Queuing Mechanism Interactions

Getting the AVB shaper, time scheduled selection, weighted priorities, and preemption to work together

Norman Finn

v02 March 10, 2015

Queue-draining mechanisms that can interact

- IEEE Std 802.1p "Strict priority"
- IEEE Std 802.1Qav "AVB shaper"
- IEEE Std 802.1Qaz "Weighted priority"
- IEEE Std 802.1Qbb "Priority Flow Control"
- IEEE P802.1Qbu "Preemption"
- IEEE P802.1Qbv "Scheduled transmission"
- IEEE P802.1Qch "Cyclic Queuing and Forwarding"

Why this presentation?

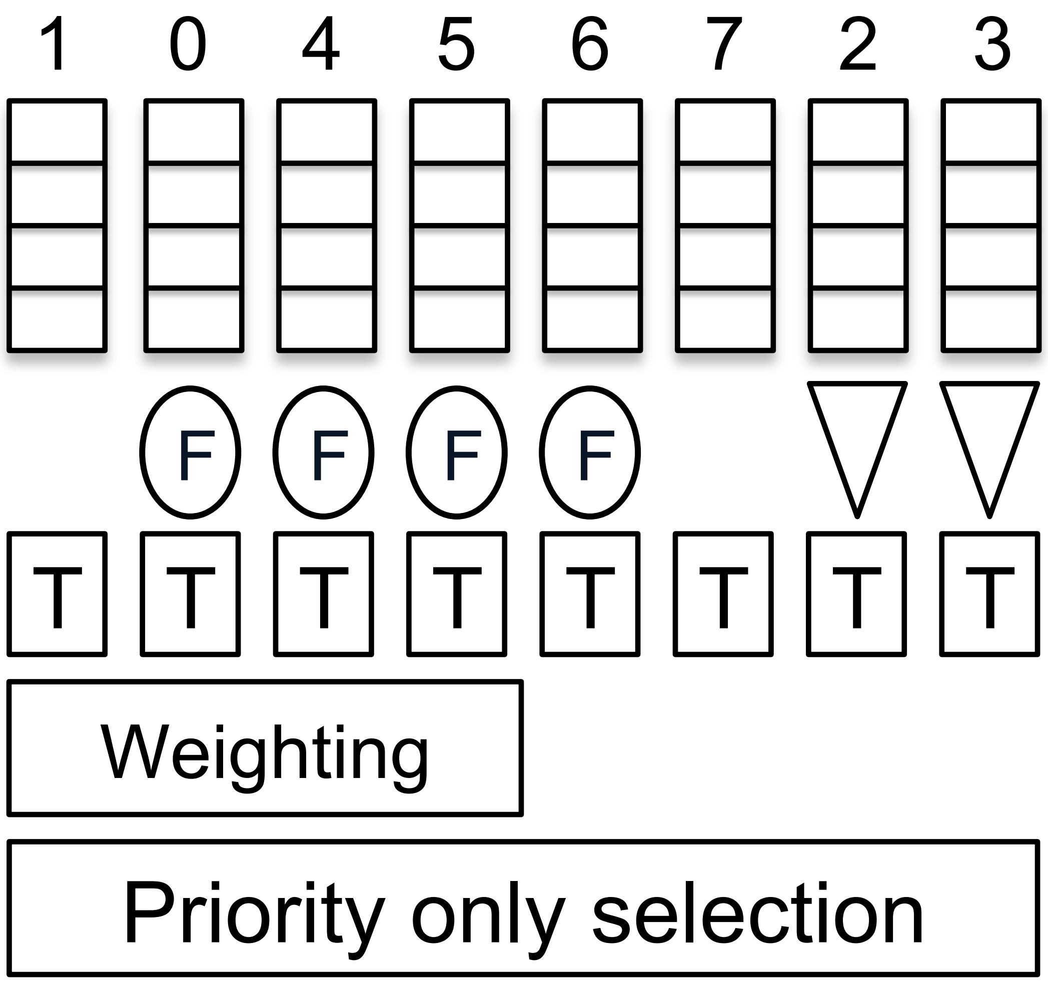
IEEE 802.1 has a number of queuing technologies (see the above list). These can be:

1. A box of random tools acquired over 20 years of tinkering.
2. An integrated mechanism that can be tuned to optimize a wide variety of critical parameters.

This is an attempt to forward the latter choice. I believe that this integration will aid the acceptance of TSN in a wider context.

Layering the transmission selection function

A (proposed) model for shaper interactions



Priority values that select these queues

Up to 8 **queues**, most important on right.
These are **service classes** 0-7, left-to-right.

AVB shaper usually on rightmost queues
PFC on some number of non-AVB queues
Time-scheduled gates on **all** queues

Priority **weighting** only on leftmost queues
(no overlap with AVB shaper allowed)

Selection of rightmost ready queue

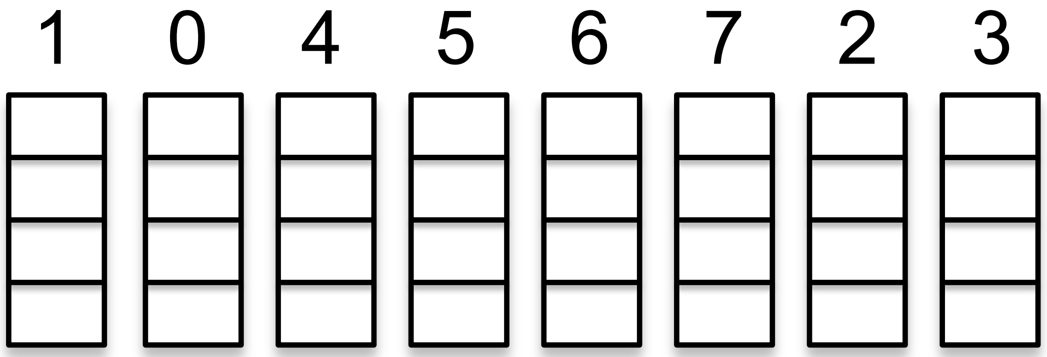
No preemption shown. We'll get to that.

NOTE: This is an example showing the use of all facilities, not a required use of priority values.

Why this particular order?

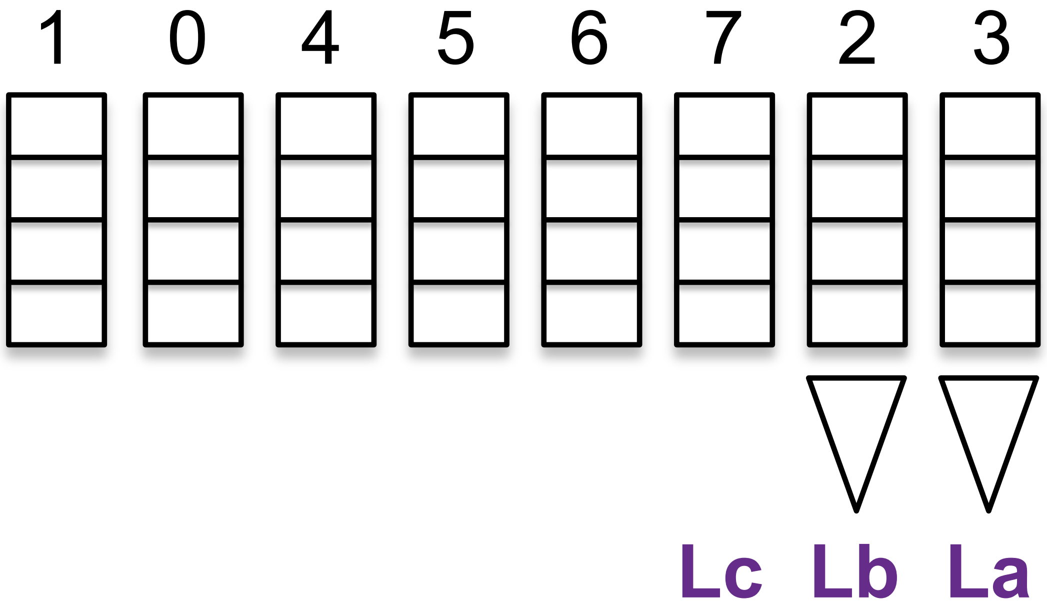
- This stacking is not obvious, and not the same as this author has presented in other slide decks.
- In the following slides, we will look at little pieces of the stack, in an attempt to find constraints on the ordering.
- **The constraints will be in green.**
- We will put those constraints together to re-create this stack.

Queue selection



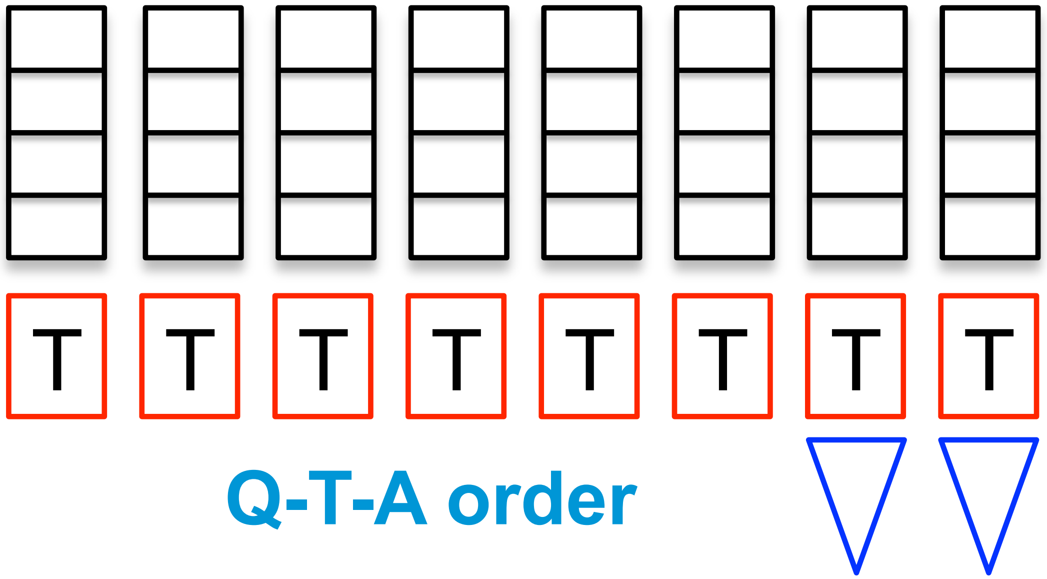
- No proposed changes to how a queue is selected in this deck.
- We can talk, later, about whether or not P802.1Qci Per-Stream Filtering and Policing needs something new.
- There are no typos in the numbers at left; 0 is more important than 1, and in this example, 2 and 3 have AVB shapers, so those frames go to the rightmost queues.
- In other words, queues are in **importance order**, which is **not** (necessarily) **numerical order**.

AVB shapers and queues



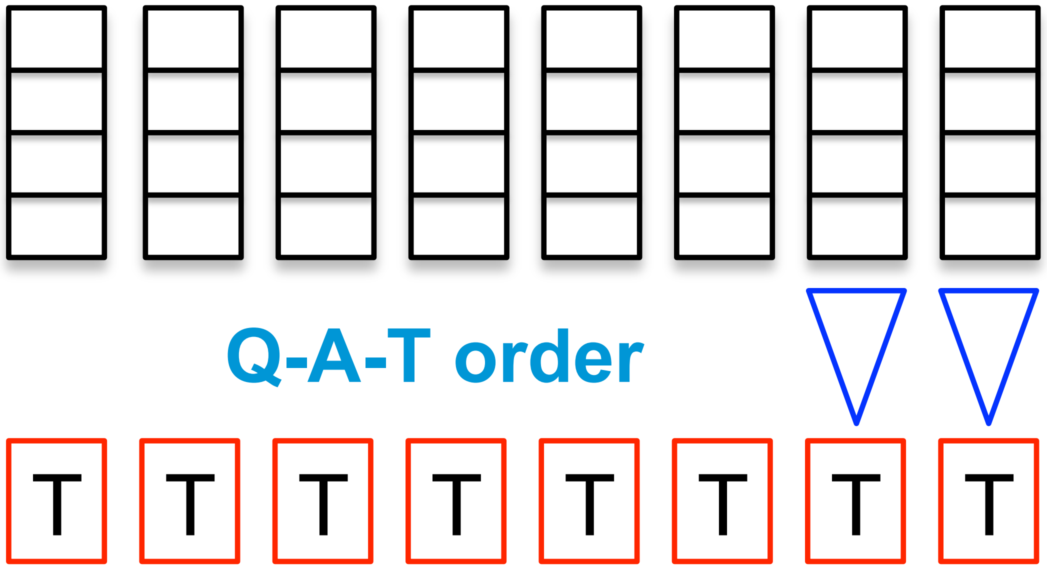
- The AVB shapers go on the most-important queues, no matter what their priority level.
- The AVB shapers guarantee a certain latency Lx to their **own queues** (**La** and **Lb**, in this example), **and to the next-lower queue** (priority 7, **Lc**, in this example).
- The biggest thing that the AVB shaper provides is an easy calculation for how much interference the most-important *n* queues cause in queue *n*+1.
- **Obviously, the AVB shapers must be below the queues they serve.**

Time gates and AVB shapers



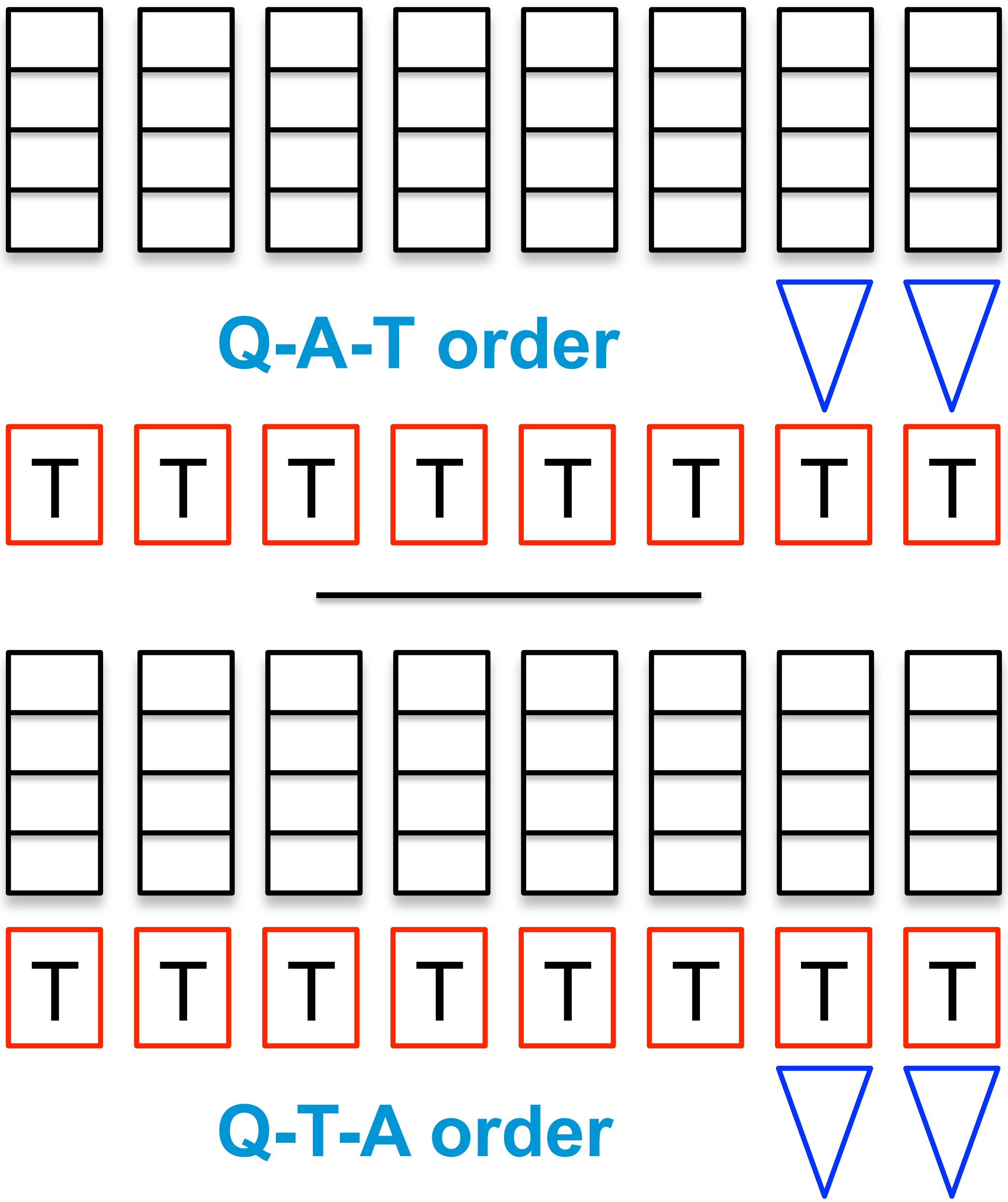
- We could put the time gates between the queues and the AVB shapers.
- This is **Q-T-A** order.

Time gates and AVB shapers



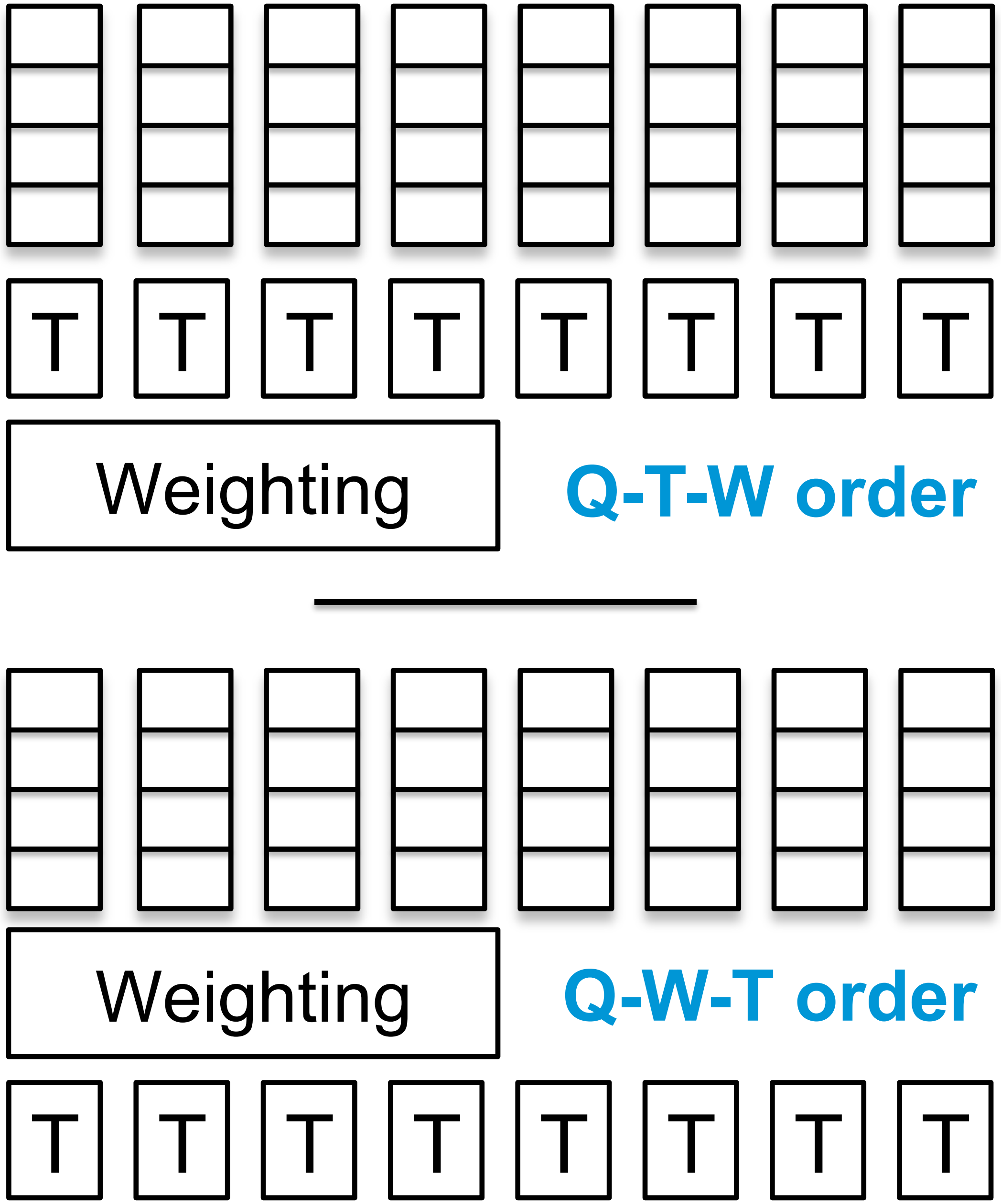
- We could put the AVB shapers between the queues and the time gates.
- This is **Q-A-T** order.

Time gates and AVB shapers



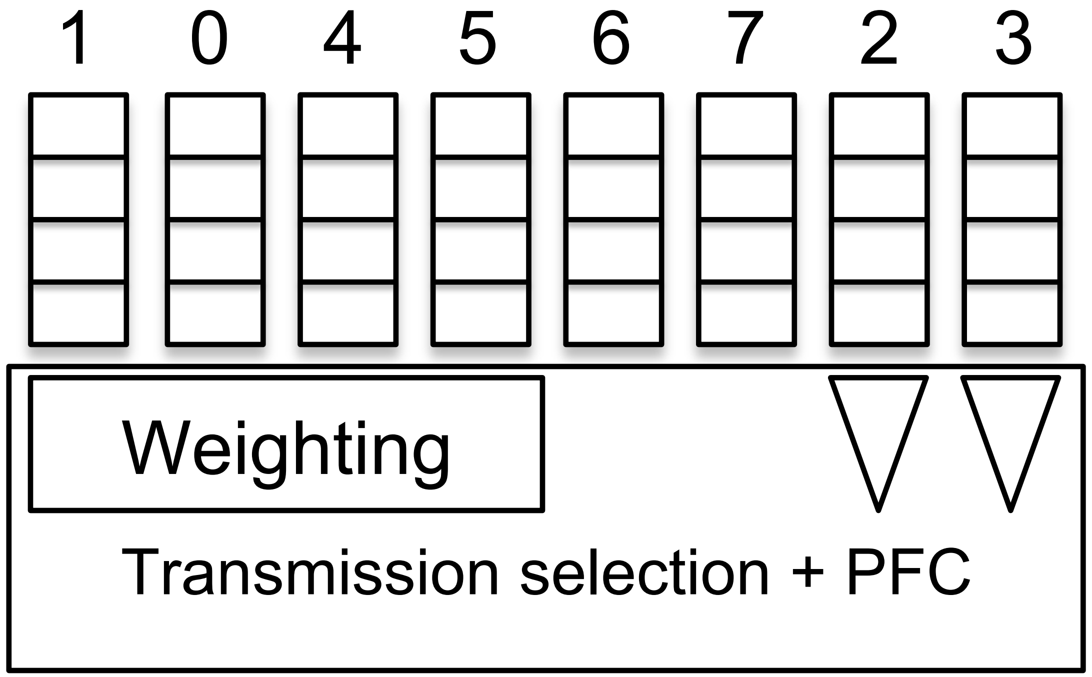
- If the application ensures that an AVB queue is empty whenever its gate is closed, there is **no difference** between the models.
- In **Q-A-T** order, a closed gate acts like interference; when the gate opens, you get a **burst** of traffic. (More about this, later.)
- In **Q-T-A** order, closing the gate when the queue is not empty leads to **unpredictable behavior**; you have reset the credit with an unknown number of packets in the queue.
- So, the order must be **Q-A-T**.

Time gates and weighting



- Similarly, the time gates can be either above (**Q-T-W**) or below (**Q-W-T**) the weighting function (Enhanced Transmission Selection).
- In **Q-T-W** order, the weighting function does not pay attention to queues with closed gates. The weighting function allows only one frame to be presented to Priority + PFC.
- In **Q-W-T** order, it would appear that the selection of the weighting function can, if that queue's gate is closed, cause head-of-line blocking for the whole weighting group.
- Therefore, we have to use **Q-T-W** order.

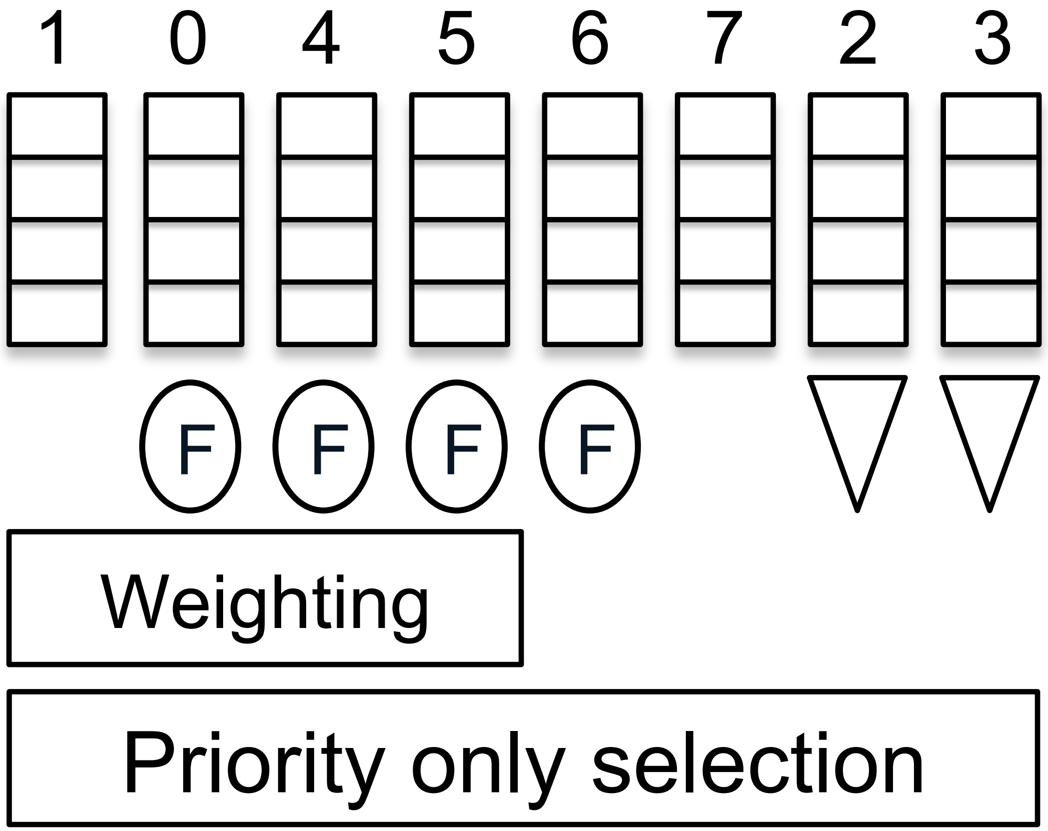
Priority, PFC, and weighting



- IEEE Std 802.1Q-2014 looks like this, IMHO.
- Weighting is on only the leftmost queues*, AVB shaping to the right†, no overlap‡. PFC is part of transmission selection.
- But, there are interactions between weighting and PFC. In particular:
  - PFC does not cause head-of-line blocking among the weighted queues.
  - PFC can be enabled on some (weighted) queues and not others.
- There is no diagram that (to my mind) clarifies the relationship of ETS, PFC, and the AVB shaper.

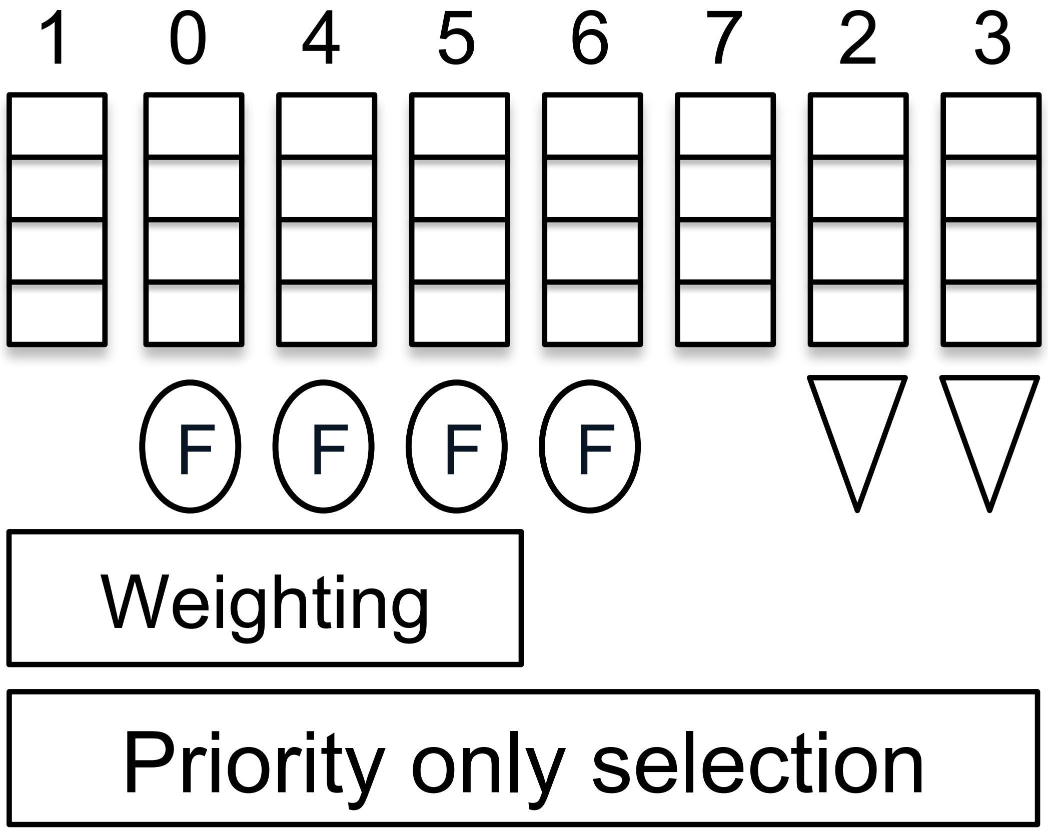
* Required (8.6.8.3:c).
† Not required; just the default.
‡ Required (Table 8-5).

Priority, PFC, and weighting



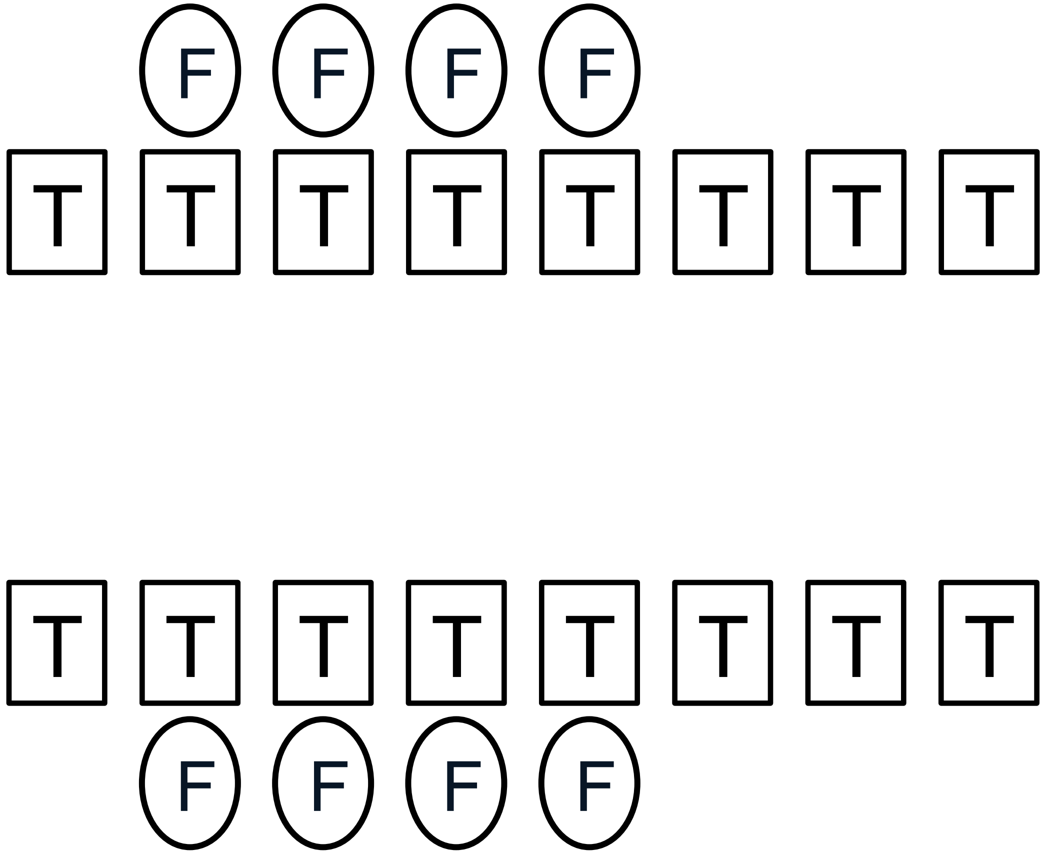
- Clauses 8.6.8 (transmission selection), 36 (PFC), and 37 (ETS = weighting) are clear: ETS and PFC are not allowed on queues using the AVB shaper.
- The descriptions in these clauses are consistent with the picture at left. In particular, **the measurable results of weighting are valid only if PFC is stable for some time** (802.1Q-2014 37.3:d), and **weighting does not select PFC-paused queues** (Clause 36).
- **F-W-P is consistent with 802.1Q-2014.**

Priority, PFC, and weighting



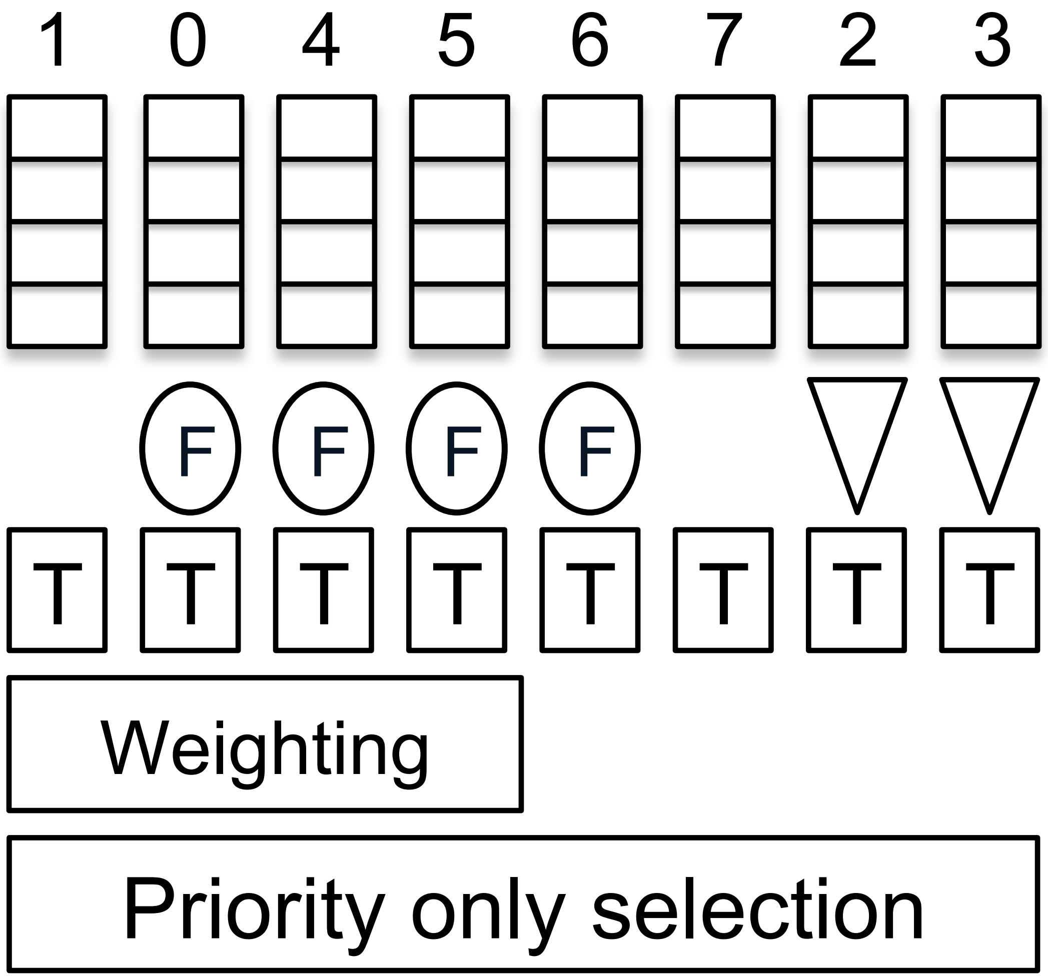
- Note that the definition of Enhanced Transmission Selection is purposely not very tight. This gives room for an implementation that uses some other order.
- This order seems to simplify the concept, however, and minimizes the interactions between the various sub-layers.
- It is also comforting to see two mutually incompatible functions (PFC and AVB shaper) at the same level; the incompatibility then is seen as a choice of alternatives for the “shaper” sublayer.

PFC and time gates



- Although Priority-based Flow Control and time-scheduled gates have considerable internal state, they both effect simple on-off switches, so their order really does not matter. (The packets generated and received by the PFC entity bypass the rest of the queuing stack, anyway. This will be addressed, later.)
- There seems to be nothing in the operation of the PFD or time gates that interact in any way that needs to be explained.
- **These can be in either order.**

A (proposed) model for shaper interactions

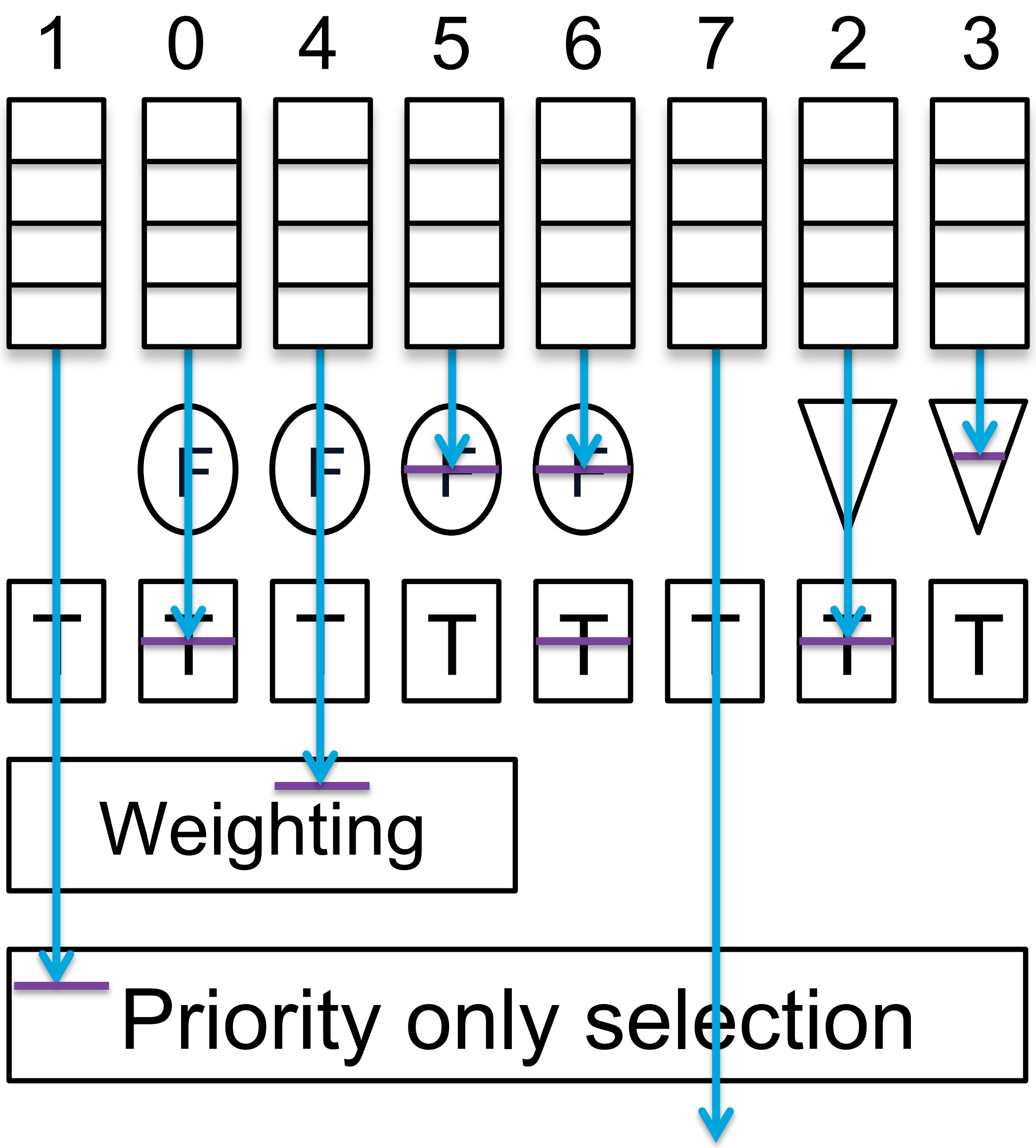


- Adding up the partial constraints gives this model for 802.1Q queuing and transmission selection.

  This minimizes the interactions between levels, e.g., it cleans up the PFC/TDS/priority interactions.

  It restores the simplicity of the final priority-only selection, so that PFC is an optional addition to, not an optional modification of, strict priority selection.

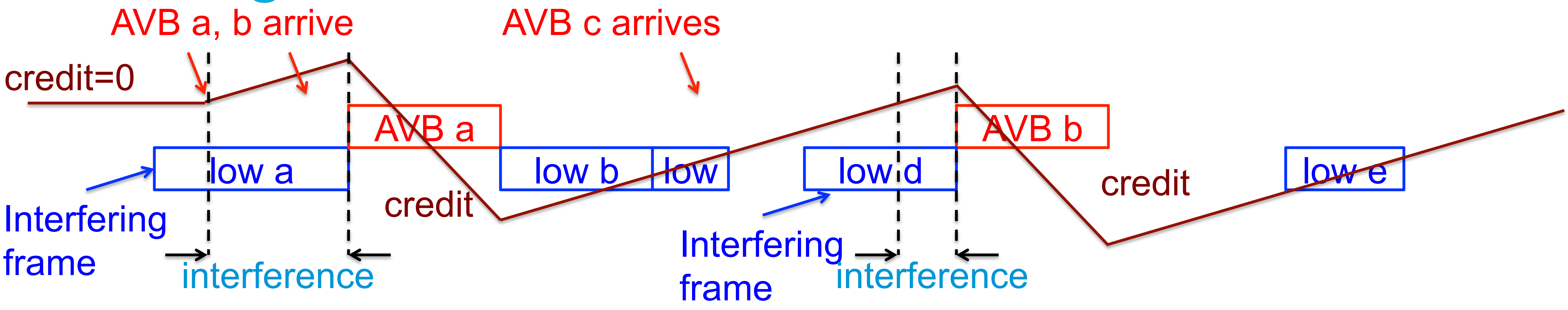
A (proposed) model for shaper interactions



- Note that, in the “down” direction, every function simply gates whether the **“queue not empty” signal is or is not passed** to the priority only selection function.
- This makes an easy-to-describe combination with a minimum of interference among the different layers, simplifying their descriptions.
- It provides a good skeleton for adding new features, as well.

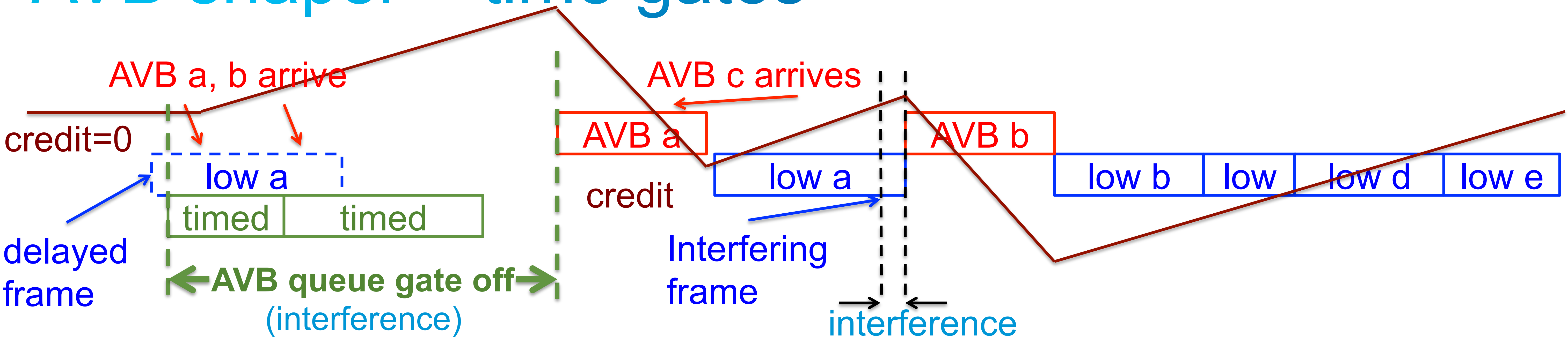
Freezing AVB credit while the time gate is closed

AVB algorithm review



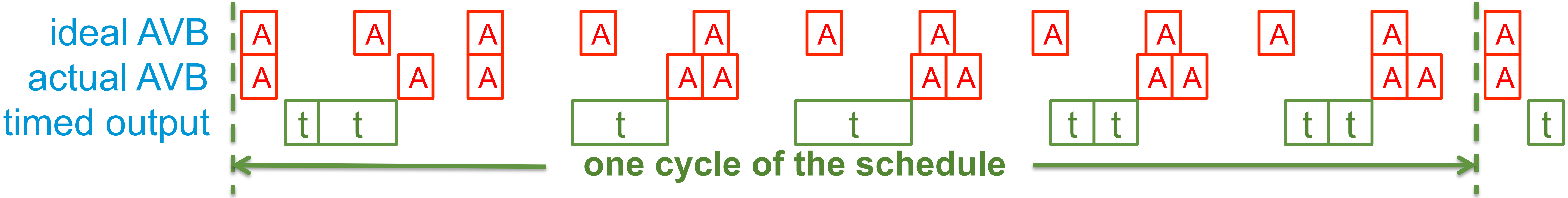
- No AVB frames in queue, so **credit** = 0;
- Three **AVB frames** arrive (arrows).
- **Credit** starts climbing when first arrives.
- Credit declines as AVB frame is transmitted.
- **Interference** from **frames** started while **credit** < 0.

AVB shaper + time gates



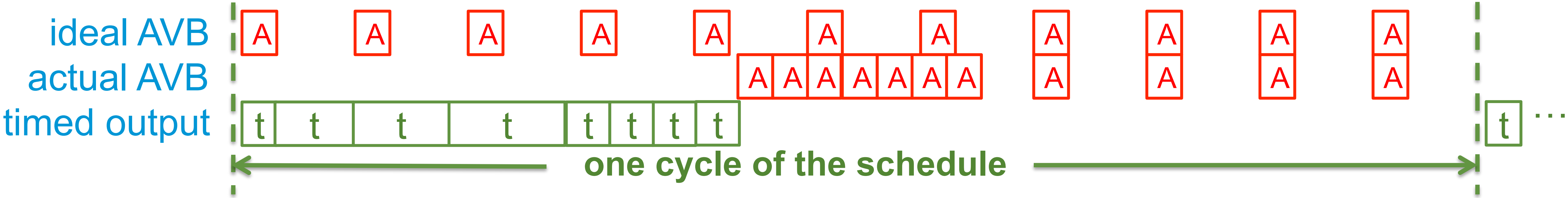
- In the current P802.1Qbv, a **closed gate** on the AVB queue is treated as interference, **just like interference** from a higher-priority AVB queue.
- **This is reasonable.** Things work like you would expect **in the small scale**. MSRP is not impacted.

AVB shaper + time gates



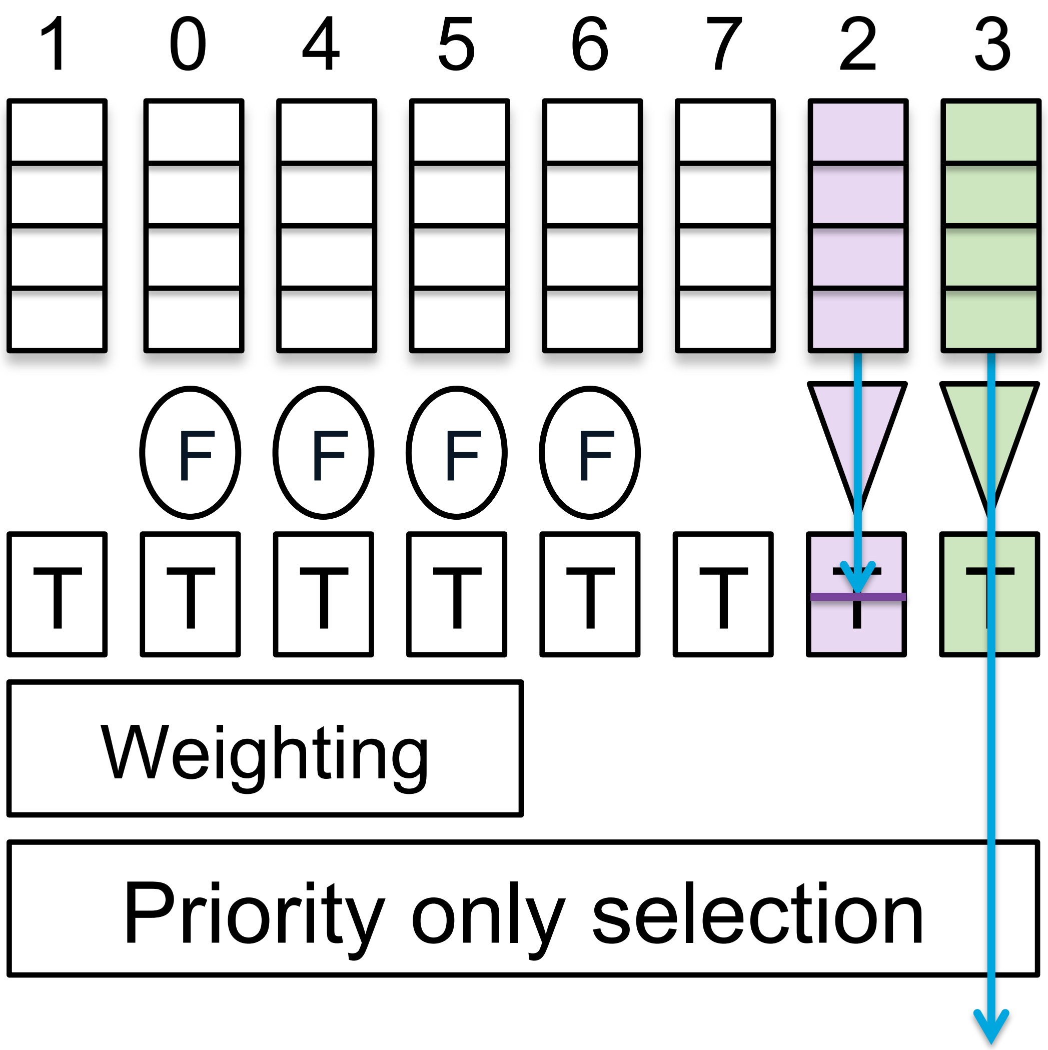
- What do I mean, “Things work like you would expect **in the small scale**”?
- If the scheduled frames are well spaced, there is not a big impact, although inevitably, **the receiver needs more buffers** to handle the bigger lumps.
- Note that **the lumps impact the latency twice**. They delay transmission in the queue with the scheduled frames, and variation in latency this causes results in a requirement for added buffers in the next hop, which equates to an addition to its worst-case latency.

AVB shaper + time gates



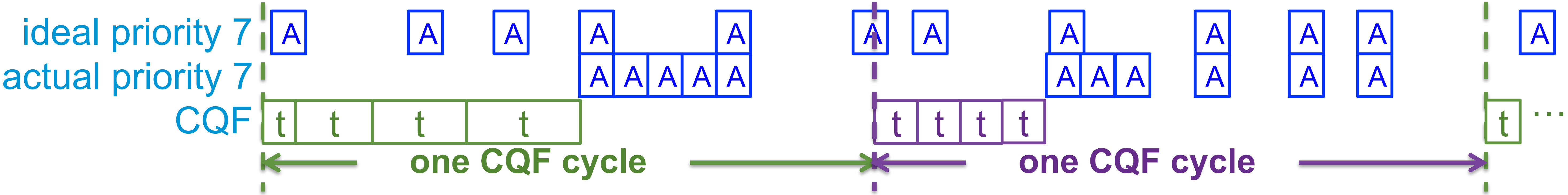
- But what if the schedule is very lumpy? Now, you have a serious lump for the receiver to deal with.
- This lump is **larger** than that caused by Class A → Class B traffic, because Class A is **shaped**, and timed traffic is **not**.
- Looked at from one point of view, this is the inevitable consequence of scheduled transmissions, and there is no problem to fix.
- Patient: “My arm hurts when I do this, Doctor.” Doctor: “Don’t do that.”

AVB shaper + CQF



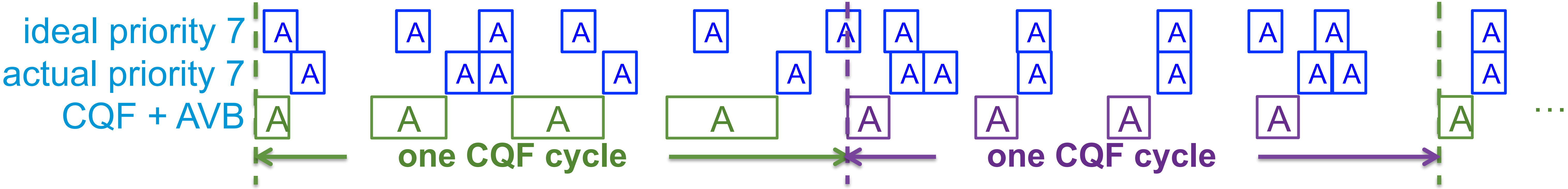
- Patient: “But Doctor, if I don’t do that, I’ll lose my wife, my dog, and my pickup truck.”
- Consider the case of Cyclic Queuing and Forwarding (CQF). In that case, I alternate between transmitting **two buffers**. That is, each CQF buffer has a duty cycle of 50%.
- (There are other models for CQF. See below.)

AVB shaper + CQF



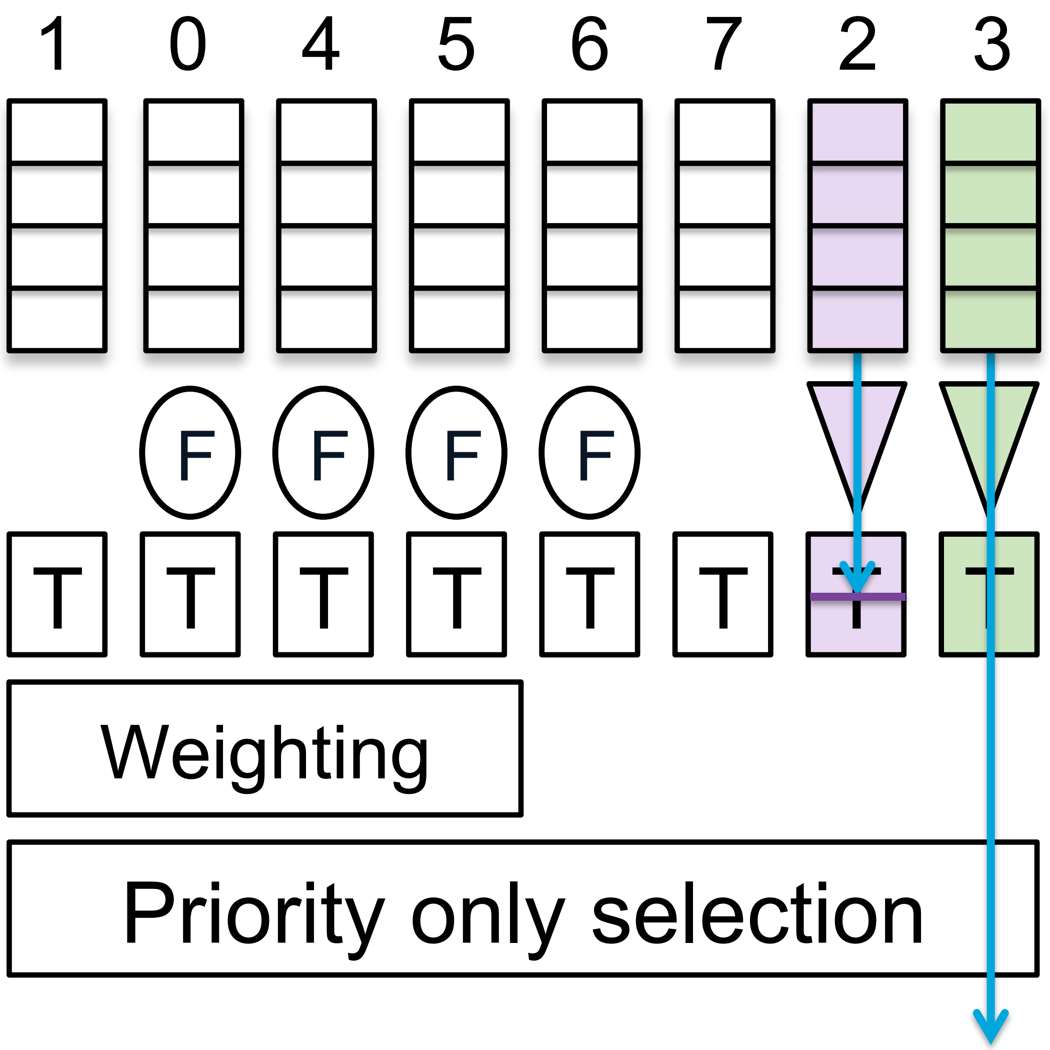
- If used without shaping, CQF does not adversely affect higher-priority AVB traffic, but it is **massively nasty** to the non-critical traffic, especially any AVB non-CQF traffic that has lower priority .

AVB shaper + CQF



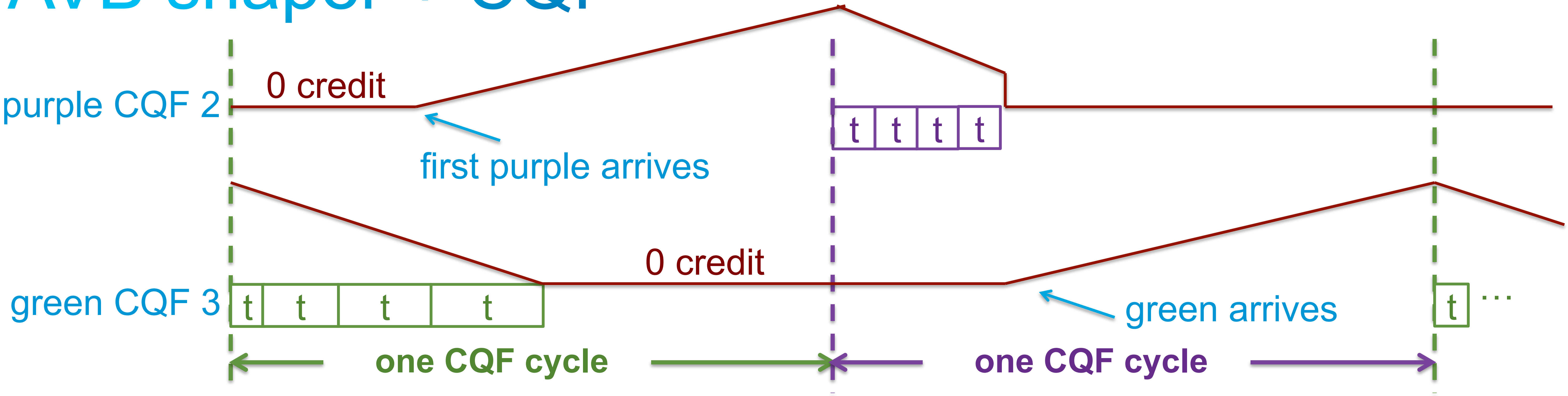
- If we **space out the CQF** traffic using AVB, then the priority 7 or lower-priority AVB traffic can get **reasonable latency**.

AVB shaper + CQF



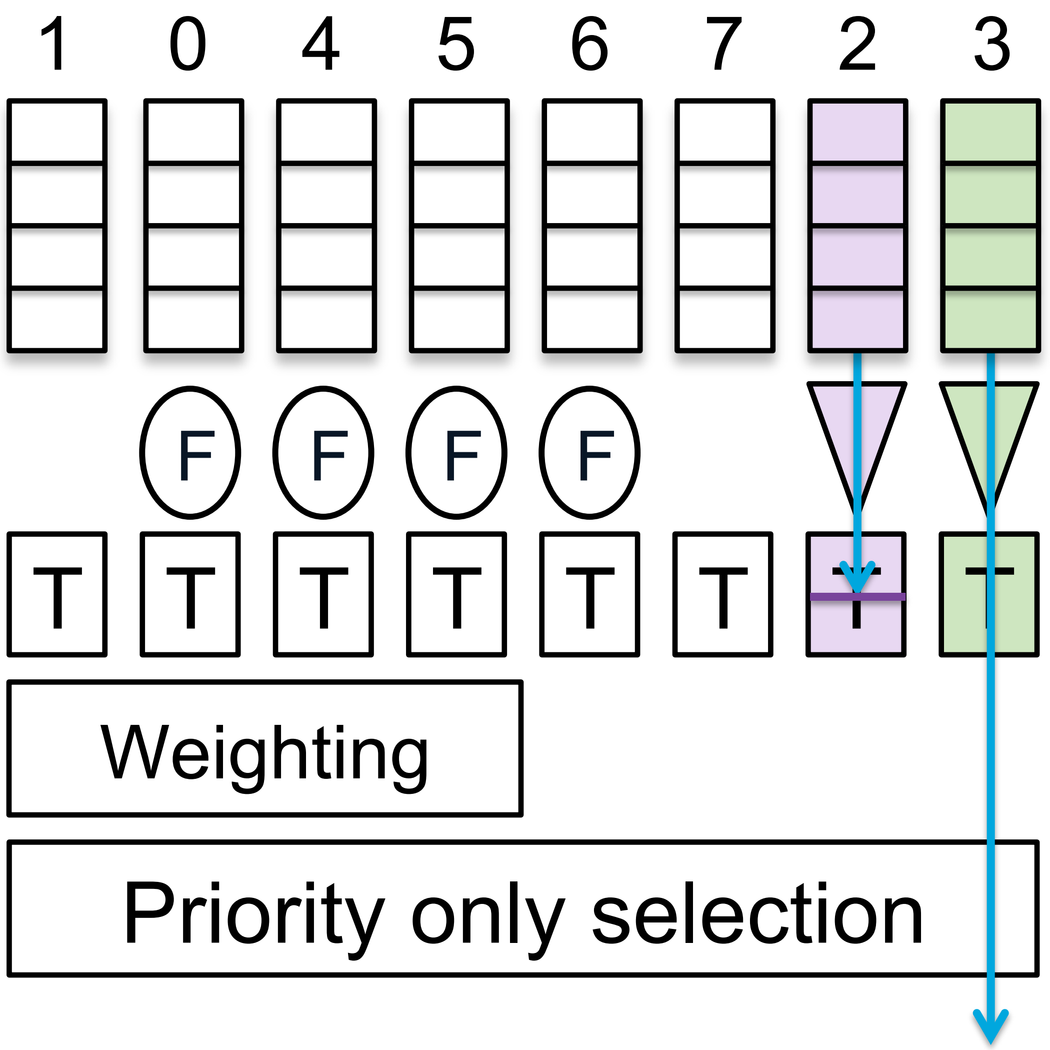
- **But, the current model, in the P802.1Qbv D2.1, cannot space out the CQF data!**
- The reason is that, while the gate is closed, the shaper above it is collecting credit.

AVB shaper + CQF



- This is **exactly** the slide we presented above, in the context of very lumpy time schedules, and said, “Don’t do that.”
- Acquiring credit while the gate is closed ensures that the CQF queue is dumped in a lump, as if the AVB shaper were not present. We’re always operating in the credit >= 0 regime.

CQF + AVB freeze

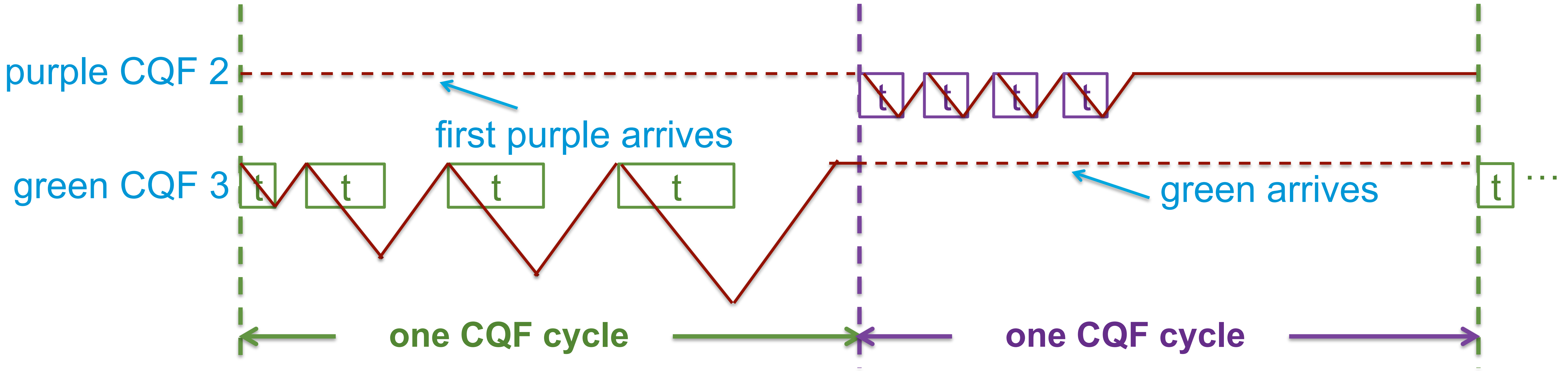


- **One way** to accomplish “Make CQF nice to other mechanisms” is to say that:

  When the time gate closes, the **AVB credit freezes**; that is, it is held at a constant value until the gate re-opens.

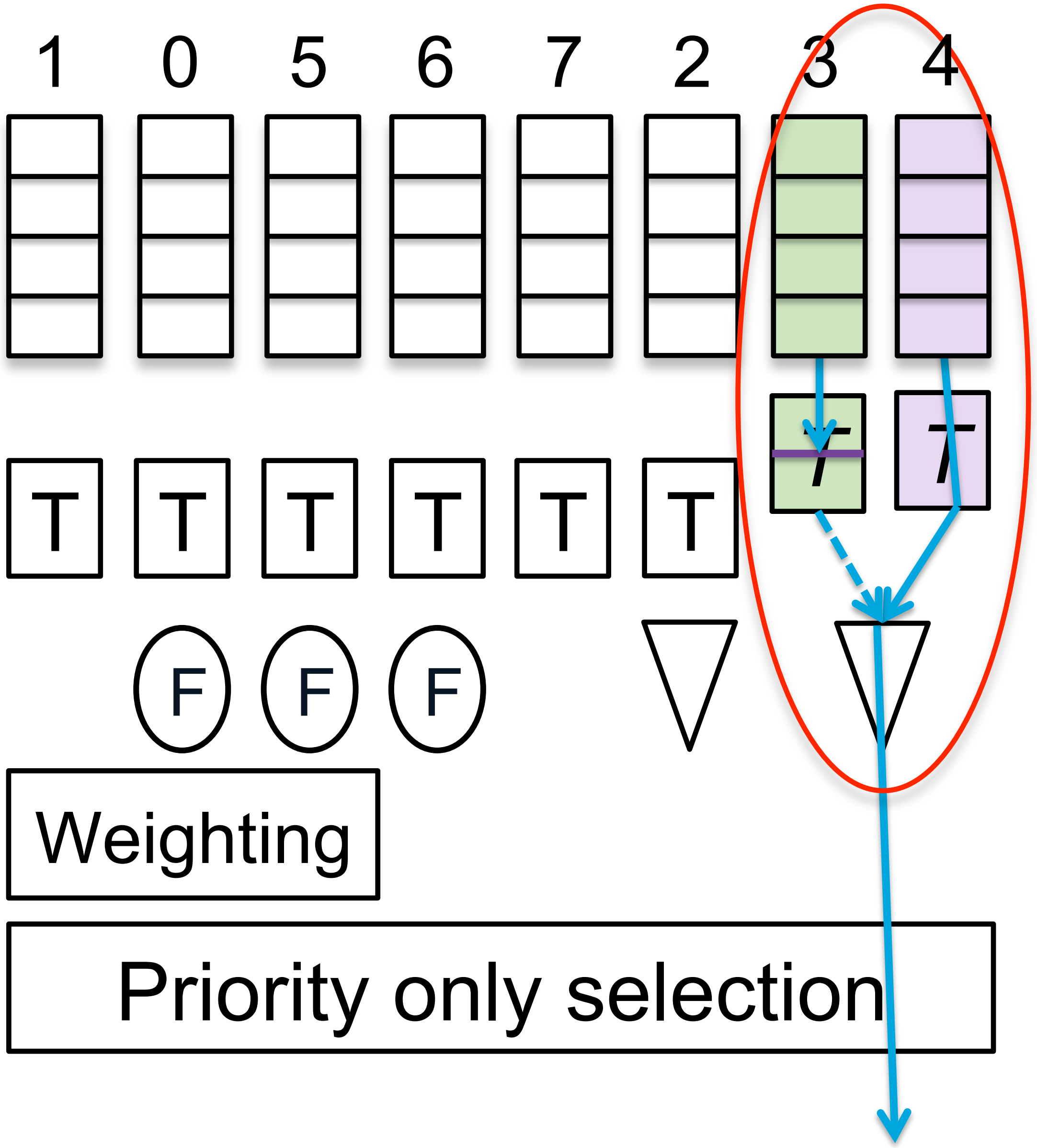
- Then, the two shapers alternate draining their queues.
- Note that each shaper is configured for exactly the current total bandwidth of the CQF traffic.

CQF + AVB freeze



- This is **exactly** the slide we presented above, in the context of very lumpy time schedules, and said, “Don’t do that.”
- Acquiring credit while the gate is closed ensures that the CQF queue is dumped in a lump, as if the AVB shaper were not present.
- We happen to be freezing a 0 credit value, but read on.

CQF + AVB shared shaper



- **Another way** to accomplish "Make CQF nice to other mechanisms" is to say that:

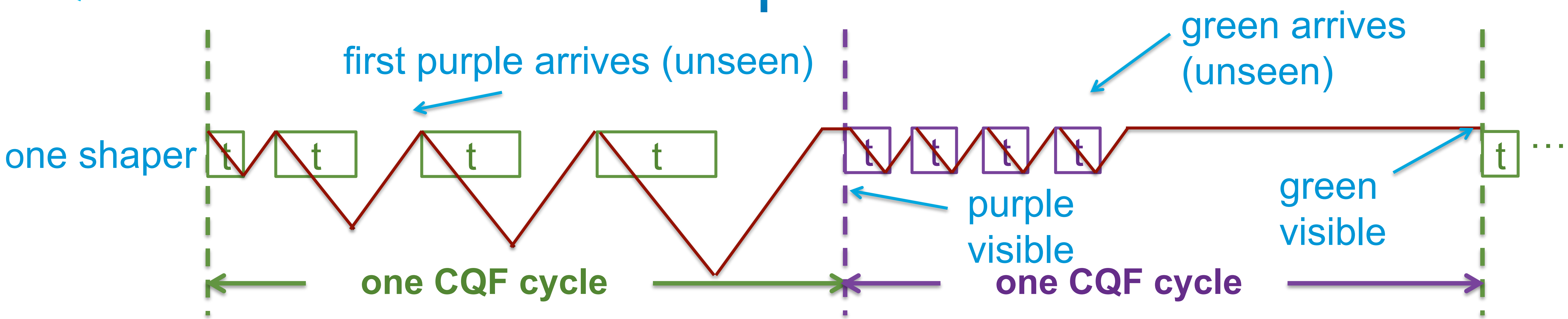
  We invert the timed gates and AVB shapers (**Q-T-A**, not **Q-A-T**).

  There are CQF time gates (*T*) that have mutual exclusion rules.

  These multiple CQF queues feed a **single shared shaper**, which works in the usual fashion, but it sees one queue at a time.

- Note that the shared shaper is configured for exactly the current total bandwidth of the CQF traffic.

CQF + AVB shared shaper

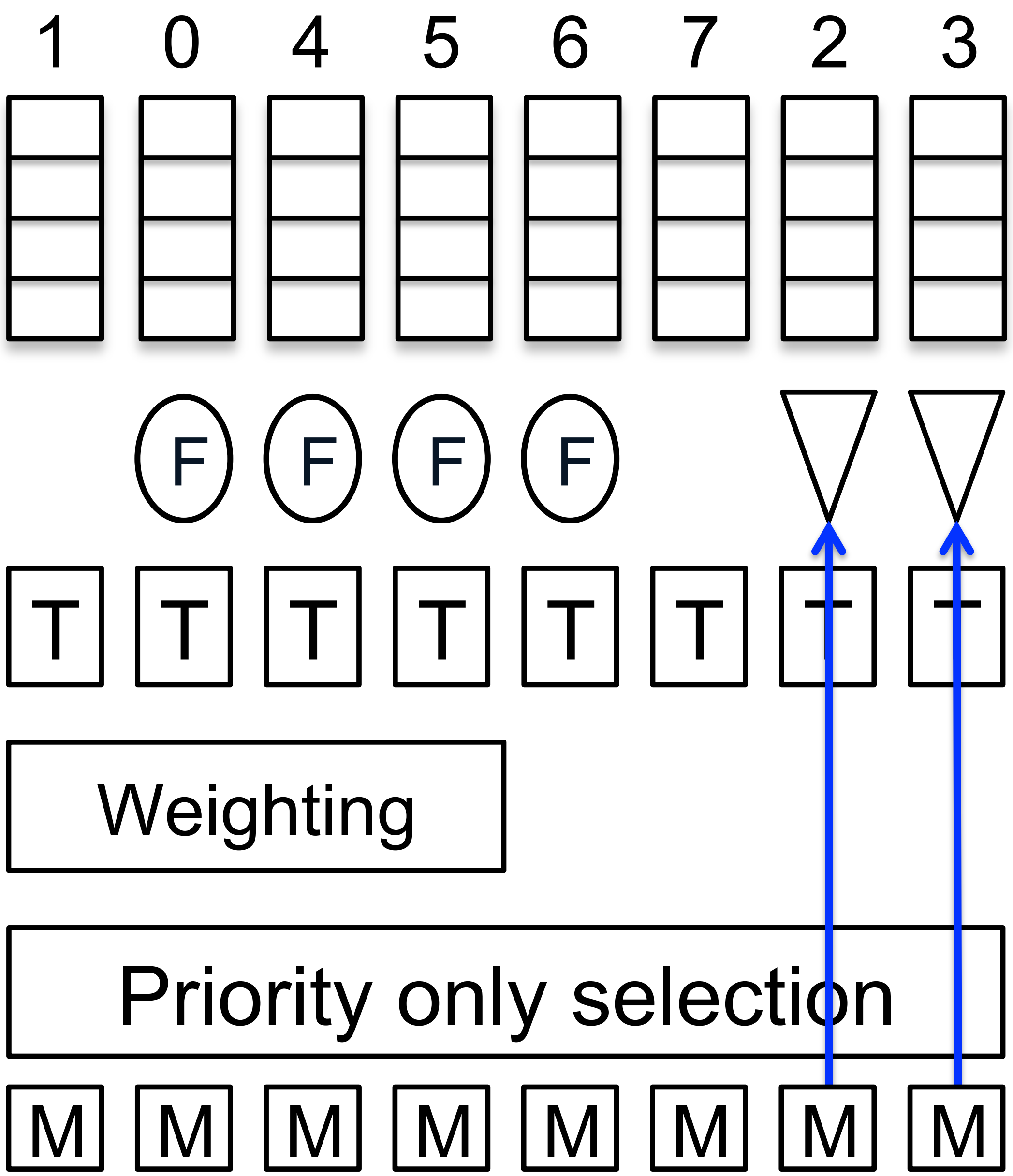


- This also solves the problem.

AVB + CQF freeze vs. shared shaper

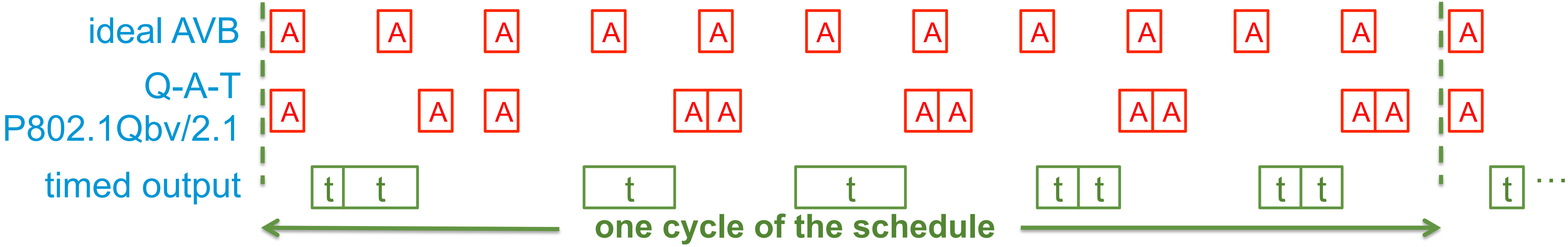
- Are these two implementations really different? **Not outside the system**.
- If CQF is working properly, then a queue will always be empty (credit = 0) at the end of a window. Therefore **the output from either model will be identical in this CQF example**.
- **We can use either or both models for CQF.**
- **But Norm!! You said the we cannot use Q-T-A. Make up your mind!!**
- The reason we couldn't use Q-T-A is that it gets confused if the gate closes with data in the queue. But in the case of CQF, that never happens.
- That doesn't mean I want to put Q-T-A in the document. It just means that, for CQF, it's not a wrong implementation, just in case that's your implementation.
- Also Don Pannell points out another possible solution: use the maximum credit parameter to limit the credit, and thus ensure gaps in the CQF transmissions. This solution is not explored in this deck.

AVB shaper and time gates



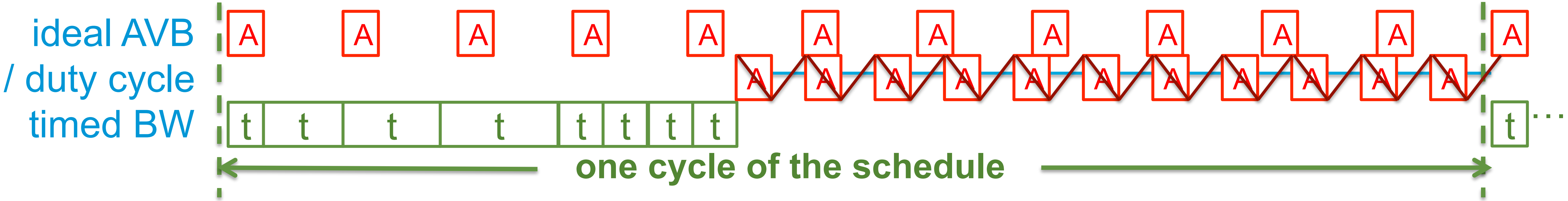
- Now, let’s look at what gets passed **up the stack**.
- Ignoring preemption for a moment, the only shaper that cares about information from the MAC (M) is the **AVB shaper**, which is modeled as using the ***transmit*** signal as an input from the lower layers.

AVB + time gates: Freeze or no.



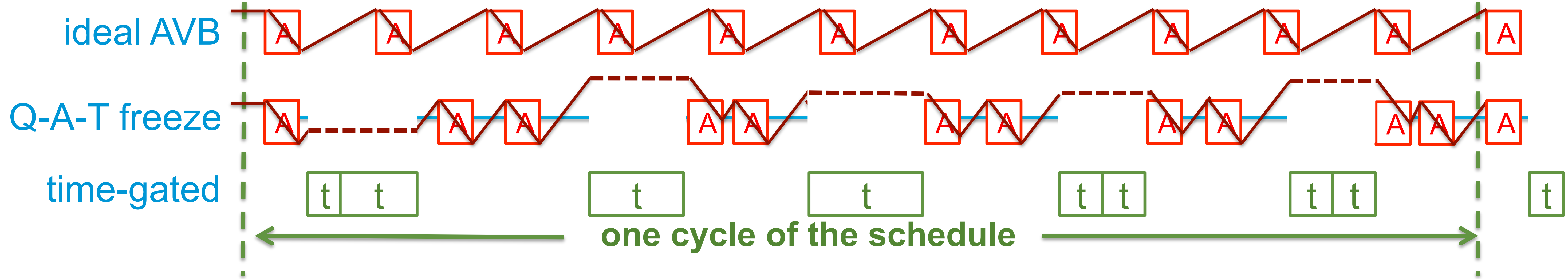
- Let’s look at the difference between P802.1Qbv D2.1 and the freeze credit proposal in the context presented before, of **time gates**, rather than CQF.
- The previous diagram for “AVB shaper + time gates” works just as before for a “nice” schedule.

AVB + time gates: credit freeze



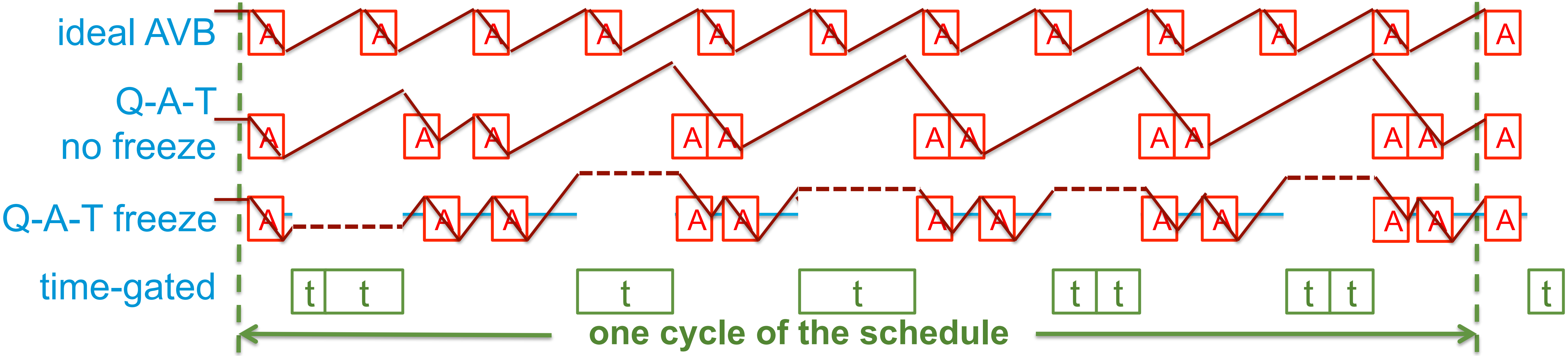
- In order to figure out how this same scenario works in for the “credit freeze” idea, we first have to figure out what value to use for the shaper’s bandwidth.
- We do this by dividing the desired BW by the duty cycle of the AVB’s time gate. This is shown graphically, above.

AVB + time gates: Q-T-A shared shaper



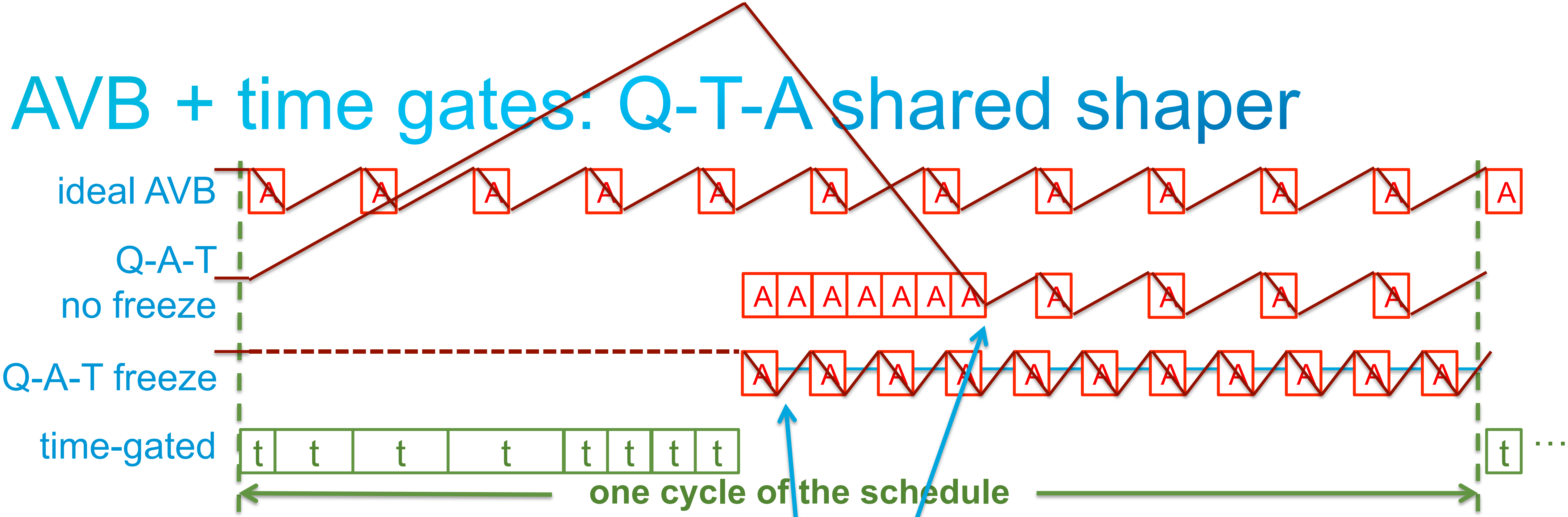
- Applying this to the example, we "freeze" the credit while the AVB shaper's time gate is closed. This is what we get, in this example.
- (Note that we are freezing non-0 credit, now, unlike the CQF case.)

AVB + time gates: Q-T-A shared shaper



- And here is everything together. The differences are not great. Mostly, the difference is that the Q-A-T freeze technique spreads the frames out a little more than the Q-T-A shared shaper, giving the Priority 7 frames better latency without sacrificing any guarantees.
- Of course, this is a “**nice**” schedule.

AVB + time gates: Q-T-A shared shaper



- This is “**nasty**” timed data. The difference now becomes significant.
- To summarize: **Freezing credit during the window closing:**
    1. Supports a **consistent Q-A-T model for CQF**, with one shaper per queue.
    2. Allows **CQF traffic to be spread out.**
    3. **Eases the pain** on priority 7 caused by lumpy schedules.

Preemption

Preemption issues

There are two questions with preemption:

- How does the method of issuing xx_UNITDATA.request primitives affect the queuing model described, above?
- How does the potential for, or act of, preemption affect the various shapers described, above?
- In making this decision, we will assume that the request primitive is issued at the moment when both the provider and the user of the service agree that it is time. There is no mechanism in the standards for this.

Request primitives

- Two basic choices have been discussed for arranging the request primitives:
  - Two SAPs, one for interruptible frames and one for express frames.
  - One SAP, and the priority parameter tells the MAC (via a table set up as a managed object) whether the frame is interruptible or express.
- Either way, it is **not** necessary for the priority selection function to offer two frames to the MAC, one interruptible and one express. That is, **the request primitive model for preemption has no effect on the transmission selection model**.
  - If there are two SAPs, then the upper layers offer the frame to the appropriate SAP until the primitive occurs.
  - If there is one SAP, then the MAC can use the priority parameter to decide when to agree to accept the primitive.

Effect of preemption on shapers

- As a reminder, the list of shapers that can be affected by preemption is:

  IEEE Std 802.1p “Strict priority”

  IEEE Std 802.1Qaz “Weighted priority”

  IEEE Std 802.1Qbb “Priority Flow Control”

  IEEE P802.1Qbv “Scheduled transmission”

  IEEE P802.1Qch “Cyclic Queuing and Forwarding”

  IEEE Std 802.1Qav “AVB shaper”

- Of these, strict priority and weighted priority seem to operate normally with any combination of interruptible and express queues. The others need to be discussed. As it turns out, only the effect of preemption on the AVB shaper presents any special issues.

Preemption and Priority-based Flow Control

- If both preemption and PFC are configured for a given queue, we know that there is no AVB shaper on the queue; AVB + PFC is disallowed.
- The effect of PFC transmissions is discussed later.
- PFC on the input side has no effect on the corresponding output queue.
- This leaves the transmit side of PFC, which is instructs the port (by means of a received PFC frame) to suspend transmission from a particular queue for a certain number of bit times.
- Since the minimum preemption fragment is the same size as the minimum frame, this author sees no way in which enabling both preemption and PFC on the same output port can make the receiver's buffering problems any worse. **There appears to be no problem with preemption plus PFC.**
- **(Unless the implementer lets PFC stall a non-first fragment.)**

Preemption and Time-scheduled transmissions

- Since time-scheduled transmissions are one of the primary justifications for preemption, this interaction has been discussed thoroughly. As a reminder:

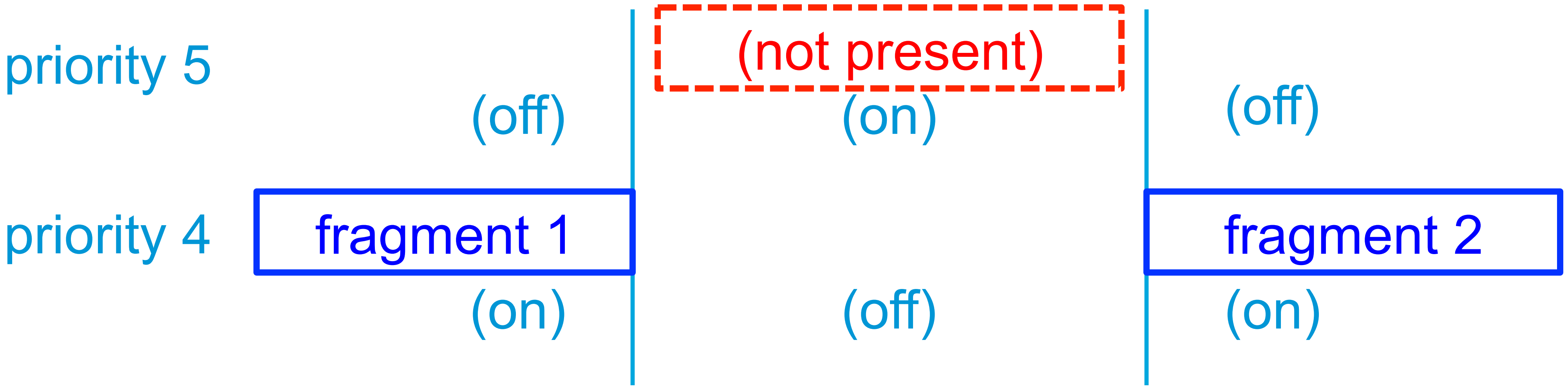
  The transmission of a frame or fragment **shall not** extend past the closing of a time gate.

  Preemption fragments have the same minimum size as ordinary frames.

  Each time a frame is interrupted, there is a "tax" of 24 octets, composed of the additional Frame Check Sequence, Inter-Frame Gap, and Preamble required to terminate one fragment and restart the next, in addition (of course) the length of the express frame(s) transmitted between the interrupted fragments. **Charging this tax to the right account is one of the primary effects of preemption on other features.**

  Every **priority value** is, to the IEEE 802.3 MAC, either interruptible or express. Given the rules so far (priority selects queue, interruptible/express values cannot be mixed on one queue), this results, at least so far, in each **queue** being interruptible or express.

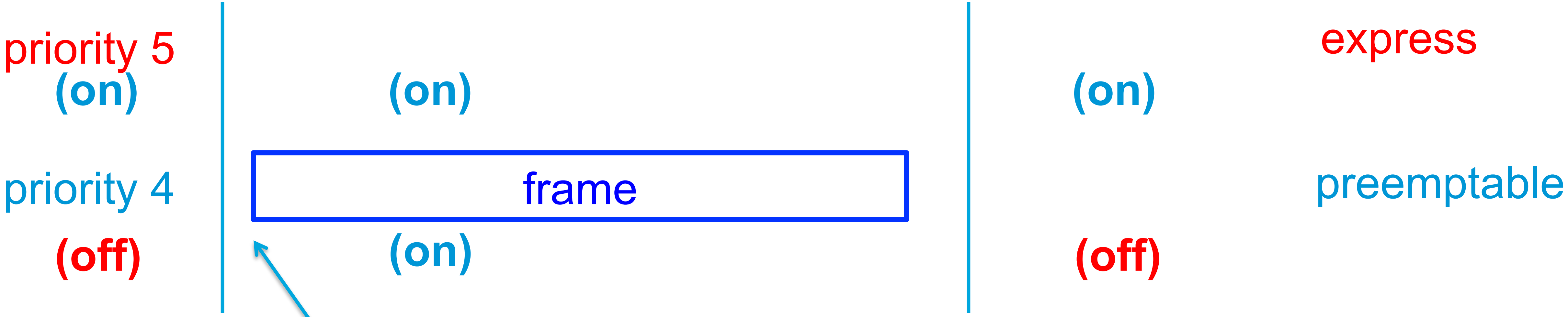
Preemption and Time-scheduled transmissions



- That does not mean there are no issues to bring up:
- A transmission cannot continue past the end of a window. But, can a frame be preempted by the end of a window, even if there is no express frame to preempt it? According to the current P802.1Qbv draft 2.1, the answer is “**Yes, it can.**” The HOLD signal handles this. So, we’re OK.

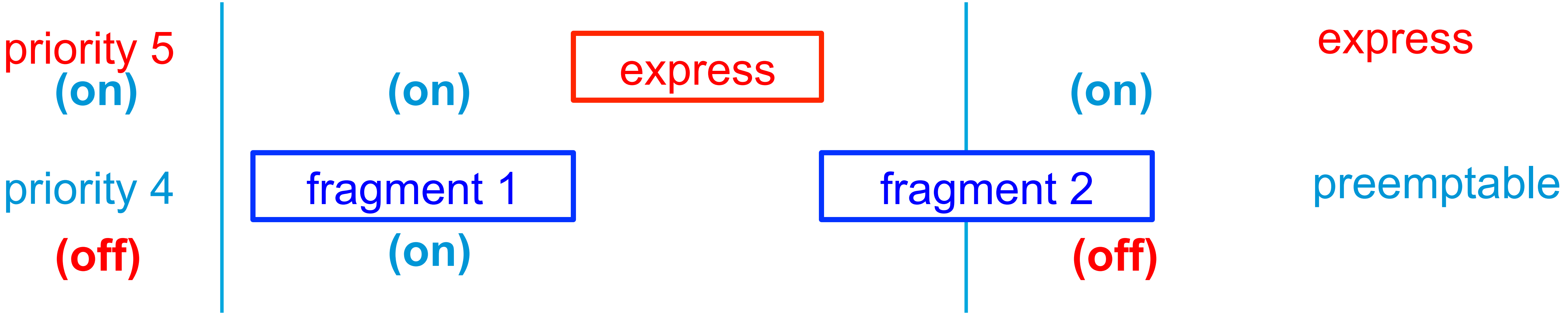
  Use case: I am opening a window for a possible transmission that must have the lowest possible latency – not even the preemption latency is acceptable.

Preemption and Time-scheduled transmissions



- Priority 4 is preemptable, and is constricted to a time window.
- Priority 5 is express, and always on.
- HOLD is not used.
- The frame **starts transmission**, because it will finish before the end of the window.

Preemption and Time-scheduled transmissions



- But, before we finish, an express frame interrupts the priority 4 frame.
- The priority 4 frame runs over the end of its window.
- The fundamental problem is that **we do not know, at UNITDATA.request time, how long it will take to transmit the frame**.
- Therefore, **we must modify the shall not transmit past the end of the window rule**.

Preemption and Time-scheduled transmissions

- Notice that we have the **same situation with cut-through frames**; we cannot know the length of the frame at the time we initiate transmission.
- What do we do? We could:

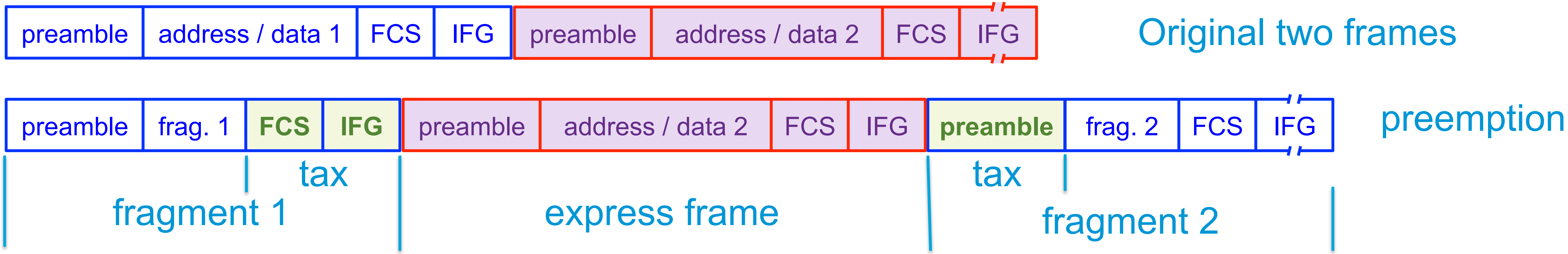
1. Rework the primitives to figure out how to make this impossible to specify.
2. Accept this, and **have an alarm that is raised when this situation occurs** in practice.
3. Accept this, and **cut off** the offending portion of the packet (to some accuracy).

- Unless/until someone comes up with an alternative formulation, I would favor raising an alarm, perhaps with an option of a cut-off (that's hard).
- We should also **modify the "shall" clause to state the exceptions**, cut-through and preemption.

Preemption and CQF

- The essential requirement for CQF is that a buffer is emptied before the start of the dead time (for transmission delay, forwarding delay, and time sync slop) at the end of each transmission cycle.
- It is unlikely, therefore, that one would give a CQF queue less importance in the priority selection than an unshaped queue.
- Making a CQF queue interruptible by preemption adds nothing to this scenario; it is perfectly permissible to do this, as long as the superior queues that can preempt the CQF queue are limited, either by time schedule or an AVB shaper, to leave enough bandwidth to empty the CQF queue in time. Therefore, we can say:
- **Preemption has no special effect on CQF,** although preemption may affect an AVB shaper attached to a CQF queue.

Preemption and the AVB shaper



- When a frame is interrupted by an express frame, for interruption, there is a **24-octet "tax"** to pay comprising an additional **FCS**, a minimum-length **IFG**, and a **preamble** that would not have been necessary if the two frames had not been interspersed.
- This tax is not included in the MSRP or AVB queue mechanisms in IEEE 802.1Q-2014, of course.

Preemption and the AVB shaper

Points to consider:

- There is a limit to the number of times a frame can be preempted, set by the minimum fragment size. However, this limit is (neglecting some corner cases) int((original frame size)/60) – 1, which makes the total worst-case tax on a 1522-octet frame 24*24 = 576 octets, which is not inconsiderable.
- The likelihood of interruptions can often not be predicted.
- If an AVB-shaped queue is interruptible, the tax cannot be counted against the configured bandwidth for that queue, because it would result in the queue being emptied much too fast in the event that no interruptions take place.
- Therefore, **the preemption tax must be charged against the worst-case interference that the interruptible shaper can impose upon the next-lower priority queue.**

Preemption and the AVB shaper

- There is another, related point: The current AVB shaper uses the notion of *transmit* signal, which indicates that a frame is being transmitted.
- Assuming that the “tax” argument, above, is correct, this simply means that **the *transmit* signal is false while the “tax” octets and the express frame are being transmitted**.
- Note that this is there is no need to “freeze” the credit during the interruption. Note also that, although the AVB shaper’s credit could rise above 0 during the preemption, a new frame cannot be initiated, because the MAC cannot accept the request primitive.

Miscellaneous interactions

Special transmissions

There are several sources of transmitted frames that, depending on how their features are implemented, can cause trouble if they bypass the queue draining mechanisms:

- IEEE Std 802.3X pause frames
- Priority-based Flow Control transmissions
- Congestion Notification PDUs
- Transmissions from the LLC “pants pocket”
- IEEE 802.1X or other frames on the uncontrolled port (assuming that the bridge relay is on the controlled port).
- Connectivity Fault Management frames from “down” MIPs/MEPs.

Special transmissions

Some of these are no problem

- IEEE Std 802.3X Pause frames are deprecated. Just don't use them.
- Congestion Notification PDUs are generated on a port and transmitted towards the relay, not towards the MAC/PHY. They pass through the queues normally on the way out the output port, so there is no problem.
- We have weasel words in 802.1Q mentioning that most all frames, including LLC frames, uncontrolled port frames, and CFM frames, should pass through the queuing mechanism in a good implementation. We may want to strengthen these words.

Special transmissions

Priority-based Flow Control **transmissions** are a **real problem**, however:

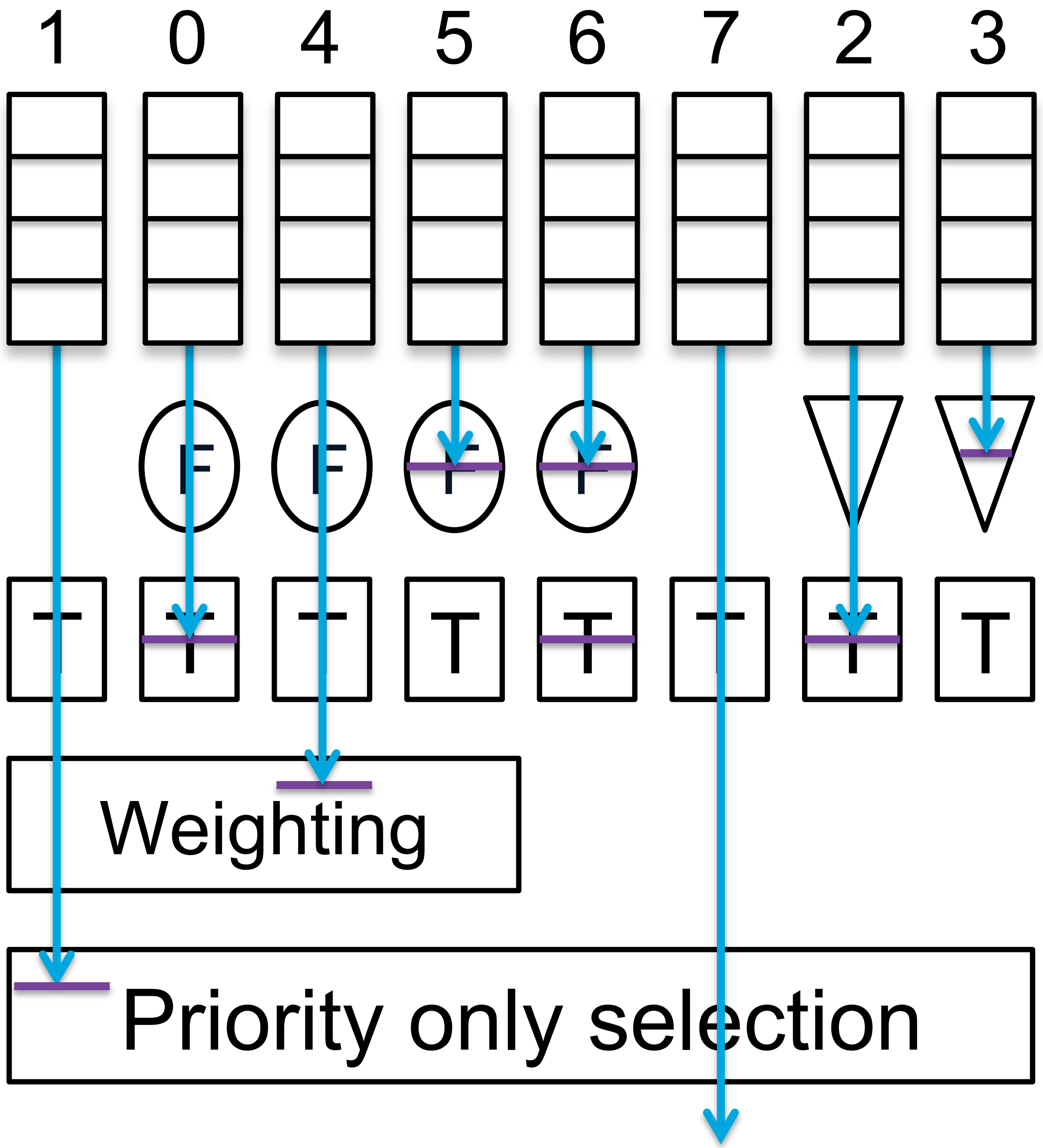
- There is no limit to the volume of PFC transmissions; it up to implementation to figure out how to avoid sending so many that they defeat their own purpose (to prevent congestion loss).
- They are very time-sensitive; they must be transmitted quickly to be of use.
- They belong to no queue; one PFC frame carries all queues' PFC states. So, the fact that **PFC shaping** cannot be configured on the same queue as AVB does not alter the fact that **PFC transmissions** can interfere with AVB.

Solution (?):

- **If** an implementation can enforce a maximum rate of PFC transmissions, it can make PFC transmission, in effect if not fact, the highest-priority AVB queue.
- Or, just **don't use PFC at all** if AVB or CQF is configured.

Summary

A (proposed) model for shaper interactions



- This model works, and minimizes inter-feature confusion, thus simplifying the text of 802.1Q.
- It allows different ways to actually implement CQF (Q-A-T or Q-T-A).

Summary: Recommendations and issues

(Comments will be provided on draft documents that point to these.)

1. **1Qbv:** A **diagram** much like the one on the previous page should be added to IEEE Std 802.1Q in order to make clear the relationships among the various shapers, along with associated disentangling of the text.
2. **1Qbv:** Compatibility of CQF with other methods will be greatly improved if we **freeze the credit of the AVB shaper when the time gate is closed**.
3. **1Qbv: Add exceptions** for cut-through and preemption to the **"shall not go past the end gate" rule**.
4. **1Qbv:** Add the ability to raise an **alarm** for a **transmit-past-the-gate event**.

Summary: Recommendations and issues

(Comments will be provided on draft documents that point to these.)

5. **1Qbu:** Make it clear that **PFC affects the start** of transmission of a frame only; it has no effect on the transmission of a non-first fragment.
6. **1Qbu:** When the AVB shaper is used with preemption, **the *transmit* signal should be false while the “tax” octets and express frame are transmitted**.

Summary: Recommendations and issues

(Comments will be provided on draft documents that point to these.)

7. **1Qcc: MSRP must charge the preemption tax** as part of the worst-case interference that the interruptible AVB shaper can impose upon the next-lower priority queue.
8. **1Qcc:** Lumps in the AVB shaper output caused by time-scheduled gate closures **impact the latency of a stream twice** – once directly by the gate events, and again in the next hop, caused by latency variation. **This item requires further study.**

Summary: Recommendations and issues

(Comments will be provided on draft documents that point to these.)

9. **1Q??:** Either an implementation must calculate its **worst-case for PFC transmissions**, and include them as the equivalent of a highest-priority AVB queue (or a scheduled queue), or it must prohibit PFC transmissions when the AVB or CQF shapers are used.
10. **1Q??:** We should make it more clear in 802.1Q that the transmissions from the LLC "baggy pants pocket" **need to pass through the queues**.

Thank you.