Suggestions for P802.1Qcc Inputs and Outputs

Norman Finn
Cisco Systems

Version 2

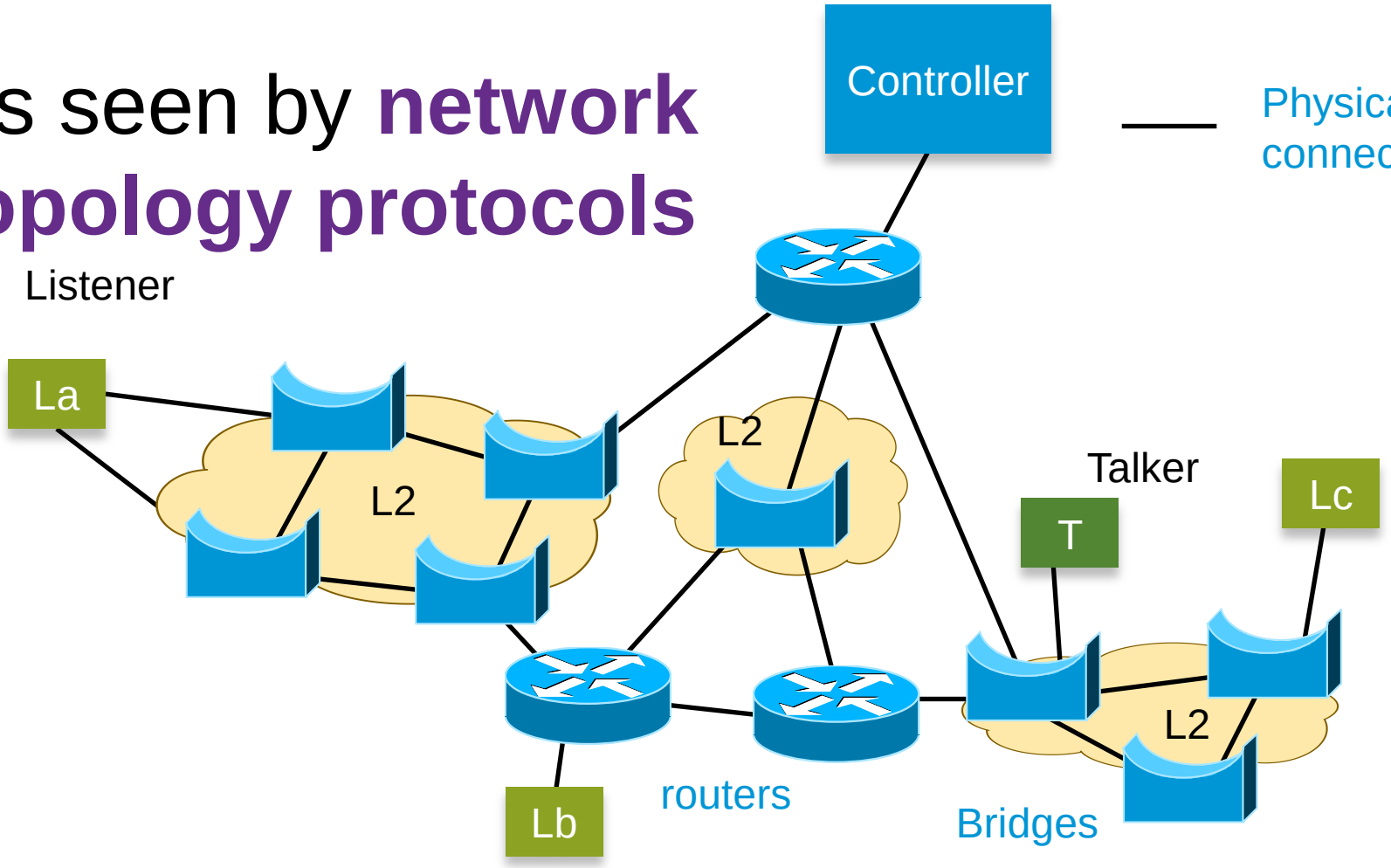
Mar. 18, 2014

This presentation

- This is cc-nfinn-Inputs-Outputs-0314-v02.
- It is based on tsn-nfinn-L2-Data-Plane-0214-v04, tsn-nfinn-L3-Data-Plane-0214-v03, and tsn-nfinn-Day-In-the-Life-0214-v02.

Reference network (from L3 Layering)

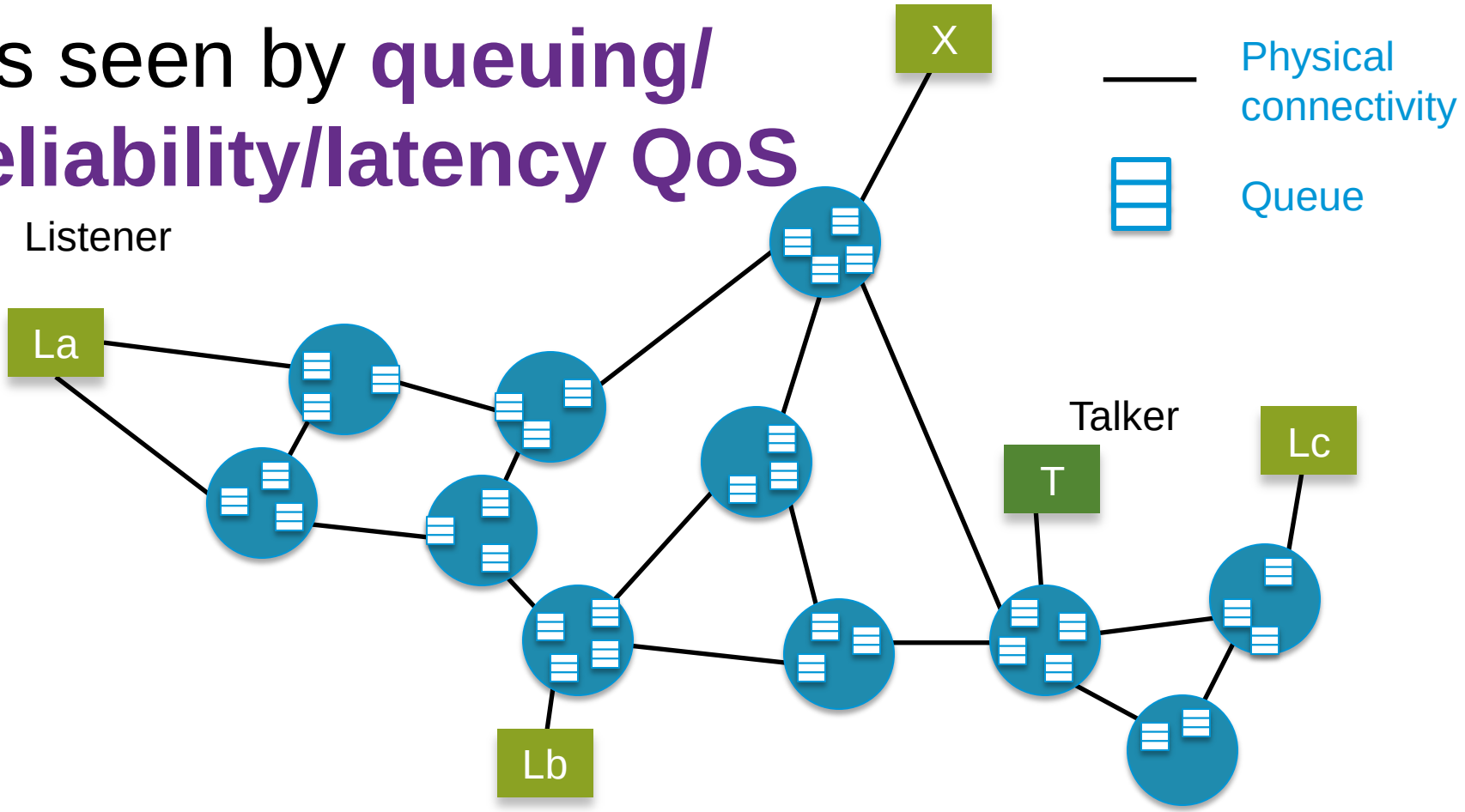
As seen by **network topology protocols**



- Gazillions of complex protocols

Reference network (from L3 Layering)

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



- Just nodes, queues, and wires!!

Why so general?

- There is a big difference between a niche networking application and the established general-purpose networking operational models.
- There already exist any number of “Real-time Ethernet” niche networking application solutions that provide most all of the IEEE 802.1 TSN capabilities, especially seamless redundancy. **You can use one, now.**
- All require that the application writer know about and work around more-or-less severe limitations on the kinds of network operations that are available to the application, compared to “normal” networking, such as:
  - Special network topologies, e.g. rings.
  - Special MAC layers, e.g. novel ASIC-interpreted headers.
  - Layer 2-only connectivity; no QoS support through routers.
  - Layer 3-only connectivity; no QoS support through bridges.

Driving assumption (from L3 Layering)

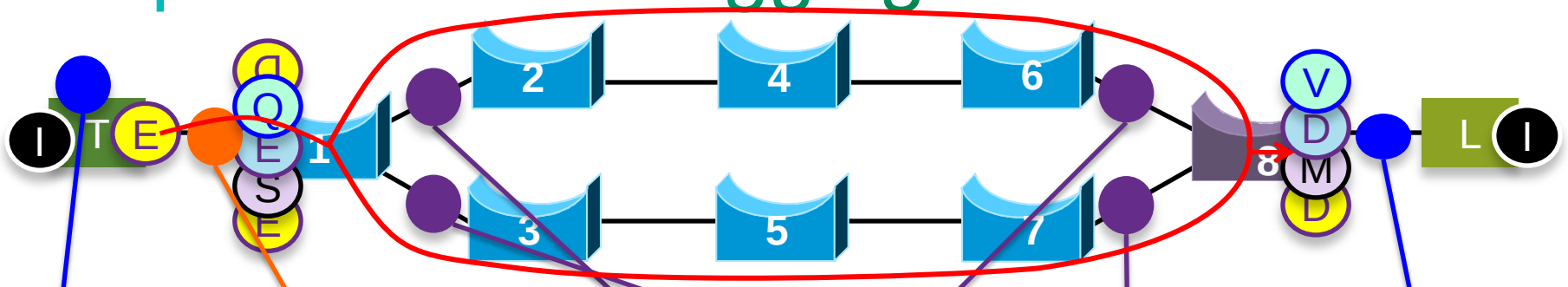
- **The goal of the TSN TG should be to write standards for Quality of Service (QoS) classes, for high reliability and low latency, that offer incremental benefit to any network, whether L2, L3, or mixed, that follow established general-purpose networking models.**
- **To the extent that TSN standards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

Data plane

The data plane

- Following is a list, from tsn-nfinn-Day-In-the-Life-0214-v02, of ten data plane scenarios.
- All of them are perfectly reasonable, and may be implemented by someone.
- There are more that we can easily imagine.
- There is a great deal of commonality among their needs for control plane information.
- **Let us ask, "What does P802.1Qcc need to do to support any one of them? All of them?"**

Summary:
Sequenced TSN tagging:



- This uses the full Split/Merge functionality with different cicuit_identifiers on the paths.

DA: Listener L
SA: Talker T
circuit_ID
ET: whatever
data

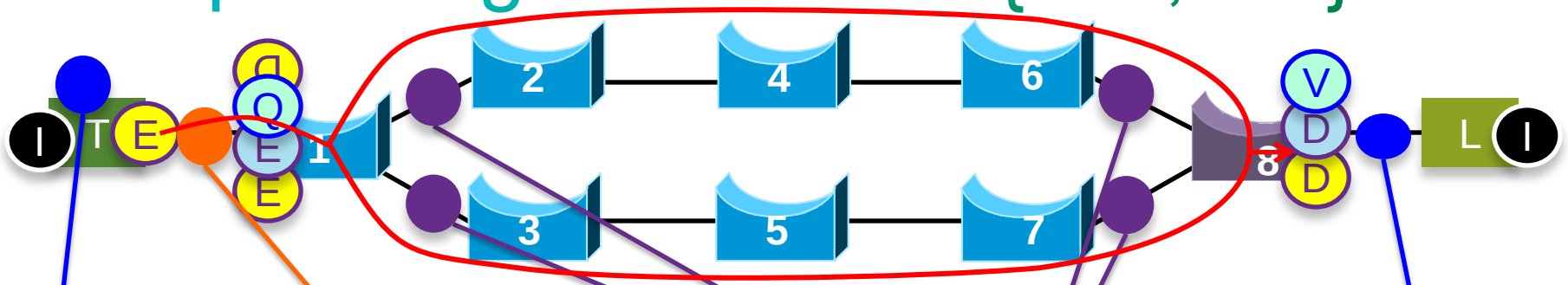
DA: TSN 734
SA: T
VLAN tag 99
ET: whatever
data

DA: TSN 7840
SA: T
VLAN tag 23
ET: TSN Seq
Sequence #
ET: whatever
data

DA: TSN 12
SA: T
VLAN tag 50
ET: TSN Seq
Sequence #
ET: whatever
data

DA: L
SA: T
VLAN 80
ET: whatever
data

Variant 1:
No Split/Merge – all same {VID, DA}



- This uses the full Split/Merge functionality with different cicuit_identifiers on the paths.

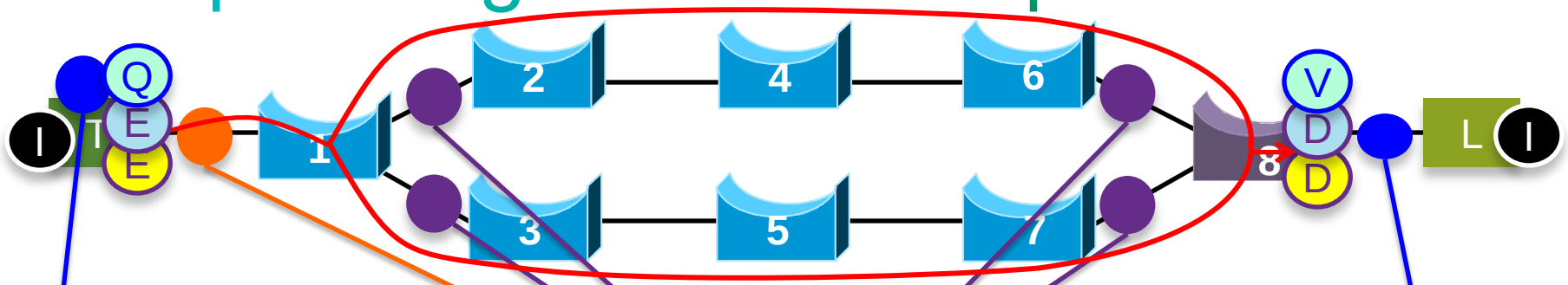
DA: Listener L
SA: Talker T
circuit_ID
ET: whatever
data

DA: TSN 734
SA: T
VLAN tag 99
ET: whatever
data

DA: TSN 734
SA: T
VLAN tag 99
ET: TSN Seq
Sequence #
ET: whatever
data

DA: L
SA: T
VLAN 80
ET: whatever
data

Variant 2:
No Split/Merge – Talker sequences



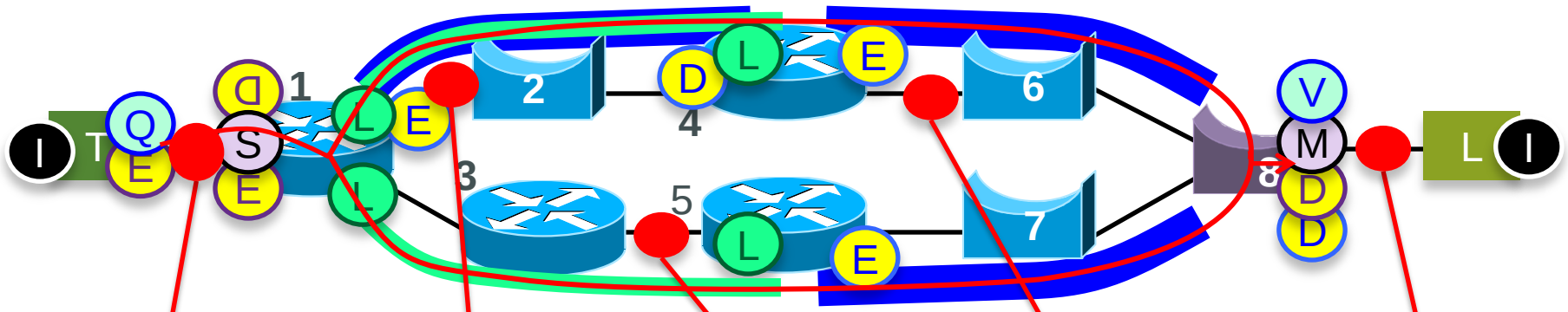
- If Talker T does the sequencing and encaps, and all paths use the same encaps, **it gets really simple!**

DA: Listener L
SA: Talker T
circuit_ID
ET: whatever
data

DA: TSN 734
SA: T
VLAN tag 99
ET: TSN Seq
Sequence #
ET: whatever
data

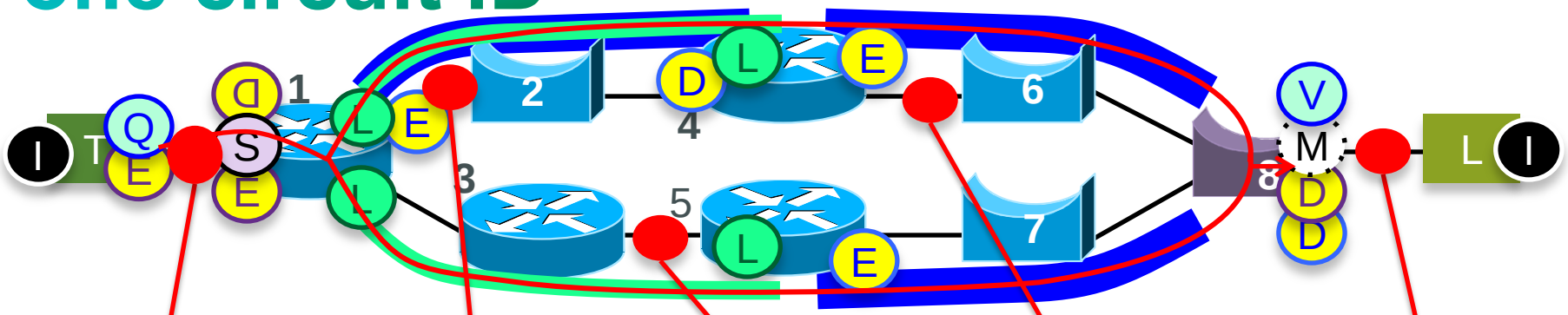
DA: L
SA: T
VLAN 80
ET: whatever
data

SUMMARY: IPgram pseudowire



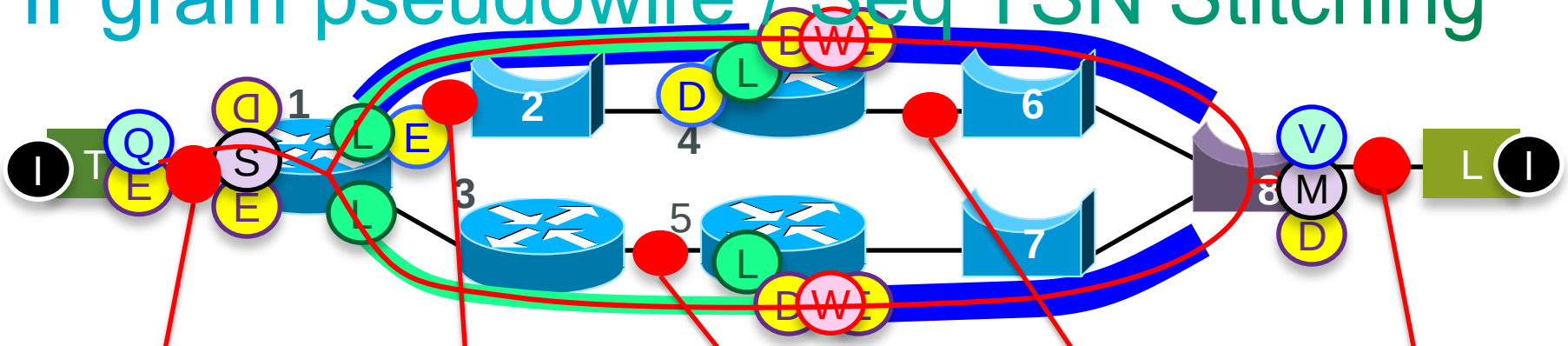
DA: Router 1	DA: TSN 140	DA: Router 5	DA: TSN 994	DA: Listener L
SA: T	SA: Router 1	SA: Router 3	SA: Router 4	SA: Router 4
ET: MPLS	VLAN tag 309	ET: MPLS	VLAN tag 7	VLAN tag 80
Pseudowire 28	ET: MPLS	ET: MPLS	Pseudowire 419	ET: IP
control (seq)	Tunnel 51	Tunnel 346	control (seq)	IPgram
IPgram	Pseudowire 449	Pseudowire 31	IPgram	
	control (seq)	control (seq)		
	IPgram	IPgram		

Variant 3: End-to-end pseudowire, one circuit ID



DA: Router 1	DA: TSN 140	DA: Router 5	DA: TSN 994	DA: Listener L
SA: T	SA: Router 1	SA: Router 3	SA: Router 4	SA: Router 4
ET: MPLS	VLAN tag 309	ET: MPLS	VLAN tag 7	VLAN tag 80
Pseudowire 28	ET: MPLS	ET: MPLS	Pseudowire 28	ET: IP
control (seq)	Tunnel 51	Tunnel 346	control (seq)	IPgram
IPgram	Pseudowire 28	Pseudowire 28	IPgram	
	control (seq)	control (seq)		
	IPgram	IPgram		

SUMMARY:
IPgram pseudowire / Seq TSN Stitching



DA: Router 1
SA: T
ET: MPLS
Pseudowire 28
control (seq)
IPgram

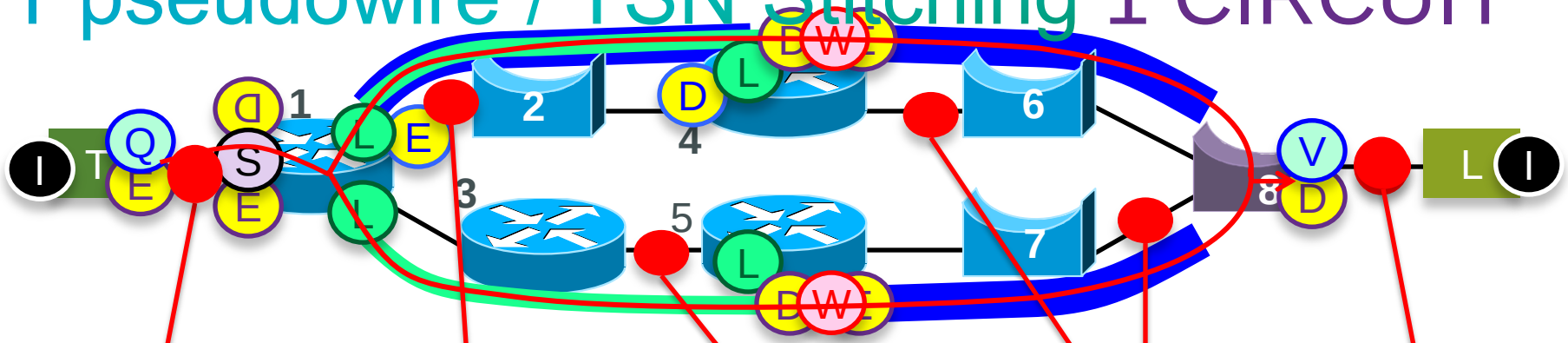
DA: TSN 140
SA: Router 1
VLAN tag 309
ET: MPLS
Tunnel 51
Pseudowire 449
control (seq)
IPgram

DA: Router 5
SA: Router 3
ET: MPLS
Tunnel 346
Pseudowire 31
control (seq)
IPgram

DA: TSN 12
SA: Router 5
VLAN tag 50
ET: TSN Seq
Sequence #
ET: IP
IPgram

DA: Listener L
SA: Router 4
ET: IP
IPgram

Variant 4:
Ppseudowire / TSN Stitching 1 CIRCUIT



DA: Router 1
SA: T
ET: MPLS
Pseudowire 28
control (seq)
IPgram

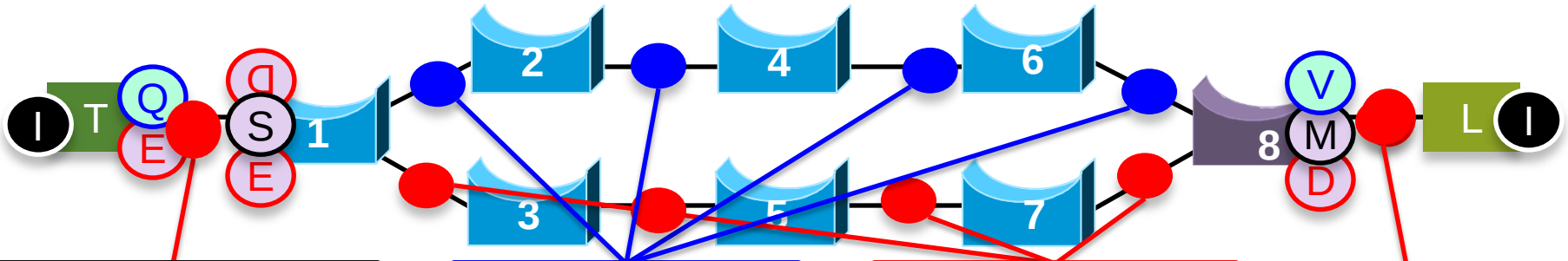
DA: TSN 140
SA: Router 1
VLAN tag 309
ET: MPLS
Tunnel 51
Pseudowire 28
control (seq)
IPgram

DA: Router 5
SA: Router 3
ET: MPLS
Tunnel 346
Pseudowire 28
control (seq)
IPgram

DA: TSN 12
SA: Router 5
VLAN tag 50
ET: TSN Seq
Sequence #
ET: IP
IPgram

DA: Listener L
SA: Router 4
ET: IP
IPgram

Summary: HSR-like tagging



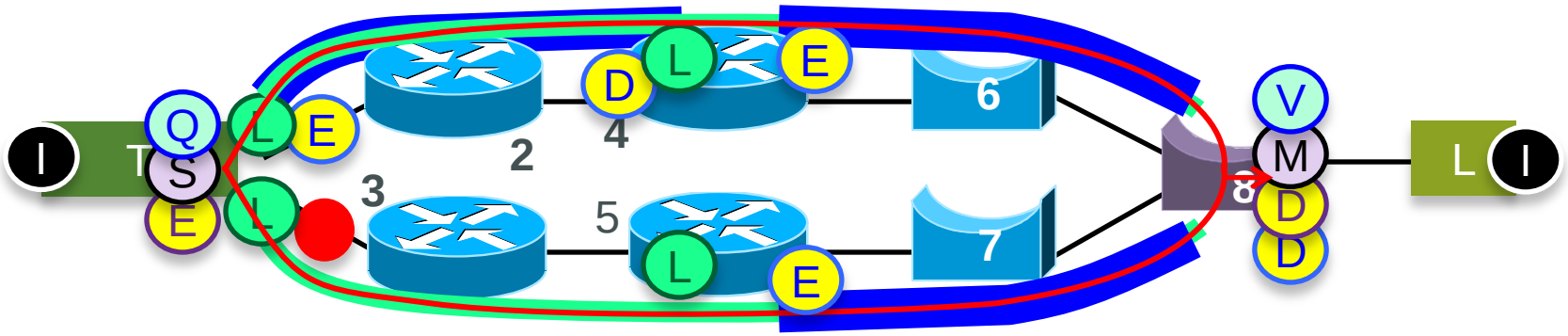
DA: TSN 734
SA: T
VLAN tag 99
HSR EtherType
pid, length, seq
DA: L
EtherType
Data

DA: TSN 7840
SA: T
VLAN tag 23
HSR EtherType
pid, length, seq
DA: L
EtherType
Data

DA: TSN 12
SA: T
VLAN tag 50
HSR EtherType
pid, length, seq
DA: L
EtherType
Data

DA: L
SA: T
VLAN tag 80
ET: IP
IPgram

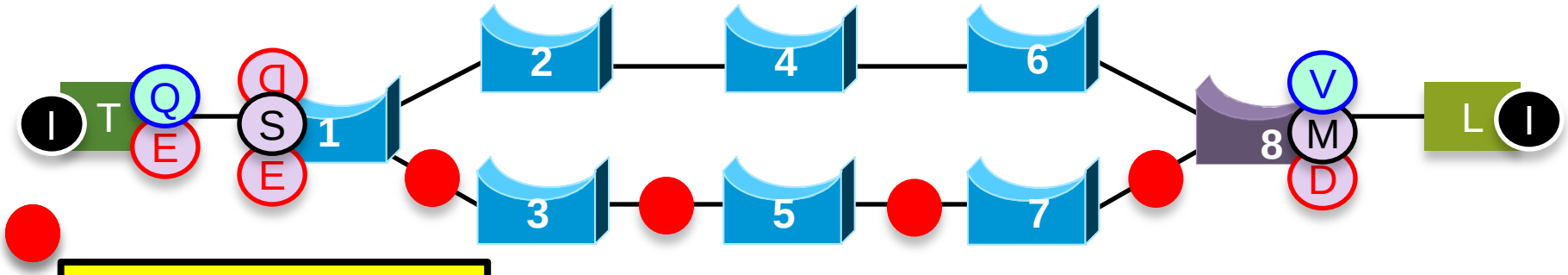
Variant 5: Dual-homed Talker



pseudowire label 28
control (sequence)
IPgram

- Talker T could be dual-homed.
- In this case, clearly T must supply the sequence numbers.
- The sequence numbers are usually part of the encapsulation.
- So, T terminates the pseudowire, not routers 2 and 3.

Layer 2 only: **PRP tagging**



DA: TSN 12
SA: T
VLAN tag 50
EtherType
Data
pid, length, sequence
HSR EtherType

- PRP would work similarly.
- This could be useful to interoperate with existing deployments.
- **A big issue with the PRP trailer is that you can't tell what it's position is in the tag layering.**

Control plane

Assumptions

- This deck makes a number of assumptions about P802.1Qcc that may or may not be shared by others.

Assumptions

- As explained in tsn-nfinn-L2-Data-Plane-0214-v04, this author believes that the object of the 802.1 TSN control plane is to set up **TSN circuits**.
- This deck will look only at the use of P802.1Qcc as a "User Network Interface (**UNI**)" protocol. That is, the interface between a Talker or Listener and the adjacent network node, whether a router or a bridge.
- We will not examine, **in this deck**, how the requests and responses propagate through the network, how or by whom the responses are computed, or what information is needed from network nodes in order to compute the responses.

Assumptions

- As explained in tsn-nfinn-L2-Data-Plane-0214-v04, we assume that what is desired is to obtain a certain TSN Quality of Service for a given flow, without restricting any parameters of that flow other than those directly affecting the QoS, e.g. bandwidth.
  - The flow may be unicast or multicast.
  - The flow may terminate at any sublayer of the OSI model (i.e. have addresses at many levels).
  - We do, however, require a reservation to be made before a data flow can obtain the TSN QoS.

Assumptions

- The applications in the Talker and Listener(s) **do know** at what layer(s) they address each other. A flow might have, for example, all of the following:
  - Source and destination MAC addresses and a VLAN ID.
  - Source and destination IPv6 addresses.
  - UDP source and destination port numbers.
  - An IEEE 1722 stream ID.
- The applications **do not know** and do not care at what layer(s) the network operates.
  - Perhaps it is a single dedicated link.
  - Perhaps it is a Bridged LAN.
  - Perhaps it is a network of routers.
  - Perhaps it is a complex mixture of all of the above.

Assumptions

- **Stream data, not scheduled data.**
- Stream data:
  - Talker transmissions are shaped, but not synchronized with other Talkers; a stream is limited only by a bandwidth contract.
  - The network must assume the worst case for interference among streams in network nodes.
- Scheduled data:
  - Talker transmissions are made at particular times in a rotating, synchronized schedule.
  - Interference between streams in network nodes is calculated and scheduled, so that sufficient buffers can be allocated.

Assumptions

- Some negotiation may be supported, provided that we keep a reasonable scope to negotiation options. For example, it may be true that:
  - A Talker is willing to vary the frame size in order to get better latency;
  - A Talker is willing to transmit at a lower bandwidth in order to accommodate other streams' needs;
  - An application can offer different features, depending on the latency that is available from the network; or
  - The possible latencies are significantly different for different destinations.

Assumptions

- A network administrator will select equipment and configure certain parameters that control what QoS options are available for streams.
  - What queuing methods are available to what network nodes.
  - Policy for conducting QoS negotiation.
  - What function(s) and protocol(s) the network uses to answer the questions posed by the UNI.

Protocol jobs

Current MSRP Jobs

- MSRP does several jobs. (One may ask whether all are necessary.)
- These have different requirements for information elements.
- Certainly, we must keep compatibility with .1Qat, but we should keep these jobs separate, at least in our minds and in the documentation, whether or not they are separable in the .1Qcc protocol, itself.

Current MSRP Jobs

1. **Configuration**: Advertise the AVB parameters configured in Bridges to other Bridges, and to potential Talkers and Listeners.
2. **Availability**: Advertise the availability of streams to potential Listeners.
3. **Native path creation**: Establish the path the stream will follow.
4. **Preliminary reservation**: Make a preliminary determination of the possibility of making the reservation.
5. **Accumulated latency**: The actual worst-case latency.
6. **Final reservation**: Commit and configure the resources in each port along the path of the stream.
7. **Approval**: Report the success or failure of the reservation to the Talker and Listeners.

Current MSRP job flow (abridged)

- Talker obtains a unique Group MAC address (perhaps via 1722).
- Talker selects a Class (A or B) and a VLAN ID from the choices presented by MSRP.
- Talker requests the reservation.
- Listener and Talker are told the actual max latency (“accumulated latency”).

Information flow questions

- What information comes from the Talker or the Listener?
- The stream address information and Tspec could actually come from either end, or from configuration.
- Is stream advertisement really a proper function of MSRP? Should it be handled at higher layers?
- The actual commitment of resources must proceed from Listener(s) to Talker.

Proposed list of .1Qcc Jobs

1. **Configuration**: Same as before.
2. **Availability**: Useful in an L2-only environment, and for backwards compatibility. But, we must allow other methods to be used, instead. (Which we do, today!)
3. **Native path following**: See below.
4. ~~**Preliminary reservation**~~.
5. ~~**Accumulated latency**~~.
6. **Final reservation**: Reserve the resources.
7. **Encapsulation**: The encapsulation to be used for the stream e.g. AVB multicast address and VLAN output from the network to both the Talker and the Listeners.
8. **Approval**: Report the success or failure of the reservation to the Talker and Listeners.

Proposed .1Qcc job flow (abridged)

- Talker and/or Listener request a reservation, specifying (among other things):
  - Native addresses of source and destination(s).
  - Tspec
  - Required min/max latency.
  - Encapsulation capabilities.
- The network returns, with a successful reservation, to both Talker and Listener(s):
  - Encapsulation to use.
  - The required min/max latency.

Native path creation job

- This is the same job being done, now, and it will still be needed for .1Qcc.
- But, “native” path creation should be based on the “native” address stack, not the encapsulation multicast address and VLAN ID, which will become outputs. (The native address stack can still be an AVB multicast and VLAN ID.)

Deleted jobs

4. ~~**Preliminary reservation**~~: This was a by-product of MSRP. It is reliable only when it says, "No." It can be dropped without affecting the Talkers' and Listeners' implementations. Dropping it reduces the complexity of the protocol.

5. ~~**Accumulated latency**~~: This was needed because the Class A/B distinctions are large, because a Talker or Listener could not ask for a specific (smaller) latency, and because no negotiation is possible. I propose fixing these omissions.

Information elements

Information elements

- Now let us look at various information elements, and fit them to the job list.
- There may be information that is:
  - Present today, and needed in P802.1Qcc.
  - Not present today, but needed in P802.1Qcc.
  - ~~Present today, but not needed in P802.1Qcc~~.

Information elements

- All layers of **native addresses** (L2 … L7) for the stream, both source and destination.
  - At present, only the MAC addresses and VLAN are present, and these are an input to MSRP from the Talker.
  - The destination MAC address and VLAN are used by MSRP, at present, for the availability and native path creation jobs.

Information elements

- All layers of **native addresses** (L2 … L7) for the stream, both source and destination.
  - As discussed in tsn-nfinn-L2-Data-Plane-0214-v04, these are the addresses that would be used by the stream, if it were **not** a TSN stream.
  - The native addresses are required for the availability and native path creation jobs.
  - At the lowest layers, this information can be local to parts of the network, and can be different at different points along the complete path (e.g. MAC addresses and VIDs in multiple bridged LANs separated by routers, or VIDs remapped by Bridges).
  - The list may include higher layer addresses (e.g. IEEE 1722 stream IDs) of interest only to the hosts.

Information elements

- The **encapsulation parameters** to be used for the stream, so that the Bridges and Routers can recognize it.
  - As discussed in tsn-nfinn-L2-Data-Plane-0214-v04, returning these to the Talker and Listeners is the **encapsulation** job.
  - These parameters are necessary during the **final reservation** job, because the bridges and routers must know how to recognize the flows.
  - (For a host using the current paradigm, the native addresses and the encapsulation have the same values.)
- Parameters:
  - Encapsulation type(s) (TSN, Pseudowire, etc.)
  - Per-encapsulation parameters (pseudowire label, TSN destination MAC address, VLAN, and priority, etc.)

Class of Service parameters

- At present, the **L2 priority code point** is an input from the Talker to the network, and used by MSRP for **availability** and **native path creation**.
- For P802.1Qcc, I would claim that L2 priority, DiffServe Code Point, and similar items should be outputs from the network to the Talker and Listener as **encapsulation parameters**, rather than inputs from the Talker.

Information elements

- A **stream identifier** for the control plane, unique over some region, used to tie together all of the MSRP jobs.
  - RSVP uses the address list for this purpose. It uses higher-layer addresses, but bottoms out at the IP layer.
  - MSRP uses a unique number built from the source MAC address.
  - Going forward, there may be no Ethernet from which to build an MSRP Stream ID, or there may be Ethernet only, and no IP addresses to build an RSVP Stream ID.
  - For our purposes, we could do both, using the native address stack, as the Stream ID, extending the stack to lower layers (plus an integer, as today). This makes the current Stream ID a valid address stack ID.

Information elements

- The **Network Spec** (required QoS), including:
  - Maximum latency.
  - Minimum latency. (Maximum – minimum = jitter)
  - Traffic Class (optional for the host, and only for backwards compatibility)
- The **Nspec** is what the Talker and/or Listener want, not what the Network promises to deliver.
- These parameters are required for making the final reservation.

Information elements

- At present, the only **Nspec** parameter is the Class (A, B, …) is used. I propose that we retain the Class A/B information, but:
  - We make this invisible to the hosts, except as required for backwards compatibility.
  - We use it in only when required by the shaper algorithms in use (e.g. the current algorithm).

Information elements

- Current **transmission specification** (**Tspec**).
  - Max number of packets sent per measurement interval
    - Will be used to determine overhead when adding encapsulations.
  - Max packet size (including preamble and gap).
    - Max size affects other streams' latencies.
  - From these, one can compute a maximum bytes per second.
    - Required to configure the shapers.

Information elements

- **Additional Tspec** parameters
  - **Measurement interval**
    - This parameter needs to be changeable, as we have discussed. I propose adding it to the Tspec.
  - A **maximum bytes per measurement interval** could be useful.
    - For example, a stream sends 10,000 bytes per interval, with 1500 bytes per frame. The bridges have to allocate 10,500 bytes/interval, because this data rate is expressed as 7 frames of 1500 bytes per interval, not as 10,000 bytes.
    - Is this worth the added complexity? (Honest question.)

Information elements

- **Host capabilities** for supporting various hardware and software protocols and options, namely:
  - Encapsulation methods supported.
- This information enables the network to match capabilities and create connections as the protocols advance in the future.

Information elements

- SRP now has a ~~**VLAN ID**~~ tied to a specific traffic class.
  - This is wrong, as has been discussed. No separate VLAN is required if the stream follows normal forwarding rules.
  - We should use 802.1Qcc as a vehicle to change 802.1Q so that, if a port is AVB-aware, it transmits all VLANs, including VLAN 1, with a tag.
- A new **VLAN ID** parameter is required for .1Qcc.
  - The VLAN ID is part of the L2 address used to set up the circuit.
  - The VLAN ID is returned on a per-stream basis as part of the TSN encapsulation.

Negotiation

Negotiation

- At present, SRP negotiation is:
  - Network operator (or default out-of-the-box values) determines what Classes (of latency) are available to the Talkers.
  - MSRP offers these classes to the Talker.
  - The Talker selects a Class with MSRP.
  - The network reports, to each Listener, the latency that will be delivered (≤ Class latency).

Negotiation

- The current negotiation implies that the available network guarantees do not depend on the locations of the Talkers and Listeners.
  - That works over the very limited scope that we chose to support.
  - That does not work over larger or more heterogeneous networks, some of which are of interest to us. (In a **real** stadium deployment, it may not be easy to guarantee < 7 hops as the network ages.)

Negotiation

- At present, if the guaranteed latency, based on the current topology, changes for the worse (because of a topology or configuration change), then the reservation is likely to be dropped by a Listener that believed the accumulated latency given it when the reservation was made.
- In other words, the Talker says, “I need 50 ms”, the network says, “I can give you 4 ms just at the moment.” If, later, this drops to the perfectly satisfactory value of 6 ms, the reservation can be lost, and there is a pointless flurry of control plane activity.
- The root cause for this wastefulness is that the user cannot request a maximum latency on any grounds smaller than the Class A/B limits, and the Listener isn’t told what that maximum is.

Negotiation

- There are useful paradigms for negotiation used by other protocols.
  - X.25 does a “speak once, no ACK needed” negotiation:
    - There are a range of possible values for a negotiated parameter, with a default value somewhere in the middle.
    - Each side states its preference. If I do not ask for the default, I must support every value between my preference and the default, including the default.
    - We use the highest (lowest) value if we are both on the low (high) side of the default, or the default, if on opposite sides.
  - 802.3 does a “Mother, may I?” power negotiation:
    - Consumer constantly supplies a current preferred value, and a list of other values that it can accept, if necessary.
    - The network returns the value to be used.
    - Both sides can request changes. Network disconnects the consumer if a timely agreement is not reached.

Negotiation

- This author would suggest negotiation along the lines of:
  - Talker and/or Listener supply:
    - Bandwidth offered (Tspec) and latences desired (Nspec).
    - 0 or more alternatives for reduced bandwidth and/or worse latency.
    - 1 or more supported encapsulations (TSN, HSR-like, etc.)
  - Network returns:
    - The bandwidth (Tspec) and encapsulation to use.
    - The guarantees provided (Nspec).

Negotiation

- Either side can request a change.
  - If timely agreement cannot be reached, network drops the reservation.
- Network and hosts base their negotiation actions upon policies that we may or may not standardize.
  - I would suggest that, if we do standardize policy, we do it as a separate PAR from P802.1Qcc, in order to avoid unnecessary delay.

Convergence

Convergence issue

- At present, if a topology change brings active reservations into conflict, necessitating dropping one or more, the oldest reservation is retained.
- But, if reservations are made at more-or-less the same time, different bridges may have different opinions about which reservation was made, first.
- This can lead to instability, and potentially, to deadlock situations.
- Including a “time of request” parameter in the reservation would help this problem; we have to think about it, more.

Requesting multiple paths

Requesting multiple paths

- In our current PARs, we have a protocol for distributing paths (P802.1Qca), but not one for requesting them.
- We could do this in P802.1Qcc, by adding to the Nspec a parameter for a worst-case acceptable value of "**Packet Loss Ratio**".
  - This would permit the network to select a path that, for example, avoids wireless links.
  - This would permit the network to create multiple paths and turn on seamless redundancy, if no one path is reliable enough.

Requesting multiple paths

- Why would we **not** forge ahead with a packet loss ratio parameter?
- Because this opens a can of worms. There are many factors influencing path choice besides a desire for higher reliability:
  - I want to pass through or avoid certain nodes that do or do not have certain features or security risks.
  - I want a path that will have the least impact on the best-effort traffic.
  - I am willing (or not) to force another flow to move, in order to let me through.

Requesting multiple paths

- Making requests for a path or multiple paths that meet certain criteria is an area that has been thoroughly explored and reasonably well standardized by the IETF PCE WG.
- I believe that further contributions are necessary in order to decide whether or how to use P802.1Qcc to request multiple paths.

Summary

Suggestions for P802.1Qcc (1)*

- Delete AVB Port Priority from 802.1Q.
- Change 802.1Q to say that an AVB port always outputs tags on every VLAN, except as explicitly configured.
- We will save the allocation of TSN VLANs for pinned-down paths, and the allocation of TSN Group MAC addresses, for a later submission.

* Marked items seem to this author to be of less importance.

Suggestions for P802.1Qcc (2)

- New input (from Talker/Listener) parameters added to P802.1Qcc reservations:
  - The entire native address stack, L1 … L7
  - Measurement interval (in Tspec)
  - Bytes per measurement interval (in Tspec)*
  - Maximum latency (in Nspec)
  - Minimum latency (in Nspec)
  - Optional alternative values for max packets/interval, max frame size, (bytes/interval), max latency, min latency*
  - Encapsulation capabilities (TSN, PRP, etc.)
  - Time that original request was first received (from first node to handle the request, not from Talker or Listener).
- NOTE: Because of the way MSRP works, these are also all output parameters – to the Listener if the Talker initiates the reservation, or to the Talker, if the Listener initiates it.

* Marked items seem to this author to be of less importance.

Suggestions for P802.1Qcc (3)

- New output (to Talker/Listener) parameters added to P802.1Qcc reservations:
  - The minimum and maximum acceptable latency (from the input list).
  - The maximum frame size, frames/interval (and bytes/interval*) to transmit (from the input list).
  - The encapsulation to use, and the parameters for that encapsulation (e.g. TSN Group destination address, VLAN ID, and L2 priority).

* Marked items seem to this author to be of less importance.

Suggestions for P802.1Qcc (4)

- Current input parameters to be deleted from P802.1Qcc except for v1 users:
  - Traffic Class and/or L2 priority.
  - (Both may still be necessary among the Bridges.)

* Marked items seem to this author to be of less importance.

Suggestions for P802.1Qcc (5)

The new work flow is then:

- Talker and/or Listener request a reservation, specifying (among other things):
  - Native addresses of source and destination(s).
  - Tspec (and options*)
  - Nspec: required min/max latency (and options*).
  - Encapsulation options.
- The network returns, with a successful reservation, to both Talker and Listener(s):
  - Option selections for Tspec and Nspec*.
  - Encapsulation selection and parameters.

* Marked items seem to this author to be of less importance.

Suggestions for P802.1Qcc (6)

- The reservation can be renegotiated, after creation, to other options among the list originally given.*
- The reservation is dropped if the worst-case requirements are no longer met (or if renegotiation is unsuccessful*).
- The only encapsulation supported in this version would be TSN Group address + VLAN ID + L2 priority, but the encapsulation is specified with an OUI, to allow expansion.

* Marked items seem to this author to be of less importance.

Thank you.