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

Norman Finn
Version 1

Feb. 17, 2014

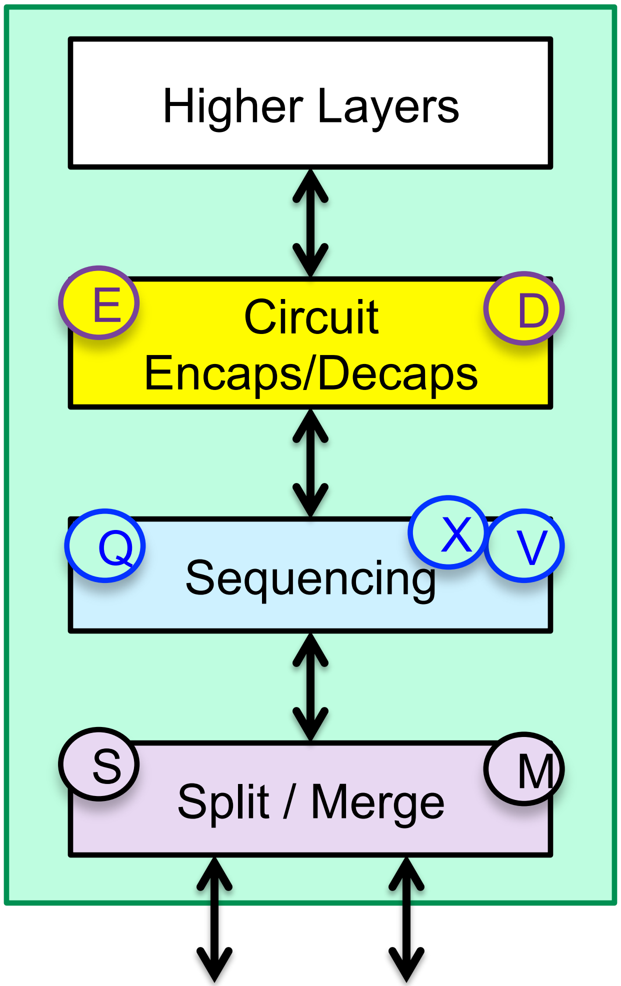
# Layering

# References

- This presentation is [cb-nfinn-How-Many-VLANs-0214-v01.pdf].
- It is based on the layering model presented in [tsn-nfinn-L2-Data-Plane-0214-v03].

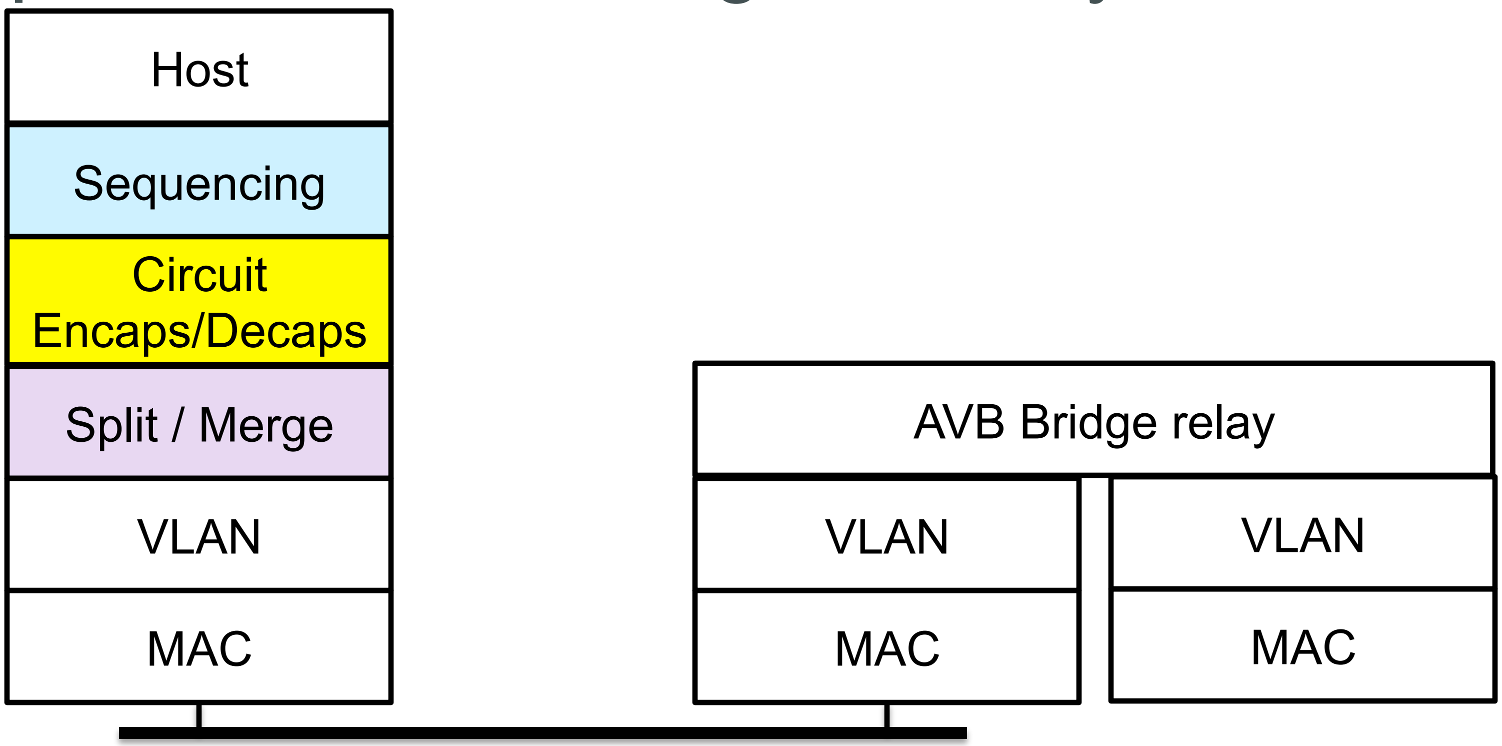
# **Host** layering (from [L2-Data-Plane])

- Higher Layers work as always.
- Circuit Encaps/Decaps E/D marks individual circuits.
- Sequence tagging Q V and discard X can detect missing packets, discard duplicate or misordered packets, and/or correct packet ordering.
- Split / Merge S M redirects and relabels packets among its 1 upper and *N* lower ports.



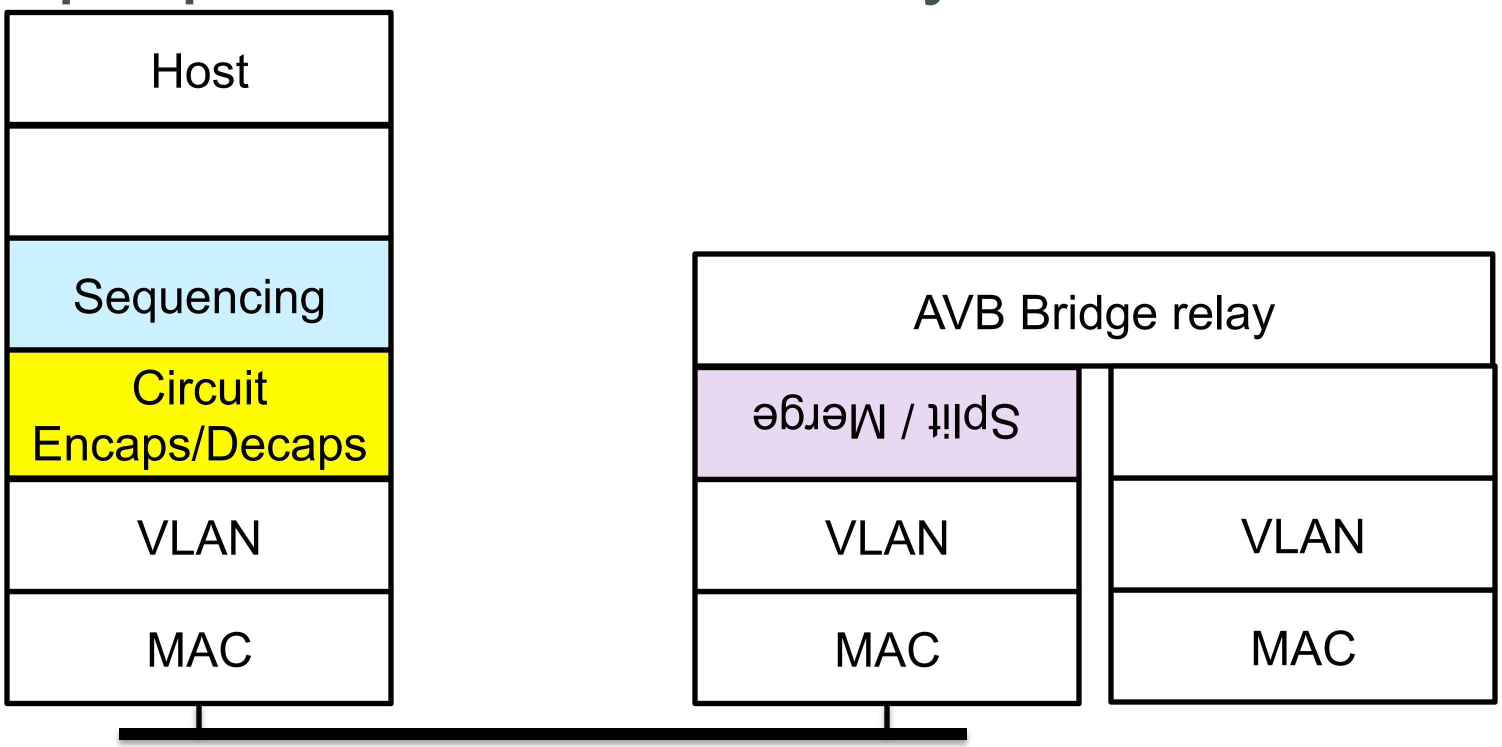
# Multiple systems: **HSCM-B** split

- This is one (extreme) model for a full TSN-aware host. The host does the frame duplication. The Bridge is today's AVB Bridge.



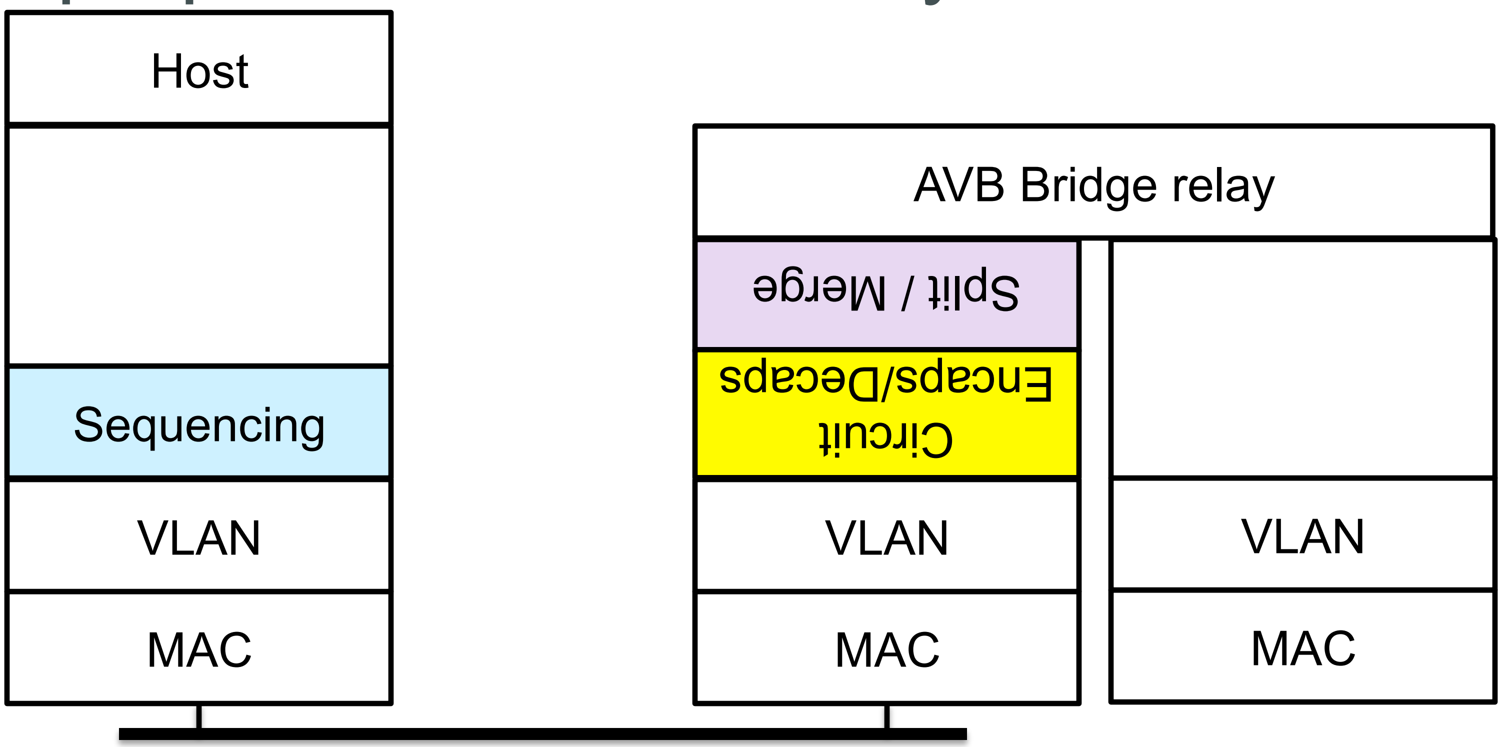
# Multiple systems: **HSC-MB** split

- As we shift TSN functions to the bridge, the stack turns upside down in order to maintain the proper order as seen by the data frames.



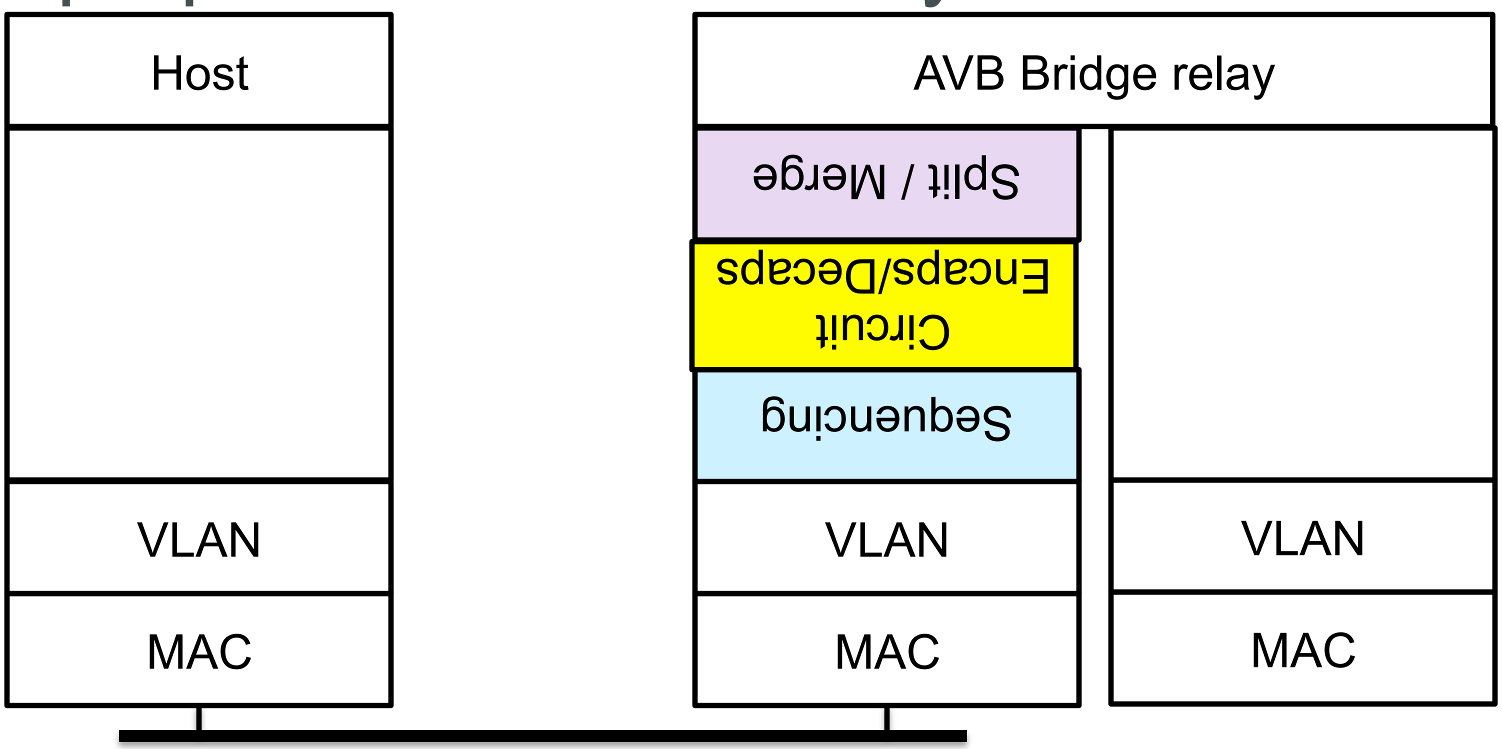
# Multiple systems: **HS-CMB** split

- As we shift TSN functions to the bridge, the stack turns upside down in order to maintain the proper order as seen by the data frames.



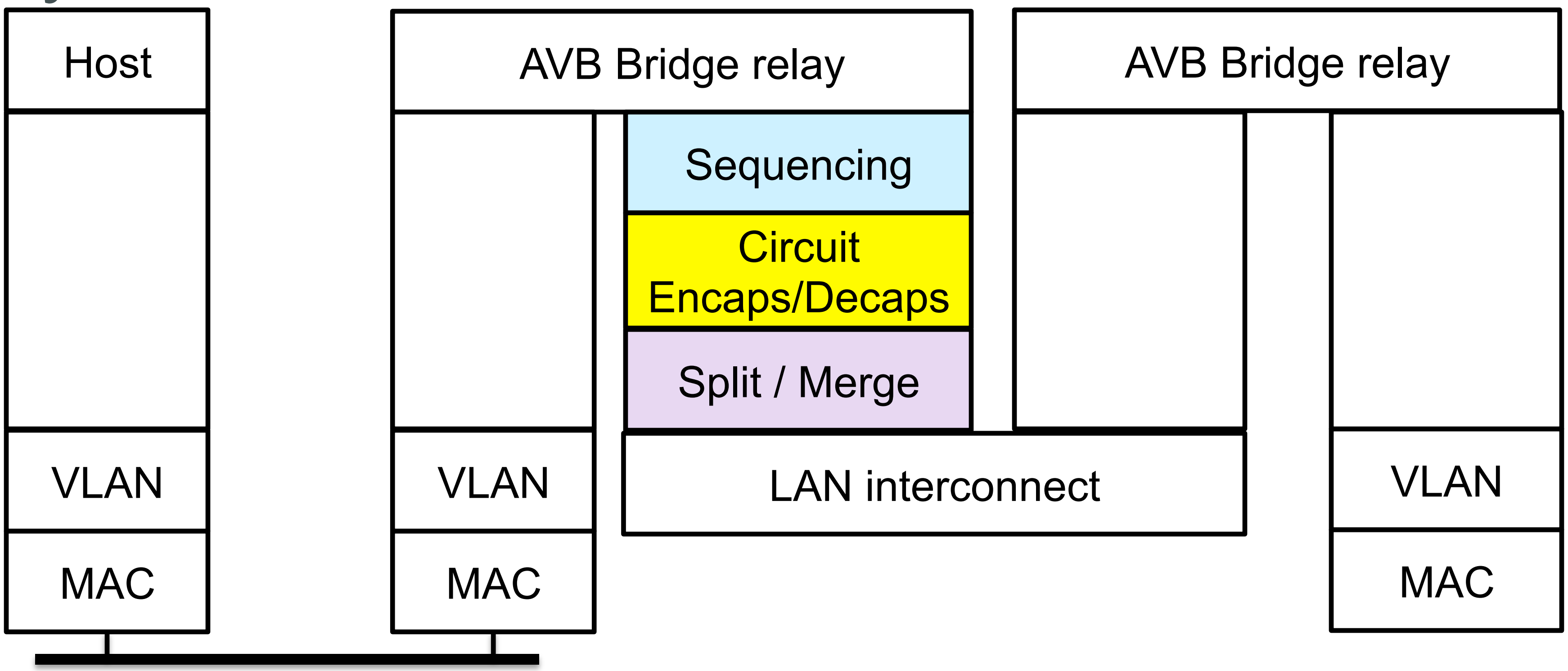
# Multiple systems: **H-SCMB** split

- As we shift TSN functions to the bridge, the stack turns upside down in order to maintain the proper order as seen by the data frames.



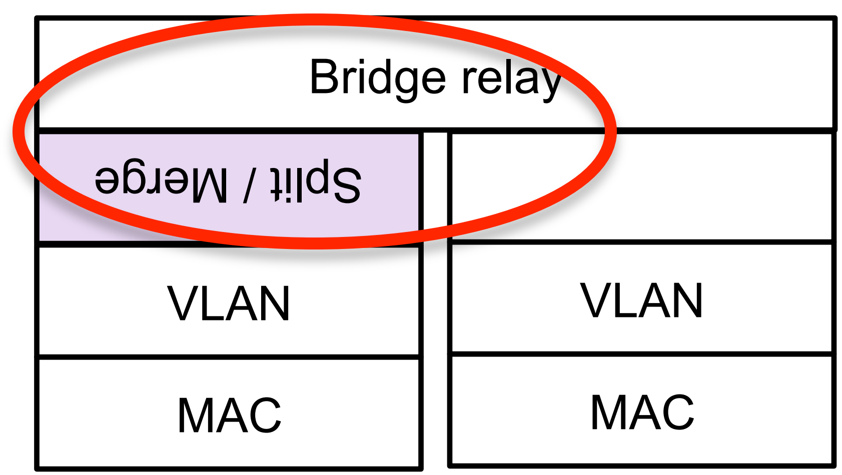
# Multiple systems: **H-SCMB** split

- This is how you draw the bridge proxy so that the higher layers are always above the lower layers.



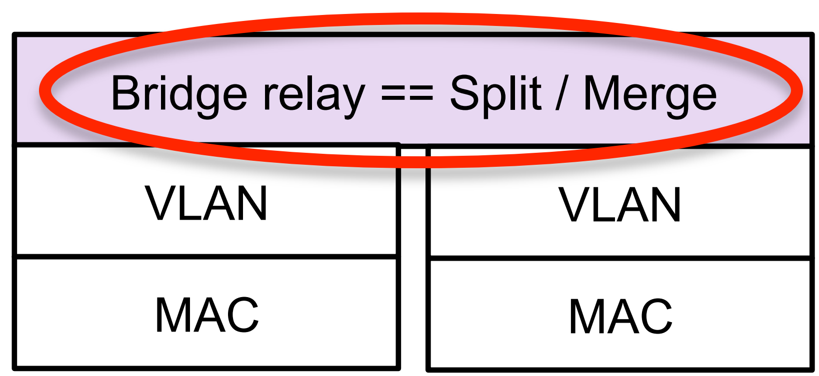
# Bridge layering

- Let's look at just the relationship between the Split / Merge function and the Bridge relay.
- We'll assume that the Circuit Encaps/Decaps and Serializing functions are in the host (**HCS-MB**), so we don't have to think about deep packet inspection.



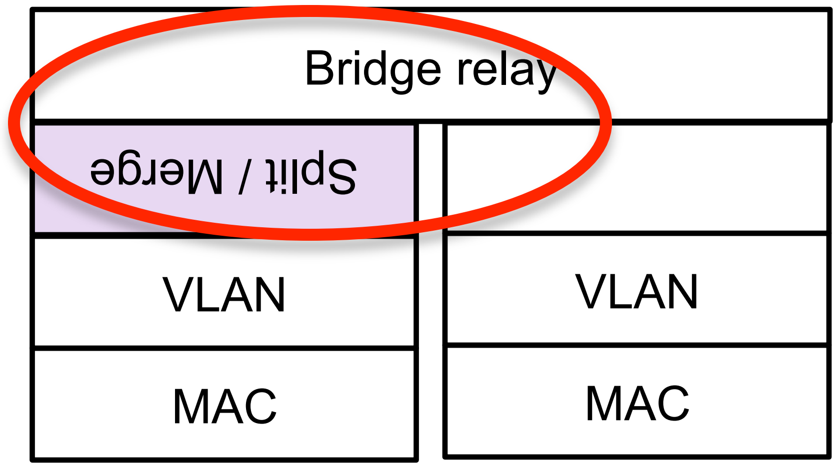
# Bridge/Split/Merge layering **Choice 1**

- Split / Merge **is** the Relay function.
- Split / Merge is the standard actions of the Filtering Database on a (multicast) frame.
- The **DA must be the same** for the two streams, because the bridge relay function cannot change it.
- The **VID is the same**, also, at least until the frame exits a port.



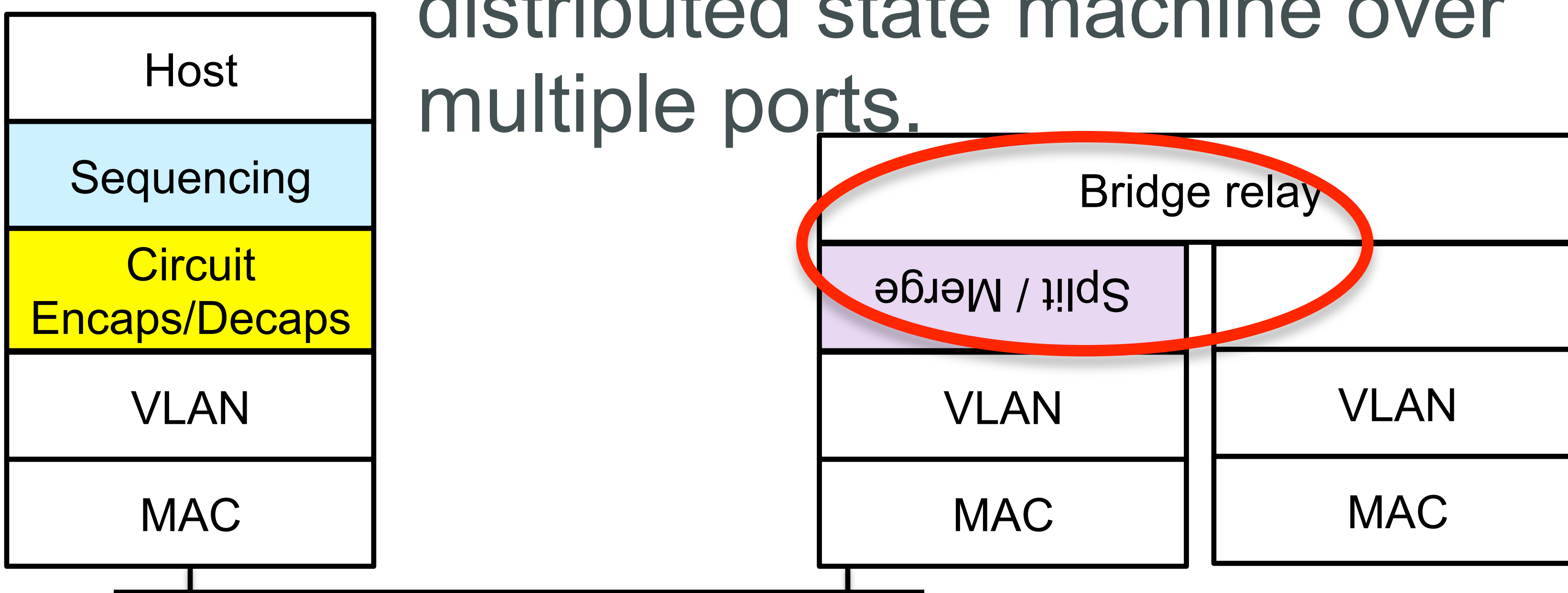
# Bridge/Split/Merge layering **Choice 2**

- Split / merge is an in-line duplication/deletion function with a single up port and a single down port.
- Split / Merge can only differentiate the two (or more) circuits on its upper port by {VLAN ID, Destination MAC} pair.
- Either the **VID or the DA must be different** among the circuits, so that the relay can send them in different directions.



# Bridge/Split/Merge layering **Choice 2**

- Note that the Split or Merge function must be on the port nearest the Talker (Split) or the Listener (Merge). Otherwise, we either have to modify the relay, or invent some kind of distributed state machine over multiple ports.



# 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**

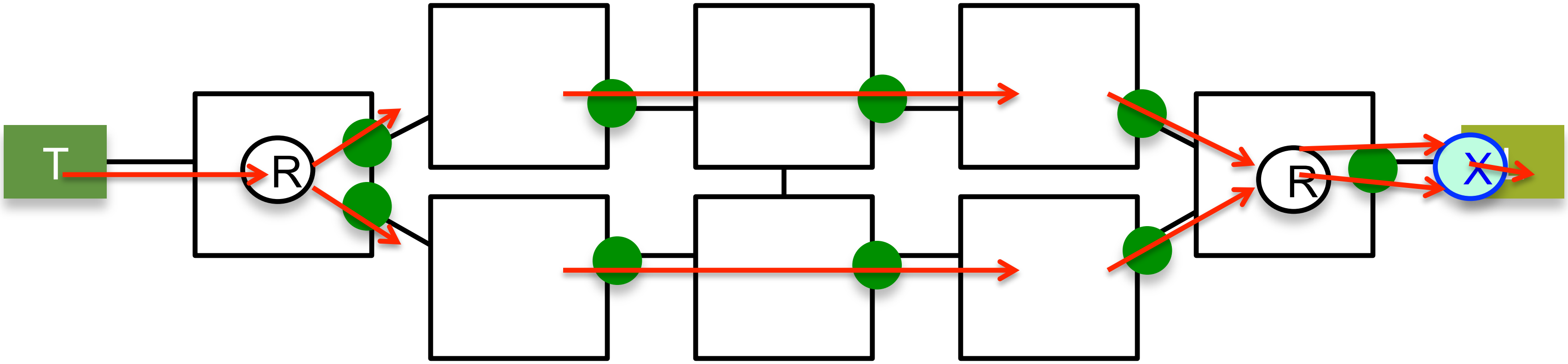

- Split/Merge functions S M are the normal Bridge relay function R .
- The flow has the **same** {VLAN ID, Destination MAC address} circuit identifier on both paths and at both ends.
- (Note: **two frames** output after the merge.)

# 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 X .)
- Each path has a **different** {VID, DA} pair, perhaps different from the outer pairs.

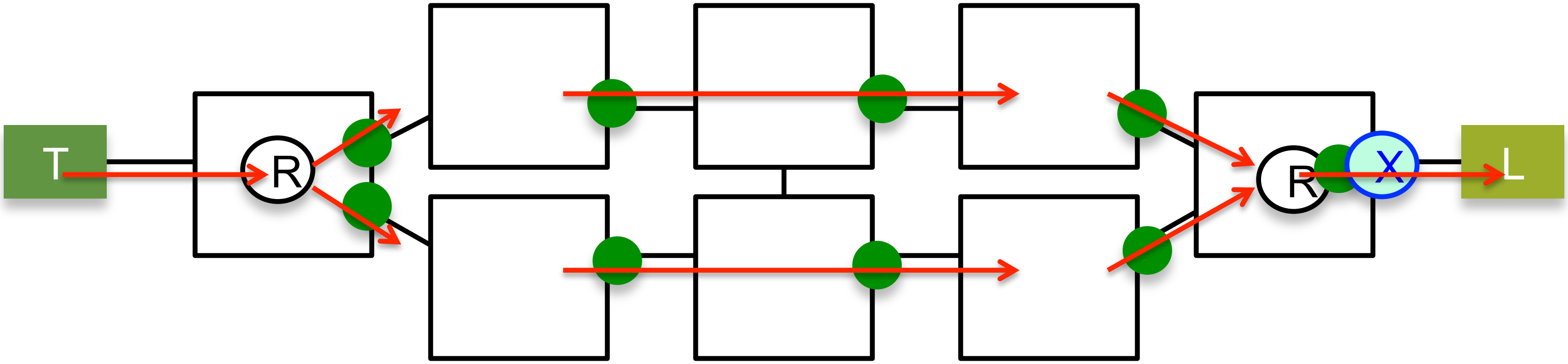
# Network view

# Proxy Split/Merge Choice 1



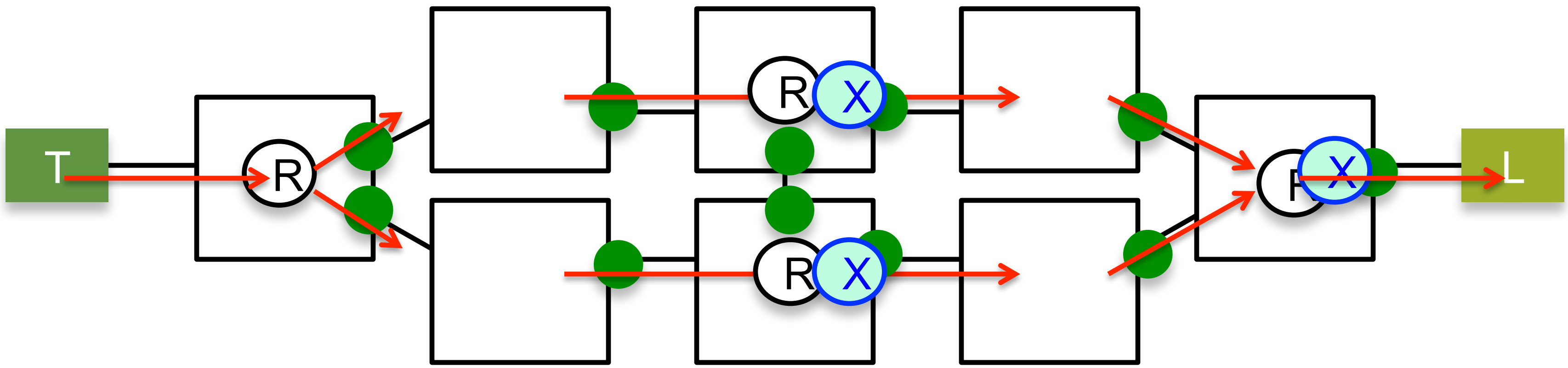

- The Split / Merge function is a relay function R
- Only these ports are enabled to pass the one {VID,DA} pair.
- If the Sequence Discard function X is in the **Listener**, then the Listener typically receives **two** copies of every frame.

# Proxy Split/Merge Choice 1



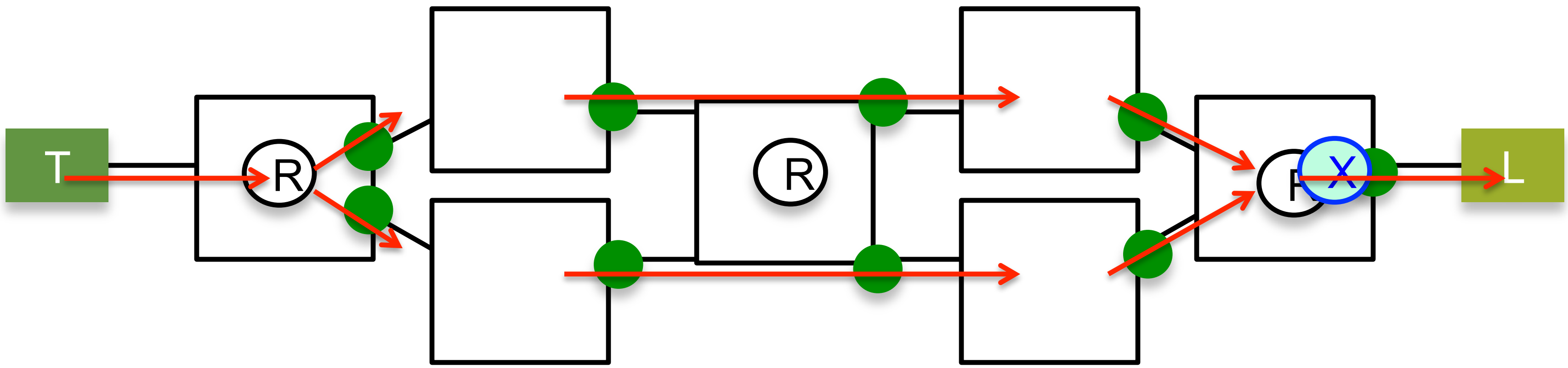

- Only three ports are enabled to pass the one {VID,DA} pair.
- If the Sequence Discard X is in the **Bridge**, then the Listener typically receives only **one** copy of every frame.

# Proxy Split/Merge Choice 1

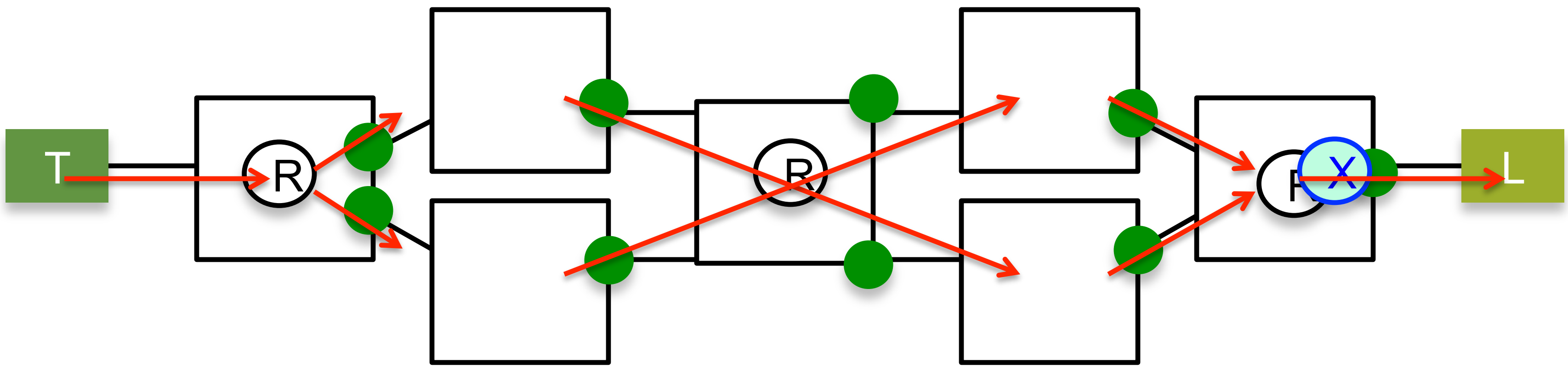



- Intermediate Split/Merge functions work.
- Only green ports ● pass the flow.
- Sequenc discard (X) eliminates duplicates.

# Proxy Split/Merge Choice 1

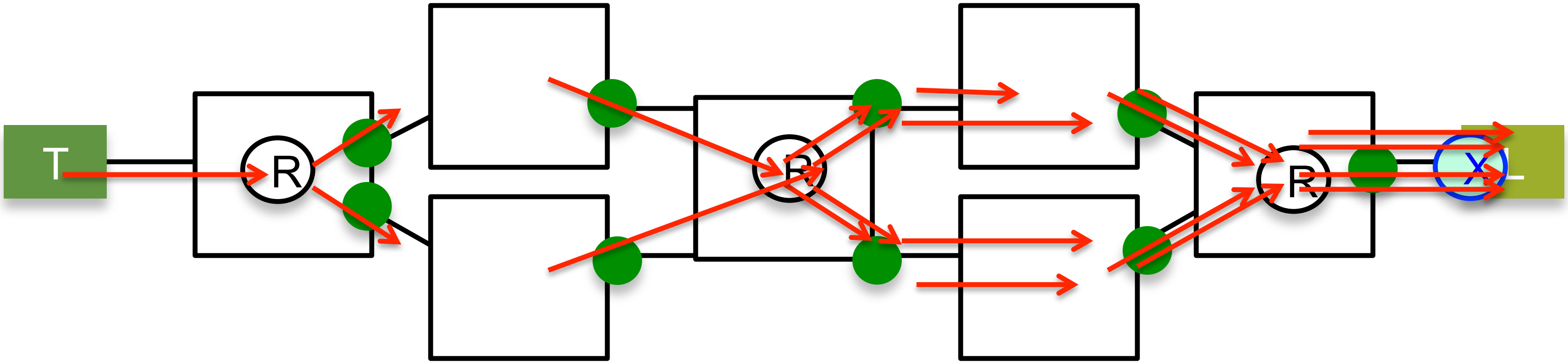



- But, what if the circuits collide?



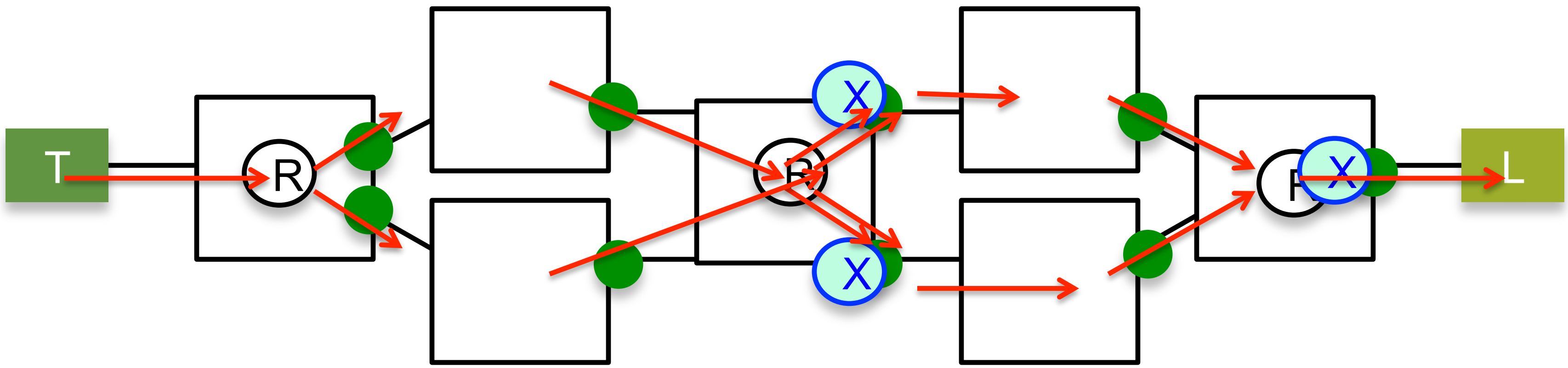

- Which flow is right?

# Proxy Split/Merge Choice 1



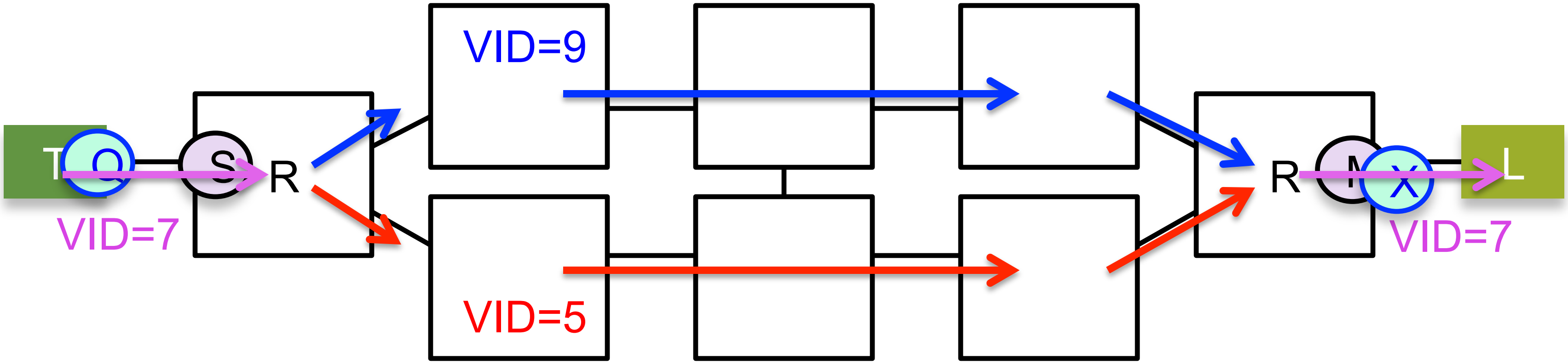

- Won’t we get extra copies of everything?
- Well, yes. **Unless** …

# Proxy Split/Merge Choice 1



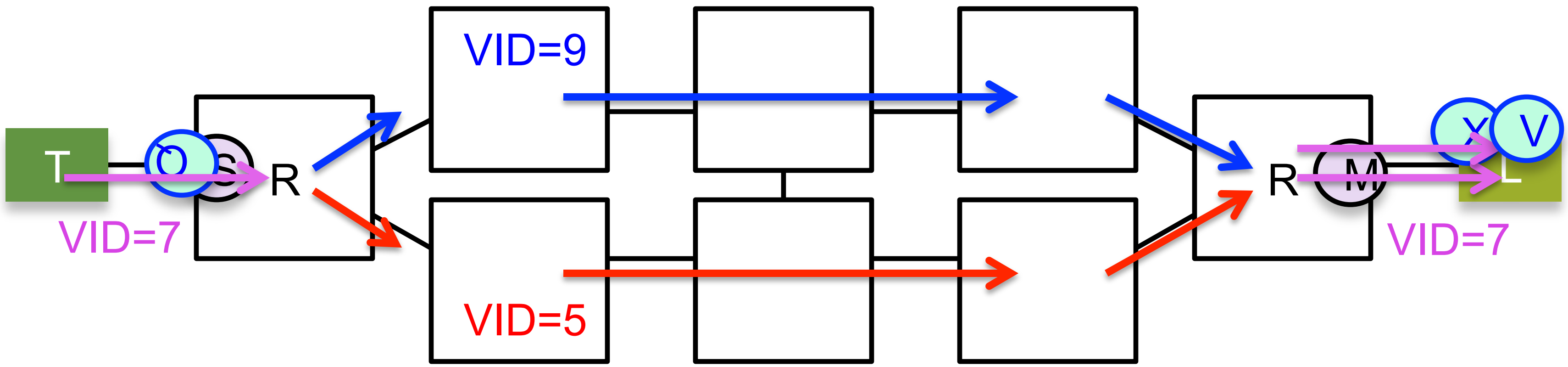

- You get extra copies unless you supply the necessary Sequence Discard functions.
- So no, you **do no criss-cross** the circuits.
- You do a **one-box re-split re-merge**.

# Proxy Split/Merge Choice 2



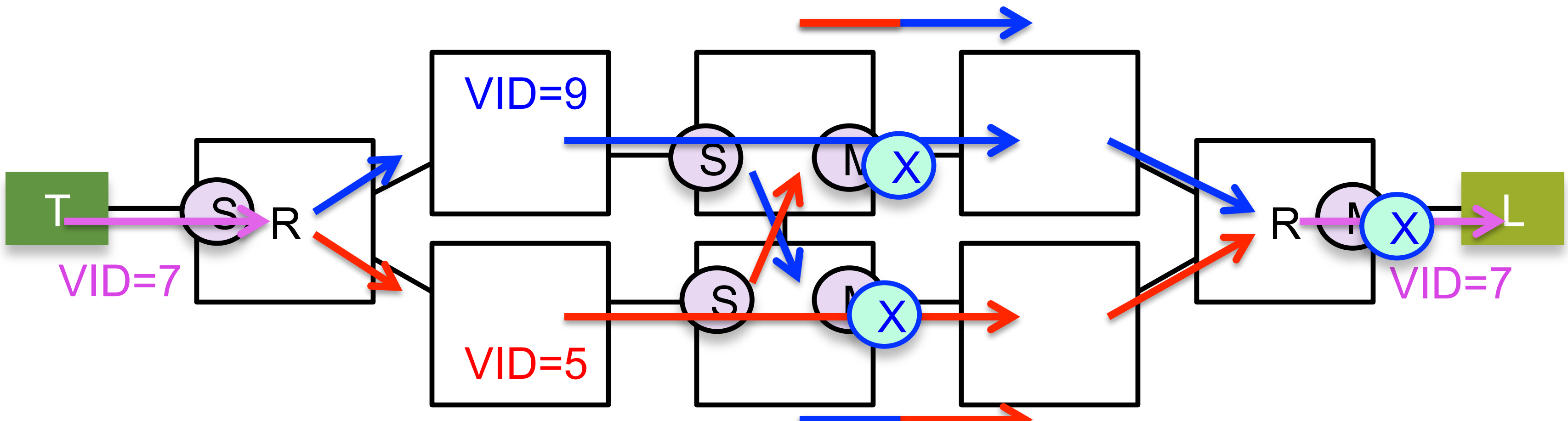

- Choice 2 requires placing the Split S and Merge M functions on the right ports.

# Proxy Split/Merge Choice 2



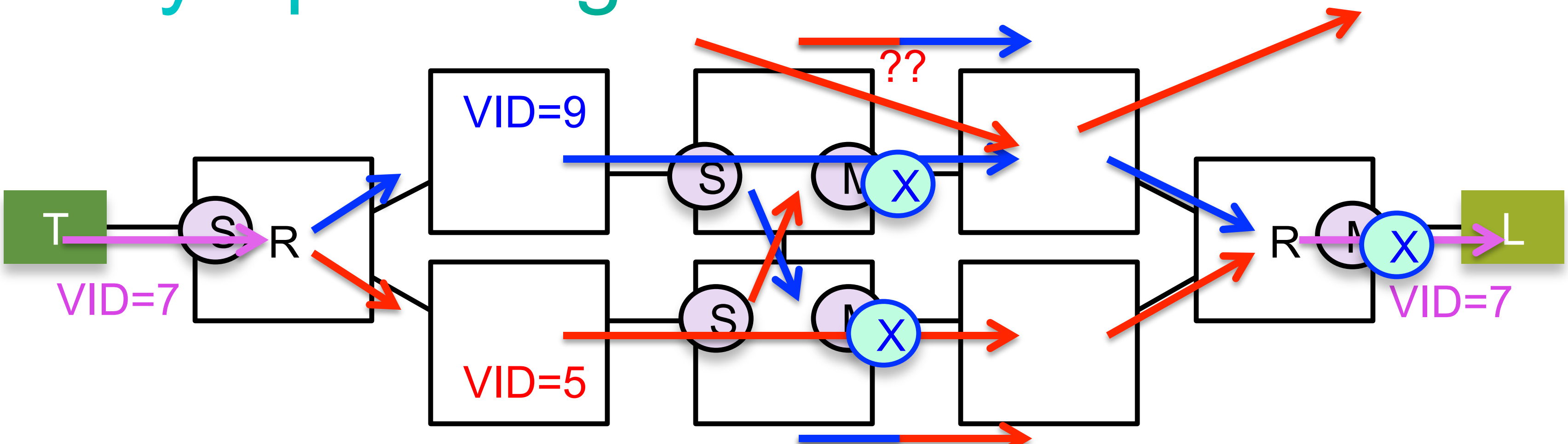

- As for Choice 1, the Sequencing functions Q, X, V can be at either end of the host links.

# Proxy Split/Merge Choice 2



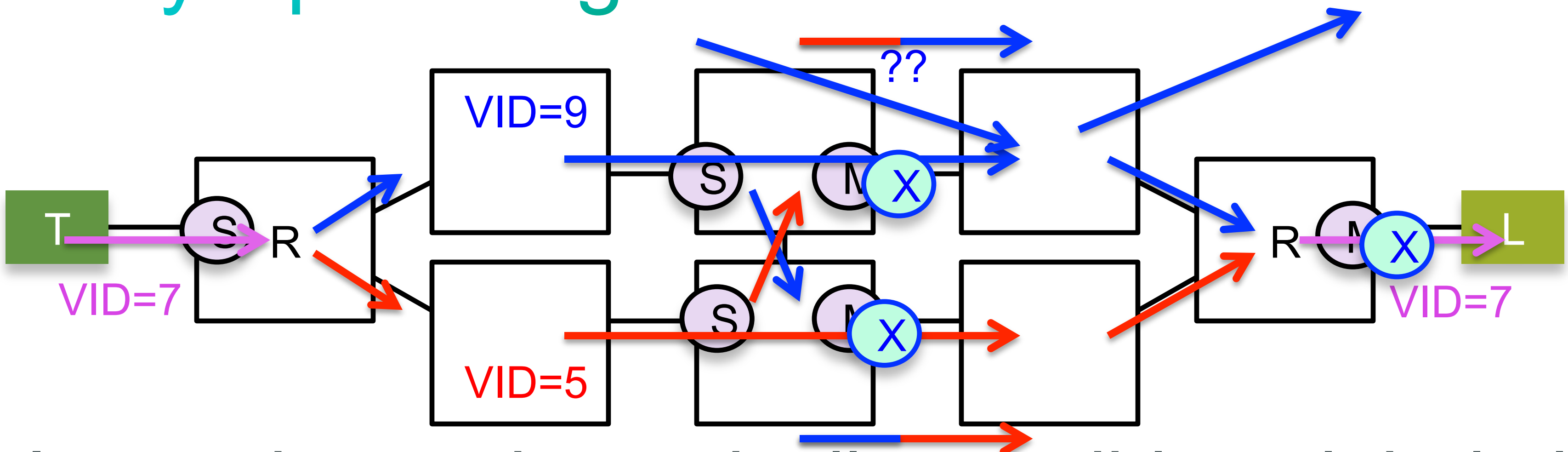

- There are issues, however, when you try to do the intermediate Split/Merge.
- The upper center bridge must change VID 5 to VID 9, and the lower center bridge must change VID 9 to VID 5, all on their output ports.
- (And/or, change the DAs.)

# Proxy Split/Merge Choice 2



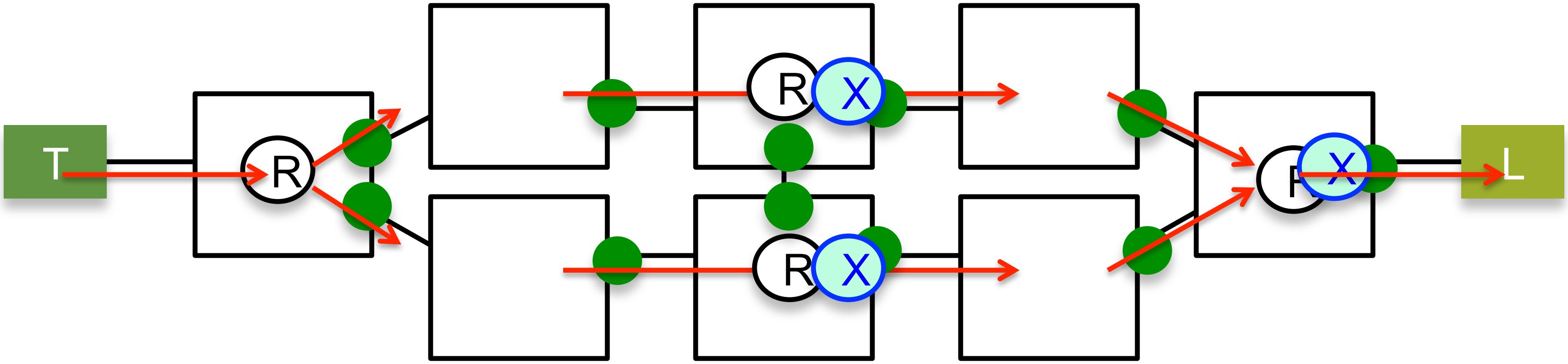

- But, what if there is some other Flow that also uses VID 5, and it does **not** need to be remapped?
- You would have to do **per-flow Circuit ID translation**. Of course, that's what the Merge function (M) does, anyway.

# 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

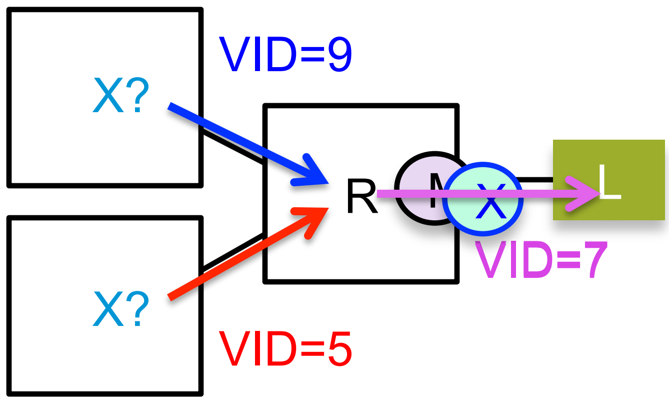


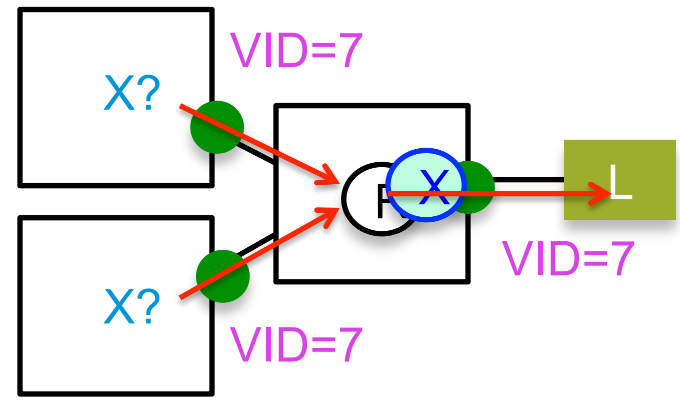

- But, it appears to this author that Choice 1 is a lot easier.

# 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 grow arbitrarily.

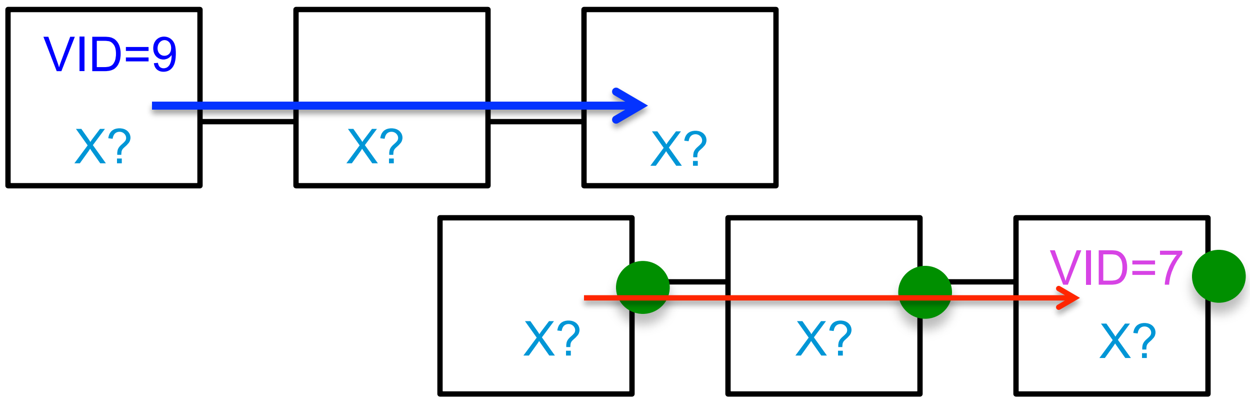
# What about latent error detection?




- 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, and avoids the need to teach a bridge how to do per-flow VID translation or destination address remapping.
- So, use one circuit label for all paths of a flow.

Thank you.