Proper Layering for the TSN Layer 2 Data Plane

Norman Finn
Version 1

Jan. 13, 2014

Moving ahead TSN L2/L3 work

- At the AVnu face-to-face meeting in December, Norm Finn made presentations on the subject of ensuring that the IEEE 802.1 Time-Sensitive Networking (TSN) efforts are compatible with the established body of work from Internet Engineering Task Force (IETF) on the Internet Protocol (IP).
- A four-step plan for advancing the TSN work was presented.

Moving ahead TSN L2/L3 work

A. Pick at least one data plane model for joint L2/L3 **stream identification**.

B. Pick at least one data plane model for joint L2/L3 **duplicate packet deletion**.

C. Pick at least one data plane model for the **dual-homed end station**.

D. Pick exactly one (hopefully) suite of **protocols** to fit the data plane choices made.

This presentation

- This is tsn-nfinn-L2-Data-Plane-0114-v01. It offers an improved model for layering the AVB/TSN concepts, which leads to a set of choices for answering the data plane questions A, B, and C. Contents of this presentation:
  - What's broken about AVB/TSN?
  - What's the Fix?
  - Alternative TSN Encapsulations
  - Summary
- This presentation does not make choices, and so cannot advance to Question D, protocols.

What is Broken?

What's broken?

- This author perceives a **layering problem** in the current AVB/TSN standards suite that is the source of a number of difficulties:
  - Integration/reconciliation with the way the IP layer uses Ethernet, especially hosts that use VLANs for separate addressing or broadcast domains.
  - Meeting the need for carrying unicast streams and the need for multiple fixed paths through the network, in a manner that does not confuse MAC address learning or waste VLAN ID space.

What's broken

- Without detracting from its achievements, especially plug-and-play, AVB remains a solution for a few very specific communities, where the vendors and users, together:
  - Have complete control over the application.
  - Have complete control over the host stack.
  - Have complete control over the network.
- Fortunately, it's not hard to fix.
- And once fixed, we have less work to do.

Complete control over the **application**

- How do **existing** applications address each other?
  - Universal Resource Locator (HTTP).
  - Unicast and multicast IPv4 and IPv6 address.
  - Individual and Group MAC address.
  - Locally-administered MAC address.
  - VLAN, often one VLAN = one IP subnet.
- How must **AVB** applications address each other?
  - Group MAC address on a particular VLAN.

Complete control over the **host stack**

- How do **existing** higher layers obtain MAC addresses?
  - Domain Name Servers are used to map host addresses to IP (or other!) addresses.
  - IPv4 stacks issue ARP requests/replies.
  - IPv6 stacks use Neighbor Discovery.
  - Fixed mapping of IPv4 multicast to MAC addresses.
- How must **AVB** stacks obtain MAC addresses?
  - Override the stack to use a 1722 address.

Complete control over the **network**

- How are **existing** networks designed?
  - Bridges provide “who” addresses, and learn geography, thus encouraging broadcast plug-and-play protocols like Bonjour and ARP.
  - Network administrators break up the network into bridged domains, separated by routers, to prevent those broadcasts from overwhelming the network.
- How must **AVB** networks be designed?
  - Everything must be one flat bridge domain.
  - 10,000 stations, 5000 streams in a stadium?!?!

:You can make it work

- Demonstrably, there is enough value in AVB for communities of developers to create applications, host stacks, and networks that operate to the AVB specification.
- But, we can make TSN applicable to users who already have their networks defined.

Driving assumption

- The goal of the TSN TG should be to write standards for new **Quality of Service** (QoS) classes for high reliability and low latency, that offer **incremental benefit** to **any network**, whether L2, L3, or mixed, that follows established general-purpose operational models.
- To the extent that TSN standards require variations from those models, their adoption will be hindered.

Questions and Answers

- What is the industry standard term for standards like, “Stream Reservation”?
- **Connection-oriented services (or circuits) over connectionless networks.**
  - When you have per-flow state in every node along the path, and rebuild the per-flow state when the path changes, that’s a “connection” or a “circuit”.

Questions and Answers

- How does the industry set up a circuit from A to B over a connectionless network?
- **A gives the network address of B to a protocol that sets up a connection, and that protocol returns to A (and B) an encapsulation, with a network address, to use for the connection's packets.**

Questions and Answers

- How does the packet get from source to destination?
- **A encapsulates the packet in a form that:**
  - **The network will get to B; and**
  - **Can be recognized by B, and by participating network nodes along the way, as a packet belonging to this particular connection.**

Questions and Answers

- Why was Nigel Bragg saying that we need one VLAN per source per path through the network (worst case) for nailed-up paths?
- **Because we didn't realize that we are tunneling circuits through the network, and thought we had to have the connectionless network addresses on the outside of the frame.**

Questions and Answers

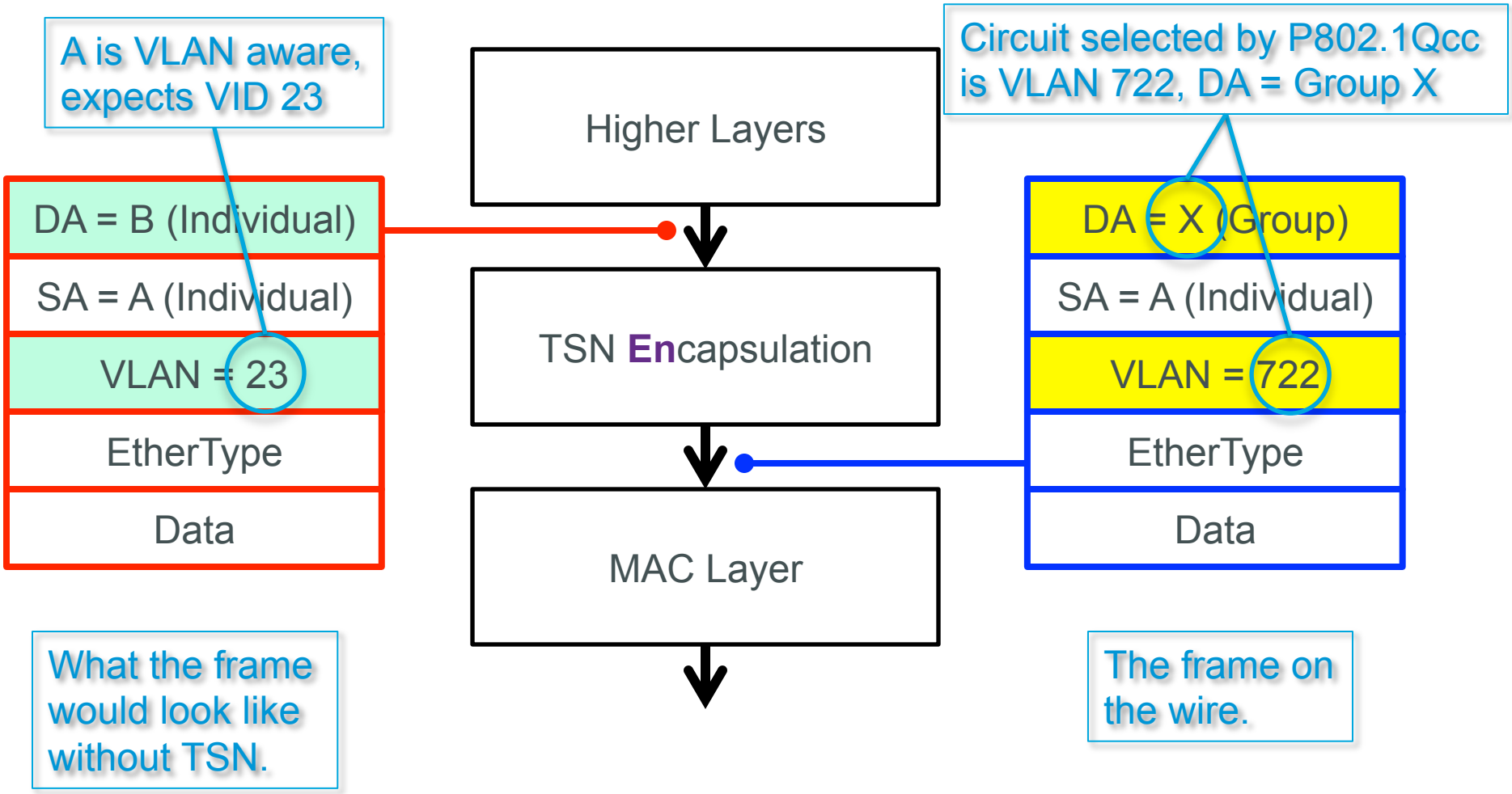
- What's wrong with the AVB frame format?
- **Nothing!**

Questions and Answers

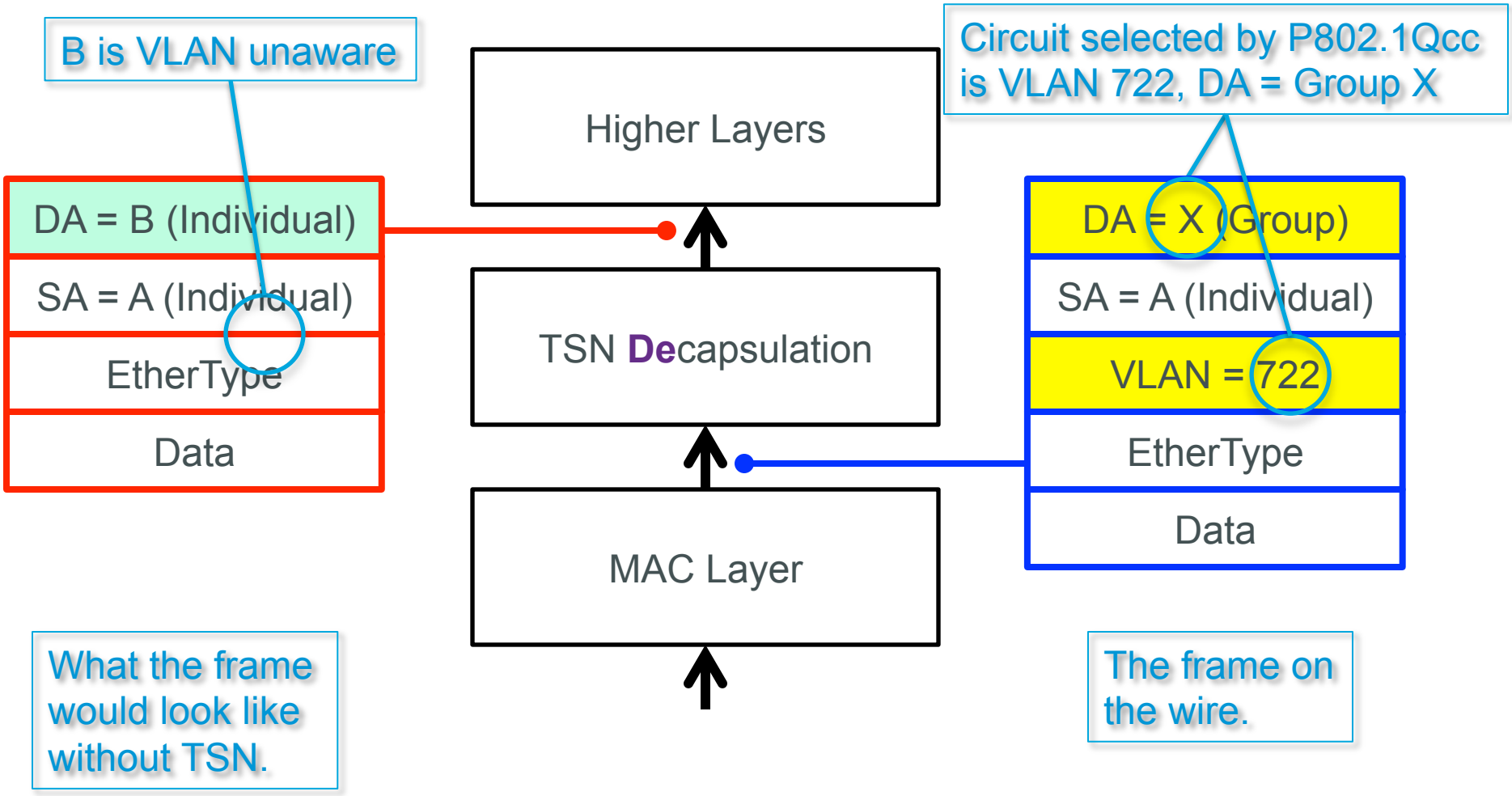
- Then, what's the fix?
- **Read on.**

What is the Fix?

Correct stack in **Talker A**



Correct stack in **Listener B**



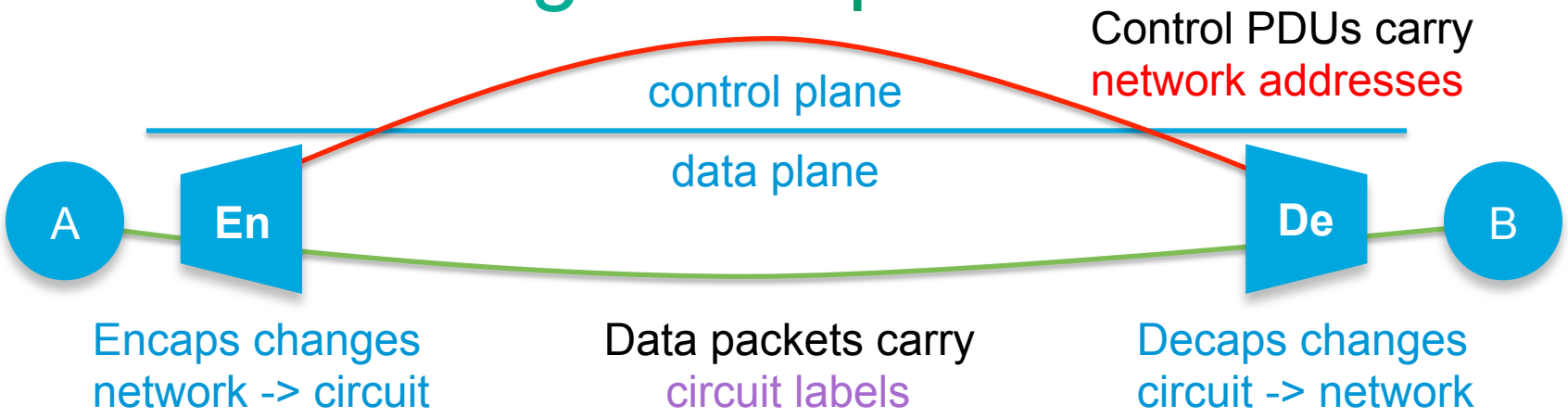
How does this get setup?

- **A**'s control functions use P802.1Qcc to say, "I want a connection to the function reached by a connectionless packet (an Ethernet frame) with VLAN 23 and Destination MAC address **B**."
- P802.1Qcc picks the encapsulation {VID 722, Group Address X}.
- Along the path from Bridge to Bridge, each bridge makes whatever modifications to the end-point addresses inside the P802.1Qcc PDU that would be made to the network packet.
  - For example, VID translations or tag removal.

How does this get setup?

- The edge Bridge adjacent to **B**, in particular, removes the VLAN tag from the P802.1Qcc request, and tells **B** about the encapsulation.
  - **B's TSN Encaps/Decaps function is VLAN-aware, even if the rest of B's protocol stack is not!**

How does this get setup?



- The information required by the TSN Encapsulation and Decapsulation functions came from the control plane.

The end result?

- No problem with IP, including unicast streams.
  - The transformations are transparent to the IP stack.
  - The IP stack works just like it always has with ARP, etc.
- No problem with any other protocol that knows about or requires particular MAC addresses or VLANs
  - The {VLAN, address} is restored as it comes up the receiving stack.
- No problem with fixed paths.
  - Every stream has a multicast address. One or two VLANs can support all fixed-path streams, even with VLAN-aware host stacks.
- No problem with MAC address learning.
  - Because fixed paths are on a VID that doesn't do learning.
- **No problem with backwards compatibility.**
  - **Existing AVB stations see the same encapsulation as always.**
  - **We'll craft P802.1Qcc to allow the Talker to supply the tunnel address.**

Why didn’t we see this, before?

- The circuit encapsulation was so trivial, we didn’t realize that we were doing circuits with labels.
- The circuit encapsulation is so trivial, you can implement an Ethernet-only application that uses the circuit label as a network address.

What do we do?

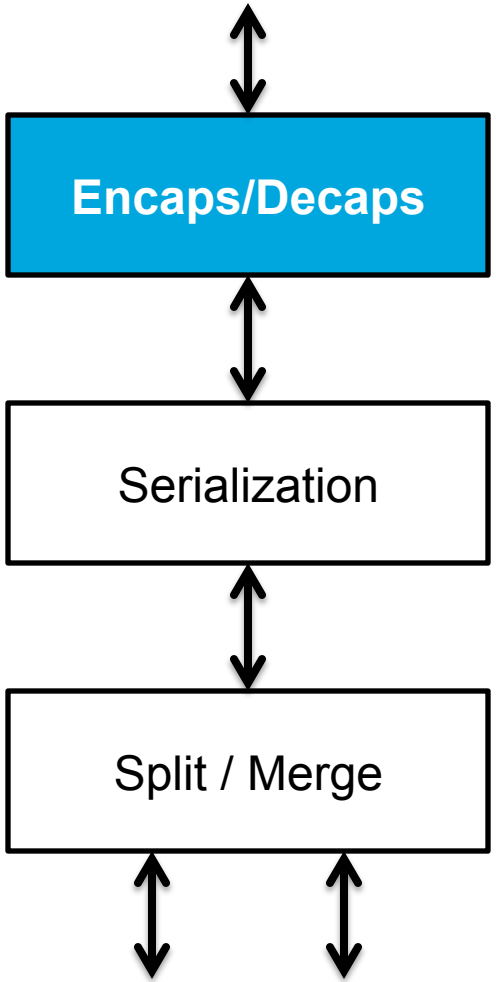
1. Revise 802.1Q to have the TSN Encaps/ Decaps layer, working as described.
2. Fix P802.1Qcc to:
   - Take the connectionless target {VID, MAC address} pair as an input.
   - Modify that pair, as needed, as P802.1Qcc proceeds through the network.
   - Give the circuit {VID, MAC address} to the endpoints, rather that taking it as an input (except as necessary for legacy reasons).

Do I have to change my ASICS?

- If what you're doing now works, don't change it. We're not changing the bits on the wire.
- How or whether you do the full reconstitution of the frame inside your host stack is an implementation matter.
- If you want to provide a totally transparent service to the upper layers, you have the option of implementing the TSN Encaps/Decaps.
- Or, of implementing (or using your existing implementation of) several other protocols.

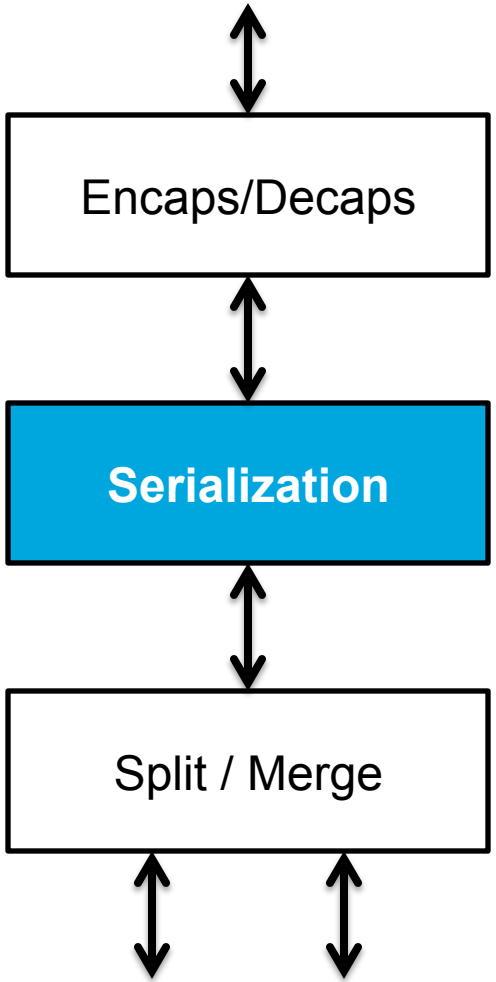
Serialization and Seamless Redundancy

Functional elements required for TSN



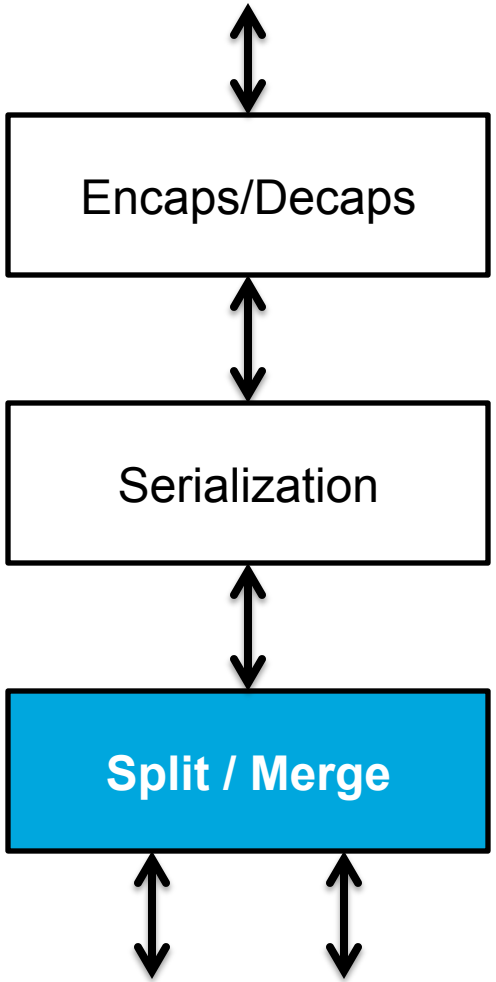
- Individual flows must be identified, and packets encapsulated, for:
  - Fixed paths;
  - Per-flow resources;
  - Seamless redundancy.

Functional elements required for TSN



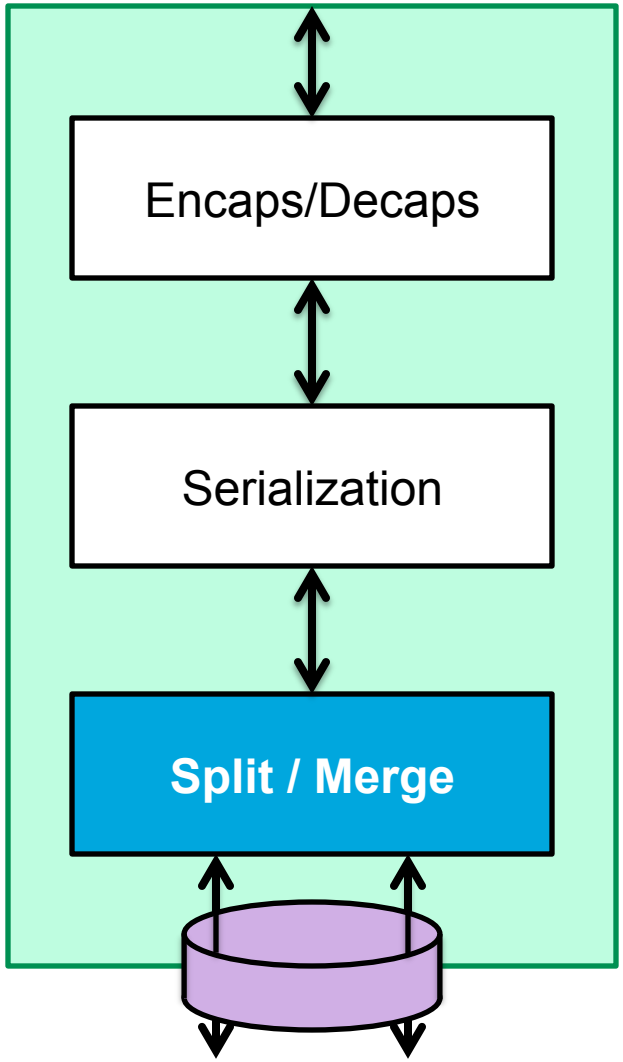
- **If** stream requires either seamless redundancy or long-range out-of-order delivery protection, then its packets must be:
  - Serialized on transmit.
  - Reordered on receive.

Functional elements required for TSN



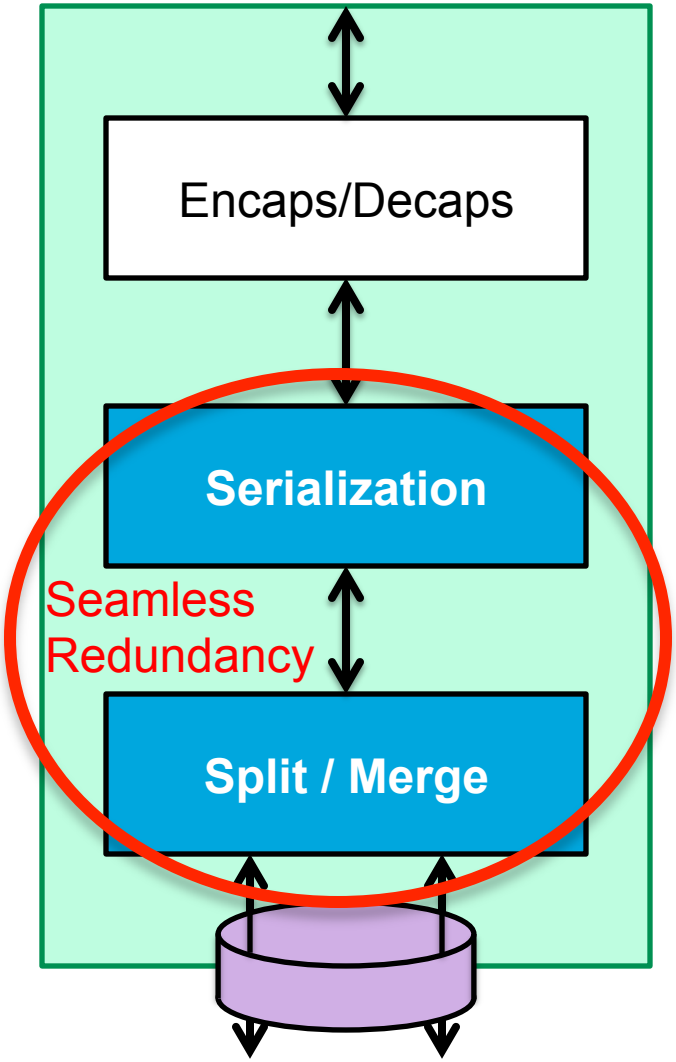
- **If** stream requires seamless redundancy, then its packets must be:
  - Serialized.
  - Split on transmit.
  - Merged on receive.

Functional elements required for TSN



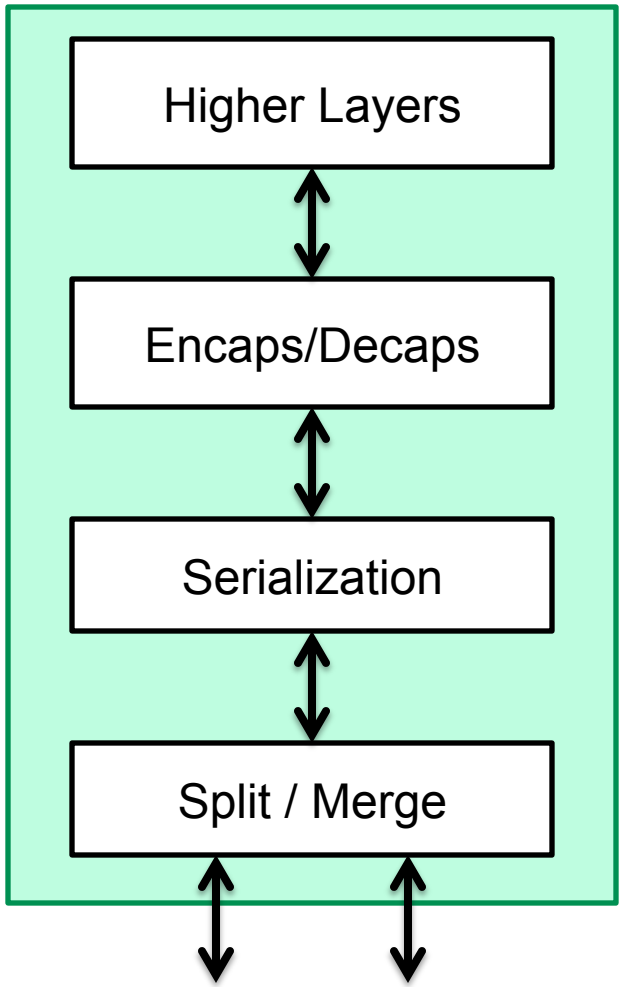
- Note that the Split Merge function does not require two physical ports.
- It may replicate / merge the streams by using different explicit in-band markers, leaving it to normal networking to make the physical split.

Functional elements required for TSN



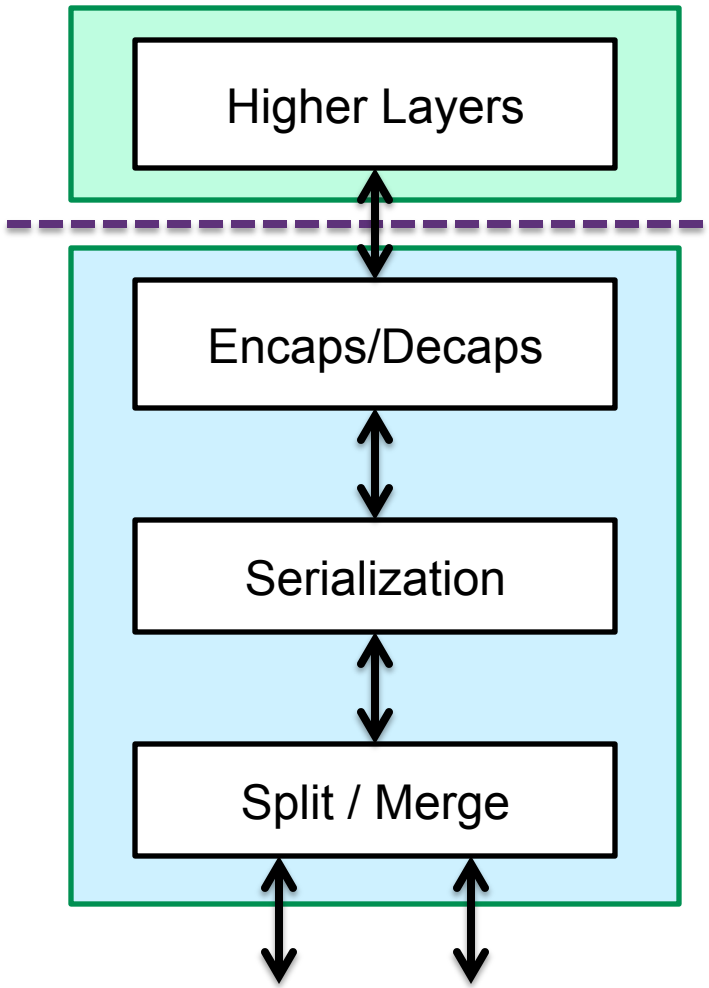
- One may want Serialization without Split/Merge.
- Split/Merge and Serialization often are combined into a single seamless redundancy function.
- All three can be combined, for that matter.

Single system



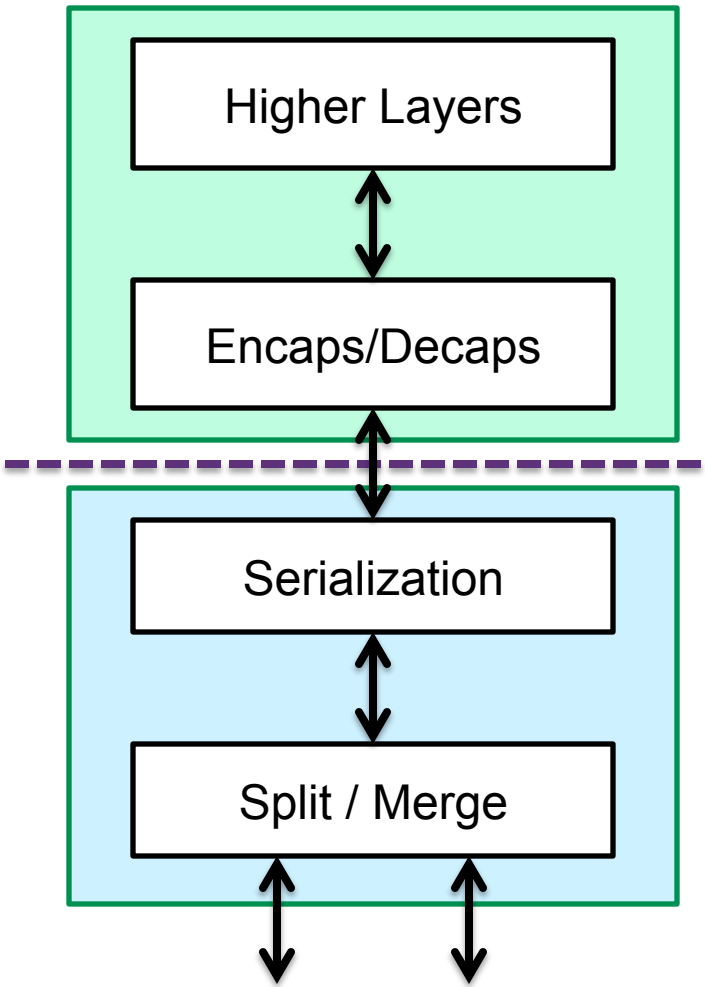
- Within a single system, Flow Identification can be **out**-of-band or **in**-band. Out-of-band methods include:
  - Socket ID.
  - Separate service instances.
  - Multiple Serialization functions.

Single system



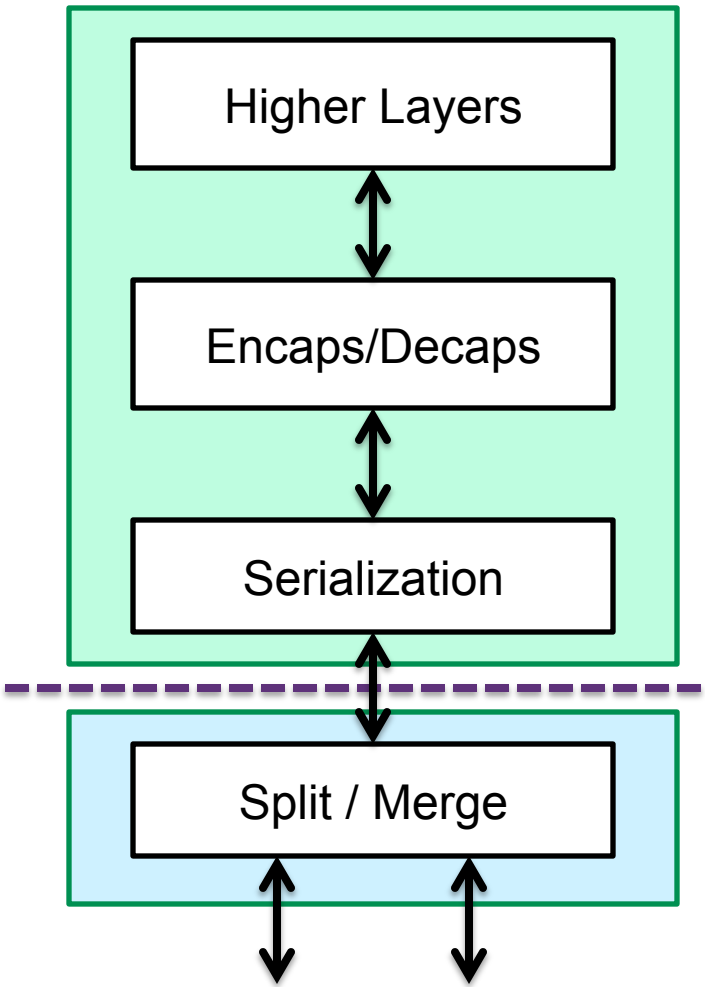
- For multiple systems, flow Identification must be **in-band**:
  - Some form of tag.
  - One or more layers of explicit addresses (e.g. VLAN ID or IP 5-tuple).
  - A flow ID buried in an application.

Single system



- For multiple systems, flow Identification must be **in-band**:
  - Some form of tag.
  - One or more layers of explicit addresses (e.g. VLAN ID or IP 5-tuple).
  - A flow ID buried in an application.

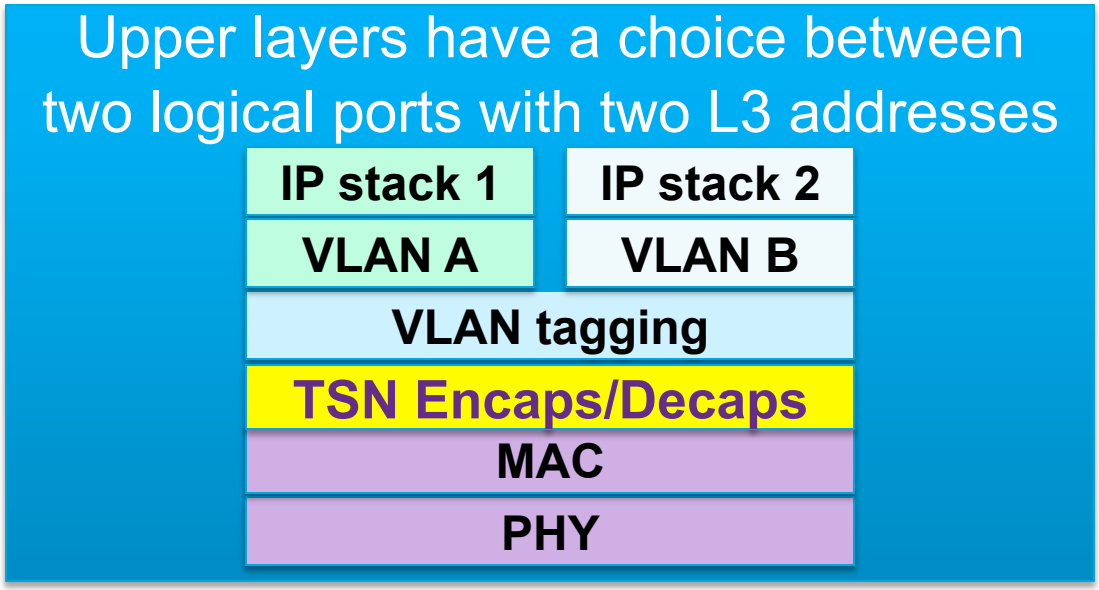
Single system



- For multiple systems, flow Identification must be **in-band**:
  - Some form of tag.
  - One or more layers of explicit addresses (e.g. VLAN ID or IP 5-tuple).
  - A flow ID buried in an application.

Single-port multiple VLAN host

- Common model for a **multi-VLAN host** with a single physical port (router or multi-VLAN server). TSN Encaps/Decaps works, but has no seamless redundancy.

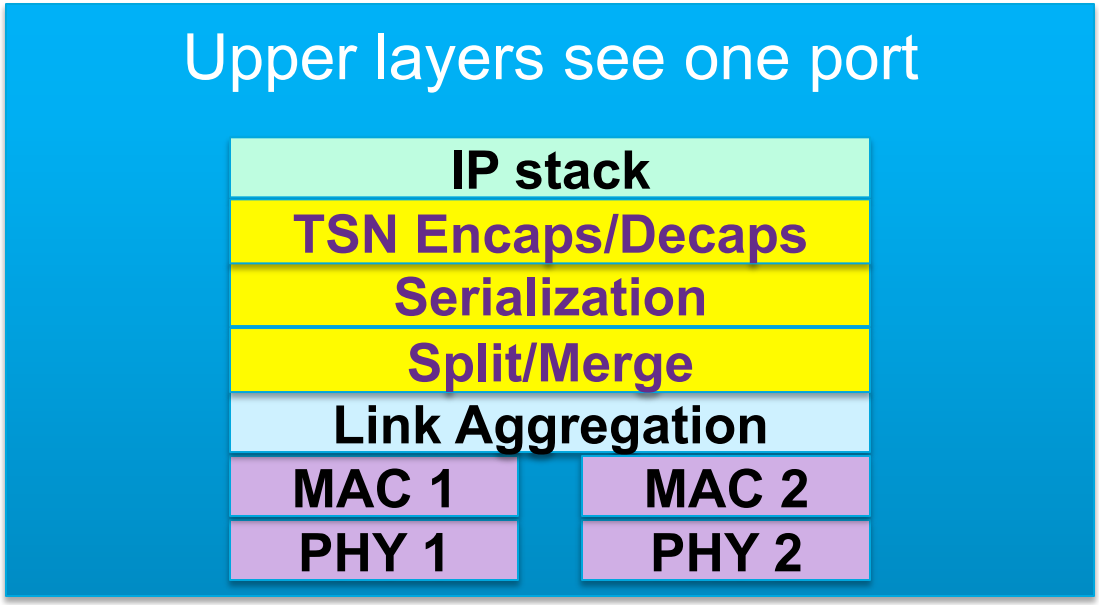


Single-port multiple VLAN host

- Even if the VLAN Tagging layer is not present (i.e., the host is VLAN-unaware), **the TSN Encaps/Decaps function is VLAN-aware**.
- No serialization or split/merge functions were shown in the diagram. They could be present. Often, however, their functions would be proxied by the adjacent bridge, in which case the TSN MAC address provides a great circuit ID.

Dual-port non-relay host (Link Aggregation)

- The **DRNI** model works transparently with Seamless Redundancy. Link Aggregation layer splits regular traffic normally, and splits SR traffic using the circuit label.

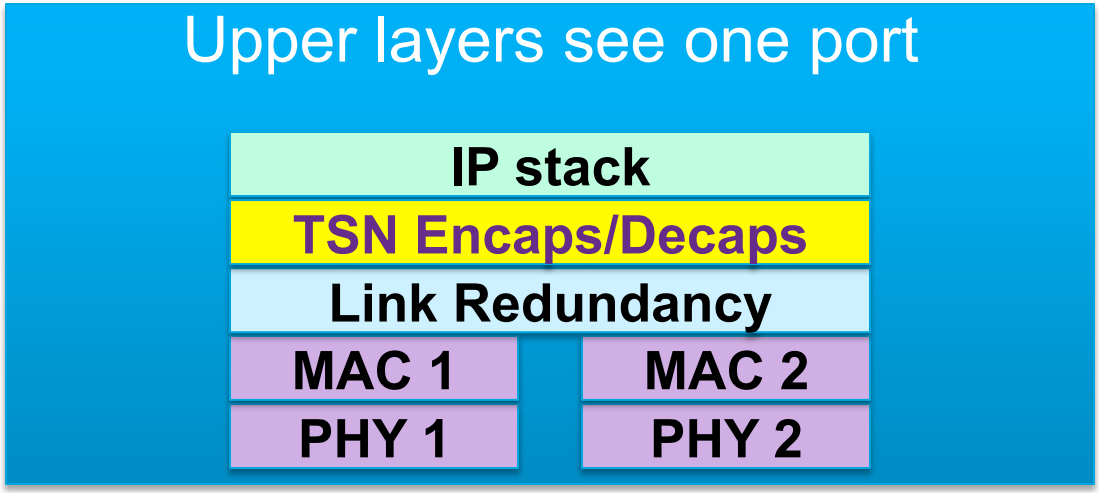


DRNI Host

- The DRNI host cannot be part of a ring – it can only be a dual-ported end station. (That's a use case, not a problem.)
- The non-TSN applications work just fine, without replication, because DRNI works.

Dual-port host (HSR or PRP)

- **IEC 62439-3 HSR/PRP** supports dual-homed hosts along with TSN. The Link Redundancy layer provides Serialization and Split/Merge capabilities, within the limits of the HSR/PRP topology assumptions.

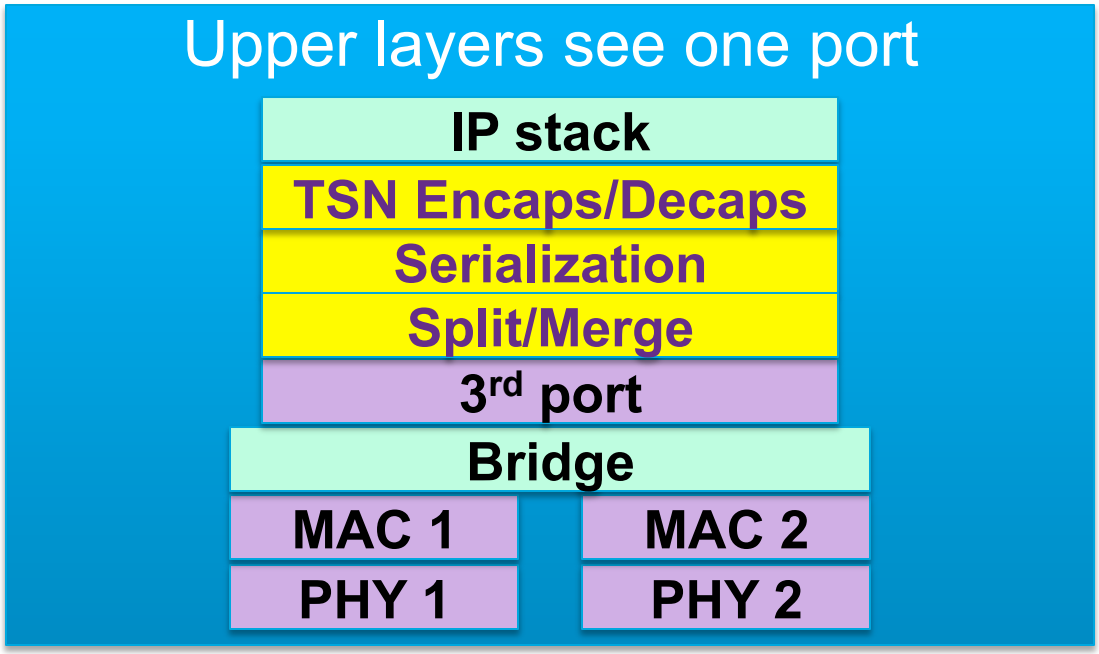


Dual-port host (HSR or PRP)

- As written, HSR supports a host that relays traffic from port to port, and PRP supports a host that does not.
- HSR requires a ring topology.
- PRP requires connections to separate networks.
- As we will see in the next section, both protocols can be easily adapted to work over a general purpose 802.1 network for TSN.

Dual-port relay host (bridged)

- The **Bridge** model works fine. The host stack creates differently-labeled circuits, and the bridge part directs them on different paths.

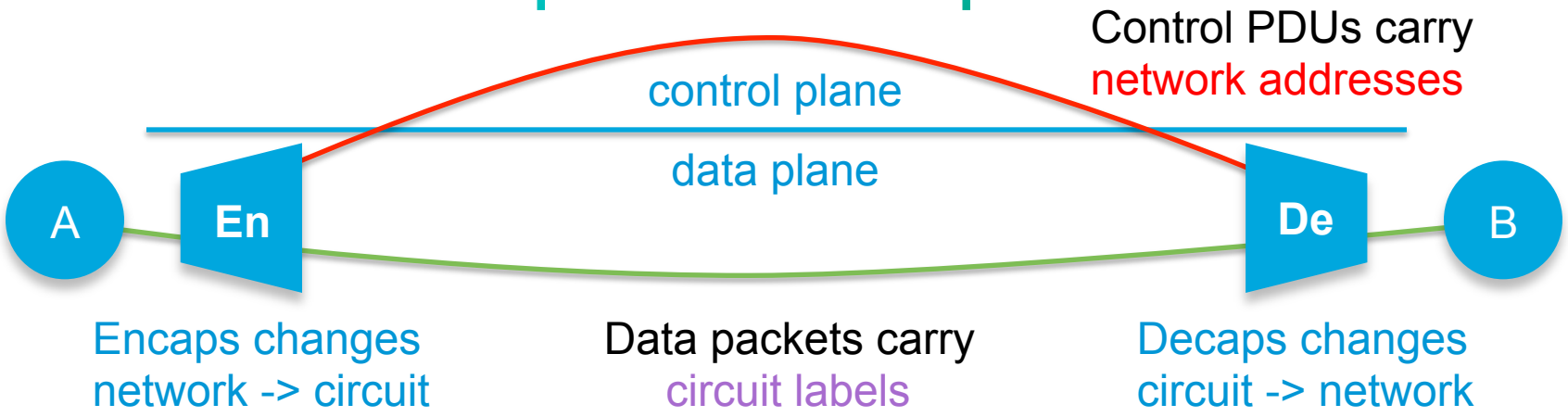


Dual-port relay host (bridged)

- This model should be of particular interest to TSN, as it is the most general, and of course, 802.1 defines bridges.
- It automatically supports rings or dual-homed stations.
- When combined with the Simple 2-port Intermediate System concept, it makes a powerful ring/chain node implementation.

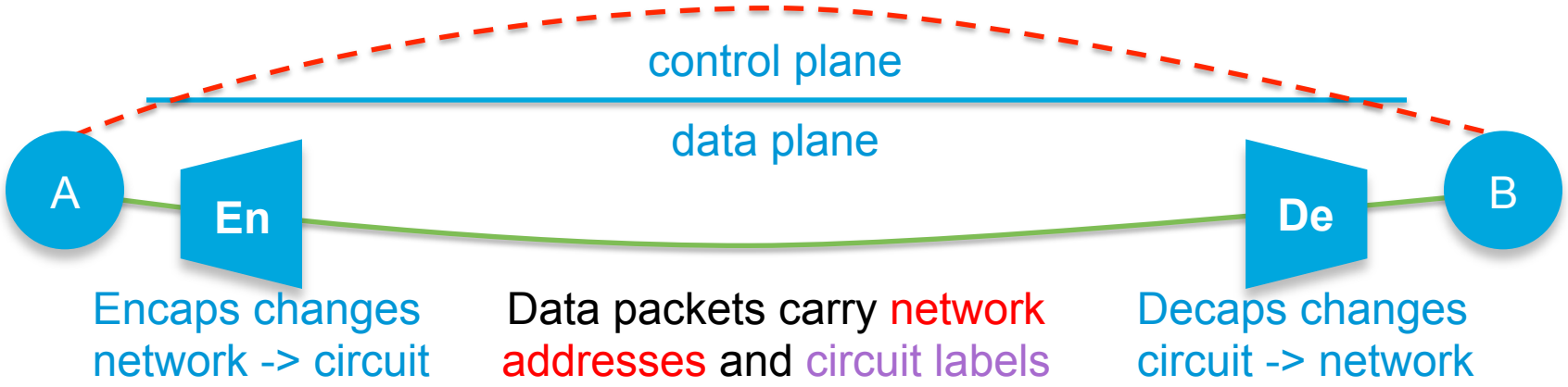
Alternative TSN Encapsulations

Are other encapsulations possible? Yes!



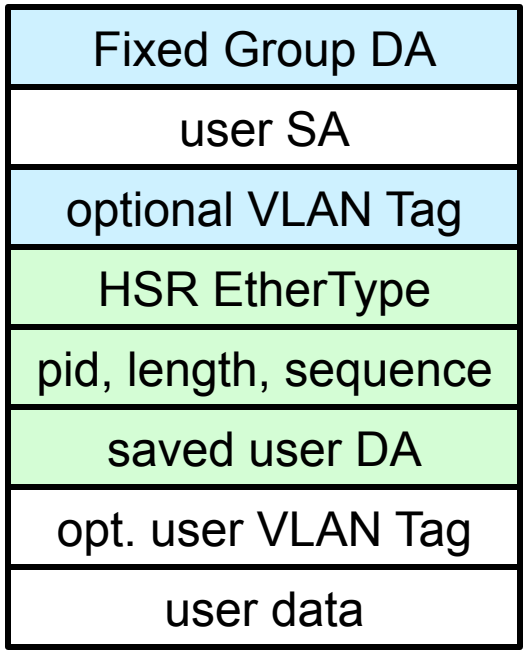
- To use the current AVB format, while making TSN services transparent to the user, the Encapsulation function destroys information, and the Decapsulation function restores it.
- Let's call this **out-of-band tunneling**.

Are other encapsulations possible? Yes!



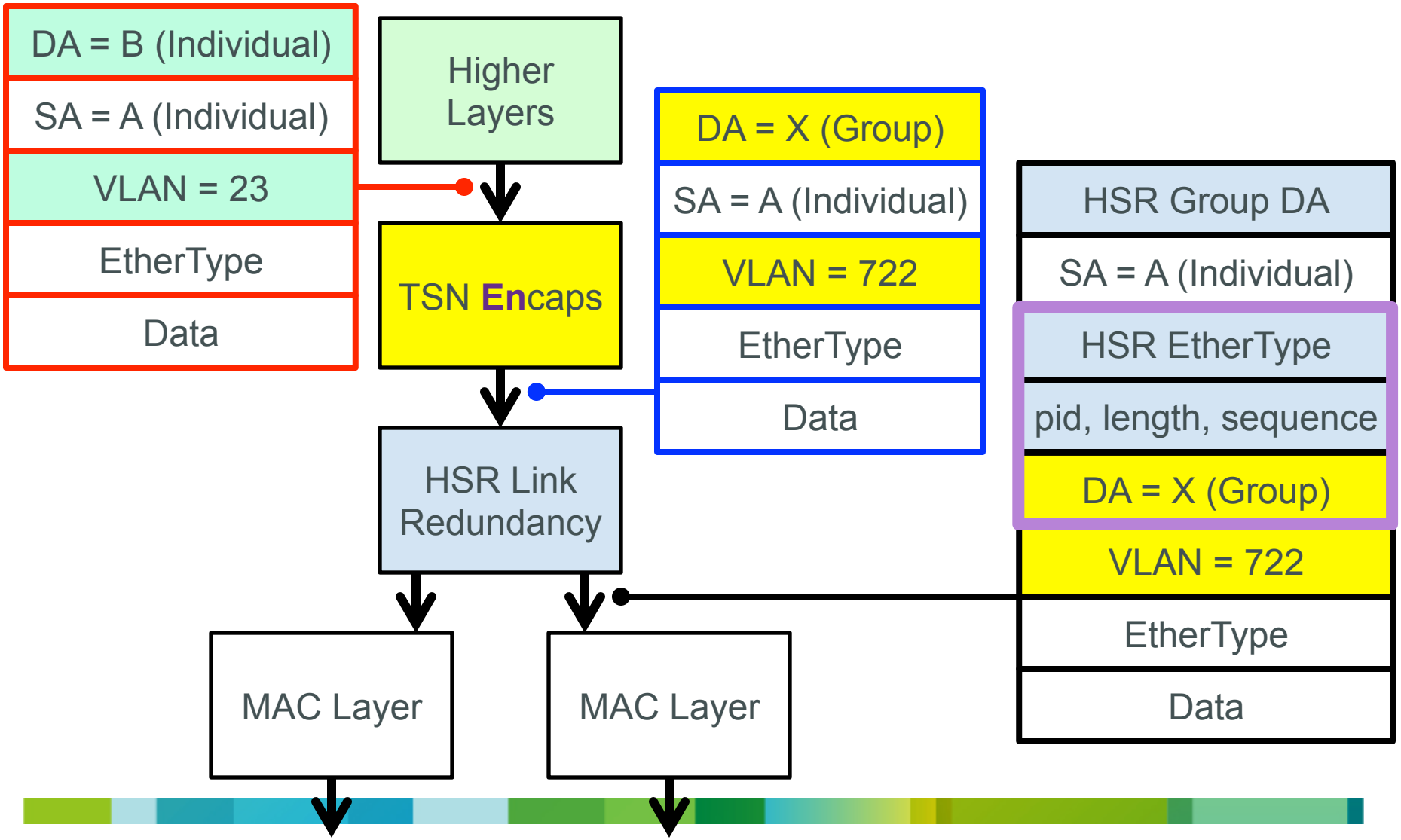
- We could do **in-band tunneling**, and encapsulate the B's address and VLAN in the data frame, itself.
- There are existing, applicable protocols that work both ways.

1. HSR seamless redundancy (1)



- The **IEC 62439-3 High-Speed Seamless Redundancy (HSR)** encapsulation provides in-band tunneling.
- Almost. On the good side:
  - HSR encapsulates the original destination MAC and VLAN, rather than using the data plane.
  - HSR includes a sequence number for seamless redundancy. (That **is** the name of the protocol!)

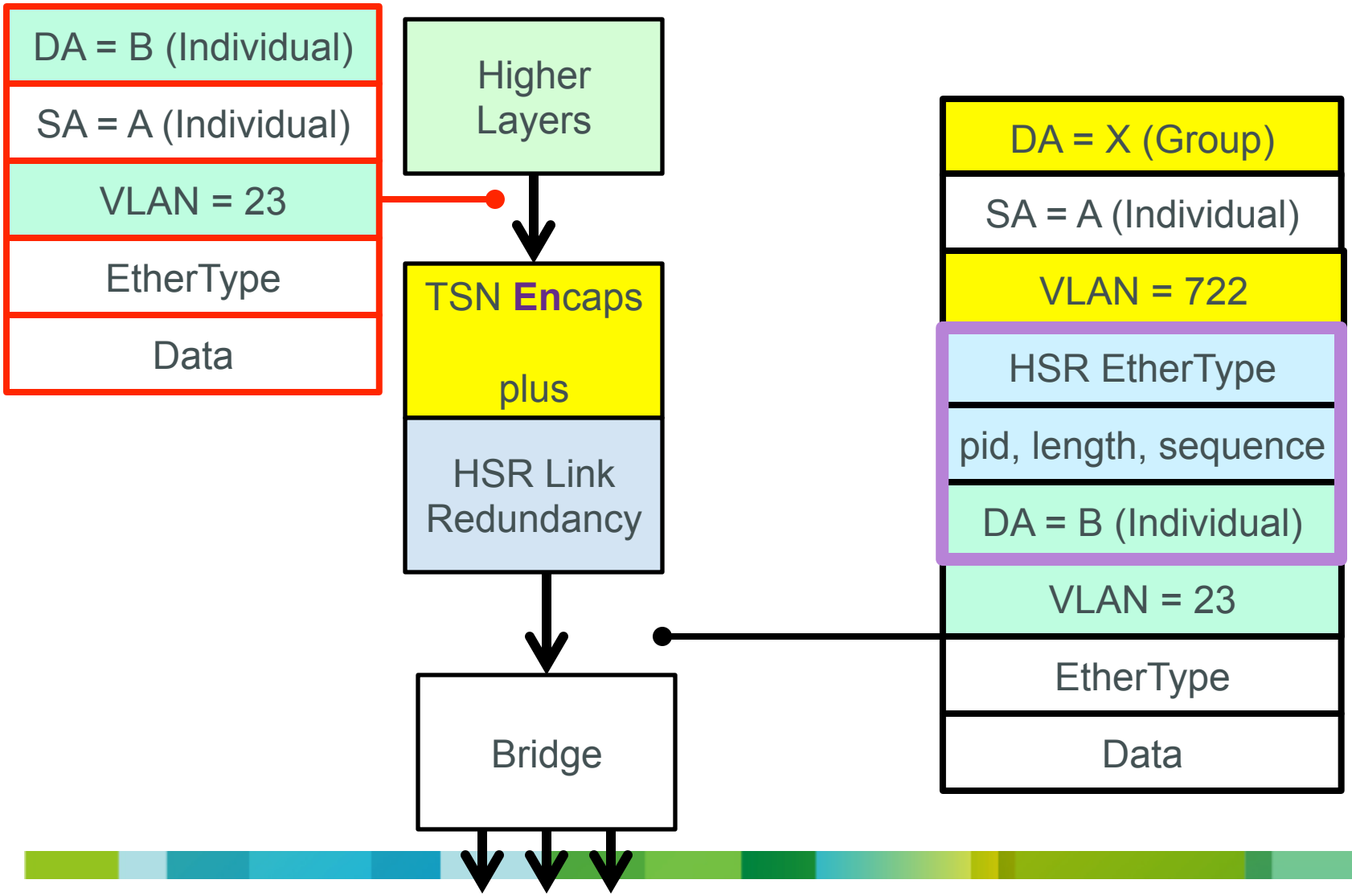
HSR seamless redundancy (1)



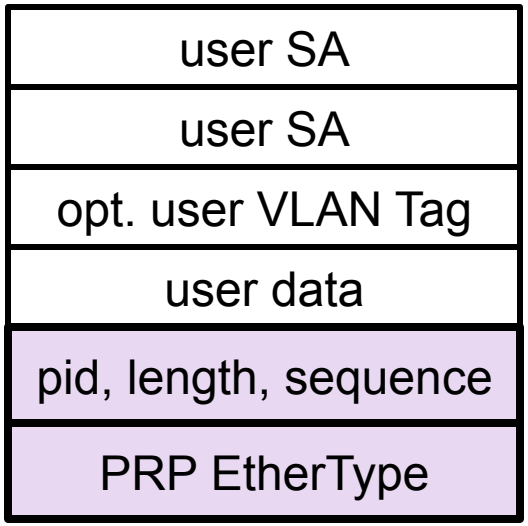
2. HSR seamless redundancy (2)

- Oops! HSR only works if the network is an HSR ring, because the TSN Circuit address has been buried behind the fixed HSR DA.
- To make HSR work over 802.1 networks:
  - One change to the PDU format:
    - We substitute the TSN circuit DA for the HSR fixed DA.
  - One change to the use of that format:
    - The Link Redundancy layer does not forward packets – that is left to a bridge function, below it.

HSR seamless redundancy (2)

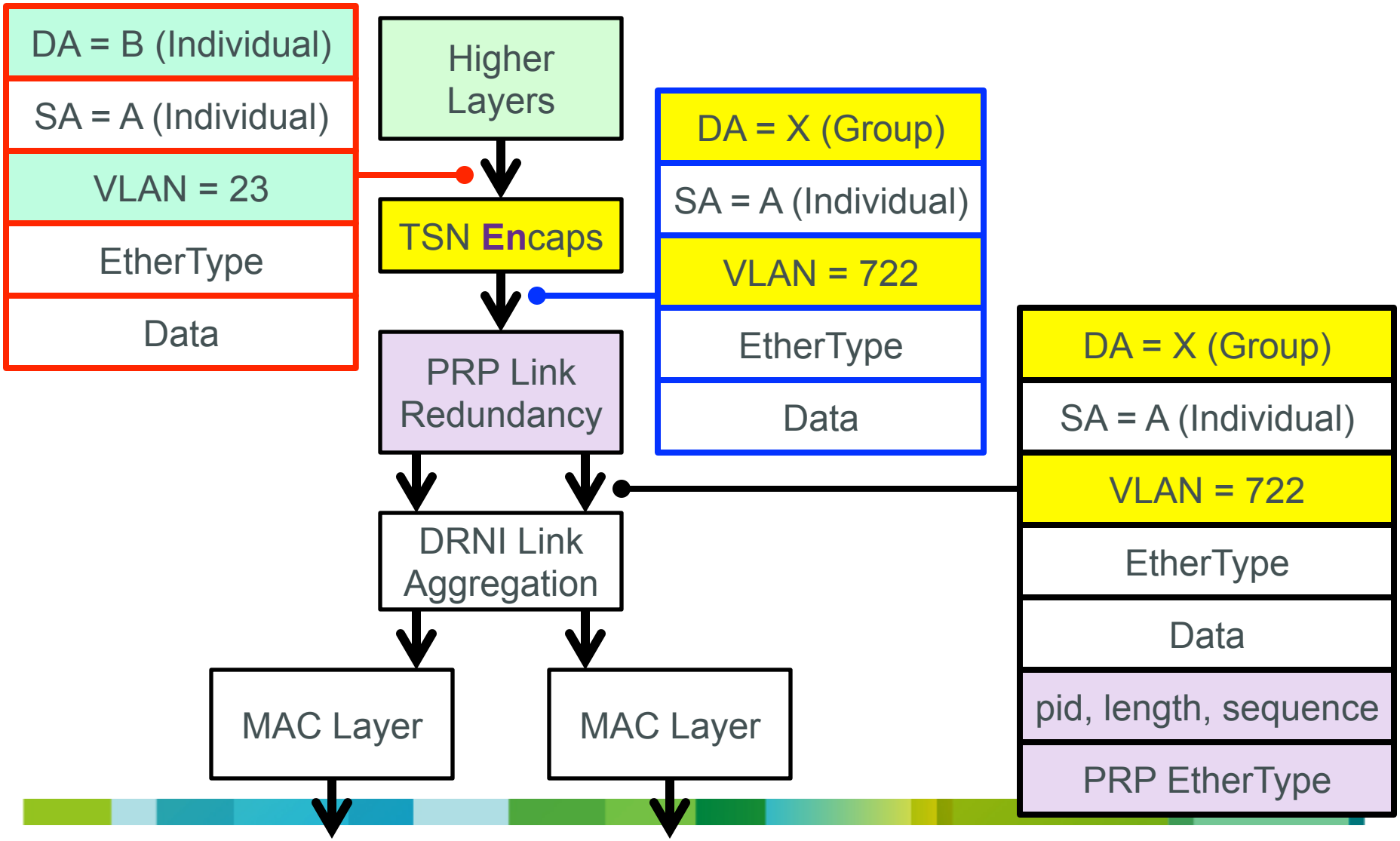


3. PRP seamless redundancy + TSN



- The **IEC 62439-3 Parallel Redundancy Redundancy (PRP)** encapsulation will work with the TSN encapsulation.
- There are serious issues with an 802.3 frame trailer. Let's leave that for another discussion.
- TSN supplies the circuit ID, and PRP supplies the split/merge capability.

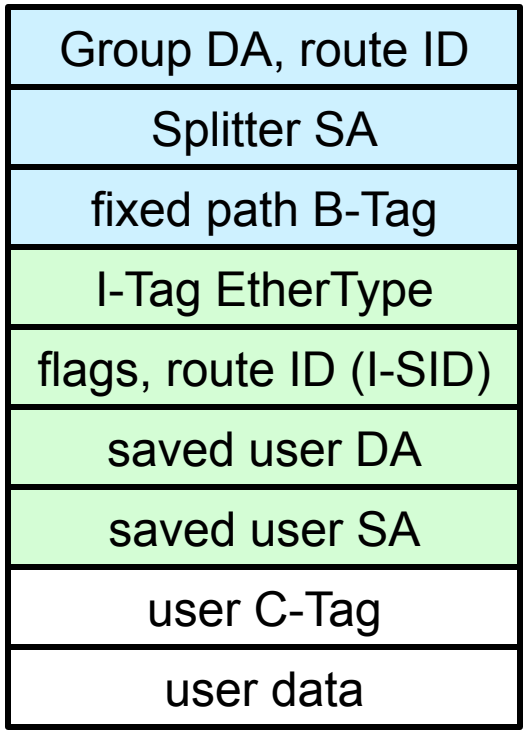
IEC 62439-3 PRP + TSN



IEC 62439-3 PRP + TSN

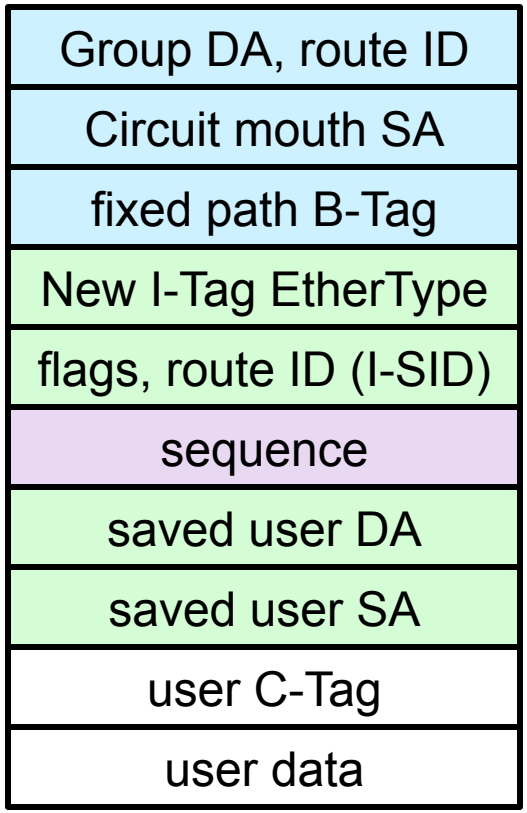
- PRP was intended to work only over separate networks; one port is connected to one network, and the other to another network. But, we can make it work over a single 802.1 network.
- It works, because the TSN circuits provide the logical equivalent of separate networks.
- A DRNI / Link Aggregation unit, as described earlier, would connect this stack to the physical ports.

4. PBB-TE for TSN



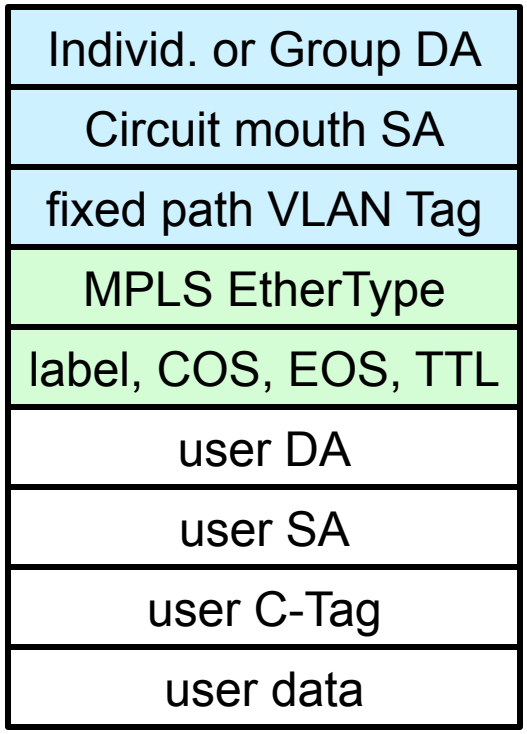
- MAC-in-MAC (802.1ah) solves the stream ID and fixed path problems using in-band tunneling.
  - It encapsulates the data now carried in the control plane.
  - But, it has no flow ID or sequence number for seamless redundancy.

PBB-TE for seamless redundancy



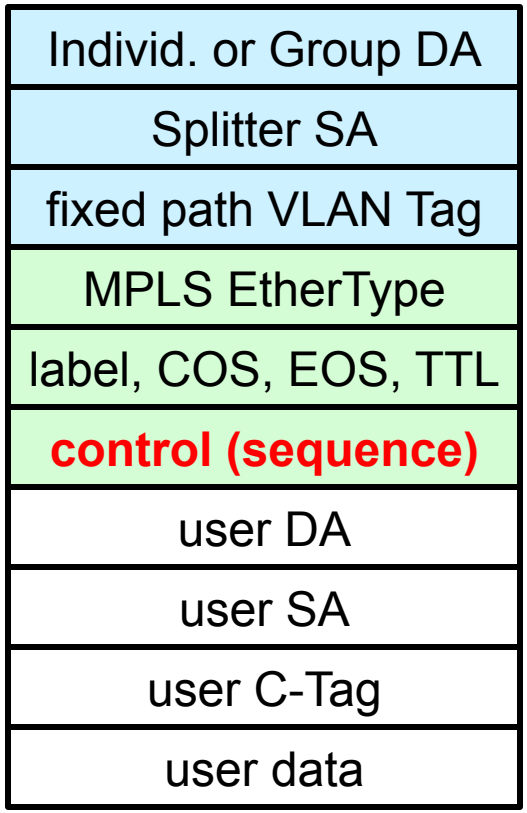
- We must either have an additional tag for the sequence number (not clear where), or a new I-Tag format that includes a sequence number (shown at left).
- Or, just use it for tunneling.

5. MPLS Ethernet Pseudowire



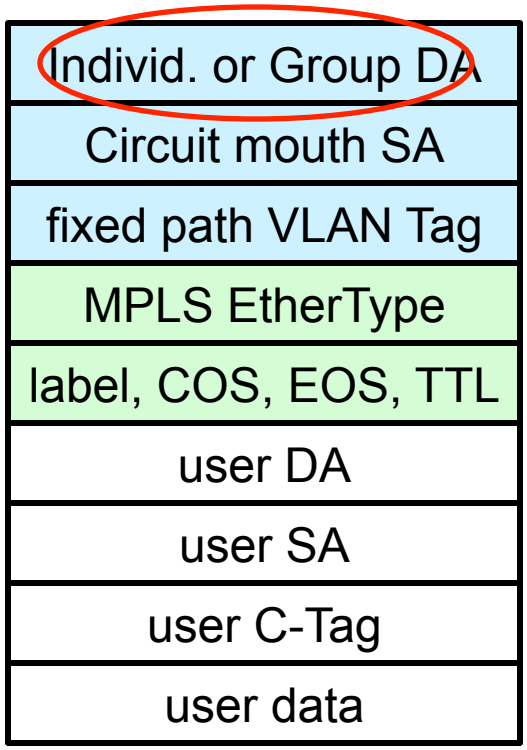
- An **MPLS pseudowire** provides almost exactly the same encapsulation structure as 802.1ah.
- Like the current AVB encapsulation, it provides out-of-band tunneling.

Pseudowires for seamless redundancy



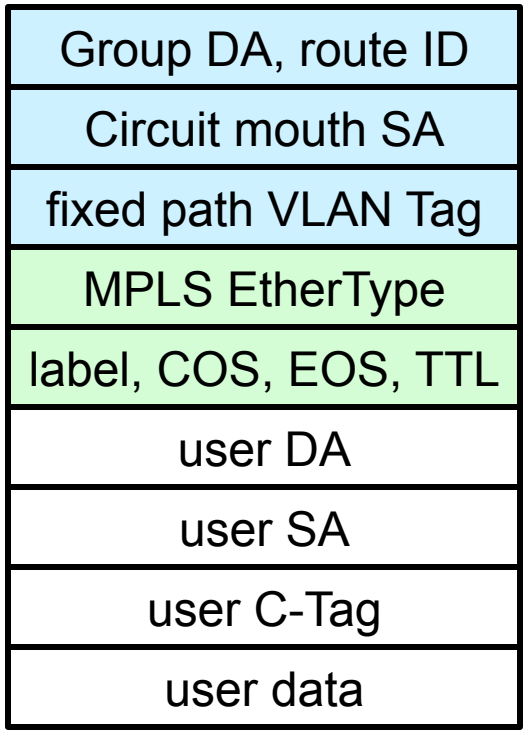
- Plus, Pseudowires have an optional control word that provides a **sequence number for seamless redundancy**.

MPLS Ethernet Pseudowire



- There is one **problem**: MPLS uses the next-hop destination, or a fixed Group DA.

MPLS Ethernet Pseudowire



- There is a **solution**: MPLS uses a fixed Group DA OUI, with the outermost MPLS label value in the low-order bits of the Group DA.
- (Just like PBB-TE.)

6. No protocol at all

- OpenFlow can recognize a stream based on common fields, such as the MAC addresses or IP 5-tuple.
- Any number of bridges have “Access Control Lists” (ACLs) that can inspect a frame and take special action.
- So, one can always leave the original frame intact, and use frame inspection to identify the circuit so that special actions can be taken.
- **But, this layering model is still important, you’re still doing circuits, and P802.1Qcc still has to change.**

Summary

Summary 1/2

- When you get the layering right, everything just works.
  - No need for deep packet inspection. No problems with existing IP stacks, too many VLAN IDs, MAC address learning, fixed paths, or backwards compatibility.
  - We must align P802.1Qcc with proper layering.
  - We must define the TSN Encaps/Decaps function.
- We shouldn't feel bad. HSR/PRP also reinvented Layer 2 networking.

Summary 2/2

- When you get the layering right, several existing protocols support TSN circuits.
  - Properly layered TSN, HSR(1), HSR(2), Pseudowires, or no protocol at all.
  - There are more.
- When you get the layering right, several existing protocols support seamless redundancy, already.
  - HSR (1), HSR (2), PRP + TSN, Pseudowires.
  - There are more.

Thank you.