This document is a further update to my detailed proposal for the reorganization of clause 8.6.5 suggested by my ballot comments on P802.1Qcr/D1.0 and referenced by a note in .1Qcr/D1.1 recirculation ballot text. This update used D1.1 text as a baseline with a goal of preserving technical changes made between D1.0 and D1.1.

This update follows discussion with Johannes Specht and Paul Congdon, with broad agreement on the lines of the proposal.

The aim of the reorganization is to start with a high level decomposition of each stage of the problem, using ideas that are hopefully already familiar to someone who has read early editions of the published 802.1Q standard, but has not been in the room while P802.1Qcr amendment is being discussed. The challenge put to me was to introduce that top-down explanation without making use of excessive forward references.

I have endeavored to solve the problem that I encountered when attempting to read P802.1Qcr drafts to date of rapidly becoming lost in the detail, referencing back and forth with mental stack overflow. I have tried to avoid any technical changes to D1.1.

In preparing this text I have used a number of editing stylistic approaches to keep things clear while remaining within a style and use of language that has been found acceptable to IEEE staff or their contract editors. I have also tried to keep the use of language consistent with early editions of 802.1Q, though undoubtedly the efforts of many editors has already introduced some variety.

This document actually consists of two parts. The first is the proposed replacement 8.6.5. With the amount of moving around text that has been involved, I have not produced a strikeout/insert change copy. In my experience such a copy is both hard to produce and to decipher when a major reorganization has been done. This text has a P802.1Qcr/D1.1 heading, but a date of July 5, 2019. The second part is the actual D1.1 text, dated June 21st (see top corner of the pages) with some PDF annotations made as I went through as a final exercise to make sure I had not missed anything. They may be useful when reviewing the proposed text.

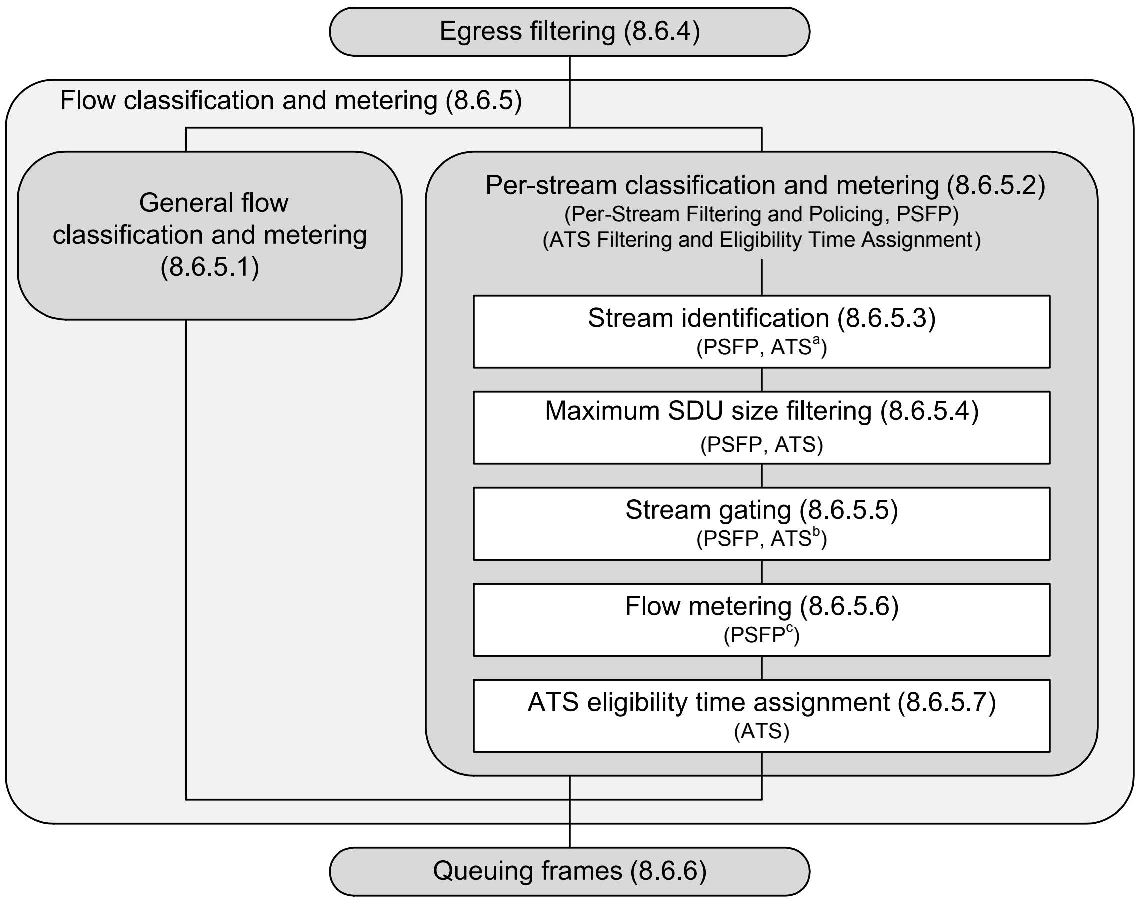
Mick Seaman, 7th July 2019

8.6.5 Flow classification and metering

The Forwarding Process can apply flow classification and metering to frames that are received on a Bridge Port and have one or more potential transmission ports. Bridge ports and end stations may support Per-Stream Filtering and Policing (PSFP), Asynchronous Traffic Shaping (ATS) filtering and eligibility time assignment, or the general flow classification rules specified in 8.6.5.1.

NOTE—The general flow classification and metering specification was added to this standard by IEEE Std 802.1Q-2005, PSPF by IEEE Std 802.1Qci-2017, and ATS by IEEE Std 802.1Qcr-2020.

PSFP and ATS share common classification and metering elements, as shown in Figure 8-13. The stream identification function specified in IEEE Std 802.1CB is used to associate each received frame with a set of stream parameters that can also identify an applicable SDU size filter, a stream gate, and a flow meter (for PSFP) or a transmission eligibility time scheduler (for ATS).



[a]Identifies the stream and the processing (PSFP, ATS, both, or neither) applied to each frame.
[b]Can determine internal frame processing priority, and can (separately) enforce permitted arrival times for scheduled streams. Both optional for any given stream, but can provide coexistence with strict time scheduled traffic.
[c]Required PSFP functionality, supports MEF 10.3 flow metering, Can use and modify discard eligible frame marking, for both PSFP and ATS.

Figure 8-13—Flow classification and metering

8.6.5.1 General flow classification and metering

Bridges that implement general flow classification and metering can identify subsets of traffic (frames), each subject to the same flow 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—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 2—Although flow metering is applied after egress (Figure 8-12), the meter(s) operate per reception Port, not per potential transmission Port(s).

8.6.5.2 Per-stream classification and metering

When Per-Stream Filtering and Policing (PSFP) or Asynchronous Traffic Shaping (ATS) is used, filtering and policing decisions for received frames are made, and subsequent queuing (8.6.6) and transmission selection decisions (8.6.8) supported, as follows:

a) Each received frame is associated with a stream filter, as specified in 8.6.5.3.
   If no matching stream filter is found, the frame is queued for transmission as specified in 8.6.6, without further frame classification and filtering processing. Wildcard stream filters can be configured to match and discard frames not associated with a specified stream.
b) The frame is subject to maximum SDU size filtering (8.6.5.4), using parameters specified by the stream filter.
   The ATS scheduler state machine operation (8.6.11) assumes that the sizes of frames that it processes are less than or equal to the associated CommittedBurstSize parameter (8.6.11.3.5).
c) If the stream filter specifies a stream gate (8.6.5.5), that is used to process the frame.
   The frame can be discarded if there is excess traffic for the stream, or it is associated with a scheduled stream and has not been received in a permitted interval. The frame’s priority can be mapped to an internal priority value (IPV) that can influence subsequent queuing decisions (8.6.6).
   A stream filter can be configured without a stream gate, if simpler behavior is desired, with an IPV equal to the frame’s priority.
d) If the stream filter specifies one or more flow meters (8.6.5.5), they are used to process the frame.
   The frame can be discarded or marked as drop eligible. A stream filter can be configured without a flow meter. A given stream filter can be configured with a flow meter and an ATS Scheduler if both PSFP and ATS are supported.

e) If the stream filter specifies an ATS scheduler (8.6.5.7), that is used to compute an eligibility time for the frame for subsequent use by the ATS transmission selection algorithm (8.6.8.5).

8.6.5.2.1 PSFP support

Each Bridge component or end station that implements PSFP supports stream identification, maximum SDU size filtering, stream gates, and flow meters, with the following:

a) A single *Stream Filter Instance Table* (8.6.5.3).
b) A single *Stream Gate Instance Table* (8.6.5.5).
c) A single *Flow Meter Instance Table* (8.6.5.6).

The relationship between stream filters, stream gates, flow meters for streams subject to PSFP processing (as identified by the stream filter) is illustrated by Figure 8-14.

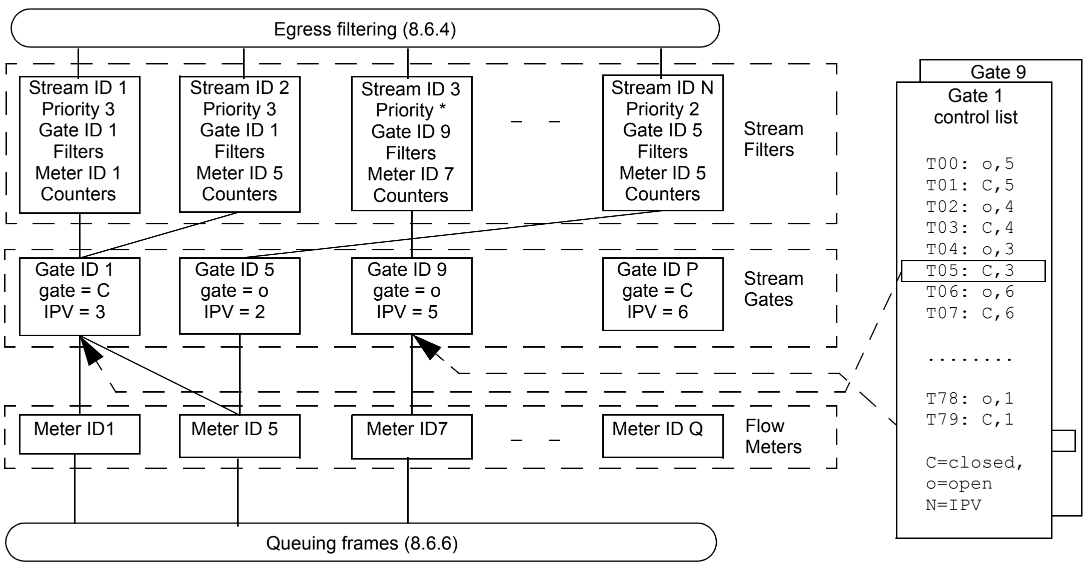


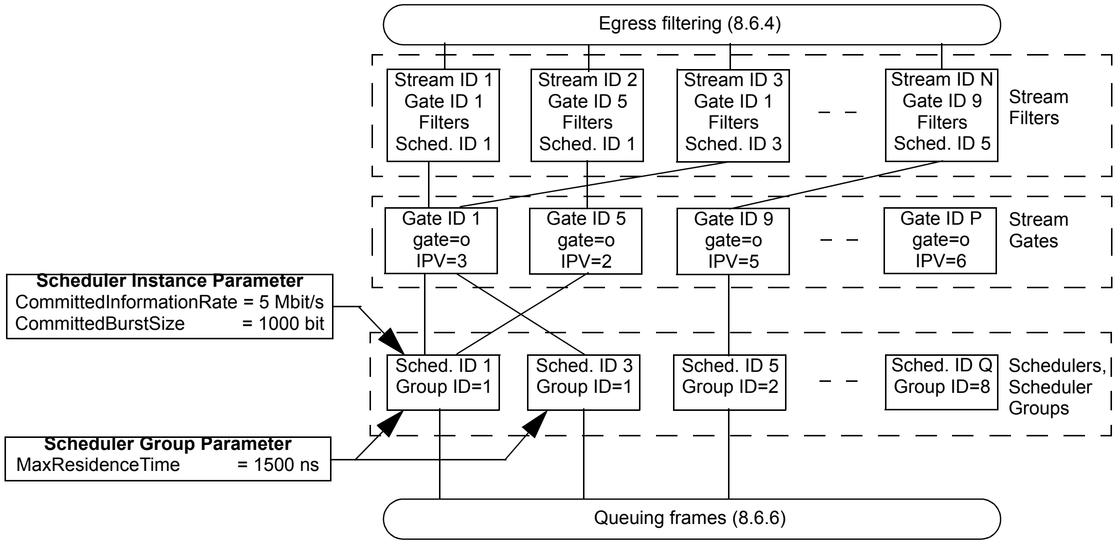
Figure 8-14—Per-stream filtering and policing

8.6.5.2.2 ATS support

Each Bridge component that implements ATS supports stream identification, maximum SDU size filtering, stream gates supporting IPV assignment, ATS schedulers, and ATS scheduler groups, with the following:

a) A single *Stream Filter Instance Table* (8.6.5.3).
b) A single *Stream Gate Instance Table* (8.6.5.5).
   If the Bridge component does not support scheduled traffic and PSFP in addition to ATS, each stream gate only supports IPV assignment (8.6.5.5). IPV can be used as part of adjusting per-hop delay bounds to meet specific networks' end-to-end delay requirements.
c) A single *ATS Scheduler Instance Table* (8.6.5.7).
d) A single *ATS Scheduler Group Instance Table* (8.6.5.7).
e) An *ATS Port Parameter Table* for each Bridge Port (8.6.5.7).

The relationship between stream filters, stream gates, and ATS schedulers for streams subject to ATS processing is illustrated by Figure 8-15.



KEY
Stream ID: stream filter instance identifier (8.6.5.3)
Gate ID: stream gate instance identifier (8.6.5.3, 8.6.5.5)
Sched. ID:ATS scheduler instance identifier (8.6.5.3, 8.6.5.7)

Figure 8-15—ATS filtering and assignment functions

ATS support in end stations is provided by a modified variant of the ATS filtering and assignment functions, as specified in clause 49.

8.6.5.3 Stream identification

Each received frame is associated with a stream filter using the frame's *stream_handle* and *priority* parameters. The *stream_handle* is a sub-parameter of the connection_identifier parameter of the ISS (6.6), provided by the stream identification function specified in Clause 6 of IEEE Std 802.1CB-2017.

Each stream filter comprises the following:

a) An integer *stream filter identifier*.
b) A *stream_handle* specification, either:
   1) A single value, as specified in IEEE Std 802.1CB.
   2) A wildcard, that matches any *stream_handle*.
c) A *priority* specification, either:
   1) A single priority value.
   2) A wildcard value that matches any priority value.
d) Maximum SDU size filtering (8.6.5.4) information, comprising:
   1) An integer *Maximum SDU size*, in octets.
   2) A boolean *StreamBlockedDueToOversizeFrameEnable* parameter.
   3) A boolean *StreamBlockedDueToOversizeFrame* parameter.
e) An integer *stream gate identifier* (8.6.5.5).
   If this parameter is absent, a stream gate is not used to discard or to assign an internal priority value (IPV) to a frame associated with the stream filter.
f) A list of of integer *flow meter instance identifiers* (8.6.5.6).
   If this list is empty, flow meters are not used to selectively discard or change the parameters of frames associated with the stream filter.

g) An integer ATS *scheduler instance identifier* (8.6.5.7).

   If this parameter is absent, frames associated with the stream filter are not subject to ATS scheduling and transmission selection.

and the following counters for frames associated with the stream filter:

h) *MatchingFramesCount*: all frames associated with that stream filter.
i) *PassingSDUCount*: frames passing the maximum SDU size filter (8.6.5.4).
j) *NotPassingSDUCount*: frames not passing the maximum SDU size filter (8.6.5.4).
k) *PassingFrameCount*: frames passing the associated stream gate (8.6.5.5).
l) *NotPassingFrameCount*: frames not passing the stream gate (8.6.5.5).
m) *RedFramesCount*: frames discarded by the flow meter (8.6.5.6).

The *stream filter identifier* uniquely identifies the stream filter, indexing a *Stream Filter Instance Table* of up to *MaxStreamFilterInstances* stream filters. Each received frame is associated with the stream filter with the lowest *stream filter identifier* whose *stream_handle* and *priority* specification match the frame's parameters, and the *MatchingFramesCount* is incremented for that filter. A *stream filter identifier* value of 0 is reserved to indicate the lack of any match.

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, for example, per-priority filtering and policing, or per-priority per-reception Port filtering and policing can be configured using these rules.

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, a stream filter can be placed at the end of the table with stream_handle and priority specifications both wild-carded and the stream gate identifier of a permanently closed stream gate.

8.6.5.4 Maximum SDU size filtering

If the SDU size of a frame exceeds the value of the associated stream filter's Maximum SDU size parameter, the frame is discarded and that stream filter's NotPassingSDUCount is incremented. If the stream filter's StreamBlockedDueToOversizeFrameEnable parameter is configured to be TRUE, the StreamBlockedDueToOversizeFrame parameter is set to TRUE and all subsequent frames will be discarded until StreamBlockedDueToOversizeFrame is administratively reset to FALSE.

Otherwise, the stream filter's PassingSDUCount is incremented (see 8.6.5.3). The default configuration of both StreamBlockedDueToOversizeFrameEnable and StreamBlockedDueToOversizeFrame is FALSE.

NOTE—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.

8.6.5.5 Stream gating

The Forwarding Process can use a stream gate to enforce scheduled use of bandwidth by discarding frames associated with scheduled traffic whose reception is not permitted at particular times. Stream gates can also map the frame's priority to an internal priority value (IPV) that is used to make subsequent queuing decisions (8.6.6), while retaining the frame's original priority for transmission.

NOTE 1—The IPV facilitates ATS per-hop delay bound adjustment to satisfy specific networks' end-to-end delay requirements. Annex T (CQF) describes another IPV use case.

Each stream gate comprises the following:

a) An integer *stream gate instance identifier*.
b) An administrative and an operational *stream gate state* parameter.
   The operational *stream gate state* can take one of two values:
   1) Open: Frames are permitted to pass through the stream gate.
   2) Closed: Frames are not permitted to pass through the stream gate.
c) An administrative and an operational *internal priority value specification*.
   The operational *internal priority value specification* can be one of the following:
   1) Null, in this case the received frame's priority parameter is used as the IPV.
   2) A specific IPV for the frame.

The *stream gate instance identifier* uniquely identifies the stream gate, indexing a *Stream Gate Instance Table* of up to *MaxStreamGateInstances* stream gates.

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 only used for IPV assignment.

If scheduled traffic and PSPF are supported, each stream gate also includes the following:

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

An instance of the stream gate control state machine (8.6.10) determines the operational values of the *stream gate state* and the *internal priority value specification* [(b) and (c) above] by the cyclical execution of the control operations (see Table 8-4) specified in the stream gate's *stream gate control list* [(d) above]. The administrative *stream gate state* and *internal priority value specification* parameters are used to determine the initial values of the corresponding operational parameters, and the administrative *stream gate control list* parameter allows configuration of a new control list prior to enabling its use in a running system.

NOTE 3—If stream gates are used for scheduled traffic co-existent with ATS traffic, the asynchronous nature of ATS traffic can require simultaneous gate state changes of multiple stream gates for ATS traffic.

If a frame is passed by a stream gate, the *PassingFrameCount* of the stream filter (8.6.5.3) associated with that frame is incremented. The *NotPassingFrameCount* is incremented if the frame is discarded.

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.

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:

i) 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.
j) 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.

8.6.5.6 Flow metering

The flow meters specified by this clause (8.6.5.6) implement the parameters and algorithm specified in *Bandwidth Profile Parameters and Algorithm* in MEF 10.3 with the additions described in this clause.

Each flow meter comprises the following:

a) An integer *flow meter 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 identifier* uniquely identifies the flow meter instance, indexing a *Flow Meter Instance Table* of up to *MaxFlowMeterInstances* flow meters.

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 can permanently discard all frames after an initial frame has been discarded. This capability is enabled if the MarkAllFramesRedEnable parameter is TRUE, and disabled if this parameter is FALSE. If 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.

Changes of the MarkAllFramesRedEnable parameter and transitions from TRUE to FALSE of the MarkAllFramesRed parameter are administrative actions. Per default, permanent frame discarding 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 (8.6.5.3) is increased.

8.6.5.7 ATS eligibility time assignment

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—ATS schedulers, as defined in this clause (8.6.5.7), realize the computational part of the overall traffic shaping operation of ATS. The complete operation is provided in combination with the ATS transmission selection algorithm (8.6.8.5), which uses the assigned eligibility times to regulate the traffic for transmission.

Each ATS Scheduler comprises the following:

a) An integer *scheduler identifier*.
b) An integer *scheduler group identifier*.
c) An integer *CommittedBurstSizeParameter* parameter, in bits (8.6.11.3.5).
d) An integer *CommittedInformationRate* parameter, in bits per second (8.6.11.3.6).
e) An internal *bucket empty time* state variable, in seconds (8.6.11.3.3).

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 schedulers that process frames from a particular reception Port and a particular upstream traffic class are in the respective ATS scheduler group.

Each ATS Scheduler group comprises the following:

f) An integer *scheduler group identifier*.
g) An integer *MaximumResidenceTime* parameter, shared by all ATS schedulers in a scheduler group, in nanoseconds (8.6.11.3.13).
h) An internal *group eligibility time* state variable, shared by all ATS schedulers in a scheduler group, in seconds (8.6.11.3.10).

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

Each ATS scheduler 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. This state machine updates the associated bucket empty time and group eligibility time state variables based on the CommittedBurstSize, the CommittedInformationRate, the MaxResidenceTime, the frame arrival times, and the frame lengths (including media-specific overhead).

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

i) An integer *DiscardedFramesCount* of frames discarded by the ATS schedulers associated with that reception Port.

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

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 in transmission ports is implementation specific.

8.6.6 Queuing frames

Change the text paragraph in 8.6.6 before Table 8-5, as indicated in red:

The Forwarding Process provides one or more queues for a given Bridge Port, each corresponding to a distinct traffic class. Each frame is mapped to a traffic class using the Traffic Class Table for the Port ~~and the frame's priority~~. The priority value used for this mapping is determined as follows:

c) If stream gates are unsupported (8.6.5.5), the frame's priority is used.
d) If stream gates are supported and the IPV specification assigned to the frame is the null value, the frame's priority is used.
e) If stream gates are supported and the IPV specification assigned to the frame is an IPV, this IPV is used.

Traffic class tables may be managed. Table 8-5 shows the recommended mapping for the number of classes implemented, in implementations that do not support the credit-based shaper transmission selection algorithm (8.6.8.2). The requirements for priority to traffic class mappings in implementations that support the credit-based shaper transmission selection algorithm are defined in 34.5. Up to eight traffic classes may be supported, allowing separate queues for each priority.

Delete clause 8.6.6.1

Insert new clause 8.6.8, 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.

8. Principles of Bridge operation

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.

This clause (8.6.5) specifies flow classification and metering elements in clauses 8.6.5.1, 8.6.5.2 and 8.6.5.3 for the use by the applications defined in clause 8.6.5.4. Bridges that do not support these applications can implement flow classification and metering according to the compatibility requirements in 8.6.5.5.

NOTE—For example, general compatibility requirements (8.6.5.5) can apply to interworking on L3 DiffServ defined in IETF RFC2475 [B33], and support for the use of Provider Bridging for Carrier Ethernet Services defined in MEF.

The ordering of the elements specified in clauses 8.6.5.1, 8.6.5.2 and 8.6.5.3 and their relationship to the requirements in 8.6.5.5 is illustrated in Figure 8-12.

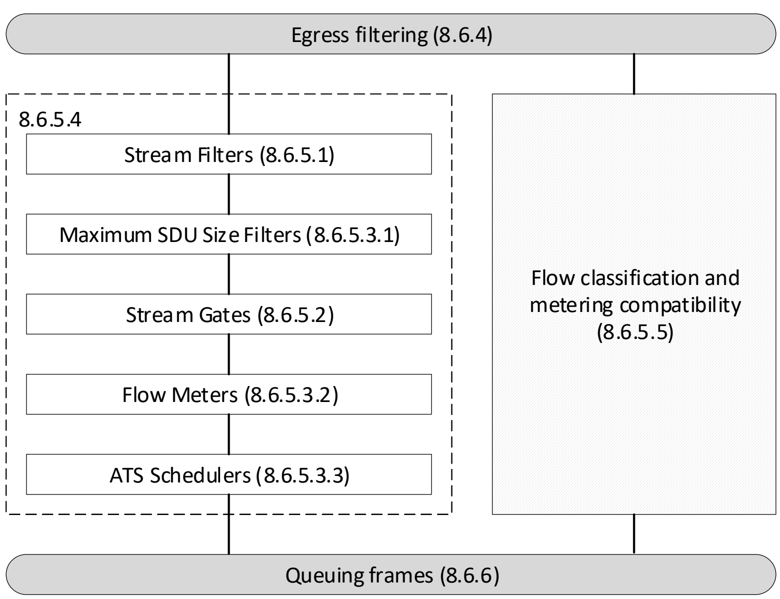


Figure 8-12—Flow classification and metering elements

The ordering of elements from 8.6.5.1, 8.6.5.2, and 8.6.5.3 affects subsequently introduced state variables of these elements (e.g., frame counters), because individual elements can discard frames. Whenever an earlier element discards a frame, the frame is not visible to subsequent elements and the remaining forwarding process.

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) A *MatchingFramesCount* counter for frames matching the stream filter.
g) A *PassingSDUCount* counter for frames that passed the Maximum SDU size filter (8.6.5.3.1).
h) A *NotPassingSDUCount* counter for frames that did not pass the Maximum SDU size filter (8.6.5.3.1).
i) A *PassingFrameCount* counter for frames passing the associated stream gate (8.6.5.2).
j) A *NotPassingFrameCount* counter for frames not passing the associated stream gate (8.6.5.2).
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-2017 [14].
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 matches, 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, 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 Internal Priority Value (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 operations based on these variables and parameters are described in the remainder of this clause (8.6.5.2). The relationship between stream filters, stream gates, and the associated variables and parameters of both is illustrated in Figure 8-13 for a number of streams.

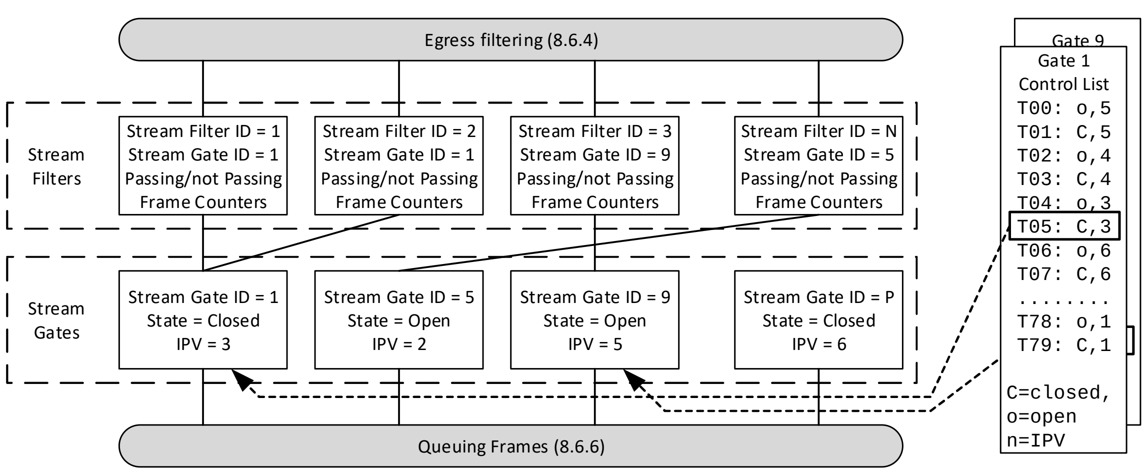


Figure 8-13—Stream Gates and Stream Filters

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

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 IPV specification. This specification is assigned to all frames that pass the associated gate, which allows to override the priority values associated with the frames during subsequent queuing decisions (8.6.6). 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 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 specification assigned to a frame co-exists with the priority value associated with the frame (i.e., an IPV does not replace this priority value).

NOTE 1—The co-existence of the IPV specification 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 for the ability to assign IPVs 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 value 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 in case a frame is discarded due to a closed gate state [8.6.5.2 item j)], and in case a frames are discarded because there are insufficient octets left (i.e., the value of the IntervalOctetsLeft parameter, as specified in 8.6.10.8, is lower than the frame length).

m) Permanent frame discarding due to a frame received during a closed gate state is enabled if the GateClosedDueToInvalidRxEnable parameter is TRUE, and disabled if this parameter is FALSE. If enabled and any frame is discarded, then the GateClosedDueToInvalidRx parameter is set to TRUE, and all subsequent frames are discarded as long as the GateClosedDueToInvalidRxEnable and GateClosedDueToInvalidRx parameters are TRUE. Changes of the GateClosedDueToInvalidRxEnable parameter and transitions from TRUE to FALSE of the GateClosedDueToInvalidRx parameter are administrative actions.
n) Permanent frame discarding insufficient left octets is enabled if the GateClosedDueToOctetsExceededEnable parameter is TRUE, and disabled if this parameter is FALSE. If enabled and any frame is discarded because there are insufficient octets left, then the GateClosedDueToOctetsExceeded parameter is set to TRUE, and all subsequent frames are discarded as long as the GateClosedDueToOctetsExceededEnable and GateClosedDueToOctetsExceededEnable parameters are TRUE. Changes of the GateClosedDueToOctetsExceededEnable parameter and transitions from TRUE to FALSE of the GateClosedDueToOctetsExceeded parameter are administrative actions.

Per default, permanent frame discarding is disabled and all associated parameters have the default value FALSE.

NOTE 2—Permanent frame discarding allows 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 schedulers

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 schedulers), 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 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 permanently discard all frames after an initial frame has been discarded. This capability is enabled if the StreamBlockedDueToOversizeFrameEnable parameter is TRUE, and disabled if this parameter is FALSE. If 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.

Changes of the StreamBlockedDueToOversizeFrameEnable parameter and transitions from TRUE to FALSE of the StreamBlockedDueToOversizeFrame parameter are administrative actions. Per default, permanent frame discarding 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 enabled if the MarkAllFramesRedEnable parameter is TRUE, and disabled if this parameter is FALSE. If 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.

Changes of the MarkAllFramesRedEnable parameter and transitions from TRUE to FALSE of the MarkAllFramesRed parameter are administrative actions. Per default, permanent frame discarding 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—ATS schedulers, as defined in this clause (8.6.5.3.3), 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 (8.6.8.5), 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 a *ATS Scheduler*

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 (8.6.11.3.5).
d) An integer *CommittedInformtionRate* parameter, in bits per second (8.6.11.3.6).
e) An internal *bucket empty time* state variable, in seconds (8.6.11.3.3).

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 (8.6.11.3.13).
h) An internal *group eligibility time* state variable, shared by all ATS scheduler instances in a scheduler group, in seconds (8.6.11.3.10).

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

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 in transmission ports is implementation specific.

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), 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.

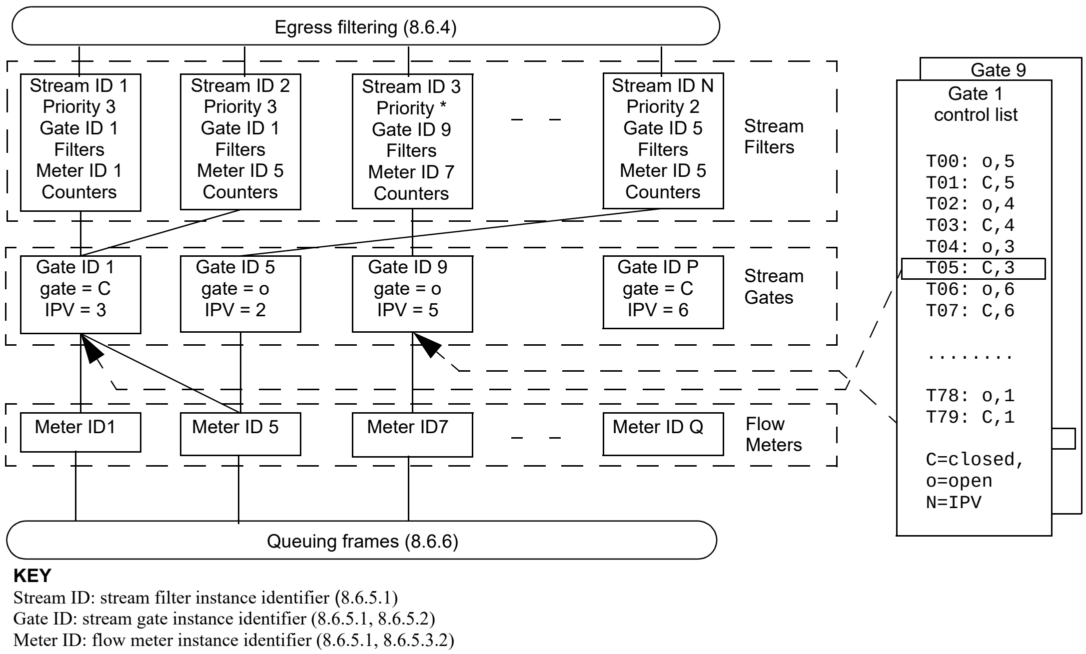


Figure 8-14—Per-stream filtering and policing

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.

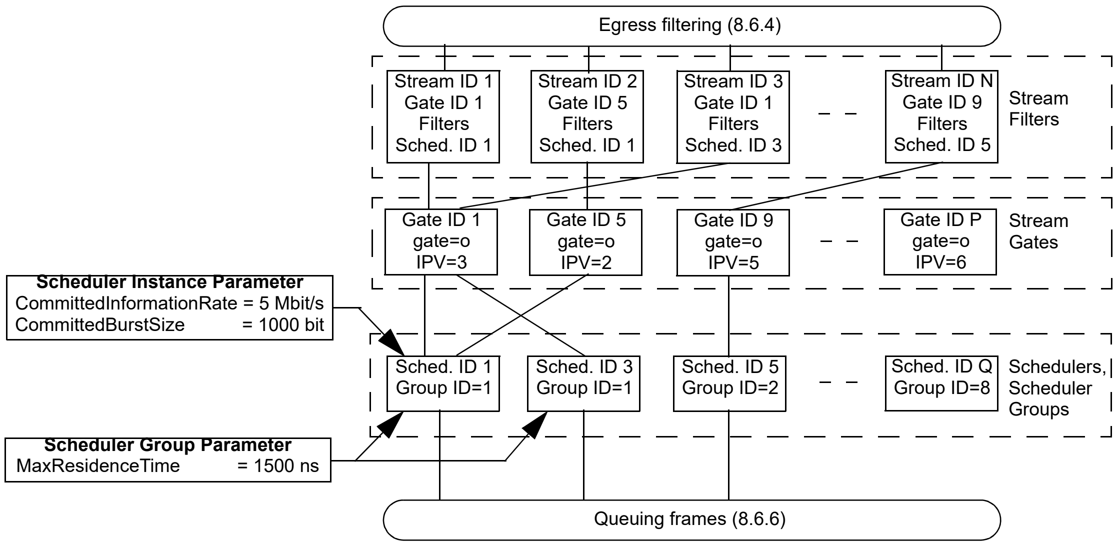
8.6.5.4.2 Asynchronous Traffic Shaping (ATS) Filtering and Assignment Functions

Bridges may support the Asynchronous Traffic Shaping (ATS) Filtering and Assignment Functions that allow filtering decisions, subsequent queuing decision (8.6.6), 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 (8.6.11.3.5). 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—IPV can be used with ATS as part of adjusting per-hop delay bounds to satisfy end-to-end delay requirements in a specific network.

NOTE 4—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 the text paragraph in 8.6.6 before Table 8-5, as indicated in red:

The Forwarding Process provides one or more queues for a given Bridge Port, each corresponding to a distinct traffic class. Each frame is mapped to a traffic class using the Traffic Class Table for the Port~~ and the frame's priority~~. The priority value used for this mapping is determined as follows:

c) If stream gates are unsupported (8.6.5.2), the frame's priority is used.
d) If stream gates are supported and the IPV specification assigned to the frame is the null value, the frame's priority is used.
e) If stream gates are supported and the IPV specification assigned to the frame is an IPV, this IPV is used.

Traffic class tables may be managed. Table 8-5 shows the recommended mapping for the number of classes implemented, in implementations that do not support the credit-based shaper transmission selection algorithm (8.6.8.2). The requirements for priority to traffic class mappings in implementations that support the credit-based shaper transmission selection algorithm are defined in 34.5. Up to eight traffic classes may be supported, allowing separate queues for each priority.

Delete clause 8.6.6.1

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