

P802.1CB

Seamless Redundancy:

Decisions to be Made

Norman Finn
P802.1CB Editor

Version 2 September 9, 2014

High priority questions



High priority

- How do we handle validation of TSN Streams and latent error detection?
 - See [slides, below](#), for one suggestion. (Now. Otherwise, the following points don't make much sense.)
 - Another is in Draft 0.4.
 - The editor much prefers the solution in the present deck to modifying the Bridge Relay Function's forwarded parameters.
 - The Stream Gates are needed for many applications, anyway.
 - Giving preferential treatment to particular flows really does require per-flow counters for debugging purposes.
 - There is nothing sacred about the current interface, and nothing terribly wrong with adding a new `input_port` parameter, but given that we need the per-flow counters, why define two mechanisms for the same purpose?
 - Also, it gets rid of the “the sequencing sublayers are not truly peer entities” phrase that this editor finds exceptionally sub-optimal.
- **DECISION: We will describe the input gates from the [slides, below](#) into Draft 0.5 Annex C, subject to further discussion, this week.**

High priority

- Are the stream discard algorithms in Draft 0.4 adequate?
 - It appears to me that, if the sender gets reset, the receiver will discard all packets until the sender gets back to where he was, before.
 - That is, for a 28-bit checksum, 2^{27} packets get discarded when a source reboots.
 - This doesn't seem ideal.

High priority

- There are at least four ways to handle the reset problem:
 1. Insist on the serializer performing a handshake with every discarder before restarting the sequence_number to 0.
 - This raises all sorts of issues with who boots first, and do you hold up the data stream if you are the splitter and you come up (reboot) after the source starts transmitting.
 2. Start with sequence_number 0, and skip over 0 when incrementing. Discarders always start with no history and accept everything. Receiving a 0 restarts the expected sequence.
 - If the packet with a 0 sequence_number is lost, it's not clear what you do.
 3. Start with a random sequence_number, and treat the reception of a packet outside the window as a reset. That is, always accept, instead of discard.
 - This makes the chance of problem on reset = (window size) / (sequence space), and the consequence of an error is the loss of one window of data, instead of 1/2 of a cycle of data.
 4. 2 + 3. Always start at 0, skip over when incrementing. Receipt of 0 or receipt of sequence outside window = reset. MY RECOMMENDATION.
- Note that a packet with a rogue sequence_number will be discarded by the Draft 0.4 algorithm, and accepted by algorithm 4, above. I do not think that is a problem. Do you?
- **DECISION: We will describe method 4 in Clause 6 of Draft 0.5. People are encouraged to look for problems with this approach.**

Medium priority questions



Medium priority

- Any advance on the terms, “intermittent streams” and “bulk streams”?
- Is support for discarding duplicates from “bulk streams” mandatory, optional, or not included in the document?
 - This requires keeping some history, perhaps in the form of a last-received sequence number and a bit vector of recent sequence numbers received.
- **DECISION: We will keep the distinction in Draft 0.5 Clause 6. We can note that one is the same as the other, with a 0-length bit vector.**
- Anyone mind if I attempt to integrate the sense of Rodney’s comments with the rest of the document, being free with the text, subject to the decisions we make, this week?
- **DECISION: No one minds.**

TSN Stream validation and latent error detection



Validation and error detection

- We define a TSN Stream Gate (TSNSG) function.
- A TSNSG can be placed on an input port.
- A TSNSG is above the TSN Circuit sublayer, so has the `circuit_identifier` available to it.
- The TSNSG has some number of Gates, each with an associated `circuit_identifier`.
- The TSNSG has one additional Gate, associated with all `circuit_identifiers` not having their own Gates in this TSNSG.
- Each Gate has a control bit, much like the P802.1Qbv gate control bit, that specifies whether the gate is open or closed.
- The TSNSG has an OPTIONAL rotating schedule, just like the P802.1Qvb schedule, that turns the Stream Gates on and off.
- The Gate controls whether a packet with that `circuit_identifier` is admitted or discarded.
- In a Bridge, this admission/discard is a part of the existing policing function.
- The TSNSG counts packets admitted/discarded per Gate.

Latent error detection

- The Seamless Redundancy Sequence Processing function counts packets passed and packets discarded.
- The Processing function has parameters that can trigger an error if the ration of packets forwarded to packets discarded is wrong.
 - I would suggest the trigger is N forwarded with 0 discarded, for a configurable N .
- If an error is reported by this mechanism, the Stream Gate counters can be examined in each system to find the problem.

Thank you.

