

PSFP: opportunities, limits and open questions

IEEE ETFA 2024 – WS04 – Time-Sensitive Networks from academia to industry: trends and challenges beyond the hype

Marc Boyer

2024-01-25

Outline

- 1 What is PSFP ?
 - Global presentation
 - Per element behaviour
- 2 What PSFP is for ?
 - PSFP for fault tolerance
 - PSFP for nominal behaviour
- 3 Conclusion

Outline

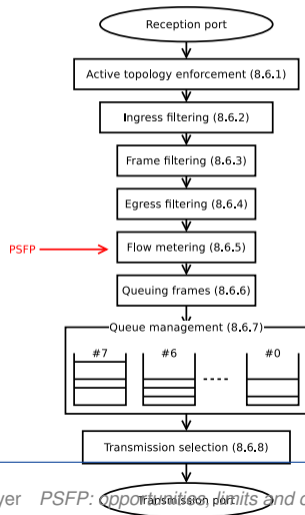
- 1 What is PSFP ?
 - Global presentation
 - Per element behaviour
- 2 What PSFP is for ?
 - PSFP for fault tolerance
 - PSFP for nominal behaviour
- 3 Conclusion

Outline

- 1 What is PSFP ?
 - Global presentation
 - Per element behaviour
- 2 What PSFP is for ?
 - PSFP for fault tolerance
 - PSFP for nominal behaviour
- 3 Conclusion

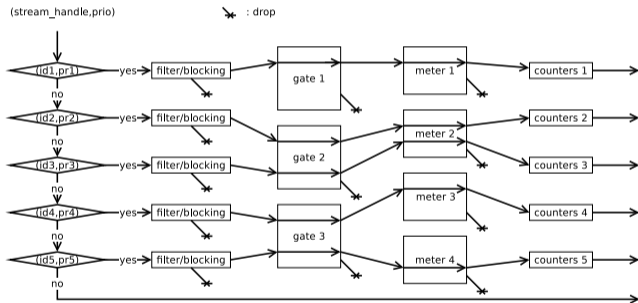
What is PSFP ?

- PSFP : Per-Stream Filtering and Policing
- defined in 802.1Qci (2017), now included in 802.1Q (2022)
- last step of the filtering pipe (between reception and queuing)
- ability to
 - count "things"
 - drop or accept frames
 - local change of frame priority



What is the architecture of PSFP ?

- An ordered list of *stream filters*
- A set of *stream gates*
- A set of *flow meters*



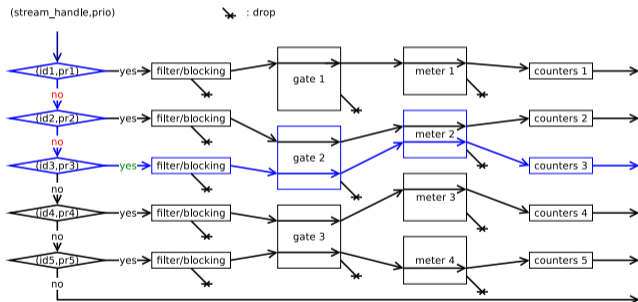
Outline

- 1 What is PSFP ?
 - Global presentation
 - Per element behaviour
- 2 What PSFP is for ?
 - PSFP for fault tolerance
 - PSFP for nominal behaviour
- 3 Conclusion

Global behaviour of PFSP

For each received frame

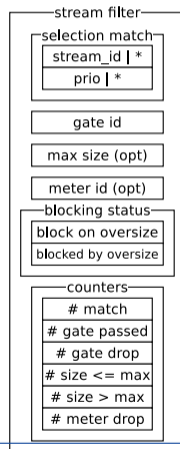
- 1 Select the first matching *filter* rule
 - look only `stream_handle` and `priority` tag
- 2 Forward to the (singles) *gate* and *meter* associated to this filter
 - can be shared by several filters
- 3 Update some counters
 - unique per filter
- 4 Forward to queuing



Each step can drop the frame

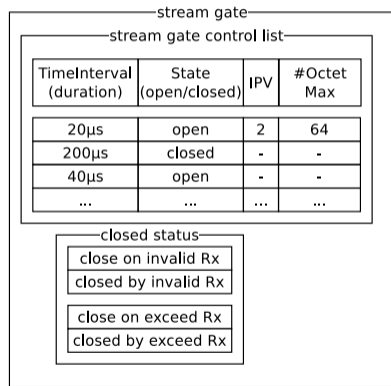
Behavior of a filter

- A rule is pair, each element either a single value or any value (*)
 - no way to have *groups* of streams (except expanding rules)
 - no way to get the input port
- A path through a *gate* and an optional *meter*
- An optional *maximal size*
 - Drop, and can lead to filter blocking
- Counters



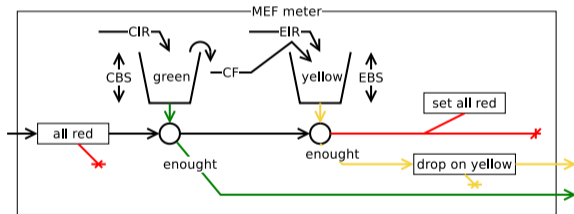
Behavior of a stream gate

- A static cyclic behaviour
 - drop frame if closed
 - can locally change the frame priority (IPV : Internal Priority Value)
 - can allow a the maximal number of octets per interval
- Like the output port GCL for *Time Aware Shaper* (TAS), but
 - several stream GCL vs. per port TAS GCL
 - more actions (IPV, max octets)



Behavior of a stream meter

- A double token-bucket
 - *Committed* token-bucket
 - *Excess* token-bucket
- Coming from non IEEE standard
- Used to mark frames
 - Green : pass
 - Red : drop
 - Yellow : drop or set drop_eligible frame bit



Outline

- 1 What is PSFP ?
 - Global presentation
 - Per element behaviour
- 2 What PSFP is for ?
 - PSFP for fault tolerance
 - PSFP for nominal behaviour
- 3 Conclusion

Outline

- 1 What is PSFP ?
 - Global presentation
 - Per element behaviour
- 2 What PSFP is for ?
 - PSFP for fault tolerance
 - PSFP for nominal behaviour
- 3 Conclusion

Whats is fault tolerance ?

- Kind of faults

Whats is fault tolerance ?

- Kind of faults
 - Loss of message (Buffer overflow, Frame corruption. . .)

Whats is fault tolerance ?

- Kind of faults
 - Loss of message (Buffer overflow, Frame corruption. . .)
 - Routing error

Whats is fault tolerance ?

- Kind of faults
 - Loss of message (Buffer overflow, Frame corruption. . .)
 - Routing error
 - Wrong timing (Synchronisation error, Wrong shaping. . .)

Whats is fault tolerance ?

- Kind of faults

- Loss of message (Buffer overflow, Frame corruption. . .)
- Routing error
- Wrong timing (Synchronisation error, Wrong shaping. . .)

- Fault protection

Whats is fault tolerance ?

- Kind of faults

- Loss of message (Buffer overflow, Frame corruption. . .)
- Routing error
- Wrong timing (Synchronisation error, Wrong shaping. . .)

- Fault protection

- *Wrong* frames may be rejected

Whats is fault tolerance ?

- Kind of faults

- Loss of message (Buffer overflow, Frame corruption. . .)
- Routing error
- Wrong timing (Synchronisation error, Wrong shaping. . .)

- Fault protection

- *Wrong* frames may be rejected
- *Lost* frames are not re-generated (in Ethernet)

Whats is fault tolerance ?

- Kind of faults
 - Loss of message (Buffer overflow, Frame corruption. . .)
 - Routing error
 - Wrong timing (Synchronisation error, Wrong shaping. . .)
- Fault protection
 - *Wrong* frames may be rejected
 - *Lost* frames are not re-generated (in Ethernet)
- Fault tolerance *containment*

Whats is fault tolerance ?

- Kind of faults
 - Loss of message (Buffer overflow, Frame corruption. . .)
 - Routing error
 - Wrong timing (Synchronisation error, Wrong shaping. . .)
- Fault protection
 - *Wrong* frames may be rejected
 - *Lost* frames are not re-generated (in Ethernet)
- Fault tolerance *containment*
 - Protect sub-system from fault in other sub-system

Whats is fault tolerance ?

- Kind of faults

- Loss of message (Buffer overflow, Frame corruption. . .)
- Routing error
- Wrong timing (Synchronisation error, Wrong shaping. . .)

- Fault protection

- *Wrong* frames may be rejected
- *Lost* frames are not re-generated (in Ethernet)

- Fault tolerance *containment*

- Protect sub-system from fault in other sub-system

Whats is fault tolerance ?

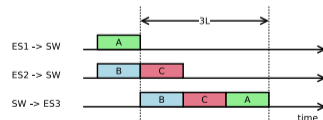
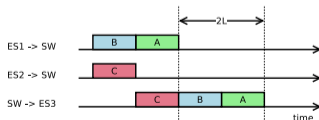
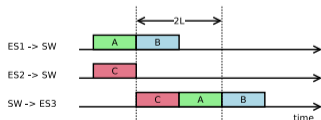
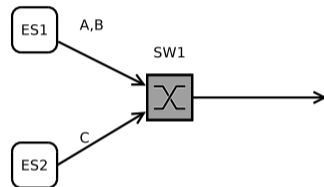
- Kind of faults
 - Loss of message (Buffer overflow, Frame corruption. . .)
 - Routing error
 - Wrong timing (Synchronisation error, Wrong shaping. . .)
- Fault protection
 - *Wrong* frames may be rejected
 - *Lost* frames are not re-generated (in Ethernet)
- Fault tolerance *containment*
 - Protect sub-system from fault in other sub-system

Networks fault containment

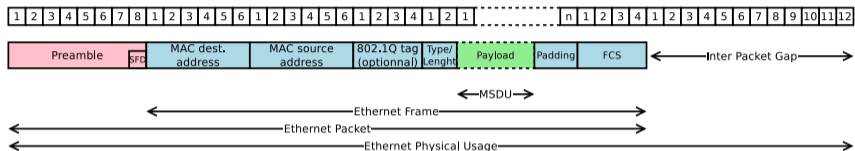
A network is said to “contain faults” if a data flow crossing only switches in nominal state experiences a quality of service conforms to the contract established.

Routing errors

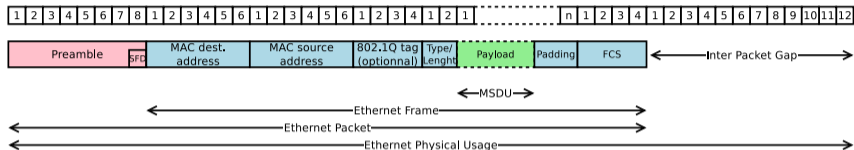
- Routing error has impact on performances
 - Routing error create bursts
 - Routing error may case unexpected delays
- Pre-TSN filtering is based on
 - source address
 - destination address
 - VLAN identifier
- PSFP has no information on input port



Frame size and media overhead

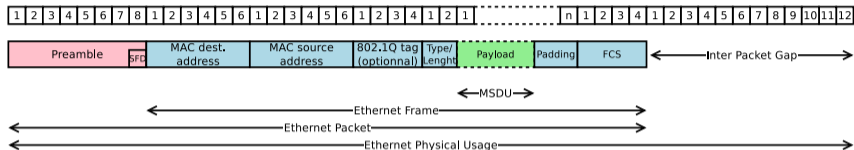


Frame size and media overhead



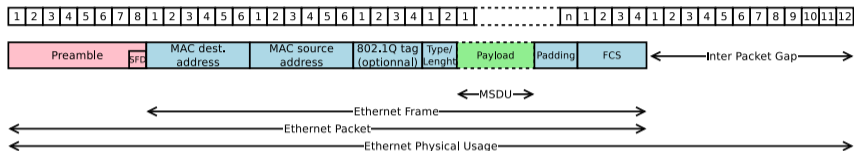
- When configuring counter and test, pay attention on what is checked

Frame size and media overhead



- When configuring counter and test, pay attention on what is checked
- PSFP counters do not count *media specific overhead* (preamble, IPG)

Frame size and media overhead

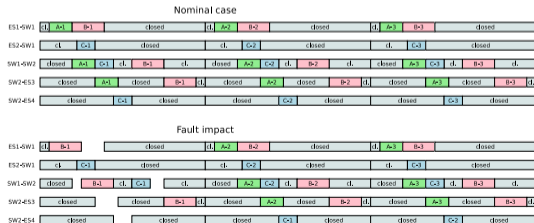
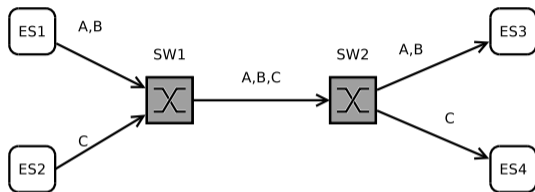


- When configuring counter and test, pay attention on what is checked
- PSFP counters do not count *media specific overhead* (preamble, IPG)
 - ⇒ Over-provisioning $\approx 30\%$

Frame size	# frames	# physical byte-time	delay (at 1GB/s)
64	1	84	672 ns
64	10	840	6.72 μ s
640	1	660	5.28 μ s
64	20	1680	13.44 μ s
1280	1	1320	10.56 μ s

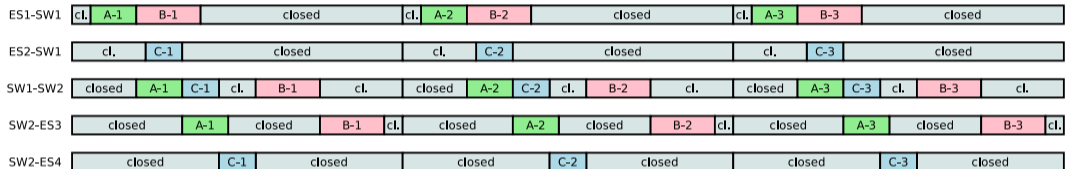
Time Aware Shaper (TAS) Fragility

- TAS recall : cf. Lukas Osswald talk
- TAS fragility
 - scheduling based on gate open/close
 - a frame can use the slot of another (if another frame is lost)
 - may break all the schedule

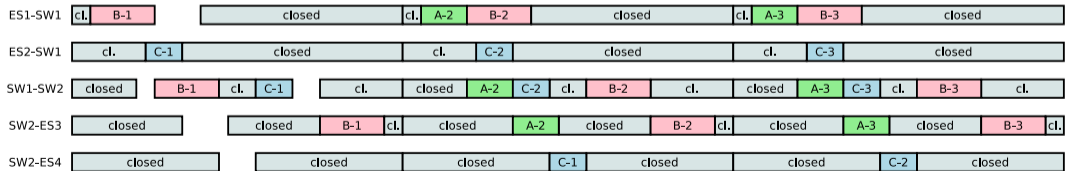


Time Aware Shaper (TAS) Fragility example

Nominal case



Fault impact



TAS Fragility solutions

Flow isolation

Build schedule such that there is never two frames from different flows in the same queue at the same time [Craciunas et al., 2016].

TAS Fragility solutions

Flow isolation

Build schedule such that there is never two frames from different flows in the same queue at the same time [Craciunas et al., 2016].

- Reduces the solution space

TAS Fragility solutions

Flow isolation

Build schedule such that there is never two frames from different flows in the same queue at the same time [Craciunas et al., 2016].

- Reduces the solution space
- PFSP could relax this requirement

TAS Fragility solutions

Flow isolation

Build schedule such that there is never two frames from different flows in the same queue at the same time [Craciunas et al., 2016].

- Reduces the solution space
- PFSP could relax this requirement
 - allow to drop frames "out of time window"

TAS Fragility solutions

Flow isolation

Build schedule such that there is never two frames from different flows in the same queue at the same time [Craciunas et al., 2016].

- Reduces the solution space
- PFSP could relax this requirement
 - allow to drop frames "out of time window"
 - research still to be done

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes
 - Long-term bandwidth bound : sum of data flow rate

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes
 - Long-term bandwidth bound : sum of data flow rate
 - Faults

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes
 - Long-term bandwidth bound : sum of data flow rate
 - Faults
 - flows sending too much

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes
 - Long-term bandwidth bound : sum of data flow rate
 - Faults
 - flows sending too much
 - violation of CBS shaping

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes
 - Long-term bandwidth bound : sum of data flow rate
 - Faults
 - flows sending too much
 - violation of CBS shaping
 - Fault containment

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes
 - Long-term bandwidth bound : sum of data flow rate
 - Faults
 - flows sending too much
 - violation of CBS shaping
 - Fault containment
 - long term : required to ensure no buffer overflow

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes
 - Long-term bandwidth bound : sum of data flow rate
 - Faults
 - flows sending too much
 - violation of CBS shaping
 - Fault containment
 - long term : required to ensure no buffer overflow
 - short term : use to get better bounds

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes
 - Long-term bandwidth bound : sum of data flow rate
 - Faults
 - flows sending too much
 - violation of CBS shaping
 - Fault containment
 - long term : required to ensure no buffer overflow
 - short term : use to get better bounds
 - ⇒ when considering faults, delay provisioning must ignore CBS benefits !

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes
 - Long-term bandwidth bound : sum of data flow rate
 - Faults
 - flows sending too much
 - violation of CBS shaping
 - Fault containment
 - long term : required to ensure no buffer overflow
 - short term : use to get better bounds
 - ⇒ when considering faults, delay provisioning must ignore CBS benefits !
- Is CBS useless ?

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes
 - Long-term bandwidth bound : sum of data flow rate
 - Faults
 - flows sending too much
 - violation of CBS shaping
 - Fault containment
 - long term : required to ensure no buffer overflow
 - short term : use to get better bounds
 - ⇒ when considering faults, delay provisioning must ignore CBS benefits !
- Is CBS useless ?
 - On a given switch, CBS preserves lower priority queues

Time conformance in case of CBS shaper

- CBS recall : cf. Lisa Maille
- The traffic contract problem : a queue receiving flows from upstream CBS queues
 - Traffic shape
 - Short-term bandwidth bound : sum of CBS slopes
 - Long-term bandwidth bound : sum of data flow rate
 - Faults
 - flows sending too much
 - violation of CBS shaping
 - Fault containment
 - long term : required to ensure no buffer overflow
 - short term : use to get better bounds
 - ⇒ **when considering faults, delay provisioning must ignore CBS benefits!**
- Is CBS useless ?
 - On a given switch, CBS preserves lower priority queues
 - Along a switch path, you can not contain CBS faults

Outline

- 1 What is PSFP ?
 - Global presentation
 - Per element behaviour
- 2 What PSFP is for ?
 - PSFP for fault tolerance
 - PSFP for nominal behaviour
- 3 Conclusion

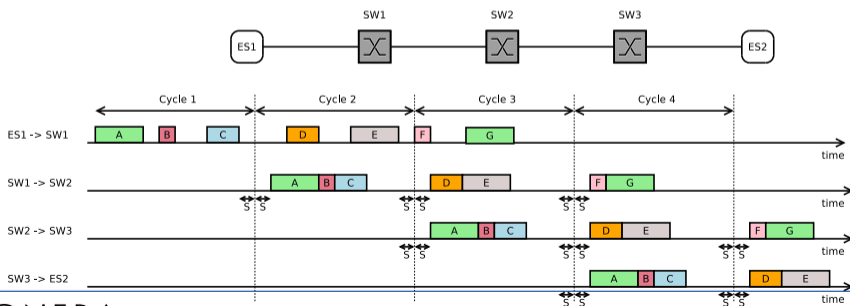
CQF : Cyclic Queuing and Forwarding

- Principles :

- Divide time into slots of duration T (even and odd)
- Frames received in one slot are forwarded next slot

- Guarantees :

- For a flow crossing h switches
- Latency in $[(h - 1)T, (h + 1)T]$
- Jitter in $[0, 2T]$



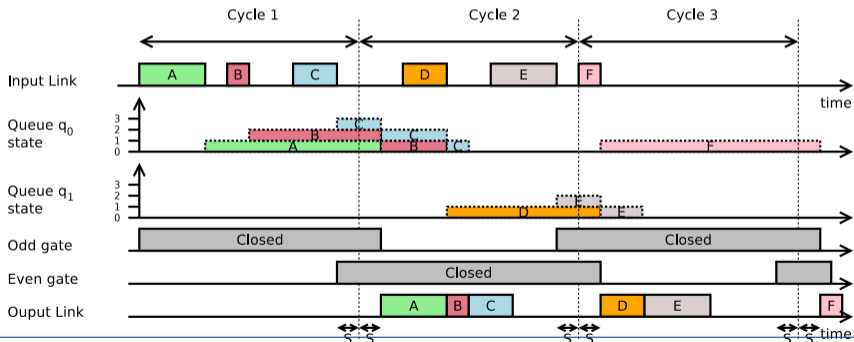
CQF implementation

- PSFP :

- Based on the *stream gate control list*
- Always open, just change IPV

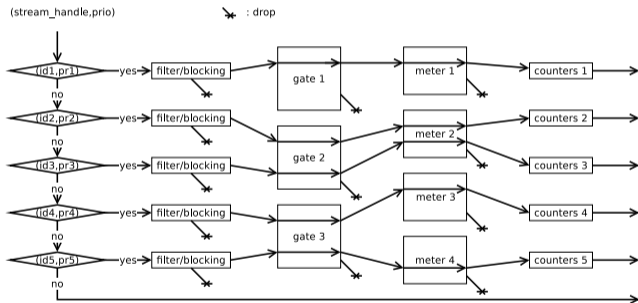
- Output GCL :

- block / open associated queues



CQF attention point

- Finite number of filters and stream gates
 - How to handle large number of streams ?
 - Use priority field ?
 - No bijection between stream_handle and streamID ?
- Synchronisation between input GCL and output GCL



Outline

- 1 What is PSFP ?
 - Global presentation
 - Per element behaviour
- 2 What PSFP is for ?
 - PSFP for fault tolerance
 - PSFP for nominal behaviour
- 3 Conclusion

Conclusion

- Faults occur in real life
- PSFP offers mechanisms to contain faults
- Research opportunities (TAS scheduling)
- Some weaknesses
 - accuracy (media overhead)
 - routing errors
 - CBS compatibility

References

- White paper on PSFP [[Boyer, 2023](#)]
- Flow isolation [[Craciunas et al., 2016](#)]
- The PSFP addenda [[TSN-Group, 2017](#)]

[[Boyer, 2023](#)] Boyer, M. (2023).

Usage of TSN Per-Stream Filtering and Policing.
working paper or preprint.

[[Craciunas et al., 2016](#)] Craciunas, S. S., Oliver, R. S., Chmelík, M., and Steiner, W. (2016).

Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks.

In *Proc. of the 24th Int. Conf. on Real-Time Networks and Systems (RTNS'16)*, RTNS'16, pages 183–192, New York, NY, USA. Association for Computing Machinery.

[[TSN-Group, 2017](#)] TSN-Group, I. (2017).

IEEE standard for local and metropolitan area networks—bridges and bridged networks—amendment 28 : Per-stream filtering and policing.