Multiple Cyclic Queuing and Forwarding

dv-finn-overlapped-CQF-0224-v01
Norman Finn
Huawei Technologies Co. Ltd
February 13, 2024

Abstract

The per-hop forwarding latency achievable by Bin Cyclic Queuing and Forwarding (BCQF), described in IEEE P802.1Qdv Draft 0.4, can be improved by realizing that an output bin can be both filling and transmitting simultaneously. This can achieve modest improvements in any BCQF network, and significant improvements in virtual or physical ring topologies.

1 Introduction

This paper assumes that the reader is familiar Multiple Cyclic Queuing and Forwarding and/or Annex Y of IEEE P802.1Qdv Draft 0.4, which provide an introduction to Bin Cyclic Queuing and Forwarding (BCQF). In these documents, BCQF is presented in a manner that, for clarity, assumes that an output bin is either available for storing data, available for transmitting data, or neither, but never for both. In fact, it is possible for the receive and transmit cycle timers to overlap cycles so that a bin is both receiving data and storing data simultaneously. This can reduce the per-hop forwarding (not counting the link delay) from the one-to-two cycles demonstrated in the draft to less than one cycle time, as shown below.

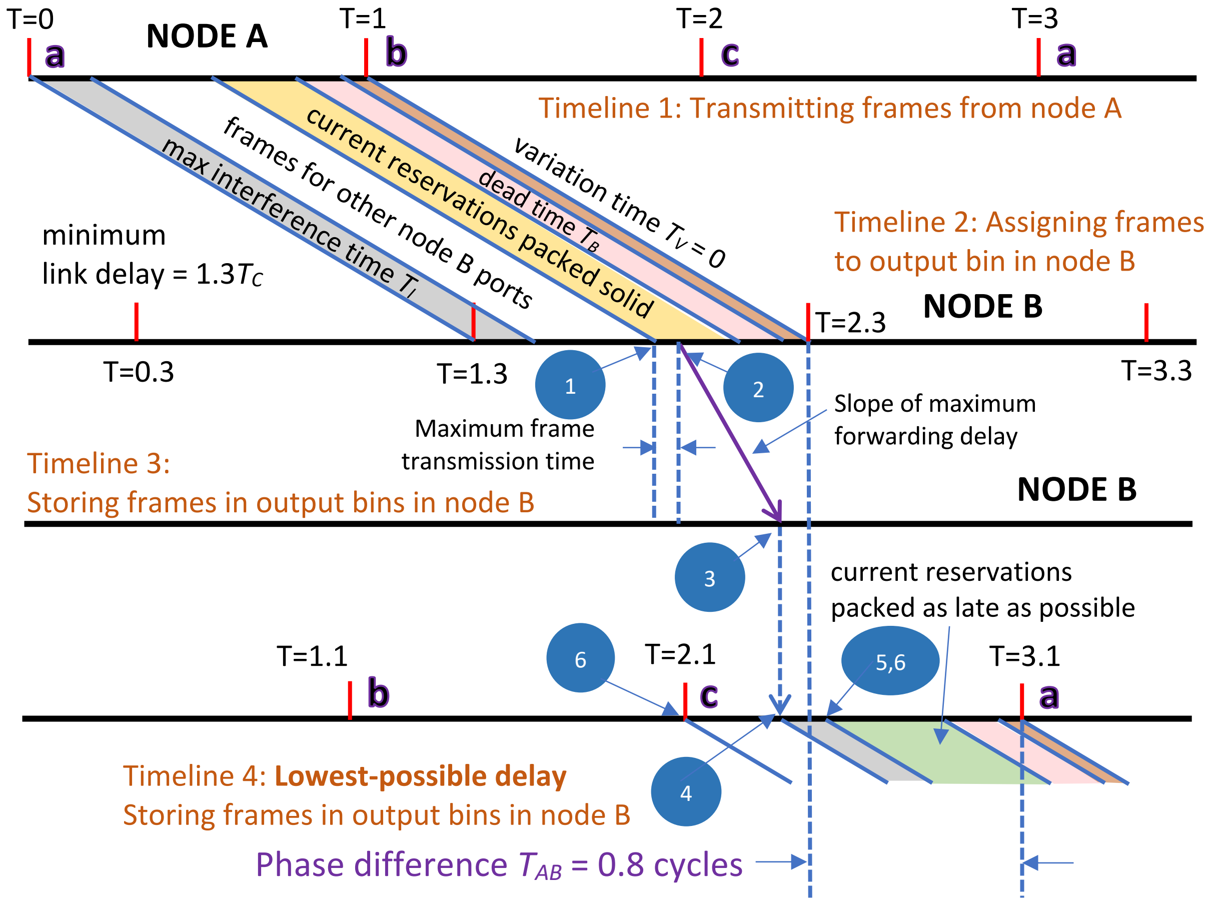
2 Minimum BCQF forwarding delay

We have two nodes, A and B. Both are running an instance of BCQF on at least one port. We will examine the flow of data from one input port of node B to one output port of node B. Node A's output port (the one feeding the input port of node B) and the output port of node B are frequency locked. We do not assume that the nodes' output bins switch in synchrony; they can be out of phase.

After a bin is enabled for transmission, node A transmits all of the frames in that bin towards receiving node B, not necessarily in a single burst. After one cycle time T_C, it starts transmitting the frames from the next bin. The transmission enabling events happen regularly, with the same period T_C. At the next hop, node B must be able to assign each received frame to a bin such that 1) frames that were in the same bin in node A, and are transmitted on the same port from node B, are placed into the same bin in node B; and 2) frames in different bins in node A are placed in different bins in node B.

Figure 1 shows an example of BCQF. This figure differs from the figures in in Multiple Cyclic Queuing and Forwarding and IEEE P802.1Qdv Draft 0.4 in that node **B**'s bin **c** is, for a time, both storing and transmitting frames.

Figure 1 BCQF maximum overlap, minimum delay


T=0
a
NODE A
T=1
b
T=2
c
T=3
a
Timeline 1: Transmitting frames from node A
variation time $T_V = 0$
dead time T_B
current reservations packed solid
frames for other node B ports
max interference time T_I
minimum
link delay = 1.3T_C
Timeline 2: Assigning frames
to output bin in node B
T=2.3
NODE B
T=0.3
T=1.3
T=3.3
1
2
Slope of maximum
forwarding delay
Maximum frame
transmission time
Timeline 3:
Storing frames in output bins in node B
NODE B
3
current reservations
packed as late as possible
T=1.1
b
6
T=2.1
c
5,6
T=3.1
a
Timeline 4: Lowest-possible delay
Storing frames in output bins in node B
4
Phase difference T_{AB} = 0.8 cycles

Node A and node B are transmitting at the same frequency, but are offset by 0.1T_C, as shown by timelines 1 and 4. In Figure 1, we use the following notation for time intervals:

T_C nominal (intended) period of the bin-swapping cycle
T_I maximum interference from lower-priority queues, one frame or one fragment
T_V sum of the variation in output delay, link delay, clock accuracy, and timestamp accuracy
T_A the part of the cycle allocable to (reservable by) streams
T_P worst-case time taken by additional bytes if this traffic class is preemptable
T_B end-of-cycle dead time optionally imposed on node A by node B
T_{AB} effective phase difference between cycle start times from input from A to output from B

The numbered blue circles in Figure 1 indicate the following points in time:

1. The last possible moment of arrival for the first bit of the first frame destined for the illustrated output port in node **B**. The yellow shape between Timeline 1 and Timeline 2 contains all of the frames passing from the illustrated input port to the output port in node **B**. There may be frames destined for other ports received before these frames. In defining this point, we assume the frames are transmitted back-to-back without being preempted.

2. The last possible moment for node B to receive the frame. It is delayed from point 1 by the transmission time required for the longest frame that can be destined for the illustrated output port. For store-and-forward frames, this is some time (presumably known to the implementer of node B) shortly after the last bit of the frame arrives. For cut-through frames, this could occur somewhat earlier. We assume that the tolerances align such that all of the pad in T_V is used up, as if $T_V = 0$. The dead time T_B still must be respected by node **A**.

3. It takes a certain maximum amount of time for node **B** to decide to which port and which bin to forward the frame, to place the frame in the output bin, and for the frame to become ready to be transmitted, if and when selected. The end of this maximum forwarding delay is point 3.

4. Point 4 is the first moment that the first bit of that first frame could be transmitted. The gray interference time T_I has been moved to a point following point 4, instead of being at the beginning of the cycle, to point out that, in this scenario, there can be a gap in the transmission of data from the bin, leading to further interference from lower-priority frames.

5. This is the latest point at which the first bit of the first frame from the input port is sure to be transmitted.

6. Point 6 is the moment by which the first bit of the first frame must start transmission, if the bin is to be emptied before the end of the transmission window. Point 6 is found by counting backwards from the end of the bin **c** cycle. The green area represents all of the frames to be transmitted from this bin, not just the frames from the particular input port illustrated; frames from other input ports could have been stored in bin **c** ahead of those from the illustrated port. It includes all interference from higher-priority traffic (BCQF queues with shorter T_C values) and additional bytes added if its frames are preempted (T_P). The dead time T_B and variation time T_V also contribute to point 6.

Point 6 must be later in time than point 5 for BCQF to work. When these two points coincide, as in Figure 1, they illustrates the minimum delay achievable with BCQF between these two ports.

3 Putting overlapped storing/transmitting to work

IEEE P802.1Qdv describes a per-port-pair additive constant that yields an output bin number from the input bin number. Procedures for initializing this constant are given in that document. The value typically chosen is the one that minimizes T_{AB} in Figure 1, and thus minimizes the per-hop frame delay. That description ignores the possibility of simultaneous input and output on the same bin, and therefore is independent of any consideration of reservations made vs. available unallocated bandwidth, or of the routing of frames. This makes the calculations dependent only upon the characteristics of the implementation and the phasing of the bin rotation cycles. This ease of calculation has advantages, but better latency is possible.

As shown in section 2 above, the amount of cycle overlap possible for any given port pair depends upon how much bandwidth is (or could be) reserved for the traffic traversing that port pair. In the worst case, when a single minimum-length from a given input port is destined for an output port that is fully allocated, the computation results in no overlap at all; a bin is never both receiving and transmitting. If nothing is known of actual or possible reservations, no overlap is possible. This point is expanded, below, in section 3.1.

Conversely, if the network designer knows about every stream reservation, then node **B** in Figure 1 could be configured for absolutely optimal delay. However, this configuration would be brittle; any change in the reservations could result in a chain of adjustments throughout the network. Changing the per-port-pair additive constants cannot, in general, be performed while BCQF frames are in flight.

One could, however, utilize configured limits on per-port-pair reservations to good advantage, as described in the following two sections. There are two ways to use the fact that storing and transmitting can be overlapped:

A. Given the phases of the input and output bin rotation cycles, a node can utilize knowledge of limits on reservations between any given pair of ports to calculate the maximum overlap possible for those ports, and thus select a value of the per-port-pair additive constant that knocks one cycle time off the per-hop delay, compared to the non-overlapped calculation (3.1, below).

B. Given knowledge of the limits on reservations, a network can be engineered by adjusting the phases of the bin rotation cycles in nodes' output ports to optimize the delay for certain streams along certain paths (3.2, below).

Both of these techniques can be used together.

3.1 Cycle phase alignments are given

For our purposes, “fan-in” is a term describing the number of input ports from which BCQF frames can be forwarded to a given output port. If the fan-in equals the number of frames that can be output from the output port, i.e. if each input port is contributing one single-frame-per-cycle stream, then no overlap is possible. (Point 1 in Figure 1 is as late as possible, and point 6 is as early as possible.) At the opposite extreme, if the fan-in is one, i.e. all of the data transmitted on a given port is coming from a single input port, then the overlap can be maximized and the per-hop latency minimized. (Point 1 in Figure 1 is as early as possible, and point 6 is as late as possible.)

If nothing is known of the reservations except that a maximum fan-in has been imposed, say a maximum of *n* input ports for a given output port, then the worst case for overlap is than 1/*n* of the total output bandwidth comes from each input port, and for a small-enough value of *n*, some overlap can be utilized when computing the per-port-pair additive constants. It is for this reason that fan-in is a configurable parameter.

If a node is configured with a maximum allocable bandwidth per port pair, then the possible per-pair overlap can be computed, and the optimum value for each per-port-pair additive constant can be chosen.

3.2 Cycle phase alignments are to be optimized

Assuming that a network designer can define maxima and minima for per-port-pair stream bandwidth reservations for each node, the phasing of output cycle boundaries can be adjusted to optimize the delay through specific port pairs in the network. It should be remembered that the phase of the output cycles at the two ends of a link are independent of each other, and that in theory, every output port of a single node can be phased independently.

Optimization by phase alignment can be limited by circular dependencies for optimization, even though, of course, no one stream travels in a circular path.

Phase optimization is also limited by the fan-in considerations given in 3.1, above; significant gains for latency can be obtained only for output ports with the lowest fan-in. However, networks may be able take advantage of limited fan-in in several ways:

a) A network may have, either logically (for the deterministic flows) or physically, a ring topology. In a ring node, it is often the case that the local (non-ring) ports’ bandwidth requirements are known exactly. Reasonable limits can also usually be set for the reserved bandwidth of the ring input port. This is an optimal situation for maximum overlap and minimum per-hop delay. There must, of course, be at least one break in the chain of phase optimization around the ring.

b) The number of endpoints for deterministic flows in more-fully-connected network can be much lower than the number of non-deterministic endpoints, leading to deterministic paths that resemble a ring topology, whether closed (ring) or not (chain).

Further study is required, but overlapped BCQF cycles on a ring appears to be amenable to cut-through forwarding.

4 Summary

BCQF can support the overlap of filling and emptying a bin. To the extent that knowledge of worst-case fan-in and/or stream reservations is known to a forwarding system, this can reduce the per-hop delay and buffering requirements for CQF, particularly for ring or chain topologies. IEEE P802.1Qdv should be updated to reflect this.