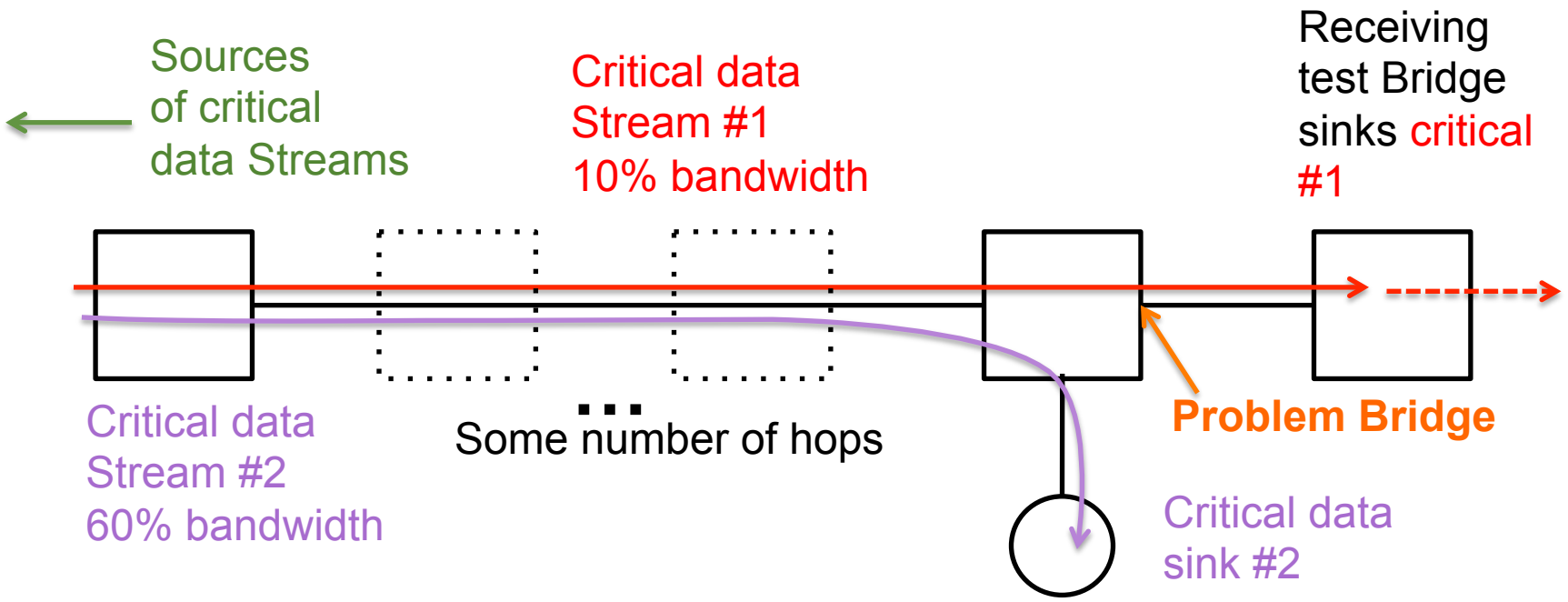
Questions about Asynchronous Traffic Shaping

Norman Finn
Cisco Systems

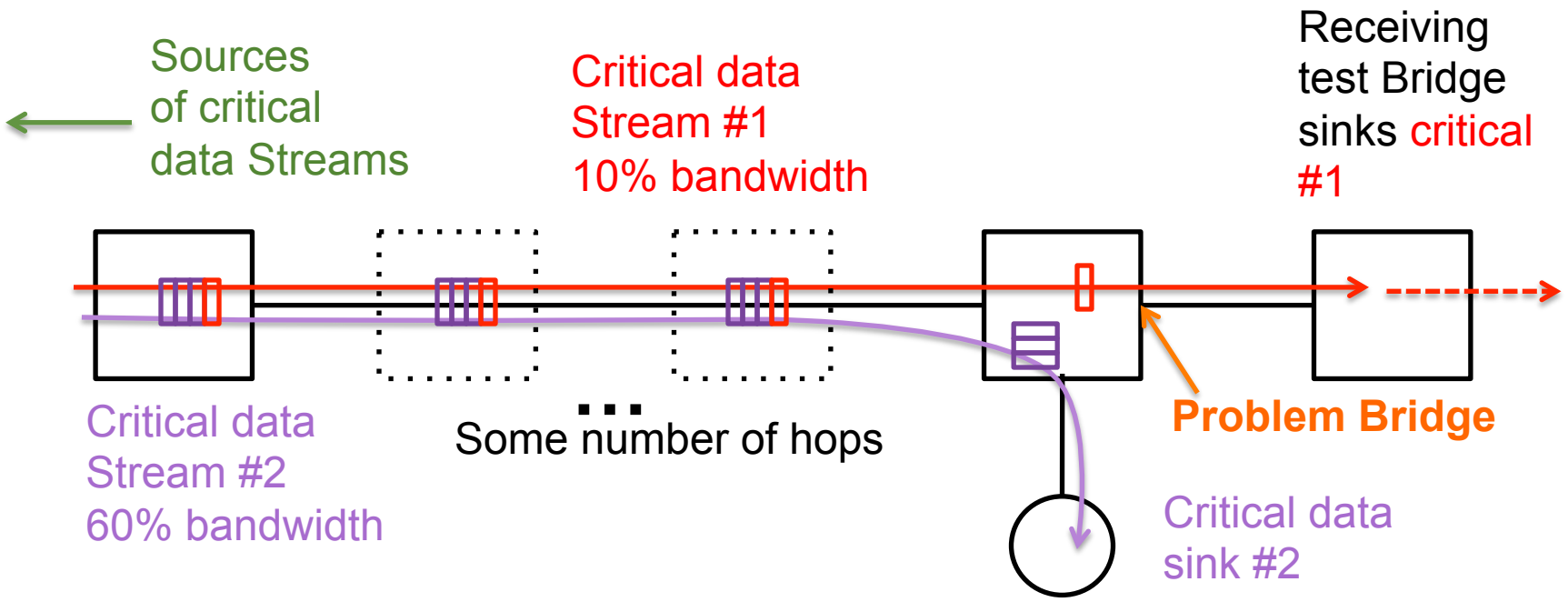
March 22, 2016

Is this a claim being made?

- Reading the P802.1Qcr PAR, it seems to claim that having one asynchronous buffer **per input port** (per output port) is sufficient to compute a maximum buffer requirement in each Bridge without reference to the overall network topology.
  - That's not totally clear, because of the words "at least" in the Scope.
  - But, that seems to be the impression held by many in the TSN TG.
- I claim that the requirement is one asynchronous buffer **per Stream** per output port to achieve independence from the network topology, and that the situation that causes the problem is easily encountered in a normal network.

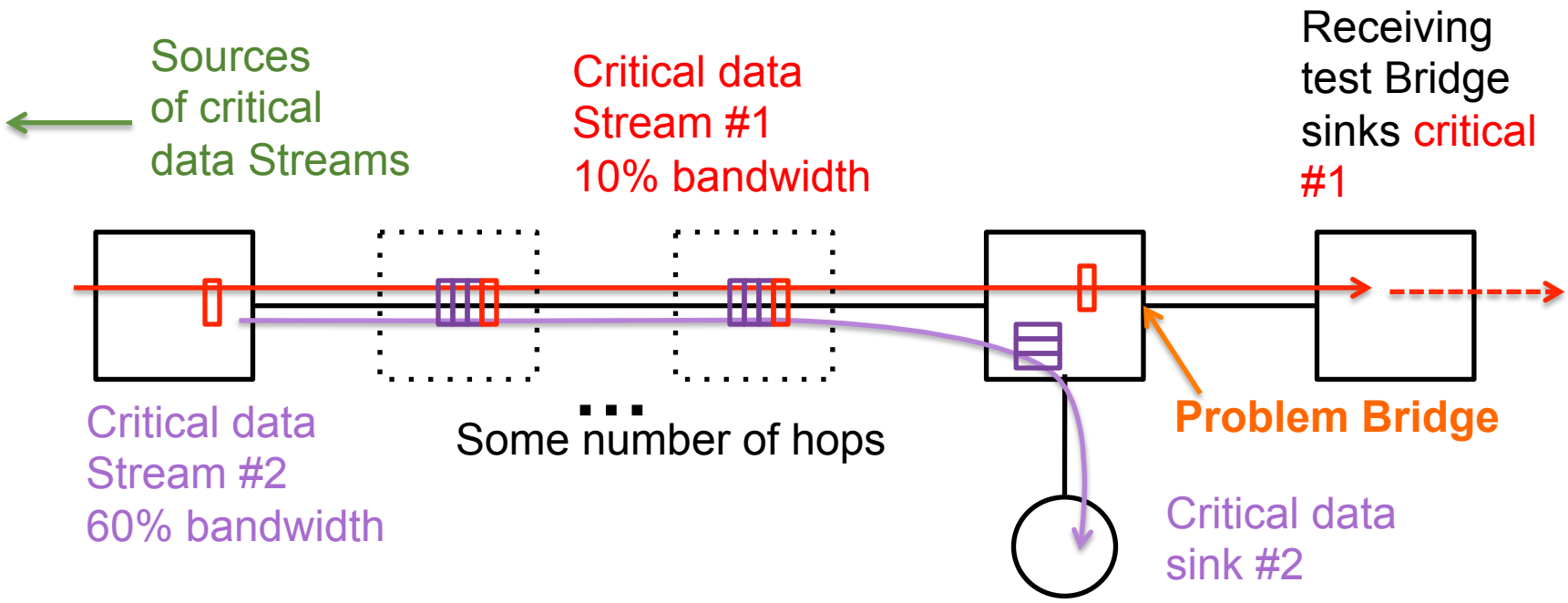


- Two critical Streams:
  - **Stream #1**: **10%** of line rate.
  - **Stream #2**: **60%** of line rate.
- Both Streams take the same route until the “Problem Bridge”.
- Both Streams use the same Asynchronous shapers until the Problem Bridge, because both take the same path.



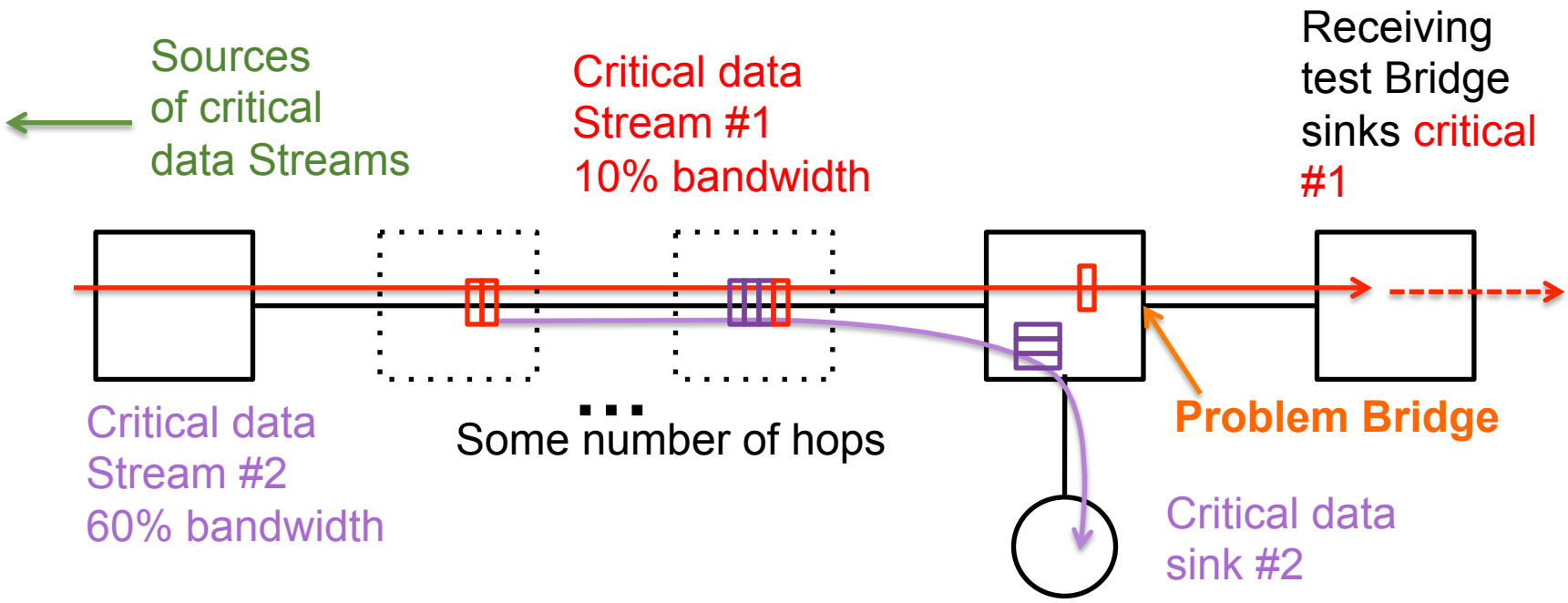
The setup:

- Queues become occupied to the expected, normal, non-0 level.
- Each asynchronous queue is partly filled with packets from Stream #1, partly filled with packets from Stream #2.
- **NOTE:** We are only looking at the asynchronous shaper queues, not the simple queues into which they dump.



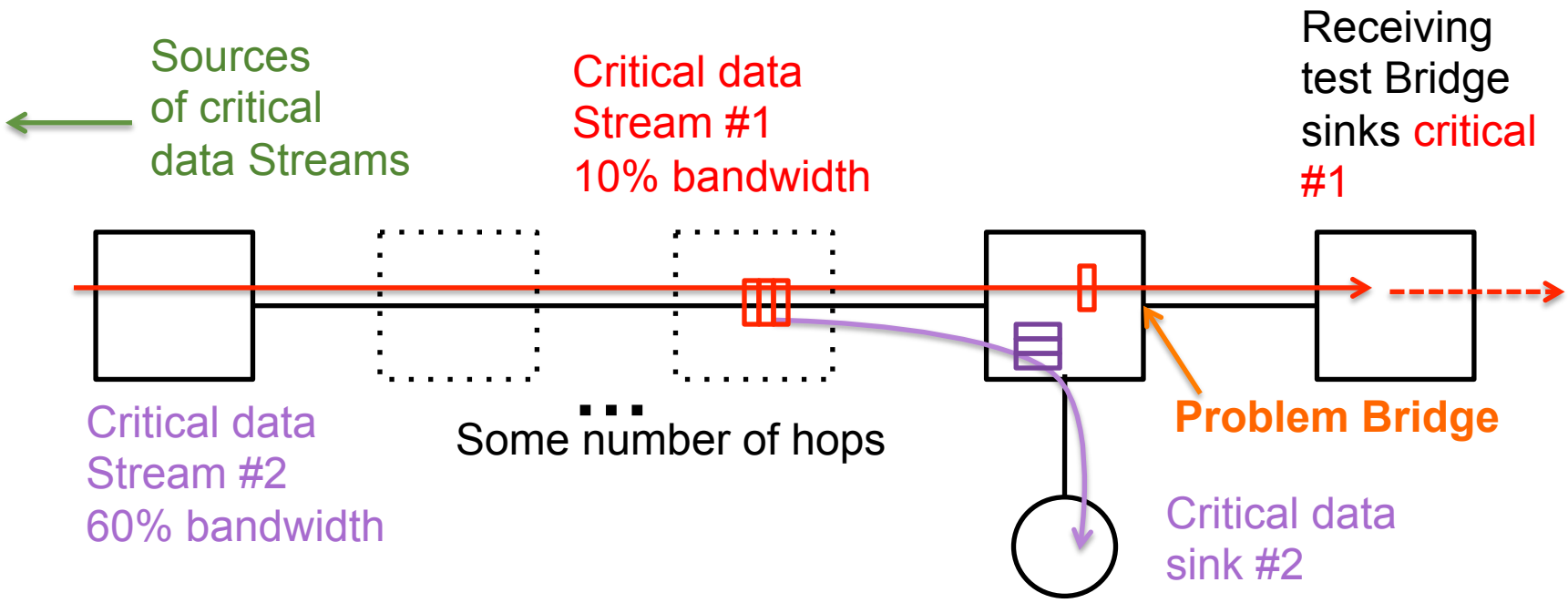
The sting:

- The source of critical Stream #2 stops transmitting.
- Critical Stream #1 starts draining at 70% line rate towards the right.



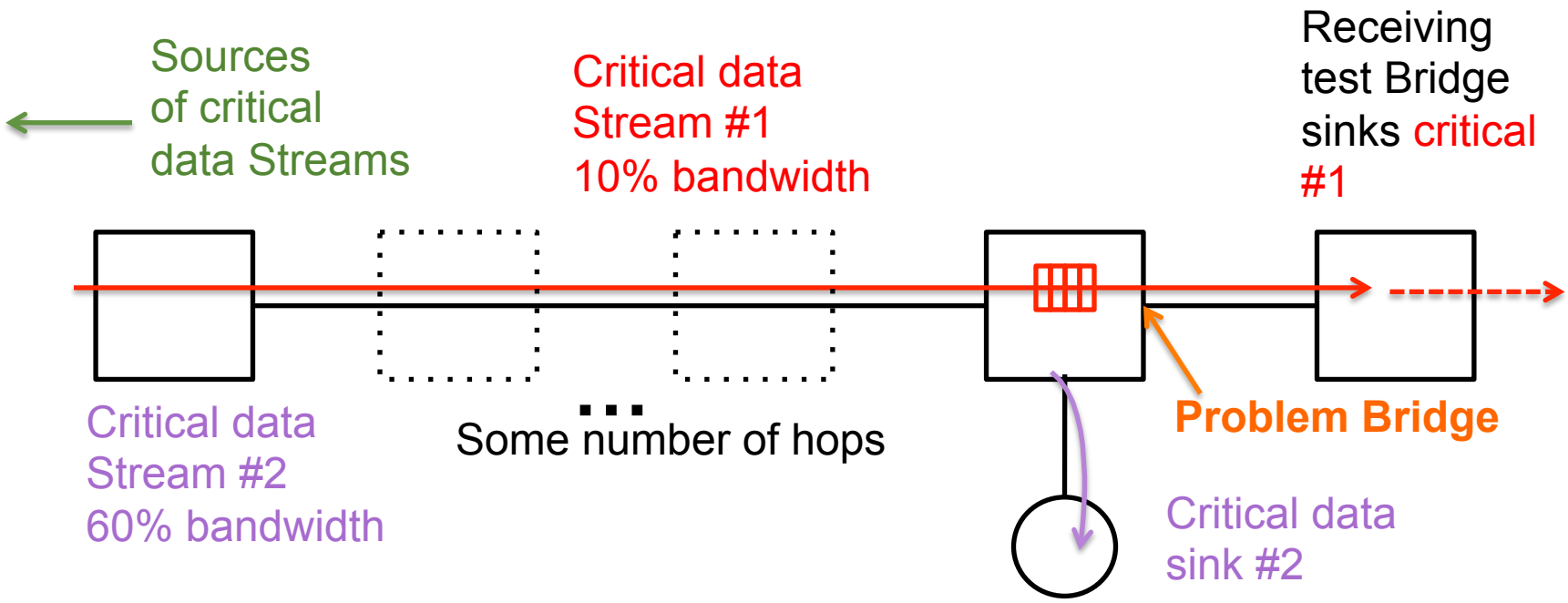
The sting:

- The source of critical Stream #2 stops transmitting.
- Critical Stream #1 starts draining at 70% line rate towards the right.



The sting:

- The source of critical Stream #2 stops transmitting.
- Critical Stream #1 starts draining at 70% line rate towards the right.



The sting:

- The source of critical Stream #2 stops transmitting.
- Critical Stream #1 starts draining at 70% line rate towards the right.
- **The asynchronous buffer in Problem Bridge is still draining at 10% line rate. It must store a number of packets proportional to the number of hops.**

Questions

1. Does this mean you can “pump” the rightmost queue until it fills, no matter how big it is?
   - Answer: “**No. It’s OK.**” If you turn Stream #2 back on, it will cause delays in Stream #1 that will allow the rightmost queue to empty before cutting off Stream #2 fills it, again.
2. Aren’t we back where we were when Christian Boiger pointed out the AVB problems?
   - It would seem so.
3. Don’t we need per-Stream asynchronous queues, instead of per-input-port asynchronous queues?
   - It would seem so. Per-Stream queues would prevent the bunch-up.

NOTE: This has been known since the DiffServ / IntServ wars of the 1990s.

Thank you.