Cyclic Queuing and Forwarding, Paternoster, and TSpecs

dd-finn-CQF-and-shaping-0120-v01
Norman Finn
January 7, 2020

Abstract

The per-flow active counter of cr-seaman-paternoster-policing-scheduling-0519-v04 can be a useful part of a network employing Cyclic Queuing and Forwarding (CQF, IEEE Std 8021Q-2018 Annex T). It can be used for 1) reducing overprovision of CQF Streams; 2) crossing boundaries where the CQF cycle time changes; and 3) crossing boundaries from TSN regions utilizing another queuing technology into CQF regions. The token-bucket parameters already defined in IEEE Std 802.1Q for policing need to be used for characterizing Stream requirements, as well.

1 Introduction

This paper assumes the reader is reasonably familiar with the enhancements to CQF presented in df-finn-multiple-CQF-0919-v01 and with the paper by Mick Seaman, cr-seaman-paternoster-policing-scheduling-0519-v04. This is the third in a series of papers on useful expansions of the basic CQF idea, which includes df-finn-multiple-CQF-0919-v01 and df-finn-CQF-latency-matching-0919-v01.

As described in section 1 of df-finn-multiple-CQF-0919-v02, the present paper is dealing with the problem of providing a service characterized by bounded latency and zero congestion loss for continuous Streams in a store-and-forward environment.

IEEE 802.1Q-2018 characterizes a Stream using MSRP parameters defined in 46.2.3.5.1 of IEEE Std 802.1Qcc-2018. These parameters include:

1. Interval: A rational number of seconds.
2. MaxFrameSize: The maximum size of a frame in the Stream.
3. MaxFramesPerInterval: The maximum number of (maximum sized) frames that the source can be transmitted for this Stream in one Interval.

On the other hand, IEEE Std 802.1Q-2018 and P802.1Qcr also characterize a Stream using the parameters:

4. CommittedInformationRate: the long-term average data rate (bits/sec) that the source promises not to exceed; and
5. CommittedButstSize: the number of excess bits that can be transmitted in advance of the committed rate.

We will show that there are two problems with using the current TSpec with CQF (or the variants of CQF described in the papers cited above:

A. Using Interval, MaxFrameSize, and MaxFramesPerInterval with CQF requires unnecessary overprovisioning that could be reduced or eliminated by using MaxFrameSize and CommittedInformationRate.
B. If a Talker transmits data using any technique other than CQF, e.g. the Credit-Based Shaper stack recommended by Figure 34-1 of IEEE Std 802.1Q-2018, then additional overprovisioning is required at every hop, to absorb momentary variations in frame transmission rates at the source.

Both of these problems can be reduced or eliminated by incorporating the CommittedInformationRate into the stream reservation protocols (2. TSpec, CommittedInformationRate and CQF), and by utilizing the per-flow active counter described in the Paternoster paper. (3. Paternoster for CQF Ingress Conditioning). These considerations lead to some recommendations for P802.1Qdd Resource Allocation Protocol (4. Recommendations for Stream Characterization). The cycle time boundary crossing problem is also explained (5. Changing CQF cycle times).

2 TSpec, CommittedInformationRate and CQF

The current TSpec gives some measure of the worst-case behavior of a Stream. Most Streams suitable for TSN can be characterized usefully by CommittedInformationRate which is, (in P802.1Qcr), the number of bits per second the Stream requires on the wire, including per-frame overhead. The resource allocated to each Stream by CQF is bit times on the wire per transmission cycle T_C. In the best case, a Stream will use exactly (T_C * CommittedInformationRate) bits per cycle. In the absence of detailed information about repeated patterns of frame sizes for a given Stream, the greatest number of bits that a Stream can occupy in one cycle occurs when the frames are divided into the worst-case assortment of sizes, in which case (MaxFrameSize – 1) bits have to be added to that best-case allocation.

Depending on the exact numbers, this overprovisioning by (MaxFrameSize – 1) bits can be annoying. However, with the current TSpec, we have to compute a value for CommittedInformationRate based on the worst cases Interval, MaxFrameSize, and MaxFramesPerInterval, and that number is higher, producing even more overallocation. (It is higher because it reflects the worst-case behavior, not the steady-state average.)

3 Paternoster for CQF Ingress Conditioning

Simply put, given CommittedInformationRate and MaxFrameSize, one can compute the minimum allocation of bits per Stream per cycle. If one uses the Paternoster algorithm at the ingress to a CQF network, with the Paternoster cycle time synchronized to the network's CQF cycle time(s), one gets the minimum overprovisioning possible for CQF (or Paternoster), and Paternoster's per-Stream state machines are required only at the edges of the network.

4 Recommendations for Stream Characterization

CQF, Paternoster, and ATS all work better with the parameters:

a. MaxFrameSize
b. CommittedInformationRate
c. CommittedButstSize

These should be the parameters used in any resource allocation protocol.

We note one further issue with the current TSpec: IEEE Std 802.1Qcc does not say how to count the frames for determining MaxFramesPerInterval. Is it the number of start-of-frame delimiters, the number of end-of-frame delimiters, the number of whole frames transmitted, or what? If it is not some particular point in the frame, e.g. the time stamp point, then when one slides the Interval test window so that the ends are in the middle of a frame, is that frame counted in the window or not? Figure 1 offers a diagram that is useful for examining this issue. Does the red interval count the right-hand frame or not?
It seems reasonable to use number of frames started.

Figure 1 TSpec Intervals

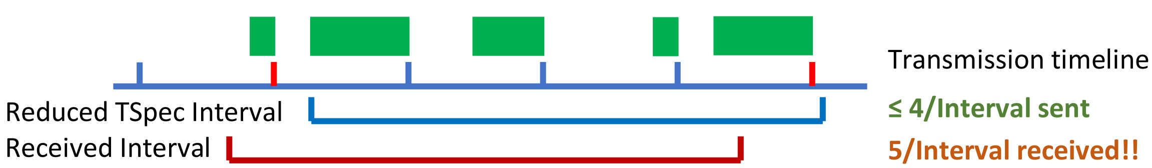

Figure 1 also demonstrates a more important point, which has an impact on using the TSpec for CQF: although the Talker never *transmits* more than 4 frames per Interval, no matter where the measuring Interval starts, the first-hop bridge can *receive* 5 frames in one Interval! Since CQF has no place to put that extra frame, it must allocate space for one extra MaxFrameSize frame per Interval. Of course, using Paternoster at ingress solves this.

I would therefore recommend that we clarify the meaning of MaxFramesPerInterval.

5 Changing CQF cycle times

There is a case where a Stream passes from a network region using cycle time (n * T_C) to a region using cycle time T_C, the two regions are synchronized in time, and the integer n is small, and no special steps need be taken; the frames belonging to n consecutive cycles in the fast (source) region are deposited naturally into a single slow cycle.

In any other case, the Paternoster per-flow active counter can be utilized to reallocate frames to different-sized cycles. The only requirement is that the Streams characterization is interpreted into a suitable resource allocation in each region. This is easy with parameter set suggested, above.