (e.g. spanning tree state).

Thank you