Scheduled queues, UBS, CFQ, and Input Gates

Norman FInn
Cisco Systems

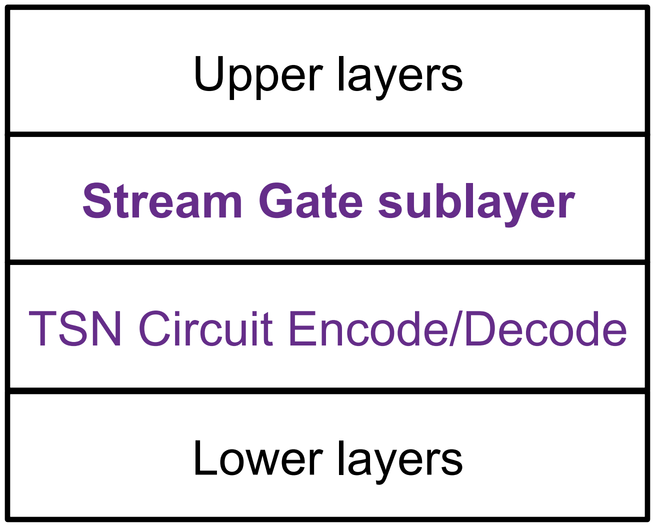
October 28, 2014

Issues addressed in this contribution

1. Building a robust network has always required “Access Lists,” meaning (to this author) the ability to perform special actions on frames that match specified criteria such as VLANs, addresses, EtherType, IP protocol, etc. Given the added importance of robustness in Time-Sensitive Networks, and the need to protect good data flows from malfunctioning end stations, perhaps it is time we defined some very basic input controls.
2. Given that only a few time-gated queues are available, it becomes difficult to control a large number of circuits, especially if some require tightly-controlled jitter.
3. Interestingly, the solution(s) suggested, here, to the above issues also happens to offer one way to describe and/or implement Cyclical Queuing and Forwarding

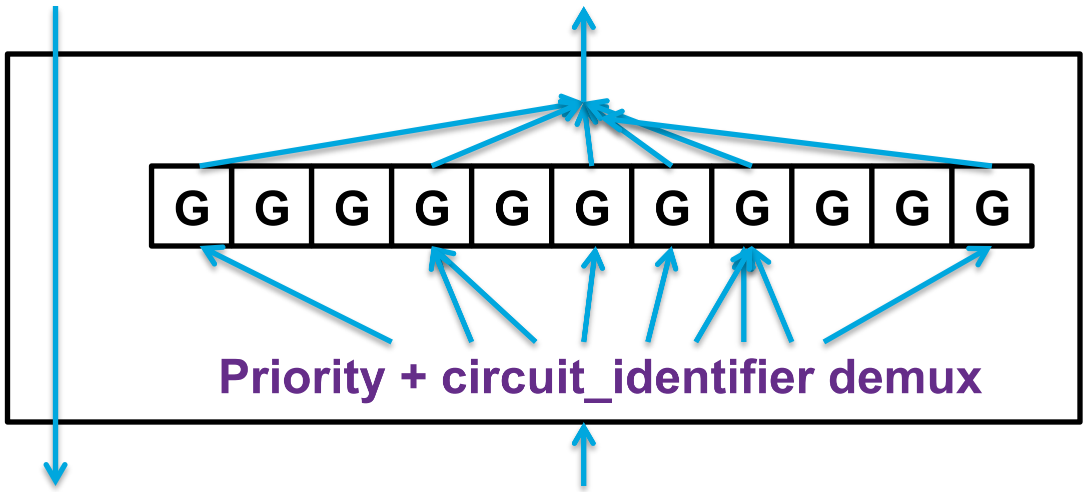
Input gates

Stream Gate sublayer in stack



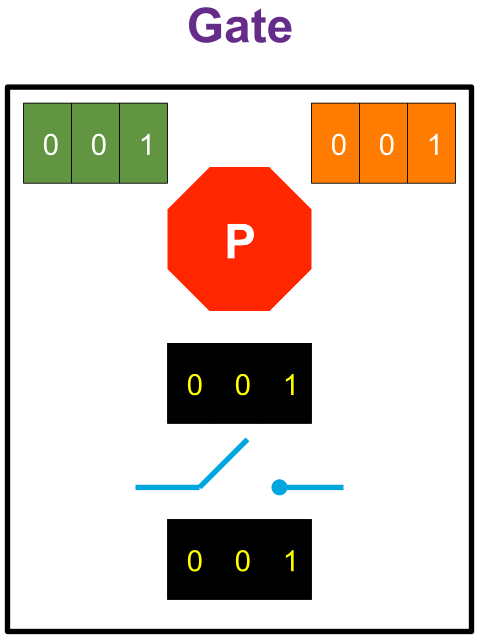
- Operates on **inputs**, not on outputs.
- Stream Gate sublayer requires a circuit_identifier, which it gets from TSN Circuit Encode/Decode function.

Stream Gate sublayer



- The Stream Gate sublayer has some number of Gates.
- The priority and circuit_identifier select to which Gate a frame is directed.
  - Either the priority, the circuit_identifier, or both, can be wildcarded (any value matches that is not explicitly in the list).
  - Priority + circuit_identifier + wildcard = you can select specific TSN circuits, non-TSN flows, TSN flows not specifically selected, etc.

Stream Gates



- Each Gate has:
  1. A pass / don’t pass **switch**. (Optionally, always closed)
  2. An (optional) standard 802.1Q **policing** function
  3. **Counters** of frames passed, marked down, and discarded.
- The pass / don’t pass switch can optionally be controlled by a circular schedule, similar to the P802.1Qbv schedule.

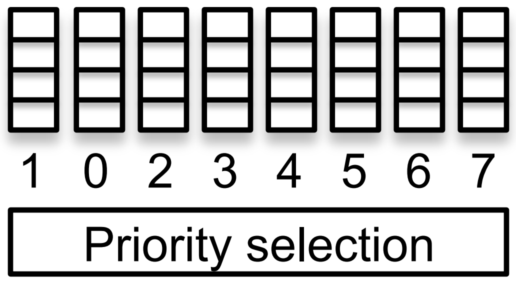
Counters and alarms

- Note that the counters, and programmable alarms associated with certain counter conditions, are important for detecting latent errors (“fool’s paradise” issues), where one leg of a redundant path fails, but the data keeps arriving on the remaining path.

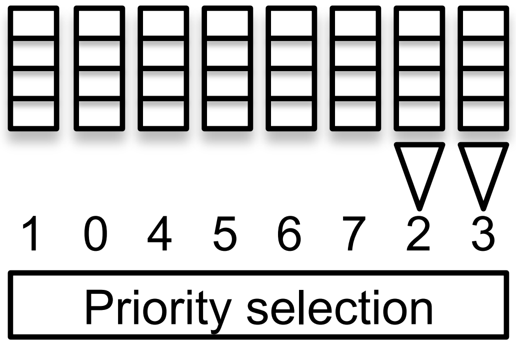
Queue selection by circuit_identifier

Priority queuing and AVB queuing

- 802.1Q: Priority (including weighted round robin)

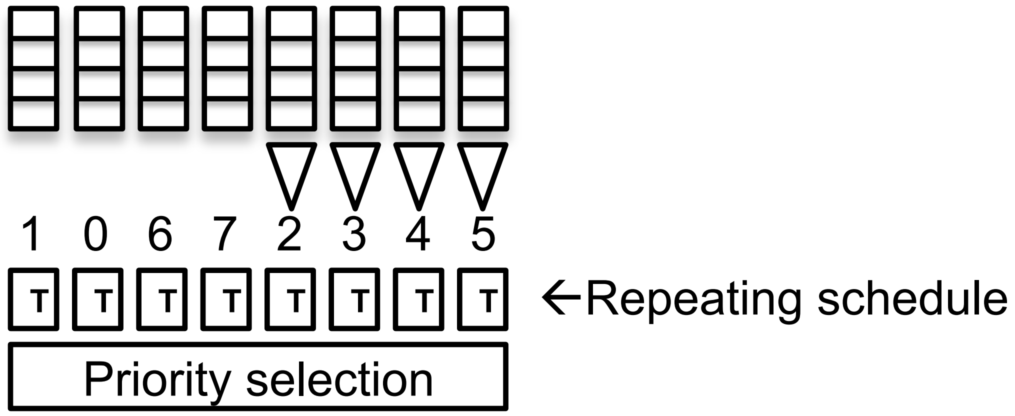


- 802.1Q-2012 (802.1Qat) adds shapers. ∇ Shaped queues have higher priority than unshaped queues, because they can still guarantee bandwidth to the highest unshaped priority.



Time-gated queues

- 802.1Qbv: A circular schedule controls a gate between each queue and the priority selection function.



- (Don’t worry about how many queues have shapers. Maybe none. Maybe most. Read on.)

Queue selection by priority

- The queue is selected by a table mapping priority to traffic class, where “traffic class” is synonymous with “queue selector”.
- An issue in industrial networks is the number of queues. Offering tightly-controlled jitter requires extra queues, so that you know what particular circuit’s frame you are gating out of the queue.
- But, there may be several circuits that need tight jitter control. These circuits may take various paths through the network that overlap In one way on one port, and another way on another port. (Flows 2+3 on one port, 1+3 on another, 1+2 on a third.) **This is very hard to do using just priorities.**

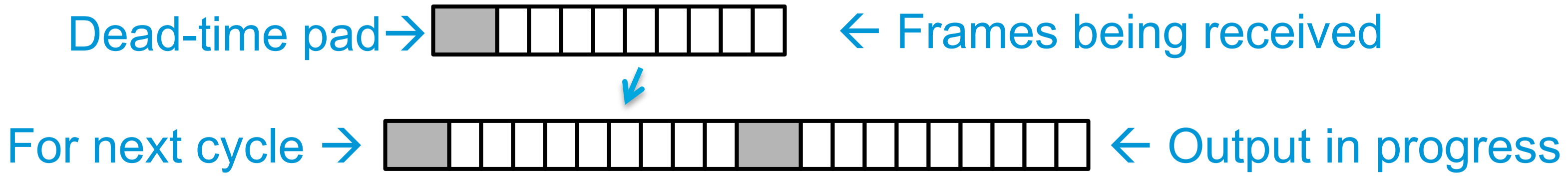
Queue selection by circuit_identifier

- If the circuit_identifier can be used in addition to the priority to determine traffic class, and thus select a queue, and especially if the arrival time of a frame for a certain circuit can be tightly controlled by input gates, then it becomes much easier to control jitter for a certain number of circuits; you can use the same queue for multiple circuits, spacing them out in time over the cycle, and know that you will not get out of synch.
- Obviously, using the circuit_identifier to select a queue is essential to Johannes Specht’s UBS.
- Not quite so obvious is that using {priority, circuit_identifier} to select a shaped queue is **all** you need for UBS.

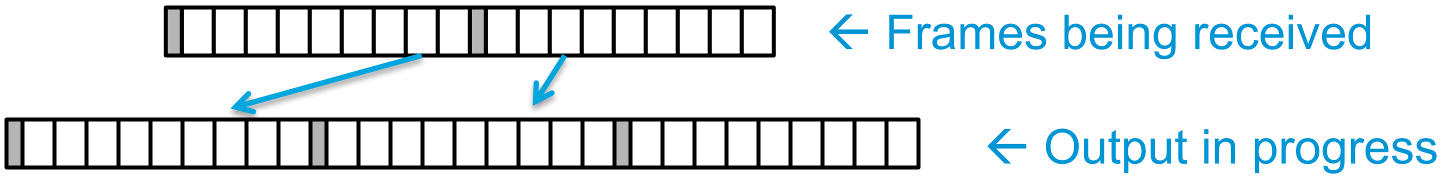
Cyclical Queuing and Forwarding

One model for CQF

- There is an issue with assigning incoming frames to the right output cycle; all frames in the same incoming cycle have to go out in the same outgoing cycle.
- Depending on the details of the length of the cycle vs. the lengths of the wires, this requires either two or three cycle-sized buffers for each output port.
  - ➢ If the wire length and bridge transit time are negligible compared to the cycle time, two cycle-buffers are sufficient.



  - ➢ Otherwise, three cycle-buffers are required.



Output queues and CQF

- Let us note that we can use credit-shaped, time-gated queues for the CQF output buffers.
- You use the time gates to alternate between the 2 (or 3) buffers.
- You use the shaper to space out the transmissions so that they do not have too large an impact on the high-priority but non-TSN traffic (such as BPDUs!), being careful to ensure that the cycle's data will always get out during the window.
- All that is needed to do CQF in addition to what we have is to use time gates to do the queue selection.
- Note that the timing for queue selection is attached to the input port, not the output port; queue selection must be synchronized with the previous hop's output.

Output queues and CQF

- And, we just talked about **input gate timing**!
- Instead of a simple on-off switch, the output of each Input Gate's schedule is a traffic class (queue selector), which varies with time. This allows CQF to be implemented without additional buffers. (Note that we also need a "no class = drop" value.)
- Note that varying what queue is selected by the frame does not change the frames priority field, at least in IEEE 802.1Q.

Summary

Summary

- The combination of:
  1. Scheduled per-circuit variation of traffic class selection; and
  2. Per-circuit policing and frame counting.
- Allows us to:
  1. Implement CFQ;
  2. Implement UBS;
  3. Provide a larger number of circuits with low jitter; and
  4. Have standards for configuring a much more robust network.
- All or part of these ideas can be incorporated either into the CFQ PAR or into a new PAR for Per-Circuit Quality of Service.

Thank you.

cisco