P802.1Qcr/D0.5 – Last Comments

Johannes Specht (University of Duisburg-Essen)

Introduction

About

The following comments P802.1Qcr/D0.5 are unresolved

- #4 and #32: Location of maximum SDU size filters
- #11: Generalization of the media-dependent overhead parameter per Port

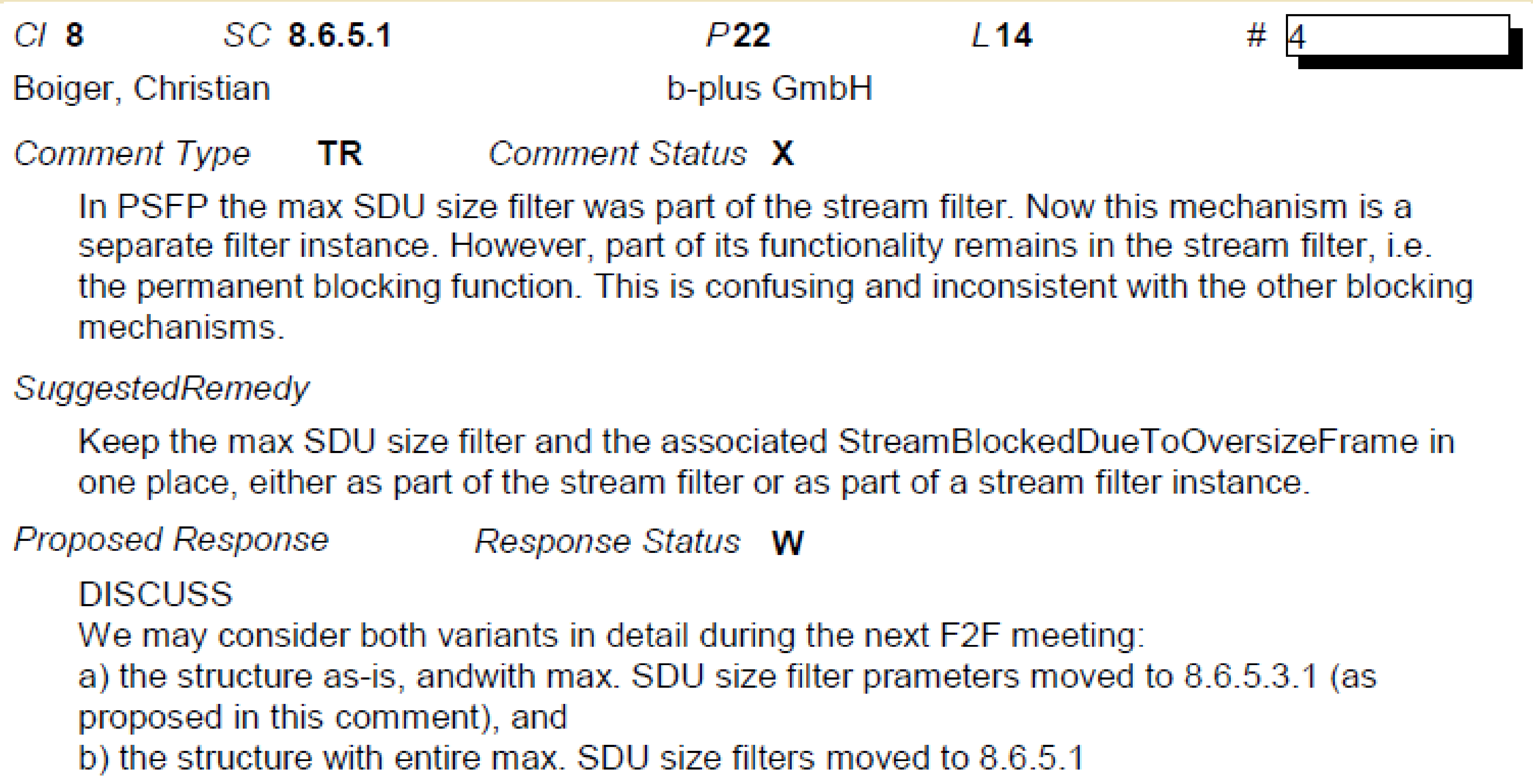
Purpose of this Presentation

- Present previews of clause 8 and 12 for offline review during the week:
  - Clause 8 – Introduces two variants:
    1. One with the model requested in #4 (clause structure as-is in D0.5)
    2. One with the model requested in #32 (stream filter specifications and maximum SDU size filtering in the stream filter clause)
  - Clause 12- Includes the generalization, as requested in #11
- Prepare to conclude the comment resolution end of this week

Comments #4 and #32

Where to describe maximum SDU size filtering…

Variant 1: As-is, with little adjustments



Cl **8** *SC* **8.6.5.1** *P* **22** *L* **14** # 4

Boiger, Christian b-plus GmbH

Comment Type **TR** *Comment Status* **X**

In PSFP the max SDU size filter was part of the stream filter. Now this mechanism is a separate filter instance. However, part of its functionality remains in the stream filter, i.e. the permanent blocking function. This is confusing and inconsistent with the other blocking mechanisms.

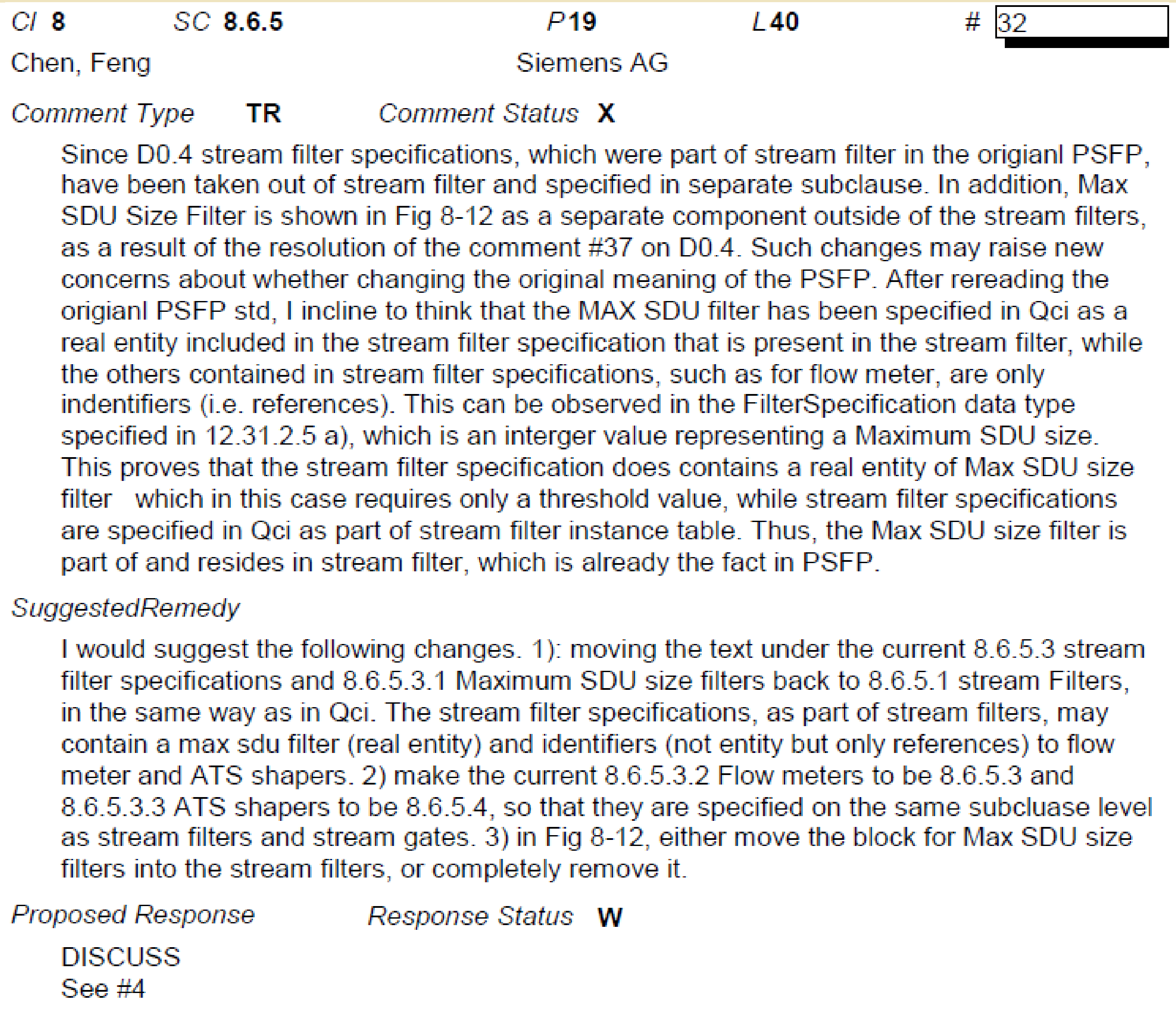
SuggestedRemedy

Keep the max SDU size filter and the associated StreamBlockedDueToOversizeFrame in one place, either as part of the stream filter or as part of a stream filter instance.

Proposed Response *Response Status* **W**

DISCUSS
We may consider both variants in detail during the next F2F meeting:
a) the structure as-is, andwith max. SDU size filter prameters moved to 8.6.5.3.1 (as proposed in this comment), and
b) the structure with entire max. SDU size filters moved to 8.6.5.1

Variant 2: max. SDU size filtering in Stream Filters



Cl **8** *SC* **8.6.5** *P* **19** *L* **40** # 32

Chen, Feng Siemens AG

Comment Type **TR** *Comment Status* **X**

Since D0.4 stream filter specifications, which were part of stream filter in the origianl PSFP, have been taken out of stream filter and specified in separate subclause. In addition, Max SDU Size Filter is shown in Fig 8-12 as a separate component outside of the stream filters, as a result of the resolution of the comment #37 on D0.4. Such changes may raise new concerns about whether changing the original meaning of the PSFP. After rereading the origianl PSFP std, I incline to think that the MAX SDU filter has been specified in Qci as a real entity included in the stream filter specification that is present in the stream filter, while the others contained in stream filter specifications, such as for flow meter, are only indentifiers (i.e. references). This can be observed in the FilterSpecification data type specified in 12.31.2.5 a), which is an interger value representing a Maximum SDU size. This proves that the stream filter specification does contains a real entity of Max SDU size filter which in this case requires only a threshold value, while stream filter specifications are specified in Qci as part of stream filter instance table. Thus, the Max SDU size filter is part of and resides in stream filter, which is already the fact in PSFP.

SuggestedRemedy

I would suggest the following changes. 1): moving the text under the current 8.6.5.3 stream filter specifications and 8.6.5.3.1 Maximum SDU size filters back to 8.6.5.1 stream Filters, in the same way as in Qci. The stream filter specifications, as part of stream filters, may contain a max sdu filter (real entity) and identifiers (not entity but only references) to flow meter and ATS shapers. 2) make the current 8.6.5.3.2 Flow meters to be 8.6.5.3 and 8.6.5.3.3 ATS shapers to be 8.6.5.4, so that they are specified on the same subcluase level as stream filters and stream gates. 3) in Fig 8-12, either move the block for Max SDU size filters into the stream filters, or completely remove it.

Proposed Response *Response Status* **W**

DISCUSS
See #4

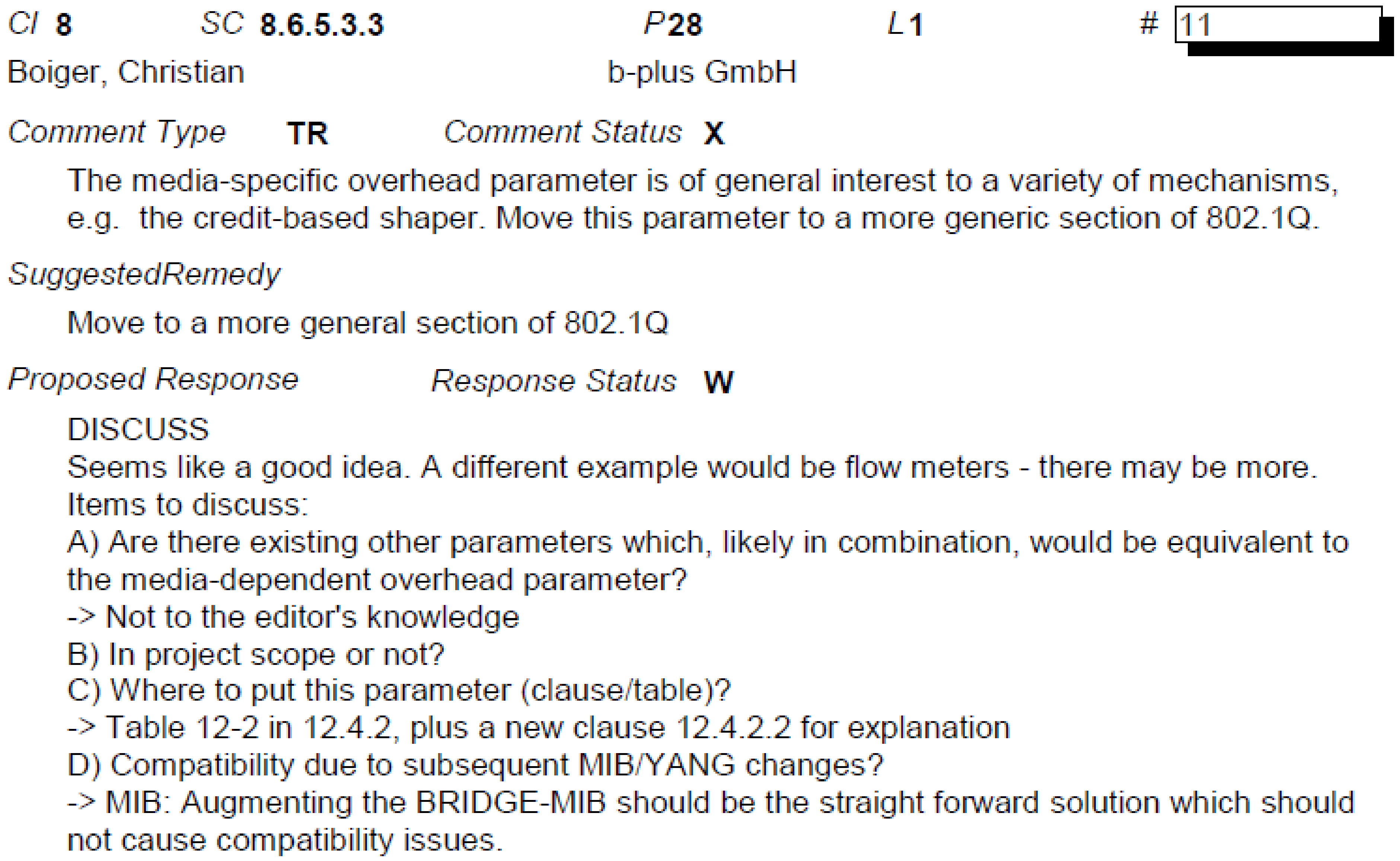
More on the clause 8 preview variants

- Variant 1 is merged in the current D1.0 document in progress:
  - Includes changes for already resolved comments on D0.5
  - Several editor's notes removed, as discussed during the last F2F meeting
  - **New** note: Max. SDU size filter parameters are in the stream filter instance table (there is a one-to-one relationship). The parameters were there in 802.1Qci, and cannot easily be moved in a separate table
  - **New** explicit statement (in consequence): At most one Max. SDU size filter per stream filter. In detail, this is nothing new (compared to Qci).
    - Multiple max SDU size filters attached to a single stream filter have **zero benefit**.
- Variant 2 is derived from Variant 1:
  - Puts the set of stream filter specifications and maximum SDU size filtering functionality into Stream Filters, as was in Qci, including changes of the clauses, figures, etc.
  - All text edits tracked, marked in blue and green

- Both variants are **technically equivalent**! It’s a matter of style.
  - Variant 1: Separates
    - action elements (gates, flow meters, ATS schedulers, **and max. SDU size filters**) from
    - identification elements (stream filters)
  - Variant 2: Closer to the original text in Qci, mixes max. SDU size filtering action with identification (stream filters)
- The editor has a slight tendency to stick with variant 1, but this is not a strong opinion. However, if the TSN group decides for variant 2, this is ok.
- There is time to review both variants during this week, although the comment will be closed at the end of this week.

Comment #11

Generalizing the media-dependent overhead


Cl **8** *SC* **8.6.5.3.3** *P* **28** *L* **1** # 11

Boiger, Christian b-plus GmbH

Comment Type **TR** *Comment Status* **X**

The media-specific overhead parameter is of general interest to a variety of mechanisms, e.g. the credit-based shaper. Move this parameter to a more generic section of 802.1Q.

SuggestedRemedy

Move to a more general section of 802.1Q

Proposed Response *Response Status* **W**

DISCUSS
Seems like a good idea. A different example would be flow meters - there may be more.
Items to discuss:
A) Are there existing other parameters which, likely in combination, would be equivalent to the media-dependent overhead parameter?
-> Not to the editor's knowledge
B) In project scope or not?
C) Where to put this parameter (clause/table)?
-> Table 12-2 in 12.4.2, plus a new clause 12.4.2.2 for explanation
D) Compatibility due to subsequent MIB/YANG changes?
-> MIB: Augmenting the BRIDGE-MIB should be the straight forward solution which should not cause compatibility issues.

Remaining Topics

- Definition in clause 12: See the preview, it is in 12.4.x.
  - Please provide feedback until the second session at the end of this week. The previews are individual contributions. You don't need to be a voting member, nor is the comment form required (verbally, e-mail, …).
- Scope: This is in scope of the project.
  - *"The amendment specifies an information model for the capabilities of asynchronous traffic shaping. It further specifies a YANG data model and Management Information Base (MIB) modules both based on that information model to support configuration and status reporting."*
- MIB: As stated in the response, augmenting the existing Bridge MIB should not be an issue. The editor can copy the required contents into the Qcr draft once they are provided and ready (i.e., a later draft).
  - Ok?
- YANG: Augmentation via the ATS YANG file is no solution. The parameter would be tied to ATS, and the goal is to generalize it → the YANG contents developed in Qcp need to be adjusted – the editor can do the adjustments (after pulling these contents into Qcr).
  - What is the best way to do so?
  - And when?

Clause 8 and 12: Variant 1

Structured as is in D0.5, max. SDU size filters in 8.6.5.3.1

8. Principles of bridge operation

<<Editor's Note: Introduction to Clause 8 in this Document

Subsequent content is based on
http://www.ieee802.org/1/files/public/docs2016/Qcr-specht-specification-mapping-proposal-0516-v01.pdf and
http://www.ieee802.org/1/files/public/docs2016/Qcr-specht-editing-1116-v01.pdf
I.e., (a) re-use procedures and tables of PSFP, (b) specify ATS functions that can be placed in reception Ports in clause 8.6.5, while (c) permitting implementation of the ATS scheduler computational part deeper in the forwarding process (i.e., in transmission ports).

In this draft of P802.1Qcr, contents of clauses 8.6.5.1 (PSFP) and 8.6.5.2 (ATS) were merged in sub-clauses 8.6.5.1 through 8.6.5.4 to simplify the descriptions/avoid redundancy. Additional slides on the restructuring are found here: http://ieee802.org/1/files/public/docs2018/cr-specht-d04-0518-v03.pdf.
- 8.6.5.1 through 8.6.5.3 define building blocks
- 8.6.5.4 now defines which of these building blocks assemble ATS and PSFP.
The initial text in 8.6.5 from 802.1Q-Rev-2018 has been moved into a separate sub-clause (8.6.5.5).>>

8.6 The Forwarding Process

Delete clause 8.6.5 and the contained figures and tables

Insert clause 8.6.5 after clause 8.6.4, including the figures and tables, as shown below (renumber existing figures and tables as necessary):

8.6.5 Flow classification and metering

The Forwarding Process may apply flow classification and metering to frames that are received on a Bridge Port and have one or more potential transmission ports.

Bridges and End Stations may implement the following elements to provide Flow classification and metering for applications defined in this standard:

a) Stream Filters (8.6.5.1)
b) Stream Gates (8.6.5.2)
c) Stream Filter Specifications (8.6.5.3):
    1) Maximum SDU Size Filters (8.6.5.3.1)
    2) Flow Meters (8.6.5.3.2)
    3) ATS Schedulers (8.6.5.3.3)

The relationship and ordering between these elements is illustrated in Figure 8-12.

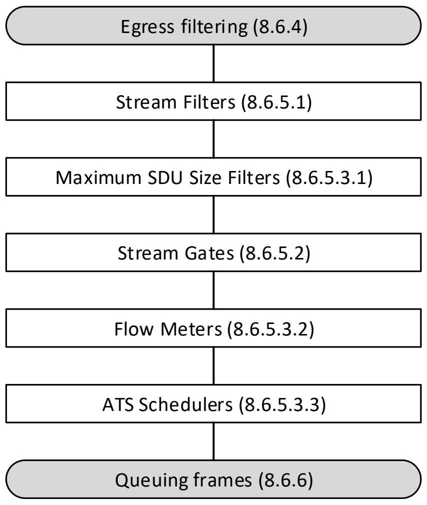


Figure 8-12—Flow classification and metering elements

Subsets of specific elements from 8.6.5.1, 8.6.5.2, and 8.6.5.3 and their use by particular applications from this standard are defined in 8.6.5.4. Applications beyond the scope of this standard might not use the elements from 8.6.5.1, 8.6.5.2, and 8.6.5.3, in which case the general compatibility requirements in 8.6.5.5 are relevant.

NOTE —Examples for applications beyond this standard include interworking with L3 DiffServ, and support for the use of Provider Bridging for Carrier Ethernet Services defined in MEF.

8.6.5.1 Stream Filters

Stream filters identify the frames of streams and associate these frames with Stream Gates (8.6.5.2) and Stream Filter Specifications (8.6.5.3) for subsequent actions.

Each Bridge component provides a *Stream Filter Instance Table* with parameters and variables of up to *MaxStreamFilterInstances* stream filter instances. Each stream filter instance has the following parameters and variables:

a) An integer *stream filter instance identifier*.
b) A *stream_handle specification*.
c) A *priority specification*.
d) An integer *stream gate instance identifier*.
e) A set of zero or more *stream filter specifications*.
f) An integer *MatchingFramesCount* counter for frames matching the stream filter.
g) An integer *PassingFrameCount* counter for frames passing the associated stream gate (8.6.5.2).
h) An integer *NotPassingFrameCount* counter for frames not passing the associated stream gate (8.6.5.2).
i) A *PassingSDUCount* counter for frames that passed the Maximum SDU size filter (8.6.5.3.1).
j) A *NotPassingSDUCount* counter for frames that did not pass the Maximum SDU size filter (8.6.5.3.1).
k) A *RedFramesCount* counter for frames that were discarded by the flow meter (8.6.5.3.2).

The stream filter instance identifier uniquely identifies the filter instance, and acts as an index to the Stream Filter Instance Table.

The values of the *stream_handle* and *priority* parameters associated with a received frame determine which stream filter is selected for the frame. The stream_handle parameter is a sub-parameter of the connection_identifier parameter of the ISS (6.6), as provided by the Stream identification function in clause 6 of IEEE Std 802.1CB. A stream filter is selected if the stream_handle value of the frame matches the value of the stream_handle specification parameter and if the priority value of the frame matches the value of the priority specification parameter.

A stream_handle specification parameter can be either of the following:

l) A single stream_handle value, as specified in IEEE Std 802.1CB.
m) A wildcard value that matches any stream_handle value.

A priority specification parameter can be either of the following:

n) A single priority value.
o) A wildcard value that matches any priority value.

NOTE 1—The use of stream_handle and priority, along with the wild-carding rules previously stated, allow configuration possibilities that go beyond the selection of individual streams, as implied by the sub-clause title; for example, per-priority filtering and policing, or per-priority per-reception Port filtering and policing can be configured using these rules.

If a received frame matches zero or multiple stream filters, the behavior is as follows:

p) If no stream filter matches, the frame is processed as it would be the case without stream filters, stream gates, and stream filter specifications.
q) If more than one stream filters match, the stream filter with the smaller stream filter instance identifier is selected.

If a stream filter is selected for a frame according to the aforementioned rules, this frame is passed to the stream gate (8.6.5.2) referred by the stream gate instance identifier parameter and the Stream Filter Specifications (8.6.5.3) in the set of stream filter specifications for further actions.

NOTE 2—If it is desired to discard frames that do not match any other stream filter, rather than such frames being processed without filtering, this can be achieved by placing a stream filter at the end of the table, in which the stream_handle and priority are both wild-carded (set to the null value), and where the stream gate instance identifier points at a stream gate that is permanently closed.

The MatchingFramesCount counter of a stream filter is increased for each frame that matches the stream_handle and priority specification parameters. If multiple stream filters match the same frame, only the MatchingFramesCount counter of the selected filter according to rule in item q) is increased.

8.6.5.2 Stream Gates

Stream gates determine whether frames are discarded or passed for further processing by the remaining forwarding process. Stream gates are also able to change the associated traffic class of frames for later queuing decisions (8.6.6) via IPV assignments.

Each Bridge component provides a *Stream Gate Instance Table* with parameters and variables of up to *MaxStreamGateInstances* stream gate instances. Each stream gate instance is associated with the following parameters and variables:

a) An integer *stream gate instance identifier*.
b) An administrative and an operational *stream gate state* parameter.
c) An administrative and an operational *internal priority value specification*.

d) An administrative and an operational *gate control list*.
e) A boolean *GateClosedDueToInvalidRxEnable* parameter.
f) A boolean *GateClosedDueToInvalidRx* parameter.
g) A boolean *GateClosedDueToOctetsExceededEnable* parameter.
h) A boolean *GateClosedDueToOctetsExceeded* parameter.

The stream gate instance identifier uniquely identifies the stream gate instance, acts as an index to the Stream Gate and Instance Table, and associates stream filter instances (8.6.5.1) with the stream gate instance.

The relationship between stream filters, stream gates, and the associated variables and parameters of both is illustrated in Figure 8-13.

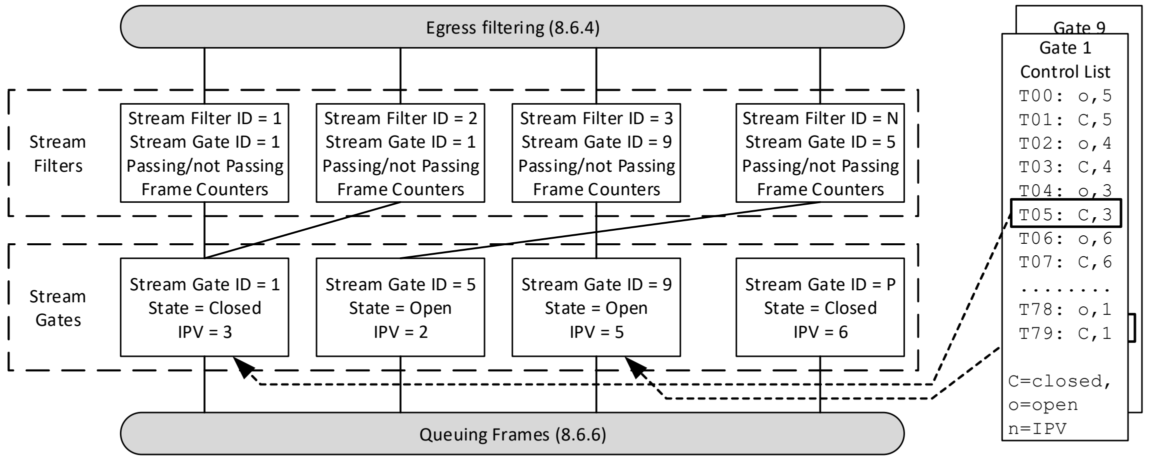


Figure 8-13—Stream Gates and Stream Filters

A stream gate can be in one of the following two states:

i) Open: Frames are permitted to pass through the stream gate.
j) Closed: Frames are not permitted to pass through the stream gate.

A frame that is permitted to pass through a stream gate is subject to subsequent actions by filter specifications (8.6.5.3) and queuing decisions (8.6.6), and the PassingFrameCount counter of the originating stream filter (8.6.5.1) is increased. A frame that is not permitted to pass through a stream gate is immediately discarded, and the NotPassingFrameCount counter of the originating stream filter is increased.

Each stream gate has an associated internal priority value specification. This specification allows to override the priority values associated with passing frames during subsequent queuing decisions (8.6.6) by assigning an *internal priority value* (IPV). An IPV specification can be either of the following:

k) A null value: The priority value associated with the frame is used to determine the frame's traffic class, using the Traffic Class Table as specified in 8.6.6. No IPV is assigned to the frame.
l) An IPV: The IPV is assigned to the frame and used, in place of the priority value associated with the frame, to determine the frame's traffic class using the Traffic Class Table as specified in 8.6.6.

An IPV assigned to a frame co-exists with the priority value associated with the frame (i.e., the IPV does not replace this priority value).

NOTE 1—The co-existence of the IPV and the priority value allows subsequent queuing decisions (8.6.6) to be based on the IPV, while the priority value is retained for transmission. A use case of IPV for ATS is the adjusting of per-hop delay bounds to satisfy end-to-end delay requirements in a specific network. Another use case for the ability to assign internal priority values can be found in Annex T (CQF).

The actual stream gate state and the IPV specification are reflected by the operational stream gate state and IPV specification parameters. Both parameters are controlled by a stream gate control state machine (8.6.10) associated with the stream gate. The state machine cyclically executes a stream gate control list that contains control operations with transitions of the operational stream gate state and IPV specification parameters, as specified in Table 8-4. The operational stream gate control list parameter reflects the actually executed list.

Table 8-4—Stream gate control operations

Operation name	Parameter(s)	Action
SetGateAndIPV	StreamGateState, IPV, TimeInterval, IntervalOctetMax	The StreamGateState parameter specifies a desired state, *open* or *closed*, for the stream gate, and the IPV parameter specifies a desired value of the IPV associated with the stream. On execution, the StreamGateState and IPV parameter values are used to set the operational values of the stream gate state and internal priority specification parameters for the stream. After *TimeInterval* ticks (8.6.9.4.16) has elapsed since the completion of the previous stream gate control operation in the stream gate control list, control passes to the next stream gate control operation. The optional IntervalOctetMax parameter specifies the maximum number of MSDU octets that are permitted to pass the gate during the specified TimeInterval. If the IntervalOctetMax parameter is omitted, there is no limit on the number of octets that can pass the gate.

The administrative stream gate state and IPV specification parameters are used to determine the initial values of the corresponding operational parameters, and in the case of the administrative stream gate control list parameter, to provide a means of configuring a new control list prior to its installation in a running system.

Stream gates are able to permanently discard frames and thus effectively override the operational gate state (i.e., the stream gate behaves as if the operational stream gate state is Closed). This capability is provided by the GateClosedDueToInvalidRx and GateClosedDueToOctetExceed functions:

m) The GateClosedDueToInvalidRx function is enabled if the GateClosedDueToInvalidRxEnable parameter is TRUE, and disabled if this parameter is FALSE. If the function is enabled and any frame is discarded because the stream gate is in the closed state, then the GateClosedDueToInvalidRx parameter is set to TRUE, and all subsequent frames are discarded as long as the GateClosedDueToInvalidRxEnable and GateClosedDueToInvalidRx parameters are TRUE.
n) The GateClosedDueToOctetsExceeded function is enabled if the GateClosedDueToOctetsExceededEnable parameter is TRUE, and disabled if this parameter is FALSE. If the function is enabled and any frame is discarded because there are insufficient IntervalOctetsLeft (8.6.10.8), then the GateClosedDueToOctetsExceeded parameter is set to TRUE, and all subsequent frames are discarded as long as the GateClosedDueToOctetsExceededEnable and GateClosedDueToOctetsExceededEnable parameters are TRUE.

Per default, the GateClosedDueToInvalidRx and GateClosedDueToOctetExceeded functions are disabled and all associated parameters have the default value FALSE.

NOTE 2—The GateClosedDueToInvalidRx and GateClosedDueToOctetsExceeded functions allow the detection of incoming frames during time periods when the stream gate is in the closed state and exceptionally large ingress bursts to result in the stream gate behaving as it is in a permanently closed state, until such a time as management action is taken to reset the condition. The intent is to support applications where the transmission and reception of frames across the network is coordinated such that frames are received only when the stream gate is open with a limited overall amount of ingress octets. Hence, frames received by the stream gate when it is in the closed state and unexpected amounts of ingress octets represent invalid receive conditions.

8.6.5.3 Stream Filter Specifications

Stream filter specifications perform actions that can result in a frame passing or failing specific rules. Frames that fail a stream filter specification are discarded. A stream filter specification can include other actions, such as setting the drop_eligible parameter to TRUE.

The following stream filter specifications are defined:

a) Maximum SDU Size Filters (8.6.5.3.1)
b) References to Flow Meters (8.6.5.3.2)
c) References to ATS Schedulers (8.6.5.3.3)

The set of stream filter specifications of a stream filter (8.6.5.1) can contain more than one stream filter specification, in which case the processing order is as shown in Figure 8-12, which is as follows:

d) Maximum SDU size filters are executed after the stream filters and before the stream gates
e) Flow meters are executed after the stream gates and before ATS shapers

The ordering of stream filter specifications and stream gates affects the internal state variables (e.g., frame counters) of these elements. Whenever a frame is discarded by a stream filter specification or a stream gate, the frame is not visible to subsequent elements.

A stream filter specification list of a stream filter contains at most one ATS scheduler reference (i.e., a frame is never processed by multiple ATS shapers), and at most one maximum SDU size filter.

8.6.5.3.1 Maximum SDU Size Filters

Maximum SDU size filters determine whether frames are discarded that exceed a given SDU size. Each Maximum SDU size filter is associated with a dedicated stream filter (i.e., Maximum SDU size filters are not shared between different stream filter instances). A Maximum SDU size filter instance is associated with the following parameters, which are located in the Stream Filter Instance Table:

a) An integer *Maximum SDU size*, in octets.
b) A boolean *StreamBlockedDueToOversizeFrameEnable* parameter.
c) A boolean *StreamBlockedDueToOversizeFrame* parameter.

NOTE 1—Due to the one-to-one relationship between Stream Filters and Maximum SDU Size Filters (8.6.5.3), the parameters of Maximum SDU Size Filters are do not require a dedicated table and are specified as part of the Stream Filter Table instead.

If the SDU size of a frame exceeds the value of the Maximum SDU size parameter, the frame is discarded. If a frame is discarded, the NotPassingSDUCount counter of the originating stream filter is increased. If a frame passes a maximum SDU size filter, the PassingSDUCount counter of the originating stream filter is increased (8.6.5.1).

NOTE 2—The Maximum SDU size is defined per stream filter and can therefore differ from the queueMaxSDU specified in 8.6.8.4. As queueMaxSDU is applied after the flow classification and metering, it is possible that a frame that passes the Maximum SDU size filter will later be discarded because its SDU size exceeds queueMaxSDU.

Maximum SDU size filters are able to to permanently discard all frames after an initial frame has been discarded. This capability is provided by a StreamBlockedDueToOversizeFrame function. The function is enabled if the StreamBlockedDueToOversizeFrameEnable parameter is TRUE, and disabled if this parameter is FALSE. If the function is enabled and the maximum SDU size filter discards a frame, then the StreamBlockedDueToOversizeFrame parameter is set to TRUE, and all subsequent frames are discarded as long as the StreamBlockedDueToOversizeFrameEnable and StreamBlockedDueToOversizeFrame parameters are TRUE. Per default, the StreamBlockedDueToOversizeFrame function is disabled and both associated parameters have the default value FALSE.

8.6.5.3.2 Flow Meters

Flow meters in this clause implement the parameters and algorithm as specified in *Bandwidth Profile Parameters and Algorithm* in MEF 10.3 with the additions described in this clause.

Each Bridge component provides a *Flow Meter Instance Table* with parameters and variables of up to *MaxFlowMeterInstances* flow meter instances. Each flow meter instance is associated with the following parameters and variables:

a) An integer *Flow meter instance identifier.*
b) An integer *Committed information rate (CIR)*, in bits per second (MEF 10.3).
c) An integer *Committed burst size (CBS)*, in octets (MEF 10.3).
d) An integer *Excess Information Rate (EIR)*, in bits per second (MEF 10.3).
e) An integer *Excess burst size (EBS) per bandwidth profile flow*, in octets (MEF 10.3).
f) A *Coupling flag (CF)*, which takes the value 0 or 1 (MEF 10.3).
g) A *Color mode (CM)*, which takes the value *color-blind* or *color-aware* (MEF 10.3).
h) A boolean *DropOnYellow* parameter.
i) A boolean *MarkAllFramesRedEnable* parameter.
j) A boolean *MarkAllFramesRed* parameter.

NOTE 1—Envelope and Rank, as defined in MEF 10.3, are not used by the flow meters described in this clause; i.e., the reduced functionality algorithm described in 12.2 of MEF 10.3 is used.

The flow meter instance identifier uniquely identifies the flow meter instance, acts as an index to the Flow Meter Instance Table, and associates stream filter instances (8.6.5.1) with the flow meter instance.

The DropOnYellow parameter indicates whether frames marked yellow by the MEF 10.3 algorithm are discarded or marked as drop eligible:

k) A value of TRUE indicates that yellow frames are discarded.
l) A value of FALSE indicates that the drop_eligible parameter of yellow frames is set to TRUE.

NOTE 2—Changing the value of the drop_eligible parameter may change the contents of the frame, depending on how the frame is tagged when transmitted, which may then require updating the frame_check_sequence. Mechanisms for conveying information from ingress to egress that the frame_check_sequence may require updating are implementation dependent.

Flow meters are able to permanently discard all frames after an initial frame has been discarded. This capability is provided by the MarkAllFramesRed function. The function is enabled if the MarkAllFramesRedEnable parameter is TRUE, and disabled if this parameter is FALSE. If the function is enabled and the flow meter discards a frame, then the MarkAllFramesRed parameter is set to TRUE, and all subsequent frames are discarded as long as the MarkAllFramesRedEnable and MarkAllFramesRed parameters are TRUE. Per default, the MarkAllFramesRed function is disabled and both associated parameters have the default value FALSE.

Each time a flow meter discards a frame, the RedFramesCount counter of the originating stream filter (8.6.5.1) is increased.

8.6.5.3.3 ATS Schedulers

Asynchronous Traffic Shaping (ATS) Schedulers assign eligibility times to frames which are then used for traffic regulation by the ATS transmission selection algorithm (8.6.8.5).

NOTE 1—Contrary to the clause name, ATS Schedulers only realize the computational part of the overall traffic shaping operation of ATS. The complete operation is provided by the combination with the ATS transmission selection algorithm, which uses the assigned eligibility times to regulate the traffic for transmission.

ATS Schedulers are organized in *ATS Scheduler Groups*. There is one ATS scheduler group per reception Port per upstream traffic class, where the latter refers to the transmitting traffic class in the device connected to the given reception Port. All ATS scheduler instances that process frames from a particular reception Port and a particular upstream traffic class are in the respective ATS scheduler group.

NOTE 2—The organization of ATS scheduler instances into groups results in a non-decreasing ordering of eligibility times of successive frames associated with a single ATS scheduler group. This permits frames of one group to be queued in a FIFO order.

Each Bridge component provides an *ATS Scheduler Instance Table* with parameters and variables of up to *MaxSchedulerInstances* ATS scheduler instances, an *ATS Scheduler Group Instance Table* with parameters and variables of up to *MaxSchedulerGroupInstances* ATS scheduler group instances, and an *ATS Port Parameter Table* with parameters and variables shared by all ATS scheduler instances associated with a reception Port.

Each ATS Scheduler instance is associated with the following parameters and variables:

a) An integer *scheduler instance identifier*.
b) An integer *scheduler group instance identifier*.
c) An integer *CommittedBurstSizeParameter* parameter, in bits (8.6.11.2.5).
d) An integer *CommittedInformtionRate* parameter, in bits per second (8.6.11.2.6).
e) An internal *bucket empty time* state variable, in seconds (8.6.11.2.3).

Each ATS Scheduler group instance is associated with the following parameters and variables:

f) An integer *scheduler group instance identifier*.
g) An integer *MaximumResidenceTime* parameter, shared by all ATS scheduler instances in a scheduler group, in nanoseconds (8.6.11.2.13).
h) An internal *group eligibility time* state variable, shared by all ATS scheduler instances in a scheduler group, in seconds (8.6.11.2.10).

Each Port is associated with the following variable for ATS shapers:

i) An integer *DiscardedFramesCount* counter for frames that were discarded by the associated ATS scheduler instances.

The scheduler instance identifier uniquely identifies the ATS scheduler instance, acts as an index to the ATS Scheduler Instance Table, and associates stream filter instances (8.6.5.1) with a particular ATS scheduler instance. The scheduler group instance identifier uniquely identifies a scheduler group instance and establishes the relationship between ATS scheduler instances and ATS scheduler group instances.

NOTE 3—Whether ATS-scheduler instances, ATS scheduler group instances, the scheduler instance table, and the scheduler group instance table are located in reception ports or transmission ports is implementation specific (see X.1).

<<Editor's Note: The reference to X.1 is a placeholder for potential informative contents on implementation details in a later draft version>>

Each ATS scheduler instance assigns eligibility times to the associated frames, and discards frames in exceptional situations. The underlying operations are performed by an ATS scheduler state machine (8.6.11) associated with an ATS scheduler instance. This state machine updates the associated bucket empty time and group eligibility time state variables based on the CommittedBurstSize parameter, the CommittedInformationRate parameter, the MaxResidenceTime parameter, the frame arrival times, and the frame lengths (including media-specific overhead).

If an ATS scheduler instance discards a frame, the DiscardedFramesCount counter of the associated Port is increased.

8.6.5.4 Stream Filter and Gate Applications

8.6.5.4.1 Per-Stream Filtering and Policing (PSFP)

Bridges and End Stations may support Per-Stream Filtering and Policing (PSFP) capabilities that allow filtering and policing decisions, and subsequent queuing decisions (8.6.6.1), to be made for received frames. These capabilities are provided by the following elements from 8.6.5.1, 8.6.5.2, and 8.6.5.3:

a) Stream Filters, as specified in 8.6.5.1.
b) Stream Gates, as specified in 8.6.5.2.
c) Maximum SDU Size Filters, as specified in 8.6.5.3.1.
d) Flow Meters, as specified in 8.6.5.3.2.

The relationship between these elements is illustrated in Figure 8-14.

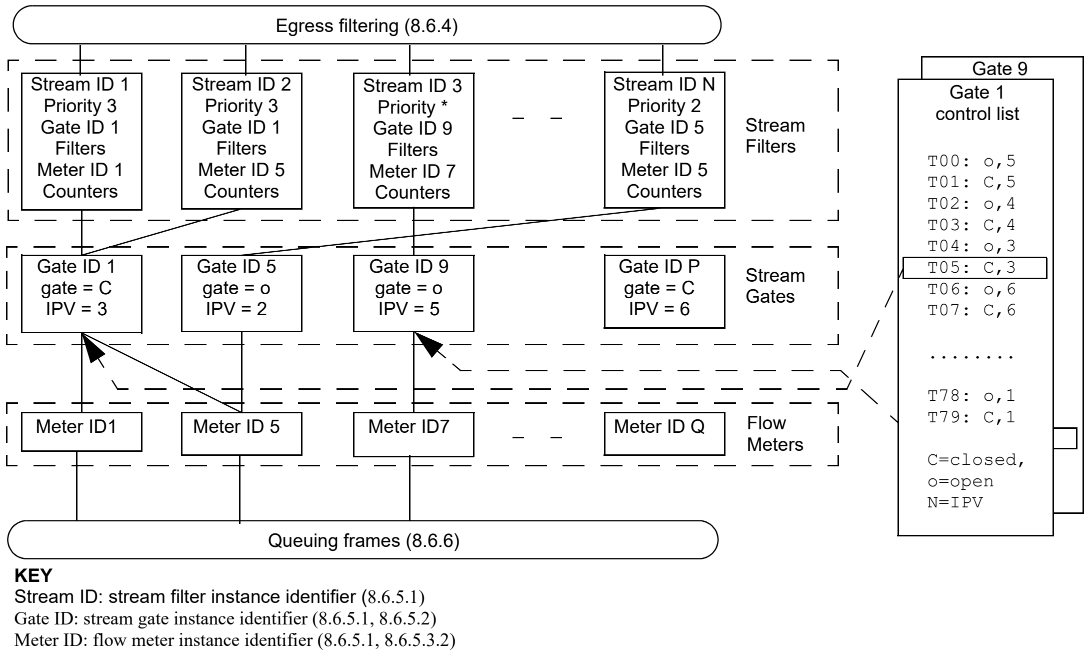

Egress filtering (8.6.4)
Stream ID 1 Priority 3 Gate ID 1 Filters Meter ID 1 Counters
Stream ID 2 Priority 3 Gate ID 1 Filters Meter ID 5 Counters
Stream ID 3 Priority * Gate ID 9 Filters Meter ID 7 Counters
Stream ID N Priority 2 Gate ID 5 Filters Meter ID 5 Counters
Stream Filters
Gate ID 1 gate = C IPV = 3
Gate ID 5 gate = o IPV = 2
Gate ID 9 gate = o IPV = 5
Gate ID P gate = C IPV = 6
Stream Gates
Meter ID1
Meter ID 5
Meter ID7
Meter ID Q
Flow Meters
Queuing frames (8.6.6)
Gate 9
Gate 1 control list
T00: o,5
T01: C,5
T02: o,4
T03: C,4
T04: o,3
T05: C,3
T06: o,6
T07: C,6
....
T78: o,1
T79: C,1
C=closed, o=open N=IPV
KEY
Stream ID: stream filter instance identifier (8.6.5.1)
Gate ID: stream gate instance identifier (8.6.5.1, 8.6.5.2)
Meter ID: flow meter instance identifier (8.6.5.1, 8.6.5.3.2)

Figure 8-14—Per-stream filtering and policing

PSFP is symmetrically implemented in bridges and end stations: All aforementioned elements are present in both. In end stations, the per-bridge component tables for stream filters (8.6.5.1), stream gates (8.6.5.2), and flow meters (8.6.5.3.2) are provided once per end station.

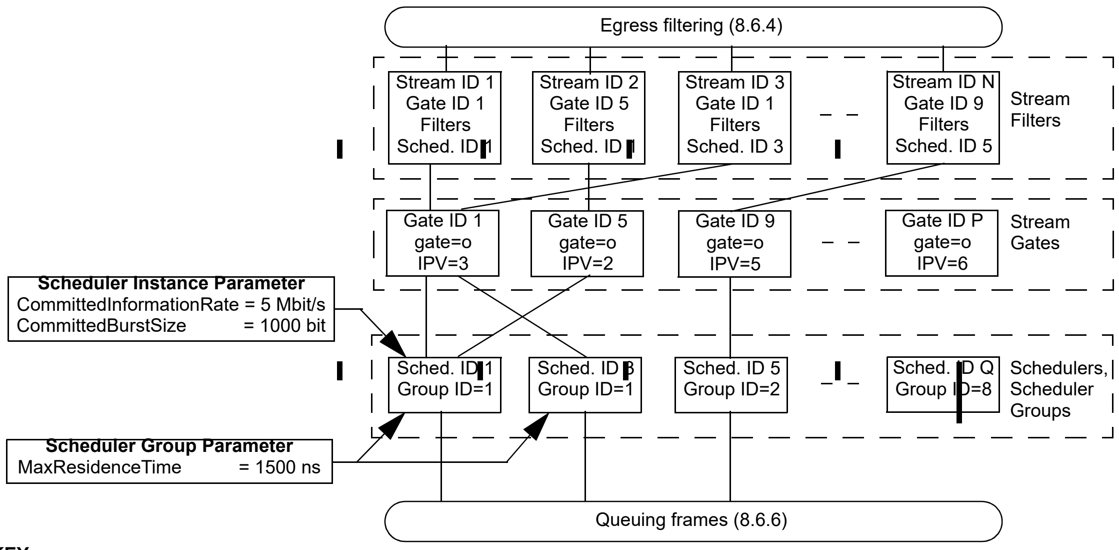
8.6.5.4.2 Asynchronous Traffic Shaping (ATS) Filtering and Assignment Functions

Bridges may support the Asynchronous Traffic Shaping (ATS) Filtering and Assignment Functions that allow filtering decisions, subsequent queuing decision (8.6.6.2), and subsequent transmission selection decisions (8.6.8.5), to be made for received frames. These functions are provided by the following elements from 8.6.5.1, 8.6.5.2, and 8.6.5.3:

a) Stream Filters, as specified in 8.6.5.1.
b) Stream Gates, as specified in 8.6.5.2, with the additional limitations stated in this clause.
c) Maximum SDU Size Filters, as specified in 8.6.5.3.1.
d) ATS Schedulers and Scheduler Groups, as specified in 8.6.5.3.3.

NOTE 1—The ATS scheduler state machine operation (8.6.11) assumes frame lengths less than or equal to the associated CommittedBurstSize parameter (8.6.11.2.5). Maximum SDU size filters can be used to avoid processing of frames that exceed the expected length.

The relationship between these elements is illustrated in Figure 8-15.



KEY
Stream ID: stream filter instance identifier (8.6.5.1)
Gate ID: stream gate instance identifier (8.6.5.1, 8.6.5.2)
Sched. ID:ATS scheduler instance identifier (8.6.5.1, 8.6.5.3.3)

Figure 8-15—ATS filtering and assignment functions

For ATS support in bridges, stream gate state transitions based on stream gate control lists (8.6.5.2) are optional. That is, it is sufficient that stream gates permanently reside either in the state open or in the state closed, and IPVs are assigned on a per-gate basis.

NOTE 2—For bridges with support for ATS, and without support for Scheduled Traffic and PSFP, stream gates of ATS traffic will never close. In this case, stream gates are permanently open and only used for IPV assignment.

NOTE 3—If the stream gate state value Closed is supported and used (e.g., for dedicated traffic classes for scheduled traffic, co-existent with dedicated traffic classes for ATS traffic), the asynchronous behavior of ATS traffic can require simultaneous stream gate state changes of multiple stream gates associated to ATS traffic.

ATS support in end stations is provided by a modified variant of the ATS Filtering and Assignment Functions, as specified in clause 49.

8.6.5.5 Flow classification and metering compatibility

Flow classification identifies a subset of traffic (frames) that may be subject to the same treatment in terms of metering and forwarding. Flow classification rules may be based on

a) Destination MAC address
b) Source MAC address
c) VID
d) Priority

Item c), specifying a VID value, is not applicable to VLAN-unaware MAC Relays.

Frames classified using the same set of classification rules are subject to the same flow meter. The flow meter can change the drop_eligible parameter associated with each frame and can discard frames on the basis of the following parameters for each received frame and previously received frames, and the time elapsed since those frames were received:

e) The received value of the drop_eligible parameter
f) The mac_service_data_unit size

The flow meter shall not base its decision on the parameters of frames received on other Bridge Ports, or on any other parameters of those Ports. The metering algorithm described in the Metro Ethernet Forum (MEF) Technical Specification 10.3 (MEF 10.3) should be used.

NOTE 1—Changing the value of the drop_eligible parameter may change the contents of the frame, depending on how the frame is tagged when transmitted, which may then require updating the frame_check_sequence. Mechanisms for conveying information from ingress to egress that the frame_check_sequence may require updating are implementation dependent.

NOTE 2—The flow meter described here can encompass a number of meters, each with a simpler specification. However, given the breadth of implementation choice permitted, further structuring to specify, for example, that frames can bypass a meter or are subject only to one of a number of meters provides no additional information.

NOTE 3—Although flow metering is applied after egress (Figure 8-11), the meter(s) operate per reception Port (see first sentence of 8.6.5), not per potential transmission Port(s).

8.6.6 Queuing frames

Change clause 8.6.6.1, as indicated:

8.6.6.1 PSFP queuing

If PSFP is supported (~~8.6.5.1~~8.6.5.4.1), and the IPV associated with the stream filter that passed the frame is anything other than the null value, then that IPV is used to determine the traffic class of the frame, in place of the frame's priority, via the Traffic Class Table specified in 8.6.6. In all other respects, the queuing actions specified in 8.6.6 are unchanged.

The IPV is used only to determine the traffic class associated with a frame, and hence select an outbound queue; for all other purposes, the received priority is used.

Insert new clause 8.6.6.2, as shown:

8.6.6.2 ATS Queuing

If ATS is supported (8.6.5.4.2), the internal priority value assigned to each frame is used to determine the traffic class of the frame.

Insert new clause 8.6.8 and Figure 8-15, as shown, re-number subsequent figures as necessary:

8.6.8 Transmission selection

Change the fourth paragraph of clause 8.6.8, as shown:

The strict priority transmission selection algorithm defined in 8.6.8.1 shall be supported by all Bridges as the default algorithm for selecting frames for transmission. The credit-based shaper transmission selection algorithm defined in 8.6.8.2, ~~and~~ the ETS algorithm defined in 8.6.8.3, and the ATS transmission selection algorithm defined in 8.6.8.5 may be supported in addition to the strict priority algorithm. Further transmission selection algorithms, selectable by management means, may be supported as an implementation option so long as the requirements of 8.6.6 are met.

Change Table 8-6, as shown:

Table 8-6—Transmission selection algorithm identifiers

Transmission selection algorithm	Identifier
Strict priority (8.6.8.1)	0
Credit-based shaper (8.6.8.2)	1
Enhanced Transmission Selection (ETS) (8.6.8.3)	2
ATS Transmission Selection (8.6.8.5)	3
Reserved for future standardization	~~3~~4–254
Vendor-specific Transmission Selection algorithm value for use with DCBX (D.2.8.8)	255
Vendor-specific	A four-octet integer, where the most significant 3 octets hold an OUI or CID value, and the least significant octet holds an integer value in the range 0–255 assigned by the owner of the OUI or CID.

Insert new clause 8.6.8.5 at the end of clause 8.6.8, as shown, re-numbering as necessary:

8.6.8.5 ATS Transmission Selection Algorithm

For a given queue that supports ATS transmission selection, the algorithm determines that a frame is available for transmission if the queue contains one or more frames eligible for transmission. A frame is eligible for transmission if the assigned eligibility time (8.6.11.2.2) is less than or equal to the current time.

The current time is determined by the TransmissionSelection Clock, which is an implementation-specific local system clock function. The TransmissionSelection Clock determines the selectability time per frame, which is the time at which this frame is queued (8.6.6) and available for transmission selection. The selectability time is used as a reference to specify the handling of device-internal implementation specific timing properties (8.6.11.2.2).

All frames that reach their selectability time are selected for transmission in ascending order of the assigned eligibility times. Transmission selection of frames with identical assigned eligibility times shall maintain the ordering requirement specified in 8.6.6.

NOTE—In the case of frames with non-identical eligibility times, the ordering requirement from 8.6.6 is automatically satisfied due the operation of the ATS scheduler state machines (8.6.12), which assign eligibility times in a non-decreasing order.

Change the contents of clause 8.6.10, as indicated:

8.6.10 Stream gate control state machines

The execution of the gate operations in a stream gate control list (~~8.6.5.1.2~~8.6.5.2) is controlled by the three state machines specified in 8.6.9:

a) The Cycle Timer state machine (8.6.9.1);
b) The List Execute state machine (8.6.9.2); and
c) The List Config state machine (8.6.9.3).

One instance of each state machine is instantiated for each stream gate control list associated with instances of stream gates in a Bridge component that supports ~~PSFP~~stream gates. An overview of the operation of these state machines can be found in Figure 8-14.

The operation of these state machines is as defined in 8.6.9, with the exception of the definitions of the ExecuteOperation() procedure, the SetGateStates() procedure, the ListPointer variable, the AdminGateStates variable, and the OperGateStates variable; amended versions of these definitions appear in 8.6.10.1 through 8.6.10.5. Table 8-8 shows the correspondence between the procedures/variables used in 8.6.9 and the ~~PSFP~~stream gate versions of these procedures/variables.

Three additional variables needed by the Execute~~PSFP~~StreamGateOperation procedure are defined in 8.6.10.6 and 8.6.10.7.

Table 8-8—Scheduled Traffic and ~~PSFP~~Stream Gate procedures/variables

Procedure/variable name in 8.6.9	~~PSFP~~Stream Gate procedure/variable name
ExecuteOperation() (8.6.9.2.1)	Execute~~PSFP~~StreamGateOperation() (8.6.10.1)
SetGateStates() (8.6.9.2.2)	Set~~PSFP~~StreamGateStates() (8.6.10.2)
ListPointer (8.6.9.4.15)	~~PSFP~~StreamGateListPointer (8.6.9.2.2)
AdminGateStates (8.6.9.4.5)	~~PSFP~~StreamGateAdminGateStates (8.6.10.4)
OperGateStates (8.6.9.4.22)	~~PSFP~~StreamGateOperGateStates (8.6.10.5)

8.6.10.1 Execute~~PSFP~~StreamGateOperation()

The Execute~~PSFP~~StreamGateOperation() procedure is responsible for fetching the next gate operation from the OperControlList, along with any parameters associated with it, and performing actions based upon the gate operation that has been fetched. The value of the ~~PSFP~~StreamGateListPointer variable (8.6.9.2.2) is used as an index into OperControlList. The procedure processes the operation according to its operation name (Table 8-4) as follows:

a) If the operation name is SetGateAndIPV, then the StreamGateState parameter value associated with the operation is assigned to the ~~PSFP~~StreamGateOperGateStates variable (8.6.10.5), the IPV parameter value is assigned to the OperIPV variable (8.6.10.7), and the TimeInterval parameter value associated with the operation is assigned to the TimeInterval variable (8.6.9.4.24). If the TimeInterval parameter value associated with the operation was 0, the TimeInterval variable is assigned the value 1. If there is an IntervalOctetMax parameter associated with the gate operation, then that parameter value is used to set the value of the IntervalOctetsLeft variable (8.6.10.8); otherwise, the IntervalOctetsLeft variable is set to a value greater than the maximum possible number of octets that the gate could pass during TimeInterval.
b) If the operation name is unrecognized, then the ~~PSFP~~StreamGateListPointer variable (8.6.9.4.15) is assigned the value of the OperControlListLength variable (8.6.9.4.23) and the TimeInterval variable (8.6.9.4.24) is assigned the value 0.
c) If there is no TimeInterval parameter associated with the operation, then the TimeInterval variable is assigned the value 0.

8.6.10.2 Set~~PSFP~~StreamGateStates()

This procedure sets the stream gate state as specified by the value of the ~~PSFP~~StreamGateOperGateStates variable (8.6.9.4.22).

8.6.10.3 ~~PSFP~~StreamGateListPointer

An integer used as a pointer to entries in the OperControlList (8.6.9.4.19), each entry consisting of a stream gate control operation with its associated parameters (Table 8-4). A value of zero points at the first entry in the list; a value of (OperControlListLength - 1) points at the last entry.

8.6.10.4 ~~PSFP~~StreamGateAdminGateStates

The initial state of the gate associated with the stream gate is set by the List Execute state machine (8.6.9.2) and is determined by the value of the ~~PSFP~~StreamGateAdminGateStates variable. The default value of ~~PSFP~~StreamGateAdminGateStates is open. The value of ~~PSFP~~StreamGateAdminGateStates can be changed by management.

8.6.10.5 ~~PSFP~~StreamGateOperGateStates

The current state of the gate associated with the stream gate. ~~PSFP~~StreamGateOperGateStates is set by the List Execute state machine (8.6.9.2), and its initial value is determined by the value of the ~~PSFP~~StreamGateAdminGateStates variable (8.6.10.4).

8.6.10.6 AdminIPV

The initial value of the OperIPV variable (8.6.10.7) associated with the stream gate is determined by the value of the AdminIPV variable. The default value of AdminIPV variable is the null value. The value of the AdminIPV variable can be changed by management.

8.6.10.7 OperIPV

The current value of the IPV associated with the stream gate. The initial value of OperIPV is set equal to the value of the AdminIPV variable (8.6.10.6). Subsequently, if there is a stream gate control list associated with the stream gate instance, its value is controlled by the contents of the operational stream gate control list and the operation of the List Execute state machine (8.6.9.2).

8.6.10.8 IntervalOctetsLeft

The current value of the IntervalOctetsLeft parameter indicates how many more MSDU octets can be passed by the stream gate during the current TimeInterval. This variable is initialized by the Execute~~PSFP~~StreamGateOperation() procedure (8.6.10.1). If a frame that would otherwise pass the gate is larger than the current value of IntervalOctetsLeft, it is treated as if the gate is in the *closed* state; i.e., it is discarded. If a frame that would otherwise pass the gate is smaller than the current value of IntervalOctetsLeft, the number of MSDU octets is subtracted from the value of IntervalOctetsLeft.

Insert the new clause 8.6.11 at the end of clause 8.6, as shown, re-numbering as necessary:

8.6.11 ATS Scheduler State Machines

The ATS scheduler state machine operation is based on the ATS scheduler clock (8.6.11.1). Each arriving frame causes invocation of the ProcessFrame(frame) procedure (8.6.11.2). Parameters contained in the ProcessFrame(frame) procedure are as defined in 8.6.5.3.3.

The ATS scheduler state machine operation is based on a token bucket shaping algorithm, as described in [B60]. This clause and its sub-clauses refer to the token bucket shaping algorithm, and the underlying terms, for informative explanation only. Specification of the ATS scheduler state machine operation does not depend on these explanations.

NOTE—A general description of the token bucket algorithm can be found in [B60].

8.6.11.1 ATS Scheduler Clock

The ATS scheduler clock is an implementation-specific local system clock function. It is used to determine the arrival time of frames (8.6.11.2.1). A Bridge component may utilize one or more ATS scheduler clock instances. In case of multiple scheduler clock instances, all ATS scheduler instances associated with the same ingress port share the same ATS scheduler clock instance (i.e., the arrival time of all frames received from a particular reception port is determined by the same ATS scheduler clock instance).

8.6.11.2 ProcessFrame(frame)

This procedure computes eligibility time, assigns the eligibility times to frames, and updates the ATS scheduler state machine variables.

The procedure is described by the following pseudo-code in a neutral manner: The arithmetic precision, and the resolution of variables, are implementation-specific, unless externally visible by management (12.31). The impact of the associated inaccuracies is discussed in X.2.

<<Editor's Note: The aforementioned reference to X.2 and 12.31 for contents related to implementation-specific inaccuracies are place holders >>

```
ProcessFrame(frame) {
      lengthRecoveryDuration     = length(frame)/
                                   CommittedInformationRate;
      emptyToFullDuration        = CommittedBurstSize/
                                   CommittedInformationRate;
      shaperEligibilityTime      = BucketEmptyTime +
                                   lengthRecoveryDuration;
      bucketFullTime             = BucketEmptyTime +
                                   emptyToFullDuration;
      eligibilityTime            = max(arrivalTime(frame),
                                       GroupEligibilityTime,
                                       shaperEligibilityTime);

      if (eligibilityTime <= (arrivalTime(frame) + MaxResidenceTime/1.0e9)){
            // The frame is valid
            GroupEligibilityTime = eligibilityTime;
            BucketEmptyTime      = (eligibilityTime < bucketFullTime) ?
                  shaperEligibilityTime :
                  shaperEligibilityTime + eligibilityTime - bucketFullTime;
            AssignAndProceed(frame,eligibilityTime);
      } else {
            // The frame is invalid
            Discard(frame);
      }
}
```

8.6.11.2.1 arrivalTime(frame)

The arrival time of the frame, in seconds. The arrival time refers to the instant of time at which a frame passes the boundary between the network physical medium and an ingress Bridge Port, as recognized by an ATS scheduler clock instance.

For all frames arriving at all ingress Bridge Ports, the arrival time is determined relative to the frame end.

NOTE—For example, the arrival time of frames may be determined based on invocation of the M_UNITDATA.indication service primitive of the ISS (6.6), as specified in IEEE Std 802.1AC.

8.6.11.2.2 AssignAndProceed(frame,eligibilityTime)

This procedure assigns an eligibility time to a frame for further processing by the transmission selection (8.6.8.5).

The assigned eligibility time, as used for ATS transmission selection decisions (8.6.8.5), is derived from the eligibilityTime parameter. The calculation of the assigned eligibility time accounts variations between the associated ATS scheduler clock instance (8.6.11.1) and the transmission selection clock (8.6.8.5), and processing delays through the forwarding process.

The ATS filtering and assignment functions are based on an ATS scheduler clock, whereas the ATS transmission selection algorithm is based on the transmission selection clock. If both are different clocks, a difference between an arbitrary instant of time t_{FA}, as recognized by the ATS scheduler clock instance, and the same instant of time t_{TS}, as recognized by the transmission selection clock, may be observed during the processing of a frame. The time difference may vary over a sequence of frames, which is characterized by

$$\text{ClockOffsetMin} \le t_{TS} - t_{FA} \le \text{ClockOffsetMax},$$

where ClockOffsetMin and ClockOffsetMax are implementation-specific constants that limit the variation.

Any frame may experience an additional, non-negative, processing delay, between its arrival time and its selectability time (8.6.8.5). This delay may vary per frame, thus that there is a delay variation over a sequence of frames. The processing delay is characterized by

$$\text{ProcessingDelayMin} \le \text{processingDelay(frame)} \le \text{ProcessingDelayMax},$$

where ProcessingDelayMin and ProcessingDelayMax are implementation-specific constants, and processingDelay(frame) denotes the processing delay of a frame, not including any potential delay introduced by the associated ATS scheduler state machine instance.

The assigned eligibility time is calculated by

$$\text{assignedEligibilityTime} = \text{eligibilityTime} + \text{ClockOffsetMax} + \text{ProcessingDelayMax}.$$

8.6.11.2.3 BucketEmptyTime

A state variable that contains the most recent instant of time at which the token bucket of the ATS scheduler instance was empty, in seconds.

The BucketEmptyTime variable is initialized with a time earlier than CommittedBurstSize/ CommittedInformationRate in the past, as perceived by the ATS Scheduler Clock. After initialization, the number of tokens in the token bucket is equivalent to the CommittedBurstSize parameter.

8.6.11.2.4 bucketFullTime

The instant of time when the number of tokens in the token bucket is equivalent to the CommittedBurstSize parameter, in seconds.

8.6.11.2.5 CommittedBurstSize

The committed burst size of the ATS scheduler instance, in bits (8.6.5.3.3). The CommittedBurstSize parameter defines the maximum token capacity of the token bucket. In the token bucket model, the number of tokens removed from the bucket by a frame equals the length of the frame, as defined in 8.6.11.2.11.

8.6.11.2.6 CommittedInformationRate

The committed information rate of the ATS scheduler instance, in bits per second (8.6.5.3.3). The CommittedInformationRate parameter defines the rate at which the token bucket is refilled with tokens until the maximum token capacity of the token bucket is reached.

8.6.11.2.7 Discard(frame)

This procedure discards the frame and increases the DiscardedFramesCount counter of the associated reception port (8.6.5.3.3). The procedure is called in exceptional situations only, e.g. misbehavior of the connected upstream system.

8.6.11.2.8 eligibilityTime

The eligibility time of the frame, without taking the implementation specific device-internal timing properties of theforwarding process into account. These timing properties are taken into account by the AssignAndProceed(frame) procedure (8.6.11.2.2).

8.6.11.2.9 emptyToFullDuration

The duration required to accumulate a number of tokens equivalent to the CommittedBurstSize parameter, in seconds.

8.6.11.2.10 GroupEligibilityTime

A state variable that contains the most recent value of the eligibilityTime variable from the previous frame, as processed by any ATS scheduler instance in the same ATS scheduler group, in seconds.

The GroupEligibilityTime variable is initialized with a time earlier or equal to the current time, as perceived by the ATS scheduler clock.

8.6.11.2.11 length(frame)

The length of the frame, in bits. This frame length includes the bits of the frame on the media that exclude transmission of a different frame. This includes media-specific framing overhead.

NOTE—For example, on IEEE Std 802.3 full-duplex point-to-point media, frame length includes Preamble, Start Frame Delimiter (SFD), Frame Check Sequence (FCS), and the interframe gap.

8.6.11.2.12 lengthRecoveryDuration

The duration required to accumulate a number of tokens equivalent to length(frame), in seconds.

8.6.11.2.13 MaxResidenceTime

The MaximumResidenceTime parameter of the ATS scheduler group instance associated with the ATS scheduler instance, in nanoseconds (8.6.5.3.3). The parameter limits the duration for which frames can reside in a bridge.

NOTE—A consistent setup of MaxResidenceTime parameter can be determined by the per hop delay bound, which is a result of the timing analysis (X.2).

8.6.11.2.14 shaperEligibilityTime

The instant of time when the number of tokens in the token bucket is at least equivalent to arrivalTime(frame), in seconds.

<<Editor’s Note: Introduction to Clause 12 in this Document

In this draft of P802.1Qcr, contents from 802.1Qci-2017, 12.31 (PSFP) and 802.1Qcr-D0.3,12.34 (ATS) were merged in 12.31. Additional slides on this restructuring are found here: http://ieee802.org/1/files/public/docs2018/cr-specht-d04-0518-v03.pdf.

>>

12. Bridge management

12.4 Bridge Management Entity

12.4.2 Port configuration

Change table Table 12-2 as indicated:

Table 12-2—Port table entry

Name	Data Type	Operations supported[a]	References
portComponentId	ComponentID	R	12.4.1.5
portPortNumber	Port Number	R	13.25
portMACAddress	MAC address	R	12.4.1.1.3 a)
portDelayExceededDiscards	counter	R	—
portMtuExceededDiscards	counter	R	—
portCapabilities	unsigned	R	—
portTypeCapabilities	unsigned	R	—
portType	enumerated	R	12.4.2.1
portExternal	Boolean	R	—
portAdminPointToPoint	unsigned	RW	IEEE Std 802.1AC
portOperPointToPoint	Boolean	R	IEEE Std 802.1AC
portName	Latin1 String (SIZE(0..32))	RW	—
portMediaDependentOverhead	unsigned	R	12.4.2.2

[a] R = Read-only access; RW = Read/Write access.

Insert clause 12.4.2.2, as shown:

12.4.2.2 Media-dependent overhead

The portMediaDepenentOverhead parameter provides the number of additional octets for media-dependent framing. The overhead includes all octets prior the first octet of the Destination Address field and all octets after the last octet of the frame check sequence.

NOTE—For example, 20 octets to account preamble, start of frame delimiter, and a nominal inter-frame gap (IFG) of 12 octets of IEEE 802.3 point-to-point media.

Change subclause 12.31 and its subclauses and tables, as indicated:

12.31 Managed objects for ~~per-stream filtering and policing~~stream filters, stream gates, and stream filter specifications

The Bridge enhancements for support of ~~per-~~stream ~~filtering~~filters, stream gates, and ~~policing~~stream filter specifications used by the applications specified in 8.6.5.4 are defined in 8.6.5.1, 8.6.5.2, and ~~the~~8.6.5.3. The associated state machines are defined in 8.6.10 and 8.6.11.

The objects that comprise this managed resource are as follows:

a) The Stream Parameter Table (12.31.1)
b) The Stream Filter Instance Table (12.31.2)
c) The Stream Gate Instance Table (12.31.3)
d) The Flow Meter Instance Table (12.31.4)
e) The Scheduler Instance Table (12.31.5)
f) The Scheduler Group Instance Table (12.31.6)
g) The Port Parameter Table (12.31.7)

12.31.1 The Stream Parameter Table

There is one Stream Parameter Table per Bridge component. The table contains a set of parameters that supports ~~PSFP(~~the applications in ~~8.6.5.1)~~8.6.5.4, as detailed in Table 12-29. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components.

12.31.1.1 MaxStreamFilterInstances

The maximum number of Stream Filter instances supported by this Bridge component (8.6.5.1).

12.31.1.2 MaxStreamGateInstances

The maximum number of Stream Gate instances supported by this Bridge component (8.6.5.2).

12.31.1.3 MaxFlowMeterInstances

The maximum number of Flow Meter instances supported by this Bridge component (8.6.5.3.2).

12.31.1.4 SupportedListMax

The maximum value supported by this Bridge component of the AdminControlListLength and OperControlListLength parameters (8.6.5.2). It is available for use by schedule computation software to determine the Bridge component's control list capacity prior to computation.

Table 12-29—The Stream Parameter Table

Name	Data type	Operations supported[a]	Conformance[b]	References
MaxStreamFilterInstances	integer	R	~~BE~~PSFP, ATS	8.6.5.1, 12.31.2
MaxStreamGateInstances	integer	R	~~BE~~PSFP, ATS	~~8.6.5.1~~8.6.5.2, 12.31.3
MaxFlowMeterInstances	integer	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.3.2, 12.31.4
SupportedListMax	integer	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.2, 12.31.3
MaxSchedulerInstances	integer	R	psfp, ATS	8.6.5.3.3, 12.31.5
MaxSchedulerGroupInstances	integer	R	psfp, ATS	8.6.5.3.3, 12.31.6

[a]R= Read only access; RW = Read/Write access.
[b]~~B = Required for Bridge or Bridge component support of PSFP.~~
~~E = Required for end-station support of PSFP.~~PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.1.5 MaxSchedulerInstances

The maximum number of ATS scheduler instances supported by this Bridge component (8.6.5.3.3).

12.31.1.6 MaxSchedulerGroupInstances

The maximum number of ATS scheduler group instances supported by this Bridge component (8.6.5.3.3).

12.31.2 The Stream Filter Instance Table

There is one Stream Filter Instance Table per Bridge component. Each table row contains a set of parameters that defines a single Stream Filter (8.6.5.1) with an associated Maximum SDU Size Filter (8.6.5.3.1), as detailed in Table 12-30. The table rows form an ordered list of filter instances, the order being determined by the StreamFilterInstance parameter. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components. Rows in the table can be created or removed dynamically in implementations that support dynamic configuration of stream filters.

12.31.2.1 StreamFilterInstance

An integer index value that determines the place of the stream filter in the ordered list of stream filter instances. The values of StreamFilterInstance are ordered according to their integer value; smaller values appear earlier in the ordered list.

Table 12-30—Stream Filter Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
StreamFilterInstance	integer	R	~~BE~~PSFP, ATS	8.6.5.1
StreamHandleSpec	stream_handle specification	RW	~~BE~~PSFP, ATS	8.6.5.1
PrioritySpec	priority specification	RW	~~BE~~PSFP, ATS	8.6.5.1
StreamGateInstanceID	integer	RW	~~BE~~PSFP, ATS	8.6.5.1, ~~8.6.5.1.2~~8.6.5.2
FilterSpecificationList	sequence of FilterSpecification	RW	~~BE~~PSFP, ATS	8.6.5.1, ~~8.6.5.1.3~~8.6.5.3, 12.31.2.5
MatchingFramesCount	counter	R	~~BE~~PSFP, ats	8.6.5.1
PassingFramesCount	counter	R	~~BE~~PSFP, ats	8.6.5.1, 8.6.5.2
NotPassingFramesCount	counter	R	~~BE~~PSFP, ats	8.6.5.1, 8.6.5.2
PassingSDUCount	counter	R	~~BE~~PSFP, ats	8.6.5.1, 8.6.5.3.1
NotPassingSDUCount	counter	R	~~BE~~PSFP, ats	8.6.5.1, 8.6.5.3.1
REDFramesCount	counter	R	~~BE~~PSFP, ats	8.6.5.1, 8.6.5.3.2
StreamBlockedDueToOversizeFrameEnable	Boolean	RW	~~BE~~PSFP, ATS	8.6.5.1, 8.6.5.3.1
StreamBlockedDueToOversizeFrame	Boolean	RW	~~BE~~PSFP, ATS	8.6.5.1, 8.6.5.3.1

[a]R= Read only access; RW = Read/Write access.
[b]~~B = Required for Bridge or Bridge component support of PSFP.~~
~~E = Required for end-station support of PSFP.~~PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.2.2 stream_handle specification data type

The stream_handle specification data type allows either of the following to be represented:

a) A stream_handle value, represented as an integer.
b) The wild card value.

12.31.2.3 priority specification data type

The priority specification data type allows either of the following to be represented:

a) A priority value, represented as an integer.
b) The wild card value.

12.31.2.4 StreamGateInstance

The StreamGateInstance parameter identifies the stream gate (12.31.3) that is associated with the stream filter. The relationship between stream filters and stream gates is many to one; a given stream filter can be associated with only one stream gate, but there can be multiple stream filters associated with a given stream gate.

12.31.2.5 FilterSpecification data type

The FilterSpecification data type can represent the following:

a) An integer value representing a Maximum SDU size (~~8.6.5.1~~8.6.5.3, 8.6.5.3.1).
b) An integer value representing a flow meter instance identifier (~~8.6.5.1, 8.6.5.1.3~~8.6.5.3, 8.6.5.3.2).
c) An integer value representing an ATS scheduler instance identifier (8.6.5.3, 8.6.5.3.3).

12.31.3 The Stream Gate Instance Table

There is one Stream Gate Instance Table per Bridge component. Each table row contains a set of parameters that defines a single Stream Gate (~~8.5.6.1.2~~8.6.5.2), as detailed in Table 12-31. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components. Rows in the table can be created or removed dynamically in implementations that support dynamic configuration of stream gates.

Table 12-31—The Stream Gate Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
StreamGateInstance	integer	R	~~BE~~PSFP,ATS	8.6.5.1, ~~8.6.5.1.2~~8.6.5.2
~~PSFP~~StreamGateEnabled	Boolean	RW	~~BE~~PSFP,ATS	8.6.9.4.14
~~PSFP~~StreamGateAdminGateStates	~~PSFPg~~StreamGateStatesValue	RW	~~BE~~PSFP,ATS	8.6.10.4, 12.29.1.2.2
~~PSFP~~StreamGateOperGateStates	~~PSFPg~~StreamGateStatesValue	R	~~BE~~PSFP,ATS	8.6.10.5, 12.29.1.2.2
~~PSFP~~StreamGateAdminControlListLength	unsigned integer	RW	~~BE~~PSFP,ats	8.6.9.4.6, 12.31.3.2
~~PSFP~~StreamGateOperControlListLength	unsigned integer	R	~~BE~~PSFP,ats	8.6.9.4.23, 12.31.3.2
~~PSFP~~StreamGateAdminControlList	sequence of ~~PSFP~~StreamGateControlEntry	RW	~~BE~~PSFP,ats	8.6.9.4.2, 12.31.3.2, 12.31.3.2.2
~~PSFP~~StreamGateOperControlList	sequence of ~~PSFP~~StreamGateControlEntry	R	~~BE~~PSFP,ats	8.6.9.4.19, 12.31.3.2, 12.31.3.2.2
~~PSFP~~StreamGateAdminCycleTime	RationalNumber	RW	~~BE~~PSFP,ats	8.6.9.4.3, 12.29.1.3
~~PSFP~~StreamGateOperCycleTime	RationalNumber (seconds)	R	~~BE~~PSFP,ats	8.6.9.4.20, 12.29.1.3
~~PSFP~~StreamGateAdminCycleTimeExtension	Integer (nanoseconds)	RW	~~BE~~PSFP,ats	8.6.9.4.4
~~PSFP~~StreamGateOperCycleTimeExtension	Integer (nanoseconds)	R	~~BE~~PSFP,ats	8.6.9.4.21
~~PSFP~~StreamGateAdminBaseTime	PTPtime	RW	~~BE~~PSFP,ats	8.6.9.4.1, 12.29.1.4
~~PSFP~~StreamGateOperBaseTime	PTPtime	R	~~BE~~PSFP,ats	8.6.9.4.18, 12.29.1.4
~~PSFP~~StreamGateConfigChange	Boolean	RW	~~BE~~PSFP,ats	8.6.9.4.7
~~PSFP~~StreamGateConfigChangeTime	PTPtime	R	~~BE~~PSFP,ats	8.6.9.4.9, 12.29.1.4
~~PSFP~~StreamGateTickGranularity	Integer (tenths of nanoseconds)	R	~~BE~~PSFP,ats	8.6.9.4.16
~~PSFP~~StreamGateCurrentTime	PTPtime	R	~~BE~~PSFP,ats	8.6.9.4.10, 12.29.1.4
~~PSFP~~StreamGateConfigPending	Boolean	R	~~BE~~PSFP,ats	8.6.9.3, 8.6.9.4.8
~~PSFP~~StreamGateConfigChangeError	Integer	R	~~BE~~PSFP,ats	8.6.9.3.1

Table 12-31—The Stream Gate Instance Table *(continued)*

Name	Data type	Operations supported[a]	Conformance[b]	References
~~PSFP~~StreamGateAdminIPV	IPV	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2, 8.6.10.6, 12.31.3.3
~~PSFP~~StreamGateOperIPV	IPV	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2, 8.6.10.7, 12.31.3.3
~~PSFP~~StreamGateClosedDueTo-InvalidRxEnable	Boolean	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2
~~PSFP~~StreamGateClosedDueToIn-validRx	Boolean	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2
~~PSFP~~StreamGateClosedDueTo-OctetsExceededEnable	Boolean	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2
~~PSFP~~StreamGateClosedDueTo-OctetsExceeded	Boolean	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2

[a]R= Read only access; RW = Read/Write access.
[b]~~B = Required for Bridge or Bridge component support of PSFP.~~
~~E = Required for end-station support of PSFP.~~PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.3.1 StreamGateInstance

An integer table index that allows the stream gate to be referenced from Stream Filter Instance Table entries.

12.31.3.2 The gate control list structure and data types

The AdminControlList and OperControlList are ordered lists containing AdminControlListLength or OperControlListLength entries, respectively. Each entry represents a gate operation as defined in Table 8-4. Each entry in the list is structured as a GateControlEntry (12.31.3.2.2).

12.31.3.2.1 ~~PSFPg~~StreamGateStatesValue

The ~~PSFPg~~StreamGateStatesValue indicates the desired gate state, *open* or *closed*, for the stream gate.

12.31.3.2.2 ~~PSFP~~StreamGateControlEntry

A ~~PSFP~~StreamGateControlEntry consists of an operation name, followed by three parameters associated with the operation, as detailed in Table 8-4. The first parameter is a ~~PSFPg~~StreamGateStatesValue (12.31.3.2.1); the second parameter is an IPV value (12.31.3.2.3), and the third parameter is a timeIntervalValue (12.31.3.2.4).

12.31.3.2.3 IPV value

The IPV value indicates the IPV (12.31.3.3) to be associated with frames that pass the gate (8.6.10.7).

12.31.3.2.4 timeIntervalValue

An unsigned integer, denoting a TimeInterval in nanoseconds (see TimeInterval in Table 8-4).

12.31.3.3 The Internal priority value specification (IPV) data type

The IPV data type represents an IPV value (~~8.6.5.1.2~~8.6.5.2); this is either the null value or an internal priority value.

12.31.3.4 Representation of times

Table 12-31 specifies times (e.g. ~~PSFP~~StreamGateAdminBaseTime) with reference to the on-the-wire timing point at which the start of a frame crosses the boundary between the physical network media and PHY. This is the message timestamp point specified by IEEE Std 802.1AS for various media.

Replace Figure 12-1, as shown

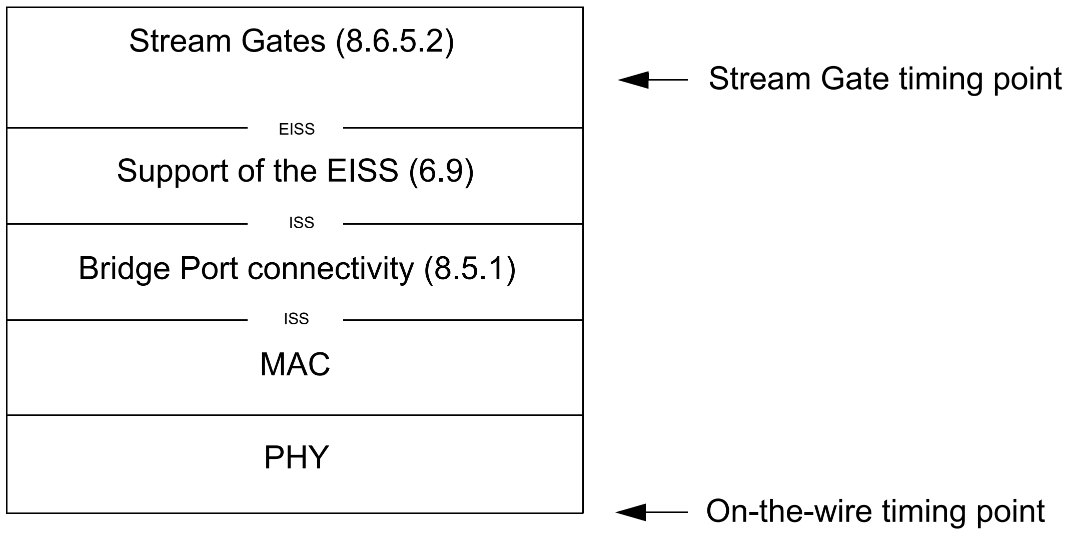


Figure 12-1—Timing points for Stream Gates

Figure 12-1 shows both the on-the-wire timing point and the ~~PSFP~~Stream Gate timing point within the interface stack (above MAC and PHY) that is used for gate open/close as described in ~~8.6.5.1~~8.6.5.2. Each timing point will have variance. A delay exists between the on-the-wire timing point and the ~~PSFP~~Stream Gate timing point. This delay will have variance that is bounded (minimum/maximum).

The Bridge contains the information needed in order to compute the minimum/maximum delay from the on-the-wire timing point to the ~~PSFP~~Stream Gate timing point. The Bridge shall adjust using the maximum delay, such that a frame's on-the-wire timing point occurs no later than the gate events represented in managed objects.

NOTE—Although the managed objects apply to a Bridge, the preceding specification of on-the-wire timing point can be applied to an end station.

12.31.4 The Flow Meter Instance Table

There is one Flow Meter Instance Table per Bridge component. Each table row contains a set of parameters that defines a single Flow Meter Instance (~~8.6.5.1~~8.6.5.3.2), as detailed in Table 12-32. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components. Rows in the table can be created or removed dynamically in implementations that support dynamic configuration of flow meters.

Table 12-32—The Flow Meter Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
FlowMeterInstanceID	integer	R	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~8.6.5.3, 8.6.5.3.2
CIR	integer, bit/s	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~8.6.5.3, 8.6.5.3.2
CBS	integer, octets	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~8.6.5.3, 8.6.5.3.2
EIR	integer, bit/s	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~8.6.5.3, 8.6.5.3.2
EBS	integer, octets	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~8.6.5.3, 8.6.5.3.2
CF	integer, 0 or 1	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~8.6.5.3, 8.6.5.3.2
CM	enumerated, color-blind or color-aware	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~8.6.5.3, 8.6.5.3.2
DropOnYellow	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~8.6.5.3, 8.6.5.3.2
MarkAllFramesRedEnable	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~8.6.5.3, 8.6.5.3.2
MarkAllFramesRed	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~8.6.5.3, 8.6.5.3.2

[a]R= Read only access; RW = Read/Write access.
[b]~~B = Required for Bridge or Bridge component support of PSFP.~~
~~E = Required for end-station support of PSFP.~~PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS..

Insert clauses 12.31.5, 12.31.6, and 12.31.7 as shown, including the contained tables:

12.31.5 The Scheduler Instance Table

There is one Scheduler Instance Table per Bridge component. Each table row in the Scheduler Instance Table comprises a set of parameters that defines a single ATS scheduler instance, as detailed in Table 12-33.

Table 12-33—The Scheduler Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
SchedulerInstanceID	integer	R	psfp,ATS	8.6.5.3.3
CommittedBurstSize	integer, bits	RW	psfp,ATS	8.6.5.3.3, 8.6.11
CommittedInformationRate	integer, bits/s	RW	psfp,ATS	8.6.5.3.3, 8.6.11
SchedulerGroupInstanceID	integer	RW	psfp,ATS	8.6.5.3.3

[a]R= Read only access; RW = Read/Write access.
[b]PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

NOTE—ATS scheduler groups establish the relationship between ATS scheduler instances and the per port management variables (12.31.7), as described in 8.6.5.3.3. As a result, the scheduler instance table does not contain references to ports.

12.31.5.1 SchedulerInstanceID

An integer table index that allows the ATS scheduler instance to be referenced from Stream Filter Instance Table entries.

12.31.5.2 CommittedBurstSize

As specified in 8.6.11.2.5.

12.31.5.3 CommittedInformationRate

As specified in 8.6.11.2.6.

12.31.5.4 SchedulerGroupInstanceID

The SchedulerGroupInstanceID parameter identifies the ATS scheduler group (12.31.6) that is associated with the ATS scheduler instance. Multiple scheduler instance can be associated to one ATS scheduler group, as detailed in 8.6.5.3.3.

12.31.6 The Scheduler Group Instance Table

There is one Scheduler Group Instance Table per Bridge component. Each table row in the Scheduler Group Instance Table comprises a set of parameters that defines a single ATS scheduler group instance (8.6.5.3.3), as detailed in Table 12-34.

Table 12-34—The Scheduler Group Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
SchedulerGroupInstanceID	integer	R	psfp,ATS	8.6.5.3.3
MaxResidenceTime	integer, nanoseconds	RW	psfp,ATS	8.6.5.3.3

[a]R= Read only access; RW = Read/Write access.
[b]PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.6.1 SchedulerGroupInstanceID

An integer table index that allows the ATS scheduler group instance to be referenced from Scheduler Instance Table entries.

12.31.6.2 MaxResidenceTime

As specified in 8.6.11.2.13.

12.31.7 The Port Parameter Table

There is one Port Parameter Table per Bridge. Each table row in the Port Parameter Table comprises a set of parameters shared by all ATS scheduler instance associated with a reception Port, as detailed in Table 12-35.

Table 12-35—The Port ParameterTable

Name	Data type	Operations supported[a]	Conformance[b]	References
PortNumber	integer	R	psfp,ATS	12.31.7.1
DiscardedFramesCount	integer	R	psfp,ATS	8.6.5.3.3, 8.6.11.2.7

[a]R= Read only access; RW = Read/Write access.

[b]PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.7.1 PortNumber

An unique index of the associated Bridge Port (12.4.2).

12.31.7.2 DiscardedFramesCount

As specified in 8.6.5.3.3 and 8.6.11.2.7.

Clause 8 and 12: Variant 2

Max. SDU size filtering in stream filters

References in 12 adjusted

8. Principles of bridge operation

<<Editor's Note: Introduction to Clause 8 in this Document

Subsequent content is based on
http://www.ieee802.org/1/files/public/docs2016/Qcr-specht-specification-mapping-proposal-0516-v01.pdf and
http://www.ieee802.org/1/files/public/docs2016/Qcr-specht-editing-1116-v01.pdf
I.e., (a) re-use procedures and tables of PSFP, (b) specify ATS functions that can be placed in reception Ports in clause 8.6.5, while (c) permitting implementation of the ATS scheduler computational part deeper in the forwarding process (i.e., in transmission ports).

In this draft of P802.1Qcr, contents of clauses 8.6.5.1 (PSFP) and 8.6.5.2 (ATS) were merged in sub-clauses 8.6.5.1 through 8.6.5.54 to simplify the descriptions/avoid redundancy. Additional slides on the restructuring are found here: http://ieee802.org/1/files/public/docs2018/cr-specht-d04-0518-v03.pdf.
- 8.6.5.1 through 8.6.5.4~~3~~ define building blocks
- 8.6.5.54 now defines which of these building blocks assemble ATS and PSFP.
The initial text in 8.6.5 from 802.1Q-Rev-2018 has been moved into a separate sub-clause (8.6.5.~~5~~6).>>

8.6 The Forwarding Process

Delete clause 8.6.5 and the contained figures and tables

Insert clause 8.6.5 after clause 8.6.4, including the figures and tables, as shown below (renumber existing figures and tables as necessary):

8.6.5 Flow classification and metering

The Forwarding Process may apply flow classification and metering to frames that are received on a Bridge Port and have one or more potential transmission ports.

Bridges and End Stations may implement the following elements to provide Flow classification and metering for applications defined in this standard:

a) Stream Filters (8.6.5.1)
b) Stream Gates (8.6.5.2)
c) ~~Stream Filter Specifications ():~~
   1) ~~Maximum SDU Size Filters ()~~
d) Flow Meters (8.6.5.3)
e) ATS Schedulers (8.6.5.4)

The relationship and ordering between these elements is illustrated in Figure 8-12.

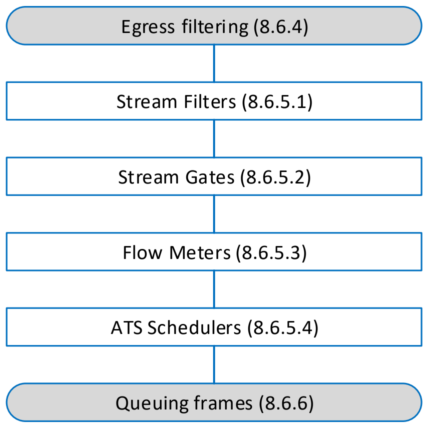


Figure 8-12—Flow classification and metering elements

Subsets of specific elements from 8.6.5.1, 8.6.5.2, ~~and~~ 8.6.5.3, and 8.6.5.4 and their use by particular applications from this standard are defined in 8.6.5.5. Applications beyond the scope of this standard might not use the elements from 8.6.5.1, 8.6.5.2, ~~and~~ 8.6.5.3, and 8.6.5.4, in which case the general compatibility requirements in 8.6.5.6 are relevant.

NOTE —Examples for applications beyond this standard include interworking with L3 DiffServ, and support for the use of Provider Bridging for Carrier Ethernet Services defined in MEF.

8.6.5.1 Stream Filters

Stream filters identify the frames of streams and associate these frames with Stream Gates (8.6.5.2), Flow Meters (8.6.5.3), and ATS Schedulers (8.6.5.4)~~and Stream Filter Specifications ()~~ for subsequent actions.

Each Bridge component provides a *Stream Filter Instance Table* with parameters and variables of up to *MaxStreamFilterInstances* stream filter instances. Each stream filter instance has the following parameters and variables:

a) An integer *stream filter instance identifier*.
b) A *stream_handle specification*.
c) A *priority specification*.
d) An integer *stream gate instance identifier*.
e) A set of zero or more *stream filter specifications*.
f) An integer *MatchingFramesCount* counter for frames matching the stream filter.
g) An integer *PassingFrameCount* counter for frames passing the associated stream gate (8.6.5.2).
h) An integer *NotPassingFrameCount* counter for frames not passing the associated stream gate (8.6.5.2).
i) A *PassingSDUCount* counter for frames that ~~passed the~~did not exceed a configured Maximum SDU size. ~~filter ().~~
j) A *NotPassingSDUCount* counter for frames that did ~~not pass~~exceed a configured~~the~~ Maximum SDU size ~~filter ()~~.
k) A *RedFramesCount* counter for frames that were discarded by the flow meter (8.6.5.3).
l) A boolean *StreamBlockedDueToOversizeFrameEnable* parameter.
m) A boolean *StreamBlockedDueToOversizeFrame* parameter.

The stream filter instance identifier uniquely identifies the filter instance, and acts as an index to the Stream Filter Instance Table.

The values of the *stream_handle* and *priority* parameters associated with a received frame determine which stream filter is selected for the frame. The stream_handle parameter is a sub-parameter of the connection_identifier parameter of the ISS (6.6), as provided by the Stream identification function in clause 6 of IEEE Std 802.1CB. A stream filter is selected if the stream_handle value of the frame matches the value of the stream_handle specification parameter and if the priority value of the frame matches the value of the priority specification parameter.

A stream_handle specification parameter can be either of the following:

n) A single stream_handle value, as specified in IEEE Std 802.1CB.
o) A wildcard value that matches any stream_handle value.

A priority specification parameter can be either of the following:

p) A single priority value.
q) A wildcard value that matches any priority value.

NOTE 1—The use of stream_handle and priority, along with the wild-carding rules previously stated, allow configuration possibilities that go beyond the selection of individual streams, as implied by the sub-clause title; for example, per-priority filtering and policing, or per-priority per-reception Port filtering and policing can be configured using these rules.

If a received frame matches zero or multiple stream filters, the behavior is as follows:

r) If no stream filter matches, the frame is processed as it would be the case without stream filters, stream gates, and stream filter specifications.
s) If more than one stream filters match, the stream filter with the smaller stream filter instance identifier is selected.

The MatchingFramesCount counter of a stream filter is increased for each frame that matches the stream_handle and priority specification parameters. If multiple stream filters match the same frame, only the MatchingFramesCount counter of the selected filter according to rule in item s) is increased.

If a stream filter is selected for a frame according to the aforementioned rules, this frame is passed to the stream gate (8.6.5.2) referred by the stream gate instance identifier parameter and the Stream Filter Specifications ~~()~~ in the set of stream filter specifications for further actions.

Stream filter specifications perform actions that can result in a frame passing or failing specific rules. Frames that fail a stream filter specification are discarded. A stream filter specification can include other actions, such as setting the drop_eligible parameter to TRUE.

The set of stream filter specifications can contain the following:

a) Integer *Maximum SDU size* parameters, in octets
b) References to Flow Meters (8.6.5.3)
c) References to ATS Schedulers (8.6.5.4)

If a maximum SDU size parameter is contained in the set of stream filter specifications, frames are discarded that exceed the given maximum SDU size. If a frame is discarded, the NotPassingSDUCount counter is increased. If a frame does not exceed the given maximum SDU size, the PassingSDUCount counter is increased.

NOTE 2—The Maximum SDU size is defined per stream filter and can therefore differ from the queueMaxSDU specified in 8.6.8.4. As queueMaxSDU is applied after the flow classification and metering, it is possible that a frame that passes the Maximum SDU size filter will later be discarded because its SDU size exceeds queueMaxSDU.

Stream Filters are able to permanently discard all frames after an initial frame has been discarded due to exceeding the Maximum SDU size. This capability is provided by a StreamBlockedDueToOversizeFrame function. The function is enabled if the StreamBlockedDueToOversizeFrameEnable parameter is TRUE, and disabled if this parameter is FALSE. If the function is enabled and the Stream Filter discards a frame, then the StreamBlockedDueToOversizeFrame parameter is set to TRUE, and all subsequent frames are discarded as long as the StreamBlockedDueToOversizeFrameEnable and StreamBlockedDueToOversizeFrame parameters are TRUE. Per default, the StreamBlockedDueToOversizeFrame function is disabled and both associated parameters have the default value FALSE.

The set of stream filter specifications of a stream filter (8.6.5.1) can contain more than one stream filter specification, in which case the processing order is as shown in Figure 8-12, which is as follows:

d) Maximum SDU size filters are executed after the stream filters and before the stream gates
e) Flow meters are executed after the stream gates and before ATS shapers

The ordering of stream filter specifications and stream gates affects the internal state variables (e.g., frame counters) of these elements. Whenever a frame is discarded by a stream filter specification or a stream gate, the frame is not visible to subsequent elements.

A stream filter specification list of a stream filter contains at most one ATS scheduler reference (i.e., a frame is never processed by multiple ATS shapers), and at most one maximum SDU size parameter.

NOTE 2—If it is desired to discard frames that do not match any other stream filter, rather than such frames being processed without filtering, this can be achieved by placing a stream filter at the end of the table, in which the stream_handle and priority are both wild-carded (set to the null value), and where the stream gate instance identifier points at a stream gate that is permanently closed.

~~The MatchingFramesCount counter of a stream filter is increased for each frame that matches the stream_handle and priority specification parameters. If multiple stream filters match the same frame, only the MatchingFramesCount counter of the selected filter according to rule in item s) is increased.~~

8.6.5.2 Stream Gates

Stream gates determine whether frames are discarded or passed for further processing by the remaining forwarding process. Stream gates are also able to change the associated traffic class of frames for later queuing decisions (8.6.6) via IPV assignments.

Each Bridge component provides a *Stream Gate Instance Table* with parameters and variables of up to *MaxStreamGateInstances* stream gate instances. Each stream gate instance is associated with the following parameters and variables:

a) An integer *stream gate instance identifier*.
b) An administrative and an operational *stream gate state* parameter.
c) An administrative and an operational *internal priority value specification*.
d) An administrative and an operational *gate control list*.
e) A boolean *GateClosedDueToInvalidRxEnable* parameter.
f) A boolean *GateClosedDueToInvalidRx* parameter.
g) A boolean *GateClosedDueToOctetsExceededEnable* parameter.
h) A boolean *GateClosedDueToOctetsExceeded* parameter.

The stream gate instance identifier uniquely identifies the stream gate instance, acts as an index to the Stream Gate and Instance Table, and associates stream filter instances (8.6.5.1) with the stream gate instance.

The relationship between stream filters, stream gates, and the associated variables and parameters of both is illustrated in Figure 8-13.

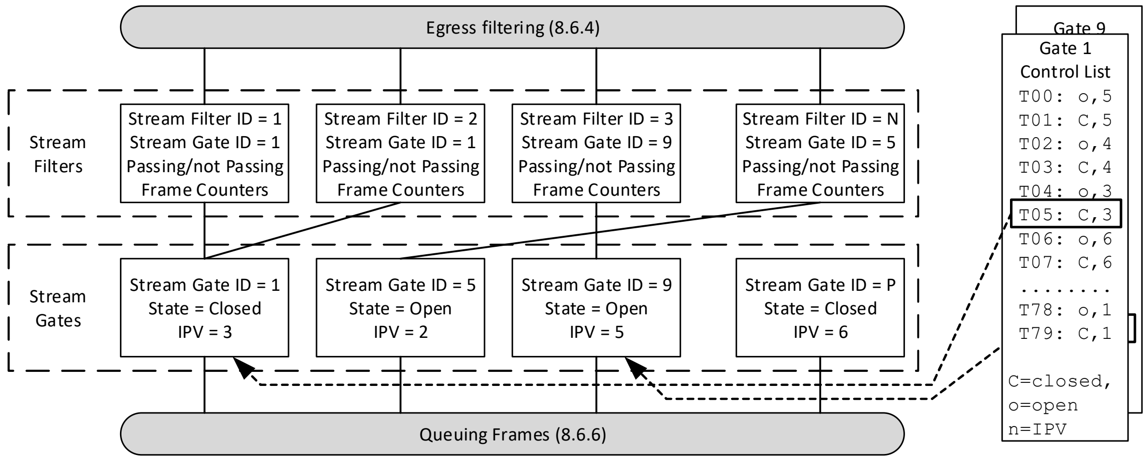


Figure 8-13—Stream Gates and Stream Filters

A stream gate can be in one of the following two states:

i) Open: Frames are permitted to pass through the stream gate.
j) Closed: Frames are not permitted to pass through the stream gate.

A frame that is permitted to pass through a stream gate is subject to subsequent actions by filter specifications () and queuing decisions (8.6.6), and the PassingFrameCount counter of the originating stream filter (8.6.5.1) is increased. A frame that is not permitted to pass through a stream gate is immediately discarded, and the NotPassingFrameCount counter of the originating stream filter is increased.

Each stream gate has an associated internal priority value specification. This specification allows to override the priority values associated with passing frames during subsequent queuing decisions (8.6.6) by assigning an *internal priority value* (IPV). An IPV specification can be either of the following:

k) A null value: The priority value associated with the frame is used to determine the frame's traffic class, using the Traffic Class Table as specified in 8.6.6. No IPV is assigned to the frame.
l) An IPV: The IPV is assigned to the frame and used, in place of the priority value associated with the frame, to determine the frame's traffic class using the Traffic Class Table as specified in 8.6.6.

An IPV assigned to a frame co-exists with the priority value associated with the frame (i.e., the IPV does not replace this priority value).

NOTE 1—The co-existence of the IPV and the priority value allows subsequent queuing decisions (8.6.6) to be based on the IPV, while the priority value is retained for transmission. A use case of IPV for ATS is the adjusting of per-hop delay bounds to satisfy end-to-end delay requirements in a specific network. Another use case for the ability to assign internal priority values can be found in Annex T (CQF).

The actual stream gate state and the IPV specification are reflected by the operational stream gate state and IPV specification parameters. Both parameters are controlled by a stream gate control state machine (8.6.10) associated with the stream gate. The state machine cyclically executes a stream gate control list that contains control operations with transitions of the operational stream gate state and IPV specification parameters, as specified in Table 8-4. The operational stream gate control list parameter reflects the actually executed list.

Table 8-4—Stream gate control operations

Operation name	Parameter(s)	Action
SetGateAndIPV	StreamGateState, IPV, TimeInterval, IntervalOctetMax	The StreamGateState parameter specifies a desired state, *open* or *closed*, for the stream gate, and the IPV parameter specifies a desired value of the IPV associated with the stream. On execution, the StreamGateState and IPV parameter values are used to set the operational values of the stream gate state and internal priority specification parameters for the stream. After *TimeInterval* ticks (8.6.9.4.16) has elapsed since the completion of the previous stream gate control operation in the stream gate control list, control passes to the next stream gate control operation. The optional IntervalOctetMax parameter specifies the maximum number of MSDU octets that are permitted to pass the gate during the specified TimeInterval. If the IntervalOctetMax parameter is omitted, there is no limit on the number of octets that can pass the gate.

The administrative stream gate state and IPV specification parameters are used to determine the initial values of the corresponding operational parameters, and in the case of the administrative stream gate control list parameter, to provide a means of configuring a new control list prior to its installation in a running system.

Stream gates are able to permanently discard frames and thus effectively override the operational gate state (i.e., the stream gate behaves as if the operational stream gate state is Closed). This capability is provided by the GateClosedDueToInvalidRx and GateClosedDueToOctetExceed functions:

m) The GateClosedDueToInvalidRx function is enabled if the GateClosedDueToInvalidRxEnable parameter is TRUE, and disabled if this parameter is FALSE. If the function is enabled and any frame is discarded because the stream gate is in the closed state, then the GateClosedDueToInvalidRx parameter is set to TRUE, and all subsequent frames are discarded as long as the GateClosedDueToInvalidRxEnable and GateClosedDueToInvalidRx parameters are TRUE.

n) The GateClosedDueToOctetsExceeded function is enabled if the GateClosedDueToOctetsExceededEnable parameter is TRUE, and disabled if this parameter is FALSE. If the function is enabled and any frame is discarded because there are insufficient IntervalOctetsLeft (8.6.10.8), then the GateClosedDueToOctetsExceeded parameter is set to TRUE, and all subsequent frames are discarded as long as the GateClosedDueToOctetsExceededEnable and GateClosedDueToOctetsExceededEnable parameters are TRUE.

Per default, the GateClosedDueToInvalidRx and GateClosedDueToOctetExceeded functions are disabled and all associated parameters have the default value FALSE.

NOTE 2—The GateClosedDueToInvalidRx and GateClosedDueToOctetsExceeded functions allow the detection of incoming frames during time periods when the stream gate is in the closed state and exceptionally large ingress bursts to result in the stream gate behaving as it is in a permanently closed state, until such a time as management action is taken to reset the condition. The intent is to support applications where the transmission and reception of frames across the network is coordinated such that frames are received only when the stream gate is open with a limited overall amount of ingress octets. Hence, frames received by the stream gate when it is in the closed state and unexpected amounts of ingress octets represent invalid receive conditions.

~~Stream Filter Specifications~~

~~Stream filter specifications perform actions that can result in a frame passing or failing specific rules. Frames that fail a stream filter specification are discarded. A stream filter specification can include other actions, such as setting the drop_eligible parameter to TRUE.~~

~~The following stream filter specifications are defined:~~

a) ~~Maximum SDU Size Filters ()~~
b) ~~References to Flow Meters (8.6.5.3)~~
c) ~~References to ATS Schedulers (8.6.5.4)~~

~~The set of stream filter specifications of a stream filter (8.6.5.1) can contain more than one stream filter specification, in which case the processing order is as shown in Figure 8-12, which is as follows:~~

d) ~~Maximum SDU size filters are executed after the stream filters and before the stream gates~~
e) ~~Flow meters are executed after the stream gates and before ATS shapers~~

~~The ordering of stream filter specifications and stream gates affects the internal state variables (e.g., frame counters) of these elements. Whenever a frame is discarded by a stream filter specification or a stream gate, the frame is not visible to subsequent elements.~~

~~A stream filter specification list of a stream filter contains at most one ATS scheduler reference (i.e., a frame is never processed by multiple ATS shapers), and at most one maximum SDU size filter.~~

~~Maximum SDU Size Filters~~

~~Maximum SDU size filters determine whether frames are discarded that exceed a given SDU size. Each Maximum SDU size filter is associated with a dedicated stream filter (i.e., Maximum SDU size filters are not shared between different stream filter instances). A Maximum SDU size filter instance is associated with the following parameters, which are located in the Stream Filter Instance Table:~~

a) ~~An integer *Maximum SDU size*, in octets.~~
b) ~~A boolean *StreamBlockedDueToOversizeFrameEnable* parameter.~~
c) ~~A boolean *StreamBlockedDueToOversizeFrame* parameter.~~

~~NOTE 1—Due to the one-to-one relationship between Stream Filters and Maximum SDU Size Filters (), the parameters of Maximum SDU Size Filters are do not require a dedicated table and are specified as part of the Stream Filter Table instead.~~

~~If the SDU size of a frame exceeds the value of the Maximum SDU size parameter, the frame is discarded. If a frame is discarded, the NotPassingSDUCount counter of the originating stream filter is increased. If a frame passes a maximum SDU size filter, the PassingSDUCount counter of the originating stream filter is increased (8.6.5.1).~~

~~NOTE 2—The Maximum SDU size is defined per stream filter and can therefore differ from the queueMaxSDU specified in 8.6.8.4. As queueMaxSDU is applied after the flow classification and metering, it is possible that a frame that passes the Maximum SDU size filter will later be discarded because its SDU size exceeds queueMaxSDU.~~

~~Maximum SDU size filters are able to to permanently discard all frames after an initial frame has been discarded. This capability is provided by a StreamBlockedDueToOversizeFrame function. The function is enabled if the StreamBlockedDueToOversizeFrameEnable parameter is TRUE, and disabled if this parameter is FALSE. If the function is enabled and the maximum SDU size filter discards a frame, then the StreamBlockedDueToOversizeFrame parameter is set to TRUE, and all subsequent frames are discarded as long as the StreamBlockedDueToOversizeFrameEnable and StreamBlockedDueToOversizeFrame parameters are TRUE. Per default, the StreamBlockedDueToOversizeFrame function is disabled and both associated parameters have the default value FALSE.~~

8.6.5.3 Flow Meters

Flow meters in this clause implement the parameters and algorithm as specified in *Bandwidth Profile Parameters and Algorithm* in MEF 10.3 with the additions described in this clause.

Each Bridge component provides a *Flow Meter Instance Table* with parameters and variables of up to *MaxFlowMeterInstances* flow meter instances. Each flow meter instance is associated with the following parameters and variables:

a) An integer *Flow meter instance identifier.*
b) An integer *Committed information rate (CIR)*, in bits per second (MEF 10.3).
c) An integer *Committed burst size (CBS)*, in octets (MEF 10.3).
d) An integer *Excess Information Rate (EIR)*, in bits per second (MEF 10.3).
e) An integer *Excess burst size (EBS) per bandwidth profile flow*, in octets (MEF 10.3).
f) A *Coupling flag (CF)*, which takes the value 0 or 1 (MEF 10.3).
g) A *Color mode (CM)*, which takes the value *color-blind* or *color-aware* (MEF 10.3).
h) A boolean *DropOnYellow* parameter.
i) A boolean *MarkAllFramesRedEnable* parameter.
j) A boolean *MarkAllFramesRed* parameter.

NOTE 1—Envelope and Rank, as defined in MEF 10.3, are not used by the flow meters described in this clause; i.e., the reduced functionality algorithm described in 12.2 of MEF 10.3 is used.

The flow meter instance identifier uniquely identifies the flow meter instance, acts as an index to the Flow Meter Instance Table, and associates stream filter instances (8.6.5.1) with the flow meter instance.

The DropOnYellow parameter indicates whether frames marked yellow by the MEF 10.3 algorithm are discarded or marked as drop eligible:

k) A value of TRUE indicates that yellow frames are discarded.
l) A value of FALSE indicates that the drop_eligible parameter of yellow frames is set to TRUE.

NOTE 2—Changing the value of the drop_eligible parameter may change the contents of the frame, depending on how the frame is tagged when transmitted, which may then require updating the frame_check_sequence. Mechanisms for conveying information from ingress to egress that the frame_check_sequence may require updating are implementation dependent.

Flow meters are able to permanently discard all frames after an initial frame has been discarded. This capability is provided by the MarkAllFramesRed function. The function is enabled if the MarkAllFramesRedEnable parameter is TRUE, and disabled if this parameter is FALSE. If the function is enabled and the flow meter discards a frame, then the MarkAllFramesRed parameter is set to TRUE, and all subsequent frames are discarded as long as the MarkAllFramesRedEnable and MarkAllFramesRed parameters are TRUE. Per default, the MarkAllFramesRed function is disabled and both associated parameters have the default value FALSE.

Each time a flow meter discards a frame, the RedFramesCount counter of the originating stream filter (8.6.5.1) is increased.

8.6.5.4 ATS Schedulers

Asynchronous Traffic Shaping (ATS) Schedulers assign eligibility times to frames which are then used for traffic regulation by the ATS transmission selection algorithm (8.6.8.5).

NOTE 1—Contrary to the clause name, ATS Schedulers only realize the computational part of the overall traffic shaping operation of ATS. The complete operation is provided by the combination with the ATS transmission selection algorithm, which uses the assigned eligibility times to regulate the traffic for transmission.

ATS Schedulers are organized in *ATS Scheduler Groups*. There is one ATS scheduler group per reception Port per upstream traffic class, where the latter refers to the transmitting traffic class in the device connected to the given reception Port. All ATS scheduler instances that process frames from a particular reception Port and a particular upstream traffic class are in the respective ATS scheduler group.

NOTE 2—The organization of ATS scheduler instances into groups results in a non-decreasing ordering of eligibility times of successive frames associated with a single ATS scheduler group. This permits frames of one group to be queued in a FIFO order.

Each Bridge component provides an *ATS Scheduler Instance Table* with parameters and variables of up to *MaxSchedulerInstances* ATS scheduler instances, an *ATS Scheduler Group Instance Table* with parameters and variables of up to *MaxSchedulerGroupInstances* ATS scheduler group instances, and an *ATS Port Parameter Table* with parameters and variables shared by all ATS scheduler instances associated with a reception Port.

Each ATS Scheduler instance is associated with the following parameters and variables:

a) An integer *scheduler instance identifier*.
b) An integer *scheduler group instance identifier*.
c) An integer *CommittedBurstSizeParameter* parameter, in bits (8.6.11.2.5).
d) An integer *CommittedInformtionRate* parameter, in bits per second (8.6.11.2.6).
e) An internal *bucket empty time* state variable, in seconds (8.6.11.2.3).

Each ATS Scheduler group instance is associated with the following parameters and variables:

f) An integer *scheduler group instance identifier*.
g) An integer *MaximumResidenceTime* parameter, shared by all ATS scheduler instances in a scheduler group, in nanoseconds (8.6.11.2.13).
h) An internal *group eligibility time* state variable, shared by all ATS scheduler instances in a scheduler group, in seconds (8.6.11.2.10).

Each Port is associated with the following variable for ATS shapers:

i) An integer *DiscardedFramesCount* counter for frames that were discarded by the associated ATS scheduler instances.

The scheduler instance identifier uniquely identifies the ATS scheduler instance, acts as an index to the ATS Scheduler Instance Table, and associates stream filter instances (8.6.5.1) with a particular ATS scheduler instance. The scheduler group instance identifier uniquely identifies a scheduler group instance and establishes the relationship between ATS scheduler instances and ATS scheduler group instances.

NOTE 3—Whether ATS-scheduler instances, ATS scheduler group instances, the scheduler instance table, and the scheduler group instance table are located in reception ports or transmission ports is implementation specific (see X.1).

<<Editor’s Note: The reference to X.1 is a placeholder for potential informative contents on implementation details in a later draft version>>

Each ATS scheduler instance assigns eligibility times to the associated frames, and discards frames in exceptional situations. The underlying operations are performed by an ATS scheduler state machine (8.6.11) associated with an ATS scheduler instance. This state machine updates the associated bucket empty time and group eligibility time state variables based on the CommittedBurstSize parameter, the

CommittedInformationRate parameter, the MaxResidenceTime parameter, the frame arrival times, and the frame lengths (including media-specific overhead).

If an ATS scheduler instance discards a frame, the DiscardedFramesCount counter of the associated Port is increased.

8.6.5.5 Stream Filter and Gate Applications

8.6.5.5.1 Per-Stream Filtering and Policing (PSFP)

Bridges and End Stations may support Per-Stream Filtering and Policing (PSFP) capabilities that allow filtering and policing decisions, and subsequent queuing decisions (8.6.6.1), to be made for received frames. These capabilities are provided by the following elements from 8.6.5.1, 8.6.5.2, and :

a) Stream Filters, as specified in 8.6.5.1.
b) Stream Gates, as specified in 8.6.5.2.
c) Maximum SDU Size Filters, as specified in .
d) Flow Meters, as specified in 8.6.5.3.

The relationship between these elements is illustrated in Figure 8-14.

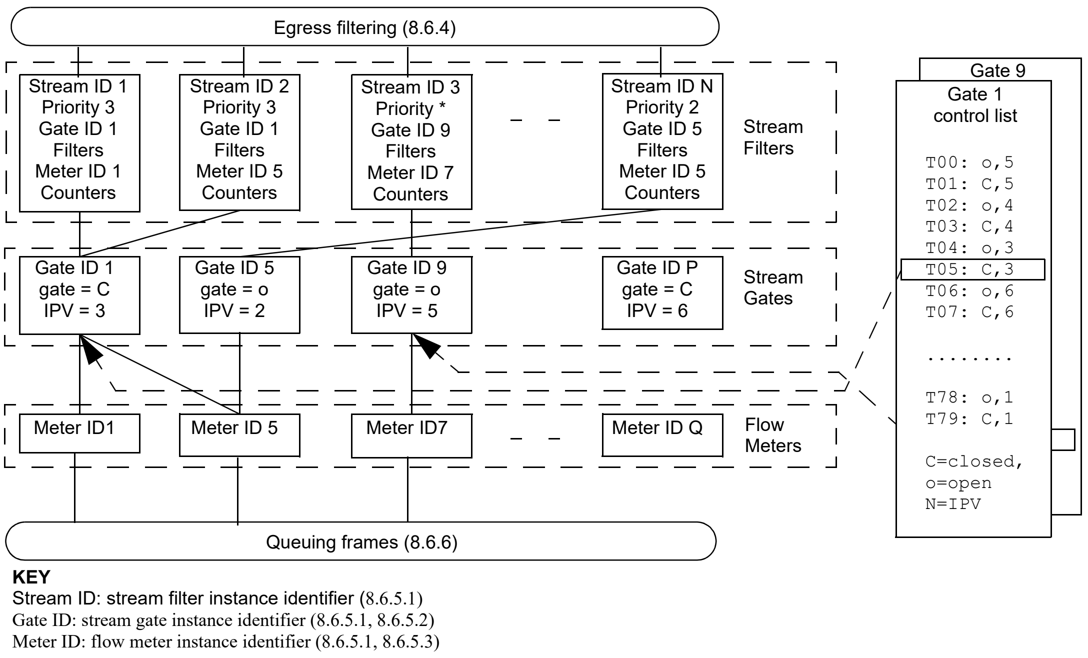


Figure 8-14—Per-stream filtering and policing

PSFP is symmetrically implemented in bridges and end stations: All aforementioned elements are present in both. In end stations, the per-bridge component tables for stream filters (8.6.5.1), stream gates (8.6.5.2), and flow meters (8.6.5.3) are provided once per end station.

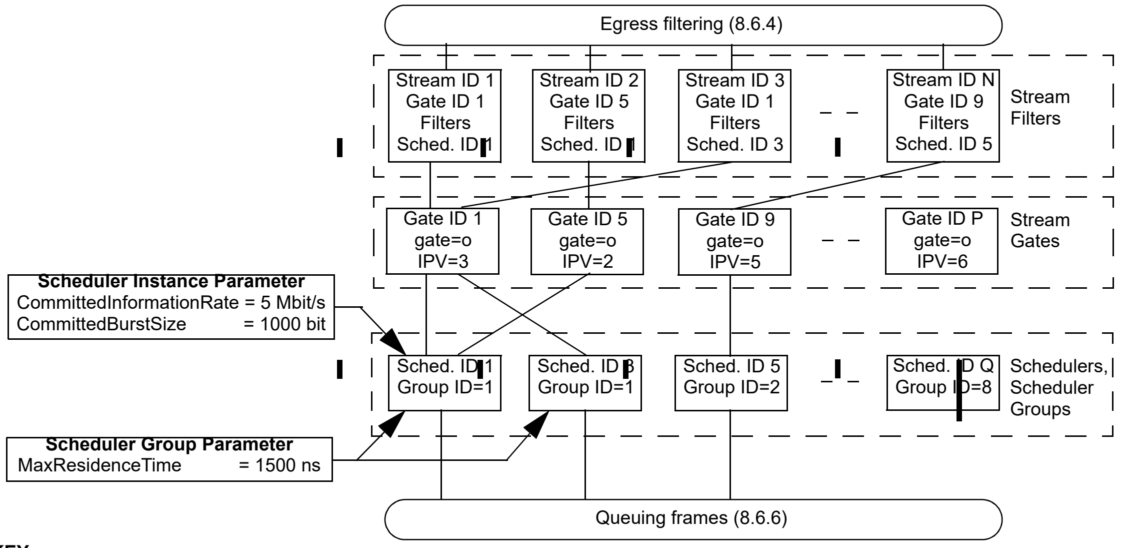
8.6.5.5.2 Asynchronous Traffic Shaping (ATS) Filtering and Assignment Functions

Bridges may support the Asynchronous Traffic Shaping (ATS) Filtering and Assignment Functions that allow filtering decisions, subsequent queuing decision (8.6.6.2), and subsequent transmission selection decisions (8.6.8.5), to be made for received frames. These functions are provided by the following elements from 8.6.5.1, 8.6.5.2, and :

a) Stream Filters, as specified in 8.6.5.1.
b) Stream Gates, as specified in 8.6.5.2, with the additional limitations stated in this clause.
c) Maximum SDU Size Filters, as specified in .
d) ATS Schedulers and Scheduler Groups, as specified in 8.6.5.4.

NOTE 1—The ATS scheduler state machine operation (8.6.11) assumes frame lengths less than or equal to the associated CommittedBurstSize parameter (8.6.11.2.5). Maximum SDU size filters can be used to avoid processing of frames that exceed the expected length.

The relationship between these elements is illustrated in Figure 8-15.


Egress filtering (8.6.4)
Stream ID 1 Gate ID 1 Filters Sched. ID 1
Stream ID 2 Gate ID 5 Filters Sched. ID 1
Stream ID 3 Gate ID 1 Filters Sched. ID 3
Stream ID N Gate ID 9 Filters Sched. ID 5
Stream Filters
Gate ID 1 gate=o IPV=3
Gate ID 5 gate=o IPV=2
Gate ID 9 gate=o IPV=5
Gate ID P gate=o IPV=6
Stream Gates
Scheduler Instance Parameter
CommittedInformationRate = 5 Mbit/s
CommittedBurstSize = 1000 bit
Sched. ID 1 Group ID=1
Sched. ID 3 Group ID=1
Sched. ID 5 Group ID=2
Sched. ID Q Group ID=8
Schedulers, Scheduler Groups
Scheduler Group Parameter
MaxResidenceTime = 1500 ns
Queuing frames (8.6.6)

KEY
Stream ID: stream filter instance identifier (8.6.5.1)
Gate ID: stream gate instance identifier (8.6.5.1, 8.6.5.2)
Sched. ID:ATS scheduler instance identifier (8.6.5.1, 8.6.5.4)

Figure 8-15—ATS filtering and assignment functions

For ATS support in bridges, stream gate state transitions based on stream gate control lists (8.6.5.2) are optional. That is, it is sufficient that stream gates permanently reside either in the state open or in the state closed, and IPVs are assigned on a per-gate basis.

NOTE 2—For bridges with support for ATS, and without support for Scheduled Traffic and PSFP, stream gates of ATS traffic will never close. In this case, stream gates are permanently open and only used for IPV assignment.

NOTE 3—If the stream gate state value Closed is supported and used (e.g., for dedicated traffic classes for scheduled traffic, co-existent with dedicated traffic classes for ATS traffic), the asynchronous behavior of ATS traffic can require simultaneous stream gate state changes of multiple stream gates associated to ATS traffic.

ATS support in end stations is provided by a modified variant of the ATS Filtering and Assignment Functions, as specified in clause 49.

8.6.5.6 Flow classification and metering compatibility

Flow classification identifies a subset of traffic (frames) that may be subject to the same treatment in terms of metering and forwarding. Flow classification rules may be based on

a) Destination MAC address
b) Source MAC address
c) VID
d) Priority

Item c), specifying a VID value, is not applicable to VLAN-unaware MAC Relays.

Frames classified using the same set of classification rules are subject to the same flow meter. The flow meter can change the drop_eligible parameter associated with each frame and can discard frames on the basis of the following parameters for each received frame and previously received frames, and the time elapsed since those frames were received:

e) The received value of the drop_eligible parameter
f) The mac_service_data_unit size

The flow meter shall not base its decision on the parameters of frames received on other Bridge Ports, or on any other parameters of those Ports. The metering algorithm described in the Metro Ethernet Forum (MEF) Technical Specification 10.3 (MEF 10.3) should be used.

NOTE 1—Changing the value of the drop_eligible parameter may change the contents of the frame, depending on how the frame is tagged when transmitted, which may then require updating the frame_check_sequence. Mechanisms for conveying information from ingress to egress that the frame_check_sequence may require updating are implementation dependent.

NOTE 2—The flow meter described here can encompass a number of meters, each with a simpler specification. However, given the breadth of implementation choice permitted, further structuring to specify, for example, that frames can bypass a meter or are subject only to one of a number of meters provides no additional information.

NOTE 3—Although flow metering is applied after egress (Figure 8-11), the meter(s) operate per reception Port (see first sentence of 8.6.5), not per potential transmission Port(s).

8.6.6 Queuing frames

Change clause 8.6.6.1, as indicated:

8.6.6.1 PSFP queuing

If PSFP is supported (~~8.6.5.1~~8.6.5.5.1), and the IPV associated with the stream filter that passed the frame is anything other than the null value, then that IPV is used to determine the traffic class of the frame, in place of the frame's priority, via the Traffic Class Table specified in 8.6.6. In all other respects, the queuing actions specified in 8.6.6 are unchanged.

The IPV is used only to determine the traffic class associated with a frame, and hence select an outbound queue; for all other purposes, the received priority is used.

Insert new clause 8.6.6.2, as shown:

8.6.6.2 ATS Queuing

If ATS is supported (8.6.5.5.2), the internal priority value assigned to each frame is used to determine the traffic class of the frame.

Insert new clause 8.6.8 and Figure 8-15, as shown, re-number subsequent figures as necessary:

8.6.8 Transmission selection

Change the fourth paragraph of clause 8.6.8, as shown:

The strict priority transmission selection algorithm defined in 8.6.8.1 shall be supported by all Bridges as the default algorithm for selecting frames for transmission. The credit-based shaper transmission selection algorithm defined in 8.6.8.2, ~~and~~ the ETS algorithm defined in 8.6.8.3, and the ATS transmission selection algorithm defined in 8.6.8.5 may be supported in addition to the strict priority algorithm. Further transmission selection algorithms, selectable by management means, may be supported as an implementation option so long as the requirements of 8.6.6 are met.

Change Table 8-6, as shown:

Table 8-6—Transmission selection algorithm identifiers

Transmission selection algorithm	Identifier
Strict priority (8.6.8.1)	0
Credit-based shaper (8.6.8.2)	1
Enhanced Transmission Selection (ETS) (8.6.8.3)	2
ATS Transmission Selection (8.6.8.5)	3
Reserved for future standardization	~~3~~4–254
Vendor-specific Transmission Selection algorithm value for use with DCBX (D.2.8.8)	255
Vendor-specific	A four-octet integer, where the most significant 3 octets hold an OUI or CID value, and the least significant octet holds an integer value in the range 0–255 assigned by the owner of the OUI or CID.

Insert new clause 8.6.8.5 at the end of clause 8.6.8, as shown, re-numbering as necessary:

8.6.8.5 ATS Transmission Selection Algorithm

For a given queue that supports ATS transmission selection, the algorithm determines that a frame is available for transmission if the queue contains one or more frames eligible for transmission. A frame is eligible for transmission if the assigned eligibility time (8.6.11.2.2) is less than or equal to the current time.

The current time is determined by the TransmissionSelection Clock, which is an implementation-specific local system clock function. The TransmissionSelection Clock determines the selectability time per frame, which is the time at which this frame is queued (8.6.6) and available for transmission selection. The selectability time is used as a reference to specify the handling of device-internal implementation specific timing properties (8.6.11.2.2).

All frames that reach their selectability time are selected for transmission in ascending order of the assigned eligibility times. Transmission selection of frames with identical assigned eligibility times shall maintain the ordering requirement specified in 8.6.6.

NOTE—In the case of frames with non-identical eligibility times, the ordering requirement from 8.6.6 is automatically satisfied due the operation of the ATS scheduler state machines (8.6.12), which assign eligibility times in a non-decreasing order.

Change the contents of clause 8.6.10, as indicated:

8.6.10 Stream gate control state machines

The execution of the gate operations in a stream gate control list (~~8.6.5.1.2~~8.6.5.2) is controlled by the three state machines specified in 8.6.9:

a) The Cycle Timer state machine (8.6.9.1);
b) The List Execute state machine (8.6.9.2); and
c) The List Config state machine (8.6.9.3).

One instance of each state machine is instantiated for each stream gate control list associated with instances of stream gates in a Bridge component that supports ~~PSFP~~stream gates. An overview of the operation of these state machines can be found in Figure 8-14.

The operation of these state machines is as defined in 8.6.9, with the exception of the definitions of the ExecuteOperation() procedure, the SetGateStates() procedure, the ListPointer variable, the AdminGateStates variable, and the OperGateStates variable; amended versions of these definitions appear in 8.6.10.1 through 8.6.10.5. Table 8-8 shows the correspondence between the procedures/variables used in 8.6.9 and the ~~PSFP~~stream gate versions of these procedures/variables.

Three additional variables needed by the Execute~~PSFP~~StreamGateOperation procedure are defined in 8.6.10.6 and 8.6.10.7.

Table 8-8—Scheduled Traffic and ~~PSFP~~Stream Gate procedures/variables

Procedure/variable name in 8.6.9	~~PSFP~~Stream Gate **procedure/variable name**
ExecuteOperation() (8.6.9.2.1)	Execute~~PSFP~~StreamGateOperation() (8.6.10.1)
SetGateStates() (8.6.9.2.2)	Set~~PSFP~~StreamGateStates() (8.6.10.2)
ListPointer (8.6.9.4.15)	~~PSFP~~StreamGateListPointer (8.6.9.2.2)
AdminGateStates (8.6.9.4.5)	~~PSFP~~StreamGateAdminGateStates (8.6.10.4)
OperGateStates (8.6.9.4.22)	~~PSFP~~StreamGateOperGateStates (8.6.10.5)

8.6.10.1 Execute~~PSFP~~StreamGateOperation()

The Execute~~PSFP~~StreamGateOperation() procedure is responsible for fetching the next gate operation from the OperControlList, along with any parameters associated with it, and performing actions based upon the gate operation that has been fetched. The value of the ~~PSFP~~StreamGateListPointer variable (8.6.9.2.2) is used as an index into OperControlList. The procedure processes the operation according to its operation name (Table 8-4) as follows:

a) If the operation name is SetGateAndIPV, then the StreamGateState parameter value associated with the operation is assigned to the ~~PSFP~~StreamGateOperGateStates variable (8.6.10.5), the IPV parameter value is assigned to the OperIPV variable (8.6.10.7), and the TimeInterval parameter value associated with the operation is assigned to the TimeInterval variable (8.6.9.4.24). If the

TimeInterval parameter value associated with the operation was 0, the TimeInterval variable is assigned the value 1. If there is an IntervalOctetMax parameter associated with the gate operation, then that parameter value is used to set the value of the IntervalOctetsLeft variable (8.6.10.8); otherwise, the IntervalOctetsLeft variable is set to a value greater than the maximum possible number of octets that the gate could pass during TimeInterval.

b) If the operation name is unrecognized, then the ~~PSFP~~StreamGateListPointer variable (8.6.9.4.15) is assigned the value of the OperControlListLength variable (8.6.9.4.23) and the TimeInterval variable (8.6.9.4.24) is assigned the value 0.

c) If there is no TimeInterval parameter associated with the operation, then the TimeInterval variable is assigned the value 0.

8.6.10.2 Set~~PSFP~~StreamGateStates()

This procedure sets the stream gate state as specified by the value of the ~~PSFP~~StreamGateOperGateStates variable (8.6.9.4.22).

8.6.10.3 ~~PSFP~~StreamGateListPointer

An integer used as a pointer to entries in the OperControlList (8.6.9.4.19), each entry consisting of a stream gate control operation with its associated parameters (Table 8-4). A value of zero points at the first entry in the list; a value of (OperControlListLength - 1) points at the last entry.

8.6.10.4 ~~PSFP~~StreamGateAdminGateStates

The initial state of the gate associated with the stream gate is set by the List Execute state machine (8.6.9.2) and is determined by the value of the ~~PSFP~~StreamGateAdminGateStates variable. The default value of ~~PSFP~~StreamGateAdminGateStates is open. The value of ~~PSFP~~StreamGateAdminGateStates can be changed by management.

8.6.10.5 ~~PSFP~~StreamGateOperGateStates

The current state of the gate associated with the stream gate. ~~PSFP~~StreamGateOperGateStates is set by the List Execute state machine (8.6.9.2), and its initial value is determined by the value of the ~~PSFP~~StreamGateAdminGateStates variable (8.6.10.4).

8.6.10.6 AdminIPV

The initial value of the OperIPV variable (8.6.10.7) associated with the stream gate is determined by the value of the AdminIPV variable. The default value of AdminIPV variable is the null value. The value of the AdminIPV variable can be changed by management.

8.6.10.7 OperIPV

The current value of the IPV associated with the stream gate. The initial value of OperIPV is set equal to the value of the AdminIPV variable (8.6.10.6). Subsequently, if there is a stream gate control list associated with the stream gate instance, its value is controlled by the contents of the operational stream gate control list and the operation of the List Execute state machine (8.6.9.2).

8.6.10.8 IntervalOctetsLeft

The current value of the IntervalOctetsLeft parameter indicates how many more MSDU octets can be passed by the stream gate during the current TimeInterval. This variable is initialized by the Execute~~PSFP~~StreamGateOperation() procedure (8.6.10.1). If a frame that would otherwise pass the gate is larger than the current value of IntervalOctetsLeft, it is treated as if the gate is in the *closed* state; i.e., it is discarded. If a frame that would otherwise pass the gate is smaller than the current value of IntervalOctetsLeft, the number of MSDU octets is subtracted from the value of IntervalOctetsLeft.

Insert the new clause 8.6.11 at the end of clause 8.6, as shown, re-numbering as necessary:

8.6.11 ATS Scheduler State Machines

The ATS scheduler state machine operation is based on the ATS scheduler clock (8.6.11.1). Each arriving frame causes invocation of the ProcessFrame(frame) procedure (8.6.11.2). Parameters contained in the ProcessFrame(frame) procedure are as defined in 8.6.5.4.

The ATS scheduler state machine operation is based on a token bucket shaping algorithm, as described in [B60]. This clause and its sub-clauses refer to the token bucket shaping algorithm, and the underlying terms, for informative explanation only. Specification of the ATS scheduler state machine operation does not depend on these explanations.

NOTE—A general description of the token bucket algorithm can be found in [B60].

8.6.11.1 ATS Scheduler Clock

The ATS scheduler clock is an implementation-specific local system clock function. It is used to determine the arrival time of frames (8.6.11.2.1). A Bridge component may utilize one or more ATS scheduler clock instances. In case of multiple scheduler clock instances, all ATS scheduler instances associated with the same ingress port share the same ATS scheduler clock instance (i.e., the arrival time of all frames received from a particular reception port is determined by the same ATS scheduler clock instance).

8.6.11.2 ProcessFrame(frame)

This procedure computes eligibility time, assigns the eligibility times to frames, and updates the ATS scheduler state machine variables.

The procedure is described by the following pseudo-code in a neutral manner: The arithmetic precision, and the resolution of variables, are implementation-specific, unless externally visible by management (12.31). The impact of the associated inaccuracies is discussed in X.2.

<<Editor's Note: The aforementioned reference to X.2 and 12.31 for contents related to implementation-specific inaccuracies are place holders >>

```
ProcessFrame(frame) {
      lengthRecoveryDuration     = length(frame)/
                                   CommittedInformationRate;
      emptyToFullDuration        = CommittedBurstSize/
                                   CommittedInformationRate;
      shaperEligibilityTime      = BucketEmptyTime +
                                   lengthRecoveryDuration;
      bucketFullTime             = BucketEmptyTime +
                                   emptyToFullDuration;
      eligibilityTime            = max(arrivalTime(frame),
                                       GroupEligibilityTime,
                                       shaperEligibilityTime);

      if (eligibilityTime <= (arrivalTime(frame) + MaxResidenceTime/1.0e9)){
            // The frame is valid
            GroupEligibilityTime = eligibilityTime;
            BucketEmptyTime      = (eligibilityTime < bucketFullTime) ?
                  shaperEligibilityTime :
                    shaperEligibilityTime + eligibilityTime - bucketFullTime;
                AssignAndProceed(frame,eligibilityTime);
        } else {
                // The frame is invalid
                Discard(frame);
        }
}
```

8.6.11.2.1 arrivalTime(frame)

The arrival time of the frame, in seconds. The arrival time refers to the instant of time at which a frame passes the boundary between the network physical medium and an ingress Bridge Port, as recognized by an ATS scheduler clock instance.

For all frames arriving at all ingress Bridge Ports, the arrival time is determined relative to the frame end.

NOTE—For example, the arrival time of frames may be determined based on invocation of the M_UNITDATA.indication service primitive of the ISS (6.6), as specified in IEEE Std 802.1AC.

8.6.11.2.2 AssignAndProceed(frame,eligibilityTime)

This procedure assigns an eligibility time to a frame for further processing by the transmission selection (8.6.8.5).

The assigned eligibility time, as used for ATS transmission selection decisions (8.6.8.5), is derived from the eligibilityTime parameter. The calculation of the assigned eligibility time accounts variations between the associated ATS scheduler clock instance (8.6.11.1) and the transmission selection clock (8.6.8.5), and processing delays through the forwarding process.

The ATS filtering and assignment functions are based on an ATS scheduler clock, whereas the ATS transmission selection algorithm is based on the transmission selection clock. If both are different clocks, a difference between an arbitrary instant of time t_{FA}, as recognized by the ATS scheduler clock instance, and the same instant of time t_{TS}, as recognized by the transmission selection clock, may be observed during the processing of a frame. The time difference may vary over a sequence of frames, which is characterized by

$$\mathrm{ClockOffsetMin} \le t_{TS} - t_{FA} \le \mathrm{ClockOffsetMax},$$

where ClockOffsetMin and ClockOffsetMax are implementation-specific constants that limit the variation.

Any frame may experience an additional, non-negative, processing delay, between its arrival time and its selectability time (8.6.8.5). This delay may vary per frame, thus that there is a delay variation over a sequence of frames. The processing delay is characterized by

$$\mathrm{ProcessingDelayMin} \le \mathrm{processingDelay(frame)} \le \mathrm{ProcessingDelayMax},$$

where ProcessingDelayMin and ProcessingDelayMax are implementation-specific constants, and processingDelay(frame) denotes the processing delay of a frame, not including any potential delay introduced by the associated ATS scheduler state machine instance.

The assigned eligibility time is calculated by

$$\mathrm{assignedEligibilityTime} = \mathrm{eligibilityTime} + \mathrm{ClockOffsetMax} + \mathrm{ProcessingDelayMax}.$$

8.6.11.2.3 BucketEmptyTime

A state variable that contains the most recent instant of time at which the token bucket of the ATS scheduler instance was empty, in seconds.

The BucketEmptyTime variable is initialized with a time earlier than CommittedBurstSize/CommittedInformationRate in the past, as perceived by the ATS Scheduler Clock. After initialization, the number of tokens in the token bucket is equivalent to the CommittedBurstSize parameter.

8.6.11.2.4 bucketFullTime

The instant of time when the number of tokens in the token bucket is equivalent to the CommittedBurstSize parameter, in seconds.

8.6.11.2.5 CommittedBurstSize

The committed burst size of the ATS scheduler instance, in bits (8.6.5.4). The CommittedBurstSize parameter defines the maximum token capacity of the token bucket. In the token bucket model, the number of tokens removed from the bucket by a frame equals the length of the frame, as defined in 8.6.11.2.11.

8.6.11.2.6 CommittedInformationRate

The committed information rate of the ATS scheduler instance, in bits per second (8.6.5.4). The CommittedInformationRate parameter defines the rate at which the token bucket is refilled with tokens until the maximum token capacity of the token bucket is reached.

8.6.11.2.7 Discard(frame)

This procedure discards the frame and increases the DiscardedFramesCount counter of the associated reception port (8.6.5.4). The procedure is called in exceptional situations only, e.g. misbehavior of the connected upstream system.

8.6.11.2.8 eligibilityTime

The eligibility time of the frame, without taking the implementation specific device-internal timing properties of theforwarding process into account. These timing properties are taken into account by the AssignAndProceed(frame) procedure (8.6.11.2.2).

8.6.11.2.9 emptyToFullDuration

The duration required to accumulate a number of tokens equivalent to the CommittedBurstSize parameter, in seconds.

8.6.11.2.10 GroupEligibilityTime

A state variable that contains the most recent value of the eligibilityTime variable from the previous frame, as processed by any ATS scheduler instance in the same ATS scheduler group, in seconds.

The GroupEligibilityTime variable is initialized with a time earlier or equal to the current time, as perceived by the ATS scheduler clock.

8.6.11.2.11 length(frame)

The length of the frame, in bits. This frame length includes the bits of the frame on the media that exclude transmission of a different frame. This includes media-specific framing overhead.

NOTE—For example, on IEEE Std 802.3 full-duplex point-to-point media, frame length includes Preamble, Start Frame Delimiter (SFD), Frame Check Sequence (FCS), and the interframe gap.

8.6.11.2.12 lengthRecoveryDuration

The duration required to accumulate a number of tokens equivalent to length(frame), in seconds.

8.6.11.2.13 MaxResidenceTime

The MaximumResidenceTime parameter of the ATS scheduler group instance associated with the ATS scheduler instance, in nanoseconds (8.6.5.4). The parameter limits the duration for which frames can reside in a bridge.

NOTE—A consistent setup of MaxResidenceTime parameter can be determined by the per hop delay bound, which is a result of the timing analysis (X.2).

8.6.11.2.14 shaperEligibilityTime

The instant of time when the number of tokens in the token bucket is at least equivalent to arrivalTime(frame), in seconds.

<<Editor’s Note: Introduction to Clause 12 in this Document

In this draft of P802.1Qcr, contents from 802.1Qci-2017, 12.31 (PSFP) and 802.1Qcr-D0.3,12.34 (ATS) were merged in 12.31. Additional slides on this restructuring are found here: http://ieee802.org/1/files/public/docs2018/cr-specht-d04-0518-v03.pdf.

>>

12. Bridge management

12.4 Bridge Management Entity

12.4.2 Port configuration

Change table Table 12-2 as indicated:

Table 12-2—Port table entry

Name	Data Type	Operations supported[a]	References
portComponentId	ComponentID	R	12.4.1.5
portPortNumber	Port Number	R	13.25
portMACAddress	MAC address	R	12.4.1.1.3 a)
portDelayExceededDiscards	counter	R	—
portMtuExceededDiscards	counter	R	—
portCapabilities	unsigned	R	—
portTypeCapabilities	unsigned	R	—
portType	enumerated	R	12.4.2.1
portExternal	Boolean	R	—
portAdminPointToPoint	unsigned	RW	IEEE Std 802.1AC
portOperPointToPoint	Boolean	R	IEEE Std 802.1AC
portName	Latin1 String (SIZE(0..32))	RW	—
portMediaDependentOverhead	unsigned	R	12.4.2.2

[a] R = Read-only access; RW = Read/Write access.

Insert clause 12.4.2.2, as shown:

12.4.2.2 Media-dependent overhead

The portMediaDepenentOverhead parameter provides the number of additional octets for media-dependent framing. The overhead includes all octets prior the first octet of the Destination Address field and all octets after the last octet of the frame check sequence.

NOTE—For example, 20 octets to account preamble, start of frame delimiter, and a nominal inter-frame gap (IFG) of 12 octets of IEEE 802.3 point-to-point media.

Change subclause 12.31 and its subclauses and tables, as indicated:

12.31 Managed objects for ~~per-stream filtering and policing~~stream filters, stream gates, and stream filter specifications

The Bridge enhancements for support of ~~per-~~stream ~~filtering~~filters, stream gates, and ~~policing~~stream filter specifications used by the applications specified in 8.6.5.5 are defined in 8.6.5.1, 8.6.5.2, and ~~the~~. The associated state machines are defined in 8.6.10 and 8.6.11.

The objects that comprise this managed resource are as follows:

a) The Stream Parameter Table (12.31.1)
b) The Stream Filter Instance Table (12.31.2)
c) The Stream Gate Instance Table (12.31.3)
d) The Flow Meter Instance Table (12.31.4)
e) The Scheduler Instance Table (12.31.5)
f) The Scheduler Group Instance Table (12.31.6)
g) The Port Parameter Table (12.31.7)

12.31.1 The Stream Parameter Table

There is one Stream Parameter Table per Bridge component. The table contains a set of parameters that supports ~~PSFP(~~the applications in ~~8.6.5.1)~~8.6.5.5, as detailed in Table 12-29. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components.

12.31.1.1 MaxStreamFilterInstances

The maximum number of Stream Filter instances supported by this Bridge component (8.6.5.1).

12.31.1.2 MaxStreamGateInstances

The maximum number of Stream Gate instances supported by this Bridge component (8.6.5.2).

12.31.1.3 MaxFlowMeterInstances

The maximum number of Flow Meter instances supported by this Bridge component (8.6.5.3).

12.31.1.4 SupportedListMax

The maximum value supported by this Bridge component of the AdminControlListLength and OperControlListLength parameters (8.6.5.2). It is available for use by schedule computation software to determine the Bridge component's control list capacity prior to computation.

Table 12-29—The Stream Parameter Table

Name	Data type	Operations supported[a]	Conformance[b]	References
MaxStreamFilterInstances	integer	R	~~BE~~PSFP, ATS	8.6.5.1, 12.31.2
MaxStreamGateInstances	integer	R	~~BE~~PSFP, ATS	~~8.6.5.1~~8.6.5.2, 12.31.3
MaxFlowMeterInstances	integer	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.3, 12.31.4
SupportedListMax	integer	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.2, 12.31.3
MaxSchedulerInstances	integer	R	psfp, ATS	8.6.5.4, 12.31.5
MaxSchedulerGroupInstances	integer	R	psfp, ATS	8.6.5.4, 12.31.6

[a]R= Read only access; RW = Read/Write access.
[b]~~B = Required for Bridge or Bridge component support of PSFP.~~
~~E = Required for end-station support of PSFP.~~PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.1.5 MaxSchedulerInstances

The maximum number of ATS scheduler instances supported by this Bridge component (8.6.5.4).

12.31.1.6 MaxSchedulerGroupInstances

The maximum number of ATS scheduler group instances supported by this Bridge component (8.6.5.4).

12.31.2 The Stream Filter Instance Table

There is one Stream Filter Instance Table per Bridge component. Each table row contains a set of parameters that defines a single Stream Filter (8.6.5.1) with an associated Maximum SDU Size Filter (), as detailed in Table 12-30. The table rows form an ordered list of filter instances, the order being determined by the StreamFilterInstance parameter. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components. Rows in the table can be created or removed dynamically in implementations that support dynamic configuration of stream filters.

12.31.2.1 StreamFilterInstance

An integer index value that determines the place of the stream filter in the ordered list of stream filter instances. The values of StreamFilterInstance are ordered according to their integer value; smaller values appear earlier in the ordered list.

12.31.2.2 stream_handle specification data type

The stream_handle specification data type allows either of the following to be represented:

Table 12-30—Stream Filter Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
StreamFilterInstance	integer	R	~~BE~~PSFP, ATS	8.6.5.1
StreamHandleSpec	stream_handle specification	RW	~~BE~~PSFP, ATS	8.6.5.1
PrioritySpec	priority specification	RW	~~BE~~PSFP, ATS	8.6.5.1
StreamGateInstanceID	integer	RW	~~BE~~PSFP, ATS	8.6.5.1, ~~8.6.5.1.2~~8.6.5.2
FilterSpecificationList	sequence of FilterSpecification	RW	~~BE~~PSFP, ATS	8.6.5.1, ~~8.6.5.1.3~~, 12.31.2.5
MatchingFramesCount	counter	R	~~BE~~PSFP, ats	8.6.5.1
PassingFramesCount	counter	R	~~BE~~PSFP, ats	8.6.5.1, 8.6.5.2
NotPassingFramesCount	counter	R	~~BE~~PSFP, ats	8.6.5.1, 8.6.5.2
PassingSDUCount	counter	R	~~BE~~PSFP, ats	8.6.5.1,
NotPassingSDUCount	counter	R	~~BE~~PSFP, ats	8.6.5.1,
REDFramesCount	counter	R	~~BE~~PSFP, ats	8.6.5.1, 8.6.5.3
StreamBlockedDueToOver-sizeFrameEnable	Boolean	RW	~~BE~~PSFP, ATS	8.6.5.1,
StreamBlockedDueToOver-sizeFrame	Boolean	RW	~~BE~~PSFP, ATS	8.6.5.1,

[a]R= Read only access; RW = Read/Write access.
[b]~~B = Required for Bridge or Bridge component support of PSFP.~~
~~E = Required for end-station support of PSFP.~~PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

a) A stream_handle value, represented as an integer.
b) The wild card value.

12.31.2.3 priority specification data type

The priority specification data type allows either of the following to be represented:

a) A priority value, represented as an integer.
b) The wild card value.

12.31.2.4 StreamGateInstance

The StreamGateInstance parameter identifies the stream gate (12.31.3) that is associated with the stream filter. The relationship between stream filters and stream gates is many to one; a given stream filter can be associated with only one stream gate, but there can be multiple stream filters associated with a given stream gate.

12.31.2.5 FilterSpecification data type

The FilterSpecification data type can represent the following:

a) An integer value representing a Maximum SDU size (~~8.6.5.1,~~).
b) An integer value representing a flow meter instance identifier (~~8.6.5.1, 8.6.5.1.3~~, 8.6.5.3).
c) An integer value representing an ATS scheduler instance identifier (, 8.6.5.4).

12.31.3 The Stream Gate Instance Table

There is one Stream Gate Instance Table per Bridge component. Each table row contains a set of parameters that defines a single Stream Gate (~~8.5.6.1.2~~8.6.5.2), as detailed in Table 12-31. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components. Rows in the table can be created or removed dynamically in implementations that support dynamic configuration of stream gates.

Table 12-31—The Stream Gate Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
StreamGateInstance	integer	R	~~BE~~PSFP,ATS	8.6.5.1, ~~8.6.5.1.2~~8.6.5.2
~~PSFP~~StreamGateEnabled	Boolean	RW	~~BE~~PSFP,ATS	8.6.9.4.14
~~PSFP~~StreamGateAdminGateStates	~~PSFPg~~StreamGateStatesValue	RW	~~BE~~PSFP,ATS	8.6.10.4, 12.29.1.2.2
~~PSFP~~StreamGateOperGateStates	~~PSFPg~~StreamGateStatesValue	R	~~BE~~PSFP,ATS	8.6.10.5, 12.29.1.2.2
~~PSFP~~StreamGateAdminControlListLength	unsigned integer	RW	~~BE~~PSFP,ats	8.6.9.4.6, 12.31.3.2
~~PSFP~~StreamGateOperControlListLength	unsigned integer	R	~~BE~~PSFP,ats	8.6.9.4.23, 12.31.3.2
~~PSFP~~StreamGateAdminControlList	sequence of ~~PSFP~~StreamGateControlEntry	RW	~~BE~~PSFP,ats	8.6.9.4.2, 12.31.3.2, 12.31.3.2.2
~~PSFP~~StreamGateOperControlList	sequence of ~~PSFP~~StreamGateControlEntry	R	~~BE~~PSFP,ats	8.6.9.4.19, 12.31.3.2, 12.31.3.2.2
~~PSFP~~StreamGateAdminCycleTime	RationalNumber	RW	~~BE~~PSFP,ats	8.6.9.4.3, 12.29.1.3
~~PSFP~~StreamGateOperCycleTime	RationalNumber (seconds)	R	~~BE~~PSFP,ats	8.6.9.4.20, 12.29.1.3
~~PSFP~~StreamGateAdminCycleTimeExtension	Integer (nanoseconds)	RW	~~BE~~PSFP,ats	8.6.9.4.4
~~PSFP~~StreamGateOperCycleTimeExtension	Integer (nanoseconds)	R	~~BE~~PSFP,ats	8.6.9.4.21
~~PSFP~~StreamGateAdminBaseTime	PTPtime	RW	~~BE~~PSFP,ats	8.6.9.4.1, 12.29.1.4
~~PSFP~~StreamGateOperBaseTime	PTPtime	R	~~BE~~PSFP,ats	8.6.9.4.18, 12.29.1.4
~~PSFP~~StreamGateConfigChange	Boolean	RW	~~BE~~PSFP,ats	8.6.9.4.7
~~PSFP~~StreamGateConfigChangeTime	PTPtime	R	~~BE~~PSFP,ats	8.6.9.4.9, 12.29.1.4
~~PSFP~~StreamGateTickGranularity	Integer (tenths of nanoseconds)	R	~~BE~~PSFP,ats	8.6.9.4.16
~~PSFP~~StreamGateCurrentTime	PTPtime	R	~~BE~~PSFP,ats	8.6.9.4.10, 12.29.1.4
~~PSFP~~StreamGateConfigPending	Boolean	R	~~BE~~PSFP,ats	8.6.9.3, 8.6.9.4.8
~~PSFP~~StreamGateConfigChangeError	Integer	R	~~BE~~PSFP,ats	8.6.9.3.1

Table 12-31—The Stream Gate Instance Table *(continued)*

Name	Data type	Operations supported[a]	Conformance[b]	References
~~PSFP~~StreamGateAdminIPV	IPV	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2, 8.6.10.6, 12.31.3.3
~~PSFP~~StreamGateOperIPV	IPV	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2, 8.6.10.7, 12.31.3.3
~~PSFP~~StreamGateClosedDueTo-InvalidRxEnable	Boolean	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2
~~PSFP~~StreamGateClosedDueToIn-validRx	Boolean	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2
~~PSFP~~StreamGateClosedDueTo-OctetsExceededEnable	Boolean	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2
~~PSFP~~StreamGateClosedDueTo-OctetsExceeded	Boolean	RW	~~BE~~PSFP,ATS	~~8.6.5.1.2~~8.6.5.2

[a]R= Read only access; RW = Read/Write access.
[b]~~B = Required for Bridge or Bridge component support of PSFP.~~
~~E = Required for end-station support of PSFP.~~PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.3.1 StreamGateInstance

An integer table index that allows the stream gate to be referenced from Stream Filter Instance Table entries.

12.31.3.2 The gate control list structure and data types

The AdminControlList and OperControlList are ordered lists containing AdminControlListLength or OperControlListLength entries, respectively. Each entry represents a gate operation as defined in Table 8-4. Each entry in the list is structured as a GateControlEntry (12.31.3.2.2).

12.31.3.2.1 ~~PSFPg~~StreamGateStatesValue

The ~~PSFPg~~StreamGateStatesValue indicates the desired gate state, *open* or *closed*, for the stream gate.

12.31.3.2.2 ~~PSFP~~StreamGateControlEntry

A ~~PSFP~~StreamGateControlEntry consists of an operation name, followed by three parameters associated with the operation, as detailed in Table 8-4. The first parameter is a ~~PSFPg~~StreamGateStatesValue (12.31.3.2.1); the second parameter is an IPV value (12.31.3.2.3), and the third parameter is a timeIntervalValue (12.31.3.2.4).

12.31.3.2.3 IPV value

The IPV value indicates the IPV (12.31.3.3) to be associated with frames that pass the gate (8.6.10.7).

12.31.3.2.4 timeIntervalValue

An unsigned integer, denoting a TimeInterval in nanoseconds (see TimeInterval in Table 8-4).

12.31.3.3 The Internal priority value specification (IPV) data type

The IPV data type represents an IPV value (~~8.6.5.1.2~~8.6.5.2); this is either the null value or an internal priority value.

12.31.3.4 Representation of times

Table 12-31 specifies times (e.g. ~~PSFP~~StreamGateAdminBaseTime) with reference to the on-the-wire timing point at which the start of a frame crosses the boundary between the physical network media and PHY. This is the message timestamp point specified by IEEE Std 802.1AS for various media.

Replace Figure 12-1, as shown

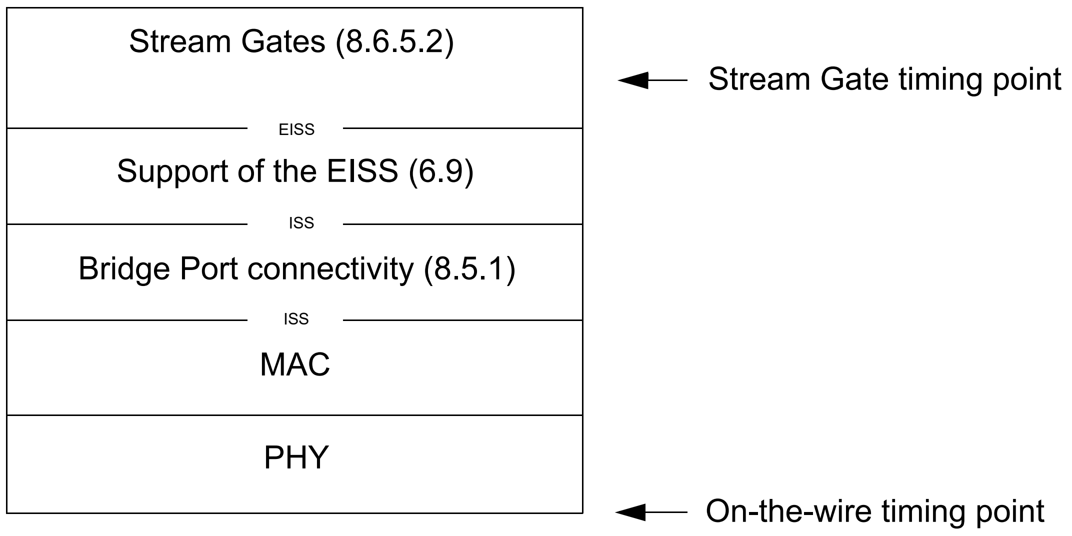


Figure 12-1—Timing points for Stream Gates

Figure 12-1 shows both the on-the-wire timing point and the ~~PSFP~~Stream Gate timing point within the interface stack (above MAC and PHY) that is used for gate open/close as described in ~~8.6.5.1~~8.6.5.2. Each timing point will have variance. A delay exists between the on-the-wire timing point and the ~~PSFP~~Stream Gate timing point. This delay will have variance that is bounded (minimum/maximum).

The Bridge contains the information needed in order to compute the minimum/maximum delay from the on-the-wire timing point to the ~~PSFP~~Stream Gate timing point. The Bridge shall adjust using the maximum delay, such that a frame's on-the-wire timing point occurs no later than the gate events represented in managed objects.

NOTE—Although the managed objects apply to a Bridge, the preceding specification of on-the-wire timing point can be applied to an end station.

12.31.4 The Flow Meter Instance Table

There is one Flow Meter Instance Table per Bridge component. Each table row contains a set of parameters that defines a single Flow Meter Instance (~~8.6.5.1~~8.6.5.3), as detailed in Table 12-32. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components. Rows in the table can be created or removed dynamically in implementations that support dynamic configuration of flow meters.

Table 12-32—The Flow Meter Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
FlowMeterInstanceID	integer	R	~~BE~~PSFP, ats	~~8.6.5.1,~~ ~~8.6.5.1.3,~~ 8.6.5.3
CIR	integer, bit/s	RW	~~BE~~PSFP, ats	~~8.6.5.1,~~ ~~8.6.5.1.3,~~ 8.6.5.3
CBS	integer, octets	RW	~~BE~~PSFP, ats	~~8.6.5.1,~~ ~~8.6.5.1.3,~~ 8.6.5.3
EIR	integer, bit/s	RW	~~BE~~PSFP, ats	~~8.6.5.1,~~ ~~8.6.5.1.3,~~ 8.6.5.3
EBS	integer, octets	RW	~~BE~~PSFP, ats	~~8.6.5.1,~~ ~~8.6.5.1.3,~~ 8.6.5.3
CF	integer, 0 or 1	RW	~~BE~~PSFP, ats	~~8.6.5.1,~~ ~~8.6.5.1.3,~~ 8.6.5.3
CM	enumerated, color-blind or color-aware	RW	~~BE~~PSFP, ats	~~8.6.5.1,~~ ~~8.6.5.1.3,~~ 8.6.5.3
DropOnYellow	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1,~~ ~~8.6.5.1.3,~~ 8.6.5.3
MarkAllFramesRedEnable	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1,~~ ~~8.6.5.1.3,~~ 8.6.5.3
MarkAllFramesRed	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1,~~ ~~8.6.5.1.3,~~ 8.6.5.3

[a]R= Read only access; RW = Read/Write access.
[b]~~B = Required for Bridge or Bridge component support of PSFP.~~
~~E = Required for end-station support of PSFP.~~PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS..

Insert clauses 12.31.5, 12.31.6, and 12.31.7 as shown, including the contained tables:

12.31.5 The Scheduler Instance Table

There is one Scheduler Instance Table per Bridge component. Each table row in the Scheduler Instance Table comprises a set of parameters that defines a single ATS scheduler instance, as detailed in Table 12-33.

Table 12-33—The Scheduler Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
SchedulerInstanceID	integer	R	psfp,ATS	8.6.5.4
CommittedBurstSize	integer, bits	RW	psfp,ATS	8.6.5.4, 8.6.11
CommittedInformationRate	integer, bits/s	RW	psfp,ATS	8.6.5.4, 8.6.11
SchedulerGroupInstanceID	integer	RW	psfp,ATS	8.6.5.4

[a]R= Read only access; RW = Read/Write access.
[b]PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

NOTE—ATS scheduler groups establish the relationship between ATS scheduler instances and the per port management variables (12.31.7), as described in 8.6.5.4. As a result, the scheduler instance table does not contain references to ports.

12.31.5.1 SchedulerInstanceID

An integer table index that allows the ATS scheduler instance to be referenced from Stream Filter Instance Table entries.

12.31.5.2 CommittedBurstSize

As specified in 8.6.11.2.5.

12.31.5.3 CommittedInformationRate

As specified in 8.6.11.2.6.

12.31.5.4 SchedulerGroupInstanceID

The SchedulerGroupInstanceID parameter identifies the ATS scheduler group (12.31.6) that is associated with the ATS scheduler instance. Multiple scheduler instance can be associated to one ATS scheduler group, as detailed in 8.6.5.4.

12.31.6 The Scheduler Group Instance Table

There is one Scheduler Group Instance Table per Bridge component. Each table row in the Scheduler Group Instance Table comprises a set of parameters that defines a single ATS scheduler group instance (8.6.5.4), as detailed in Table 12-34.

Table 12-34—The Scheduler Group Instance Table

Name	Data type	Operations supported[a]	Conforman ce[b]	References
SchedulerGroupInstanceID	integer	R	psfp,ATS	8.6.5.4
MaxResidenceTime	integer, nanoseconds	RW	psfp,ATS	8.6.5.4

[a]R= Read only access; RW = Read/Write access.
[b]PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.6.1 SchedulerGroupInstanceID

An integer table index that allows the ATS scheduler group instance to be referenced from Scheduler Instance Table entries.

12.31.6.2 MaxResidenceTime

As specified in 8.6.11.2.13.

12.31.7 The Port Parameter Table

There is one Port Parameter Table per Bridge. Each table row in the Port Parameter Table comprises a set of parameters shared by all ATS scheduler instance associated with a reception Port, as detailed in Table 12-35.

Table 12-35—The Port ParameterTable

Name	Data type	Operations supported[a]	Conforman ce[b]	References
PortNumber	integer	R	psfp,ATS	12.31.7.1
DiscardedFramesCount	integer	R	psfp,ATS	8.6.5.4, 8.6.11.2.7

[a]R= Read only access; RW = Read/Write access.
[b]PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.7.1 PortNumber

An unique index of the associated Bridge Port (12.4.2).

12.31.7.2 DiscardedFramesCount

As specified in 8.6.5.4 and 8.6.11.2.7.

Thank you for your Attention!

Questions, Opinions, Ideas?

Johannes Specht
Dipl.-Inform. (FH)

Dependability of Computing Systems	Schuetzenbahn 70
Institute for Computer Science and	Room SH 502
Business Information Systems (ICB)	45127 Essen
Faculty of Economics and	GERMANY
Business Administration	T +49 (0)201 183-3914
University of Duisburg-Essen	F +49 (0)201 183-4573

Johannes.Specht@uni-due.de
http://dc.uni-due.de