TSN Command-Line Guide

Application Note
V1.0

Table of Contents

1. Introduction

This document explains how to setup Time-Sensitive Networking (TSN) features.

Time-Sensitive Networking is a number of IEEE 802 standards that are defined by the IEEE TSN task group. These standards define mechanisms for deterministic real-time communication over Ethernet networks.

The following links gives an overview of the purpose and current status of the various TSN standards:

- https://en.wikipedia.org/wiki/Time-Sensitive_Networking
- http://www.ieee802.org/1/pages/tsn.html

2. Using TSN functions with PTP time synchronisation

When using TAS and PSFP between network elements, it is required to have a common global time reference provided by PTP. When booting the device, it will take some time for a configured PTP application to get locked to the common time reference. It may cause malfunctioning of TAS and PSFP if config-change is issued before PTP time is in a Locked or Locking state. A function which can delay the issue of config-change until PTP is Locked/Locking or a configurable time has passed, can be configured with the CLI command: tsn ptp-check

The configuration of PTP is out of scope for this configuration guide. The

syntax for TSN delayed start function is:

```
tsn ptp-check procedure { none | ptp | wait }
tsn ptp-check ptp-port <0-3>
tsn ptp-check timeout <10-200>
no tsn ptp-check procedure
no tsn ptp-check ptp-port
no tsn ptp-check timeout
```

Where:

none	Procedure: Start TSN functions immediately without any delay
ptp	Procedure: Monitor the status of PTP time. Start if it is Locking or Locked. If Locking or Locked is not achieved within wait time, then start anyway
wait	Procedure: Wait timeout number of seconds before starting TSN functions
	ptp-port The PTP port to use for sensing PTP status
timeout	Set ptp-check-procedure timeout in seconds

An example is shown below:

```
(config)# tsn ptp-check procedure ptp
(config)# tsn ptp-check ptp-port 2
(config)# tsn ptp-check timeout 30
```

It is intended that after configuring tsn ptp-check, it should be saved to startup-config, and the delay function will only be executed once after a 'power on' or 'reload cold'.

3.Time Aware Shaper

Time Aware Shaper is defined in the IEEE 802.1Qbv standard and is the ability to allow transmission from each queue to be scheduled relative to a known global timescale.

The global time is maintained by using a specific version of Precision Time Protocol (PTP) as defined in IEEE 802.1AS-Rev.

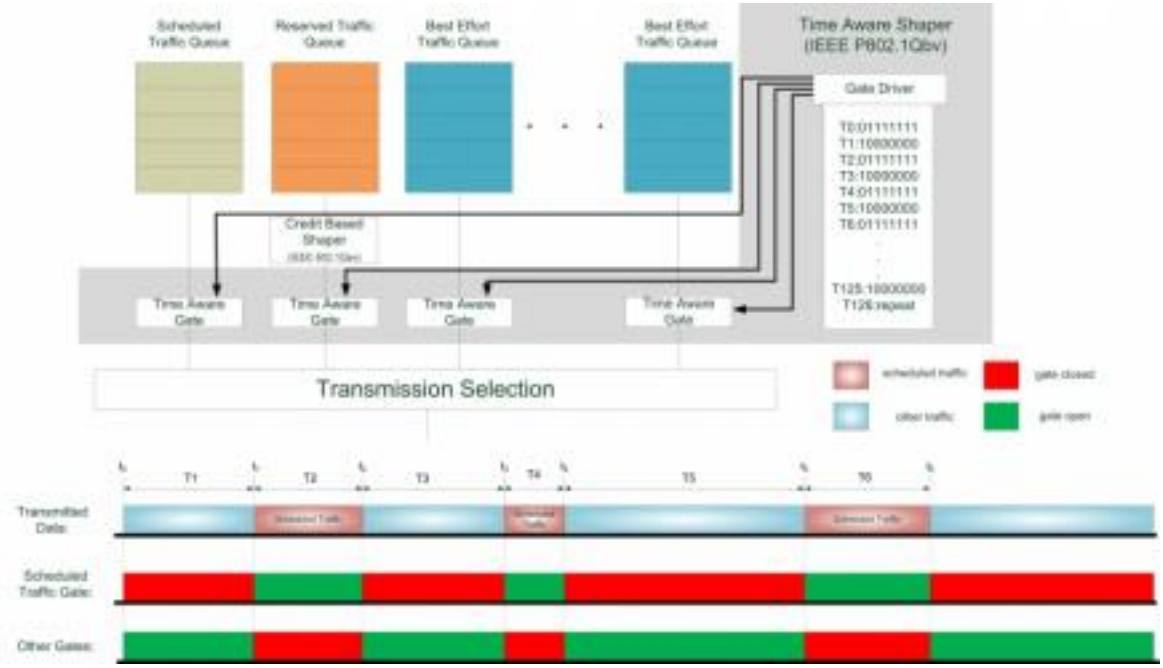

Figure 2. Time Aware Shaper

3.1. Configuration of parameters

The syntax for TSN TAS global level CLI configuration command is:

```
tsn tas always-guard-band
no tsn tas always-guard-band
```

Where:

```
always-guard-band:     Guard band is implemented for any queue to scheduled queues
                       transition.
no always-guard-band:  Guard band is implemented for non-scheduled queues to
                       scheduled queues transition.
```

The syntax for TSN TAS interface level CLI configuration command is:

```
tsn tas base-time seconds <seconds> nanoseconds <nanoseconds>
tsn tas config-change
tsn tas control-list index <index> gate-state queue <queue> { open | closed } time-interval
<interval> [ operation { set | set-hold | set-release } ] tsn tas control-list-length <length>
tsn tas cycle-time <time> { ms | us | ns }
tsn tas cycle-time-extension <extension>
tsn tas gate-enabled
tsn tas gate-states queue <queue> { open | closed }
tsn tas max-sdu queue <queue> <sdu>
```

Where:

```
The following parameters are defined in IEEE802.1Q: ieee8021STMib
base-time                   Admin Base Time.
config-change               Start a configuration change
control-list                Admin Control List
control-list-length         Admin Control List Length
cycle-time Admin            Cycle Time
cycle-time-extension Admin  Cycle Time Extension
gate-enabled                Enabled state of Time Aware Shaping
gate-states                 Initial gate state for each queue
max-sdu                     Queue Max SDU configuration
queue                       Traffic class. 0-7.
index                       Index of Gate Control Entry
gate-state                  Admin Gate State
time-interval               Time Interval in Nanoseconds
operation                   set | set-hold | set-release
```

3.2. Configuration example

IMPORTANT Add Web interface screen dumps and explanation here! The

equivalent CLI commands are:

```
! Create and start a Time Aware Shaper schedule on port 1.
! The schedule contains three gate control entries:
! 0: Open queue 7 and close all other queues for 20 milliseconds
! 1: Open queue 5-6 and close all other queues for 30 milliseconds
! 2: Open queue 0-4 and close all other queues for 50 milliseconds
! The schedule is repeated every 110 milliseconds
! TAS is scheduled to start at seconds 4300 nanoseconds 500
# configure terminal
! Disable always-guard-band
(config)# no tsn tas always-guard-band
(config)# interface 10GigabitEthernet 1/1
!
! Set max sdu size for queue 5 to 512 bytes
(config-if)# tsn tas max-sdu queue 5 512
!
! Enable Time Aware Shaper
(config-if)# tsn tas gate-enabled
!
! Set cycle-time to 110 milliseconds
(config-if)# tsn tas cycle-time 110 ms
!
! Set tsn tas cycle-time-extension 9000
(config-if)# tsn tas cycle-time-extension 9000
!
! Set gate state for queues 0-3 to closed
(config-if)# tsn tas gate-states queue 0-3 closed
!
! Set start time of schedule
(config-if)# tsn tas base-time seconds 4300 nanoseconds 500
(config-if)# !
!
! Configure gate control list
(config-if)# tsn tas control-list-length 3
(config-if)# tsn tas control-list index 0 gate-state queue 7 open time-interval 20000000
operation set-hold
(config-if)# tsn tas control-list index 1 gate-state queue 5-6 open time-interval 30000000
operation set-release
(config-if)# tsn tas control-list index 2 gate-state queue 0-4 open time-interval 50000000
operation set
! Start schedule
(config-if)# tsn tas config-change
```

```
Show configuration using 'show running-config'.
# show running-config
no tsn tas always-guard-band
[...]
interface 10GigabitEthernet 1/1
tsn tas max-sdu queue 5 512
tsn tas gate-enabled
tsn tas gate-states queue 4-7 open
tsn tas cycle-time 110 ms
tsn tas cycle-time-extension 9000
tsn tas base-time seconds 4300 nanoseconds 500
tsn tas control-list-length 3
tsn tas control-list index 0 gate-state queue 7 open time-interval 20000000 operation set-
hold
tsn tas control-list index 1 gate-state queue 5,6 open time-interval 30000000 operation
set-release
tsn tas control-list index 2 gate-state queue 0-4 open time-interval 50000000 operation
set
tsn tas config-change
```

3.3. Status

IMPORTANT Add Web interface screen dumps and explanation here! The

equivalent CLI commands are:

```
! Show status using 'show tsn tas status'.
# show tsn tas status interface 10GigabitEthernet 1/1
interface
GateEnabled                    : TRUE
OperGateStates                 : 0x1f
OperCycleTime                  : 110 ms
OperCycleTimeExtension         : 9000 nanoseconds
OperBaseTime                   : 4300 seconds, 500 nanoseconds
ConfigChangeTime               : 4300 seconds, 500 nanoseconds
TickGranularity                : 0 tenths of nanoseconds
CurrentTime                    : 4311 seconds, 827669856 nanoseconds
ConfigPending                  : FALSE
ConfigChangeError              : 0
SupportedListMax               : 256
OperControlListLength          : 3
GateControlEntry 0             : GateStates 0x80, TimeInterval 20000000
nanoseconds, GateOperation set-hold
GateControlEntry 1             : GateStates 0x60, TimeInterval 30000000
nanoseconds, GateOperation set-release
GateControlEntry 2             : GateStates 0x1f, TimeInterval 50000000
nanoseconds, GateOperation set
```

```
! Disable Time Aware Shaper on port 1.
# configure terminal
(config)# interface 10GigabitEthernet 1/1
(config-if)# no tsn tas gate-enabled
```

3.4. Configuration considerations

The max-sdu parameter is used only to calculate the guard band time: gbt = max sdu*8 / LINK_SPEED. Please note, that on Fireant, frames larger than max-sdu are not rejected.

The max-sdu is defined for each interface,queue and as a result, gbt's can be configured for each traffic class on an interface.

The required guard band time can be reduced if preemption is used. If the traffic class being closed consist of preemptible frames, and the class being opened consists of express frames, then a set-hold operation can be included as part of the gate operation. This causes any currently transmitting preemptible frame to be preempted, reducing the latency before the port is ready to transmit express frames.

When a gate operation closes an input in a scheduler element, that input is permanently blocked until another gate operation opens it again. Similarly, a set-hold on a port remains in effect until another gate operation does a set-release.

This needs to be kept in mind when stopping a TAS list. If the last gate operation in the TAS list leaves any scheduler element input closed, or leaves a set-hold in effect, they can be left indefinitely, possibly causing frames to be blocked in the switch.

If one of the gate operations in a TAS list opens everything, then the TAS list can be arranged so that this is the last operation in the TAS list. A TAS list always completes its cycle before stopping, thus this leaves everything open after the TAS list is stopped.

Alternatively, after stopping a TAS list that leaves inputs closed or set-hold in effect, it is necessary to configure a dummy TAS list with an "open all" gate operation and run it for one cycle.

3.4.1. Configuration of Gate Control Entry

A GCE consists of 3 elements:

- gate-state: Specify for each queue wether the gate shall be open or closed in this interval.
- time-interval: The time in nanoseconds where the gate have the open/close state as defined by the gate-state parameter.
- operation: The value may be one of set, set-hold, set-release.
  These options are as defined by IEEE 802.1Q-2018, table 8-7.
  - set: The gates are immediately set to the states indicated in the gate-state parameter. After time-interval have elapsed control passes to the next gate operation.

- set-hold: Performs all of the actions defined for the set operation. In addition, the start of this operation marks the point in the sequence of gate operations at which the MAC associated with the port is to have stopped transmitting preemptible frames. If frame preemption is not supported or not enabled, this operation behaves the same as set operation.
- set-release: Performs all of the actions defined for the set operation. In addition, the start of this operation marks the point in the sequence of gate operations at which the MAC associated with the port is permitted to resume transmitting preemptible frames; if an express frame is currently being transmitted by the MAC, the release takes effect at the end of that transmission. If frame preemption is not supported or not enabled this operation behaves the same as set operation.

The value of a time-interval should always be larger than the guard band time (as specified thru the values of max-sdu and LINK_SPEED).

An open queue will always be opened for a small amount of time, even if the guard band time is larger than the configured time-interval.

There are some restrictions:

For a GCE with set-hold, all queues opened must be Express queues. For a GCE with set-release all queues opened must be Preemptable queues. The same queue cannot be open in both a set-hold and a set-release operation.

3.4.2. Configuration of Gate Control List

A Gate Control List (GCL) is a list of gate control entries (GCE). A GCL is configured by the control-list parameter. The number of GCEs in a control-list is defined by control-list-length parameter.

When defining a control-list, start with setting control-list-length. Then configure each GCE. The sum of all time-interval in a control-list must be equal to or less than the cycle-time. Each queue must be open in at least one QCE.

3.4.3. Configuration of always-guard-band option.

The always-guard-band defines how the guard band values are calculated and has the following effect:

If a GCL do not contain set-hold and/or set-release operations the always-guard-band has no effect.

If a GCL do contain set-hold and set-release operations then:

- When always-guard-band=true a guard band is implemented on all queues, both Express and Preemptible queues.
- When always-guard-band=false a guard band is only implemented on Preemptible queues.

3.4.4. Calculation of Guard Band Times.

The Maximum SDU size parameter is used to calculate the guard band time: gbt = max_sdu[]*8 / LINK_SPEED
If frame preemption is enabled and a gate operaton is set-hold, the guard band time in preemptable queues is automatically selected as the frame preemption min fragment size plus 64 bytes.

NOTE : A queue is said to be preemptible, if frame preemption is enabled, and if this queue is not opened in a set-hold gate operation.

3.4.5. Using config-change and base-time

The command "tsn tas config-change" signals the start of a configuration change. If the value of parameter base-time is in the future, the configuration change will be executed at base-time. If base-time is in the past, the configuration change will be executed as soon as possible. In practice it will be within approx 2 seconds, at a time which is an integral number of cycle-times ahead of the configured value of base time. This way, the synchronisation between schedules in elements across a sceduled network can be maintained.

3.4.6. Uncertainty related to last frame in TAS gate open interval

In the Fireant implementation of Time Aware Shaper function (IEEE 802.1Q-2018, Enhancements for Scheduled Traffic), frames that are buffered in the disassembler fifo on the egress port when the TAS gate closes are transmitted after the gate close time Ts.

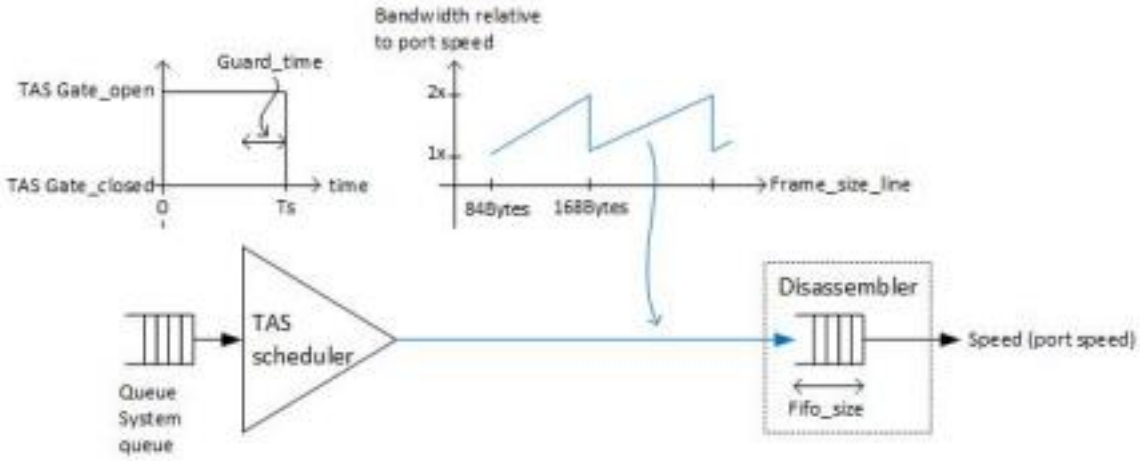

Figure 3. Time Aware Shaper

In a test where same size frames are transmitted using TAS, the maximum number of frames transmitted, N, is calculated as follows:

Ts: gate open time interval
max_sdu_line: configured value (including 8 bytes preamble + 12 bytes IFG) speed: link speed
guard_time: max_sdu_line * 8 / speed
frame_size: size of frames sent
frame_size_line: size of frames sent including 8 bytes preamble + 12 bytes IFG fifo_size: size of fifo on the egress port. Depends on port speed
N = (Ts - guard_time) * speed/frame_size_line + Max(fifo_size/frame_size, 1)

Example:

Ts = 100 us
speed = 1000,000,000 bit/sec
frame_size = 148 bytes
frame_size_line = 168 bytes
max_sdu_line = 276 bytes
guard_time = 276 * 8 / 1000,000,000 = 2.208 us
fifo_size: 1024 bytes (1G interface)
N = (100 us - 2.208 us) * 1000,000,000 / (168*8) + Max(1024/148, 1)
N = 72,76 + 6,9 = 79.68 frames

Frames can be prevented from transmission after Ts by increasing the guard time to accommodate for the amount of traffic queued up in the disassembler fifo as follows:

guard_time_safe * speed/frame_size_line = Max(fifo_size/frame_size, 1) <=>
guard_time_safe = Max(fifo_size/frame_size, 1) * frame_size_line/speed

Using the parameters from Example 1:

guard_time_safe = 6,9 * 168 bytes * 8bits/byte / 1000,000,000 bit/sec = 9.3us

And thereby:

max_sdu_safe = guard_time_safe * speed / 8 bits/byte = 1162 bytes
max_sdu_line = 276 bytes + 1162 bytes = 1438 bytes.

4. Frame Preemption

Frame Preemption is defined in the IEEE 802.1Qbu and IEEE 802.3br standards.

Frame Preemption is the ability to suspend the transmission of a non time-critical frame, and allow for one or more time-critical frames to be transmitted. When the time-critical frames have been transmitted, the transmission of the non time-critical frame is resumed. A non time-critical frame could be preempted multiple times.

The use of Frame Preemption with Time Aware Shaping reduces the guard band needed from the size of the largest possible interfering frame to the size of the largest possible interfering fragment as shown below.

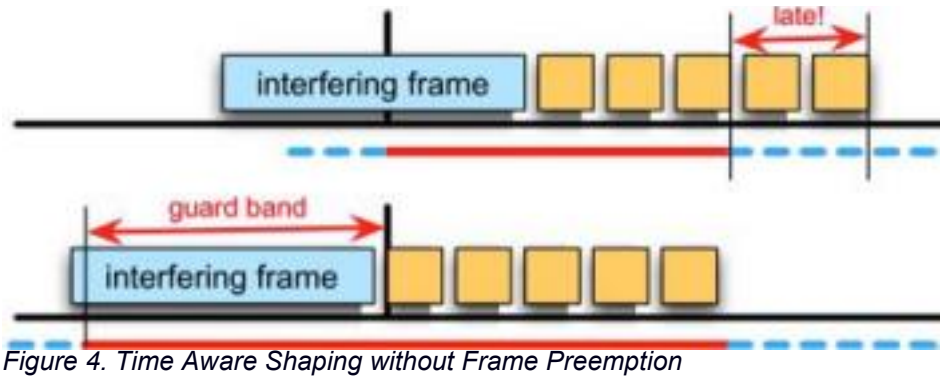

Figure 4. Time Aware Shaping without Frame Preemption

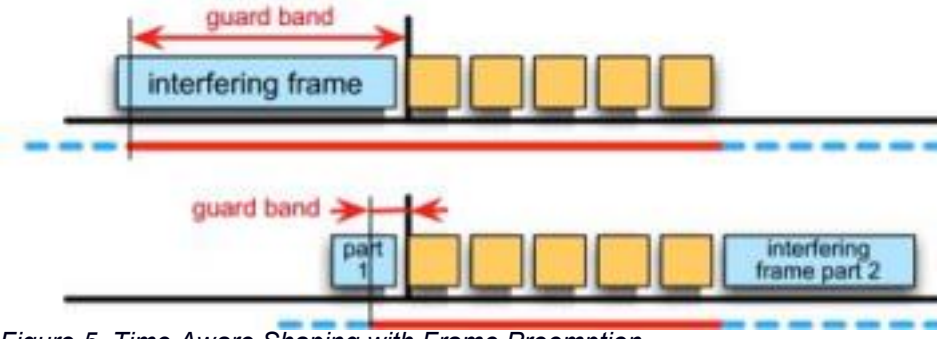

Figure 5. Time Aware Shaping with Frame Preemption

Frame Preemption must be enabled on both egress port level and on egress port/ queue level.

Frame Preemption is disabled by default on port level and disabled by default on port/ queue level.

4.1. Configuration

IMPORTANT Add Web interface screen dumps and explanation here! The

equivalent CLI commands are:

```
! Enable frame-preemption on port level for port 1.
# configure terminal
(config)# interface GigabitEthernet 1/1
! Enable frame-preemption on port (disabled by default)
(config-if)# tsn frame-preemption
```

```
! Do not wait to receive lldp message before enabling frame-preemption in transmit direction
```

```
! Disable frame-preemption on on queue level for port 1, queue 0 and 1. # configure terminal
(config)# interface GigabitEthernet 1/1
! Disable frame-preemption on queue 0 and 1
(config-if)# no tsn frame-preemption queue 0-1
```

```
! Show configuration using 'show running-config'.
! Note that frame-preemption on port level is disabled by default
! and not shown unless the 'all-defaults' option is used.
# show running-config
[...]
tsn frame-preemption queue 0
tsn frame-preemption queue 1
```

4.2. Status

IMPORTANT : Add Web interface screen dumps and explanation here!

The equivalent CLI commands are:

```
! Show frame preemption port status using 'show tsn frame-preemption status'. # show
tsn frame-preemption status interface GigabitEthernet 1/1
interface GigabitEthernet 1/1
HoldAdvance                 : 1016 nanoseconds
ReleaseAdvance              : 1016 nanoseconds
PreemptionActive            : FALSE
HoldRequest                 : FALSE
StatusVerify                : disabled
LocPreemptSupported         : TRUE
LocPreemptEnabled           : TRUE
LocPreemptActive            : FALSE
LocAddFragSize              : 0 (64 octets)
```

5. Per-Stream Filtering and Policing

Per-Stream Filtering and Policing (PSFP), as defined in the IEEE 802.1Qci standard, provides filtering, policing and service class selection for a stream.

A stream is defined in the same way as in FRER but the stream IDs may be different.

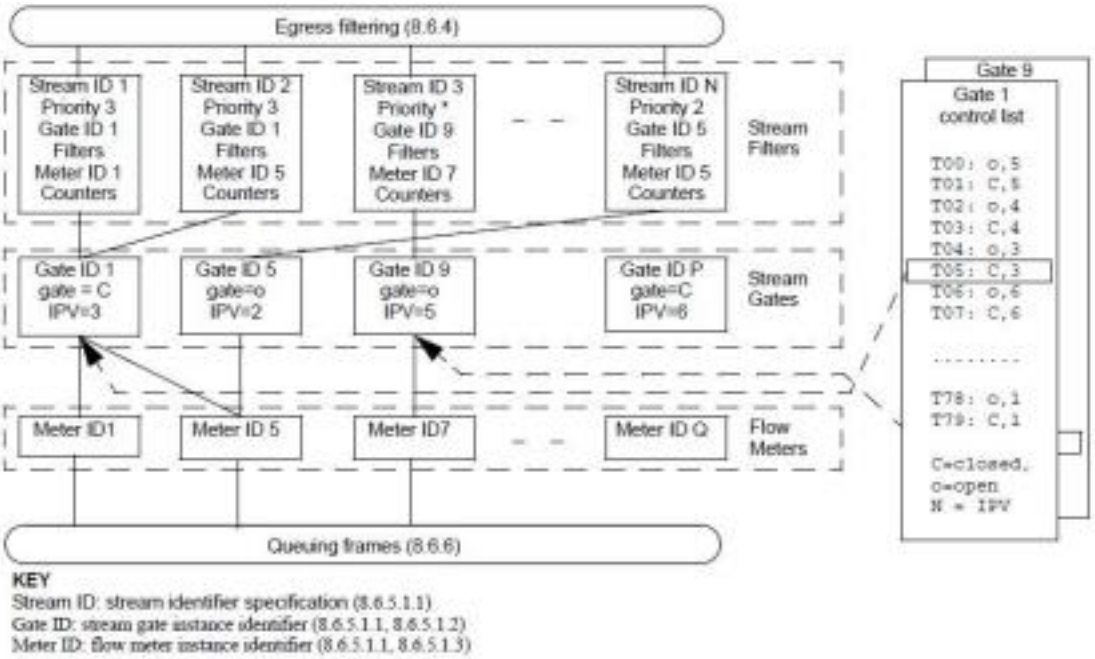

Figure 6. Per-Stream Filtering and Policing

The stream filter instance table is an ordered list of stream filters where smaller identifier values appear earliere in the ordered list. This is especially important to know if wild-card match parameters are used. If a frame matches more than one stream filter, the stream filter that is selected is the one that is located earliest in the list.

Each stream filter instance has s reference to a stream gate instance and optionally a reference to a flow meter instance.

The port parameter in the stream filter instance definition is not a part of the current standard, but is added in order to support different filters for each of the replicated streams in FRER.

When configuring stream filters for PSFP the hardware imposes some restrictions on the configuration:

- The maximum number of stream filter instances using the same stream ID is two.
- Both of these instances must be configured with the same priority (specific or wildcard).
- Both of these instances must be configured with different ports.and both ports must be specific (non wildcard).

The easiest way to avoid error messages due to these restrictions is to first configure priority and port and then stream ID.

IMPORTANT :
The Seville2 platform does not support simultaneous use of queue policers and PSFP flow meters. Queue policers are silently ignored when the first PSFP flow meter is created and re-enabled again when the last PSFP flow meter is deleted.

5.1. Configuration

IMPORTANT :
Add Web interface screen dumps and explanation here!

```
! Associate PSFP stream ID 10 with source MAC address 00:00:00:00:xx:xx, VLAN 1: configure terminal
stream 10
dmac any
smac 00-01-c1-00-00-00 / ff-ff-ff-00-00-00
outer-tag vid 1
exit
! To match double tagged traffic
stream 10
inner-tag vid 2
exit
! Adding masks and pcp, dei, s-tag
stream 10
inner-tag vid 1 / 4092 pcp 3 / 4 dei 0 s-tag
exit
! Select ipv4 traffic
stream 10
ipv4
exit
! Select specific ipv4 traffic
stream 10
ipv4 proto udp sip 10.208.23.00/255.255.255.0 dip 10.208.23.00/255.255.255.0 exit
! Select ipv6 traffic
stream 10
ipv6
exit
! Select specific ipv6 traffic
stream 10
ipv6 proto udp sip fe80::997b:9ec1:c74a:e826/64 dip fe80::201:c1ff:fe01:c550/64 exit
! Select snap traffic
stream 10
snap any
exit
! Select llc traffic
stream 10
llc dsap 1 ssap 2 control 3
exit
stream 10
etype 0x0600
exit
! Associate PSFP stream 10 with 10GigabitEthernet 1/7 and 10GigabitEthernet 1/8 interface 10GigabitEthernet 1/7-8
```

```
stream-id 10
end
```

```
!
Create Flow Meter Instance 30 with the following properties:
! Committed Information Rate is 400 kbps.
! Committed Burst Size is 8192 bytes.
! Block stream permanently if frames are exceeding the configured rate. !
configure terminal
tsn flow meter 30
cir 400
cbs 8192
mark-red-enable
exit
!
! Create Stream Gate Instance 20 with the following properties:
! Admin cycle time is 100 milliseconds.
! Gate is closed for 99 milliseconds and open for 1 millisecond where internal priority is modified
to 6.
! Block stream permanently if frames are received during the closed time-interval. !
tsn stream gate 20
cycle-time 100 ms
close-due-to-invalid-rx-enable
base-time 2020/02/15 12:00:00
control-list-length 2
control-list index 0 gate-state closed time-interval 99 ms
control-list index 1 gate-state open time-interval 1 ms ipv 6
enable
exit
!
! To change configuration of enabled gate use "config-change"
tsn stream gate 20
state open
config-change
exit
! Create Stream Filter Instance 1 with the following properties:
! Match on stream ID 10 and priority 4.
! Limit maximum SDU size to 512 bytes.
! Use Stream Gate Instance 20.
! Use Flow Meter Instance 30.
! Block stream permanently if frames greater than maximum SDU size are received. !
tsn stream filter 1
stream-id 10
priority 4
gate id 20
max-sdu 512
flow-meter id 30
block-due-to-oversize-enable
end
```

```
! See if stream filter has been blocked by oversize frames
# show tsn stream filter status
Filter Stream blocked due to
Id oversize frame
------ --------------------
1 false
! Reopen stream filter after is has been blocked by oversize frames:
# clear tsn stream filter 1 stream-blocked-due-to-oversize-frame
```

```
! See if stream gate has been closed either due to invalid rx or due to octets exceeded.
# show tsn stream gate 1 status
State: open
Cycle time: 40000000 ns
Cycle time extension: 0 ns
Base time: 1970/01/01 00:00:30
Config change time: 1970/01/01 00:00:30
Tick granularity: 1 ns
Current time: 1970/01/01 16:12:48 009500712
Config pending: false
Config change errors: 0
Priority value: 0
Closed-due-to-invalid-rx : true
Closed-due-to-octets-exceeded: true
! Reopen stream gate after is has been blocked.
clear tsn stream gate 1 closed-due-to-invalid-rx
clear tsn stream gate 1 gate-closed-due-to-octets-exceeded
```

5.2. Status

IMPORTANT Add Web interface screen dumps and explanation here! The

equivalent CLI commands are:

```
! Show Stream Filter Instance statistics (frame counters):
# show tsn stream filter statistics
Filter   Matching     Passing      Not passing   Passing      Not passing   Red
Id       frames       frames       frames        sdu count    sdu count     frames
-------- ------------ ------------ ------------- ------------ ------------- ----------
       1            0            0             0            0             0          0
#
```

```
! Clear Stream Filter Instance statistics (frame counters):
# clear tsn stream filter 1 statistics
#
```

```
! Show Stream Gate Instance status
# show tsn stream gate 20 status
State                    : closed
Cycle time               : 0 ns
Cycle time extension     : 0 ns
Base time                : 1970/01/01 00:00:00
Config change time       : 1970/01/01 00:00:00
Tick granularity         : 0 ns
Current time             : 1970/01/01 00:49:41 454345112
Config pending           : false
Config change errors     : 0
Priority value           : 0
#
```

6. Frame Replication and Elimination for Reliability

Frame Replication and Elimination for Reliability (FRER), as specified in the IEEE 802.1CB-2017 standard, provides increased reliability (reduced packet loss rates) for a stream.

This is achieved by

1. Sequence numbering and replicating packets in the talker (source) end system and/or in relay systems in the network. In the following, this is called generation.
2. Eliminating those replicates in the listener (destination) end system and/or in other relay systems. In the following, this is called recovery.

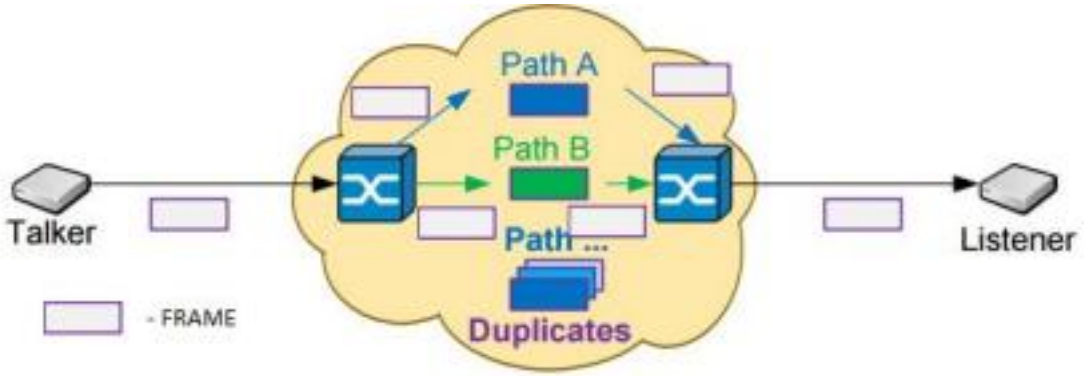

Figure 7. Frame Replication and Elimination for Reliability

FRER is supported by Microchip's SparX-5i and LAN9668 switch chips running IStaX software. There are differences between the two, which will be explained as we go along.

6.1. Overview

The basic idea is to send frames from the talker end through two or more paths in the network to increase the likelihood of the frames reaching their destination (listener end).

In order not to and send multiple copies of a given frame to the listener, the FRER generator augments every frame with a six byte so-called R-tag, as specified in clause 7.8 of 802.1CB-2017, before transmitting the frame on two or more ports.

The R-tag contains a two byte sequence number, which is incremented for every frame received by the generator. This sequence number is used by subsequent recovery functions in the FRER network to eliminate copies of a frame before presenting it to the Listener or to the next switch in the FRER network.

Several system components come into play when configuring FRER:

1. Streams are used for identifying and classifying frames ingressing a switch to a given FRER instance,
2. VLANs and/or MAC addresses are used to guide the frames the correct way in the FRER network,
3. MSTP is used to prevent loops, and
4. FRER configuration is used to connect the dots and configure a FRER instance as either a sequence generator or a sequence recovery instance.

It is often desirable to perform central management of the switches making up the FRER network, which in turn may result in layer 2 loops unless guarded by a loop prevention protocol such as spanning tree. In this guide, we provide examples that not only show how to configure FRER, but also how to configure the FRER switches for central, loop-free management. In the entire guide, we assume that the management VLAN is running untagged in VLAN 1.

In a FRER network, besides the R-tag, it is common practice to augment every frame egressing the generator with a VLAN tag containing a VLAN ID - the so-called FRER VLAN - which is used further down the FRER network to classify the frames as belonging to a given FRER instance. This FRER VLAN tag is supposed to be removed on the last switch in the FRER network before presenting the original frames to the listener end system.

Although not impossible to operate untagged, the presence of a VLAN tag containing a FRER VLAN makes network administration easier.

In the following, we assume that all FRER-protected frames carry an outer FRER VLAN tag. With this in place, there are two methods to guide the frames in the right direction:

1. VLAN flooding and
2. MAC address table forwarding

Let’s examine these two methods separately.

6.1.1. Method 1: VLAN Flooding

The idea is that frames egress the FRER generator with this outer FRER VLAN tag and get flooded to all ports on subsequent switches that are members of this VLAN.

All switches in the FRER network must disable learning in the FRER VLAN and make sure that frames only egress ports towards the final recovery function.

To avoid loops, the ingress ports along the way in the FRER network may **not** be members of the FRER VLAN.

With this method, the network administrator must assign a unique FRER VLAN to every FRER instance.

6.1.2. Method 2: MAC Address Table Forwarding

With this method, the listener's MAC address is pre-provisioned in every switch along the way, so that the <FRER VLAN, Listener MAC> tuple points out the egress ports.

With this method, one single FRER VLAN can accommodate all FRER instances, but it requires the listener's MAC address to be known beforehand.

Later on, we shall see examples of configuring switches with both methods. For now, let's look at FRER configuration parameters.

6.2. Configuration of parameters

A FRER instance represents a unidirectional function which can be either in generation or recovery mode. When a FRER instance is disabled, it has no impact on the frames passing through the switch.

Upon enabling a FRER instance, a sanity check on the provided parameters will be performed and if the combination of parameters lies within the acceptable range, FRER will start to operate.

When you configure a system which implements a full FRER network, you also have to consider configuration of one or more of the following components: VLAN, Stream, MAC address table.

The syntax for FRER global level CLI configuration command is:

```
tsn frer <inst>
```

Where :

```
inst FRER instance number
```

The syntax of FRER configuration level CLI Command is :

```
admin-state {enable | disable}
egress interface (<port_type> [<port_list>])
frer-vlan <vid>
ingress stream-id-list <stream_list>
mode {generation | recovery}
recovery algorithm {match | vector [history-length <history_len>]}
recovery individual
recovery latent-error-detection [difference <diff>] [period <period>] [paths <paths>] [reset-
period <reset_period>]
recovery reset-timeout <reset_timeout>
recovery take-no-sequence
recovery terminate
```

Where :

admin-state	Enable or disable a FRER instance
egress	Select egress ports that this FRER instance will hit
frer-vlan	Select the VLAN ID that ingress flows get classified to mode Choose this FRER instance's mode of operation (generation or recovery)
recovery	Set a recovery mode parameter
stream-id-list	Select the ingress streams that should map to this FRER instance. Only one stream ID can be specified in generator mode
generation	FRER instance generates R-tags
recovery	FRER instance operates in recovery mode
algorithm	Choose which recovery algorithm to run
match	Run match recovery algorithm (802.1CB, clause 7.4.3.5) vector Run vector recovery algorithm (802.1CB, clause 7.4.3.4) history-
length	Select the vector algorithm's history length
individual	When individual recovery is enabled, each member stream runs the recovery function before presenting it to the compound recovery function
latent-error-detection	Enable recovery's latent error detection function
difference	Set the maximum allowed difference between discarded packets and passed packets before triggering the detection of a latent error
period	Set the period with which the latent error test function runs
reset-period	Set the period between running the latent error reset function
reset-timeout	Configure recovery function's reset timeout
take-no-sequence	Accept all frames whether they are R-tagged or not. terminate This option allows to strip an R-tag from a frame before presenting it on egress
no egress interface	Unset the egress interfaces of this FRER instance
no frer-vlan	Default the VLAN ID that ingress frames get classified to
no mode	Default this FRER instance's mode of operation
no ingress stream-id-list	Clear the list of ingress stream IDs
no recovery algorithm	Default the recovery algorithm (vector)
no recovery individual	When individual recovery is disabled, all member streams are sent directly to the compound recovery function without performing individual recovery first
no recovery latent-error-detection	Disable recovery's latent error detection function
no recovery reset-timeout	Default the recovery function's reset timeout no recovery
take-no-sequence	The recovery function discards frames that are not R-tagged (default)
no recovery terminate	Do not strip an R-tag from a frame before presenting it on egress

6.3. Simple Configuration Example

This example shows a Talker and a Listener connected by two switches, A and B, with two paths in-between. These two paths are protected by FRER, which implements splitting and recovery such that if one of the paths between the two switches fails, traffic will continue to flow uninterrupted between Talker and Listener.

The System Switch allows for managing both FRER switches. It must run the Multiple Spanning Tree Protocol (MSTP) at least in the management VLAN. Configuration of this switch is not otherwise part of this guide.

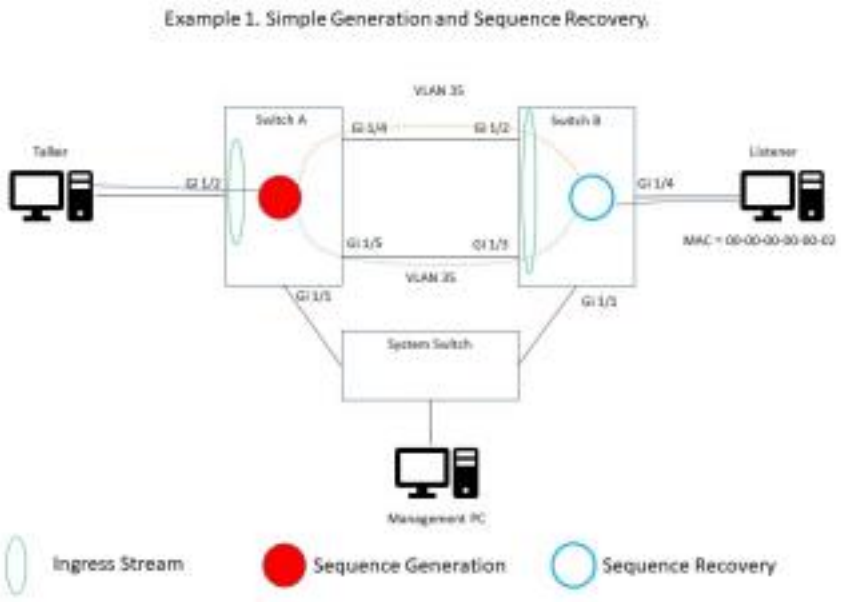

Figure 8. FRER Generation and Sequence Recovery.

Switch A takes care of classifiying the frames from the Talker to a FRER instance and splitting the stream into two paths. Switch B takes care of merging the two paths and presenting the original frames to the Listener.

Switch A's frame classification happens through the use of so-called streams. It is outside the scope of this configuration guide to list all the parameters that make up a stream identification, but among the parameters are both L2, L3, and L4 properties.

In this example, we have chosen to map all frames ingressing Gi 1/2 with a destination MAC address (DMAC) matching that of the Listener to the FRER instance. The Listener's DMAC is 00-00-00-00-00-02.

Switch A's FRER instance is configured to be in generation mode, and generates two identical copies of the frame on its configured egress ports.

When the frames exit Gi 1/4 and Gi 1/5 of Switch A, they include a 6-byte R-tag containing a sequence number.

On top of that, it is common practice to also push an outer VLAN tag containing a VLAN ID used to identify the stream (or possibly multiple streams) on subsequent switches in the FRER network. The VLAN ID of this VLAN tag holds the so-called FRER VLAN.

It is typically the last switch in the FRER network that takes care of popping both the R-tag and the FRER VLAN tag from the frame before presenting the original frame to the Listener.

If the frame itself contained a VLAN tag on ingress to Switch A, that VLAN tag will become the inner tag upon egress of Switch A. A frame can therefore have between one and three tags on egress of Switch A:

1. Possibly 4 bytes outer VLAN tag with VLAN ID set to the FRER VLAN. This tag is only pushed if the egress port is configured to tag the FRER VLAN. The TPID is determined by the egress port's tag type (C- or S-),
2. 6 bytes R-tag containing the generated sequence number,
3. Possibly 4 bytes inner VLAN tag containing the original ingress frame's VLAN tag. Only available if the original frame contained a VLAN tag.

In both the VLAN flooding example and the MAC Address Table Forwarding examples, switch B will be configured to look for frames arriving on Gi 1/2 and Gi 1/3 with an outer VLAN tag containing a VLAN ID corresponding to the FRER VLAN.

In the MAC Address Table Forwarding example, this matching will also match on the frames' DMAC.

The goal is to let the Listener receive the same frames as the Talker transmitted while providing reliability, so that either of the two links between Switch A and Switch B can go down without the two parties noticing.

6.3.1. Example 1-1: VLAN Flooding

This first example will make use of VLAN flooding.

Switch A: Generation using VLAN Flooding

Command	Purpose
# configure terminal	Enter configuration mode.
(config)# prompt Switch-A	Give this switch a name
Switch-A(config)# vlan 1 Switch-A(config-vlan)# interface GigabitEthernet 1/1 Switch-A(config-if)# switchport access vlan 1 Switch-A(config-if)# switchport mode access	Make sure the management VLAN is defined and that the management port is a member of this VLAN. These are all default settings and will not be shown in the running-config
Switch-A(config)# interface * Switch-A(config-if)# spanning-tree Switch-A(config-if)# exit	Make sure all ports (could be just the ones we show in the figure) run MSTP. This is the default and will not be shown in the running-config, but is include for clarity.

Switch-A(config)# spanning-tree mst te vlan 35	By default, all 4K VLANs are included in the CIST MSTI, so if MSTP finds a loop, it will block all 4K VLANs. We know that there is a loop on the FRER VLAN between Switch A and Switch B, so we would like to exclude this VLAN from blocking. We do that by putting it into a special MSTI called te, which stands for Traffic Engineering. VLANs in the TE MSTI are always forwarding.
Switch-A(config)# stream 17 Switch-A(config-stream)# dmac 00-00-00-00-00-02 Switch-A(config-stream)# exit	Create a stream for classifying frames from the Talker to a FRER instance. The stream ID (17) is arbitrarily chosen and must be unique within this switch. It must not be used for other purposes than FRER (e.g. PSFP). Let the stream match frames with DMAC equal to that of the listener and exit the stream configuration mode.
Switch-A(config)# interface GigabitEthernet 1/2 Switch-A(config-if)# stream-id 17	Assign Stream ID 17 to Gi 1/2, the ingress port receiving frames from the talker.
Switch-A(config-if)# no switchport hybrid ingress-filtering Switch-A(config-if)# switchport hybrid allowed vlan none Switch-A(config-if)# switchport mode hybrid	**This is a very important step to understand.** Later on in the configuration, we will make sure that all frames matching the newly configured stream get classified to the FRER VLAN. If Gi 1/2 is not member of the FRER VLAN, then frames would get discarded already on ingress with the default VLAN configuration (switchport mode access). We don’ t want the ingress port to be member of the FRER VLAN, because we don’ t want frames received on other ports that happen to get classified to the FRER VLAN to egress our ingress port. Therefore, we disable ingress filtering on Gi 1/2, which will allow it to receive and forward frames classified to any VLAN, including the FRER VLAN. In order to be able to disable ingress filtering, the port must be put into hybrid mode. Notice that ingress filtering is disabled by default in hybrid mode, so show running-config will not print this line. We leave the port to not being member of any VLAN, but it could be configured to being member of any VLAN
Switch-A(config-if)# interface GigabitEthernet 1/4,5 Switch-A(config-if)# switchport hybrid allowed vlan 1,35 Switch-A(config-if)# switchport hybrid native vlan 1 Switch-A(config-if)# switchport hybrid port-type c-port Switch-A(config-if)# switchport mode hybrid Switch-A(config-if)# exit	Configure our two egress ports, Gi 1/4and Gi 1/5. We can do this in one tempo, because we - in this example - configure them identically. This might not be the case in other scenarios. We make the ports members of VLAN 1, the management VLAN and VLAN 35, the FRER VLAN. The ports could be members of more VLANs, but they must be members of at least the FRER VLAN. In order to force a C-tag on R-tagged frames, we keep the native VLAN (PVID) at 1 and make the port a C-port before switching to the newly configured hybrid mode. Notice that neither the native vlan nor the port-type lines will appear in show running-config ,

	because these are the defaults in hybrid mode. They are included here for clarity.Finally exit interface configuration mode.
Switch-A(config)# no mac addresstable learning vlan 35	As said in the beginning of this example, we make use of VLAN flooding to make sure that frames classified to the FRER VLAN get transmitted on all ports that are members of the FRER VLAN. In order to make sure that this happens, we must disable learning and enable flooding in the FRER VLAN. If learning was not disabled, a frame ingressing Gi 1/2 to a particular destination MAC address (DMAC), which is present in the MAC address table on a particular port in the FRER VLAN, would only be sent on that port and not on all ports in the FRER VLAN. Flooding is enabled in all VLANs by default, so nothing required there.
Switch-A(config)# tsn frer 10 Switch-A(config-frer)# mode generation Switch-A(config-frer)# ingress stream-id-list 17 Switch-A(config-frer)# frer-vlan 35 Switch-A(config-frer)# egress interface GigabitEthernet 1/4,5 Switch-A(config-frer)# admin-state enable	Create a FRER instance with ID 10 (arbitrarily chosen, but must be unique within this switch) and put it in generation mode. Let it work on the stream we just created and have frames classified to our FRER VLAN (35). When using the VLAN flooding model, it is important that the FRER VLAN is not being used for any other purposes than FRER on this particular FRER instance; different FRER instances must use different FRER VLANs. If you want to have only a single FRER VLAN for multiple FRER instances, you must use the MAC table forwarding model, as explained later. Despite its name, the stream-id-list can only have one single member in generation mode. Also specify the egress ports, Gi 1/4 and Gi 1/5. Replicated frames egressing these ports will have the same R-tag pushed. Finally enable this instance.
Switch-A(config-frer)# end Switch-A#	We are now done with configuration of Switch A.

NOTE :
If other ports are members of the FRER VLAN, frames will also egress those. On LAN9668, the frames will egress with an R-tag and on SparX-5i, they will egress without an R-tag.

To summarize , here's a list of commands, where default and irrelevant commands are omitted.

```
prompt Switch-A
vlan 1
no mac address-table learning vlan 35
spanning-tree mst te vlan 35
stream 17
dmac 00-00-00-00-00-02
interface GigabitEthernet 1/2
switchport hybrid allowed vlan none
switchport mode hybrid
stream-id 17
interface GigabitEthernet 1/4,5
switchport hybrid allowed vlan 1,35
switchport mode hybrid
tsn frer 10
mode generation
ingress stream-id-list 17
frer-vlan 35
egress interface GigabitEthernet 1/4,5
admin-state enable
```

The vigilant reader will notice that the stream can be instantiated on more than one ingress port. This is a perfectly valid scenario, allowing streams from multiple ports towards DMAC 00-00-00-00-00-02 to be FRER protected by the same FRER instance.

Switch B: Recovery using VLAN Flooding

In this example, Switch B terminates the FRER flows and will transmit it on the designated egress port(s) without R-tags.
We define Switch B's FRER VLAN to be 100. This is in order to emphasize that the terminating switch's FRER VLAN doesn't need to be the same as on previous switches inside the FRER network, as long as it's unique on the terminating switch. It could, however, just as well be the same as on the previous switches in the FRER network.

Regarding management, we will skip describing the steps taken to making Gi 1/1 an access port in VLAN 1 and how to enable MSTP on all ports, because this is default and the same thing as on **Switch-A** . We will, however, emphasize how to prevent MSTP from blocking the FRER VLANs.

Command	Purpose
# configure terminal	Enter configuration mode.
(config)# prompt Switch-B	Give this switch a name.

Switch-B(config)# spanning-tree mst te vlan 100	As on Switch-A, we need to add VLANs that are not supposed to be blocked to the TE MSTI. In this case, we must have the new FRER VLAN ID (VLAN 100) in the TE MSTI, because we need this VLAN ID to always be forwarding.
Switch-B(config-stream)# stream 20 Switch-B(config-stream)# outer-tag vid 35 Switch-B(config-stream)# exit	Create a stream (arbitrarily chosen stream ID) that matches the R-tagged frames coming from Switch A. Since - in the VLAN flooding model – it is "guaranteed" that only one stream uses this VLAN ID, we don't need to match any other properties of the frames.
Switch-B(config)# interface GigabitEthernet 1/2,3 Switch-B(config-if)# stream-id 20 Switch-B(config-if)# switchport hybrid allowed vlan 1 Switch-B(config-if)# switchport mode hybrid	Instantiate the stream on both ingress ports and put them in hybrid mode in order for it to forward all VLANs (because ingress filtering - by default - is disabled in hybrid mode). In this particular example, we have chosen the ports only to be a member of the Management VLAN (1), so that management frames can flow freely between Switch-A and Switch-B, unless blocked by MSTP. It is up to the network administrator to make the ingress ports members of therequired VLANs.
Switch-B(config-if)# interface GigabitEthernet 1/4 Switch-B(config-if)# switchport access vlan 100 Switch-B(config-if)# exit Switch-B(config)# vlan 100 Switch-B(config-vlan)# exit	In this example, we configure the egress port as a simple access port in Switch B's FRER VLAN (100). Access ports are only members of their native VLAN (PVID), and the frames get untagged on transmission. Access ports require their PVID to be explicitly created (or they won't be members of any VLANs), so we also create VLAN 100.
Switch-B(config)# no mac addresstable learning vlan 100	This step is not so important when we only have one egress port. However, if we were not recovering and terminating a FRER flow with one egress port, but recovering and resplitting the flows into several egress ports, this is an important step (see explanation in Switch A's configuration).
Switch-B(config)# tsn frer 20 Switch-B(config-frer)# mode recovery Switch-B(config-frer)# ingress stream-id-list 20 Switch-B(config-frer)# frer-vlan 100 Switch-B(config-frer)# egress interface GigabitEthernet 1/4 Switch-B(config-frer)# recovery terminate Switch-B(config-frer)# admin-state enable	Create a FRER instance (ID is arbitrarily chosen, but must be unique within this switch) and put it in recovery mode. Let it work on the stream we just created and have frames classified to VLAN 100. Specify the egress interface where the recovered frames are to be transmitted. The recovery terminate line configures the switch to remove the Rtag from frames egressing the specified egress interfaces.
Switch-B(config-frer)# end Switch-B#	We are now done with configuration of Switch B.

NOTE :
If another port is member of VLAN 100 and that port is not specified in the FRER instance's egress interface list, then frames transmitted on that port will not be recovered, that is, multiple copies will be transmitted - one per member flow. Furthermore, on SparX-5i, these frames will still contain the R-tag, whereas on LAN9668, the R-tag will be stripped.

To summarize, here's a list of commands, where default and irrelevant commands are omitted.

```
prompt Switch-B
vlan 1,100
no mac address-table learning vlan 100
spanning-tree mst te vlan 100
stream 20
outer-tag vid 35
interface GigabitEthernet 1/2,3
switchport hybrid allowed vlan 1
switchport mode hybrid
stream-id 20
interface GigabitEthernet 1/4
switchport access vlan 100
tsn frer 20
mode recovery
ingress stream-id-list 20
frer-vlan 100
egress interface GigabitEthernet 1/4
recovery terminate
admin-state enable
```

6.3.2. Example 1-2. MAC Table Forwarding

The previous example uses VLAN flooding to get frames from ingress to egress on a particular switch. An alternative is to pre-provision the Listener's MAC address in Switch A and Switch B's MAC address tables. This requires somewhat more configuration than the VLAN flooding method, but is at the same time safer, because frames only egress the ports pre-provisioned with the Listener's MAC address, and not all ports that are members of the FRER VLAN.
Another advantage of using the MAC table forwarding approach is that all FRER instances may use the same FRER VLAN.

A disadvantage of using the same FRER VLAN for all FRER instances is that broadcast and multicast frames can't be controlled per FRER instance. If this is required, each FRER instance must use a separate FRER VLAN. With this in place, either VLAN flooding or MAC table forwarding can be used.

The following configuration utilizes the same IDs and setup as the previous example. Focus will be on the differences to the previous example.

In the example, we assume that the Listener's MAC address is 00-00-00-00-00-02. Switch A: Generation using MAC Table Forwarding

Command	Purpose
# configure terminal	Enter configuration mode.

(config)# prompt Switch-A Switch-A(config)# spanning-tree mst te vlan 35 Switch-A(config)# stream 17 Switch-A(config-stream)# dmac 00-00-00-00-00-02 Switch-A(config-stream)# exit Switch-A(config)# interface GigabitEthernet 1/2 Switch-A(config-if)# switchport hybrid allowed vlan none Switch-A(config-if)# switchport mode hybrid Switch-A(config-if)# stream-id 17 Switch-A(config-if)# interface GigabitEthernet 1/4,5 Switch-A(config-if)# switchport hybrid allowed vlan 1,35 Switch-A(config-if)# switchport mode hybrid Switch-A(config-if)# exit	This is the same and with the same explanations as in Example 1-1.
Switch-A(config)# mac address-table static 00-00-00-00-00-02 vlan 35 interface GigabitEthernet 1/4,5 Switch-A(config)# no mac addresstable learning vlan 35	When using MAC table forwarding, we tell the switch that the Listener can be reached on two egress ports (Gi 1/4 and Gi 1/5) on the FRER VLAN. This ensures that frames destined to the Listener only egresses those two ports, no matter which other ports are members of the FRER VLAN, making it possible for several FRER instances to use the same FRER VLAN - possibly on other egress ports. We also - as in the previous example - disable learning in the FRER VLAN – for safety.
Switch-A(config)# vlan 35 Switch-A(config-vlan)# no flooding Switch-A(config-vlan)# exit	This is the second half of using MAC table forwarding. Disabling flooding in the FRER VLAN ensures that incoming frames that get classified to the FRER VLAN for one or the other reason, get dropped on ingress **unless** the frame’ s DMAC is in the MAC address table.
Switch-A(config)# tsn frer 10 Switch-A(config-frer)# mode generation Switch-A(config-frer)# ingress stream-id-list 17 Switch-A(configfrer)# frer-vlan 35 Switch-A(config-frer)# egress interface GigabitEthernet 1/4,5 Switch-A(config-frer)# admin-state enable Switch-A(config-frer)# end Switch-A#	This is the same and with the same explanations as in Example 1-1.

To summarize, here's a list of commands, where the default commands are omitted.

```
prompt Switch-A
vlan 1
vlan 35
no flooding
no mac address-table learning vlan 35
mac address-table static 00-00-00-00-00-02 vlan 35 interface GigabitEthernet 1/4,5 spanning-tree mst te vlan 35
stream 17
dmac 00-00-00-00-00-02
interface GigabitEthernet 1/2
switchport hybrid allowed vlan none
switchport mode hybrid
stream-id 17
interface GigabitEthernet 1/4,5
switchport hybrid allowed vlan 1,35
switchport mode hybrid
tsn frer 10
mode generation
ingress stream-id-list 17
frer-vlan 35
egress interface GigabitEthernet 1/4,5
admin-state enable
```

Switch B: Recovery using MAC Table Forwarding

Command	**Purpose**
Switch-B# configure terminal	Enter configuration mode.
(config)# prompt Switch-B Switch-B(config)# spanning-tree mst te vlan 100 Switch-B(config-stream)# stream 20 Switch-B(config-stream)# outer-tag vid 35 Switch-B(config-stream)# dmac 00-00-00-00-00-02 Switch-B(config-stream)# interface GigabitEthernet 1/2,3 Switch-B(config-if)# switchport hybrid allowed vlan 1 Switch-B(config-if)# switchport mode hybrid Switch-B(config-if)# stream-id 20 Switch-B(config-if)# interface GigabitEthernet 1/4 Switch-B(config-if)# switchport access vlan 100 Switch-B(config-if)# vlan 100	This is the same and with the same explanations as in Example 1-1, except for one thing shown in the first NOTE below.
Switch-B(config-vlan)# no flooding Switch-B(config)# no mac addresstable learning vlan 100 Switch-B(config)# mac address-table static 00-00-00-00-00-02 vlan 100	Here, we make sure that frames that get classified to VLAN 100 are discarded unless they are present in the MAC table. We also ensure that the Listener' s MAC address is in the MAC table on the correct port.

interface GigabitEthernet 1/4	
Switch-B(config)# tsn frer 20 Switch-B(config-frer)# mode recovery Switch-B(config-frer)# ingress stream-id-list 20 Switch-B(config-frer)# frer-vlan 100 Switch-B(config-frer)# egress interface GigabitEthernet 1/4 Switch-B(config-frer)# recovery terminate Switch-B(config-frer)# admin-state enable Switch-B(config-frer)# end Switch-B#	This is the same and with the same explanations as in Example 1-1.

NOTE :
This is very important: LAN9668 only supports stream matching beyond two tags, where the R-tag is one of them. So if the frames contain both a FRER VLAN tag, an R-tag and an inner VLAN tag, it is only possible to match on DMAC and SMAC, and not other, higher layer frame properties. If the frames ingressing the generator end are untagged, it is still possible to match on higher layer frame features in recovering switches.
SparX-5i, on the other hand, supports stream matching on three tags, which makes it possible to match on frame features beyond L2, also on recovering switches in the FRER network.

NOTE :
If the DMAC is added to the MAC table on another egress port but it' s not specified in the egress interface … of the FRER instance, frames will egress that other port unrecovered and therefore in multiple copies. On SparX-5i, they will in addition still contain the R-tag,
whereas on LAN9668, the R-tag will be stripped.

To summarize, here' s a list of commands, where the default commands are omitted.

```
prompt Switch-B
vlan 1
vlan 100
no flooding
no mac address-table learning vlan 100
mac address-table static 00-00-00-00-00-02 vlan 100 interface GigabitEthernet 1/4 spanning-
tree mst te vlan 100
stream 20
dmac 00-00-00-00-00-02
outer-tag vid 35
interface GigabitEthernet 1/2,3
switchport hybrid allowed vlan 1
switchport mode hybrid
stream-id 20
interface GigabitEthernet 1/4
switchport access vlan 100
tsn frer 20
mode recovery
ingress stream-id-list 20
frer-vlan 100
egress interface GigabitEthernet 1/4
recovery terminate
admin-state enable
```

6.4. Individual Recovery

The following figure shows an example of how to configure Switch B to perform individual recovery.

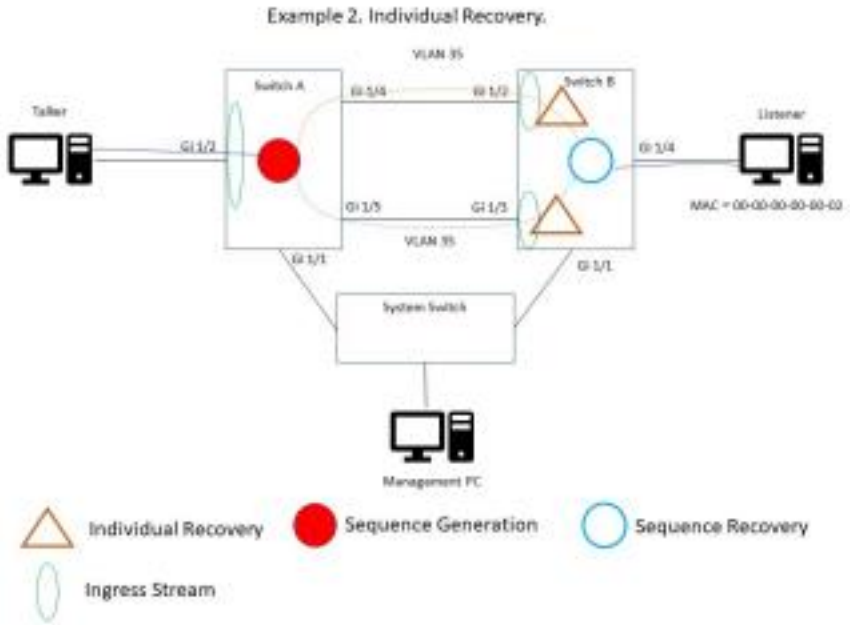

Figure 9. FRER Generation and Individual Recovery.

The first thing to note compared to the figure of Example 1 is that the two ingress ports of Switch B have its single ingress stream split in two. This allows the switch to perform individual recovery on each of the two member streams.

Individual recovery means that a member stream undergoes recovery before it reaches the compound recovery function. The compound recovery function sits on each and every egress port in the FRER instance, and is what we have used until now.

So what is the purpose of individual recovery?
The one and only thing that individual recovery can do that compound recovery can't is to filter out member streams that keep presenting the same R-tag sequence number because of a defect transmitter. It goes like this:

Suppose the transmitter of member stream 1 is working perfectly. It will send out frames with an increasing sequence number and wrap back to 0 after 65536 frames. Suppose the transmitter of member stream 2 is sending out the same frame with the same sequence number, X, over and over again.
If we only had a compound recovery function, that function would at times be presented with frames with sequence number X from stream 1 and sequence number X from stream 2, and the first of these two frames would be sent to the egress port. So - depending on timing - sometimes the frame with sequence number X would come from stream 1 and sometimes it would come from the erroneous stream 2. The effect of enabling individual recovery is to have the individual recovery function for stream 2 filter out all identically numbered frames before they are presented to the compound recovery function.

Let me say it straight away: This is a very unlikely situation, and I think most network administrators will not need individual recovery.

Moreover, individual recovery is very expensive in terms of hardware resources: Every ingress stream needs an individual recovery function per egress port. So if a FRER instance defines 8 ingress streams and 8 egress ports, the switch needs 64 individual recovery instances - just for this one FRER instance.

6.4.1. Example 2: Configuring Individual Recovery

Anyhow, here is how you configure individual recovery on Switch B using the same methodology as was used in Example 1-1. Only changed commands are shown.

Command	Purpose
Switch-B(config)# stream 20 Switch-B(config-stream)# outer-tag vid 35 Switch-B(config-stream)# stream 21 Switch-B(config-stream)# outer-tag vid 35	We need to create two separate streams with the same properties.
Switch-B(config-stream)# interface GigabitEthernet 1/2 Switch-B(config-if)# stream-id 20 Switch-B(config-if)# interface GigabitEthernet 1/3 Switch-B(config-if)# stream-id 21	Instantiate stream 20 on Gi 1/2 and instantiate stream 21 on Gi 1/3.
Switch-B(config-if)# tsn frer 20 Switch-B(config-frer)# ingress stream-id-list 20,21 Switch-B(config-frer)# recovery individual Switch-B(config-frer)# end Switch-B#	Map both ingress streams to this FRER instance and enable indiviudal recovery.

This boils down to this configuration :

```
prompt Switch-B
vlan 1,100
no mac address-table learning vlan 100
spanning-tree mst te vlan 100
stream 20
outer-tag vid 35
stream 21
outer-tag vid 35
interface GigabitEthernet 1/2
switchport hybrid allowed vlan 1
switchport mode hybrid
stream-id 20
interface GigabitEthernet 1/3
switchport hybrid allowed vlan 1
switchport mode hybrid
stream-id 21
interface GigabitEthernet 1/4
switchport access vlan 100
tsn frer 20
mode recovery
ingress stream-id-list 20,21
frer-vlan 100
egress interface GigabitEthernet 1/4
recovery individual
recovery terminate
admin-state enable
```

6.5. Recovery Algorithm

IEEE 802.1CB-2017 requires implementations to provide two different recovery function algorithms, match and vector .

match is the simplest algorithm: It basically says: Discard all packets with a sequence number equal to the last sequence number seen. Accept all others. The algorithm also comes with a reset timer that - when it expires - causes the algorithm to accept any sequence number - even the same as the previous. The reset timer is restarted every time a packet is accepted.

The match algorithm counts the number of times the reset timer has expired and the number of passed, discarded, and out-of-order packets. Out-of-order happens when the sequence number of a given packet is not one higher than the previous (and the timer hasn't expired).

vector is somewhat more complicated. When a packet with a given sequence number arrives, it must be within the previous accepted packet's sequence number +/- a configurable history length, or it will be discarded. If the packet is already seen

(within the history length window), it is also discarded.
Also this algorithm comes with a reset timer that - upon expiration - causes the algorithm to accept any sequence number next time a packet arrives. The reset timer is restarted every time a packet is accepted.

The vector algorithm counts the number of times the reset timer has expired and the number of passed, discarded, out-of-order, and so-called rogue packets. Out-of order happens when the sequence number of a given packet is "older" than a previous packet's (taking wrap-around into account), and the packet hasn't been accepted before. Out-of-order packets are accepted.
Rogue packets are packets with a sequence number beyond the history length window. Rogue packets are also counted as discarded.
Furthermore, the vector algorithm counts lost packets, that is, the number of unreceived sequence numbers when the history window moves.

Both algorithms also count the number of packets arriving without an R-tag. This is done with the tagless counter. By default, such packets will be discarded. A per-FRER instance parameter recovery take-no-sequence , however, allows such frames to pass through.
Notice: The 802.1CB standard utilizes the frerSeqRcvyTakeNoSequence only in the vector algorithm, but the switch chips that the present guide is meant for also utilize it in the match algorithm.
Notice: This feature should only be used on terminating switches, because such tagless packets will be R-tagged (with sequence number 0) on their way out on non terminating switches.

The selected algorithm on a given FRER instance will be used in both compound and individual recovery functions.

Default is the vector algorithm with a history-length of 2 and a reset timeout of 1000 milliseconds and tagless packets are discarded.

6.5.1. Recovery Algorithm Examples

Command	**Purpose**
Switch-B# configure terminal	Enter configuration mode.
Switch-B(config)# tsn frer 20 Switch-B(config-frer)# recovery	This selects the match algorithm on this FRER instance. It times out after 4000

algorithm match Switch-B(config-frer)# recovery reset-timeout 4000	milliseconds.

Command	**Purpose**
Switch-B# configure terminal	Enter configuration mode.
Switch-B(config)# tsn frer 20 Switch-B(config-frer)# recovery algorithm vector history-length 32 Switch-B(config-frer)# recovery reset-timeout 500 Switch-B(config-frer)# recovery take-no-sequence	This selects the vector algorithm with a history length of 32 on this FRER instance. Here, we have configured the algorithm to reset after 500 milliseconds, and we allow it to pass frames arriving without an R-tag (default is *not* to allow this).

6.6. Latent Error Detection

The purpose of latent error detection is to raise a flag if the number of discarded packets is "relatively few" compared to the number of passed packets.

The algorithm relies on four user inputs:

1. period : The number of milliseconds between invoking the test function . Default is 2000.
2. reset-period : The number of milliseconds between invoking the reset function . Default is 30000.
3. paths : The number of member streams expected to ingress this FRER instance. Default is 2.
4. difference : The number of packets "allowed" to be in difference without raising the flag. Default is 100.

The reset function algorithm is as follows: Every reset-period milliseconds, read number of passed and discarded packet counters, and set a per-FRER instance variable, CurDiff , as follows:

```
CurDiff = passed_packets * (paths - 1) - discarded_packets;
```

The test function algorithm is as follows: Every timeout milliseconds, read the discarded and passed packet counters, and perform the following:

```
diff = Abs(CurDiff - (passed_packets * (paths - 1) - discarded_packets));
if (diff > difference) {
raise_flag();
}
```

Basically, it says: If you expect N member streams to ingress this FRER instance, N-1 of these member streams are expected to be discarded, and only one is expected to pass.

To allow for some slack due to random packet losses and the fact that counters are not neccessarily read simultaneously, set the difference to account for that.

The reset function makes sure that CurDiff is updated to avoid that occassional packet losses don’t accumulate forever.

Once the flag is raised, it will become sticky in the sense that it requires a network administrator to clear it. The value of the sticky flag can be read with show tsn frer <instance> and it may give rise to a JSON notification and an SNMP trap. To clear it, use the tsn frer <instance> reset latent-error CLI command.

6.6.1. Latent Error Detection Example

This example shows how to configure latent error detection on switch B from example 1-1

Command	**Purpose**
Switch-B# configure terminal	Enter configuration mode.
Switch-B(config)# tsn frer 20	Enter configuration of our FRER instance.
Switch-B(config-frer)# recovery latent-error-detection or Switch-B(config-frer)# recovery latent-error-detection paths 2 difference 1000 period 1000 resetperiod 10000 or Switch-B(config-frer)# no recovery latent-error-detection	First example shows how to enable latent error detection with default parameters. Second example shows how to enable it with custom parameters. Last example shows how to disable latent error detection.
Switch-B(config-frer)# end	End configuration mode.

The following shows how to see the current latent error detection status and how to clear the raised flag.

```
Switch-B# show tsn frer 20
Inst  Operational   State     Mode         Latent Errors
----  ----------------------- ----------   ----------------
20    Active                  Recovery     Yes
Switch-B# tsn frer 20 reset latent-error
Switch-B# show tsn frer 20
Inst  Operational   State     Mode         Latent Errors
----  ----------------------- ----------   ----------------
20    Active                  Recovery     No
```

8.7. Advanced Configuration Example

Before diving into status and statistics, let's have a look at an advanced configuration example.
The following figure is very much alike Figure C-4 from IEEE-802.1CB-2017. 2022-03-31 Confidential

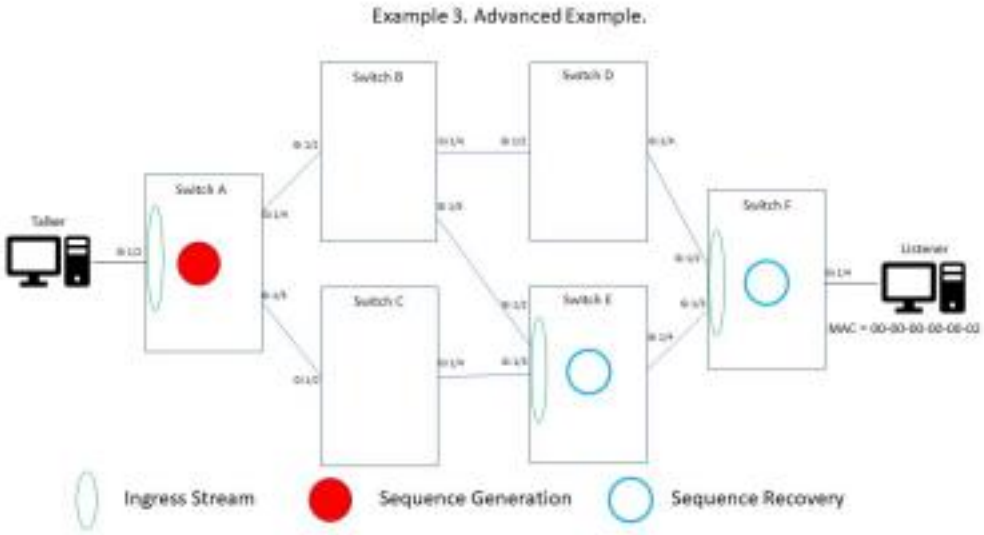

Figure 10. Advanced Configuration Example.

Here, the Talker is transmitting a stream, but has no FRER functions. Switch A transforms the ingress stream into a compound stream by sequencing the packets and splitting them into two streams.
These two streams go into switch B and C, where B splits one of the member streams further and D simply relays them from ingress to egress.
The - now - three member streams go into switch D and E. Switch D makes a simple relay of one of the member streams and switch E performs a recovery on the two others before presenting one recovered stream on egress.
The - now - two member streams then go into switch F, which performs recovery before egressing the restored stream to the Listener, which also doesn't have any FRER functions.

Switch A, E, and F are required to have FRER functions, whereas switch B, C, and D can be simple layer 2 (yet managed) forwarding switches.

Let's see how this plays out when using the MAC Table Forwarding method for ensuring that frames come from ingress to egress on any particular switch.

We assume that the Listener is having MAC address 00-00-00-00-00-02. For simplicity, configuration of the

management VLAN is left out.

6.7.1. Switch A

The configuration is identical to that of Switch A from Example 1-2 and boils down to: 2022-03-31

```
vlan 35
no flooding
no mac address-table learning vlan 35
mac address-table static 00-00-00-00-00-02 vlan 35 interface GigabitEthernet 1/4,5 spanning-
tree mst te vlan 35
stream 17
dmac 00-00-00-00-00-02
interface GigabitEthernet 1/2
switchport hybrid allowed vlan none
switchport mode hybrid
stream-id 17
interface GigabitEthernet 1/4,5
switchport hybrid allowed vlan 1,35
switchport mode hybrid
tsn frer 10
mode generation
ingress stream-id-list 17
frer-vlan 35
egress interface GigabitEthernet 1/4,5
admin-state enable
```

6.7.2. Switch B

A simple split of one frame into two can be done in many ways. Here, let's provision the egress ports with the Listener's MAC address.

Since the ingress stream is C-tagged with the FRER VLAN and since we carefully manage that VLAN, we can simply let both ingress and egress ports carry all VLANs (trunk mode) and disable flooding and learning in the FRER VLAN (VLAN 35) while using the MAC address table to forward.

We must also prevent MSTP from blocking VLAN 35.

This gives the following configuration.

```
vlan 35
no flooding
no mac address-table learning vlan 35
mac address-table static 00-00-00-00-00-02 vlan 35 interface GigabitEthernet 1/4,5 spanning-
tree mst te vlan 35
interface GigabitEthernet 1/2,4,5
switchport mode trunk
```

6.7.3. Switch C and Switch D

These switches are also FRER-unaware and we use the same type of configuration as for Switch B, but with only one egress port.

```
vlan 35
no flooding
no mac address-table learning vlan 35
mac address-table static 00-00-00-00-00-02 vlan 35 interface GigabitEthernet 1/4 spanning-tree
mst te vlan 35
interface GigabitEthernet 1/2,4
switchport mode trunk
```

6.7.4. Switch E

This switch performs recovery as does Switch B from Example 1-2. The only difference is that frames must be tagged with the FRER VLAN on egress and we are not terminating the recovery, so we don't strip the R-tag (no recovery terminate).

```
vlan 35
no flooding
no mac address-table learning vlan 35
mac address-table static 00-00-00-00-00-02 vlan 35 interface GigabitEthernet 1/4 spanning-tree
mst te vlan 35
stream 20
dmac 00-00-00-00-00-02
outer-tag vid 35
interface GigabitEthernet 1/2,3
switchport mode trunk
stream-id 20
interface GigabitEthernet 1/4
switchport mode trunk
tsn frer 20
mode recovery
ingress stream-id-list 20
frer-vlan 35
egress interface GigabitEthernet 1/4
admin-state enable
```

6.7.5. Switch F

This switch performs recovery exactly the same way as does Switch B from Example 1-2.

Whether to configure the egress port that connects to the Listener as an access port or as a trunk/hybrid port is up to the network administrator. Here, we have re-used that from Example 1-2.

```
vlan 1
vlan 100
no flooding
no mac address-table learning vlan 100
mac address-table static 00-00-00-00-00-02 vlan 100 interface GigabitEthernet 1/4 spanning-
tree mst te vlan 100
stream 20
dmac 00-00-00-00-00-02
outer-tag vid 35
interface GigabitEthernet 1/2,3
switchport hybrid allowed vlan 1
switchport mode hybrid
stream-id 20
interface GigabitEthernet 1/4
switchport access vlan 100
tsn frer 20
mode recovery
ingress stream-id-list 20
frer-vlan 100
egress interface GigabitEthernet 1/4
recovery terminate
admin-state enable
```

6.8. Status

FRER instances' status is displayed with the show tsn frer [instance(s)] [details] ICLI command in EXEC mode.

An overview is displayed if omitting details , like the following:

```
# show tsn frer
Inst     Operational          State Mode          Latent Errors
-----    -----------------    ----------------    ------------------
  3      Admin  disabled       Generation
 10             Active         Generation
 14      Admin disabled        Recovery
101      Active (warnings)     Recovery                  No
```

In this particular case, we have four FRER instances, two of which are in generation mode, and two in recovery. Two are administratively disabled and two are active, one of which (FRER instance #101) has operational warnings.
The active recovery instances also have a column with Latent Errors , which may show Yes or No if latent error discovery is enabled. Otherwise, it shows Check disabled .

It is interesting to know why FRER instance #101 has warnings. To see them, ask for details:

```
# show tsn frer 101 details
Instance: 101
Operational state: Active
Mode: Recovery
Ingress Stream IDs: 20
FRER VLAN: 35
Egress interfaces: GigabitEthernet 1/4
Recovery algorithm: Vector (with history-length 2)
Reset timeout: 1000 ms
Take-no-sequence: No
Terminating FRER: No
Latent error detection: Enabled (with difference 100 packets, period 2000 ms, paths = 2, and
reset period 30000 ms)
Latent errors: No
Operational warnings: At least one of the ingress streams doesn't exist #
```

The operational warnings are shown at the bottom of the output. Here, it says that at least one of the ingress streams doesn't exist, meaning that the stream with ID 20 has not yet been created (or has been deleted), so the FRER instance is active but no frames will get mapped to it. Once stream ID 20 has been created and installed on at least one ingress port, will the status become updated automatically.
Besides the operational warnings, the details contain various configuration properties and whether any latent errors have been discovered.

Besides the operational warnings, the details contain various configuration properties and whether any latent errors have been discovered.

6.9. Statistics

The counters mentioned throughout this configuration guide can be shown with show tsn frer statistics [instance(s)] [details] ICLI command in EXEC mode.

An overview, containing number of resets, number of passed packets, and number of discarded packets, is displayed if omitting details :

```
# show tsn frer statistics
```

Inst	Oper	State	Mode	Egress I/F	Resets	Passed	Discarded
3	Admin	disabled	Generation		0		
10	Active		Generation		1		
14	Admin	disabled	Recovery				
20	Active		Recovery	Gi 1/4	1	1004	1004
				Gi 1/5	1	1004	1004

To see detailed counters, add details to the command. In the following example, only details for one instance is shown:

```
# show tsn frer statistics 20 details
Instance                : 20
Operational state       : Active
Mode                    : Recovery

Egress interface        : GigabitEthernet 1/4
Passed                  : 1004
Discarded               : 1004
Out-of-order            : 0
Rogue                   : 0
Lost                    : 1
Tagless                 : 0
Resets                  : 6
Latent error resets     : 1863
Egress interface        : GigabitEthernet 1/5
Passed                  : 1004
Discarded               : 1004
Out-of-order            : 0
Rogue                   : 0
Lost                    : 1
Tagless                 : 0
Resets                  : 6
Latent error resets     : 1863
```

This FRER instance is in non-individual recovery mode, so it only has compound counters per egress interface.

If the ingress stream is split into two (one per ingress port) and individual recovery is enabled, the output of the detailed statistics could look like the following:

```
# show tsn frer statistics 20 details
Instance                : 20
Operational state       : Active
Mode: Recovery
Egress interface        : GigabitEthernet 1/4
Ingress Stream IDs      : Compound          20          21
                          ----------  ----------  ----------
Passed:                       1004        1004        1004
Discarded:                    1004           0           0
Out-of-order:                    0           0           0
Rogue:                           0           0           0
Lost:                            2           2           2
Tagless:                         0           0           0
Resets:                          4           4           4
Latent error resets:             0           0           0
Egress interface        : GigabitEthernet 1/5
Ingress Stream IDs      : Compound          20          21
                          ----------  ----------  ----------
Passed:                       1004        1004        1004
Discarded:                    1004           0           0
Out-of-order:                    0           0           0
Rogue:                           0           0           0
Lost:                            2           2           2
Tagless:                         0           0           0
Resets:                          4           4           4
Latent error resets:             0           0           0
```

Now, the statistics is presented in a table per egress interface. The compound counters are shown in the first column. Subsequent columns hold individual counters per ingress stream ID.

In generation mode, there is only one single counter, namely the number of times the sequence number generator is reset. This brings us to the following section.

6.10. Reset Control

6.10.1. Reset Recovery Functions

Although of doubtable use, it is possible to force the recovery function(s) to reset for a particular FRER instance. This corresponds to setting frerSeqRcvyReset (802.1CB-2017, clause 10.4.1.4). It is not possible to control this per egress port or per individual recovery function; all functions get reset. Here's how:

```
# show tsn frer 20 statistics
Inst Oper. State Mode Egress I/F Resets Passed Discarded ---- -------------- ---------- ---------- -----
------- ------------ ------------
20 Active Recovery Gi 1/4 6 1004 1004 Gi 1/5 6 1004 1004
# tsn frer 20 reset
# show tsn frer 20 statistics
Inst Oper. State Mode Egress I/F Resets Passed Discarded ---- -------------- ---------- ---------- -----
------- ------------ ------------
20 Active Recovery Gi 1/4 7 1004 1004 Gi 1/5 7 1004 1004
```

The example above first shows counters for the FRER instance, then performs the reset, and then shows counters again. This shows that the number of resets has increased from 6 to 7.

6.10.2. Reset Sequence Number Generator

If the FRER instance is in generation mode, the same command has another effect: It resets the sequence number generator to provide sequence number 0 in the next transmitted packet. Example:

```
# show tsn frer statistics 10
Inst Oper. State Mode Egress I/F Resets Passed Discarded ---- -------------- ---------- ---------- -----
------- ------------ ------------ 10 Active Generation 1
# tsn frer 10 reset
# show tsn frer statistics 10
Inst Oper. State Mode Egress I/F Resets Passed Discarded ---- -------------- ---------- ---------- -----
------- ------------ ------------ 10 Active Generation 2
```

Here, we also show the counters for the FRER instance before and after, just to emphasize that it has an effect to issue the reset command.

One thing worth noting is that once a FRER instance - whether in recovery or generation mode - is getting enabled (admin-state enable), an automatic reset happens, causing the counters to count to 1.