P802.1CBec

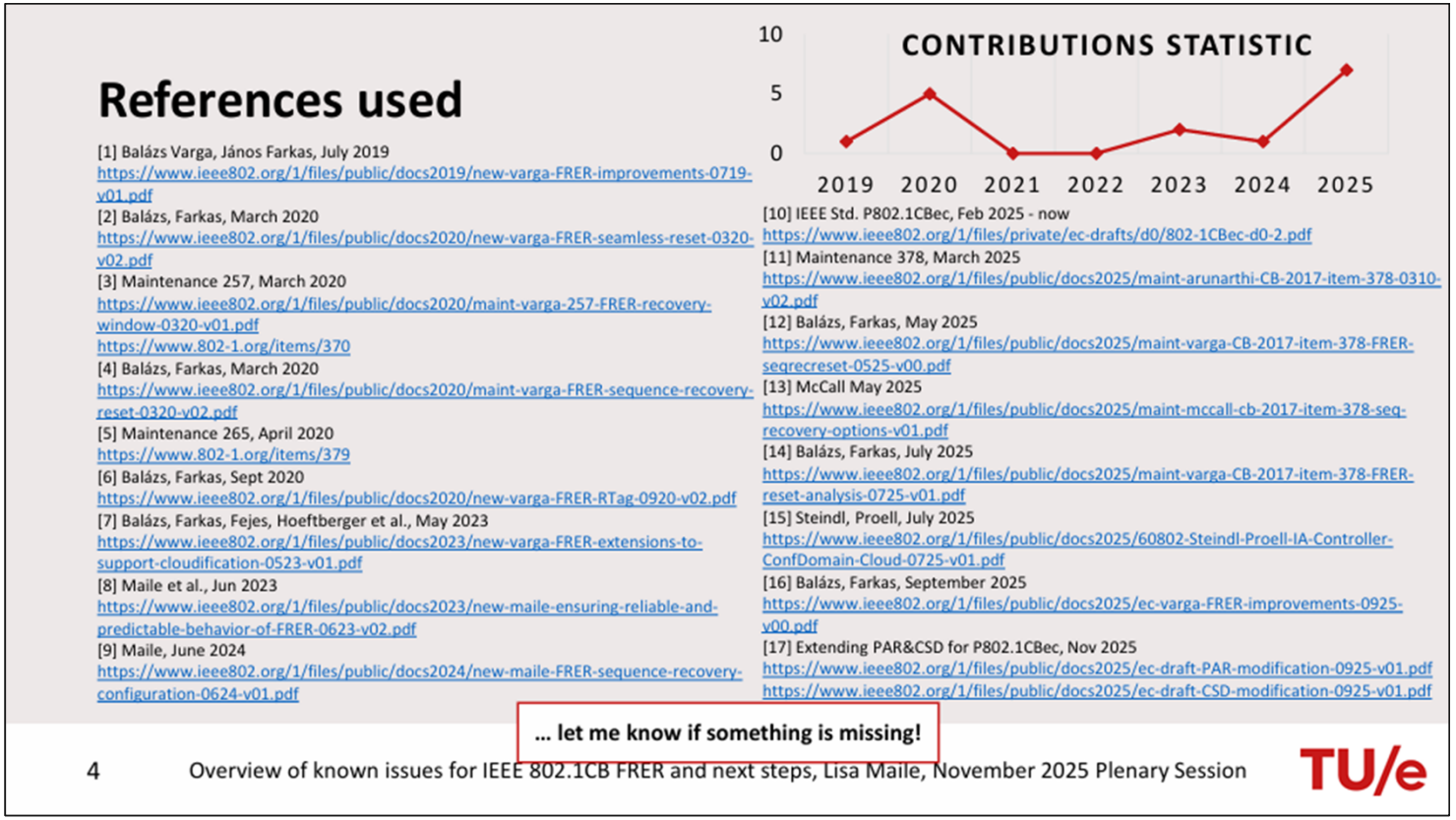
FRER Analysis + Solutions

Discussion material:
Lisa Maile,
David McCall,
Christophe Mangin,
Max Turner,
Balazs Varga

IEEE 802.1 TSN TG
2026 May

2026-05-11

History



https://www.ieee802.org/1/files/public/docs2025/ec-maile-FRER-issues-overview-1125-v03.pdf

Starting point for solution search

- Analysis of reset event details: The side effects of "one-size-fits-all"
  https://www.ieee802.org/1/files/public/docs2025/maint-varga-CB-2017-item-378-FRER-reset-analysis-0725-v01.pdf
  - RESULT: separation of Reset / Timeout functionality
    - The current single reset function is under-specified and sub-optimal …
    - Exclude the reaction to out-of-step situation from reset, and make it as an integral part of SeqRcvy

      Reset: frerSeqRcvyReset=True, e.g., for troubleshooting at duplication;
      Timeout: No new packets received at elimination. **Currently jointly covered.**
- Discussion on scenarios: Overview of known issues for IEEE 802.1CB FRER … and next steps
  https://www.ieee802.org/1/files/public/docs2025/ec-maile-FRER-issues-overview-1125-v03.pdf
  - RESULT: selected in-scope-scenarios
    - Member stream jitter [slide12]
    - Talker pause [slide13]
    - Lost on all member streams (optional) [slide14]
    - NFV (SeqGen reset, SeqRecov reset) [slide15]

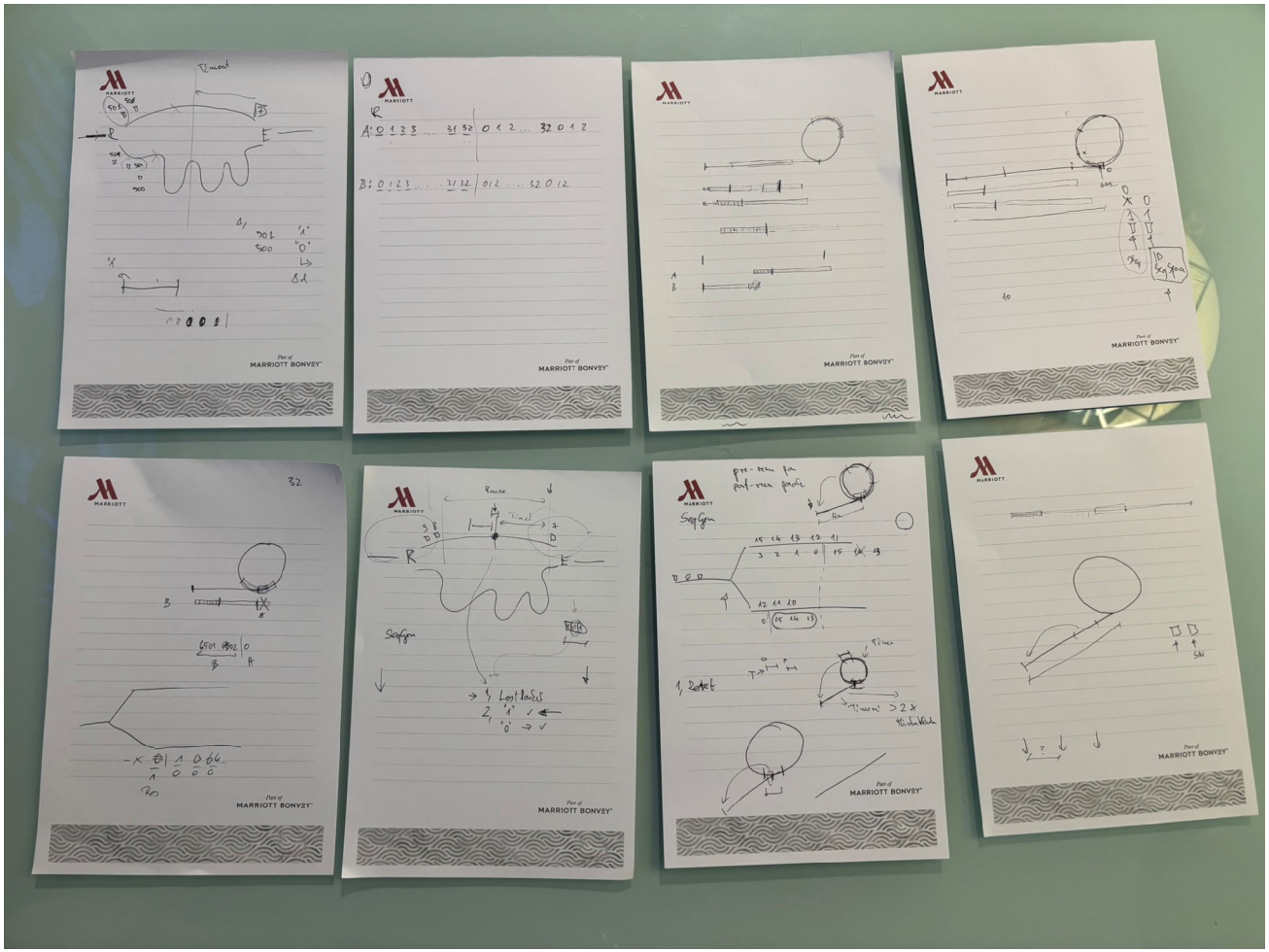
Two set of scenarios

- Set-1: Issue = "timeout" related (no immediate reaction needed)
  - Set1.1: Member stream jitter
  - Set1.2: Talker pause
  - Set1.3: Lost on all member streams (optional)

- Set-2: Issue = "reset" related (needs immediate reaction from the functionalities)
  - Set2.1: NFV SeqGen reset
  - Set2.2: NFV SeqRcvy reset

- LEMMA-1: IF "reset" and "timeout" are separated
  THEN the contradicting requirements on timeout are eliminated !!! (Timeout can be slow …)

Discussion notes …



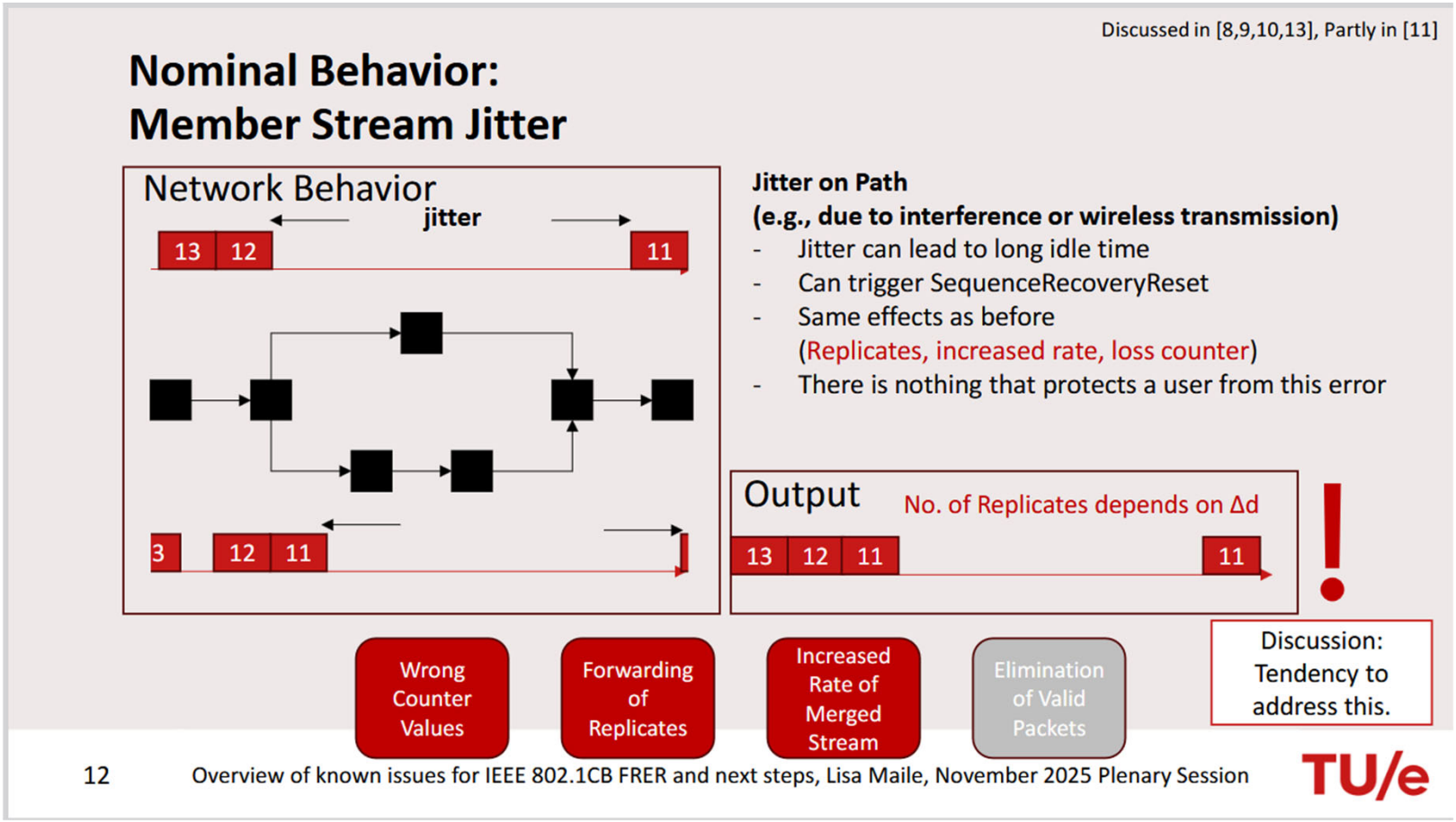
RECOVERY_TIMEOUT

Timeout specifics …

- SeqRcvy state is encoded in SequenceHistory
- Before RECOVERY_TIMEOUT (normal operation)
  - SequenceHistory determines what SeqNums are within the history window
- After RECOVERY_TIMEOUT (something happened)
  - No packets were received for a while
  - SequenceHistory might not represent the stream's state (outdated)
  - TakeAny to accept any received packet of the stream and initialize SequenceHistory

- Questions
  1. How to determine a safe RECOVERY_TIMEOUT period?
  2. What to do with SequenceHistory after a RECOVERY_TIMEOUT happened and how to count LostPackets?

Set1.1, Member stream jitter



https://www.ieee802.org/1/files/public/docs2025/ec-maile-FRER-issues-overview-1125-v03.pdf

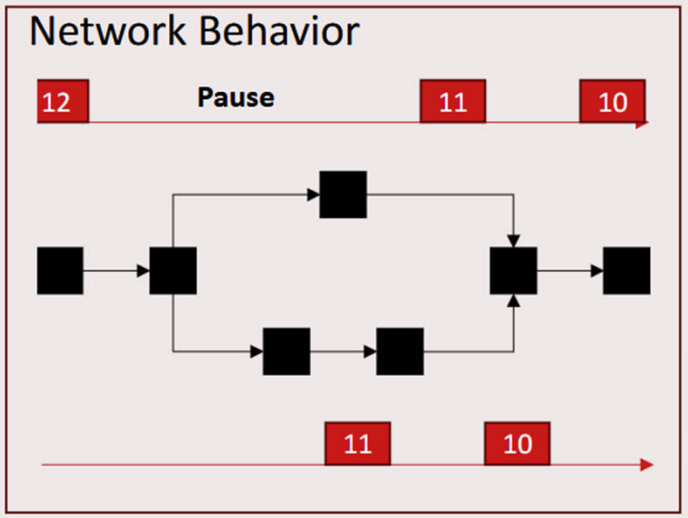
Set1.1, Member stream jitter

- Jitter on the path (e.g., due to interference or wireless transmission)
  - Root cause of the issue:
    timeout happens too early, a slower RECOVERY_TIMEOUT is needed
  - LEMMA-2:
    - There is a maximum/minimum network latency (MaxNetLatency/MinNetLatency)
    - If packet with “SN=n” received, then after “MaxNetLatency-MinNetLatency” no packet with “SN<=n” can arrive
- Solution components:
  - No explicit mechanism, make the reader aware of the situation in the standard
  - Design rules:
    - ΔD = MaxNetLatency – MinNetLatency, (ΔD_{max} = MaxNetLatency)
    - Wait long enough means: RECOVERY_TIMEOUT = ΔD (or higher …)
    - Engineered history window:
      frerSeqRcvyHistoryLength = Peak Rate of TSN stream / ΔD (or higher …)

Set1.2, Talker pause

Discussed partly in [14]

Nominal Behavior: Talker Pause



Talkers are allowed to pause or send less

- Pauses can be arbitrarily long and **arbitrarily often**
- Will trigger SequenceRecoveryReset
- Each time loss counter increases by #HistoryLength

Wrong Counter Values

Forwarding of Replicates

Increased Rate of Merged Stream

Elimination of Valid Packets

Discussion: Tendency to address this.

13 Overview of known issues for IEEE 802.1CB FRER and next steps, Lisa Maile, November 2025 Plenary Session

TU/e

https://www.ieee802.org/1/files/public/docs2025/ec-maile-FRER-issues-overview-1125-v03.pdf

Set1.2, Talker pause

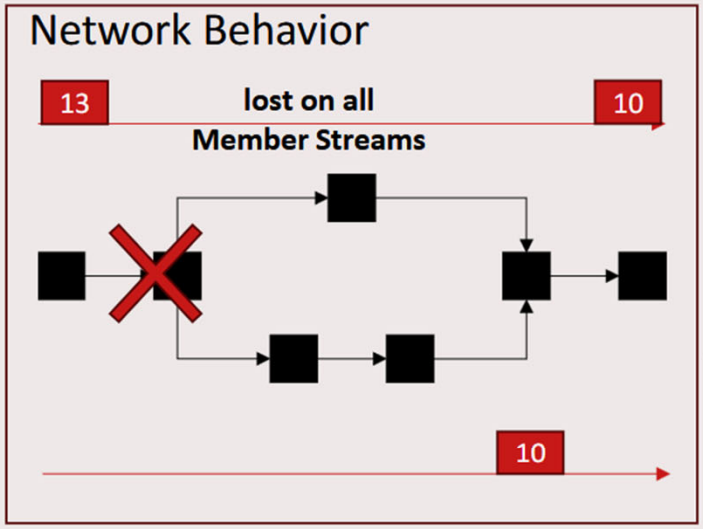
- Talker stops sending packets (might be rare events)
  - Keeping the SequenceHistory state is good as it is useful if the talker continues
  - Waiting longer for timeout does not hurt ("but…" will follow in Set1.3)
  - Root cause of the issues:
    - After timeout SequenceHistory filled with "0"s -- > LostPackets counter wrong
- Solution components
  - RECOVERY_TIMEOUT = must be at least ΔD (or even longer …)
  - Increased history window:
    frerSeqRcvyHistoryLength = Peak Rate of TSN stream / ΔD (or higher …)
  - Evaluate /report "0"s in SequenceHistory when timeout happens
    (missed packet statistics are relevant immediately, i.e., for the time period before talker paused!)
  - Possible solutions:

Fill SequenceHistory with "1"s	Fill SequenceHistory with "0"s
(safe to do so, as no packets in-flight within the network after timeout)	+ modify LostPacket counter ("0" does not count as loss until first "1" seen in SequenceHistory)

Set1.3, Lost on all member streams (optional)

Discussed in [8,9,10,13], Partly in [2,12,14]

Failure Behavior: Failure before Duplication



All Replicates are lost
(e.g., due to failure before duplication if FRER is in relay node or failure on all links)

- Time of failure can be arbitrary
- Next Sequence Number can be arbitrarily higher (e.g., 0x13 or any higher number)
- Valid packets discarded (e.g., 0x13 and following) → outside current Seq. Window
- Will trigger SequenceRecoveryReset after timeout, then same issues as for **Talker Pause** (loss counter)
- Can be covered as trade-off between the Reset Timeout value and History Length, but careful config. required.

Wrong Counter Values | Forwarding of Replicates | Increased Rate of Merged Stream | Elimination of Valid Packets

Discussion: Open.

15 Overview of known issues for IEEE 802.1CB FRER and next steps, Lisa Maile, November 2025 Plenary Session

TU/e

https://www.ieee802.org/1/files/public/docs2025/ec-maile-FRER-issues-overview-1125-v03.pdf

Set1.3, Lost on all member streams (optional)

- No packets reach SeqRcvy function
  - Hard to distinguish from talker pause (not always possible)
  - SequenceHistory state is good only for a while (= ΔD)
  - Root cause of the issues:
    - TakeAny: First packet may not have the lowest SeqNum of the in-flight packets
    - After timeout SequenceHistory filled with "0"s -- > LostPackets counter is wrong
- Solution components
  - RECOVERY_TIMEOUT = ΔD (waiting longer hurts – the "but" compared to Set1.2)
  - Engineered history window: frerSeqRcvyHistoryLength = Peak Rate of TSN stream / ΔD
  - Evaluate /report "0"s in SequenceHistory when timeout happens
  - One possible solution:

Fill SequenceHistory with "0"s

+ modify LostPacket counter ("0" does not count as loss until first "1" seen in SequenceHistory)

Sum of new functionalities for Set1

- RECOVERY_TIMEOUT = ΔD
  (timeout = the duration when the SequenceHistory get outdated after the last accepted packet, if a bigger history window is used, timeout should be adapted)
- Engineered history window:
  frerSeqRcvyHistoryLength = Peak Rate of TSN stream / ΔD

- Evaluate "0"s in SequenceHistory when timeout happens
  (missed packet statistics are relevant immediately, i.e., for the time period before talker paused!)

- Fill SequenceHistory with "0"s when timeout happens
  + modify LostPacket counter rules for a while ("0" does not count as loss until first "1" seen in SequenceHistory)

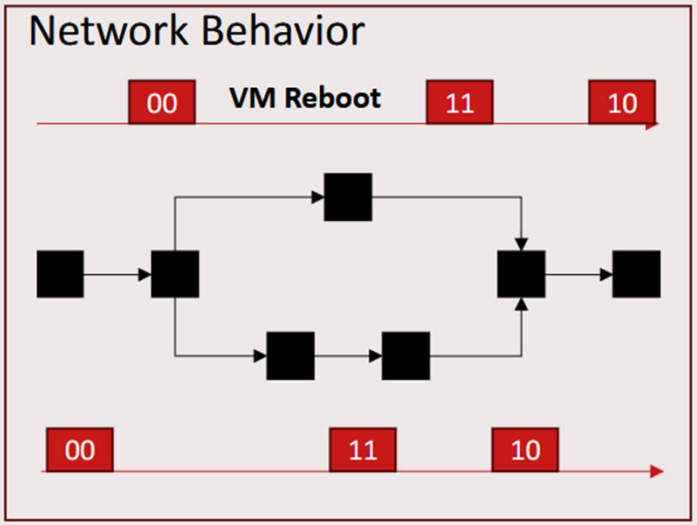
RESET

Set2.1, Reset of SeqGen

Discussed in [1,2,4,6,7,12,16]

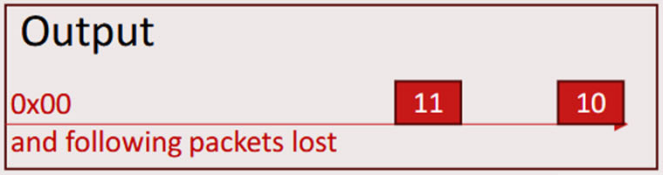
Nominal Behavior:

Network Function Virtualization (NFV)



Virtualization (e.g., containers, virtual machines)

- FRER functionality might be part of the cloud
- Virtual instances can restart fast (e.g., replication function)
- Next SeqNum is the first value of the SeqNumSpace (i.e., 0x00)
- No timeout → Seq. Window not reset
- Valid packets discarded (e.g., 0x00 and following) → outside current Seq. Window
- Will trigger SequenceRecoveryReset after timeout, then same issues as for **Talker Pause** (loss counter)



Wrong Counter Values

Forwarding of Replicates

Increased Rate of Merged Stream

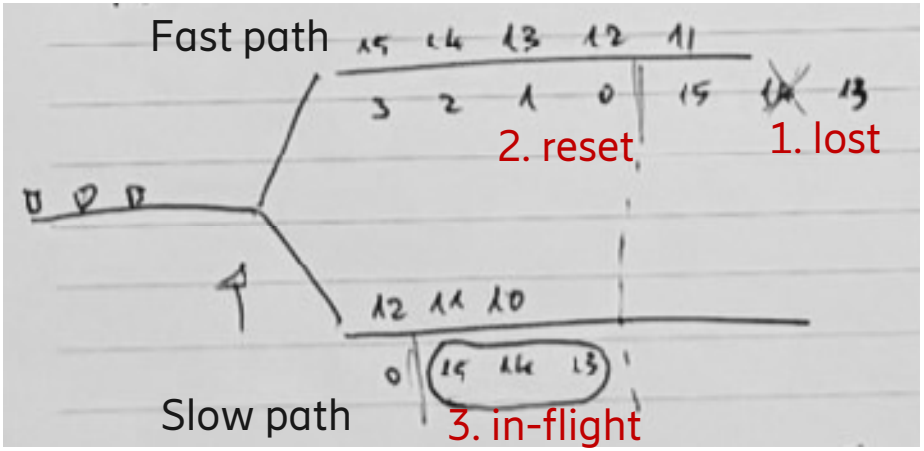
Elimination of Valid Packets

Discussion: Tendency to address this.

14 Overview of known issues for IEEE 802.1CB FRER and next steps, Lisa Maile, November 2025 Plenary Session TU/e

https://www.ieee802.org/1/files/public/docs2025/ec-maile-FRER-issues-overview-1125-v03.pdf

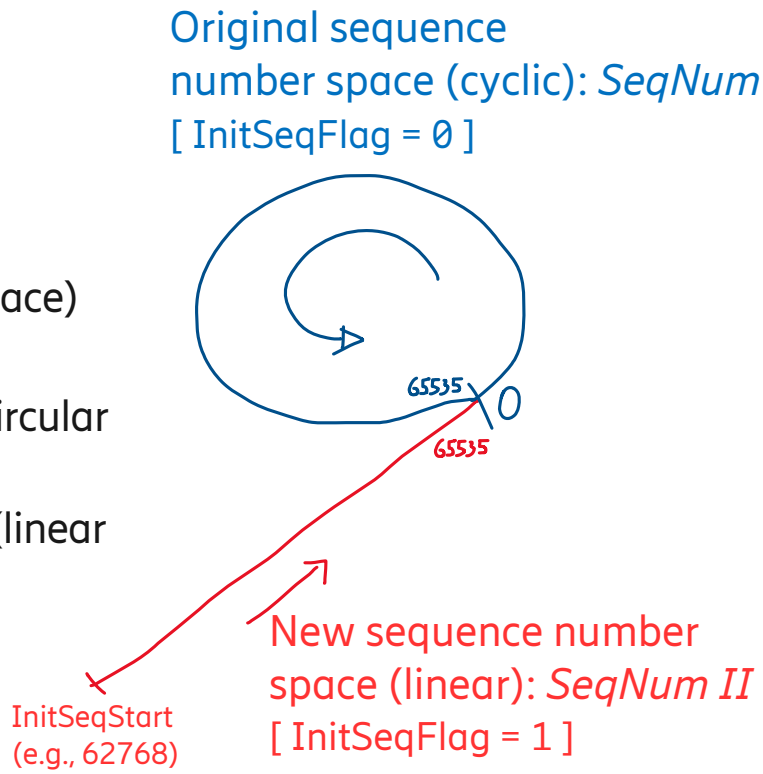
Set2.1, Reset of SeqGen



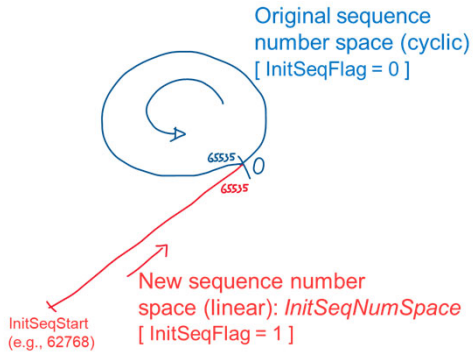
- Reset of SeqGen
  - Possible "jump" of SeqNum in the packet stream
  - Needs immediate reaction within the network
  - Can be non-intrusive if
    (1) reset is done fast on the SeqGen and
    (2) downstream functions can recognize the reset
  - Root cause of the issues:
    - RC-1: SeqNum gets out of history-window -- > drop (of post-reset packets)
    - RC-2: SeqNum within history-window ("history overlap") -- > drop (pre/post-reset packets)
  - How to distinguish pre/post-reset packets? What to do with in-flight packets?
- Solution:
  - InitSeqNumSpace: distinguish pre/post-reset packets (solves RC-1 and RC-2, deals with pre/post-reset in-flight packets)

802.1CBec improvements
RESET

- How: change Sequence Number Space characteristics
  1. extending the sequence number space (additional new linear initial sequence number space)
  - Extending the SeqNum space
    - SeqNum space I.: used during normal operation (circular sequence number space)
    - SeqNum space II.: used after SeqGenReset event (linear sequence number space)
    - SeqNum spaces are distinguished via a flag "InitSeqFlag" ["0": SeqNum-I, "1": SeqNum-II]

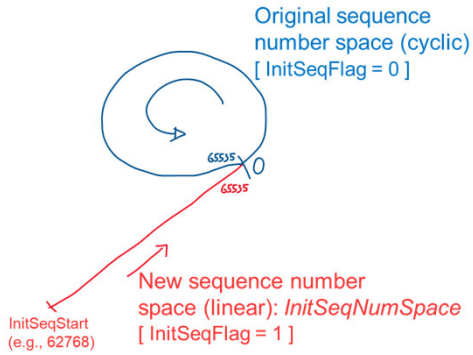


NEW Sequence Number Space
Basics of operation



- Solution: introducing a new additional sequence number space SeqNum space II ("*InitSeqNumSpace*"), which is used after initialization /reset of the sequence generation function.
  - When this new number space is exhausted the sequence generation function starts to use the original sequence number space, which is illustrated by the blue cyclic sequence number space.
- New flags:
  - Usage of the new Sequence Number Space is marked by a new flag, namely the "InitSeqFlag".
- Examples for detection of a SeqGen reset on downstream nodes:
  - IF SeqNum space I. is in use and a packet received with set InitSeqFlag, THEN it refers to the event that the SeqGen was reset. Packets in both spaces (new and "in-flight") are handled safely.
  - IF SeqNum space II. is in use and a packet received with set InitSeqFlag AND its SeqNum is lower than current history window, THEN a new reset of SeqGen is detected. But packets "in-flight" before this new reset are not tracked.

NEW Sequence Number Space
Details of operation (InitSeqNumSpace)



"*InitSeqNumSpace*" is used as follows:

— When sequence generation function was initialized/reset, the "*InitSeqNumSpace*" is used to generate the sequence number for the packets.

— "*InitSeqNumSpace*" has its own set of variables that are used to process the sequence_number of a packet by the Replication function and Elimination function.
Note: *InitSeqNumSpace* variable names are derived from the variable names of the original sequence number via appending them with an "*Init*" prefix, e.g., "*InitGenSeqNum*", "*InitRecovSeqNum*", "*InitSequenceHistory*", "*InitTakeAny*", "*InitRemainingTicks*", etc.

— "*InitSeqNumSpace*" length can be determined "automatically" from the sequence interval length (should be not too long, not too short)

Set2.2, Reset of SeqRcvy

Currently under discussion

- Reset of SeqRcvy
  - Intrusive by nature: Lost of local state (SequenceHistory)
  - Duration of reset depends on implementation (no packet processing during reset!)
  - Root cause of the issues:
    - There might be in-flight packets in the network
    - Risk of duplicates / drops due to the lost state
    - Which (duplicate/drop) hurts more? (-- > use-case dependent)
- Solution:
  - TimeOfReset: duration of executing reset (implementation specific), Network designer should consider duration of reset when setting PostResetGuardTimer
  - PostResetGuardTimer: delays start of packet processing of SeqRcvy (packets are dropped until timer expires)
    - PostResetGuardTimer = "0", results in minimum packet loss, but possible risk of duplicates
    - TimeOfReset + PostResetGuardTimer = "ΔD", results in no duplicates / no drops
    - TimeOfReset + PostResetGuardTimer > "ΔD", risk of packet loss

Sum of new functionalities

- SeqGen, SeqRcvy
  - Using additional sequence number space (InitSeqNumSpace)
  - Flag: InitSeqFlag (indicator of SeqNumSpace)

Changes in .1CBec

Tbd.

Discussion

Root cause / Fixing
Main rules

Root cause of issues (It is all about SeqNum)

- SeqGen/SeqRcvy have their own internal states those drive their operation.
- SeqGen/SeqRcvy functions require to be "in-step" (i.e., having the same view on the expected SeqNum range of the served Stream's packets being under forwarding in the network)
- There is no state communication between SeqGen/SeqRcvy, so identification of "out-of-step" situation is based on timer-based guesses driven by the observed SeqNum of received Stream's packets.
- Timer based reaction to "SeqNum out-of-step" situations is hard to achieve and has side effects.

Fixing proposal (Explicit Signaling)

- Adding explicit signaling on SeqNum specific actions between SeqGen and SeqRcvy
  - Signaling can be in-band & unidirectional (SeqGen -> SeqRcvy)
- Keep timer based reaction for extreme failure scenarios as a last resort (e.g., all paths fail)
- Optional: adding flexibility to SeqGenReset operation (i.e., start value of SeqNum)