

Text Proposal for: A (sub-)section on Shaper Interoperability in IEEE 802.1DG

Max Turner
Ethernovia
(BMW, Ruetz System Solutions)
June 2021, v4

1 Glossary

1.1 Latency

The time it takes for a full frame to pass from a transmitter to a receiver. It is often referred to as the time “first bit out to last bit in”.

1.2 Delay

The time it takes for a particular element of a frame (usually the start of the frame) to pass from a transmitter to a receiver. It is often referred to as the time “first bit out to first bit in”.

1.3 TAS – Time Aware Shaper

Originally specified in IEEE Std. 802.1Qbv Enhancements to Traffic Scheduling.

1.4 CBS – Credit Based Shaper

Originally specified in IEEE Std. 802.1Qav as Forwarding and Queuing Enhancements for Time-Sensitive Streams (FQTSS).

1.5 ATS – Asynchronous Shaper

Originally specified in IEEE Std. 802.1Qcr Asynchronous Traffic Shaping.

2 IEEE802.1Q Egress Model

Within the transmission selection model of IEEE802.1Q, it is possible to logically couple CBS, TAS, and SP in series. This means in order for a frame to be transmitted, all three elements must allow the transmission at the exact same time and a frame must be available for transmission.

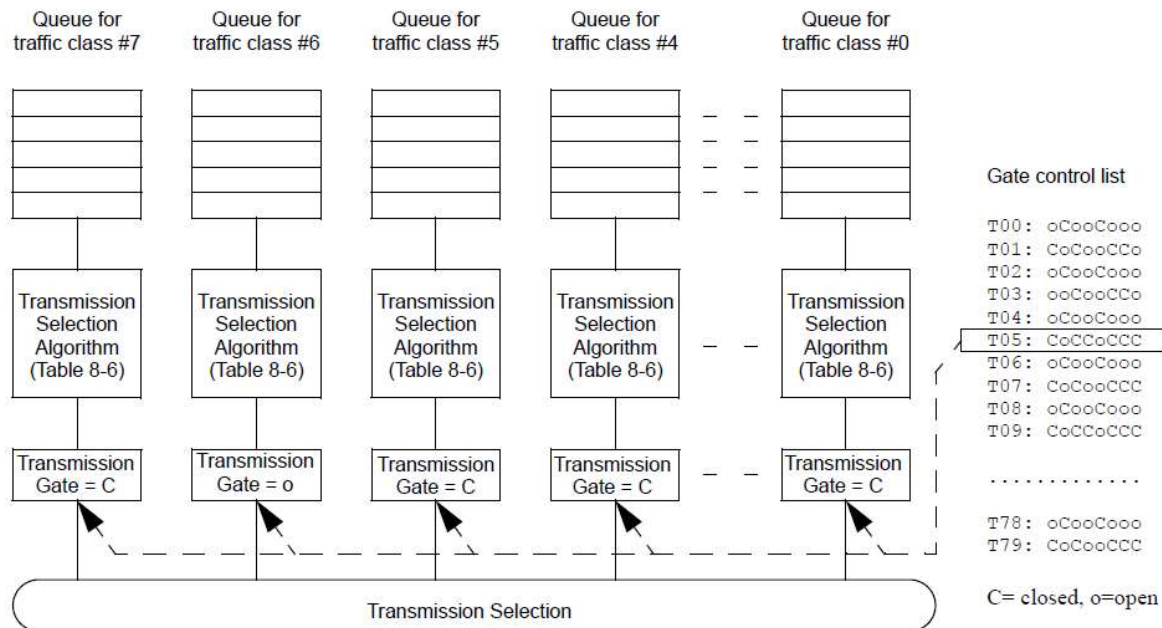


Figure 8-14—Transmission selection with gates

The TAS gate is completely unaware of the state of the queue (is a frame available?). It is controlled by a clock cycle, where two basic patterns of control are discussed below.

The CBS has a (dynamically) configured bandwidth expressed as the `operIdleSlope`, which accumulates credit only if a frame is in the queue and the TAS gate is open.

- g) *sendSlope*. The rate of change of *credit*, in bits per second, when the value of *credit* is decreasing (i.e., while *transmit* is TRUE). The value of *sendSlope* is defined as follows:

$$\text{sendSlope} = (\text{idleSlope} - \text{portTransmitRate})$$
- h) *transmitAllowed*. Takes the value TRUE when the *credit* parameter is zero or positive; FALSE when the *credit* parameter is negative.

The ATS determines a transmission time based on the arrival time of the frame, potentially using a different time base than the TAS.

3 Time Aware Shaper (TAS)

3.1 TAS Bus Mode

In what this document refers to as the Bus Mode, the gate control of all ports on a LAN are controlled by a single Gate Control List, whose Gate Open Times and Gate Open Duration are identical. This means the whole LAN is reserved for the transmission of a single set of messages. It is called Bus Mode as it resembled the operation of a Bus systems with a TDMA schema. In Bus Mode care must be taken to choose the Gate Open Duration long enough for the frame to propagate the maximum number of hops required. The Gate Open Duration must be at least the Latency of the set of messages to be transferred. This will usually mean having to increase the Gate Open Duration when a hop is added to the network and thereby potentially changing the whole TAS Gate Schedule in each hop of the LAN. Clock synchronisation inaccuracies across all hops need to be taken into account when determining the schedule. As a rough approximation the Gate Open Duration will be at least the number of hops times the length of the set of messages to be transported. For

100Mbit/s link speeds this will likely be in the 100's of μs , for 1Gbit/s link speeds this will likely be in the 10's of μs . These numbers are given to give an indication on the required resolution of the required time synchronization.

The number of such protected gate times is assumed to be limited as the whole LAN is blocked for the duration. If Bus Mode is allowed, the Gate Control List shall have at least 8 entries with a resolution of at least $10\mu\text{s}$.

While the frames in question are allowed to pass undisturbed through the LAN, it must be considered that this time is forced onto other frames as delay.

3.2 TAS Phased Mode

In what this document refers to as the Phased Mode, the gate control of each port a set of messages traverses is timed to allow the transmission without any queuing delay after reception. In a daisy chain this means the port gate of the next hop opens about the transmission latency plus the store and forward latency later than on the previous hop. The naming has been derived from phased traffic lights, which allow vehicles to get to the next intersection when the traffic lights are green. This requires exact a priori knowledge of the path the messages to be protected will travel along. It also means that any minute change in the transmission timing needs to be considered along the full path and will usually lead to a recalculation of the schedule for each hop. To increase the net usage of any egress port it will also require more accurate time resolution than Bus Mode.

The number of such protected gate times is assumed to be high compared to Bus Mode. If Phased Mode is allowed, the Gate Control List shall have at least 32 entries with a resolution of at least 64 byte times of the line rate of the port in question ($64 \times 8\text{bit}/100\text{Mbit/s} = 5\mu\text{s}$, $64 \times 8\text{bit}/1\text{Gbit/s} = 0.5\mu\text{s}$).

3.3 TAS Blockage Protection

As TAS will check for the length of a frame to ensure it can be transmitted in the allocated gate open cycle, it must be ensured that no frame with a length larger than the maximum gate open time within a Gate Control List is allowed in the egress queue. As such a mismatch could constitute an unrecoverable blockage of the port, it is required for any node implementing TAS to also implement a policer on frame length.

Note: IEEE Std. 802.1Q section 8.6.8.4 states: "frames that exceed queueMaxSDU are discarded [item b8] in 6.5.2]."

NOTE in 802.1Q section 8.6.9.2.2: SetGateStates(): A frame is not sent when TX can't be completed before the gate closes. "NOTE—If the OperGateStates value differs from the previous gate states, the SetGateStates() procedure causes gateopen and/or gate-close events to occur (3.97, 3.98). It is possible that, on a given queue, the maximum time interval between any gate-open event and a subsequent gate-close event is smaller than the value of queueMaxSDU, in which case, frames that are too long to transmit but are shorter than queueMaxSDU could be queued on that queue. Such frames would never be transmitted, but would eventually be discarded because they exceed the maximum frame lifetime (6.5.2, 6.5.6). It should also be noted that a Bridge is allowed to discard frames that could never be transmitted [item b8] in 6.5.2]."

4 Buffer Overrun Protection

Misconfiguration of any TSN shaping algorithm may lead to a (slow) build up of residual queue occupancy. It is required for any node to implement a watchdog on queue occupancy, which is able to clear the queue (loss of data!) in case it reaches a threshold occupancy. While this will lead to the loss of data, it prevents the overall flow of data from being completely stopped.

The maximum occupancy threshold should be set at a few 10kbyte, given a regular max. Size Ethernet frame has 1.5kByte.

Further action and error recording of such an event is left to the implementer.

Note: IEEE Std. 802.1Q section 6.5.6 suggests a Maximum Bridge transit delay of 1 to 4 seconds. At 100Mbit/s line rate, this could require a buffer of up to 50Mbyte.

5 Combination of CBS and TAS

5.1 No-Interference Stability Condition

Assuming there is no interfering traffic for a CBS controlled queue because the TAS gate is open for only this one queue, there is a simple stability condition that needs to be met, in order to not overrun the buffer of the queue.

The CBS streams can deliver a maximum of $cbsBits = (operIdleSlope * OperCycleTime)$ during one TAS gate cycle. The CBS stream is assumed to be quantized in n Frames of equal length L . While the gate is open, $(Rate * GateOpenTime) \geq cbsBits$ can be transmitted.

From $(operIdleSlope * OperCycleTime) = (n * L)$ one can find an

$$n = \lceil (operIdleSlope * OperCycleTime) / L \rceil$$

and due to $(n * L) \leq (Rate * GateOpenTime)$, the configuration is stable if and only if:

$$\lceil (operIdleSlope * OperCycleTime) / L \rceil * L \leq (Rate * GateOpenTime) \dots\dots\dots(5.1-1)$$

Defining an average bandwidth (idleSlope) for which the same number of bits can be transmitted during GateOpenTime as will be received on average during OperCycleTime, leads to:

$$operIdleSlope * OperCycleTime = idleSlope * GateOpenTime \dots\dots\dots(5.1-2)$$

The following external parameters are associated with each queue that supports the operation of the credit-based shaper algorithm:

- c) **portTransmitRate**. The transmission rate, in bits per second, that the underlying MAC Service that supports transmission through the Port provides. The value of this parameter is determined by the operation of the MAC.
- d) **idleSlope**. The rate of change of *credit*, in bits per second, when the value of *credit* is increasing (i.e., while *transmit* is FALSE and the transmission gate for the queue is open [8.6.8.4]). The value of *idleSlope* can never exceed *portTransmitRate*. If the enhancements for scheduled traffic (8.6.8.4) are not supported, or if GateEnabled is FALSE (8.6.9.4.14), the value of *idleSlope* for a given queue is equal to the value of the *operIdleSlope(N)* parameter for that queue, as defined in 34.3. If the enhancements for scheduled traffic (8.6.8.4) are supported, and GateEnabled is TRUE (8.6.9.4.14), then

$$idleSlope = (operIdleSlope(N) \times OperCycleTime / GateOpenTime)$$

where OperCycleTime is as defined in 8.6.9.4.20 and GateOpenTime is equal to the total amount of time during the gating cycle that the gate state for the queue is Open.

NOTE 1—When scheduled traffic operation is enabled, *credit* is accumulated only while the gate is open; therefore, the effective data rate of the *idleSlope* is increased to reflect the duty cycle for the transmission gate associated with the queue; however, the value of *operIdleSlope(N)* for the queue remains unchanged.

As per the CBS bandwidth definition the reservation was done according to:

$$operIdleSlope = cbsBits / observationInterval \dots\dots\dots (5.1-3)$$

$$operIdleSlope = (operIdleSlope * OperCycleTime) / observationInterval$$

Forcing:

$$OperCycleTime = observationInterval \dots\dots\dots (5.1-4)$$

The egress port will actually be busy transmitting for:

$$\begin{aligned} tAllStreams &= cbsBits / Rate = (operIdleSlope * observationInterval) / Rate \\ &= (operIdleSlope * OperCycleTime) / Rate = (idleSlope * GateOpenTime) / Rate \dots\dots\dots (5.1-5) \end{aligned}$$

during GateOpenTime.

5.1.1 Burst Operation

For the special case of:

$$tAllStreams = GateOpenTime \dots\dots\dots (5.1-6)$$

$$tAllStreams = (operIdleSlope * OperCycleTime) / Rate$$

$$operCycleTime = Rate * tAllStreams / operIdleSlope \dots\dots\dots (5.1-7)$$

$$TASBand = Rate * GateOpenTime / OperCycleTime \dots\dots\dots (5.1-8)$$

$$operCycleTime = TASBand / (GateOpenTime / OperCycleTime) * (tAllStreams / operIdleSlope)$$

$$operIdleSlope = TASBand \dots\dots\dots (5.1-9)$$

The GateOpenTime allocated in the TAS results in an identical bandwidth to what is reserved by operIdleSlope, the output will be a single burst while the TAS gate is open, no other gate can be allowed to be open and no effect of the CBS will be visible to the receiver on the link.

individual contribution, to be presented 2021-06-29 – updated since 2021-06-01 with input from inside and outside the working group

For this burst operation case the CBS shall be disabled!

5.1.2 Singular Operation

Still assuming the queue in question is the only one allowed to transmit, but the bandwidth allowed by the TAS gate timing is exactly the average bandwidth (idleSlope) to fulfil the stability criterion of (5.1-2), meaning:

$$\text{TASBand} = \text{idleSlope} \dots\dots\dots (5.1-10)$$

$$\text{TASBand} = \text{Rate} * (\text{GateOpenTime} / \text{OperCycleTime})$$

$$\text{idleSlope} * \text{GateOpenTime} = \text{operIdleSlope} * \text{OperCycleTime}$$

$$\text{Rate} * (\text{GateOpenTime} / \text{OperCycleTime}) = \text{operIdleSlope} * (\text{OperCycleTime} / \text{GateOpenTime})$$

$$\text{operIdleSlope} / \text{Rate} = (\text{GateOpenTime} / \text{OperCycleTime})^2$$

from (5.1-4): $\text{OperCycleTime} = \text{observationInterval}$

$$\text{GateOpenTime} = \sqrt{\text{operIdleSlope} / \text{Rate}} * \text{observationInterval} \dots\dots\dots (5.1-11)$$

Leads to a fixed configuration of the TAS, based on the CBS parameters.

5.2 Multi Gate Operation

Section 5.1 and its subsections make two assumptions: They ignore any interfering traffic, as only one TAS gate is open and they assume a single observationInterval, meaning a single SR-Class for all CBS controlled queues.

If at least one more gate is open along with the CBS controlled queue in question, it is important to configure the gate open time to allow for any interfering traffic along with the bandwidth reserved for CBS in this queue. This requires:

$$\text{TASBand} > \text{idleSlope}$$

$$\text{GateOpenTime} > \text{tAllStreams} * \text{Rate} / \text{idleSlope}$$

$$\text{Max Latency} = t_{\text{Device}} + t_{\text{MaxPacketSize+IPG}} + (t_{\text{AllStreams}} - t_{\text{StreamPacket+IPG}}) \times \text{Rate}/\text{MaxAllocBand} + t_{\text{StreamPacket}}$$

where

t_{Device} = the internal delay of the device (in increments of 512 bit times)

NOTE 3— t_{Device} is an integral multiple of 512 bit times so that it scales with the speed of the media.

$t_{\text{MaxPacketSize+IPG}}$ = the transmission time for a maximum size interfering frame (1522 octets to 2000 octets) plus its preamble and start of frame delimiter (SFD) (8 octets), and the following inter-packet gap (IPG) (12 octets)

$t_{\text{StreamPacket}}$ = the transmission time for the maximum frame size of the stream that is being reserved, plus its preamble and SFD (8 octets)

$t_{\text{StreamPacket+IPG}}$ = the transmission time for the maximum frame size of the stream that is being reserved, plus its preamble and SFD (8 octets) and the following IPG (12 octets)

Rate = the transmission rate of the port

MaxAllocBand = maximum allocatable bandwidth, the maximum amount of bandwidth the AVB system is able to allocate for Class A streams on the port

t_{Interval} = the Class A observation interval or 125 μs

$t_{\text{AllStreams}} = (\text{MaxAllocBand} \times t_{\text{Interval}}) / t_{\text{Rate}}$ = the sum of the transmission times of all Class A stream frames the AVB System is able to allocate in an observation interval (125 μs) on a port

gives a lower limit for GateOpenTime:

$$\text{GateOpenTime} \geq t_{\text{Device}} + t_{\text{MaxPacketIPG}} + (t_{\text{AllStreams}} - t_{\text{StreamPacketIPG}}) * \text{Rate} / \text{MaxAllocBand} + t_{\text{StreamPacket}} \dots \dots \dots (5.2-1)$$

With:

$$\text{MaxAllocBand} = \text{idleSlope} \dots \dots \dots (5.2-2)$$

determining the temporal spread of packets due to CBS.

$$\text{operIdleSlope} * \text{OperCycleTime} / \text{Rate} = \text{idleSlope} * \text{GateOpenTime} / \text{Rate}$$

$$\text{Rate} / \text{idleSlope} = (\text{Rate} * \text{GateOpenTime}) / (\text{operIdleSlope} * \text{OperCycleTime})$$

$$\text{GateOpenTime} \geq t_{\text{Device}} + t_{\text{MaxPacketIPG}} + t_{\text{StreamPacket}} + (t_{\text{AllStreams}} - t_{\text{StreamPacketIPG}}) * (\text{Rate} * \text{GateOpenTime}) / (\text{operIdleSlope} * \text{OperCycleTime})$$

$$\text{GateOpenTime} * (1 - (t_{\text{AllStreams}} - t_{\text{StreamPacketIPG}}) * \text{Rate} / (\text{operIdleSlope} * \text{OperCycleTime})) \geq t_{\text{Device}} + t_{\text{MaxPacketIPG}} + t_{\text{StreamPacket}}$$

$$\text{GateOpenTime} \geq (t_{\text{Device}} + t_{\text{MaxPacketIPG}} + t_{\text{StreamPacket}}) / (1 - ((t_{\text{AllStreams}} - t_{\text{StreamPacketIPG}}) * \text{Rate} / (\text{operIdleSlope} * \text{OperCycleTime}))) \dots \dots \dots (5.2-3)$$

Yields a configuration limit for given OperCycleTime and Rate as well as device, interference, and CBS parameters.

The denominator of (5.2-3) can be further analysed:

$$((t_{\text{AllStreams}} - t_{\text{StreamPacketIPG}}) * \text{Rate}) < (\text{operIdleSlope} * \text{OperCycleTime})$$

$$\text{OperCycleTime} / \text{Rate} > (t_{\text{AllStreams}} - t_{\text{StreamPacketIPG}}) / \text{operIdleSlope}$$

individual contribution, to be presented 2021-06-29 – updated since 2021-06-01 with input from inside and outside the working group

$$\text{OperCycleTime} / \text{Rate} > (\text{cbsBits} - \text{streamPacketIPGBits}) / (\text{operIdleSlope} * \text{Rate})$$

$$\text{OperCycleTime} > (\text{cbsBits} - \text{streamPacketIPGBits}) / \text{operIdleSlope} \dots\dots\dots(5.2-4)$$

This yields only a very weak requirement, but shows that one can now choose OperCycleTime in (5.2-3) independently from the SR-Class (observationInterval), but only dependent on the amount of data and the average bandwidth required.

Since $\text{idleSlope} \leq \text{Rate}$,

$$\text{Rate} \geq \text{operIdleSlope} * (\text{OperCycleTime} / \text{GateOpenTime}) \dots\dots\dots(5.2-5)$$

Can be expressed as:

$$\text{operCycleTime} / \text{Rate} \leq \text{GateOpenTime} / \text{operIdleSlope} \dots\dots\dots(5.2-6)$$

Gives a configuration criterion which will be used further in section 5.3.

5.2.1 Contiguous Control List Entries

Equation (5.2-3) only holds, if the CBS queue can transmit its frames at the intervals defined and especially directly after an interfering frame. As the TAS gate may decline transmission of a frame if it can not be completed within the current 'open' interval (see section 3.3), for any queue controlled by CBS and TAS, the TAS gate control entries shall be configured in such a way that the gate for the CBS controlled queue only opens and closes at most once per TAS cycle. This means the GateOpenTime can not be fragmented across the OperCycleTime, but must be one contiguous time block, even if it extends over multiple control list entries. Otherwise an interfering frame would have to be considered for every 'open' gate control entry and the total GateOpenTime would have to increased significantly.

5.3 Maximum CBS bandwidth with TAS gates enabled

IEEE Std. 802.1Q suggests to only reserve a maximum of 75% of a port's bandwidth for CBS (see 5.3.1 for a rationale), meaning $\text{operIdleSlope} \leq (75\% * R)$, which was equivalent to $(\text{tAllStreams} / \text{observationInterval}) \leq 75\%$ when operating without the TAS gates.

With the TAS gates enabled one still can start the configuration at:

$$\text{operIdleSlope} = \text{cbsBits} / \text{observationInterval} \dots\dots\dots(5.3-1)$$

From (5.2-6):

$$\text{operIdleSlope} \leq \text{Rate} * (\text{GateOpenTime} / \text{OperCycleTime})$$

One can postulate:

$$(\text{GateOpenTime} / \text{OperCycleTime}) \geq \text{operIdleSlope} / (75\% * \text{Rate}) \dots\dots\dots(5.3-2)$$

This condition can be achieved in two ways:

- 1) Either $(\text{GateOpenTime} / \text{OperCycleTime})$ is set to fulfil the desired condition through initial configuration,
- 2) or the $(\text{GateOpenTime} / \text{OperCycleTime})$ must be adjusted for every new (de-)reservation of a CBS stream, e.g. based on the information delivered by SRP.

This document strongly suggests for the implementer to configure $(\text{GateOpenTime} / \text{OperCycleTime})$ to the maximum required bandwidth in order to prevent complex re-configuration at runtime.

Using (5.1-8) one can also express (5.3-2) as:

$$\text{operIdleSlope} \leq 75\% * \text{TASBand} \dots\dots\dots(5.3-3)$$

and using (5.2-2) and (5.1-2) results in:

$$\text{MaxAllocBand} = 75\% * \text{TASBand} * (\text{OperCycleTime} / \text{GateOpenTime})$$

$$\text{MaxAllocBand} = 75\% * \text{Rate} \dots\dots\dots(5.3-4)$$

5.3.1 One rational for 75% CBS limit

There are multiple more or less arbitrary ways to argue for the limit stated in IEEE Std. 802.1Q section 34.3.1: “The recommended default value of *deltaBandwidth(N)* for the highest numbered traffic class supported is 75%”.

From the automotive perspective one can assume 100Mbit/s links at the time this statement was drafted. On such a link a 1500Byte max. Ethernet Frame takes about 120usec to transmit (ignoring e.g. IPG for now). The observation interval for SR-Class A is 125usec (driven from the audio use-case), which was at the time presumed to be pervasive.

Doing a very rough approximation one can assume: $t_{\text{MaxFrame}} \approx \text{observationIntervalA}$ (at 100Mbit/s), leading to: $\text{observationIntervalA} * (1 - 75\%) * 4 \approx t_{\text{MaxFrame}}$

Meaning an interfering max. Frame can be “made up” within a burst of less than 4 observation Intervals ($t_{\text{MaxFrame}} < \text{observationIntervalA}$), equivalent to 4 max. Frame sizes and translating into a $4 * 1500\text{byte} = 6\text{kbyte}$ maximum buffer requirement for a transmit or receive queue. This was rather arbitrarily considered an acceptable limit.

In further discussions an SR Class C of 1333usec ([AVnu AVB Spec](#)) was found to be more efficient.

Here $75\% * 1333\text{usec}$ means that $\text{observationIntervalC} * (1 - 75\%) \approx 3 * t_{\text{MaxFrame}}$ (still on 100Mbit/s links) allows for “make up” of an interfering Frame within the observation interval as well as the use of more than one max. Frame in the CBS shaped streams, which is important for video transport.

5.3.2 An alternate CBS bandwidth limit

One could also have postulated:

$$\text{idleSlope} \leq 75\% * \text{TASBand} \dots\dots\dots(5.3.2-1)$$

$$\text{operIdleSlope} * (\text{OperCycleTime} / \text{GateOpenTime}) \leq 75\% * \text{Rate} * (\text{GateOpenTime} / \text{OperCycleTime})$$

$$\text{operIdleSlope} \leq 75\% * \text{Rate} * (\text{GateOpenTime} / \text{OperCycleTime})^2$$

$$(\text{GateOpenTime} / \text{OperCycleTime})^2 \geq \text{operIdleSlope} / (75\% * \text{Rate}) \dots\dots\dots(5.3.2-2)$$

This is obviously a much stronger restriction than the result in (5.3-2), as $(\text{GateOpenTime} / \text{OperCycleTime}) < 1$, $(\text{GateOpenTime} / \text{OperCycleTime})^2 << (\text{GateOpenTime} / \text{OperCycleTime})$.

Without further arguments there seems no reason to restrict the CBS bandwidth further than what (5.3-2) requires.

5.4 Maximum CBS Latency with TAS gates enabled

The maximum latency given by IEEE Std 802.1BA-2011 page 15:

$$\text{maxLatencyBA} = t_{\text{Device}} + t_{\text{MaxPacketIPG}} + t_{\text{StreamPacket}} + (\text{tAllStreams} - t_{\text{StreamPacketIPG}}) * \text{Rate} / \text{MaxAllocBand} \dots\dots\dots(5.4-1)$$

should be modified to allow for different influences.

5.4.1 Gate Open Delay

The GateOpenDelay is determined by the time it takes for the TAS gate to open, which is given by:

$$\text{GateOpenDelay} = \text{OperCycleTime} - \text{GateOpenTime} \dots\dots\dots (5.4-2)$$

if Contiguous Control List Entries are observed. It should be added to (5.4-1), if TAS is combined on a queue with CBS and synchronous frame arrival can not be guaranteed.

If TAS gates are synchronized across the network as described in sections 3.1 and 3.2 it is important to ensure sufficient timing accuracy and/or guard bands in order to prevent this effect.

5.4.2 Over Provisioning

This aspect is not technically related to the combination of CBS and TAS, but may be used here as well due the likely static configuration suggested by (5.3-2).

IEEE Std 802.1BA-2011 page 15 assumes:

$$\text{tAllStreamsBA} = \text{MaxAllocBand} * \text{observationInterval} / \text{Rate} \dots\dots\dots (5.4.2-1)$$

Meaning the bandwidth reserved for CBS (operIdleSlope) is identical with the bandwidth required by the streams transmitted (cbsBits / observationInterval, as used in (5.3-1)).

If one instead configures:

$$\text{operIdleSlope} > \text{cbsBits} / \text{observationInterval} \dots\dots\dots (5.4.2-2)$$

one can use (5.3-4) and

$$\text{tAllStreams} = \text{cbsBits} / \text{Rate}$$

to get:

$$(\text{tAllStreams} - \text{tStreamPacketIPG}) * \text{Rate} / \text{MaxAllocBand} = (\text{cbsBits} - \text{streamPacketIPGBits}) / (75\% * \text{Rate}) \dots\dots\dots (5.4.2-3)$$

Which can lead to a significant reduction in latency for the CBS streams while other data flows are not withheld from using the excess bandwidth allocated.

5.4.3 Worst case CBS Latency with TAS gates enabled

Taking the worst case for sections 5.4.1 and 5.4.2 into account, the worst case latency can be expressed as:

$$\begin{aligned} \text{wcLatencyCBSTAS} = & \text{tDevice} + \text{tMaxPacketIPG} + \text{tStreamPacket} + (\text{OperCycleTime} - \text{GateOpenTime}) + \\ & ((75\% * \text{Rate} * \text{GateOpenTime}) - \text{streamPacketIPGBits}) / (75\% * \text{Rate}) \dots\dots\dots (5.4-4) \end{aligned}$$

with streamPacketIPGBits = tStreamPacket * Rate.

5.5 Conclusions on CBS and TAS gate combination

Section 5.1 shows that without interference the configuration of the TAS is forced by the CBS configuration.

If interference is allowed, equation (5.3-2) made clear, that a dynamic configuration, e.g. by using SRP, of the combined CBS with TAS gates system is not desirable, as CBS reservation changes would also result in TAS gate schedule changes.

Equation (5.4-4) shows a clear need to have synchronized TAS gate opening times, in order to prevent (OperCycleTime – GateOpenTime) to be added to the latency.

It is also important to point out, that:

$$t_{\text{MaxPacketIPG}} < (\text{cbsBits} - \text{streamPacketIPGBits}) / (75\% * \text{Rate}) \dots\dots\dots (5.5-1)$$

in may cases, particularly for video transport, where an SR-Class with observationInterval > 1ms will likely be used and where cbsBits is potentially an integer multiple of (tMaxPacketIPG * Rate).

$$\begin{aligned} \text{maxLatencyCBSTAS} = \\ t_{\text{Device}} + t_{\text{MaxPacketIPG}} + t_{\text{StreamPacket}} + (\text{OperCycleTime} - \text{GateOpenTime}) + \\ (\text{cbsBits} - \text{streamPacketIPGBits}) / (\text{idleSlope}) \dots\dots\dots (5.5-2) \end{aligned}$$

This also shows, that designing a combined CBS/TAS queue to meet a certain latency will be difficult, as it results in a somewhat circular relation, since GateOpenTime determines idleSlope, which in turn influences MaxLatency, which puts a lower limit on GateOpenTime (even ignoring the GateOpenDelay).

Equations (5.2-3) and (5.2-6) can be used to find a static configuration, but are hardly suitable to allow for dynamic changes during run-time.

5.5.1 Alternate approach to CBS with TAS gates

A possible solution could be to choose a configuration where:

$$\text{OperCycleTime} \approx \text{GateOpenTimeCBS} \dots\dots\dots (5.5.1-1)$$

Meaning the gate for the CBS queue is closed only for a small time during the OperCycleTime.

This would lead to:

$$\text{idleSlope} \approx \text{operIdleSlope} \dots\dots\dots (5.5.1-2)$$

If (OperCycleTime – GateOpenTimeCBS) ≈ tMaxPacketIPG, taking (5.5-1) into account and further simplifying tStreamPacket ≈ tMaxPacketIPG and cbsBits ≈ (n * tMaxPacketIPG / Rate)

$$\begin{aligned} \text{maxLatencyALT} = \\ t_{\text{Device}} + 3 * t_{\text{MaxPacketIPG}} + (n - 1) * t_{\text{MaxPacketIPG}} / (\text{Rate} * \text{idleSlope}) \dots\dots\dots (5.5.1-3) \end{aligned}$$

The GateOpenDelayCBS = GateOpenTimeTAS, which is the time the gate is open for in this case a single other TAS stream, has now been reduced to just another interfering frame for the CBS stream.

For the TAS stream GateOpenTimeTAS << OperCycleTime, meaning it can suffer a significant delay GateOpenDelayTAS = (OperCycleTime – GateOpenTimeTAS) if it arrives out of sync. Since GateOpenTimeTAS is now in the order of a max. Frame (tMaxPacketIPG), this may lead to even more stringent requirements on the time-synchronisation, especially on high line rate links.