# Seamless Reduncancy Issues

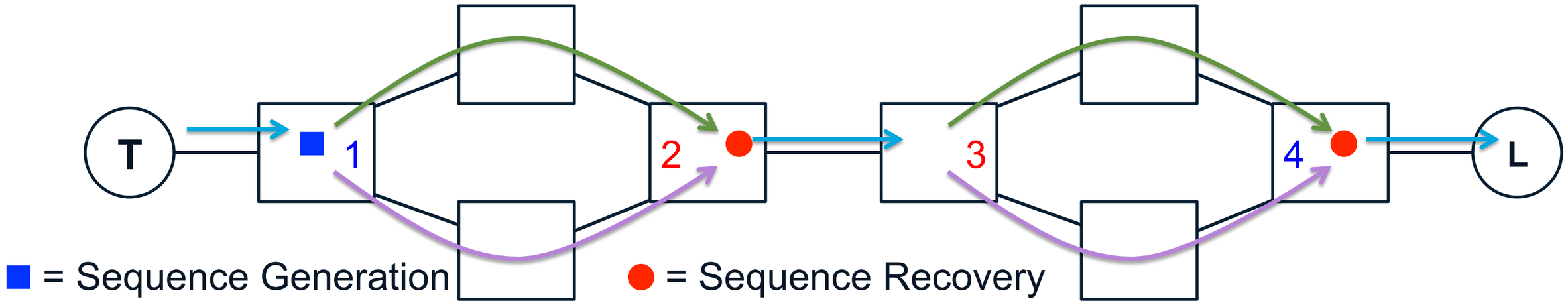
## Supporting comments on P802.1CB Draft 2.0

**Norman FInn**
Editor, P802.1CB

October 14, 2015

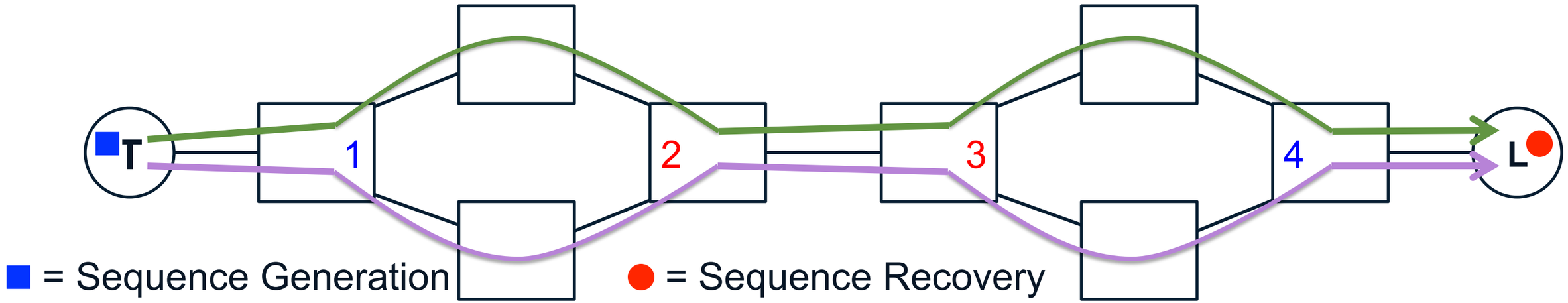
# Question 1: Stream splitting

# Seamless Redundancy in the **relay** systems




- This is **one way** to use Seamless Redundancy in a network that provides a less-than-ideal topology.
- Bridge 1 sequences and splits, Bridges 2 and 4 merge and recover, and Bridge 3 only splits (which may be simply a multicast forward operation).
- All of the SR is in the relay systems; the end systems are SR-unaware.

# Seamless Redundancy in the **end** systems




- This is **another way** to use Seamless Redundancy in that same topology.
- The Talker sequences and splits, the Listener merges and recovers, and all of the Bridges simply forward.
- All of the SR is in the end systems; the relay systems are SR-unaware.

# Both are viable, as is a mixture

- Neither use is “wrong” or “right”.  There are good reasons for either one.

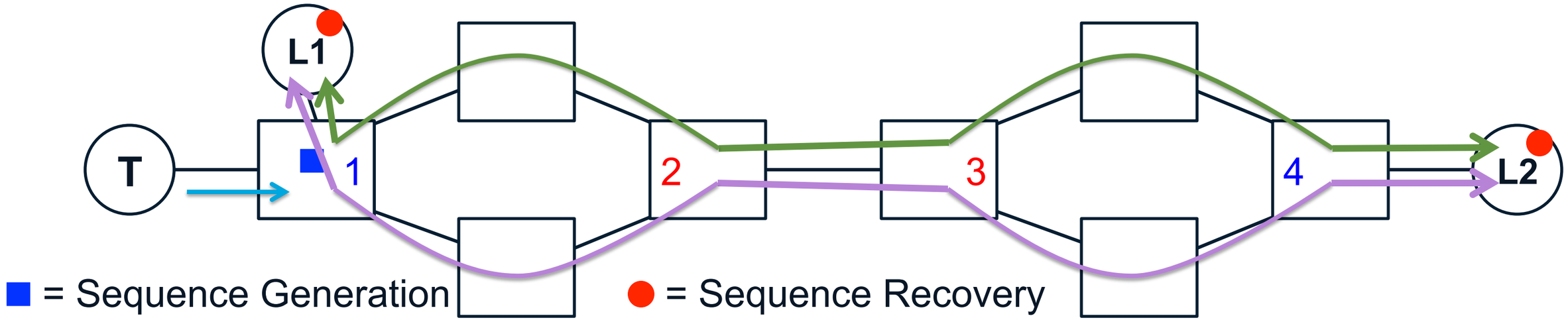
  A link’s reliability may or may not be improved by multiple time-delayed transmissions.

  Individual relay and end systems can be SR-aware or SR-unaware.

  Bandwidth can be a resource that is scarce or plentiful on different links.

- Especially in a “brownfield” environment, one may have a mixed topology, with all four combinations of {SR-aware,SR-unaware}{end,relay} systems.
- Systems that today are using existing SR solutions, e.g. IEC 62439-3 HSR/PRP, may or may not expect to receive duplicates, and complain if they do or do not receive them.

# The issue



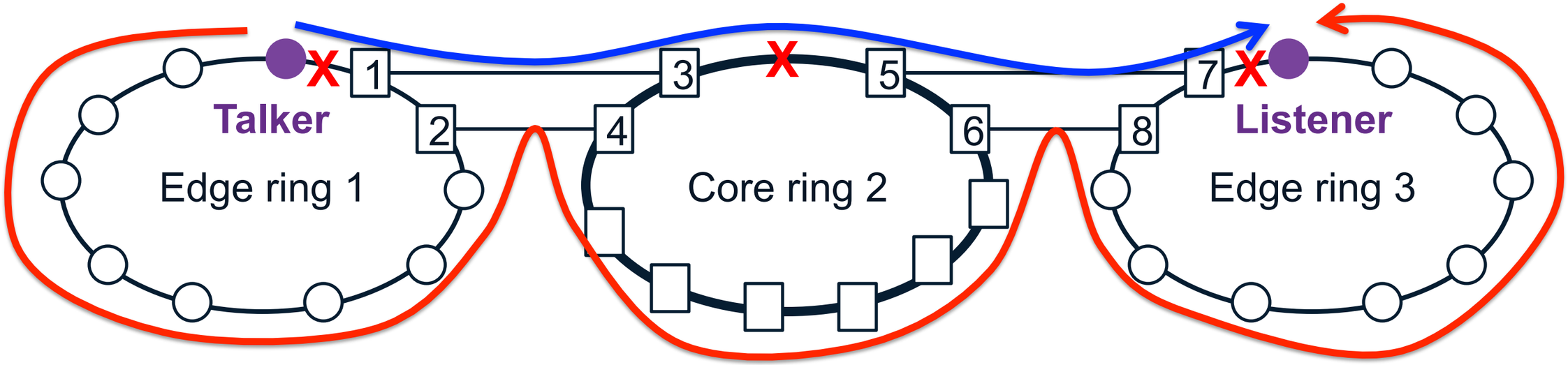

- **Mixed scenario**: Bridge 1 sequences and splits, **both Listeners** expect to merge and recover, and Bridges 2, 3, and 4 simply forward.
- SR is implemented in Bridge 1 and in both Listeners; most of the relay systems and the Talker are SR-unaware.
- **Bridge 1 cannot be configured this way in P802.1CB Draft 2.0.** (See editor's note, P802.1CB D2.0 p82 L26.)

# Proposed resolution of Question 1: Allow multiple outputs from one input.

- Allowing this configuration requires replicating two copies of one input packet, perhaps identical, perhaps different, on the same output link.
- Editor recommends that this capability be required, and that the managed objects be altered to support it.
- Working Group conclusion:

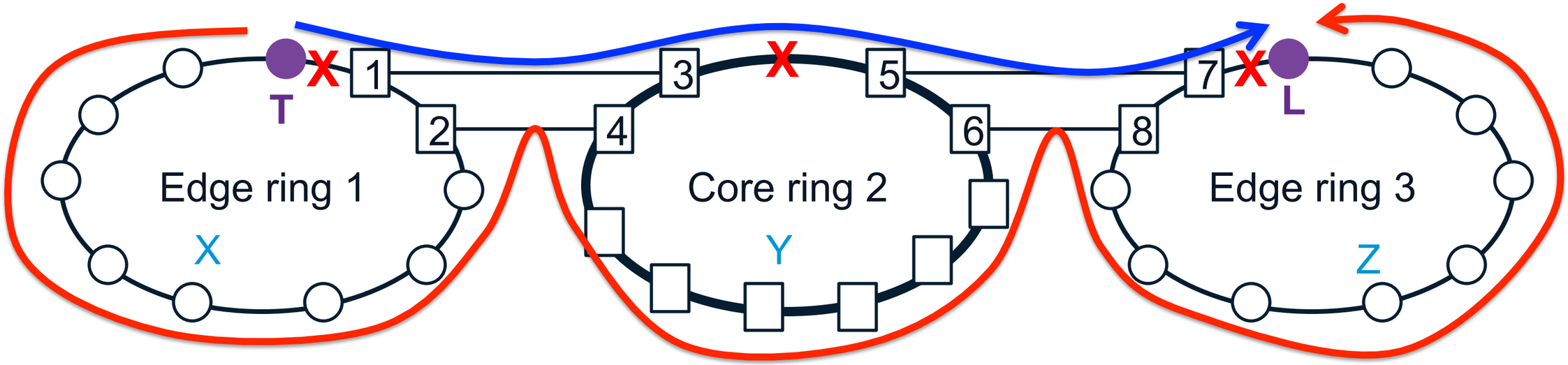
# Question 2: Delay function

# Additive delays



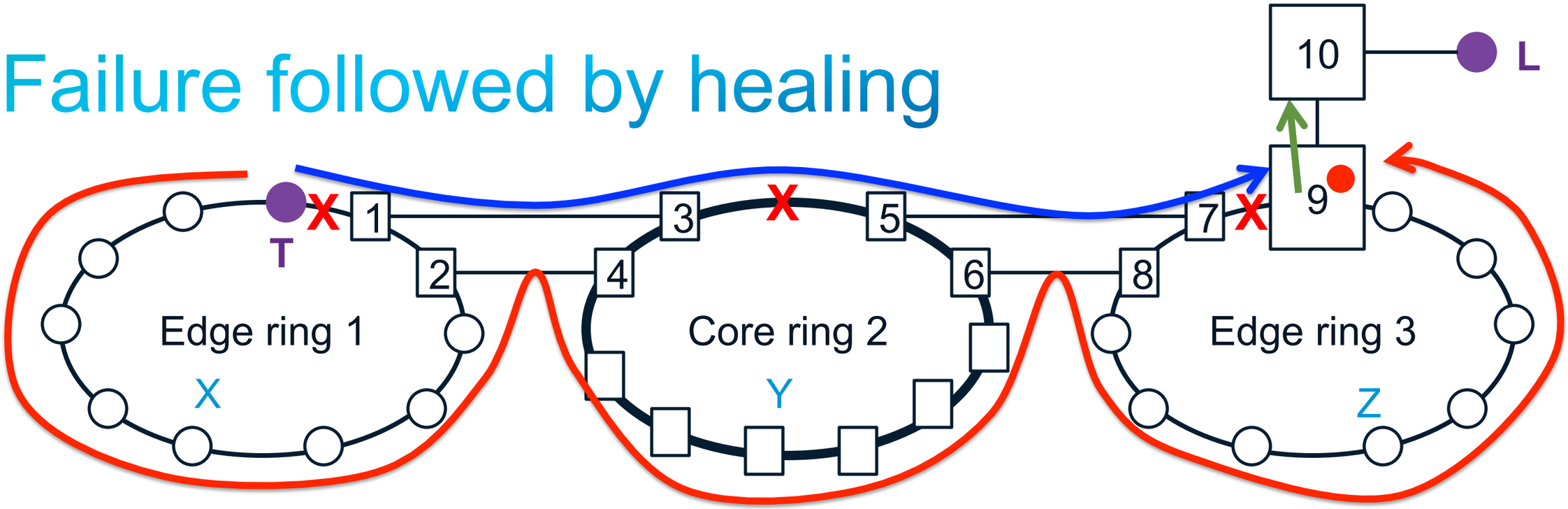

- This is a common deployment. It supports **3 link failures**,
- The **shortest path** from Talker to Listener is the one along which the Listener usually receives its packets.
- The **longest path** is used, however, if the 3 link failures are exactly the ones shown.

# Additive delays



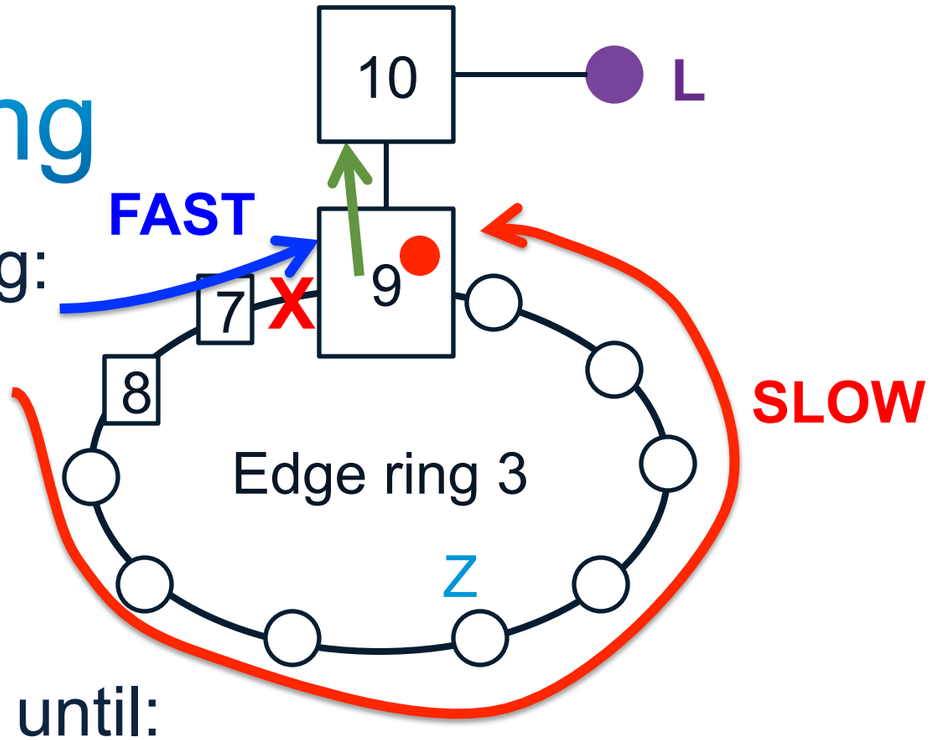

- $\Delta t_1 = t(TX2) - t(T1)$, $\Delta t_2 = t(4Y6) - t(35)$, $\Delta t_3 = t(8ZL) - t(7L)$
- Total delta Talker to Listener $\Delta t_{TL} = \Delta t_1 + \Delta t_2 + \Delta t_3$
- If we're using TSN for zero congestion loss, the **allocated buffer space in every system in Edge ring 3 must account for the total $\Delta t_T$**, in addition to the normal requirements for TSN.
- This is nothing for intermittent Streams, but it is **critical for bulk Streams**.

# Failure followed by healing




- Let us look at what happens if all three links fail, and then heal, each at the same time. Note that we have added Bridge 9, which performs sequence recovery and sends a single TSN Stream to the Listener via Bridge 10.
- (This is not a silly scenario! Human-induced network management errors and the unfortunate placement of big electric motors are two causes.)
- Remember, we are looking at bulk Streams.

# Double rate output after healing




- Network is in steady state. Bridge 9 is receiving:
  1040, 1000*, 1041, 1001*, 1042, 1002*, …,
  where the **starred packets are discarded**.
- When the **fast links fail**, Bridge 9 receives:
  5040, 5000* **(fail)**, 5001*, 5002*, …
- The discards continue with **no packets output** until:
  5038*, 5039*, 5040*, 5041, 5042, 5043, … Normal output rates resume.
- When the **fast links heal**, Bridge 9 receives:
  7998, 7999, **(heal)** 8040, 8000, 8041, 8001, 8042, 8002, …,
  and Bridge 9 **continues double-rate output (or buffering)** until:
  8078, 8038, 8079, 8039, 8080, 8040*, 8081, 8041*, 8082, 8042*, …
  Normal output rates resume.

# Just to make it worse …

- Properly sequencing the failures and healings of *n*–1 links can result in double, triple, quadruple, …, *n*-tuple burst rates for shorter periods of time.
- In the above example, quadruple rate output is possible with just the right (wrong) sequence for healing the links.

# Now let's really twist the knife …

- One could say, "I don't care about bulk Streams." But, let us observe that **Cyclic Queuing and Forwarding serves only bulk Streams**.

  Packets stop at every relay node along the path.

  Therefore, more nodes in one path = more packets in flight on that path = bulk Streams.

- CQF has no place to put **even one** packet that arrives at double the normal rate, e.g. after a **single packet loss** and later reception via the slow path. (And increasing the reservations is difficult – see the previous slide.)
- **Therefore, as currently specified, Seamless Redundancy and CQF cannot, in practice, be used together.**
- **Similarly, with AVB queuing, the combination of SR and bulk Streams means that ALL systems require enough extra buffers to handle both the AVB formulas and the (perhaps larger) SR deltas.**

# Proposed resolution of Question 2: Add Delay Function to P802.1CB

- If the given is that CQF distributes packets from input to output based on time, then the obvious solution is to introduce deliberate timed delay FIFOs at certain points in the network to equalize the path delays.
- One FIFO per group of Streams with the same path deltas is sufficient.
- These delay functions could be specified in P802.1Qch, in P802.1CB, or in some other 802.1Q amendment, but we can observe that:
  - The delay function is useful to both AVB bulk Streams and CQF Streams.
  - There is no need for a delay function if Seamless Redundancy is not employed.
  - Therefore, P802.1CB seems to be the right place for it.
- Working Group decision:

Thank you.