Proper Layering for the TSN Layer 3 Data Plane

Norman Finn, Rudy Klecka, Pascal Thubert, Cisco Systems

Version 1

Jan. 17, 2014

This presentation

- This is part 2 of a two-part presentation.
- Part 1, tsn-nfinn-L2-Data-Plane-0114-v02, introduces concepts on which this presentation depends. Part 1 should be read before Part 2.

Terminology note

- We will often use "**host**," not "end station."
  - "End station" is too 802.1-centric. A router is an end station to L2, which confuses things.
  - But, remember that we're not always talking about a device that meets the Host Requirements RFC.
- We will often use "**node**" or "**network node**", not "bridge," "router," "brouter," "switch," etc., since it usually doesn't matter what the device actually is.
- We will often use "**packet**," and not "frame," unless we are talking specifically about an Ethernet frame.

Why does IEEE 802.1 care about L3?

- This author perceives a disconnect between the current AVB/TSN protocol suite and the long-term needs of a broader marketplace.

Why does IEEE 802.1 care about L3?

- This is best illustrated by our work to date on P802.1Qcc. Among our goals are:
  - Support a network of tens of thousands of end stations with thousands of AVB/TSN flows.
  - Support the convergence of ordinary traffic and mission-critical traffic.
- On the other hand:
  - Our scheme only works over a flat bridged network.
  - There is a consensus in the industry that a flat bridged network of that size would collapse under (among other things) the broadcast load, especially if converged with ordinary traffic.

Going forward

- A number of people have spent considerable effort over the last few months to figure out how to reconcile this conflict.
- We have started by asking, “How can I most easily adapt existing applications, existing, host stacks, and existing networking equipment to take advantage of TSN?”
- This means using existing networking models, existing layering models, and to the greatest extent possible, existing protocols.

Driving assumption

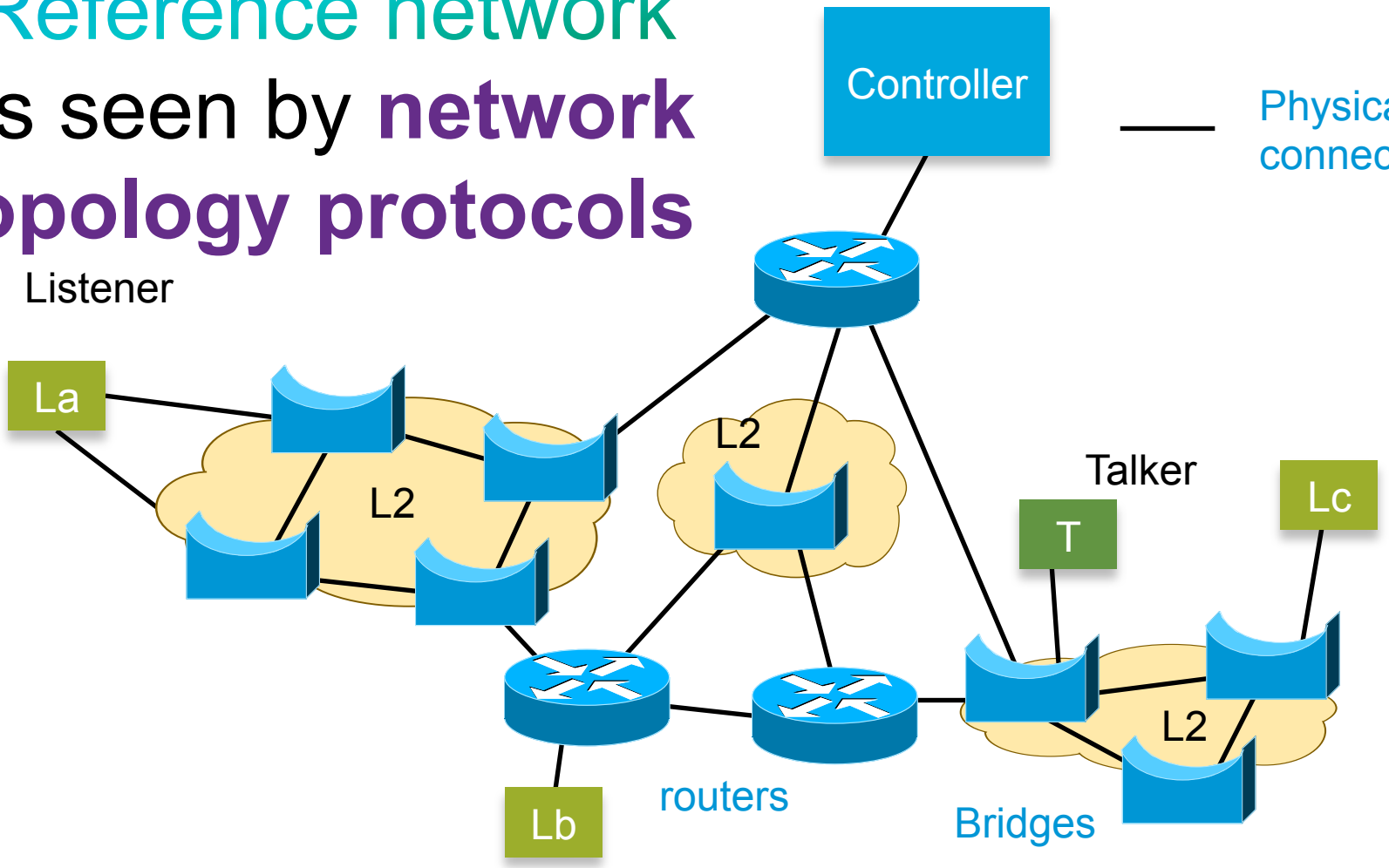
- **The goal of the TSN TG should be to write standards for new Quality of Service (QoS) classes for high reliability and low latency, that offer incremental benefit to any network, whether L2, L3, or mixed, that follow established general-purpose operational models.**
- **To the extent that TSN standards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

Which leads to a Reference Network

- The network is some combination of bridges and routers, illustrated on the following slide, that follow the existing norms for networking.
- We have the usual plethora of protocols running (including, perhaps, L2 protocols from ITU-T, ODVA, or ISO, instead of IEEE).

Reference network

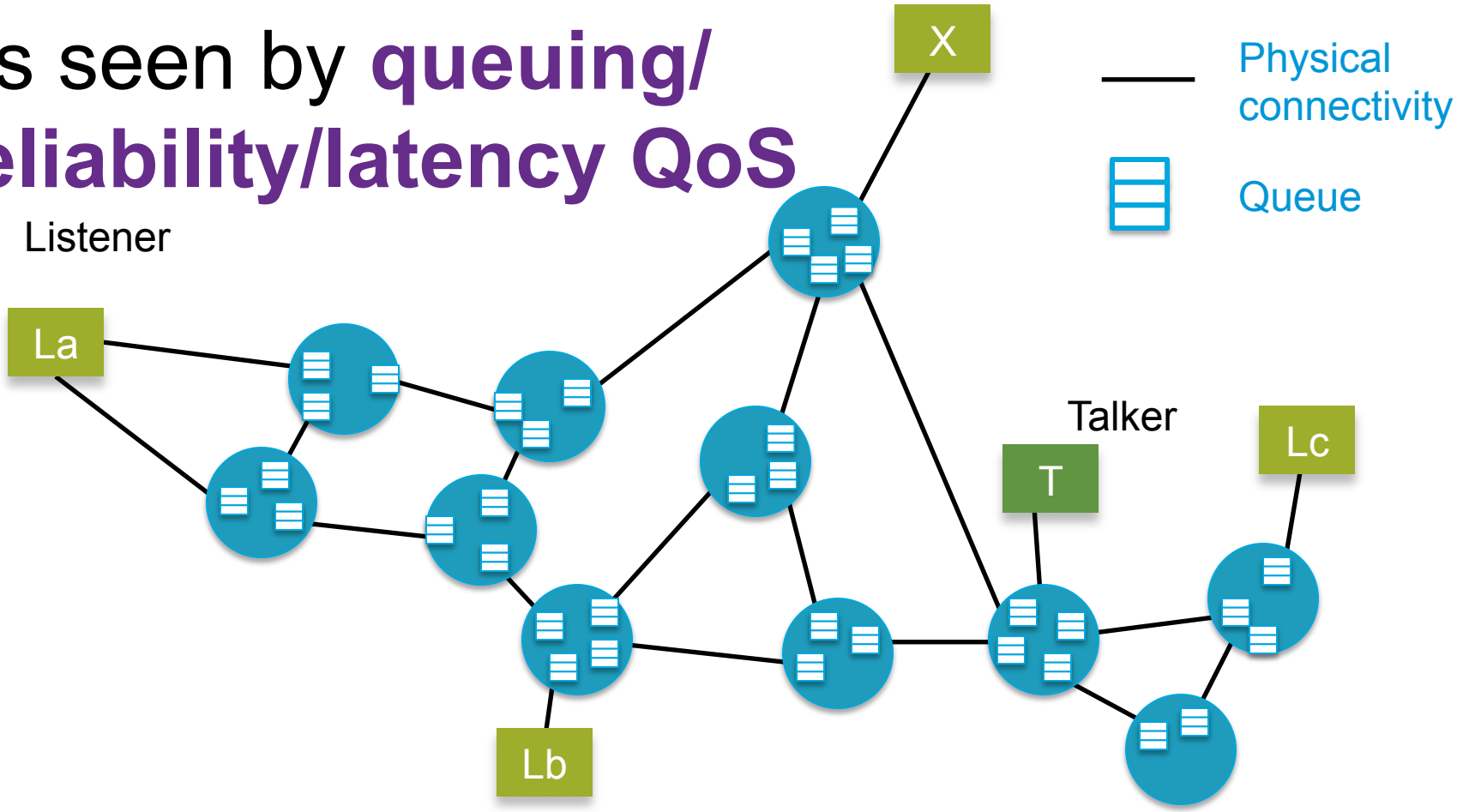
As seen by **network topology protocols**



- Gazillions of complex protocols

Reference network

As seen by **queuing/ reliability/latency QoS**



- Just nodes, queues, and wires!!

Just nodes, queues, and wires

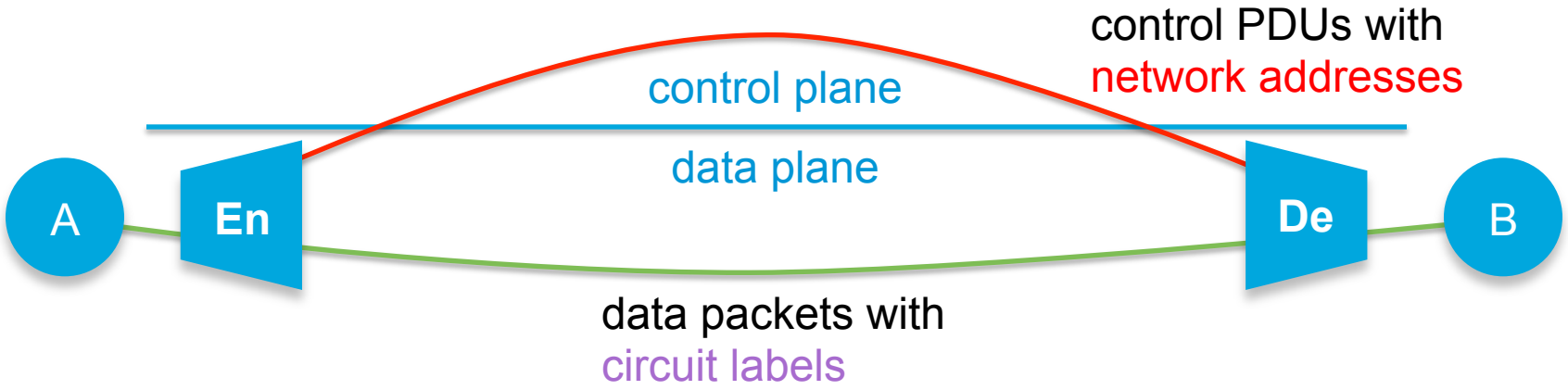
- To build a circuit that makes guarantees, every box along the path has to participate in the circuit.
  - We've known that for some time, in AVB/TSN.
- Therefore, in a mixed L2/L3 network, every box along the path has to participate in the data plane and in the control plane.

Outline

- Part 1 covers a number of issues, mostly in the Layer 2 end-to-end world, summarized here in two slides.
- This Part 2 covers:
  1. Peering principles
  2. Circuit identification
  3. MPLS and Pseudowires
  4. L2 Peers vs. L3 Peers
  5. Summary

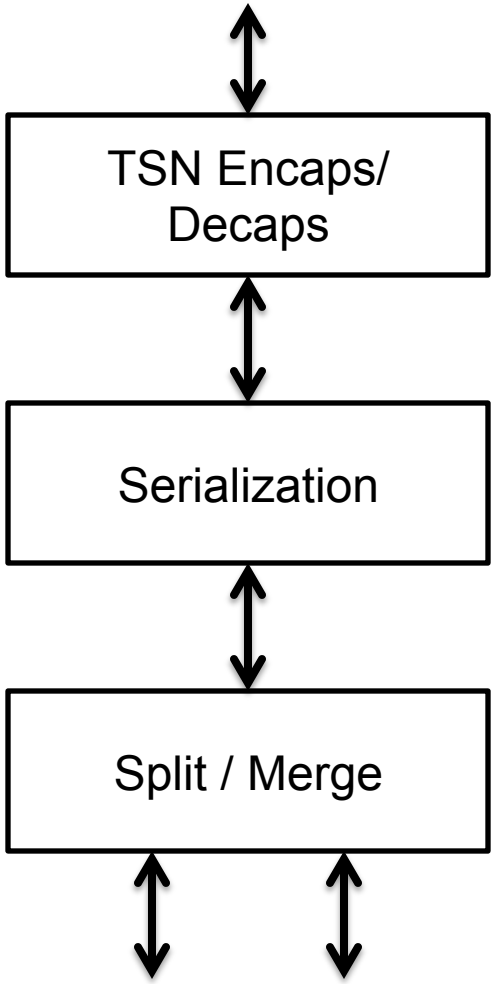
L2 Layering

Reminder of conclusions from Part 1



- From a pure layering standpoint, **this is what we're doing right now** with AVB/TSN.
- A **TSN Encaps** function is substituting the network addresses of the endpoints with circuit labels (tunnel addresses), and at the end of the circuit, a **TSN Decaps** function restores them.

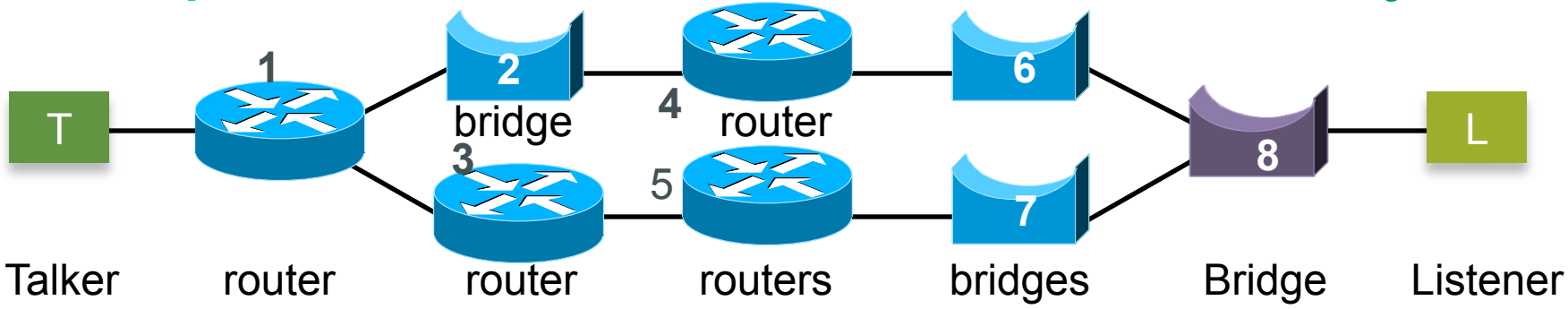
Functional elements required for TSN



- Individual flows must be identified, and packets encapsulated.
- Packets must be serialized.
- Streams must be split among multiple paths on transmission, and merged on receipt.

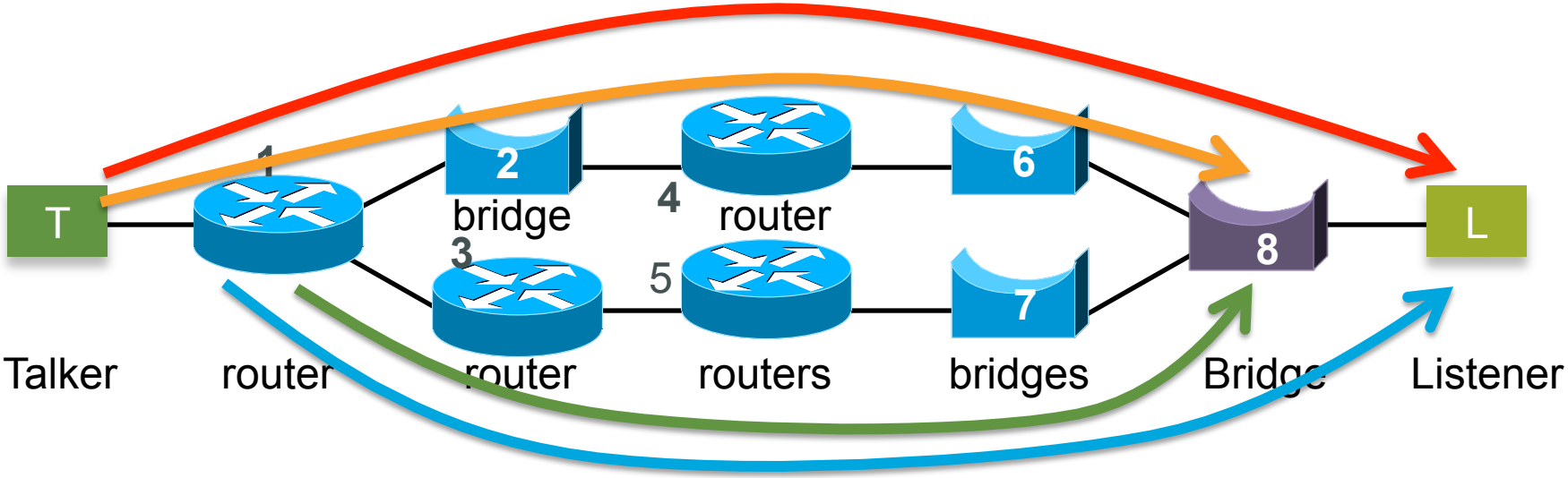
1. Peering

Who peers with whom, at what sublayer?



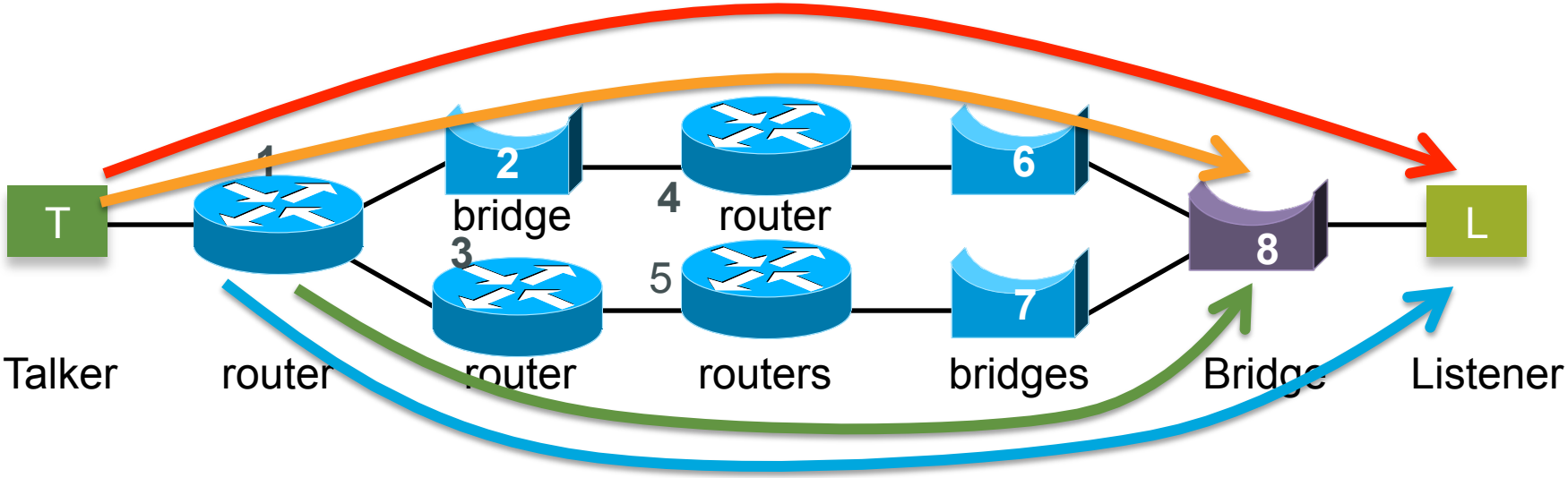
- In a complex network, **the split/merge peers in the TSN stack must operate at the same sublayer**.
- The TSN stack resides in a system. It does not matter with what Layer that system primarily concerns itself.

Who peers with whom, at what sublayer?



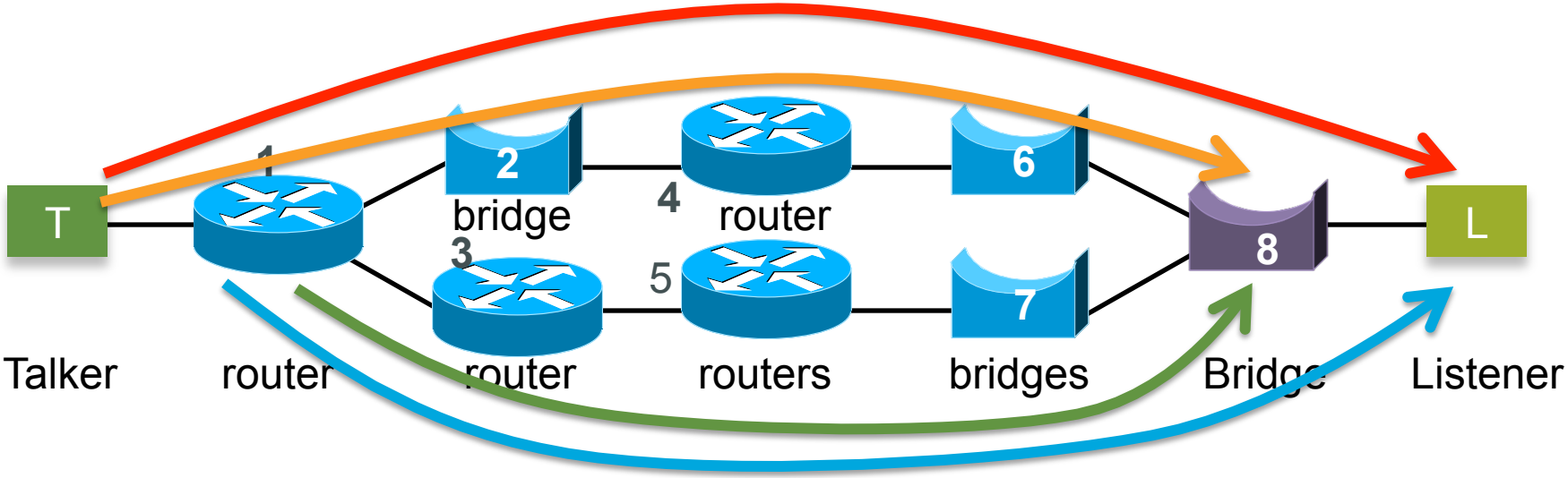
- This is especially important when the network nodes are performing proxy functions for hosts or for other network nodes.
  - This diagram shows where split/merge peers can reside. It does not show multiple paths for one flow.

Who peers with whom, at what sublayer?



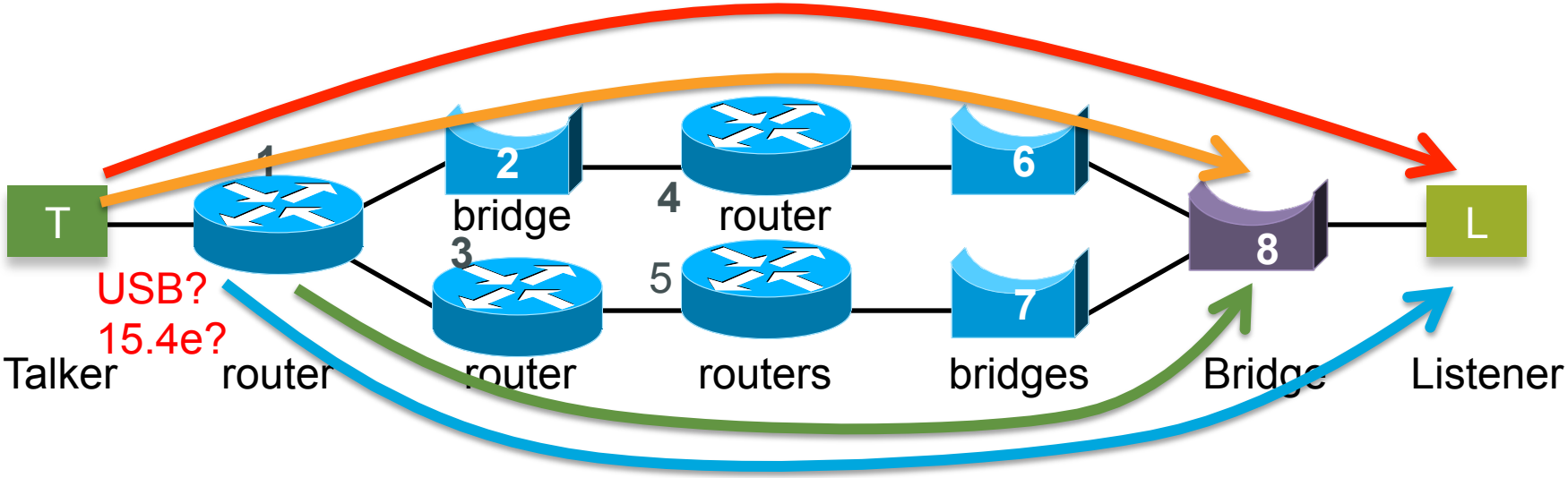
- **None of the split/merge peers, here can be at the Ethernet layer.** Not even the end-to-end Talker-to-Listener peers. Why?
- It's **not** because you have a router at one end and a bridge at the other; the system's Layer association doesn't matter.

Who peers with whom, at what sublayer?



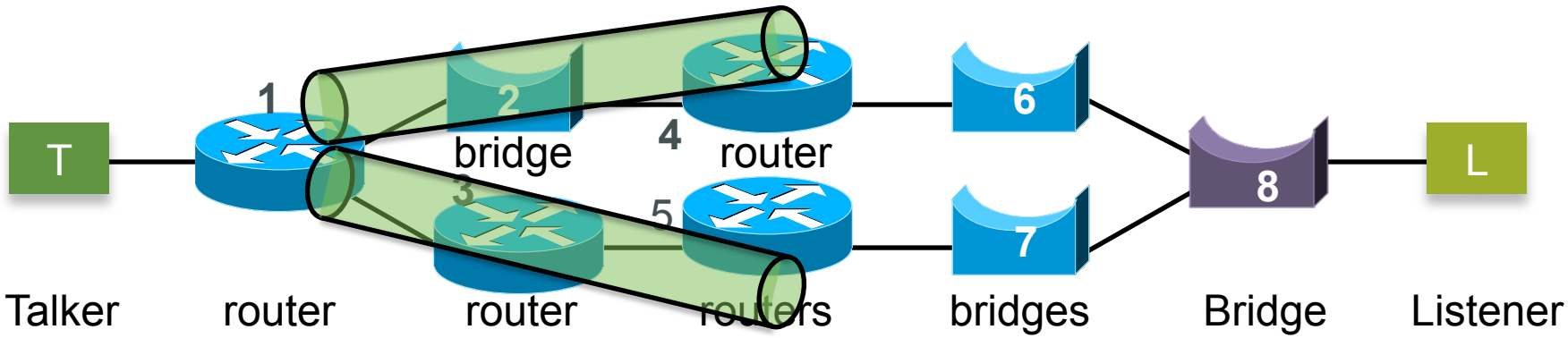
- None of the split/merge peers, here can be at the Ethernet layer. Not even the end-to-end Talker-to-Listener peers. Why?
- Because the **Talker and Listener do not have an Ethernet relationship in the base network**.

Who peers with whom, at what sublayer?



- For example, the Listener is Ethernet; it's connected to a bridge.
- But the Talker may be connected to its router via USB, or via IEEE 802.15.4e.

Who peers with whom, at what sublayer?



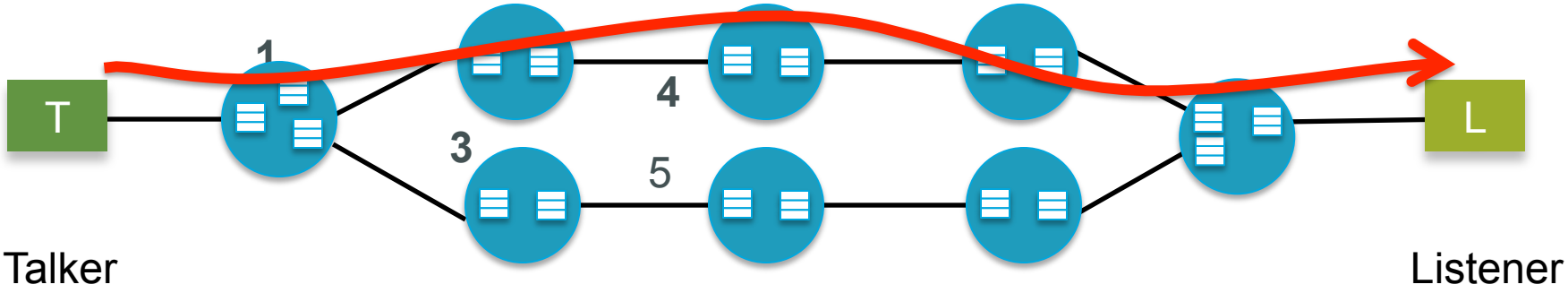
- One can always create tunnels using Ethernet-over-XYZ technology, and make the Talker and Listener Ethernet peers.
- But, again, **if the whole world is Ethernet, then the world doesn't work**.

The current TSN stack peers Ethernet

- At present, the TSN stack peers only Ethernet protocols.
- This is not sufficient for a mixed L2/L3 network.
- So, we need a TSN stack that peers at higher layers.
- As we will see, this is not as hard, or as alien to TSN's current work, as it may seem at first.

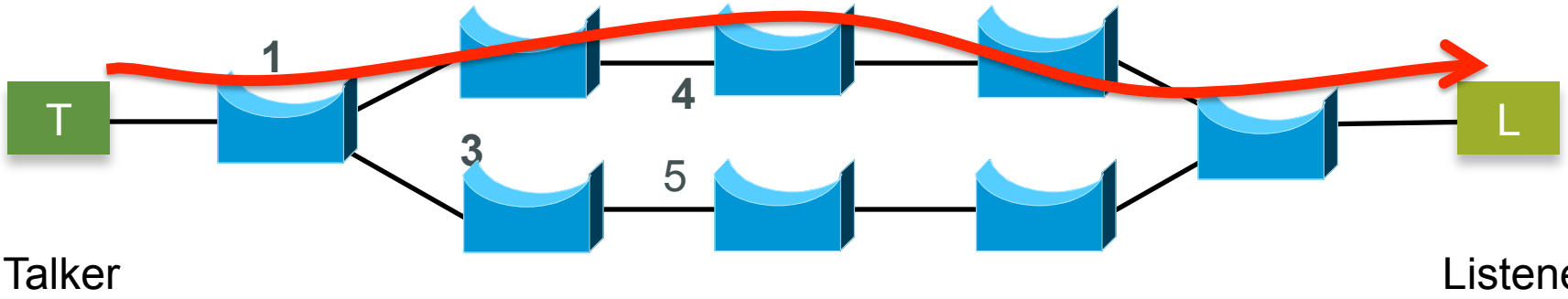
2. Circuit identification

Circuit identification



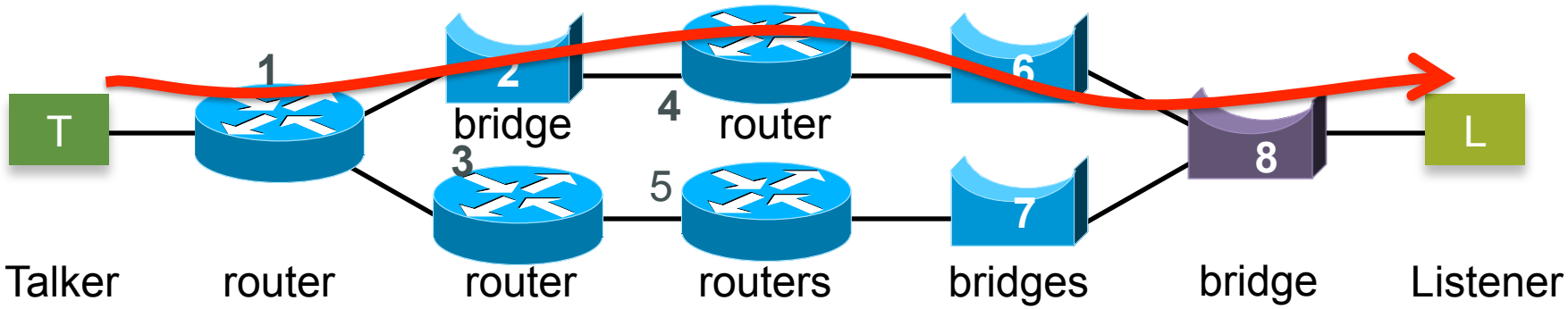
- Every network node along the path must be able to recognize the circuit, in order to provide it with the per-circuit services it requires.

Circuit identification



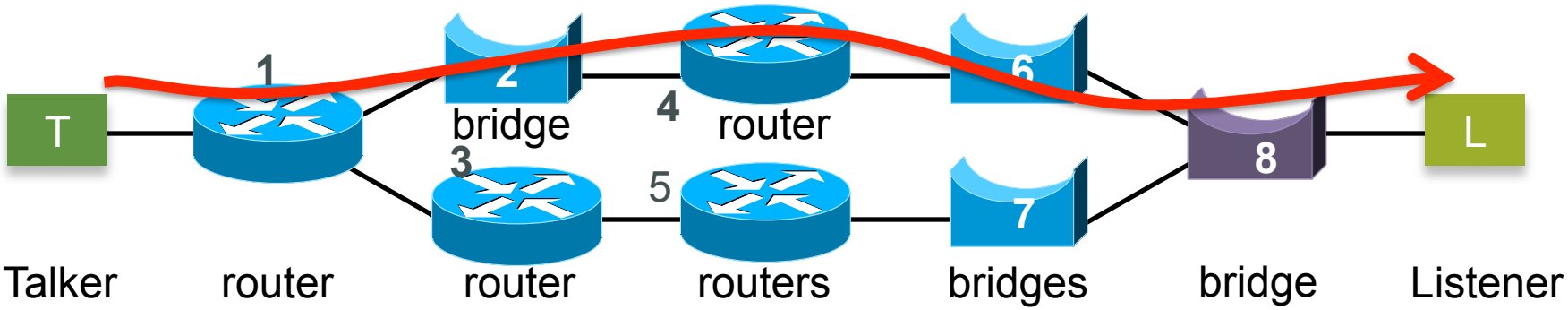
- For all-Ethernet networks, we have several candidates for circuit identification (from Part 1):
  - Current AVB/TSN frame format (VLAN + destination MAC address).
  - HSR (VLAN + destination MAC address).
  - PBB-TE (VLAN + destination MAC address).
  - Ethernet over pseudowire (VLAN + destination MAC address).

Circuit identification



- But when the boxes are mixed L2/L3, {VLAN, MAC address} pairs don't work:
  - There are boxes that are not Ethernet.
  - There are no end-to-end Ethernet addresses!

Circuit identification



- What we need is to have the circuit ID available to every box. How? Options:
  - Deep packet inspection? Not the first choice.
    - It's difficult and expensive.
    - Security can make it impossible.
  - An L2 tag? No!
    - There is no L2 that runs end-to-end.

There is no L2 tag that runs end-to-end?

- Well, actually, there is.
- It's called, Multi-Protocol Label Switching (MPLS).
- Remember Part 1? The pseudowire format has a circuit label buried in it, which was copied to the Ethernet DA.

3. MPLS and Pseudowires

Multi-Protocol Label Stack

- Each MPLS label is 32 bits, including a 20-bit "label value" that identifies a flow for the purposes of routing.
- MPLS labels can be stacked to any depth, even more so than IEEE 802.1 tags.
- An MPLS label is marked whether it is the last label in the stack or not (End Of Stack = EOS bit):
  - After a not-the-last label is another label.
  - What's after the last label is identified by the label.
  - An EtherType is **not** needed between or after labels.

Label Switched Paths (LSP)

- An LSP is a path through the network from a Label Edge Router (LER) through some number of Label Switching Routers (LSRs, an MPLS “switch”) to one or more destination LERs.
- At every hop:
  - The label value tells the LSR how to forward the packet.
  - At each hop, the outermost label value changes and the TTL (8 of the 32 bits in the label) is decremented.

Setting up LSPs

- There are many protocols for setting up LSPs.
- For example, an LSP can be set up to carry IPgrams.
  - This can be the same path that the IPgram would have followed.
  - This can be a path that is different from what the routing protocol would normally do with the packet. E.g., a Path Computation Element (PCE) can pick a path.
  - The LSP can change to follow the topology, or it can be fixed until explicitly torn down.

Label stacks

- The network pushes and pops labels as flows enter or leave layered LSPs (tunnels). An LER (edge) function is at the mouth of every LSP.
- Note that this paradigm supports scaling up to huge numbers of (AVB/TSN) streams:
  - You can aggregate bundle of streams by pushing an extra label.
  - This can be treated as a single flow with a bandwidth equal to (or greater than) the sum of its constituent flows.

What follows the last label

- **Anything** that the end points have the means to agree upon. This is the power of MPLS! E.g.:
  - A bare IPgram.
  - An Ethernet or Frame Relay frame.
- The endpoints creating the Label Switched Path decide, though the protocol used to create the LSP, what they are encapsulating.
- Typically,
  - the **next-to-last** label value routes the packet, and
  - The **last** label Identifies the format of what follows the label stack.

MPLS over Ethernet frame format

- The outermost MPLS label governs the progress of the packet through the LSRs.
- In current MPLS-over-Ethernet, the DA is the address, Individual or Group, of the MPLS device(s) intended to receive the packet, perhaps over a Bridged LAN.
- MPLS-over-X is defined for all X.

There is only one thing missing …

- The MPLS label identifies, to the LSRs, the flow to which the packet belongs, for both routing and QoS.
- But, the Bridges need to know this, also.
- But, we have a TSN Encaps/Decaps layer!
  - So, whatever destination MAC address and VLAN would be used, normally, to carry the MPLS-labeled packets, we can change them, in order to carry it over a TSN circuit to its destination.

A choice – all will work

Bridges do MPLS

- The label value changes at each hop.
- Since things are properly layered, we can't prevent this from working or from being implemeted.

MPLS uses a fixed Group + label value DA

- The MPLS frame is encapsulated via TSN.

Pseudowires

- There is a class of “what follows the last label” that is supported by control protocols and in lots of ASICs, called “pseudowires.”
- An essential feature of a pseudowire is that it can guarantee ordered delivery.
  - A pseudowire has a control word following the label, and preceding the payload.
  - This control word carries a sequence number.
- Hence, the mention of Ethernet pseudowires in Part 1.
- **But, wait! There’s more!**

Pseudowires

- It is an interesting fact that the most common algorithm for pseudowires to eliminate out-of-order deliveries, simply discarding out-of-sequence packets, also happens to eliminate duplicates for seamless redundancy.
  - It works fine, in the case where transmissions are infrequent relative to delivery delay, which is the industrial use case.
  - It contains sufficient information to enable TSN to define algorithms suitable for high-volume streams, should we choose to do so.

How many labels on a pseudowire?

- “Naked” pseudowires, where the outermost label is also the last label, are not encouraged by many people in IETF, although there are cases where they are appropriate.
- In essence, the pseudowire label is unambiguously the flow ID, and IETF prefers that this be wrapped in a label that is unambiguously a route ID.
- But, this is a presentation, not a standard. We’ll work that out.

Layering

- The Internet Protocol, we can safely say, is a Layer 3 protocol.
- People call VLAN Bridging Layer 2. Many would argue with this assessment.
- Most people consider MPLS a Layer 2 with no single Layer 1.

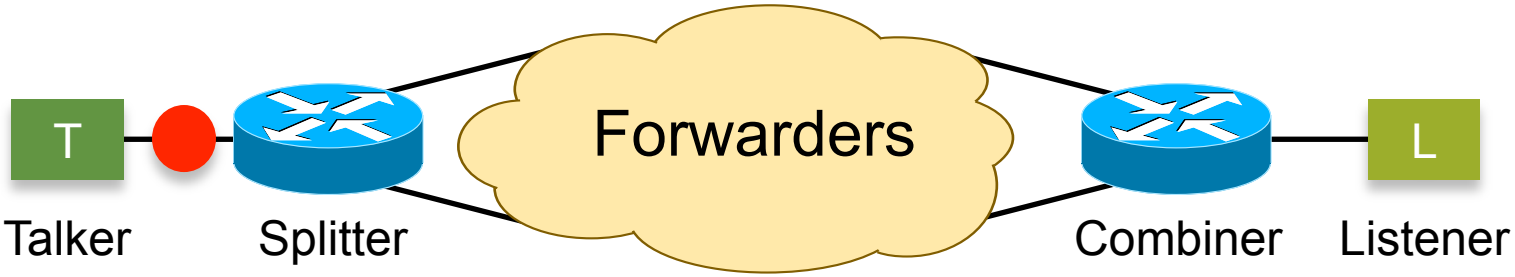
Layering

- What we can say is that, from top (Layer 7) to bottom (Layer 1) the natural layering is:
  - Internet Protocol
  - Pseudowires
  - MPLS
  - Bridged LANs
- Of course, you can always have X over Y over Z over X over Y over P over X over Q.

Are TSN features applicable to MPLS?

- Certainly!
- Note that identifying individual flows in the data plane is necessary for AVB for basic flow reservation and cancellation (route ID).
- The MPLS label provides:
  1. A route ID
  2. A flow ID, where separation is required
  3. A 3-bit priority (COS = Class Of Service bits)

Fixed routes: L3 end-to-end

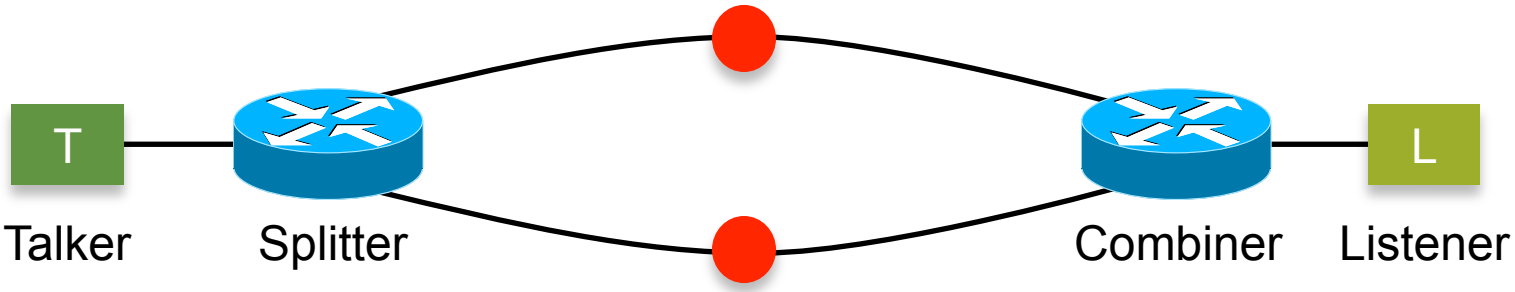


user packet

- How do Splitter and Combiner communicate in the L3/MPLS/pseudowire world?

Original packet

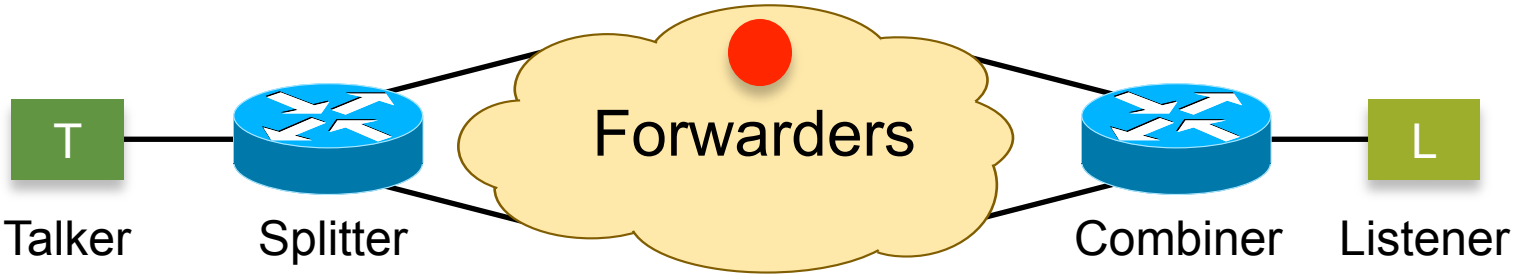
Fixed routes: Mixed L2/L3 end-to-end



pseudowire label
control word
user packet

- The Splitter adds a pseudowire label and control word.
- The Combiner eliminates the control word, and passes on the user data.

Fixed routes: Mixed L2/L3 end-to-end



routing label
pseudowire label
control word
user packet

- To get through an MPLS cloud, an extra MPLS label would be necessary, because the pseudowire label typically has end-to-end meaning.

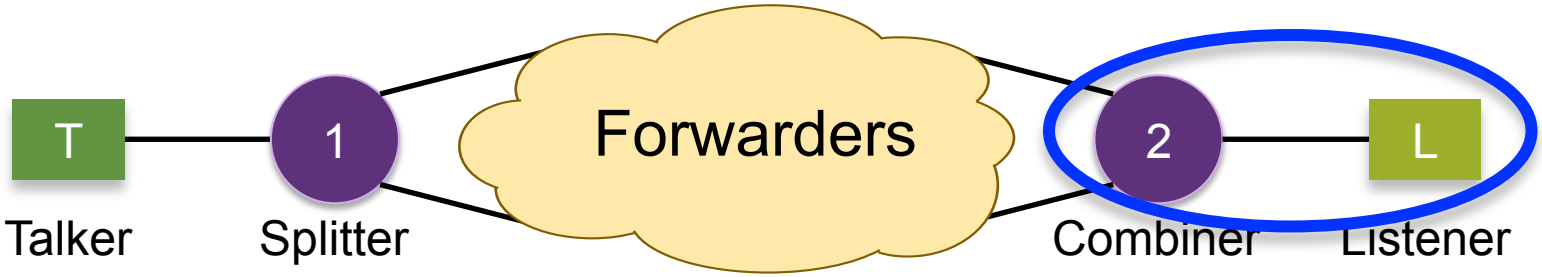
4. L2 Peers vs. L3 Peers

4. L2 peers vs. L3 peers

- Given all that discussion, this author claims that there are only two cases of interest:

1. The Talker and Listener are peers at Layer 3 or above, for example, IP, CLNS, TCP Port 21, or IEEE 1722.
2. The Talker and Listener are using a protocol that is tied to a particular Layer 2 technology, e.g., Ethernet, **and** they are peers on the same (real or virtual) Layer 2 network.

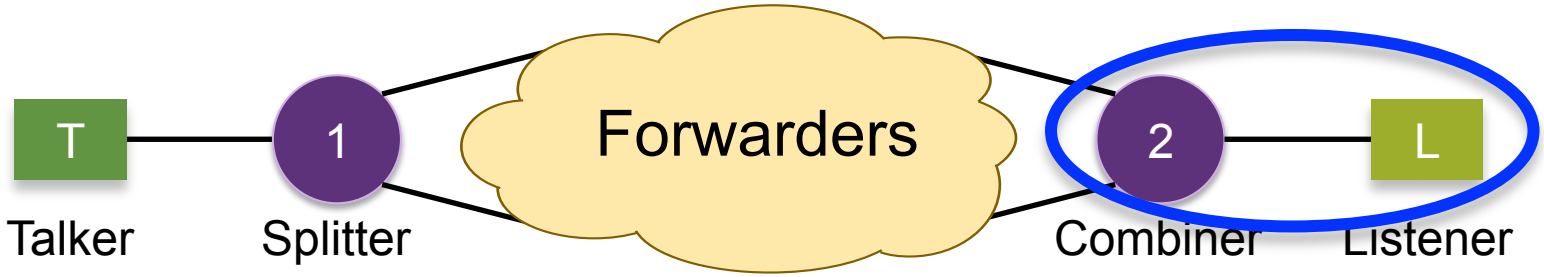
1: T/L are L3 peers. 2: T/L are L2 peers.



Original Layer X payload

- Splitter and Talker, or Combiner and Listener, may or may not be combined into the same system.
- We **don't care** whether Splitter and Combiner are bridges or routers.

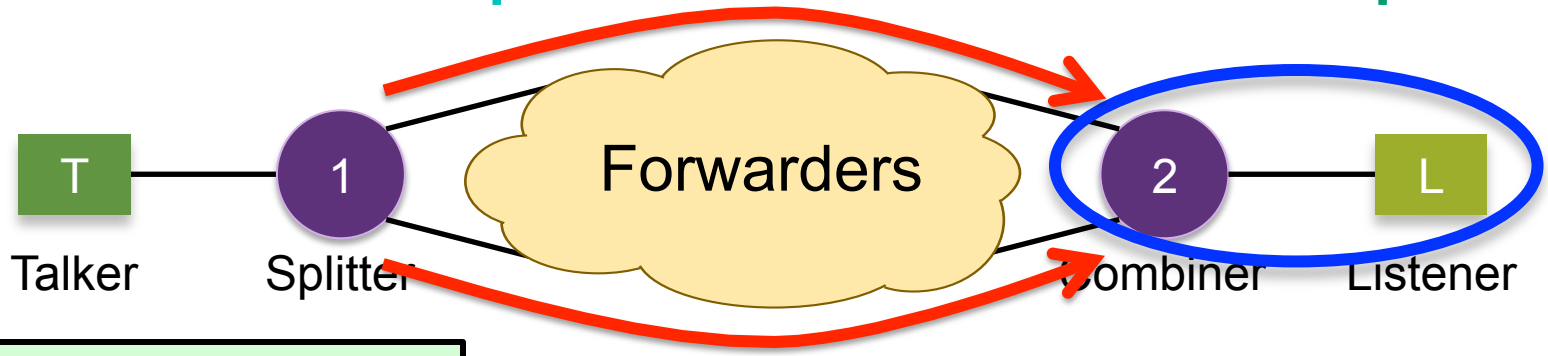
1: T/L are L3 peers. 2: T/L are L2 peers.



pseudowire label
control word
user packet

- Splitter originates a pseudowire carrying exactly the user packet.
- Case 1: User packet is an L3 packet.
- Case 2: User packet is an L2 frame (e.g., Ethernet)

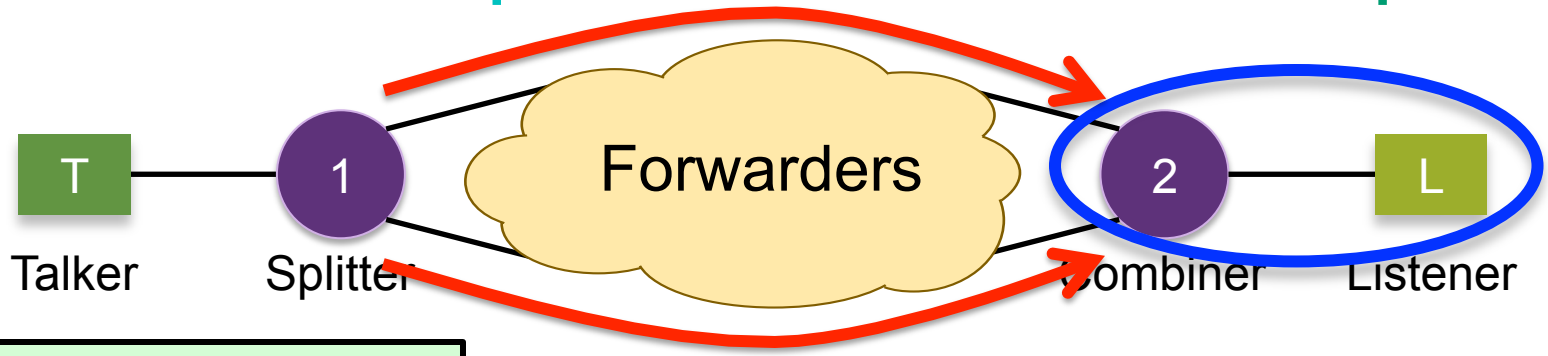
1: T/L are L3 peers. 2: T/L are L2 peers.



pseudowire label
control word
user packet

- Splitter has a peer-to-peer relationship to its Combiner via the pseudowire.
- Pseudowire packet is carried over various media as appropriate, and gets TSN QoS along the way.

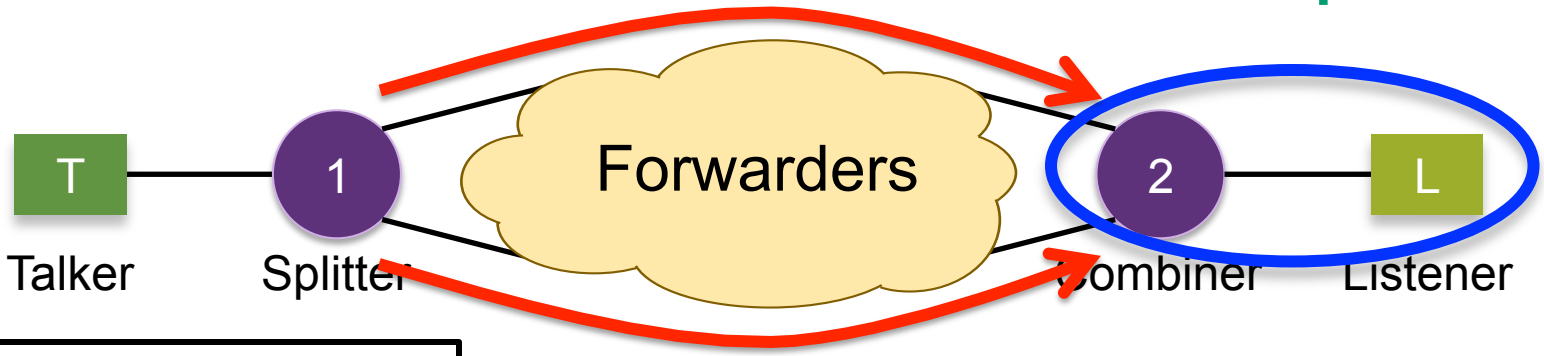
1: T/L are L3 peers. 2: T/L are L2 peers.



pseudowire label
control word
user packet

- Whether the Splitter or Combiner is a Router or a Bridge, and whether acts for itself or proxies for a Router or for a Listener, it always operates at this level of pseudowire.
- The proxy relationships and PW type (case 1 vs. case 2) govern the details of the interfaces to the Talker and the Listener.

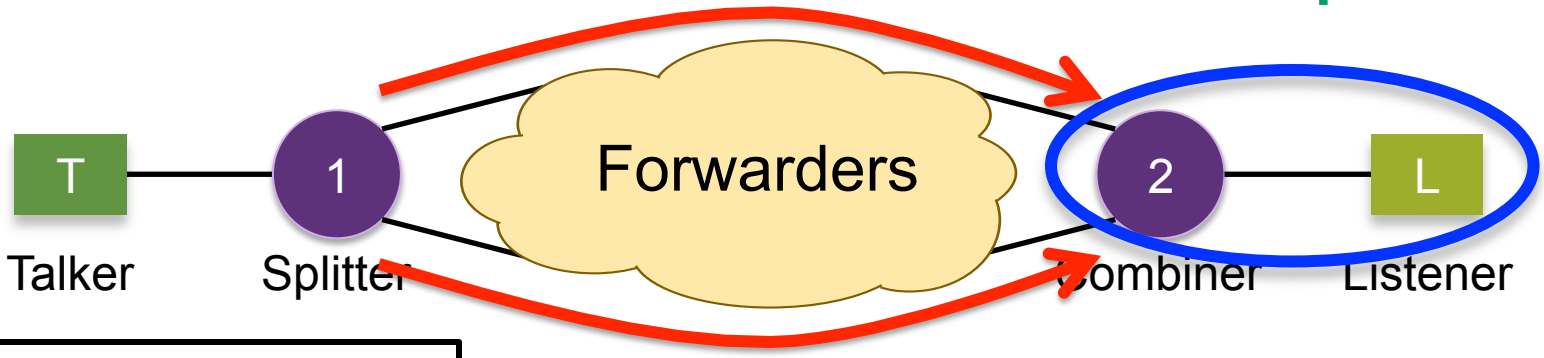
Case 1: Talker & Listener are L3 peers



Group DA, route ID
last hop SA
VLAN as needed
ET: pseudowire
pseudowire label
control word
user packet

- If Talker and Listener are L3 peers, and Forwarders use Ethernet, frame format is at left. Note that L2 addresses don't matter to L3 peers.
- Splitter/Combiner are L3 functions. It doesn't matter whether they reside in L2 or L3 boxes.
- Penalty: **8 bytes**. **4** if sequence numbers not required!

Case 2: Talker & Listener are L2 peers



Group DA, route ID
last hop SA
VLAN as needed
ET: pseudowire
pseudowire label
control word
user DA
user SA
user tag(s)
user data

- If Talker and Listener are Ethernet peers, and Forwarders use Ethernet, frame format is at left. (Another Layer 2 technology would use another pseudowire type and L2 encapsulation.)
- Splitter/Combiner are still **L3 functions**. It doesn't matter whether they reside in L2 or L3 boxes.

Yes, there are more details to work out …

- We have to talk about what MAC addresses are used where, depending on who is proxying for whom.
- The reason that the overhead is so low for Case 1 (L3 peers) is that the MAC addresses, customer VLAN ID (if needed), and payload EtherType are supplied by the Collector, based on the flow ID.
  - But, as shown in Part 1, this is what we're doing, today!

Yes, there are more details to work out …

- Undoubtedly, some new pseudowire types will have to be defined. (We might want to define an IEEE 1722 pseudowire.)
- As shown in Part 1, we still have a number of possible encapsulations on the table. This is not a problem, because we have a solid layering framework. In fact, none of the solutions presented can be ruled out.
- **We cannot and must not prevent any of them, but we only need to encourage, via protocols or profiles, only a very few.**

Detailed “day in the life of a packet”

- The author has more than a hundred slides showing packet formats for various scenarios.
- Rather than reading them, the reader would be better served taking the basic functional stack, the reference network, and the frame formats from this deck and from Part 1, and making up his or her own diagrams.
- With a proper layering plan, everything just works.

5. Summary

5. One-slide summary

- There is one obvious candidate for an end-to-end circuit ID not tied to Ethernet – an MPLS label.
- There is one obvious candidate for end-to-end seamless redundancy that work at either L2 or L3 – an MPLS pseudowire.
- Those two solutions work well with bridges and the current AVB/TSN data plane.
- With the right layering model – presented here – there are many ways to put together properly layered solutions.

Thank you.