TSN Control & Configuration

Norman Finn
Cisco Systems

May 27, 2015

Introduction

- This contribution is in response to the excellent analyses presented in cc-goetz-MRPv2-MSP-v12 and tsn-sexton-feature-priority-request.
- It is an attempt to map the shortest route from the current state of the IEEE 802.1 TSN Task Group to a viable set of TSN standards.

There are four scenarios of interest

1. **Static:** The Talkers, Listeners, and relay systems are configured before power-up. There are no run-time changes to reservations.
2. **Central:** On top of (1), there is a central network controller that learns what was preconfigured, and is responsible for coordinating any changes to those configured reservations with any new reservations. Reservations can be made by Talkers, Listeners, and third-parties (e.g. applications controllers).
3. **Peer-to-peer:** On top of (1), there is no central controller. Talkers and Listeners are responsible for making additional reservations using a peer-to-peer protocol (MSRP or MSRP++).
4. **Mixed:** Some relay systems know about the central controller, and some only know the peer-to-peer protocol.

Scenario 1: Configuration

Minimum requirements for configuration

- Let us ask, “What is the minimum amount of standards and development work that must be done to enable configuration (scenario 1)?” We need:

1. A means for the system designer to specify his/her/its decisions.
2. A distinction between volatile and non-volatile flows, the later of which persist across a reset/reboot.

- For these, we require a full set of read/write managed objects for paths, seamless redundancy, shapers, schedules, PTP, etc. In short, for everything.

Configuration **solution**

- Note that the requirement to configure reservations can be very simply met by a set of managed objects that permit the configuration of an MSRP++ reservation. That is, MSRP++ without the PDUs or timeouts.
- This capability is necessary, not optional. Configurability of these data flows is a “shall,” both for implementers and for standards writers.

Scenario 2:
Central Controller (only)

Minimum requirements for central controller

- Let us ask, "What is the minimum amount of standards and development work that must be done to enable a central controller (scenario 2)?" We need:

1. A means for the controller to determine the topology of the network, including all of the relay systems, Talkers, and Listeners.
2. A means for the controller to find out what reservations have been configured in the network beforehand.
3. Provision for a backup controller, in case the active controller fails.
4. A means for the controller to distribute its decisions.
5. Some networks will require a means for an applications controller to submit (lists of) reservations.

Topology discovery **issue**

- No current topology protocol gives the locations of the Talkers and Listeners.
- No current topology protocol gives the locations of both routers and bridges in a mixed L2/L3 network. (Not necessarily a consideration for 802.1 TSN.)
- No current topology protocol separates out the physical components of an Aggregated Link.
- The managed objects associated with LLDP do, and those associated with MSRP should, allow a network management station (or central controller) to detect the relay system's physical port(s) to which each Talker or Listener is attached. These same managed objects also reveal the connectivity of bridges and routers.

Topology discovery **solution**

- The network controller uses the LLDP MIBs (and/or YANG model, when available) to discover the network topology.
- Relay systems **must** run LLDP and support the managed objects.
- End stations **can** run LLDP and support the managed objects.
- Note that SNMP (and hopefully, YANG models) support management configuration of Notifications when monitored variables change, e.g. when a topology change occurs.

Central controller managed objects

- Every relay system must, and Talkers and Listeners may, provide the managed objects of scenario 1 for configuration.
- These managed maintain three classes of reservations:
  - Non-volatile configuration (remembered across boot/reset).
  - Volatile configuration (human or central controller).
  - Peer-to-peer protocol (MSRP++).
- These objects may include the system's opinion of who the controller is, if any.
- This supports non-volatile configuration, decision distribution by the controller, and human interference, and provides a means for a backup controller to obtain the complete active database when it acquires control of the network. (Note that this path has been well explored by the IETF PCE WG.)

Central controller applications interface

- Even if Talkers and Listeners do not participate in reservations, we can imagine a need for an applications controller to submit requests for reservations and receive responses from the network controller.
- Because this is so similar to the mixed scenario 4, we will defer this discussion to that scenario.
- Not all networks require this interface; the applications controller can be embedded with the network controller.

Central controller **solution**

- LLDP + managed objects handle scenario 2 with **no additional required protocols**.
- 802.1 TSN does have to complete the set of managed objects. At the very least, we need to distinguish between volatile and non-volatile configured, and dynamic, reservations.
- We also need to think about what managed objects and/or keep-alive mechanisms are required to support primary and backup controllers.
- Remember, scenario 2 does not include the Talker/Listener or applications controller registrations that are part of scenario 4.

Scenario 3:
Peer-to-peer (only)

Minimum requirements for peer-to-peer

- Every Talker and Listener that wants to control reservations must implement a User Network Interface (UNI) Protocol to ask for reservations.
- Even if all new/dynamic reservations are made via a peer-to-peer protocol, volatile and non-volatile configured reservations (scenario 1) must be supported.
- We are assuming the non-existence of a central controller. Therefore some TSN features cannot be supported, except by pre-configuration. In particular, we do not believe that it is practical to support P802.1Qbv scheduled transmissions with a peer-to-peer protocol.
- Therefore, the UNI does not have to support all TSN features. (But, see scenario 4, mixed controller/peer-to-peer.)

Peer-to-peer **solution**

- MSRP is chatty (sends lots of packets) and unreliable. Because it has no transport layer, either the loss of MSRP PDUs due to buffer exhaustion, or low bandwidth caused by transmission throttling, will limit the number of reservations that can be kept per link.
- Using the MSRP scheme for distribution data, but using an ISIS LSP-based method for transferring data from system to system, offers a significant improvement to MSRP. Let's call that MSRP++.

Scenario 4:
Mixed central and peer-to-peer

Mixed controller and peer-to-peer

- For the mixed scenario, we have all of the requirements of scenario 2, the central controller, plus:

1. A means for a relay system to notify the central controller of the receipt of an MSRP++ registration.
2. A means for the controller to tell a relay system to respond to an MSRP++ registration.
3. A means for an applications controller to submit (perhaps a large number of) reservation requests to the network controller on behalf of Talkers and Listeners that do not run MSRP++.

Mixed controller and peer-to-peer **solution**

- If we use an LSP-based MSRP++, then the relay-system-controller protocol, the applications-controller-network-controller, and UNI protocol can be almost the same, except for the outer “carrier” shell of the PDU:

1. We need a nearest-neighbor shell for the end-system-relay-system UNI MSRP++ PDUs.
2. We need a transport protocol (TCP) to carry the relay-system-controller PDUs and applications-controller-network-controller PDUs.

- We need TLVs in the LSPs that support making a registration on behalf of another party.

  Note that a “reply” to a registration is, itself, a registration!

Optional behaviors

Optional behaviors

- There is nothing wrong with running SPB and/or 802.1Qca.
- There is nothing wrong with building a controller and relay systems that use SPB and 802.1Qca to obtain topology and distribute multipath information.
- However, requiring those protocols for TSN compliance is not, in our opinion, a good idea:

  We still need a solution to finding the Talkers' and Listeners' locations in the topology, and that solution (LLDP) gives the same information as SPB to the controller.

  We still need a solution for controlling non-path managed objects (e.g. schedules) in the relay systems, and that solution (YANG/MIBs) can carry the same information as Qca.

  SPB + Qca imposes upon small, cheap three-port bridge+station devices the burden of maintaining the LSPs containing entire network database, including TSN additions. That is too much to require.

Summary

The role of MSRP++

- The big difference between this contribution and cc-goetz-MRPv2-MSP-v12 is that this contribution does not use MSRP++ to distribute the decisions of the central controller.
- That is, the path for a mixed reservation to a known Listener in this contribution is:

  Talker → edge relay system → controller →→ all affected relay systems, in parallel →→ responses back to the controller, in parallel → response to edge relay system → Talker.

- While the path for cc-goetz-MRPv2-MSP-v12 seems to be:

  Talker → edge relay system → controller → path information to edge relay system → hop-by-hop along the path through the affected relay systems → hop-by-hop back to the Talker.

The role of MSRP++

- The reasons for preferring this contribution's use of MSRP++ to that in cc-goetz-MRPv2-MSP-v12 include:
  - The information flow in this contribution, just described, is inherently faster, because all relay systems work and respond in parallel.
  - MSRP++ is a thoroughly impractical vehicle for distributing the port schedules, because each port's schedule is particular to that particular port, contains data from many different flows, and each flow can originate from a different Talker's registration. Therefore, the alternative presented here is necessary, anyway.
  - There are use cases for scenario 2, where the Talkers and Listeners do not issue registrations, that have no need for MSRP++ at all.
  - (My apologies if my interpretation of cc-goetz-MRPv2-MSP-v12 is incorrect.)

The role of MSRP++

- As the IETF PCE WG has discovered, the RSVP-TE scenario has interesting security considerations – namely, the routers along the path need to validate the path given them via RSVP-TE. One solution requires “cookies” to be distributed from the PCE by RSVP-TE, for validation between the relay systems and the PCE (central controller).
- Using a secure relay-system-controller connection bypasses this issue; all reservations come from the controller on a secure channel.

MSRP++ is certainly necessary!

- MSRP++ is required, in its peer-to-peer form, for scenario 3, no controller. This requires a next-hop-only "carrier".
- MSRP++ is required, with a next-hop-only carrier, for a UNI in a mixed Talker-driven controller-driven scenario.
- MSRP++ is required, with an L3 transport carrier (TCP) for the relay-system-controller communication required to allow the controller to make the decisions required to interface to a peer-to-peer MSRP++ or UNI MSRP++ neighbor to a controller-attached relay system.
- **But, MSRP++ should not be used to relay information from a controller to the relay systems that it controls.**

Summary

- MSRP++ is not a “shall,” but a “may,” for TSN systems.
- Our highest priority should be ensuring that we have a complete set of managed objects, including LLDP, containing the variables needed for volatile and non-volatile configuration, as well as controller-relay-system communication, for the controller-only scenario.
- LLDP and the TSN managed objects are each a “shall” for relay systems and a “may” for end systems.
- SPB and Qca are each a “may,” not a “shall.”
- The IEEE 802.1 TSN TG should spend its limited resources on an LSP-based MSRP++ to the exclusion of another MRP-based version of MSRP that will have a very short useful lifetime.

Thank you.