# 802.1CB FRER maintenance/improvements

## Reset of Sequence Recovery Function

IEEE 802.1 Maintenance TG
May, 2025

Balázs Varga, János Farkas Ericsson Research 2025-05-19

# 802.1CB FRER maintenance
## Overview

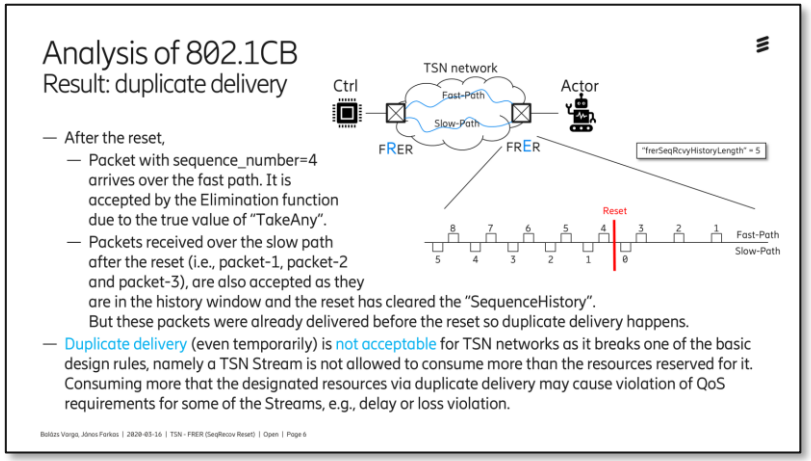
- **Problem:**
  **the reset of sequence recovery function defined in 802.1CB-2017 may cause stream anomalies**

- **Target:**
  **minimize the impact of sequence generation function's reset**

- How:
  to be discussed, e.g., fine-tune reset functionality

# Summary
## Impact of SequenceRecoveryReset

- Identified Issues of SequenceRecoveryReset
  - Duplicates
    - Caused by “too fast” reset procedure (“SequenceHistory” array is cleared during reset (filling with “0”)), where replicas of already received (before reset arrived) packet are still “in flight” in the network




  - Incorrect Loss Counter: frerCpsSeqRcvyLostPackets
    - Caused by clearing the “SequenceHistory” array (filling with “0”)
    - If no loss then “frerCpsSeqRcvyLostPackets” is increased by “#HistoryLength-1”
      else it is increased more

Note: Counters of LatentErrorDetection are not changed during Reset

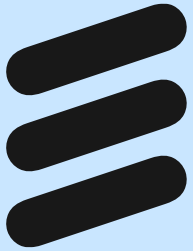
# Summary
## Impact of SequenceRecoveryReset

- 802.1CB-2017 defines three reasons to reset the Sequence recovery function:
  1. BEGIN event (initialization/reset), i.e., Node reboot
     - Issues: Duplicates, Incorrect Loss Counter
       Note: duplicates depends on initialization duration, if "TIME_init >> network latency", then no duplicates are expected
  2. Management event (frerSeqRcvyReset=true) and i.e., Operator action
     - Issues: Duplicates, Incorrect Loss Counter
       Note: this is just a reset of the recovery function what is pretty fast
  3. RECOVERY_TIMEOUT event (timeout mechanism expired). i.e., Stream issues
     - Issues: Incorrect Loss Counter
       Note: duplicates are not expected if RECOVERY_TIMEOUT is properly designed

# Summary
## Towards a solution

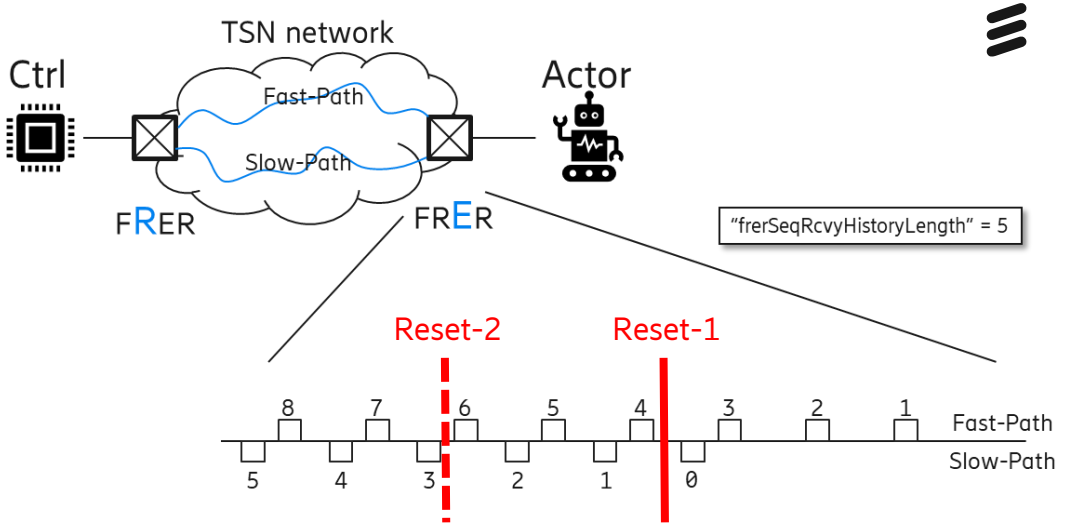
- 802.1CB-2017 defines three reasons to reset the Sequence recovery function:
  1. BEGIN event (initialization/reset), i.e., Node reboot
     - Issues: Duplicates, Incorrect Loss Counter
  2. Management event (frerSeqRcvyReset=true) and i.e., Operator action
     - Issues: Duplicates, Incorrect Loss Counter
  3. RECOVERY_TIMEOUT event (timeout mechanism expired). i.e., Stream issues
     - Issues: Incorrect Loss Counter
       Note: duplicates are not expected if RECOVERY_TIMEOUT is properly designed
     - **A possible solution (other may exist)**:
     - Replace :
       - "7.4.3.3 SequenceRecoveryReset" sets the "RecovSeqNum" to "RecovSeqSpace - 1", **clears** the "SequenceHistory" array and sets "TakeAny" to true.
     - With:
       - "7.4.3.3 SequenceRecoveryReset" sets the "RecovSeqNum" to "RecovSeqSpace - 1", **sets** the "SequenceHistory" array and sets "TakeAny" to true.

# Questions ...



Ericsson

# Recap: Which knob is what for?




- Proper sizing of frerSeqRcvyHistoryLength (SequenceHistory)
  - is a network engineering task
  - impacts the design of frerSeqRcvyResetMSec as well (timeout period in milliseconds for the RECOVERY_TIMEOUT event)
- There are contradicting requirements on sizing (can be also use-case specific)
  - frerSeqRcvyHistoryLength: Proper sizing is based on the (1) TSN Stream characteristics AND (2) FRER topology (latency difference of the paths used by the Member Streams)
    - As small as possible: e.g., to protect against failures / malicious nodes
    - As big as possible: e.g., to provide flexibility in case of short network failure(s)
  - frerSeqRcvyResetMSec: Proper sizing is based on the (1) TSN Stream characteristics AND (2) latency difference of the paths used by the Member Streams AND (3) value of frerSeqRcvyHistoryLength
    - As small as possible: e.g., to react fast to failure(s) and avoid unnecessary out of "history window" drops
    - As big as possible: e.g., to avoid unnecessary RECOVERY_TIMEOUT events
- A good design needs to find a trade off …

# Background slides

— Details of the problem and solution directions

# History on Contributions for Cloudification (e.g., vPLC)
## 2019-2020

- Reset causing duplicate delivery
  - https://www.ieee802.org/1/files/public/docs2020/maint-varga-FRER-sequence-recovery-reset-0320-v02.pdf
- Seamless Reset
  - https://www.ieee802.org/1/files/public/docs2020/new-varga-FRER-seamless-reset-0320-v02.pdf
- Explicit signaling of Reset via “SeqResetFlag”
  - http://www.ieee802.org/1/files/public/docs2019/new-varga-FRER-improvements-0719-v01.pdf

# History on Reset Related Maintenance Items
## 2025

- https://www.ieee802.org/1/files/public/docs2025/maint-arunarthi-CB-2017-item-378-0310-v02.pdf
- https://www.ieee802.org/1/files/public/docs2025/maint-mangin-CB-2017-item-378-0125-v01.pdf
- https://www.ieee802.org/1/files/public/docs2025/maint-mccall-cb-2017-item-378-v01.pdf

# Problem with existing reset mechanism

# Analysis of 802.1CB FRER
## 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.

# Analysis of 802.1CB FRER
## Notes

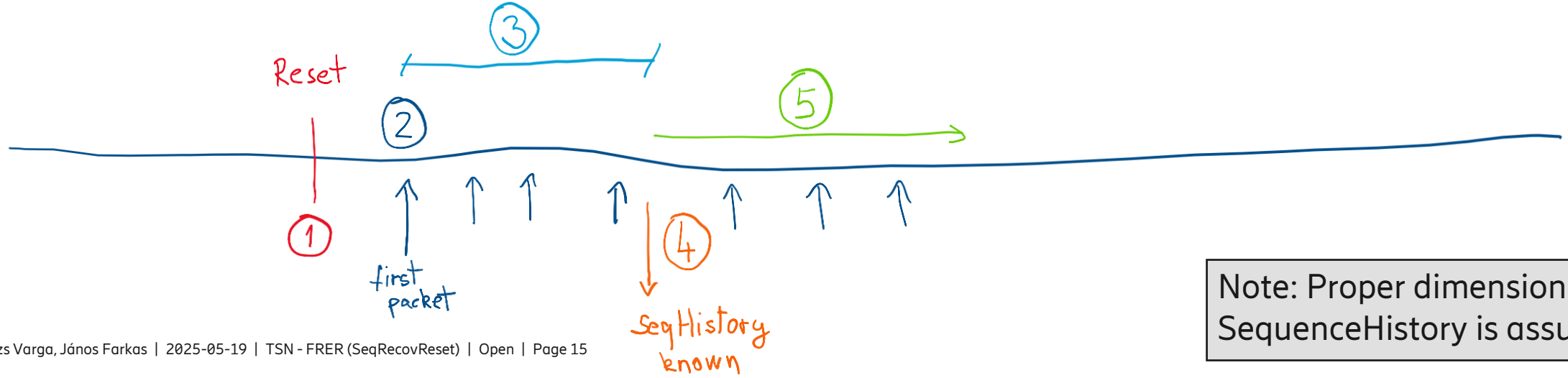
- Reset is an invasive network event
  - A transient on the operation of the system is expected,
    and such a transient SHOULD be treated as business as usual

- Reset (usually) happens in case of an "out-of-normal-operation"
  - 802.1CB-2017 defines three reasons to reset the Sequence recovery function:
    1. BEGIN event (initialization/reset), i.e., Node reboot
    2. Management event (frerSeqRcvyReset=true) and i.e., Operator action
    3. RECOVERY_TIMEOUT event (timeout mechanism expired). i.e., Stream issues

# Analysis of 802.1CB FRER
## Optimal operation in case of reset

— Seems to be quite complex

1. RESET happens
2. First packet of the stream received (path not known!) with SeqNum=N
3. Packet with SeqNum <= N may arrive for a time (PathsMaxDiffLatency), wait and collect SeqNums of incoming frames
4. All information about frame sequence numbers is known to properly fill in the SequenceHistory and e.g., identify missing frames
5. Back to normal operation

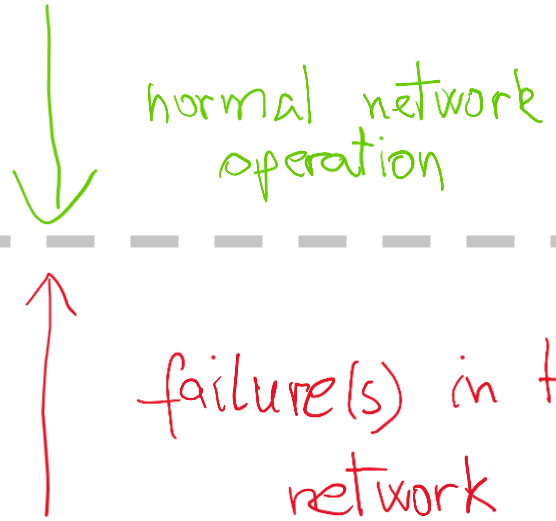



Note: Proper dimensioning of SequenceHistory is assumed

# Analysis of 802.1CB FRER
## Scenarios

Scenarios where RESET impacts e.g., stream counters

- Stream is off/pause -- > RECOVERY_TIMEOUT event -- > Stream on -- > e.g., incorrect loss counter




- Stream on + RESET event -- > possible duplicate delivery, incorrect loss counter, etc.
- Stream on + Loss on some Member Streams + RESET -- > possible duplicate delivery, incorrect loss counter, etc.
- Stream on + Fast-path is back from failure + RESET -- > possible duplicate delivery, incorrect loss counter, etc.
- ...

# Towards a solution …

# Analysis of 802.1CB FRER
## What do we intend to solve ???

Before discussing the solutions

- What do we intend to solve?
  - Under normal operation we do not want to have either false or false positive alarms!

- Can we live with some “multiple temporary anomalies at the same time” cases?
  - Transient after the RESET in single/multiple failure scenarios …
  - Some duplicates/false loss alarms/etc. during the transient
  - Wait for correct operation/alarms/counters e.g., 2 x “frerSeqRcvyLatentErrorPeriod” time