How many VLAN IDs are required for 802.1CB seamless redundancy?

Norman Finn
Version 2

Mar. 4, 2014

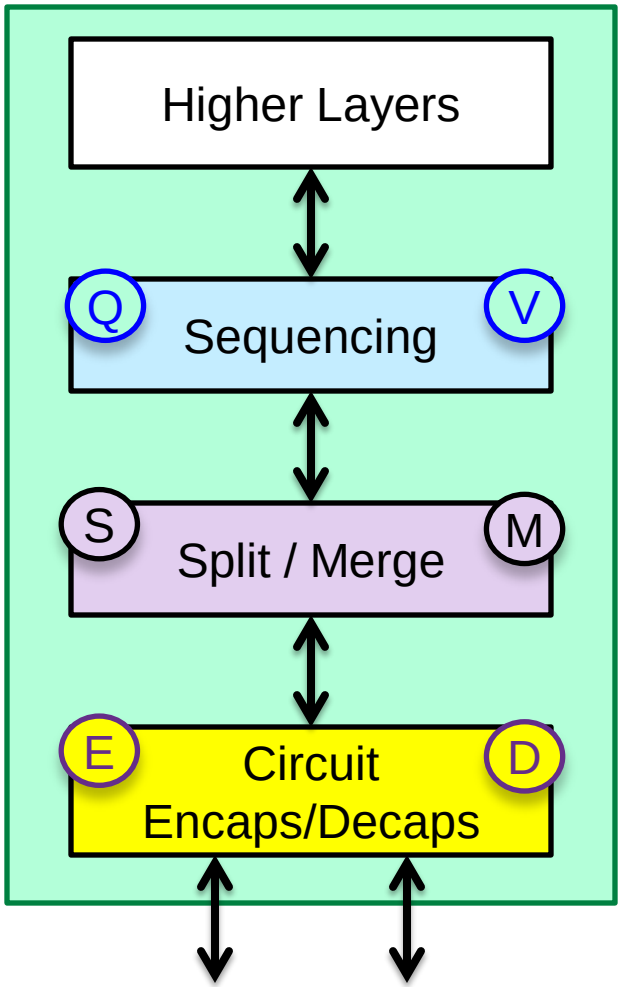
References

- This presentation is cb-nfinn-How-Many-VLANs-0214-v02.pdf.
- It is based on the layering model presented in tsn-nfinn-L2-Data-Plane-0214-v04.
- It also draws from tsn-nfinn-Day-In-The-Life-0214-v02.

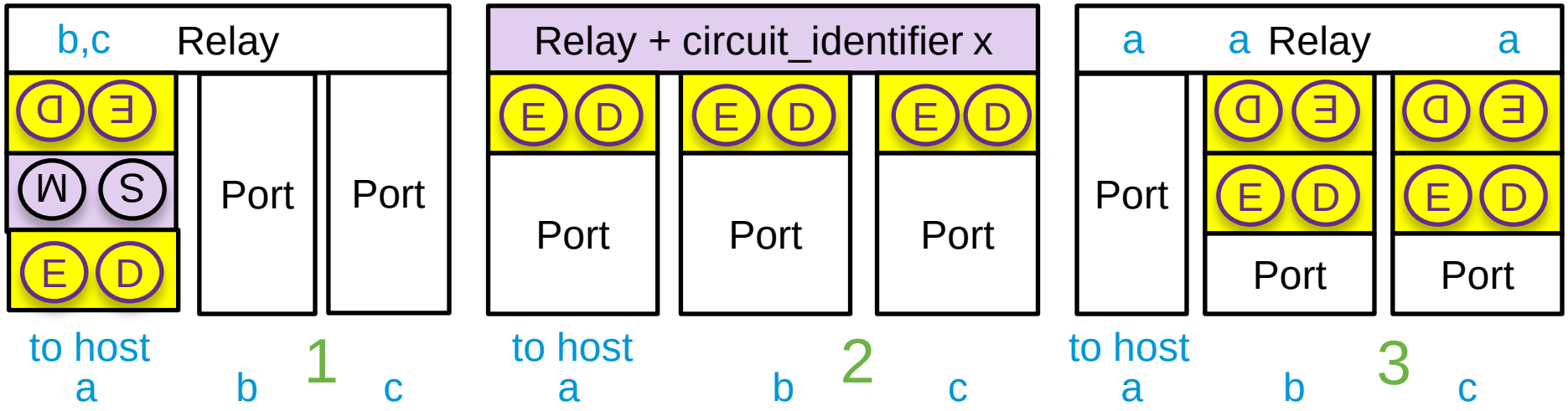
Layering

Host layering (from L2-Data-Plane)

- Higher Layers work as always.
- Sequence number generation (Q) and discard (V) can detect missing packets, discard duplicate or misordered packets, and/or correct packet ordering.
- Split (S) replicates and relabels packets' circuit_identifiers. Merge (M) only relabels.
- Circuit Encaps/Decaps (E) / (D) packs/unpacks parameters in packets.



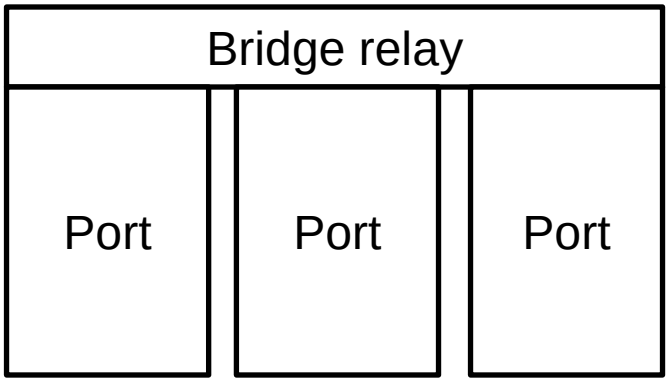
Splitting and merging circuit identifiers



- Some ways to describe splitting and merging:

1. Split/Merge replicates packets and remaps circuit_identifier **a** to **b** and **c** on the way up.
2. Relay transfers circuit_identifier **x** along with packet among independent Encaps/Decaps.
3. Per-exit-port per-flow encapsulation remap **a→b** and **a→c**.

Splitting and merging circuits



- No matter how you describe it in detail, the observables outside the bridge are the same.
  - If the different paths have **different** external encapsulations, then the bridge has to do **per-circuit remapping** of those encapsulations.
  - If all paths have the **same** external encapsulations, then the bridge does its **ordinary job**.

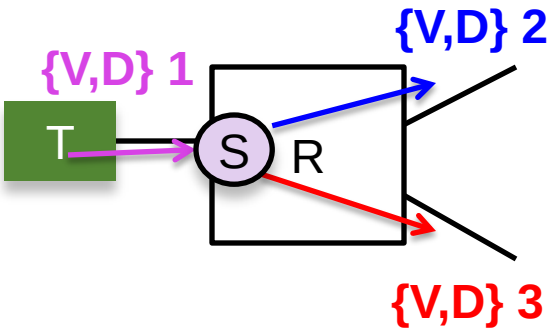
Bridge/Split/Merge layering

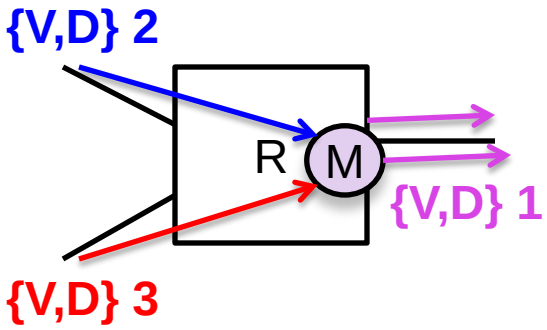
- **Choice 1**: Split/Merge is the Bridge relay, functioning as normal, so the root and N paths all have the same {VID,DA} pair (circuit label), at least within a single bridge.
- **Choice 2**: Split/Merge replicate and reconcile different {VID,DA} pairs for each circuit (root and *N* paths), and the Bridge relay functions as normal.
- **Not a choice**: A new kind of Bridge relay.

Proxy Split/Merge **Choice 1**

- Split/Merge functions (S) (M) are the normal Bridge relay function (R) .
- The flow has the **same** {VLAN ID, Destination MAC address} circuit identifier on both paths and at both ends.
- (Note: **two frames** output after the merge.)

Proxy Split/Merge **Choice 2**

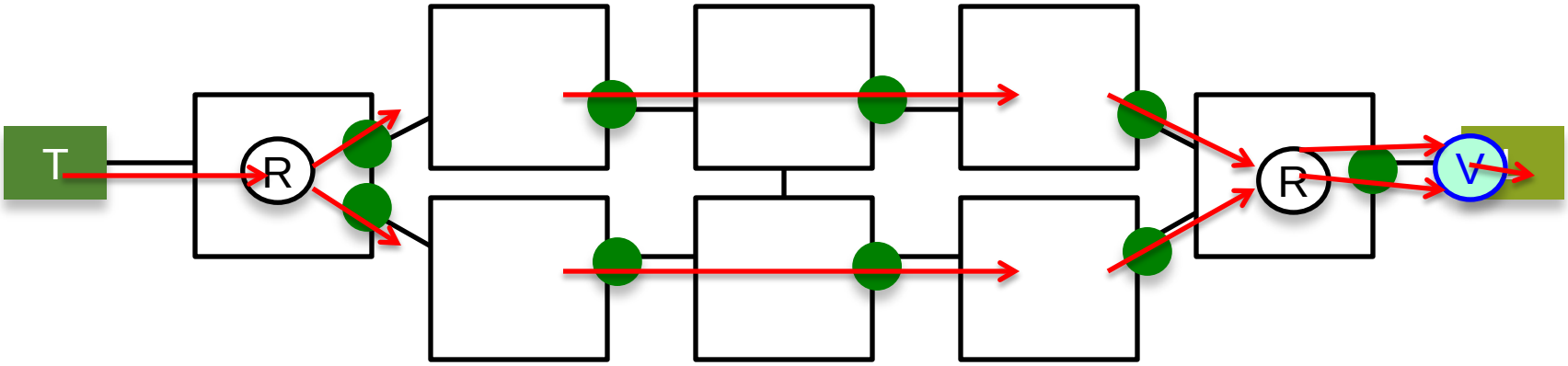




- Split functions (S) changes input {VID, DA} to different values for the two (or more) paths.
- Merge function (M) combines different circuit IDs into a third circuit ID. (Extra frames are eliminated by Sequence discard (V) .)
- Each path has a **different** {VID, DA} pair, perhaps different from the outer pairs.

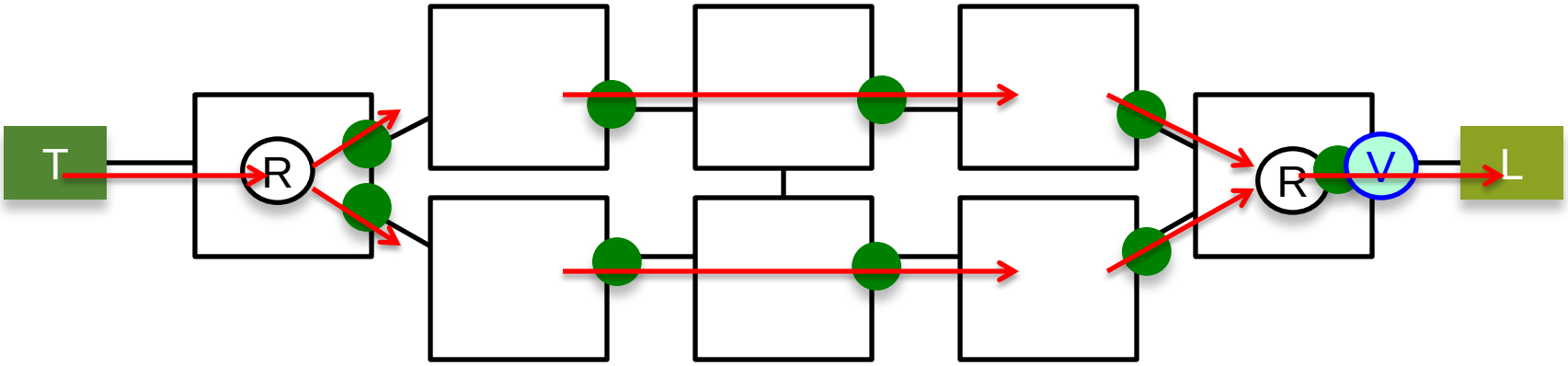
Network view

Proxy Split/Merge Choice 1



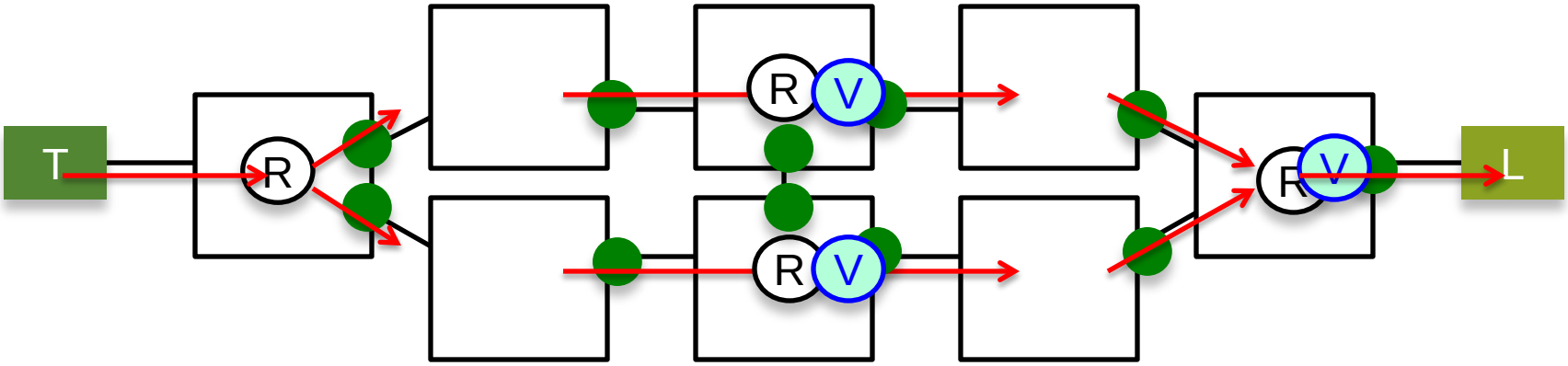
- The Split / Merge function is a relay function (R)
- Only these ports are enabled to pass the one {VID,DA} pair.
- If the Sequence Discard function (V) is in the **Listener**, then the Listener typically receives **two** copies of every frame.

Proxy Split/Merge Choice 1



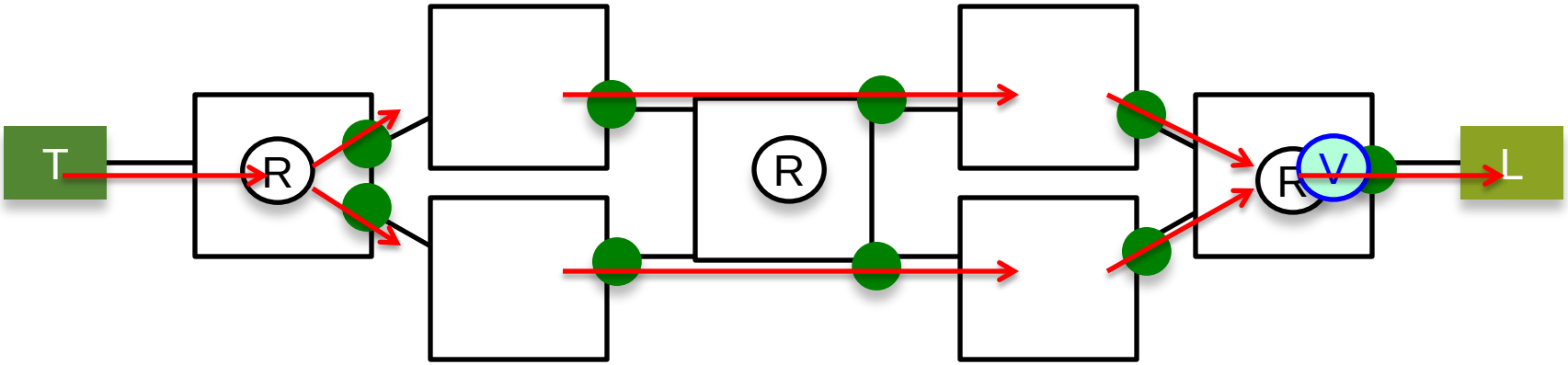
- Only three ports are enabled to pass the one {VID,DA} pair.
- If the Sequence Discard (V) is in the **Bridge**, then the Listener typically receives only **one** copy of every frame.

Proxy Split/Merge Choice 1

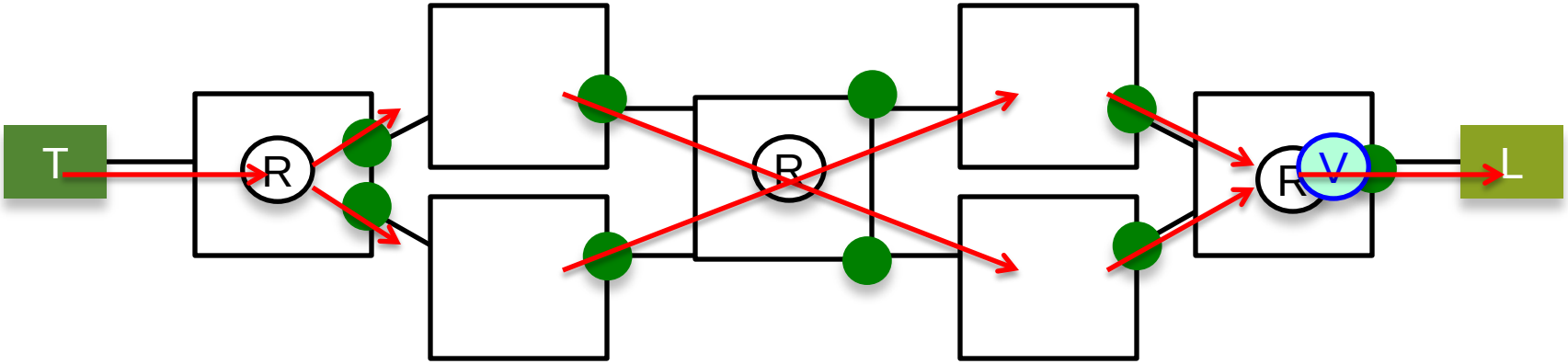


- Intermediate Split/Merge functions work.
- Only green ports ● pass the flow.
- Sequenc discard (V) eliminates duplicates.

Proxy Split/Merge Choice 1

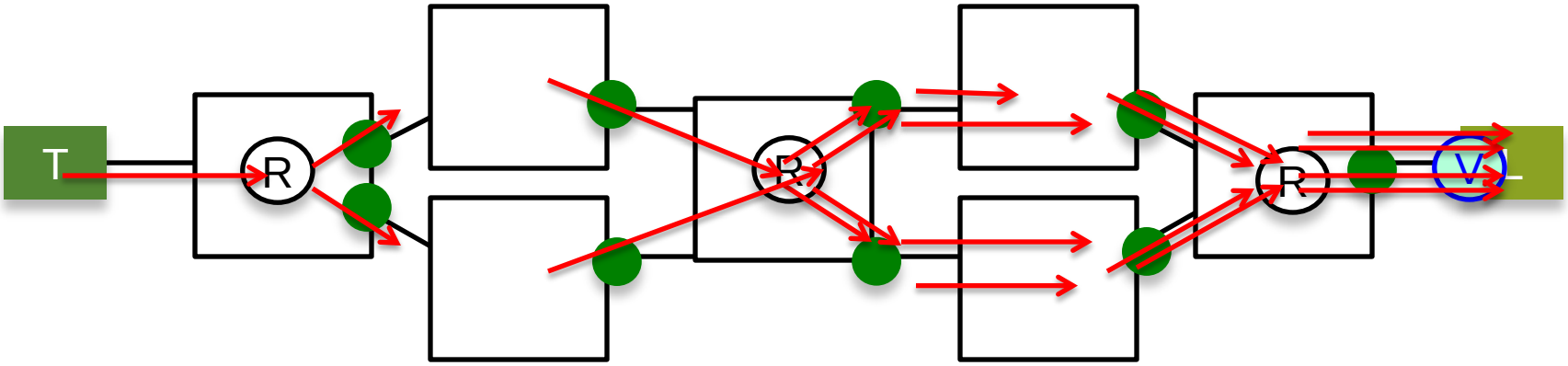


- But, what if the circuits collide?



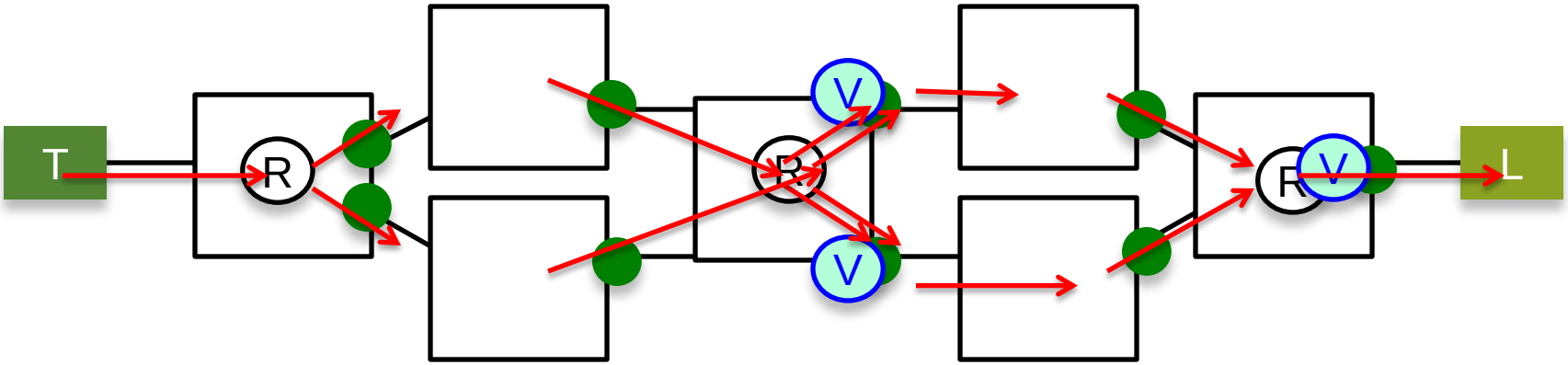
- Which flow is right?

Proxy Split/Merge Choice 1



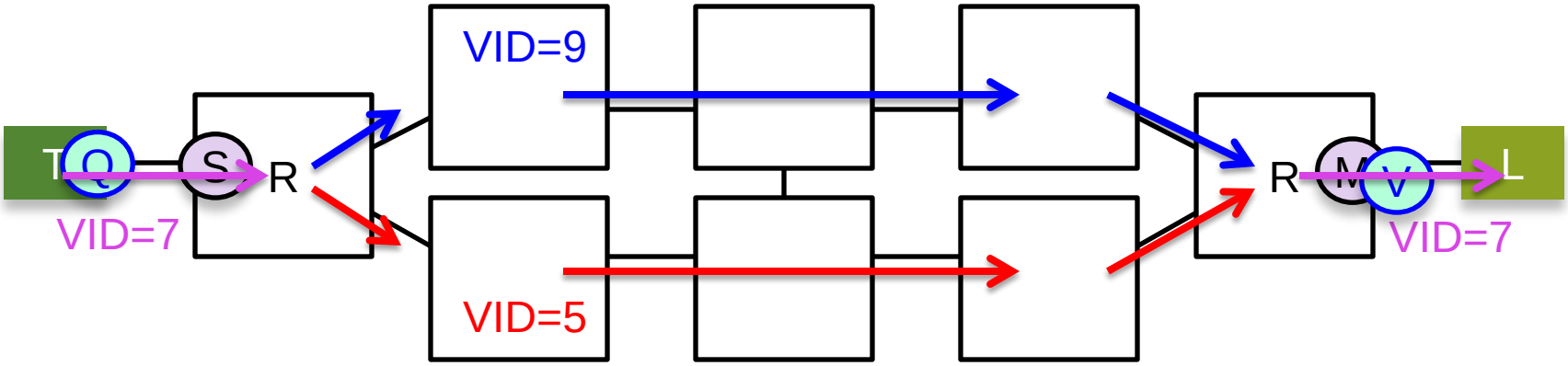
- Won't we get extra copies of everything?
- Well, yes. **Unless** …

Proxy Split/Merge Choice 1



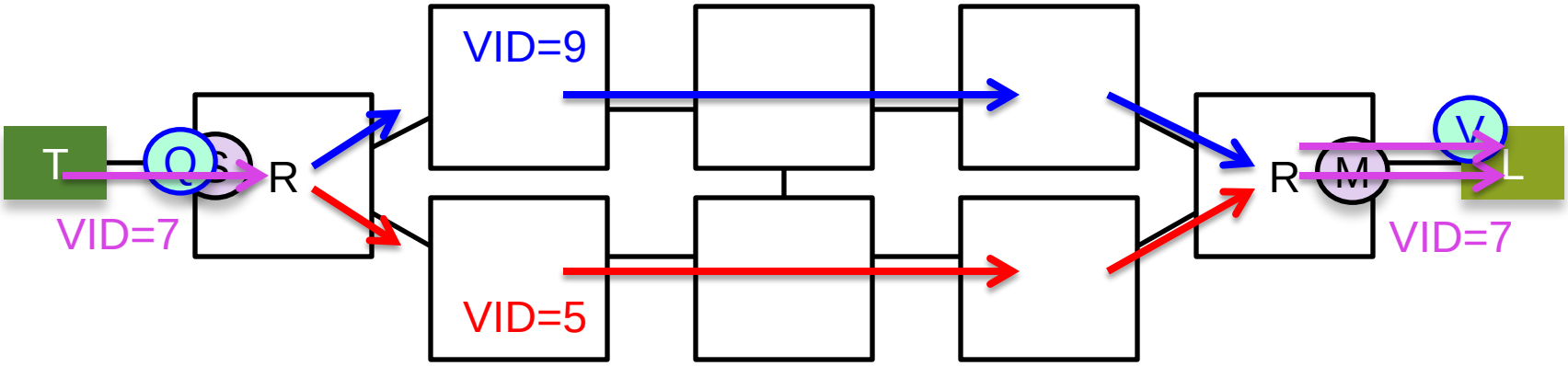
- You get extra copies unless you supply the necessary Sequence Discard functions.
- So no, you **do no criss-cross** the circuits.
- You do a **one-box re-split re-merge**.

Proxy Split/Merge Choice 2



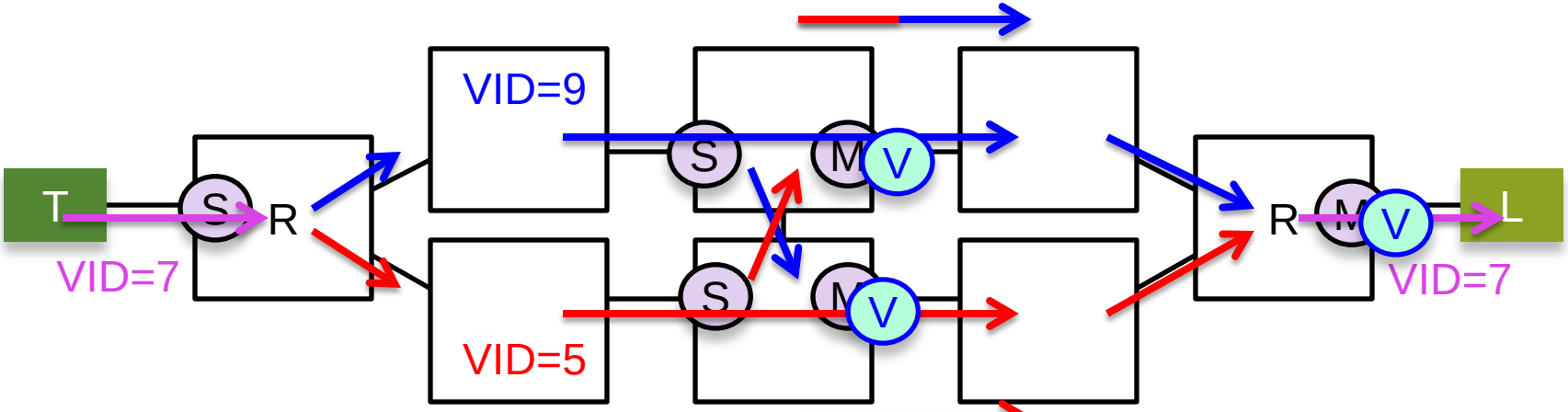
- Choice 2 requires placing the Split (S) and Merge (M) functions on the right ports.

Proxy Split/Merge Choice 2



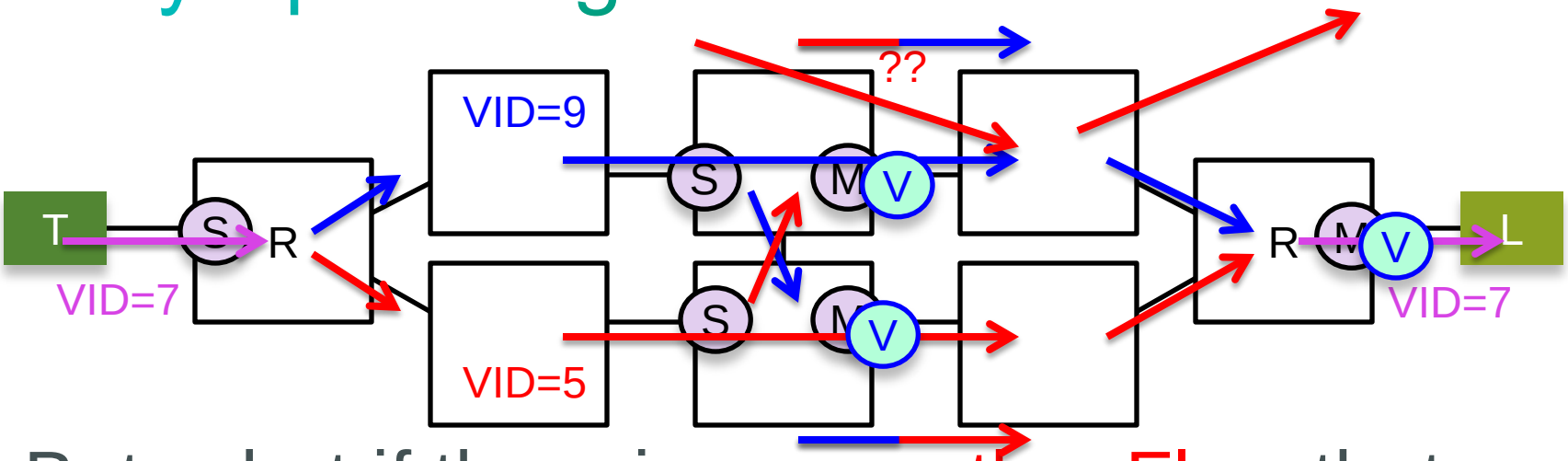
- As for Choice 1, the Sequencing functions Q, V can be at either end of the host links.

Proxy Split/Merge Choice 2



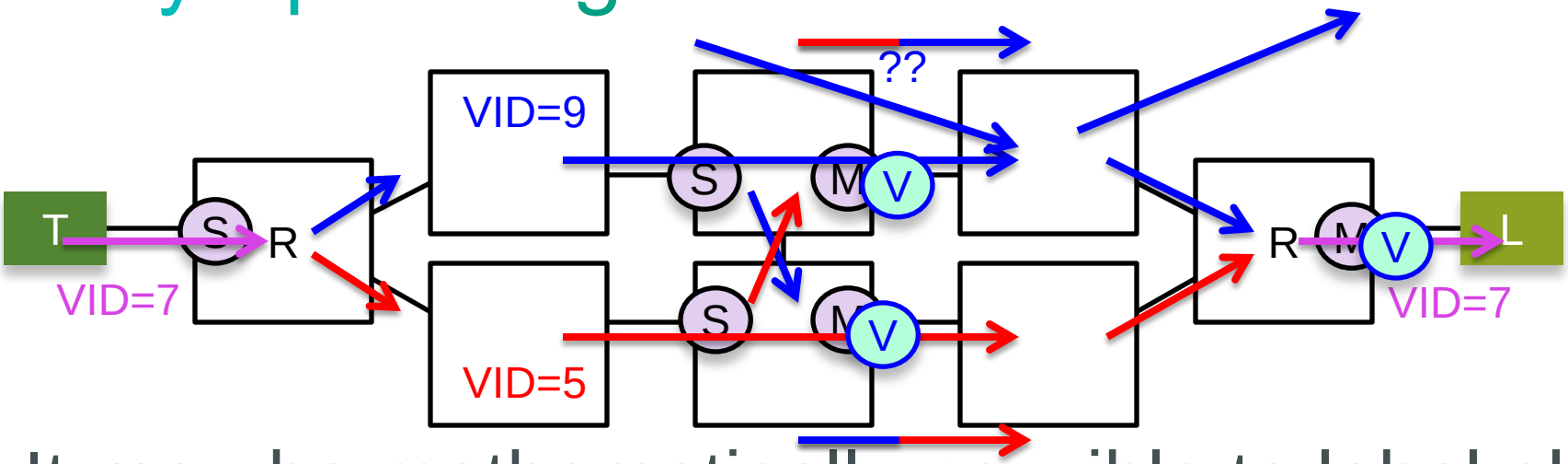
- There are issues, however, when you try to do the intermediate Split/Merge.
- The upper center bridge must change VID 5 to VID 9, and the lower center bridge must change VID 9 to VID 5, all on their output ports.
- (And/or, change the DAs.)

Proxy Split/Merge Choice 2



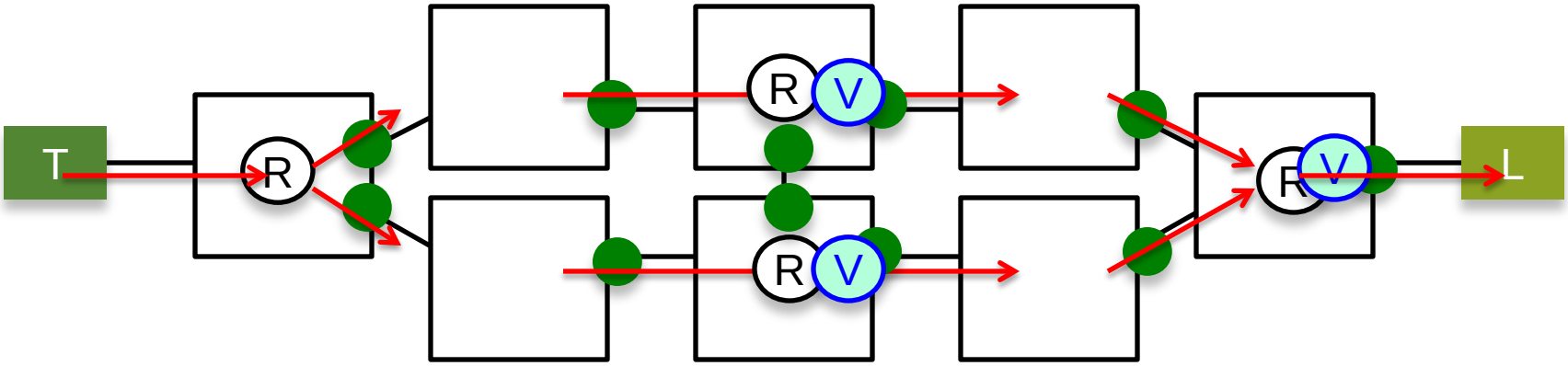
- But, what if there is some other Flow that also uses VID 5, and it does **not** need to be remapped?
- You would have to do **per-flow Circuit ID translation**. Of course, that's what the Merge function (M) does, anyway.

Proxy Split/Merge Choice 2



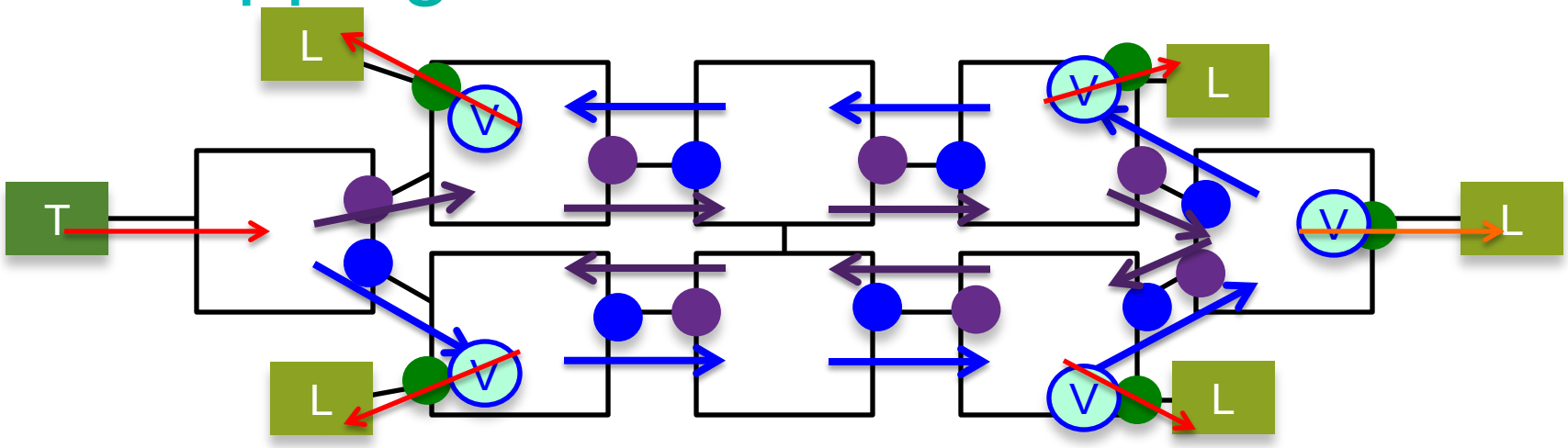
- It **may** be mathematically possible to label all ports in the network as either a “red” or a “blue” port, so this problem never comes up.

Proxy Split/Merge Choice 1



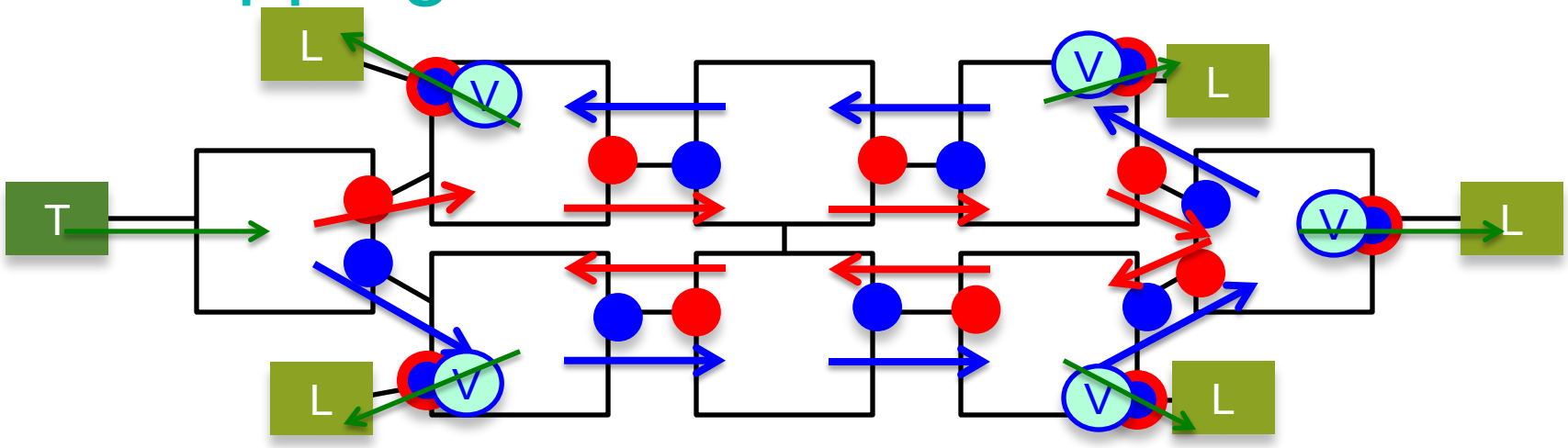
- But, it appears to this author that Choice 1 is a lot easier.

Overlapping distribution Choice 1



- A single VLAN also works for overlapping distribution, as in the ring, above. (Packets go both clockwise ● and counterclockwise ●.)
- We then depend upon the fact that a multicast is never transmitted on the port on which it was received.

Overlapping distribution Choice 2

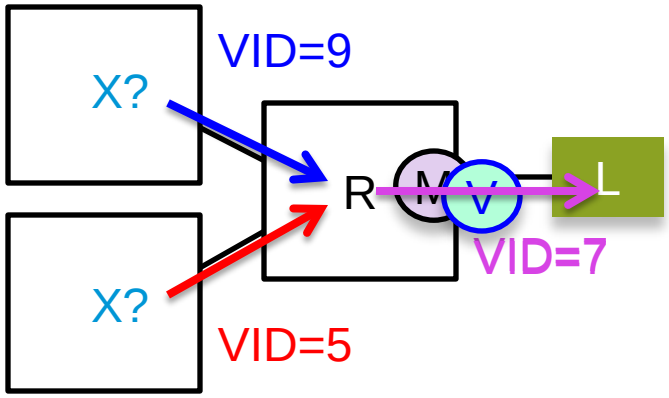


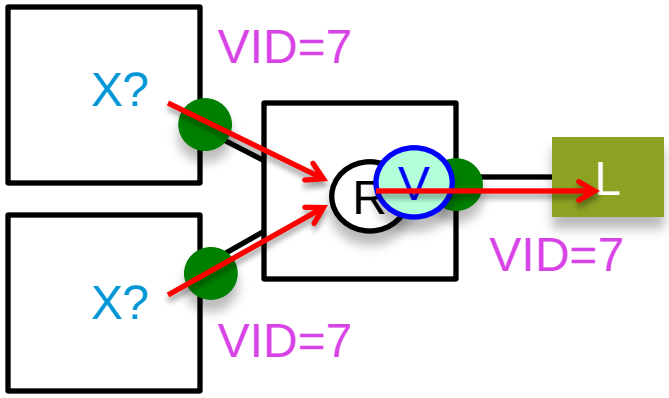
- Clearly, two VLANs work, also.
- Some discussion is in order to decide whether there are significant control plane advantages to using two VLANs. (If we never flood, it doesn't seem to matter, much.)

Isn't that dangerous?

- Doesn't choice 1 enable a misconfiguration that could blow up the network with a multicast storm caused by a circular path?
- It seems slightly more probable that Choice 1 would blow up than Choice 2; but both can blow up.
- To ensure against loops, you would need a new pair of VLANs (or DAs) **at every hop**. The requirement for VLAN IDs or DA remapping would be large.
- In either case, **when using pinned-down paths, input filters that block frames from arriving on unexpected ports are an important safety feature against broadcast storms**.

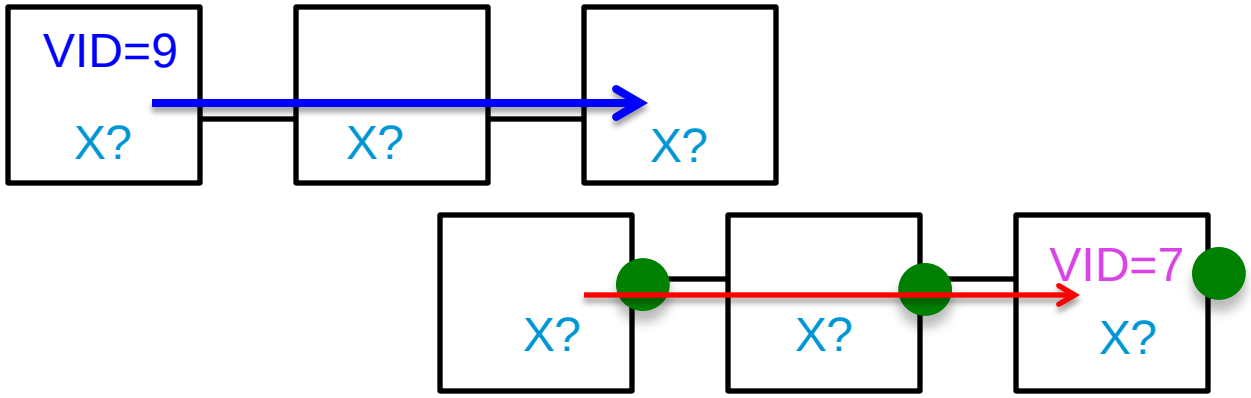
What about latent error detection?





- We must detect the situation where one path has failed, but not both, so that we have no protection. This is **latent error detection**.
- It is easier to identify which leg has the problem at the merge sequence point, if the paths have different labels? **Yes**.

What about latent error detection?



- But, you also have to figure out **where** the error occurred, and the different VIDs are no help, there.

Summary

Summary

- You can use one {VID, DA} circuit label pair for all of the sub-flows making up a stream protected by seamless redundancy.
- Doing so makes life simpler for the data plane, and avoids the need to teach a bridge how to do per-flow VID translation or destination address remapping.
- There may be control plane reasons to prefer more than one VID or DA.
- This author favors the use of one circuit label for all paths of a flow.

Thank you.