802.1CB maintenance/improvements

FRER
Fixing the Elimination function (Sequence Recovery Reset)

IEEE 802.1 Maintenance TG
March, 2020

Balázs Varga, János Farkas Ericsson Research 2020-03-16

Topics

- Proposal
  - Correcting of “Sequence Recovery Reset” process in order to avoid duplicate delivery
- Background
  - Details of problem and solutions

802.1CB maintenance
Overview

- **Problem: reset of Sequence recovery function defined in 802.1CB-2017 may cause temporary duplicate delivery**

- **Target: ensure that no duplicate delivery happens due to reset of sequence generation function**
- How: change reset, two ways are proposed
    1. Adding a reset-guard period after the reset, where received packets are dropped.
    2. Modifying the reset procedure to be root-cause dependent.

    - Which one is preferred?
        - implementation of the reset-guard solution is simpler, but the reset-guard solution somewhat increases the outage because packets are dropped after the reset event for a while.
        - implementation of the solution where the root-cause of reset is distinguished is more complex, but it can minimize the impact of the reset event.

Problem with existing reset mechanism

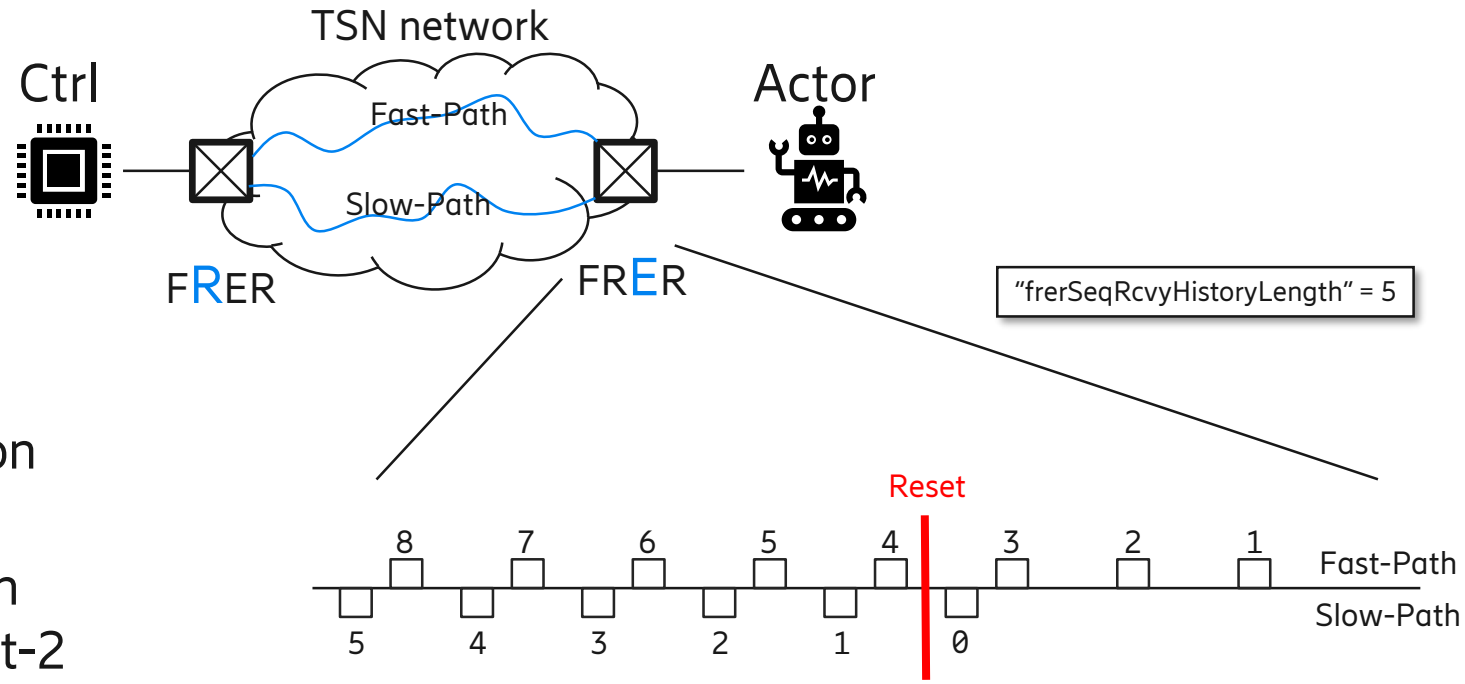
Analysis of 802.1CB
Reset of Sequence Recovery Function

- 802.1CB-2017 defines three reasons to reset the Sequence recovery function:
  1. BEGIN event (initialization/reset),
  2. Management event (frerSeqRcvyReset=true) and
  3. RECOVERY_TIMEOUT event (timeout mechanism expired).

- Current procedure:
  - "7.4.3.3 SequenceRecoveryReset" sets the "RecovSeqNum" to "RecovSeqSpace - 1", clears the "SequenceHistory" array and sets "TakeAny" to true.

- In case-1 (BEGIN) and case-2 (Management), the slow path may be transferring packets whose corresponding duplicates were already received and processed (forwarded) by the Elimination function of a node before the reset was generated. Such scenarios result in duplicate delivery. (See next slide ...)

Analysis of 802.1CB
Result: duplicate delivery



- After the reset,
  - Packet with sequence_number=4 arrives over the fast path. It is accepted by the Elimination function due to the true value of "TakeAny".
  - Packets received over the slow path after the reset (i.e., packet-1, packet-2 and packet-3), are also accepted as they are in the history window and the reset has cleared the "SequenceHistory". But these packets were already delivered before the reset so duplicate delivery happens.
- Duplicate delivery (even temporarily) is not acceptable for TSN networks as it breaks one of the basic design rules, namely a TSN Stream is not allowed to consume more than the resources reserved for it. Consuming more that the designated resources via duplicate delivery may cause violation of QoS requirements for some of the Streams, e.g., delay or loss violation.

1, Adding a reset-guard period

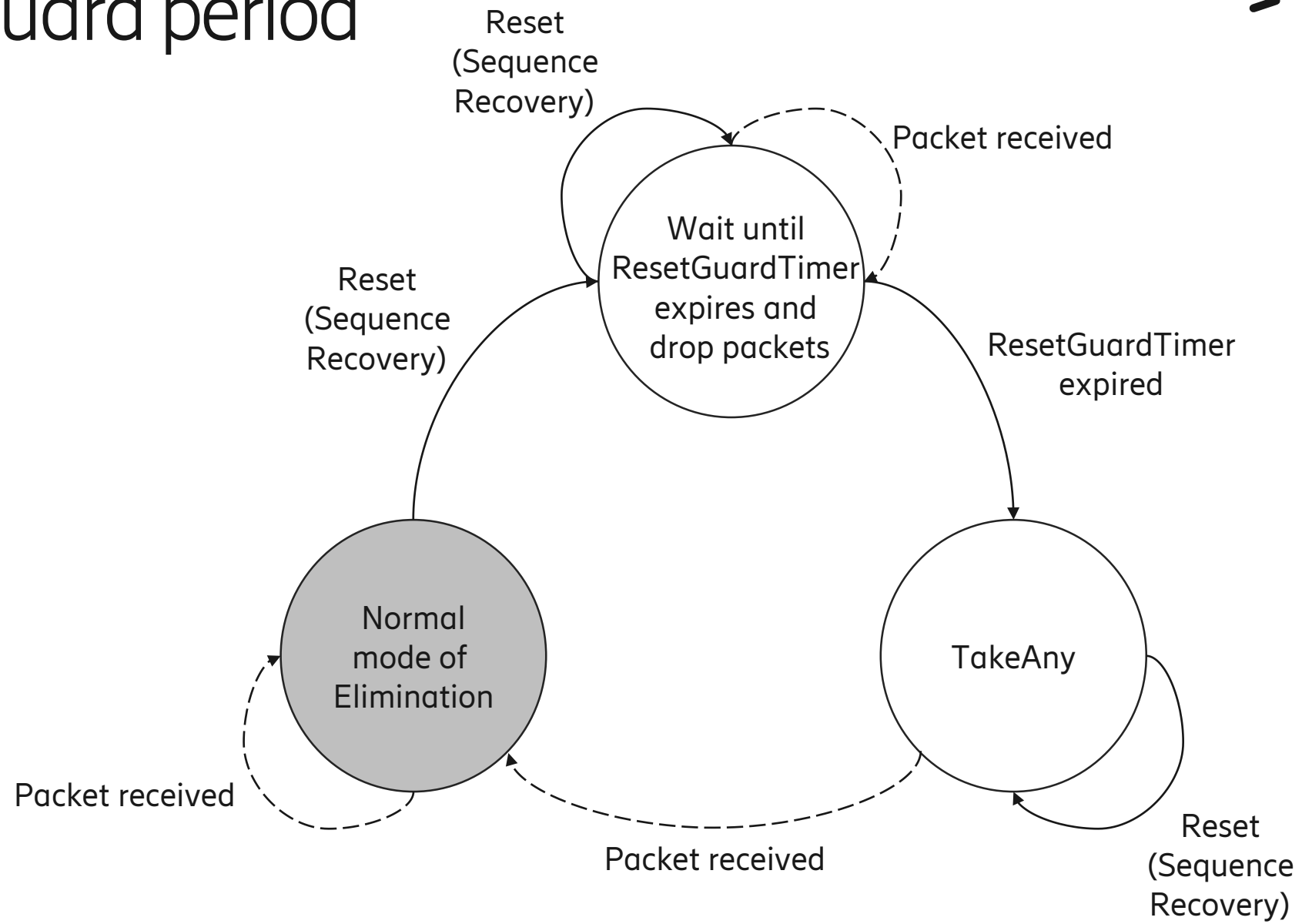
Adding a reset-guard period
Basics of operation

— Solution: extend the reset mechanism with an added reset-guard period, where received packets are dropped. The duration of the reset-guard period depends on the delay difference of the different paths used by the Member Streams.

— *MaxPathDelayDiff* = maximum $_{\text{for all "i" and "j"}}$ ($\text{PathDelay}_i - \text{PathDelay}_j$)
where "n" denotes the number of Member Streams and i, j = {1 … n}

— The reset-guard period timer ("ResetGuardTimer") is used as follows on the Elimination function:
  — When reset, set "TakeAny" true, set the "ResetGuardTimer" to the "MaxPathDelayDiff" of the Stream and clear the history ("SequenceHistory")
  — Drop all received packets of Member Streams until "ResetGuardTimer" expires.
  — Accept the first packet received and update "RecovSeqNum" and "SequenceHistory" accordingly
  — Back to normal operation of Elimination

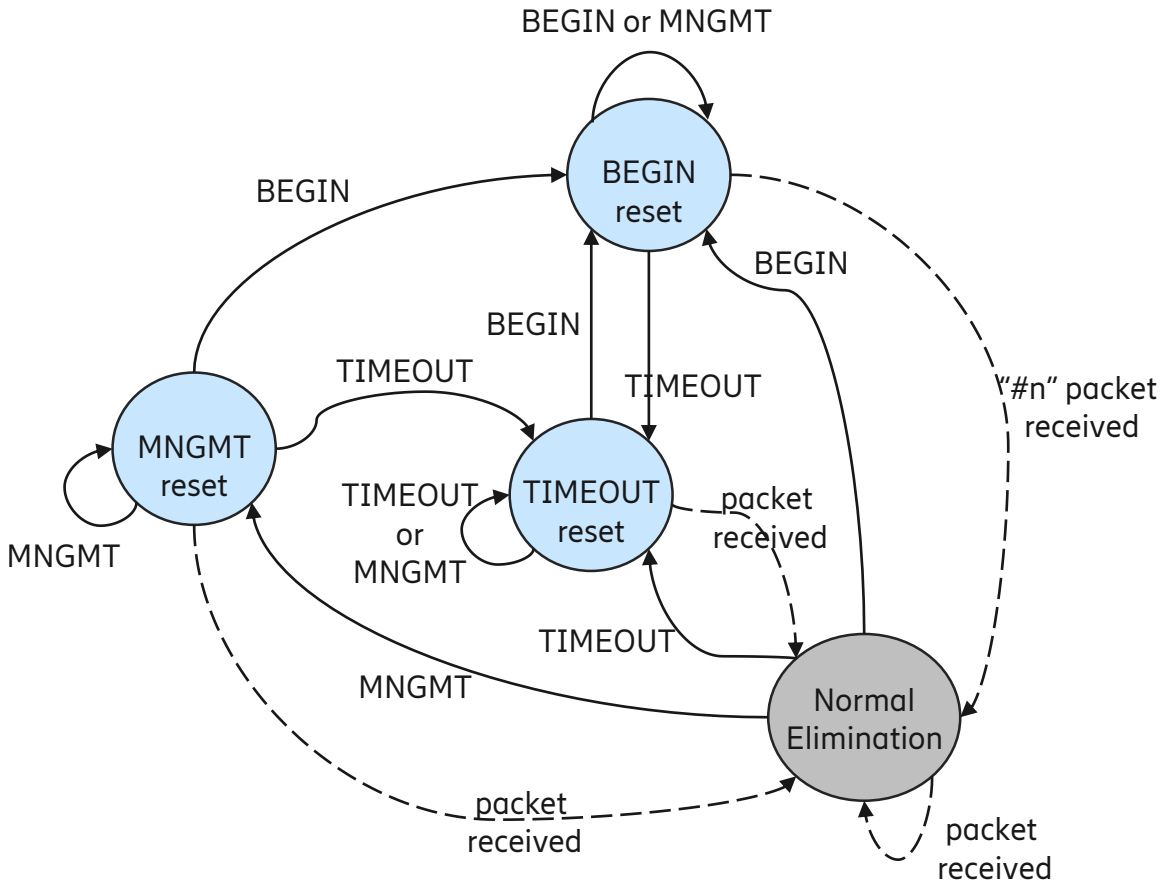
Adding a reset-guard period
Basics of operation …



2, Root-cause dependent reset procedure

Root-cause dependent reset procedure
Basics of operation

- Solution: reset procedure is modified according to the root-cause of the reset.
  - BEGIN event
  - MNGMT event
  - RECOVERY_TIMEOUT event



Root-cause dependent reset procedure
Basics of operation (BEGIN)

- BEGIN event: initialization or a full-reset of the node

- Elimination function:
  - When reset (BEGIN), collect and store the first "n" pieces of the received packets.
  - Select the packet with the largest sequence_number. If multiple packets exists with the largest sequence_number pick one of them). Note that the largest sequence_number is selected by taking into account the cyclic characteristics of the sequence number space. (Note also that, in correctly designed systems, the difference of the sequence_numbers of the "n" packets is within the history window.)
  - Forward only the selected packet and discard all the other stored packets.
  - Set "RecovSeqNum" to the sequence_number of the selected packet and set the history to all "1" ("SequenceHistory" = [1, … , 1])
  - Go back to normal operation of the Elimination function

"n" denotes the number of Member Streams

Root-cause dependent reset procedure
Basics of operation (MNGMT)

- MNGMT event: sequence recovery function reset requested by the management system via the "frerSeqRcvyReset" variable.
- Elimination function:
  - When reset (MNGMT), it is checked whether or not the first received packet can be accepted as follows:
    - If the sequence_number of the received packet is out of the history window (HSW: history window {RecovSeqNum + d; .. ; RecovSeqNum - d + 1}, where d = "frerSeqRcvyHistoryLength"), then packet is accepted. "RecovSeqNum" is set to the sequence_number of the received packet, and the history is updated to show the acceptance ("SequenceHistory" = [1, 0, ... , 0]).
    - If the sequence_number of the received packet is within the history window, then it is evaluated against the preserved values of "RecovSeqNum" and "SequenceHistory" whether or not the packet has been already received. If the packet has been already received, then the packet is dropped. If the packet has not been received yet, then packet is forwarded, and "RecovSeqNum" and "SequenceHistory" are updated accordingly.
  - Go back to normal operation of the Elimination function

Root-cause dependent reset procedure
Basics of operation (RECOVERY_TIMEOUT)

- RECOVERY_TIMEOUT event: the timeout mechanism triggers the reset

- Elimination function:
  - When reset (RECOVERY_TIMEOUT), accept the first packet received.
  - Forward the packet.
  - Set “RecovSeqNum” to the sequence_number of the received packet and update the history to (“SequenceHistory” = [1, 0, … , 0])
  - Go back to the normal operation of the Elimination function

Which solution to select?

Pros and Cons
Reset-guard vs. Root-cause dependent reset procedure

- Reset-guard
  - the implementation is simpler.
  - somewhat increases the outage because packets are dropped after the reset event for a while.
  - Note that many operational people treat “reset” functions as being not part of normal operation and delayed restore of operation caused by the guard-period might be negligible compared to the duration of the misbehaving characteristics that triggered the reset.

- Root-cause dependent reset procedure
  - the implementation is more complex.
  - can minimize the impact of the reset, which may be a requirement for some mission critical TSN applications. In such cases, it can be a better choice.

Impact on 802.1CB-2017

Target and Solutions at-a-glance
Avoid duplicate delivery due to reset of seq.recov function

- **Problem: reset of Sequence recovery function defined in 802.1CB-2017 may cause temporary duplicate delivery**

- **Target: ensure that no duplicate delivery happens due to reset of sequence generation function**
- How: change reset, two solutions are proposed
    1. Adding a reset-guard period after the reset, where received packets are dropped.
    2. Modifying the reset procedure to be root-cause dependent.

- **Impact: Duplicate delivery (even temporarily) is not acceptable for TSN networks as it breaks one of the basic design rules, namely a TSN Stream is not allowed to consume more than the resources reserved for it.**

802.1CB impact
What needs to be changed/added …

- Elimination node
  - Change “7.4.3.3 SequenceRecoveryReset”

Questions …