Delay Analysis of AVB traffic in Time-Sensitive Networks (TSN) with application to multi-packet Video Frames

Dorin Maxim, Ye-Qiong Song

Delay Analysis of AVB traffic in Time-Sensitive Networks (TSN) with application to multi-packet Video Frames

Dorin Maxim and Ye-Qiong Song

LORIA, 615 Rue du Jardin Botanique, 54600 Villers-lès-Nancy, France

Abstract

Future autonomous vehicles and ADAS (Advanced Driver Assistance Systems) need real-time audio and video transmission together with control data traffic (CDT). Although audio/video packet delay analysis has been largely investigated in AVB (Audio Video Bridging) context, it needs to be further extended with the presence of the CDT in the new TSN context and also extended to the response time analysis of large-size video frames which are often transmitted in multiple TSN Ethernet packets. In this paper we first present a local delay analysis of AVB packets under hierarchical scheduling of credit-based shaping and time-aware shaping on TSN switches. We analyze the effects of time aware shaping on AVB traffic, how it changes the relative order of transmission of packets leading to bursts and worst case scenarios for lower priority AVB streams. We also show that these bursts are upper-bounded by the Credit-Based Shaper, hence the worst-case transmission delay of a given stream is also upper-bounded. We present the analysis to compute the worst case delay for a packet, as well as the feasibility condition necessary for the analysis to be applied. Furthermore, we extend our analysis to the case of realistic video frames where each frame (image) needs to be split into several TSN packets and re-assembled at the destination node. The analysis of such video frames has an extra level of complexity as the entire video frame needs to be transmitted within a given time limit, as opposed to just each packet having to respect a deadline. Additionally, a video frame may be transmitted over several cycles of the Time Aware Shaper and this is the first work, to the best of our knowledge, that considers such a model. Our methods (analysis and simulation) are applied to an automotive use case, which is defined within the Eurostars RETINA project, and where both control data traffic and AVB traffic must be guaranteed. For the case of multi-packet video streams we have analyzed a large amount of randomly generated stream sets in order to assess the pessimism of the analysis and the variation in feasibility with respect to the characteristics of the stream sets.

Keywords: In-vehicle network, Time Sensitive Networking, AVB traffic, Delay Analysis, Credit-Based Shaper, Time Aware Shaper

1. Introduction

IEEE 802.1 TSN (Time-Sensitive Networking) [1] is the most important next evolution of switched Ethernet technology for time critical in-vehicle and process control networks. It comes from Ethernet AVB by adding several new features such as precise time synchronization through IEEE 802.1AS-2011 [2] (a specific profile of IEEE 1588 PTP) and TAS (Time-Aware Shaper) for supporting both hard real-time and soft real-time constraints.

Enabled by the precise time synchronization, a key new feature of TSN is the definition of the new traffic shaping mechanism TAS, which is capable of accommodating hard real-time

streams with deterministic end-to-end delays. TAS uses a pre-defined TDMA-like scheduler that guarantees timely transmission of the most critical CDT (Control Data Traffic). Furthermore, a *guard band* is added before any scheduled CDT window. Guard band and CDT forms together a *protected window* for ensuring that non-CDT such as audio, video and best effort traffic will not interfere with the CDT. The TAS defines a fixed and periodic schedule cycle of time slots, which specifies when a traffic queue is enabled (opened) or disabled (closed) for transmission during its time slot. This schedule is configured offline. Opening/closing is implemented by associating each queue of the output port (up to 8 queues) with a time-aware transmission gate. A TAS cycle represents a complete CDT transmission period, which is 500μs according to IEEE 802.1 TSN for accommodating to time-critical applications [1], [3].

Although the offline building of the TDMA schedule could be complex, CDT can always be guaranteed by using TAS [4], [5], [6] if there is enough bandwidth. Nevertheless, the use of TAS has a direct impact on the delay of the other TSN traffic classes and especially on the AVB traffic as it has to meet some real-time constraints. So the previous delay analysis of the AVB without TAS impact cannot be directly applied to it in this new TSN context.

In this paper we mainly focus on the delay analysis of the AVB traffic taking into account the impact of TAS in the TSN context. This is motivated by the automotive application use case defined within Eurostars RETINA project[1] by extending the first use case of [3], where video streams are main bandwidth consumers for autonomous driving and active safety.

In our model, after traffic classification according to eight IEEE802.1Q priorities, each audio or video stream is first constrained (or shaped) by a CBS (Credit-Based Shaper), then transmitted during a non-CDT slot which is shared by all traffic that is non-CDT (audio, video and best effort) according to their priorities. The main purpose of applying a traffic shaper like CBS to those non-CDT classes is to guarantee a necessary bandwidth to each of them without starving the lower priority traffic.

In CBS, the credit increases with an *idle slope* (i.e. reserved bandwidth) when packets are queued and decreases with a *send slope* (i.e. the total link bandwidth minus the reserved bandwidth) during its packet transmissions. In TSN, a CBS shaped AVB packet can only be sent outside of the CDT windows and if its credit is not negative. This operating mode for transmitting AVB traffic is called hereafter *CBS+TAS model*. It arises two additional issues for real-time guarantee of AVB traffic: how to determine the necessary bandwidth (via the idle slope) to guarantee deadlines and how to accurately (i.e. with minimum pessimism) analyze the worst-case delay for a given AVB stream with its reserved bandwidth.

Delay analysis of CBS shaped AVB traffic without TAS has been extensively studied using different worst-case delay analysis approaches: network calculus [7], busy period based real-time schedulability analysis [8] and trajectory analysis [9], and eligible interval in [10] and [11] that provides tight delay bounds. We can also find some work related to the delay analysis of TSN with other shapers than CBS for AVB, such as with peristaltic shaper in [12] and Burst limiting shaper [13]. Another work deals with AVB ST model [14], which is quite similar to our CBS+TAS model. However as will be explained later, several differences exist in these two models, making the result in [14] inapplicable to the CBS+TAS model.

So it is necessary to develop an analysis on this new CBS+TAS model in TSN context for making practical use of audio and video streaming in autonomous driving. In [15] we have extended, for the first time, the *Eligible Interval* approach of [10] and [11] to the delay analysis of the CBS constrained AVB traffic under TAS constraint. Later on, this same CBS+TAS model has also been analyzed in [16] using network calculus approach, first for the case of two

[1] `http://retinaproject.eu/`

AVB classes (priorities), and then further improved in [17] by extending the analysis to more than two AVB classes and by reducing the pessimism with the definition of new arrival and service curves. We note that in [18], extending the eligible interval approach to the CBS+TAS model is also indicated as future work.

Most of the state of the art analyses only focus on per Ethernet packet delay analysis, with numerical applications to only small size examples. However practical delay constraint is expressed on the large size video frames (images) which need to be split into several Ethernet packets to be transmitted in TSN. This practical issue should also be carefully examined during automotive application design step [19].

Contributions: The main contribution of this paper is to give a comprehensive analysis of AVB traffic under hierarchical CBS+TAS scheduling, knowing that the transmission of these packets is influenced by two shapers (directly by CBS and indirectly by TAS) as well as by their priorities (with respect to other AVB classes) and by FIFO ordering with respect to packets from other streams but sharing the same AVB class output queue. For this, we describe the impact of TAS over AVB streams showing how TAS influences the relative order of transmission of AVB streams, leading to bursts of packets and worst case-transmission delays for lower priority streams. Further extension of [15] includes a formal proof of the worst-case delay of CBS shaped AVB streams under TAS impact as well as new CBS credit assignment rules that take into account the TAS occupied bandwidth [2]. We also extend our analysis to take into account realistic video frames that are split into multiple packets and their transmission spans over several TAS cycles. We apply our analysis framework to an automotive inspired use-case and compare the results of the analysis with simulation results, showing that the analysis is safe for all cases that we have considered and the tightness varies depending on the load of the TSN switch and the complexity of the TAS scheduling table (also called gate control lists).

We note that this paper mainly focuses on providing tight local delay bounds, based on the eligible interval approach. How to also obtain tight end-to-end delay bound using eligible interval analysis is still an open issue that we leave as future work. But we believe that providing tight local delay bound is of paramount importance towards obtaining tight end-to-end delay bounds, since the remaining key issue is to take into account the serialization effect of packets in a multi-hop context and in this way reducing the pessimism generated by considering that the worst case arrival combinations occurs in each switch through which a flow passes. Furthermore, we emphasize the fact that our local analysis has its place in the toolbox of a network designer, as it can be used to (1) take a closer look at congested nodes in the network and more appropriately distribute the traffic to balance the load and (2) appropriately compute/verify the *idleSlope* and *sendSlope* of the AVB traffic to optimize/balance their local transmission delays and implicitly the end-to-end delays.

Organization of the paper: in Section 2 we analyze the related work on which our own contribution is based. Sections 3 presents the model of the network that we are interested in analyzing, the effects of the time aware shaper on the AVB traffic and the formalization of the problem that we solve in the paper. Section 4 describes the first major contribution of our work, which is the worst case transmission delay analysis of CBS shaped AVB packets in a TSN switch, as well as the way to reserve necessary bandwidth in CBS considering the TAS occupied bandwidth. Section 5 presents the second major contribution of our work, an extension of the analysis to the case of realistic video frames that are split into multiple packets and transmitted over several TAS cycles. Section 6 gives the experiments that we have conducted to assess

[2]The formulas initially elaborated for AVB in IEEE802.1Qav in `www.ieee802.org/1/files/public/docs2009/av-fuller-queue-delay-calculation-0809-v02.pdf` could be re-adapted according to our CBS assignment rules

the efficiency of our analysis. These experiments include an automotive case study that we simulate and analyze with our proposed solution as well as extensive analyses of randomly generated Stream-Sets in order to assess how their feasibility varies when certain parameters of the Stream-Sets are varied. We conclude the paper in Section 7.

2. Related Work

Our work is in the context of the IEEE 802.1Qbv TSN standard [1], [20] which enhances Ethernet networks to support time sensitive applications, meant to be used in the automotive and industrial control domains. A key new feature of TSN is the definition of the new traffic shaping mechanism TAS (Time Aware Shaper), which is capable of accommodating hard real-time streams with deterministic end-to-end delays and jitters. TAS provides latency guarantees by using a pre-defined scheduler that guarantees timely transmission of Control Data Traffic (CDT). Furthermore, the existence of a mechanism called guard band is assumed that ensures that the traffic that is non-CDT will not interfere with control data traffic. The TAS defines a fixed and periodic schedule of time intervals (slots), which specifies when a traffic queue is enabled (opened) or disabled (closed) for transmission. This schedule (also called gate control list) is configured offline for each port of the switch. The presence of the CDT will impact of course the delay performance of other traffic classes, such as audio/video and best effort ones.

TSN standards being released recently, there are few research contributions in the domain of Time Sensitive Networks. One of the first papers in the domain is [3], which presents three traffic shapers which were under discussion in the IEEE802.1TSN working group, at the time when [3] was being written (autumn 2014): Time Aware Shaper (TAS), Burst Limiting Shaper (BLS) and Peristaltic Shaper (PS). Analyses are proposed for each of them, to compute the end-to-end latency of a packet over n hops. Evaluations are performed showing that: (1) only TAS is able to schedule the CDT within the maximal delays imposed by the standard; (2) BLS has delays larger by approximately 20% with respect to the limits; (3) PS has delays almost twice as large than those permitted by the standard. An improvement is proposed to the Peristaltic Shaper in the form of a Guard Band which reduces the delays, but not enough to make them less than what is permitted by the standard.

In the same period [12] proposed a formal worst-case analysis for the Time-Aware Shaper (TSN/TAS) and Peristaltic Shaper (TSN/PS). For each shaper two analyses are derived: (a) one analysis for the streams of packets that are controlled by the shaper itself and (b) one analysis for the streams that are not controlled by the shaper (the shaper also has an impact on these streams in terms of transmission delays). In the case of TSN/TAS it is considered that critical streams have link access (i.e. their gate is opened) during special time intervals that repeat periodically and in which no other stream is allowed to transmit. All non-critical streams transmit outside of these TAS intervals and need to compete for access. This assumptions is quite restrictive on the possible scheduling table that can be used on the network. Also, the credit of non-TAS streams is not considered, i.e. there is always enough credit to transmit. In [13] the same analysis has been applied to TSN with the Burst-Limiting Shaper. Their worst case delay evaluation is mainly based on the busy period analysis of the flow schedulability. We see that there was no study on CBS+TAS model.

Since then, two main research topics have been developed on TSN: one on TAS time slot scheduling with several challenging issues like the optimal scheduling that has minimal impact on the other traffic, joint time slot scheduling and path routing optimization, or still the clock drift tolerant slot time scheduling; another on providing tight upper bounds on the worst case delay of the other real-time traffic classes such as audio/video that are shaped by CBS and under interference of the TAS scheduled CDT. In fact, although TAS and CBS are interesting features

of TSN, their practical use is complex. So we must provide answers to the two corresponding questions on how to design the optimal TAS schedule for a given CDT and how to assign the CBS credits to guarantee the desired delay for different other real-time streams.

In this paper we only focus on analyzing the worst case delay bounds of AVB traffic classes under CBS and with interference of predefined CDT schedule. Inversely, for guaranteeing bounded transmission delays, we also provide rules for obtaining the necessary CBS credits.

As for the CDT scheduling problem, readers can refer to [4], [5], and [6].

For choosing a suitable worst case delay analysis method, we first review the different approaches that have been used for analyzing CBS shaped AVB traffic without the interference of the CDT (i.e. without the presence of TAS).

Delay analysis of CBS shaped AVB without TAS has been extensively studied using mainly three different worst-case delay analysis approaches: 1) network calculus [7], 2) busy period based real-time schedulability analysis [8] and trajectory analysis [9], and 3) recently developed eligible interval [10], [11], [18]. Among them, the eligible interval approach is particularly interesting as it provides the tightest delay bound with very low computing complexity [18].

The first contribution [11] presents the *Eligible Interval* analysis for AVB packets, in the case of either lower-priority interference or higher-priority interference upon the packet under analysis – the case of both types of interference is left as future work. An eligible interval is an interval in which a stream has pending load available and also has non-negative credit, i.e. the packets can be sent unless the output port is otherwise occupied. The analysis returns the worst-case response time, which is a tighter upper-bound with respect to the one returned by the busy period analysis [8].

Subsequently, [10] extends the eligible intervals analysis presented in [11] to take into account interference from both lower- and higher-priority traffic at the same time. The analysis does not rely on any assumptions on interfering priority classes other than those enforced in the Ethernet AVB standard [2], i.e. there is no need to know the arrival patterns or transmission requirements of the interfering traffic. This independence from detailed information of the traffic is achieved by taking advantage of the effect of CBS on the interfering traffic, i.e. replenishing (idle) and consumption (send) slopes of the credit.

In [18], the authors extended the eligible interval to a general case where several higher and lower priorities are considered. As all the higher priority flows have to be considered, the obtained bound is no longer guaranteed tight. However numerical comparisons with the busy period analysis of [8] still show its relative tightness. We note that extending the eligible interval approach from only CBS to the CBS+TAS model is also indicated as future work in [18].

For meeting deadline constraints, this result allows to easily evaluate the tight worst-case delays of the AVB traffic through one AVB switch for the given CBS idle and send slopes. In [21], the same authors have also proposed two algorithms to compute the minimum sufficient AVB bandwidth allocation (idle slope) for satisfying the deadline constraints. The idea is straightforward. The algorithms just impose that the delay is equal to the deadline to directly deduce which value of the corresponding idle slope (so bandwidth credit) is needed.

The *Eligible Interval* analysis was proven to be tight for AVB networks (for at least the case of two AVB classes), and it also gives a way to design CBS bandwidth allocation. For this reason we choose to extend it to the TSN case where interference from the control data traffic shaped by TAS must be considered.

In [15] we have extended, for the first time, the *Eligible Interval* approach of [10] and [11] to the delay analysis of CBS shaped AVB traffic under TAS constraint. In this paper we will further extend it to dealing with large size video streams.

The work in [14] presents a model very similar to the IEEE 802.1Qbv TSN, called AVB ST (AVB Scheduled Traffic). The main difference between TSN and AVB ST lies in the way

the protected windows are created for the CDT. While in TSN a scheduling table (gate control list) is used to define which traffic-gates are closed or opened at each moment in time, AVB ST does not use such a table. Instead AVB ST considers that each CDT packet (called scheduled traffic in this approach) is scheduled in a deterministic manner and the switch has complete knowledge of when each packet arrives and needs to be transmitted. Hence each CDT packet has its own protected window and implicitly a guard band before its arrival. The delay of AVB is then bounded considering that CDT impact can be modeled as length rate quotient or leaky-bucket, which is not exactly the case of TSN TAS shaped CDT. Another notable difference between AVB ST and TSN is that in TSN the credit of classes AVB_A and AVB_B does not increase when their transmission gates are closed by the time aware shaper, while in AVB ST their credit still increases (according to the relevant *idleSlope*) when ST packets are being transmitted. This means that the *Eligible Interval* analysis of [11] and [10] can not be applied on AVB ST as the arrival and transmission of an ST packet allow for the credits of AVB streams to evolve and may in this way change their transmission order [22]. Hence [14] uses the busy period approach, leading to more pessimistic results than what can be obtained with the *Eligible Interval* approach.

Recently, the same CBS+TAS model has also been analysed in [16] using network calculus approach to derive the worst case delays for AVB traffic in TSN, concentrating on the benefits of preemption over the non-preemptive case. The experimental results show that the preemption overheads are covered by the gain in bandwidth due to the reduction in the size of Guard-Bands. However the work presented in [16] introduces important pessimism and is limited to the case of only two CBS shaped traffic classes. In a recent technical report [17], the authors proposed an improvement by considering more than two CBS shaped classes and by reducing the pessimism with new aggregate arrival curves. The most significant improvement is that the aggregate arrival curves to an intermediate switch take into account both the physical link shaping and CBS shaping, preventing simultaneous arrival bursts. But their arrival curve assumes that it is not possible to transmit an AVB packet or a best effort packet during the guard band, while in the TSN standard and in our model, any packet that can finish its transmission before the end of the guard band can still be transmitted (which is the case when setting the length of the guard band to be equal to the transmission time of the largest lower priority packet). Considering the potentially large guard band (because of large lower priority packet sizes), ignoring it may also lead to considerable pessimism.

In our work we concentrate on the local analysis in the non-preemptive case, with an extension to the case when frames are split into several packets that are sent over several TAS cycles. We only consider the case of two CBS shaped AVB traffic classes as it is the case for many practical applications. Its extension to more than two CBS shaped traffic classes could be done by inspiring from the work in [18]. Moreover, how to provide tight end-to-end delay in multihop TSN using eligible interval is still an open issue that we leave as future work, and for this purpose the idea of considering physical link and CBS shaping to prevent arrival bursts to intermediate switches [17] may be helpful.

Another line of research that is relevant to our work is the transmission over the network of large video frames that are split into several Ethernet TSN packets. To the best of our knowledge, the only contribution in this direction currently published is [19] which looks at End-to-End delays of multi-packet frames in a TSN-like network (i.e. it has most characteristics of TSN except for the guard-band that should precede the transmission of Control Data Traffic). As [19] is an industrial paper, it does not provide details for the theoretic delay analysis used nor the implementation of the simulation tool. Nevertheless, the paper presents valuable insight on the transmission of segmented frames over TSN. The theoretical delays are computed using Network Calculus (NC), without giving details on the particularities and the modifications done

to the state of the art analysis to make it applicable for TSN. The modeling and simulation results are obtained with the RTaW-Pegase tool, a commercial product of Real-Time-at-Work company. A particularity of [19] is the fact that the video frames are split in 30 packets that are separated either by $16ms$ or $33ms$ depending on the Frames Per Second (FPS) rate (60 FPS, respectively 30 FPS) of the camera that generates the frames. The evaluations in [19] show that the AVB standard classes are not enough to ensure the communication needs of the traffic streams in their automotive application example, such as multi-packet video frames. On the other hand TSN is effective at improving the latency, but this comes with a cost of configuration and timing verification, which are complex activities in TSN.

Another contribution that deals with large frames that are split in several packets is [23] though it is not in the context of TSN. The paper is concerned with the dimensioning of the network in order to ensure bounded delays for the video frames generated by various types of cameras, and hence with various requirements and number of packets per frame. The video frames are of the type MPEG as defined by the H.264 codec [24], and this is the model that we will be using in our extended analysis for video frames sent over TNS.

3. TSN Model and Notations

In this section we describe the model of the network we are interested in analyzing as well as the different classes of traffic that can be transmitted on the network. We are interested in Time Sensitive Networks (TSN) embedded in the automotive domain. The network is meant to convey Control Data Traffic (CDT), Audio/Video traffic (AVB) and Best Effort traffic (BE). Packets are sent from end-nodes on the network through one or multiple switches until they reach their destination nodes.

Table 1 gives a typical example of CDT, Audio (A), Video (B) and Best-effort (BE) traffic classes that a TSN network link of bandwidth $BW = 100Mbps$ should support. A stream (or flow) of a given traffic class i denoted by τ_i is characterized by $\tau_i = (C_i, T_i, D_i)$ where C_i is the transmission time of a packet of size $Size_i$ of the stream τ_i, which is obtained by $C_i = Size_i/BW$, T_i is the period or the minimal inter-arrival of the consecutive packets of τ_i, and D_i the end-to-end deadline the network should meet. Of course, one traffic class contains one or several streams having similar real-time constraints. In this case, instead of using one additional index like $\tau_i^j \in eqp(i),\ j \neq i$ to denote another stream j of class i, we simplify its notation in this paper by $\tau_j \in eqp(i),\ j \neq i$ that will not introduce any confusion in the context that we use it. TSN standard imposes the minimal CDT period to 500μs and 5 hops delay not exceeding $D_i = D_i^{global}$=100μs. Audio packets are shaped with the period of 125μs and must respect 2ms maximum 7 hops delay, while video packets with period of 250μs and their 7 hops delay must be bounded by 50ms. We note that audio and video codecs output streams should be shaped at the end system to respect 125μs and 250μs periods.

Figure 1 presents an example of such a network with three switches and twelve end nodes. Table 2 gives details of the ECUs (Electronic Control Units) together with the traffic classes they transmit/receive. It extends the model of [3] and has been defined within the Eurostars RETINA project according to the automotive application requirements.

3.1. Network Model

On the egress port of each switch the TSN standard specifies that there are up to eight gates, meaning that the traffic that passes through a switch can be split into at most eight different classes or priorities. At least one of the eight gates is reserved for the control data traffic (CDT). The gates that are not reserved for the CDT are available to be used by the AVB and BE traffic. The difference between the CDT and other traffic is explained in detail below.

Class	$Size_i$	C_i	T_i	D_i^{local}	D_i^{global}
CDT	175B	14μs	500μs	20μs	100μs/5hops
A	325B	26μs	125μs	285μs	2ms/7hops
B	325B	26μs	250μs	7142μs	50ms/7hops
BE	325B	26μs	125μs	na	na

Table 1: Traffic classes

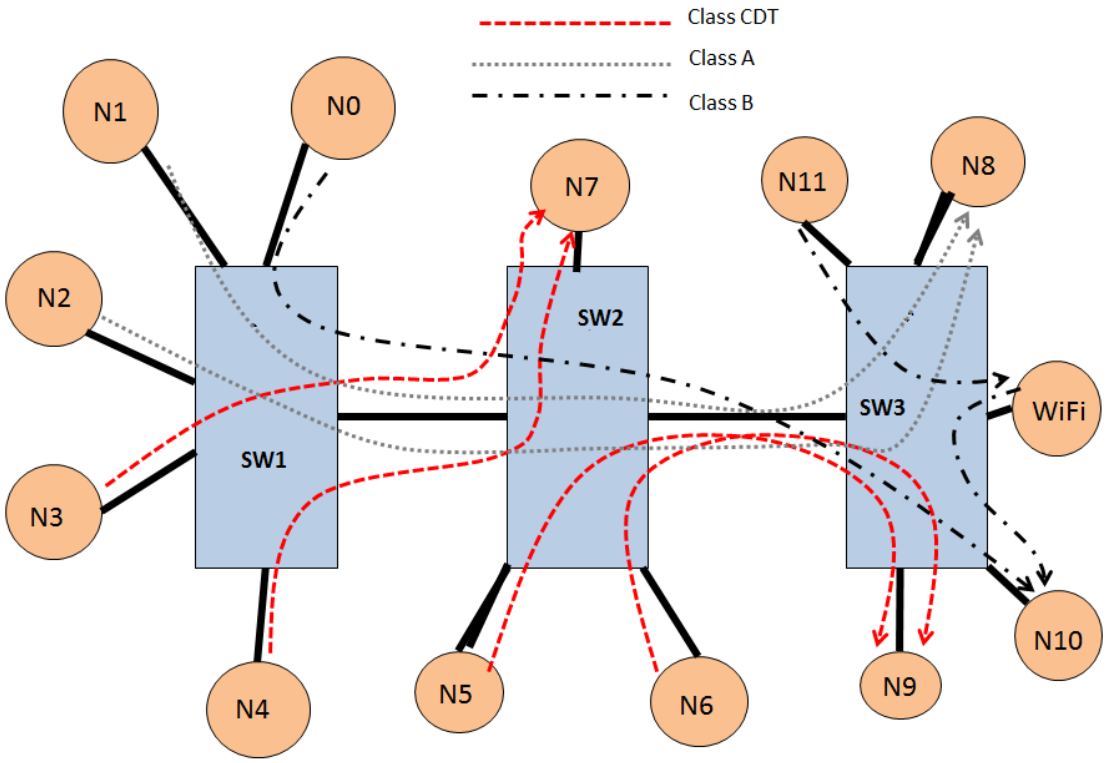


Figure 1: Network topology

A graphical depiction of an egress port can be seen in Figure 2. The port also has a Schedule table (also called Gate-Control-List) implemented with the purpose to reserve bandwidth for the CDT and isolate it from the AVB and BE traffic by using a guard band. When a gate is opened (marked as 1 in the schedule table) then the traffic queued up in the gate can be transmitted, else, if the gate is closed (marked as 0 in the table) then transmission is blocked, except for the last ongoing transmission which can continue its transmission as it should be able to finis before the end of the guard band. AVB traffic is further restricted using CBS (detailed in Section 3.2.2) for fairly sharing the remaining bandwidth among concurrent streams other than CDT.

In this paper we assume transmission without preemption and we only focus on the delay analysis through one egress port. An end-to-end delay upper-bound of a flow crossing several switches can be obtained by performing the addition of all the single hop delays, but this simple summation is not necessarily tight as it does not consider the serialization effect of multi-hop transmission. We will show by simulation an example of the pessimism introduced with such an

Node	Node Type	Stream Type	I/O
N0	Back Camera	Video (Class B)	I
N1	Voice Assistance	Audio (Class A)	I
N2	Media Audio	Audio (Class A)	I
N3	Speed Sensor	CDT	I
N4	Angular Rotation Sensor	CDT	I
N5	Wheel Angle Sensor	CDT	I
N6	Steering Wheel Sensor	CDT	O
N7	Engine Speed Actuator	CDT	I
N8	Sound Output Device	Audio (Class A)	I
N9	Steering Wheel Actuator	CDT	I
N10	Video Output Device	Video (Class B)	I
N11	Front Camera	Video (Class B)	I
WiFi	Wireless module	AVB	I/O

Table 2: Stream details

estimation. How to accurately and tightly compute the end-to-end delay is still an open issue and left as our future work.

3.2. Traffic model

According to the TSN standard, traffic on the network is classified in four classes: CDT (control data traffic), A (Audio), B (video) and BE (best-effort). Classes A and B are also called AVB classes. The CDT class is the novelty that the TSN standard [1] brings with respect to the AVB standard [2] upon which it builds.

3.2.1. Control Data Traffic

The CDT class is meant to contain flows of traffic of a critical nature, with strict timing constraints (delays and jitters), necessary for the functioning of the automobile in which the network is embedded. For this reason the TSN standard specifies shaping mechanisms in order to isolate the CDT class from all other traffic and to guarantee that the imposed delays and jitters are respected. The shaping mechanism that is emerging as the best candidate is called Time Aware Shaper (TAS) and it works by dividing the access to the link in a TDMA fashion in order to isolate the CDT in its own time slot(s) such that AVB and BE traffic can not interfere with it.

CDT is considered to be deterministic as it can be controlled on the sending node. Each output port (also called egress port) of each TSN switch has a TAS implemented in the form of a scheduling table which indicates the moments in time when each gate (different classes of traffic pass through different gates) is closed or opened. In order to isolate CDT streams from all other traffic, the TAS scheduling table necessarily contains at least one time slot when only the gate for the CDT streams is opened and all other gates are closed. CDT packets arrive at the switch inside a CDT time-slot and are immediately sent without being blocked by AVB traffic. Before each CDT time-slot the scheduling table specifies a guard-band (GB), a time-slot in which AVB packets can not start transmission, but they can finish transmission if they started being sent before the guard-band was activated. This mechanism is necessary just before the CDT time-slot is activated to ensure that no other packet would block the transmission of CDT packets as preemption is not allowed. The size of the guard band is equal to the transmission time of the size of the largest packet (over all classes of traffic except CDT) that may pass through the switch. This is to ensure that any packet that begins transmission just before that

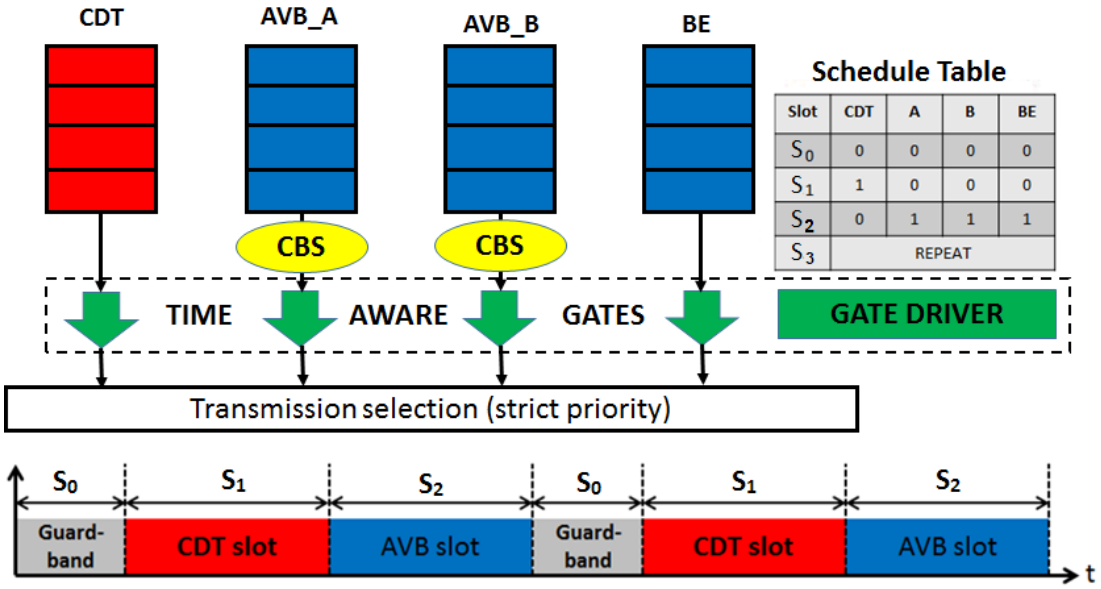


Figure 2: Egress port

guard band is activated, will finish transmission before that CDT-slot begins and so will not block the transmission of CDT packets. The CDT-slot together with the GB form the *Protected Window*.

Setting up the scheduling tables for each TSN switch is a complex problem and a gross-grain solution for it was proposed in [4]. Other related work can be found in [5], [6]. In this work we consider that the scheduling tables for each switch are given.

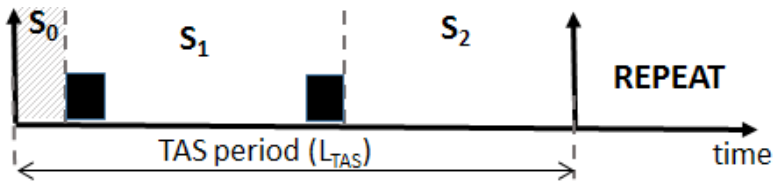
A TAS scheduling table is a list of time points, indicating which gate is open at each time point, with the last entry in the list being the *REPEAT* command. The scheduling table has a period L_{TAS} which is split into different time slots in a TDMA fashion and this splitting repeats for the entire life-time of the system. The TSN standard specifies that $L_{TAS} = 500$ μs by default.

We distinguish between two types of TAS configurations, depending on the granularity of their scheduling tables:

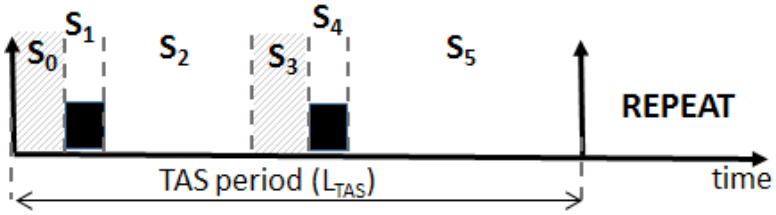
1. With a single Protected Window per TAS cycle, depicted in Figure 3a,
2. With multiple Protected Windows per TAS cycle, depicted in Figure 3b.

Both TAS schedules of Figure 3 need to transmit two CDT packets, represented by the black boxes, but a different amount of bandwidth is reserved in the two cases. The TAS scheduling tables that give rise to the two configurations are presented in Table 3

In Figure 3a the TAS splits time into just three slots: (1) S_0 is the guard band necessary before each CDT-slot, (2) S_1 is the CDT-slot and (3) S_2 is the bandwidth left for the AVB and BE traffic. This kind of TAS has the advantage that it is easy to derive and implement as all CDT packets are grouped into a single block of time. The potential disadvantages are the long blocking time increasing the delay of other traffic and the fact that the bandwidth between the first and the last CDT packets could be wasted, is there is not enough CDT packets to transmit during that time interval. The schedule table for this TAS configuration can be seen in Table 3a.



(a) Single CDT-slot



(b) Multiple CDT-slots

Figure 3: Different TAS configurations

Slot	CDT	A	B	BE
S_0	0	0	0	0
S_1	1	0	0	0
S_2	0	1	1	1
REPEAT				

(a) Single CDT-slot

Slot	CDT	A	B	BE
S_0	0	0	0	0
S_1	1	0	0	0
S_2	0	1	1	1
S_3	0	0	0	0
S_4	1	0	0	0
S_5	0	1	1	1
REPEAT				

(b) Multiple CDT-slots

Table 3: Different TAS scheduling tables

As opposed to the single Protected Window configuration of Figure 3a, the TAS schedule of Figure 3b depicts the more complicated scheduling Table 3b, which splits time in several slots: (a) two guard-bands (S_0 and S_3), (b) two CDT-slots (S_1 and S_4) and (c) two slots left for the transmission of AVB and BE traffic (S_2 and S_5). In this way the TAS only reserves as much bandwidth as necessary for the transmission of CDT packets, freeing up bandwidth to be used by AVB and BE traffic (slot S_2 is gained with respect to the schedule in Figure (3a). Of course this kind of scheduling table can be generalized to an arbitrary number of protected windows not just two. The disadvantage of this kind of configuration is the potentially large complexity of the scheduling table, which increases with the number of CDT flows in the network. Deriving such a scheduling table is not trivial. In addition, each protected window also reserves bandwidth for the guard-band preceding the CDT-slot, which is a potential waste of bandwidth as AVB and BE flows may not always be able to claim it for their transmissions. Also, it may not be worth the effort of making a complex table if the time between CDT packets is small and it is entirely reserved by the guard-band, completely removing the possibility that it is used by AVB or BE flows. In this case we might as well group multiple CDT packets per time-slot and simplify the scheduling table.

3.2.2. AVB Traffic and Credit-Based Shaper

The AVB traffic is transmitted outside of the protected windows which reserves bandwidth for the CDT class. Besides being blocked by the time aware shaper, AVB traffic is also influenced by several other mechanisms:

- classes AVB_A and AVB_B are shaped by the Credit Based Shaper (CBS) specified by the AVB and TSN standards. This shaping mechanism is necessary in order to avoid bursts of packets of the A and B class arriving all at the same time, i.e. to impose a minimal separation between packets of each stream. As an effect of credit based shaping, bandwidth is left available for best effort (BE) streams.
- as in each gate there are packets of different streams (of the same class), these packets are dispatched in FIFO ordering of their arrival at the egress port.
- when packets of different classes are eligible for transmissions (i.e. have non negative credit), then conflicts are resolved according to the fixed priority policy given by the AVB standard which specifies that class A has higher priority than class B which has higher priority than class BE.

Hence the transmission of an AVB stream is influenced by four mechanisms: TAS, CBS, FIFO and FPNS (fixed priority non-preemptive scheduling).

Credit Based Shaping mechanism: Each gate that transmits AVB_A or AVB_B traffic has a CBS mechanism implemented. The main purpose of using CBS (or any other TSN traffic sahpers such as Burst-Limiting Shaper or Peristaltic Shaper) rather than strict priority is to provide a way to reserve enough bandwidth (not all the bandwidth in case of strict priority) while still not starving lower priority classes (notion of fairness). Streams of class $i \in \{$AVB_A, AVB_B$\}$ queued on the output port are allowed to start transmission when the credit of the class is larger or equal to 0. The credit is set to zero by default. When class i transmits, its credit decreases at a rate of α_i^- which is called the *sendSlope* of class i and is measured in bits per second. Alternatively, the credit of the class may increase when either its credit is negative (and it is not currently transmitting), or when there are packets of class i queued in the gate but can not transmit because a class of higher priority is transmitting - in this case the credit of class i may increase even if it is already greater than 0. A positive credit is reset to zero when the corresponding i class queue becomes empty. The increase rate of the credit of class i is denoted by α_i^+, called *idleSlope*. In TSN, with the presence of CDT windows, the credit is kept constant (i.e., frozen) during CDT time slots and can not increase during the guard band (but can still decrease for allowing the last packet before gate closing to finish its transmission during the guard band, before of course the end of the guard band to prevent any interference with the following CDT slot). According to [1] the relation between α_i^+ and α_i^- is established as $\alpha_i^+ + \alpha_i^- = BW$, where BW is the transmission rate of the output port (i.e. the bandwidth of the port). The *idleSlope* (α_i^+) represents the desired bandwidth reservation for the traffic class i. Although the TSN standard does note impose any idleSlope value, in practice the total α_i^+ of n classes for $i = 1, 2, ..., n$ plus the CDT reserved bandwidth can not be larger than the total bandwidth BW of the port. In Section 4.2.2 we will introduce a formal technique for computing appropriate values for the *idleSlope* and *sendSlope*.

TAS influence on the CBS: The IEEE TSN standard specifies that when a gate which transmits AVB traffic is closed (i.e. during the guard-bands and the CDT-slots) the credit of the traffic in that gate is frozen and can no longer *increase*. Also, as AVB streams are allowed to transmit during the guard-band (finish transmitting of packets that had started before the activation of the guard-band), then the credit of that class does *decrease* during the guard-band. These effects are represented graphically in Figure 4 (best viewed in color on a screen), where

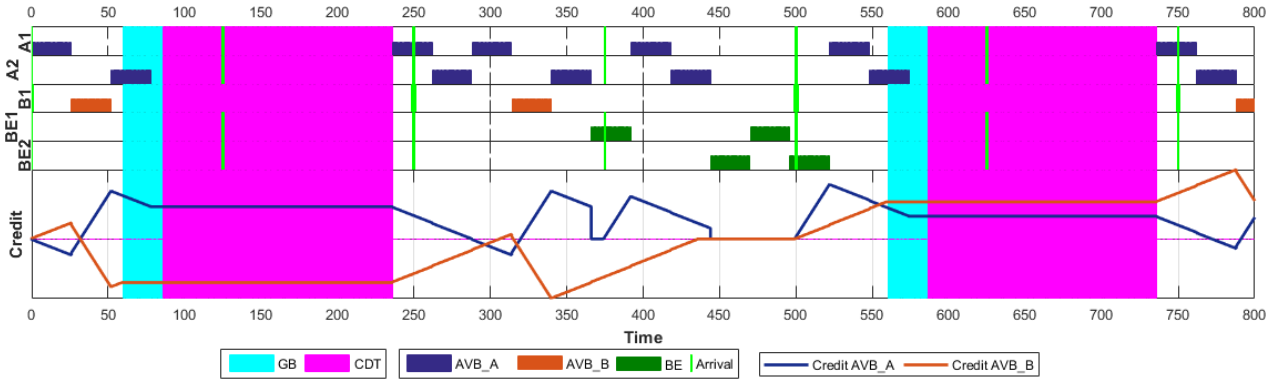


Figure 4: The effect of TAS on credit consumption and replenishment

two AVB_A flows, one AVB_B flow and two BE flows are being transmitted in a TSN switch (in this case from switch SW1 to switch SW2 of Figure 1). The credit of the AVB_A class is represented by the blue line while the credit of the AVB_B class is represented by the red line. Packets of class AVB_A and BE are available for transmission at $t = \{0, 125, 250, 500, 625, 750\}$; packets of class AVB_B are available for transmission at $t = \{0, 250, 500, 750\}$; the guard-band is activated at $t = \{60, 560\}$ and the CDT-slot is activated at $t = \{86, 586\}$. Green lines mark the moments in time when packets are available to be transmitted. The *idleSlope* and *sendSlope* of the two AVB classes are set, in this illustrative example, to 80Mbps for the *idleSlope* of class AVB_A and 20Mbps for the *idleSlope* of class AVB_B (we will discuss in the next section on how to set the value of the *idleSlopes* in the presence of the guard band and CDT that are represented by the protected windows). All packets have an equal size of 26μs. The guard-band also has a size of 26μs as this is the size of the largest AVB and BE packet in our numerical example of Table 1[3]. The length of the CDT-slot is equal to 150μs.

It can be seen that at $t = 52$, stream AVB_A2 starts transmitting a packet, and it continues transmitting until $t = 78$ with the guard-band being active since $t = 60$. Between $t = 60$ and $t = 78$ the credit of class AVB_A continues to decrease. Once the packet finishes its transmission, then the credit of class AVB_A stays constant until the end of the CDT-slot. As opposed to class AVB_A, class AVB_B has negative credit at $t = 52$, hence its credit is incremented according to its *idleSlope*, until $t = 60$ at which moment its gate is closed due to the activation of the guard-band. The credit of class AVB_B stays constant during the guard band and the CDT-slot, even though it has a negative value. We recall that the credits of AVB classes can decrease during the guard-band but not increase, and it can neither increase nor decrease during the CDT-slot, where it stays at constant value.

Note that at $t = 366$ and at $t = 444$ the credit of class AVB_A drops from a positive value to zero as at those time instants there are no AVB_A packets waiting to be transmitted. This is a characteristic of the IEEE AVB [2] standard regarding credit base shaping and it still applies to TSN switches.

A further effect of the TAS on AVB flows that greatly contributes to their worst case transmission delays is presented in detail in Section 4.2 (particularly Figure 6) hence we omit

[3]In general the length of the guard-band corresponds to the size of the largest non-CDT packet in the switch. This value is upper-bounded by the time needed to transit one Ethernet MTU packet of size 1542 bytes (e.g. 123.36μs in a 100Mbps switch) without preemption and lower-bounded by the time needed to transmit two Minimal Ethernet packets of 64 bytes (e.g. 2x5.12=10,24 μs on a 100Mbps switch) with preemption

it here and refer the reader to Section 4.2.

3.3. Problem Description

The problem that we address in this paper can be expressed as follows: given a TSN switch and the traffic passing through it, compute the worst case delay of packets of AVB classes A and B in the switch, knowing that their transmission times are influenced by the two shapers simultaneously (TAS and CBS) as well as by the interference and blocking from other AVB packets of higher and lower priority (FPNS) and by packets of same priority as the packet under analysis (FIFO). To the best of our knowledge, only two previous work exists on the delay analysis of AVB in TSN context: the first one is our previous work [15] by extending the eligible interval approach and the second [16] and [17] using the network calculus approach. Furthermore, in Section 5 we will extend the traffic model to the case when video frames are split into several packets which are transmitted over multiple TAS cycles and the entire frame needs to respect a given deadline, i.e. the delay of the frame is computed as the amount of time passed from when its first packet is ready for transmission until the moment when its last packet finishes transmission.

3.4. Notations

We summarize in Table 4 the notations that are used throughout the paper.

Notation	Definition
CDT, A, B, BE	Control Data Traffic, Audio, Video, Best Effort classes
L_{TAS}	TAS cycle duration, $L_{TAS} = 500\mu$s by default
L_{CDT}	CDT slot duration
L_{GB}	Guard Band duration
L_{PW}	Protected Window duration $L_{PW} = L_{GB} + L_{CDT}$
$\tau_i = (C_i, T_i, D_i)$	A stream of class i
$\tau_j \in eqp(i)$	Other streams of the same class i, a simplified form of $\tau_i^j \in eqp(i),\ j \neq i$
C_i	Packet transmission time of a stream of class i
T_i	Period or minimal interarrival of consecutive packets of a stream of class i
D_i	Relative deadline of any packet of a stream of class i
BW	Bandwidth (date rate) of an Ethernet link
α_i^+ and α_i^-	Idle and send slopes (CBS credits) reserved to the queue of class i
α_H^+ and α_H^-	Idle and send slopes reserved to a class of higher priority than a given class i
α_L^+ and α_L^-	Idle and send slopes reserved to a class of lower priority than a given class i
$R_{FIFO}(\tau_i)$	Worst case delay of stream τ_i in FIFO queue (classic Ethernet)
$R_{AVB}(\tau_i)$	Worst case delay of stream τ_i in AVB network
$R_{FIFO}(\tau_i)$	Worst case delay of stream τ_i in TSN
BR_i	Bandwidth reservation ratio of class i queue
U_i, U_H, U_L	Utilization (workload ratio) of respectively classes i, High and Low
T_I	I-frame interval (or GOP interval)
T_f	Minimal interval of any two consecutive video frames
B_I	Number of the packets split from an I-frame
B_p	Number of the packets split from a P-frame
C_v	Size of a packet of a video frame (considered identical for I or P-frames)
β	Number of TAS cycles a video frame overlaps: $\beta = \lceil \frac{T_f}{L_{TAS}} \rceil$

Table 4: Notations

4. Local analysis for AVB streams in a TSN switch

In this section we introduce the main contribution of our work, the formal worst case delay analysis for an AVB packet in a TSN switch. For this we extend the eligible interval analysis of [11] and [10] to account for possible blocking created by the TAS. This part is largely inspired from our previous work [15] with two notable improvements: providing a formal proof of the CBS shaped AVB traffic delay influenced by TAS of TSN; and explicitly giving the rules for the bandwidth allocation (idle slope) that can allow to guaranteeing the deadline of a given AVB traffic class.

We first start with the case when there is a single protected window (i.e. a single CDT slot and its preceding guard-band) per TSN TAS cycle, and then generalize the result to the case of multiple protected windows per TSN TAS cycle.

4.1. Local analysis for AVB switches

Before we proceed with the proposed analysis, we briefly recall here the worst case delay analysis in the case of AVB networks as described in [10]. The analysis computes the worst case delay of a packet of stream τ_i, taking into account the worst-case blocking due to higher priority packets (C_H^{max}) as well as the worst-case blocking due to lower priority packets (C_L^{max}) which are "dilated" to take into account the worst case delay due to insufficient credit. The rationale for "dilating" packets can be found in [8] and we do not detail it here. In the rest of this paper we will use the letter L to denote the class of streams of lower priority than τ_i. Similarly, we will use the letter H to denote the class of streams of higher priority than τ_i. The equation for computing the worst case AVB delay for a stream τ_i is the following (from [10]):

$$R_{AVB}(\tau_i) = R_{FIFO}(\tau_i) + C_L^{max} \times (1 + \frac{\alpha_H^+}{\alpha_H^-}) + C_H^{max} \tag{1}$$

where $R_{FIFO}(\tau_i)$ is the delay due to the FIFO ordering of packets of the same class on the gate through which the packet under analysis is transmitted, computed as follows (from [10]):

$$R_{FIFO}(\tau_i) = C_i + \sum_{\tau_j \in eqp(i),\ j \neq i} C_j \times (1 + \frac{\alpha_i^-}{\alpha_i^+}) \tag{2}$$

In Equations (1) C_L^{max} represents the maximum transmission time of any packet over all flows of lower priority than the flow under analysis τ_i and is used as a blocking time caused by a lower priority packet due to the fact that transmission is assumed non-preemptive (note that we are only interested in non-preemptive functioning of TSN in this paper). C_H^{max} represents the maximum transmission time of any packet over all flows of priority higher than the flow under analysis. In Equation (2), $eqp(i)$ represents the set of other flows of the same priority (i.e., same class of traffic) as the flow under analysis. α_i^+ and α_i^- are the *idleSlope* and respectively *sendSlope* of the class under analysis, with $\alpha_i^+ + \alpha_i^- = BW$. So the maximum utilization for class i is $\frac{\alpha_i^+}{BW}$. It is achieved when only class i queue is in transmission with always queued packets.

It is important to note that for Equations (1) and (2) to be applicable, class i needs to have bounded transmission delays which is ensured if its bandwidth reservation respects the following utilization condition:

$$\sum_{eqp(i)} \frac{C_i}{T_i} \leq \frac{\alpha_i^+}{\alpha_i^+ + \alpha_i^-} = \frac{\alpha_i^+}{BW} \tag{3}$$

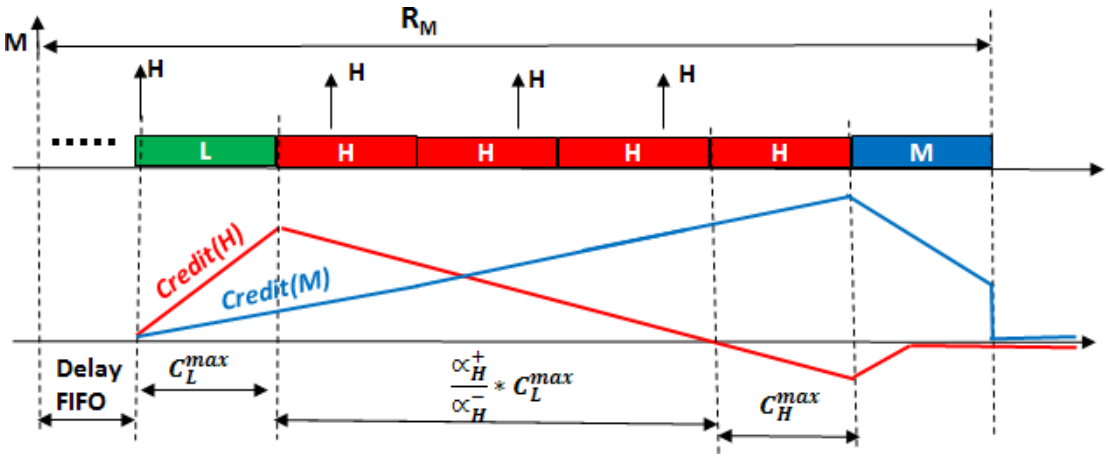


Figure 5: Worst case delay of a packet when there is no Protected Window.

Cao et al. [10] proved that Equation (1) gives a tighter bound than other previous existing results on the worst-case interference that a packet suffers in an AVB switch. We also note that according to the proof of Lemma 1 in [10], Equation (2), which is deduced from the busy period analysis of [8], gives an upper bound on the worst-case delay.

A generic worst case scenario in the AVB context is depicted in Figure 5 where the transmission of a packet of medium priority M is delayed by packets of the same priority in the FIFO queue through which it passes, then by a lower priority packet L and then by several packet of the higher priority class H (depending on the amount of credit that the class H managed to accumulate during the transmission of packet L), leading to Equation 1.

Once the transmission time of an AVB packet is computed we can then include the further blocking that the packet can suffer due to the time aware shaper of the TSN switch which prevents the AVB transmission during the time slots reserved to CDT.

4.2. Analysis in the case of a single Protected Window per TAS cycle

In this section, we will give both the AVB traffic delay bound in TSN context (which improves our first result in [15] by providing a formal proof) and the way to choose the necessary bandwidth reservation through α_i^+, under assumption that the worst case delay or deadline does not exceed the duration of one TAS cycle (typically 500μs). This last assumption will be relaxed in Section 5 when dealing with multi-packet video streams where multiple TAS cycles are considered.

4.2.1. Worst case delay analysis

Let us denote by L_{CDT} the length of the CDT-slot. That is, the amount of time during which the entire bandwidth of the link is reserved for the transmission of CDT packets.

Similarly, let us denote by L_{GB} the length of the Guard Band preceding the CDT-slot. That is, the amount of time during which the entire bandwidth of the link is reserved in order to protect the transmission of CDT packets. Note that the length of the Protected window is equal to $L_{PW} = L_{CDT} + L_{GB}$.

As the credits of AVB streams do not increment when their gates are closed, i.e. neither during the CDT slot, nor during the Guard Band, then when a protected CDT slot occurs it freezes the increase of the credits of AVB flows. Subsequently, once the protected window has passed, the transmission of AVB packets can continue from where they left off. Note that

credit can continue to decrease during the guard band if a packet is still being transmitted, as described in Section 3.2.2.

Definition 1. *We define the minimal bandwidth reservation ratio BR_i of a class $i \in \{AVB\_A, AVB\_B\}$ in a TSN switch as the proportion of bandwidth (BW) of a link that is given to class i through its credit replenishment with idleSlope and consumption with sendSlope, after we have subtracted the portion of bandwidth that can not be used in the worst case by AVB classes, that is during the guard band L_{GB} and CDT slot L_{CDT}:*

$$BR_i = \frac{\alpha_i^+}{\alpha_i^+ + \alpha_i^-} \times (1 - \frac{L_{GB} + L_{CDT}}{L_{TAS}}) \tag{4}$$

where L_{TAS} is the TAS cycle duration.

Feasibility condition: In order to guarantee that AVB packets have finite worst-case transmission times in a TSN switch, a necessary condition is that the utilization of the class is less than its minimal bandwidth reservation ratio BR_i:

$$U_i = \sum_{eqp(i)} \frac{C_i}{T_i} \leq BR_i \tag{5}$$

Theorem 1. *If the feasibility condition given by Equation 5 is respected, then the worst case delay of an AVB flow τ_i in a TSN switch can be computed using the following equation:*

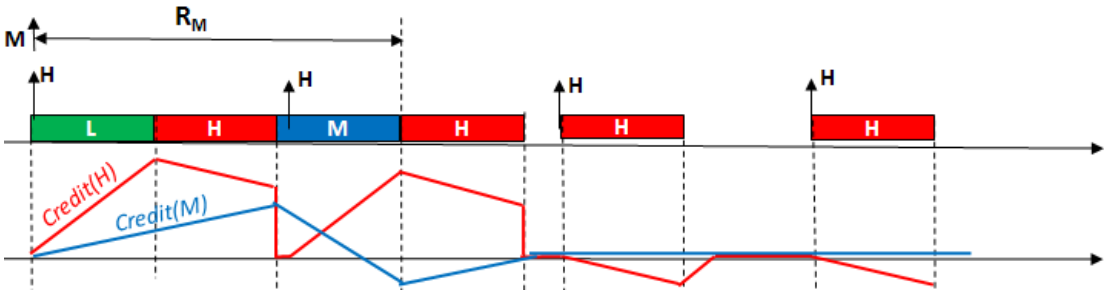
$$R_{TSN}(\tau_i) = R_{AVB}(\tau_i) + L_{CDT} + L_{GB} \tag{6}$$

where $R_{AVB}(\tau_i)$ is obtained using Equation (1).

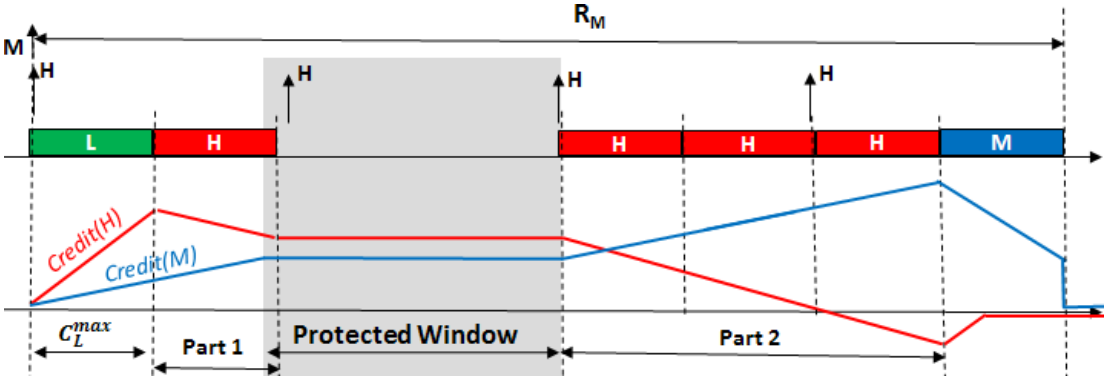
This is a surprisingly simple result that means in the TSN context, the worst-case delay is just that of AVB context plus the protected window added by the TAS of TSN. However it is not straightforward and need to develop the following proof.

Proof 1. *An effect of the TAS that greatly contributes to the worst case transmission delay of an AVB class can be seen in Figure 6, where the relative order of transmission is changed due to the Protected Window. Let us consider the transmission of the packet denoted by M. In Figure 6a, packet M is transmitted after a lower priority packet L and a higher priority packet H. In Figure 6b, the Protected Window is activated right at the end of the first H packet, blocking in this way the transmission of packet M. Furthermore, several H-packets become available for transmission during the Protected Window, and as they have higher priority than packet M they get transmitted right after the Protected Window, further blocking the transmission of packet M and changing the relative transmission order with respect to Figure 6a. We note that this transmission inversion is caused by the presence of a TAS protected window, hence it can not appear in the AVB context, but only in TSN. An appropriate and safe analysis in the TSN context needs to take such effects into consideration.*

Nevertheless, the amount of H-packets that can block the transmission of packet M is limited by the idleSlope *and* sendSlope *of stream H. This was proven in [10] for the case of AVB networks and it still holds true in TSN due to the fact that the credit of class H can not increase during the Protected Window. That is, regardless of how many H packets are ready for transmission at the end of the protected window, only a subset of these can be transmitted before their credit becomes negative. An intuition of this upper-bound on H packets can be seen in Figure 5, which is the worst case transmission delay for packet M when there is no Protected*



(a) Schedule without blocking by Protected Window



(b) Schedule with blocking by Protected Pindow

Figure 6: The effect of a TAS Protected Window on packet transmissions

Window. The credit of stream H can reach a maximal value of $\alpha_H^+ \times C_L^{max}$ *and then it decreases at a rate of* α_H^- *requiring* $\frac{\alpha_H^+ \times C_L^{max}}{\alpha_H^-}$ *time units before reaching zero, at which point stream H can still send one more packet. This scenario leads to the worst case delay of packet M in an AVB switch, given by Equation 1. As the Credit of class H does not increase during the Protected Window, this upper-bound on the maximal blocking by higher priority streams over stream M continues to hold true, leading to a worst case transmission delay of an AVB packet in a TSN switch given by Equation (6). Figure 6 shows the effect of a protected window over the transmission order of AVB packets which leads to bursts of higher priority packets to be transmitted right after the Protected Window, blocking lower priority streams and resulting in a worst case transmission delay where otherwise it wouldn't be the case. In Figure 6b the sum* $Part\ 1 + Part\ 2$ *is at most equal to* $\frac{\alpha_H^+ \times C_L^{max}}{\alpha_H^-} + C_H^{max}$ *of Figure 5 which leads to Equation (6) by adding* $L_{CDT} + L_{GB}$ *to Equation (1).*

□

Equation 6 gives a conservative upper-bound as it supposes that the packet under analysis needs to wait for both the GB and CDT slots to pass before it can be transmitted, in addition to its worst case AVB delay. The pessimism comes from the GB since in practice it can be used (but not always) for transmitting one largest AVB or BE packet since L_{GB} is set to the transmission time of such a packet to allow the transmission of the last packet available just before the end of the non-CDT slot.

Equations (4), (5) and (6) provide a sufficient analysis framework for computing delays for AVB flows in TSN switches and consequently for verifying the schedulability of the system, by

comparing the computed delays with the deadlines on the packets.

Inversely, for satisfying a given deadline (worst-case delay), one can also find what is the suitable CBS credit value for a traffic class.

4.2.2. CBS credit allocation

It is important to notice that Equation (6) assumes that the packet under analysis can only meet one protected window during its worst case delay (single protected window per TAS cycle), that is to assume that the worst case delay is less than L_{TAS}. In fact, reserving a bandwidth ratio higher than a traffic class utilization, as given in equation (5), only leads to a finite response time, but does not guarantee that it is less than a given deadline (D_i) or amount of time (e.g. L_{TAS}). So a practical issue would be how to reserve sufficient but still minimum bandwidth to a class of AVB traffic allowing to guarantee the deadline constraint of that class. It is obvious, and also as indicated in [21], that a tighter deadline will need more bandwidth reservation. So the minimum bandwidth reservation algorithms in [21], which are only applicable to AVB networks (i.e. not TSN), can be extended to the TSN context as follows, where AVB traffic is delayed by the protected window ($L_{PW} = L_{GB} + L_{CDT}$).

Let us define α_H^+ and α_H^- as the Audio traffic credit replenishment and consumption slopes, and α_M^+ and α_M^- as the Video traffic credit replenishment and consumption slopes respectively. Each audio and video flow is characterized by $\tau_i = (C_i, T_i, D_i)$ where D_i is the deadline per Ethernet packet. The choice of α_H^+ and α_M^+ are bounded by equations (7) and (9). The utilization of the audio (respectively video) flows is denoted by U_H (respectively U_M).

A minimal value for α_H^+ can be derived using the following equation:

$$\alpha_H^+ \geq max(\frac{U_H}{1 - \frac{L_{PW}}{L_{TAS}}}, \max_{\tau_i \in H} \frac{\sum_{\tau_j \in H, j \neq i} C_j}{D_i - C_i - C_{M,L}^{max} - L_{PW}}) \times BW \tag{7}$$

Equation (7) gives a lower bound on the minimal bandwidth reservation that the Audio class must have to guarantee the required deadline. It could be in turn either $\frac{U_H}{1-\frac{L_{PW}}{L_{TAS}}} \times BW$ (derived from Equation (4)) if the utilization constraint is dominant (i.e., the deadline is large) or the second part of the max function if the deadline constraint is dominant (i.e. the deadline is so small that one has to reserve more bandwidth than the audio traffic load). The proof of this second part $\max_{\tau_i \in H} \frac{\sum_{\tau_j \in H, j \neq i} C_j}{D_i - C_i - C_{M,L}^{max} - L_{PW}}$ is straightforward from equation (6) by putting $R_{TSN}(\tau_i) = D_i$. Again we note that this is only true for the case of $D_i \leq L_{TAS}$. In case that $D_i > L_{TAS}$, we need to impose a virtual deadline on the stream, which is at most equal to the length of the TAS cycle, for example $D_i' = min(D_i, L_{TAS})$.

Once α_H^+ has been derived such that it verifies Equation (7), this value needs to also verify the following upper-bound equation:

$$\alpha_H^+ \leq (1 - \frac{L_{PW}}{L_{TAS}}) \times BW \tag{8}$$

Equation (8) is the upper bound of the bandwidth that the Audio class can use. It is just the total output port bandwidth minus the part reserved for the protected window. We note that this is a safe bound, but it could be slightly pessimistic since one audio packet is allowed to transmit during the guard band (as any transmission that starts just before the starting of the guard band can finish before the end of this guard band).

Similarly, a minimal value for α_M^+ can be derived using the following equation:

$$\alpha_M^+ \geq max(\frac{U_M}{1-\frac{L_{PW}}{L_{TAS}}}, \max_{\tau_i \in M} \frac{\sum_{\tau_j \in M, j \neq i} C_j}{D_i - C_i - C_L^{max}(1+\frac{\alpha_H^+}{\alpha_H^-}) - C_H^{max} - L_{PW}}) \times BW \tag{9}$$

Equation (9) once again gives the minimal bandwidth reservation that the Video class must have to guarantee the required deadline. It could be in turn either $\frac{U_M}{1-\frac{L_{PW}}{L_{TAS}}} \times BW$ if the utilization constraint is dominant (i.e., the deadline is large) or the second part of the max function if the deadline constraint is dominant (i.e. the deadline is so small that one has to reserve more bandwidth than the video traffic load). The proof of this second part $\max_{\tau_i \in M} \frac{\sum_{\tau_j \in M, j \neq i} C_j}{D_i - C_i - C_L^{max}(1+\frac{\alpha_H^+}{\alpha_H^-}) - C_H^{max} - L_{PW}}$ is also straightforward from equation (6) by putting $R_{TSN}(\tau_i) = D_i$. One can also refer to [21] for a similar approach to the case of only AVB without TAS.

$$\alpha_M^+ \leq (1 - \frac{L_{PW}}{L_{TAS}}) \times BW - \alpha_H^+ \tag{10}$$

Equation (10) is the upper bound of the bandwidth that the Video class traffic can use. It is just the total output port bandwidth minus both the parts reserved for CDT and for the Audio class traffic. We also note that this bound could be pessimistic since one video packet may be allowed to transmit during the guard band, and all reserved bandwidth for the Audio class through α_H^+ is not necessarily used out by the audio streams.

Example 1. *In order to see how Equations (7), (8), (9) and (10) are applied, we present here a simple numerical example.*

Let $\Gamma = \{CDT_1, AVB\_A_1, AVB\_A_2, AVB\_B_1, BE_1, BE_2\}$ *be a Stream-Set passing through a TSN switch of transmission rate* $BW = 100Mbps$*. The parameters of the streams are as given by Table 1. There is a single Protected Window which isolates flow* CDT_1 *from the AVB and BE traffic. The size of the protected window is* $L_{PW} = L_{CDT} + L_{GB} = 14\mu s + 26\mu s = 40\mu s$ *and the TAS cycle is* $L_{TAS} = 500\mu s$*.*

Let us suppose that class A has a deadline of $D_A = 285\mu s$*, then by applying Equation (7) we obtain that* $\alpha_A^+ \geq max(\frac{0.416}{0.92}, \frac{26}{285-26-26-40}) \times BW = max(0.4521, 0.1347) \times BW = 0.4521 \times BW$*. This is a minimal required bandwidth reservation for class AVB_A.*

We can choose any value larger than $0.4521 \times BW$ *for* α_A^+ *in order to ensure bounded delays for class AVB_A. For example let us choose* $\alpha_A^+ = 0.46 \times BW$*, which is a safe value.*

Once a safe value for α_A^+ *is found using Equation (7), this value needs to also satisfy Equation (8). The condition to be checked is* $0.46 \times BW \leq (1 - \frac{40}{500}) \times BW = (1 - 0.08) \times BW = 0.92 \times BW$*. We conclude that indeed there is enough bandwidth (*$0.92 \times BW$*) to accommodate the requirements for class AVB_A.*

Similarly, let us suppose that class B has a deadline of $D_B = \min\{500, 7132\}$*, then by applying Equation (9) we obtain that:*

$\alpha_B^+ \geq max(\frac{0.104}{0.92}, \frac{0}{500-26-26\times(1+\frac{80}{20})-26-40}) \times BW = max(0.113, 0) \times BW = 0.113 \times BW$*. This is a minimal required bandwidth reservation for class AVB_B.*

Let us choose $\alpha_B^+ = 0.15 \times BW$*, which is a safe value and will ensure bounded delays of class AVB_B. In order to verify Equation 10 we check that* $0.15 \times BW \leq (1 - \frac{40}{500}) \times BW - 0.46 \times BW = (1 - 0.08) \times BW - 0.46 \times BW = 0.46 \times BW$*. The inequality is respected.*

As we have found feasible values for both α_B^+ *and* α_A^+ *we can conclude that the stream-set is feasible on the switch under consideration and we can now use Equation (6) to compute the worst case local delay for each class.*

In this example, both class A and class B are utilization dominant as it is the first part of the max function in Equations (7) and (9) that gives the lower bounds for α_A^+ and α_B^+ respectively. If we change the deadline of class AVB_A to be $D_A = T_A = 125\mu s$ then we have that $\alpha_A^+ \geq max(0.4521, \frac{26}{125-26-26-40}) \times BW = max(0.4521, 0.7877) \times BW = 0.7879 \times BW$, and class AVB_A would be deadline dominant as it is now the second part of the max function that gives the lower bound of α_A^+.

Equations (9) and (10) can be further extended to the general case when there is an arbitrary number of traffic classes that are of higher priority than the class under analysis, by making sure that the equation removes from the total link bandwidth the bandwidth reserved for all higher priority classes as well as the bandwidth reserved for the control data traffic.

Equation (4) removes from the total bandwidth of the output port the bandwidth that is blocked by the TAS for the Protected Window. A fraction of the bandwidth that is left is reserved for class i giving the value of BR_i. This reservation is similar to the reservation in AVB switches (Equation 3), where a fraction of the *total* bandwidth is reserved for an AVB class i.

4.3. Generalization to arbitrary number of Protected Windows per TAS cycle

The case of multiple Protected Windows is more complex to analyze due to the possible inter-leaving of AVB and CDT reservations of the bandwidth.

We denote by L_{CDT}^k the length of the k-th CDT slot, in the set of n CDT slots that my occur during the entire TAS cycle. Each such CDT slot is preceded by a guard band of length L_{GB}, i.e. all guard bands are of equal lengths.

A conservative upper-bound on the worst-case delay of an AVB packet in a TSN switch can be obtained by generalizing Equation (6) in the following way:

$$R_{TSN}(\tau_i) = R_{AVB}(\tau_i) + \sum_{k=1}^{n} L_{CDT}^k + n \times L_{GB} \tag{11}$$

This equation holds under the generalized utilization condition:

$$\sum_{eqp(i)} \frac{C_i}{T_i} \leq BR_i^n \tag{12}$$

where

$$BR_i^n = \frac{\alpha_i^+}{\alpha_i^+ + \alpha_i^-} \times (1 - \frac{n \times L_{GB} + \sum_{k=1}^{n} L_{CDT}^k}{L_{TAS}}) \tag{13}$$

In order to compute this upper-bound we need to know the length of each CDT slot in the switch. Still, this upper-bound is pessimistic as it supposes that all protected windows arrive immediately one after the other, which is probably not the case in well designed networks, i.e. there are gaps between protected windows in which AVB packets can be transmitted. The analysis can be further refined since not all flows will be blocked by all protected windows. Indeed, some higher priority streams may suffer less blocking from TAS, while lower priority flows will potentially be blocked by all protected windows in a TAS period. Distinguishing the exact amount of TAS blocking that a flow may suffer is a non-trivial problem and we leave it as future work.

The analysis framework given by Equations (11), (12) and (13) is a direct generalization of the case when there is a single Protected Window in the TAS cycle (Section 4.2), and hence it follows the same reasoning of correctness (but with higher pessimism).

To ensure that delays are bounded and the analysis can be applied, we need to ensure that the bandwidth reservations α_A^+ and α_B^+ are safe. That is, Equations (7) and (9) need to be satisfied with the mention that the bandwidth reservation for protected windows is generalized to $L_{PW} = n \times L_{GB} + \sum_{k=1}^{n} L_{CDT}^k$.

5. Extension to multi-packet Video Streams

In this section we extend our analysis to cope with streams that have transmission times larger than the TAS-cycle of the TSN switch. For this, we extend the video class AVB_B to a realistic model used by an industrial partner[4] in the conception of automotive embedded video cameras. This model is in accordance with the ones in the state of the art [24], [23] presented in Section 2.

5.1. Realistic Video Model

A realistic video stream is generally composed of two types of frames:

- I-frames, of size C_I,
- P-frames, of size $C_P = \frac{C_I}{p}, p \geq 1, p \in \mathbb{R}$.

A visual representation of the frames generated by a video stream can be seen in Figure 7. The transmission of each I-frame is followed by the transmission of m P-frames then another I-frame and m P-frames and so on, the pattern repeating during the whole video stream. An I-frame together with the m P-frames that follows it form what is called a GOP (Group Of Pictures) of size $m + 1$. Note that m can be equal to 0, in which case there are no P-frames in the GOP, just a single I-frame. Similarly, if p (the scaling factor of P-frames) is equal to 1 then the size of each P-frame is effectively equal to that of an I-frame, meaning that there is no difference between I-frames and P-frames.

A new I-frames is available for transmission every T_I time units, which is called the I-period (or GOP-period), while the minimal separation between frames is denoted by T_f. That is, the separation between two consecutive P-frames as well as between an I-frame and the P-frame immediately following it is equal to T_f. In this paper we suppose that all video streams have the same period T_f. Generally, $T_f = 40000\mu$s and the GOP is formed of $49 + 1 = 50$ frames, leading to an I-period of $T_I = 50 \times 40000\mu s = 2s$.

As the size of a video frame generated by a camera can be considerably larger than what is allowed by the TSN standard, the video frames are split in multiple packets. Hence, an I-frame is composed of B_I packets, while a P-frame is composed of B_P packets.

The size of any packet of a video stream is denoted by C_v and the number of packets that a frames is divided into is given by the ratio between the size of the frame and the size of the packet. That is, I-frames are split into $B_I = \lceil \frac{C_I}{C_v} \rceil$ packets, while P-frames are split into $B_P = \lceil \frac{C_P}{C_v} \rceil$ packets. As it can be seen in Figure 7, the last packet of a frame may be smaller in size than C_v. For this work we consider that this last packet is padded so that it has a size equal to C_v.

When a video frame is generated it is split into packets, and then all the packets are available for transmission at the same time and queued in the internal buffer of the end-system that generated them. It is then up for the network interface to induce a minimal separation between packets. We denote this time interval by T_P and it corresponds to the period of the AVB_B

[4] ALKIT Communications, `http://www.alkit.se/`

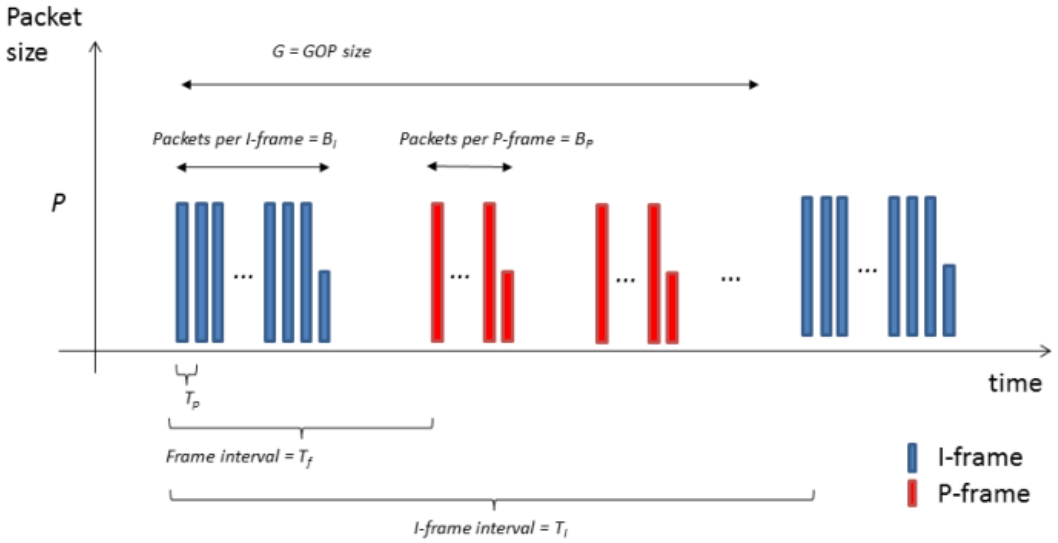


Figure 7: RTP Video traffic characterization

class as given in Table 1. Once in the network, the packets corresponding to a frame can once more be buffered in the egress port of a switch and effectively become available for transmission all at the same time, i.e. the minimal separation T_P is nullified and the stream has a bursty behavior that is then corrected by the Credit Based Shaper.

Once all the packets of a frame reach their destination end-system, they are reassembled to form the video frame. The transmission delay of an entire video frame needs to be within a deadline, starting from the moment the first packet becomes available for transmission until the moment when the last packet finishes transmission. Hence, in this section we focus our attention on computing the transmission delay of the last packet of the frame, considering that all the other $B_I - 1$ packets are also buffered in the FIFO queue of the egress port, leading to the worst case delay of the entire video frame.

Furthermore, in order to place ourselves in the worst case condition we suppose that the scaling factor of P-frames is $p = 1$, i.e. the P-frames are identical to I-frames in number of packets and their size. This assumption regards all the video streams if the class AVB_B and through the rest of this section we use the term video frame to mean both I-frame and P-frame. The worst case transmission delay of a video frame occurs when all other video streams in the egress port have frames that become ready for transmission at the same time when the frame under analysis becomes ready for transmission.

5.2. Analysis of multi-packet video streams in a TSN switch

In order to compute the worst case delay of a realistic video stream we first need to modify Equations (1) and (2) so that they take into account the novel nature of the video frames that are split into packets. The delay that occurs due to the FIFO ordering of packets in the output gate is given by (generalization of Equation 2):

$$R_{FIFO}(\tau_i) = C_v + (B_i - 1) \times C_v \times (1 + \frac{\alpha_i^-}{\alpha_i^+}) + \sum_{\tau_j \in eqp(i),\ j \neq i} B_j \times C_v \times (1 + \frac{\alpha_i^-}{\alpha_i^+}) \qquad (14)$$

Equation (14) is interpreted as follows: the worst case delay of the last packet of a video frame under analysis (i.e. the delay of the entire video frame) in the FIFO queue is composed of the transmission time (C_v) of the packet itself, plus the transmission time of the other $B_I - 1$ packets ("inflated" by the CBS) of the frame under analysis ($(B_i - 1) \times C_v \times (1 + \frac{\alpha_i^-}{\alpha_i^+})$), plus the transmission times of all packets ("inflated" by the CBS) of all the video frames of all the other streams in the FIFO queue ($\sum_{\tau_j \in eqp(i),\ j \neq i} B_j \times C_v \times (1 + \frac{\alpha_i^-}{\alpha_i^+})$).

Equation (1) stays unchanged in its formulation, with the exception that, if any of the higher priority streams is (also) composed of frames that are split into several packets (i.e. video streams), then

$$C_H^{max} = max(max_{j \in AVB_B}(C_j^v); max_{j \in AVB_A}(C_j)) \tag{15}$$

A similar observation needs to be made for the case when one or more lower priority streams are generating frames that are split into packets:

$$C_L^{max} = max(max_{j \in AVB_B}(C_j^v); max_{j \in AVB_{BE}}(C_j)) \tag{16}$$

Once $R_{AVB}(\tau_i)$ is computed, the worst case transmission delay of a frame of video stream τ_i is obtained by solving the following recurrence equation:

$$R_{TSN}^{k+1}(\tau_i) = R_{AVB}(\tau_i) + \left\lceil \frac{R_{TSN}^k(\tau_i)}{L_{TAS}} \right\rceil \times L_{PW} \tag{17}$$

where:

- L_{PW} is the total amount of bandwidth reserved by the TAS within a cycle of length L_{TAS} (be it the case of a single or multiple Protected Windows). That is, $L_{PW} = n \times L_{GB} + \sum_{k=1}^{n} L_{CDT}^k$, where n is the number of protected windows in the TAS cycle.
- The value $\left\lceil \frac{R_{TSN}(\tau_i)}{L_{TAS}} \right\rceil$ represents the number of TAS cycles that will affect the transmission of the frame under analysis. Figure 8 provides a visual representation of how the term $\left\lceil \frac{R_{TSN}(\tau_i)}{L_{TAS}} \right\rceil$ of Equation (17) is computed.

At the first step of the computation, $R_{TSN}^0(\tau_i)$ is instantiated with the value of $R_{AVB}(\tau_i)$ and the result obtained at step k is used as input for step $k + 1$. The computation stops when the result obtained at step $k+1$ is equal to that obtained at step k, and this result is the worst case transmission time of a video frame of the stream under analysis. Alternatively, the computation can also be stopped when the following condition is no longer respected:

$$R_{TSN}(\tau_i) \leq T_f(\tau_i) \tag{18}$$

The condition given by Equation (18) is interpreted as follows: when the computed delay exceeds the period of the stream the analysis is stopped as it means that the current frame interferes with the transmission of the next frame of the stream and so the transmission times would increase more and more for every subsequent frame, i.e. would be unbounded. A tighter stopping condition can be given by replacing $T_f(\tau_i)$ in Equation 18 with the local deadline $D^{local}(\tau_i)$ of the stream under analysis, provided that the local deadline is less than or equal to the period, i.e. $D^{local}(\tau_i) \leq T_f(\tau_i)$:

$$R_{TSN}(\tau_i) \leq D^{local}(\tau_i) \tag{19}$$

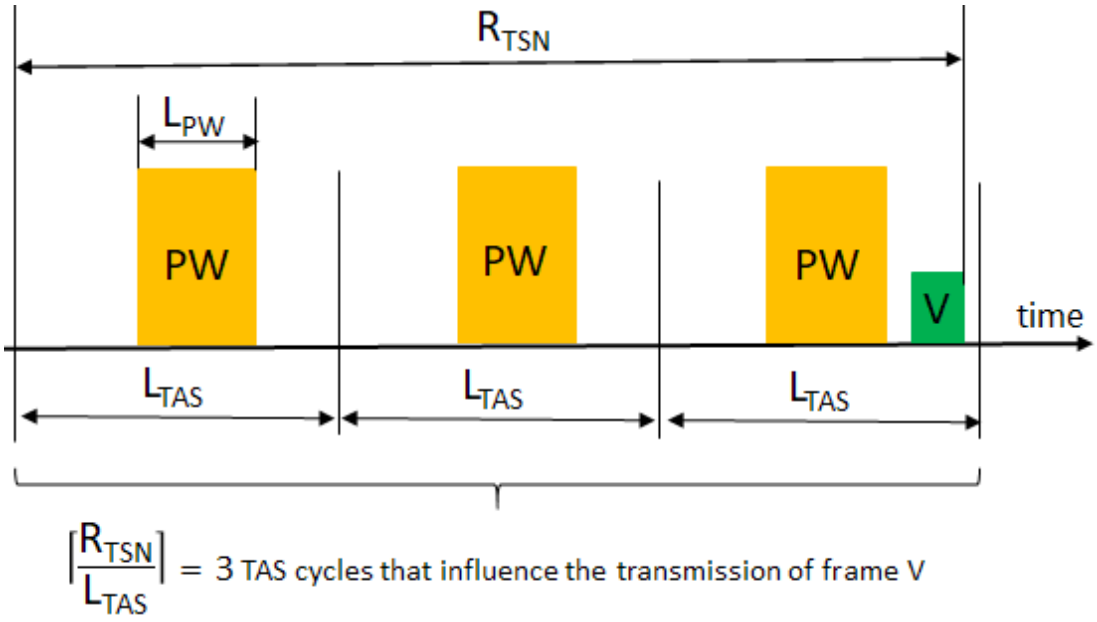


Figure 8: The impact of multiple TAS-cycles on the transmission of a frame.

meaning that the computation is stopped once it is determined that the frame can not be transmitted within its local deadline. This condition implies that a local deadline for the stream can be precisely determined and this deadline should be strictly respected.

5.3. Bandwidth reservation requirement

In this paper we suppose that all video streams have the same period T_f. We define the worst case utilization of the video class U_v (v represents video which is equivalent to the medium class M of the previous sections) as:

$$U_v = \sum_{i \in eqp(v)} \frac{C_I^i}{T_f} = \sum_{i \in eqp(v)} \frac{B_I^i \times C_v}{T_f} \tag{20}$$

Equation (20) computes the worst case utilization of the video class as it only takes into account the I-frames. This utilization may be too pessimistic if the amount m of P-frames following each I-frames is large. Nevertheless, in order to provide a worst case utilization we need to only take into account I-frames. An average utilization may be derived by taking into account P-frames as well as I-frames, but this also needs an average delay analysis, as the worst case delay analysis we propose in this section is not compatible with an average utilization condition and may produce optimistic results. Another, better option of taking into account the reduced transmission times of P-frames is to perform a probabilistic analysis [25] of the delays incurred by video frames. Such an analysis would take advantage of the fact the I-frames may rarely synchronize to produce a scenario in which deadlines would be missed. Rather, at any given time, the video gate of the egress port would be filled with P-frames and few (or none at all) I-frames. This would be the average case scenario. We leave the derivation of such an analysis for future work and for now we contend ourselves with proposing a worst case delay analysis for video streams, knowing that the delay of I-frames is a safe (though pessimistic) upper-bound on the delay of P-frames.

Let us denote by β the number of TAS-cycles that any video frame of a video stream overlaps with:

$$\beta = \lceil \frac{T_f}{L_{TAS}} \rceil \tag{21}$$

The bandwidth reservation of a video class v is defined as:

$$BR_v = \frac{\alpha_v^+}{\alpha_v^+ + \alpha_v^-} \times (1 - \frac{\beta \times L_{PW}}{T_f}) \tag{22}$$

which is a generalization of Equation (13).

Then Equation (12) can be generalized to the following necessary bandwidth reservation condition, for the case of realistic video streams:

$$U_v \leq BR_v. \tag{23}$$

From Equations (22) and (23) we can derive a minimal requirement for α_v^+ such that the transmission delay of the video stream under analysis is bounded by the period of the stream T_f:

$$U_v \leq \frac{\alpha_v^+}{\alpha_v^+ + \alpha_v^-} \times (1 - \frac{\beta \times L_{PW}}{T_f}) <=> \tag{24}$$

$$\frac{U_v}{1 - \frac{\beta \times L_{PW}}{T_f}} \leq \frac{\alpha_v^+}{BW} <=> \tag{25}$$

$$\frac{U_v}{1 - \frac{\beta \times L_{PW}}{T_f}} \times BW \leq \alpha_v^+ \tag{26}$$

Equation (9) can be update to reflect the fact that the video frame can span across several TAS cycles, resulting in a tighter bound on the bandwidth reservation in the case that the deadline of the flow is the dominant constraint (rather than the utilization of the video class):

$$\alpha_v^+ \geq max(\frac{U_v}{1 - \frac{\beta \times L_{PW}}{T_f}}, \max_{\tau_i \in V} \frac{\sum_{\tau_j \in V, j \neq i} C_j}{D_i - C_i - C_L^{max}(1 + \frac{\alpha_H^+}{\alpha_H^-}) - C_H^{max} - \beta \times L_{PW}}) \times BW \tag{27}$$

5.4. Correctness

Equation (17) provides a safe upper-bound over the transmission delay of a video stream τ_i because the order in which packets are transmitted is given by $FIFO$ and the worst case blocking suffered by a packet of τ_i is already computed by Equations (1) and (14). Furthermore, the blocking caused by the protected windows does not change the transmission order of packets nor the worst case blocking suffered by the packet under analysis due to lower and higher priority AVB-streams, but instead it delays the transmission of all the packets in the ready-queue by the same amount of time, i.e. the length of the protected window. Also, due to the fact that credits can not increase during the protected windows, it means that no matter how many new higher and lower-priority packets become ready for transmission during the protected window, the worst case blocking inflicted by other streams to the packet under analysis is bounded by the value computed with Equation (1), which makes use of the credit slopes of the various AVB classes passing through the egress port under consideration.

We note here that Equation (17) reduces to Equation (11) when the transmission time of the packet under analysis is at most equal to the length of the TAS-cycle, L_{TAS}, i.e. $\left\lceil \frac{R_{TSN}(\tau_i)}{L_{TAS}} \right\rceil = 1$.

Example 2. *Let us consider a link of a TSN switch through which the following stream-set is transmitted:*

- *two AVB_A streams* $A_1 = A_2 = \{C_A = 1, T_A = 12, \alpha_A^+ = 20\% \times BW\}$*;*
- *two video streams* $B_1 = B_2 = \{C_I = 3, T_I = 15, C_v = 1, m = 2, \alpha_B^+ = 80\% \times BW\}$*, which means that* $B_I = 3$ *and* $B_P = 2$*;*
- *two BE streams* $BE_1 = BE_2 = \{C_{BE} = 1, T_{BE} = 12\}$.

The Time-Aware Shaper has a cycle equal to $L_{TAS} = 7$ *of which it reserves* 1 *time unit for the CDT slot and another* 1 *time unit of the guard band that precedes the CDT slot, leaving* 5 *time units for the AVB and BE traffic.*

In order to compute the worst case transmission time of a video frame on the TSN switch, we first need to compute its transmission time in an AVB context. In the worst case, we assume that B_1 *is always transmitted before* B_2*. We first apply Equation (2) to flow* B_2*:*

$R_{FIFO}(B_2) =$
$C_v + \sum_{\tau_j \in eqp(B_2),\ j \neq B_2} B_j \times C_v \times (1 + \frac{\alpha_B^-}{\alpha_B^+}) + (B_i - 1) \times C_v \times (1 + \frac{\alpha_B^-}{\alpha_B^+}) =$
$1 + 1 \times 3 \times (1 + \frac{20}{80}) + 1 \times 2 \times (1 + \frac{20}{80}) = 1 + 3 \times \frac{5}{4} + 2 \times \frac{5}{4} =$
$1 + \frac{15}{4} + \frac{10}{4} =$
$1 + \frac{25}{4} =$
7.25

Subsequently we apply Equation (1):

$$R_{AVB}(B_2) = R_{FIFO}(B_2) + C_{BE} \times (1 + \frac{\alpha_A^+}{\alpha_A^-}) + C_A = 7.25 + 1 \times (1 + \frac{20}{80}) + 1 = 9.5$$

Once we have computed $R_{AVB}(B_2)$ *we can use Equation (17) to compute the worst case transmission time of an I-frame of the video stream by initializing* $R^0_{TSN}(B_2) = 9.5$*:*

$$R^1_{TSN}(B_2) = 9.5 + \lceil \frac{9.5}{7} \rceil \times 2 = 9.5 + 4 = 13.5 \neq R^0_{TSN}$$

$$R^2_{TSN}(B_2) = 9.5 + \lceil \frac{13.5}{7} \rceil \times 2 = 9.5 + 4 = 13.5 = R^1_{TSN}$$

hence the computation stops after the second iteration as $R^2_{TSN}(B_2) = R^1_{TSN} = 13.5$ *and we have determined a safe upper-bound for the video streams* B_1 *and* B_2 *in the TSN switch to be equal to* 13.5 *time units.*

Figure 9 presents a visual representation of the worst case transmission delay of the video flows analyzed in this example. Each rectangle represents a packet of size $1\mu s$ *and the numbers* n *of the video packets represents that the packet is the* n*th one of the frame. For example the packet that is transmitted between* $t = 57$ *and* $t = 58$ *has the number* 3 *on it, meaning that it is the* 3^{rd} *and last packet of the frame. A new frame starts with the packet marked* 1*. The packet transmitted between* $t = 57$ *and* $t = 58$ *became available for transmission at* $t = 47$ *while the frame to which it belongs became available for transmission at* $t = 45$*. This means that the delay of the entire frame is equal to* $58 - 45 = 13\mu s$ *and this is the worst case transmission delay of the video stream on the switch as it is indicated by our analysis.*

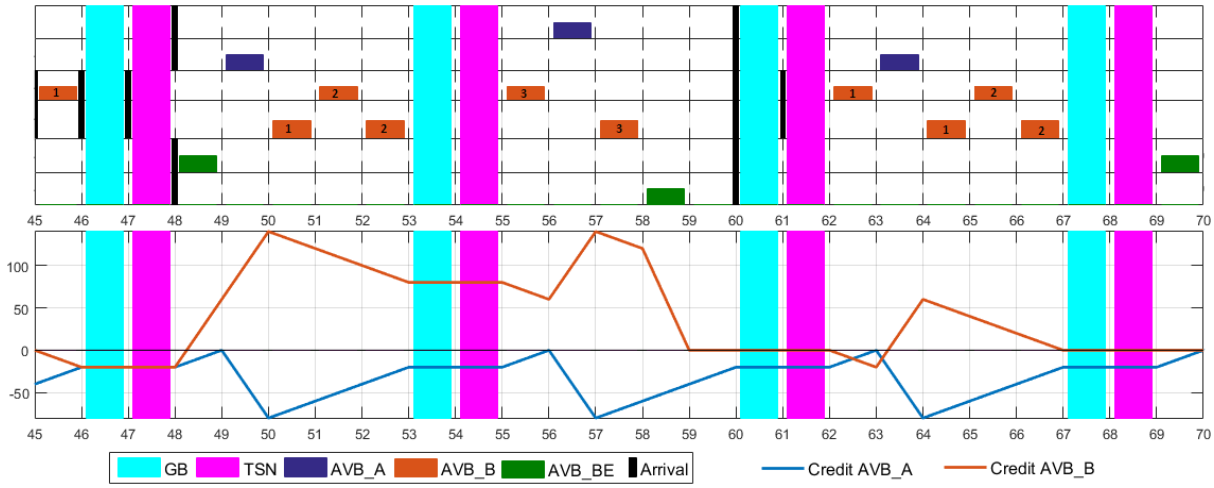


Figure 9: The transmission of a video frame over TSN.

6. Experimental results

In this section we further validate our delay analysis for AVB traffic in TSN switches by applying it on various stream-sets. Firstly, in Section 6.1 we take a look at an automotive-inspired TSN system which we simulate and the observed delays are compared with the the delays computed using our theoretical analysis. Then, in Section 6.2 we perform a series of experiments to assess the efficiency of the proposed analysis for multi-packet video frames.

6.1. The case of standard AVB classes

This first set of experiments comes in support of the analysis presented in Section 4.2 and Section 4.3.

Network and traffic: The topology of the network is presented in Figure 1. It has three switches and 13 end-nodes. The details of each node and the type of traffic they generate or receive are presented in Table 2. Figure 1 also presents the sources, destinations and paths that each CDT and AVB stream takes through the network.

The default parameters of each stream are presented in Table 1. The period (T_i) and end-to-end deadline (D_i^{global}) of each stream are specified in the IEEE TSN standard. The local deadline (D_i^{local}) is computed by dividing the end-to-end deadline to the number of hops it is specified upon. Other ways for computing the local deadlines exist, for example taking into consideration the load on each switch through which the stream passes, but for our example it suffices to divide the end-to-end deadline by the number of hops.

The size parameter ($Size_i$) of a stream is made up of the payload of a packet of the stream plus its header, i.e. $Size_i = Payload_i + Header_i$. The transmission time C_i of a packet of stream i is computed by dividing the size of the stream (converted in *bits*) to the port transmission rate of the TSN switch:

$$C_i = \frac{Size_i \times 8}{BW} = \frac{(Payload_i + Header_i) \times 8}{BW} \tag{28}$$

For example, in Table 1, a transmission rate of 100 Mbps is considered, resulting in the transmission times C_i presented in the third column of the table. For simplicity, all throughout the section we reason in terms of size (as opposed to just the payload of the packet), including in

this way the header. For this experiment we consider that the *sendSlope* and *idleSlope* of classes AVB_A and AVB_B are given and they are as follows: $\alpha_A^- = 20\% \times BW$, $\alpha_A^+ = 80\% \times BW$, $\alpha_B^- = 80\% \times BW$ and $\alpha_B^+ = 20\% \times BW$.

The two CDT streams are separated by 100μs on the sending nodes, so that they are never available for transmission to block each-other. That is, if the CDT stream of node N_3 has a packet available for transmission at instant t, then the CDT stream of node N_4 will have a packet available for transmission at instant $t + 100$.

For our investigation we use two TAS configurations, as presented in Table 3. The period of the TAS is $L_{TAS} = 500\mu$s in both cases.

- The first configuration has a single Protected Window composed by a CDT-slot of size $L_{CDT} = 150\mu$s (slot S_1 in Table 3a), preceded by a guard-band of size $L_{GB} = 26\mu$s (slot S_0 in Table 3a). This leaves $500 - 150 - 26 = 324\mu$s available for the transmission of AVB and BE traffic (slot S_2 in Table 3a).
- The second configuration has two CDT-slots of equal sizes $L_{CDT} = 14\mu$s (slots S_1 and S_4 in Table 3b), each CDT-slot being preceded by a guard-band of size 26μs (slots S_0 and S_3 in Table 3b). Due to the separation of 100μs of the two CDT streams, the two Protected Windows are spaced out so that there are 60μs between them, available for AVB and BE traffic (slot S_2 in Table 3b). Slot S_5 in the scheduling Table 3b, of size 360μs is also available for the transmission of AVB and BE traffic.

All AVB and BE streams are first instantiated at $t = 0$ and then subsequently instantiated according to their periods T_i. All throughout our experiments we have varied the starting time of the TAS by giving it an offset (let us denote it by ϕ) that varies between 0 and $L_{TAS} = 500\mu$s in increments of 1μs. This means that slot T_0 starts at $t = \phi$, then slot S_1 starts after the previous slot has passed, i.e. at $t = S_0 + \phi$, and so on. For each value of the offset, a simulation of 100000μs was performed. Hence, each experiment contains 500 simulations, one per each value of ϕ and the largest observed delay over all simulations was recorded. Offsetting the TAS with respect to the first arrivals of non-CDT traffic allows for some AVB and BE packets to be transmitted before the first guard-band is activated, modifying the credits accordingly, leading to larger delays for later packets as credits are already negative at the instant when the guard-band is activated. The simulator used is a prototype that we have implemented based on the run-time simulator PAnSim [26] and that we make available for free use to the community[5].

6.1.1. Default experiment

As a first experiment we simulate the network depicted in Figure 1, at a transmission rate of 100Mbps for each link and with 9 flows crossing the network as given in Table 2: four CDT flows passing through the protected windows formed by the TAS, two AVB_A flows, one AVB_B flow and two BE flows. The parameters of each traffic class are presented in Table 1. Except for the BE flows, all flows are depicted on the network in Figure 1. The BE flows are not depicted in the figure as they are broadcasted in the entire network, passing through all switches and reaching all end-nodes.

As an example of the kind of results that our local delay analysis provides, we present in Table 5 the results for the AVB flows transmitted between switch SW1 and switch SW2. On this link there are two CDT flows, two AVB_A flows, one AVB_B flow and two BE flows.

The utilization of the link between SW1 and SW2 of Figure 1 (U_{SW1SW2}) can be computed as the sum of the utilization of the protected windows, the AVB_A and AVB_B classes and the

[5] `https://members.loria.fr/DMaxim/artifact-rtns2017-tsnavb/`

			Single Protected Window		Multiple Protected Windows	
Stream	AVB Analisys	AVB Simulation	TSN Analysis	TSN Simulation	TSN Analysis	TSN Simulation
A1	85	81	261	260	165	124
A2	85	81	261	260	165	124
B1	182	52	358	317	262	139

Table 5: Worst-case delays on switch SW1 for the default experiment

BE class: $U_{SW1SW2} = U_{PW} + 2 \times U_A + U_B + 2 \times U_{BE}$. The utilization of the class AVB_A (keeping in mind that there are two streams of this class on the link) is $U_A = 2 \times (26/125) = 2 \times 0.208 = 0.416 = 41.6\%$. The utilization of the class AVB_B is $U_B = 26/250 = 0.104 = 10.4\%$. This means that the utilization of the AVB streams is $U_{AVB} = U_A + U_B = 41.6\% + 10.4\% = 52\%$ of the total link bandwidth, i.e. 52Mbps. The utilization of the CDT streams is actually given by the percentage of bandwidth that is reserved by the protected windows. For example, in the case of two protected windows, each of the size $L_{PW} = L_{GB} + L_{CDT} = 26\mu s + 14\mu s = 40\mu s$, occurring every 500us, we obtain a utilization of the two protected windows equal to $U_{PW} = 2 \times 40/500 = 0.16 = 16\%$. If we add this to the AVB utilization we obtain a link load of $68\% = 68Mbps$. The BE traffic, with a worst case utilization of $U_{BE} = 0.416$ and without delay constraints, is added to fully load the link and generate worst case scenarios for the AVB class. In the case of a single protected window of size $L_{PW} = L_{GB} + L_{CDT} = 26\mu s + 150\mu s = 176\mu s$ the PW utilization is $U_{PW} = 176/500 = 0.352 = 35.2\%$. Adding this to the AVB utilization we obtain a link load of $52\% + 35.2\% = 87.2\%$. Again, the BE class is meant to load the link and generates worst case transmission scenarios for the AVB classes.

The network was first simulated in an AVB setting, i.e. no CDT traffic, hence no bandwidth blocked by protected windows. The worst case delays for the AVB setting were also calculated with the eligible interval technique presented in [11] and [10]. Then the protected windows were activated to enhance the network to transmit CDT flows in the two TAS configurations discussed in the previous section. The simulated delays are then compared to the delays computed using our proposed analysis in Section 4.2 and Section 4.3 for the two TAS configurations respectively. As it can be seen in Table 5 the delays computed by our analyses are strictly larger than the delays observed during simulation, yet very close even in the case when these are two CDT-slots. This indicates that the analysis is safe and tight, being able to correctly calculate the delay that a packet of a given flow will suffer in the switch. For example, for class AVB_A (streams A1 and A2) in the case of a single Protected Window, our analysis computes a worst case delay on switch SW1 equal to 261μs while the largest delay we observe during simulation for the stream is 260μs. In the case of two Protected Windows, for the same class our analysis computes a delay of 165μs while the largest delay observed during simulation is just 124μs. The difference between the two delays is 41μs which is about the size of a protected window, i.e. $L_{GB} + L_{CDT} = 26 + 14 = 40\mu$s. This indicates that the stream is not affected by the second Protected Window in the TAS cycle, denoting the pessimism of our generalized analysis. We leave for future work the decrease of this pessimism.

Using our local analysis we can compute a bound for the end-to-end delays of the streams by applying the analysis on each switch that the stream traverses. For example, for stream A2, for the case of a single Protected Window, the analysis computes an end-to-end delay of 574μs while during simulation the largest delay observed is equal to 324μs. Similarly in the case of multiple Protected Windows the analysis computes an end-to-end delay of 382μs while the largest simulated delay is equal to 198μs. In both cases we notice an increased amount of

pessimism which was to be expected as our analysis is a local one, hence it assumes an arrival pattern that leads to the worst case blocking on each switch through which the stream passes. This assumption is not necessarily true in all switch as packets get serialized from one switch to the other resulting in less blocking and smaller end-to-end delay with respect to what a local analysis can compute. To further reducing this pessimism an end-to-end analysis needs to be proposed. Several approaches can be investigated, for example a solution based on the trajectory approach [9], on the forward end-to-end delay analysis [27] or on networkcalculus [16, 28]. This is one of our future objectives.

6.1.2. Extended experiment

In order to test the generality of our analysis, for the second experiment we have extended the amount of traffic in the network while keeping the same topology. In order to accommodate more flows we increase the transmission rate of each switch in the network to 1Gbps. The consumption and replenishment slopes are adjusted accordingly: $\alpha_A^+ = 800$Mbps, $\alpha_A^- = 200$Mbps, $\alpha_B^+ = 200$Mbps and $\alpha_B^- = 800$Mbps. Without loss of generality we consider that nodes can send multiple flows simultaneously and the routes, source nodes and destination nodes of the flows are the same as presented in Figure 1, as well as the transmission rates and deadlines are those presented in Table 1. For the sake of simplicity we present here only the results for the link between switch SW1 and switch SW2 (which we will refer to as SW1-SW2).

For this experiment we consider the size of a CDT packet to be equal to $C_{CDT} = 2\mu$s. There are two CDT streams on the link, generating from nodes N3 and N4. The CDT streams are separated by 100μs, hence we do not investigate the case of a singe Protected Window as the time between the two flows would be too big of a waste relative to the size of the packets in the network. That is, we limit our investigation to the case of two Protected Windows with minimal wastage of bandwidth, i.e. $L_{CDT} = 2\mu$s.

As the largest size of any packet on a TSN switch can be 1534 bytes, which means approximately 12μs on a 1Tb switch, we considered that there are 12 AVB_A streams passing through the link SW1-SW2, with transmission times varying from 1μs to 12μs, in increments of 1μs such that each stream has a distinct transmission time, i.e. $C_{A_7} = 7\mu$s . The utilization of class AVB_A is $U_A = 0.624$ which is less than its bandwidth reservation ratio $BR_A = 0.7552$, meaning that all streams of the class have bounded delays. Similarly we consider that there are six streams of class AVB_B passing through the link SW1-SW2, with transmission times varying from 1μs to 6μs. The utilization of class AVB_B is $U_B = 0.084$ which is less than its bandwidth reservation ratio $BR_B = 0.1888$, i.e. all AVB_B streams have bounded delays. The transmission times C_i of AVB streams are presented in the second column of Table 6. All periods T_i and deadlines (D_i^{local} and D_i^{global}) are the same as in Table 1. There are also 10 streams of class BE passing through the link SW1-SW2 and each of them has a transmission time of 12μs. Hence, in total there are 30 streams passing through the SW1-SW2 link: two CDT, 12 $\times$ AVB_A, 6 $\times$ AVB_B and 10 $\times$ BE.

The theoretical worst case-delays for the AVB traffic computed with our proposed analysis, as well as the largest delays observed during simulation for each stream are presented in Table 6. Once again we observe that the computed delays correctly and tightly upper-bound the delays observed during simulation. In this experiment, the pessimism induced by the Protected Windows is reduced (compared to the previous experiment) as the Protected Windows are relatively small with respect to the size of the TAS cycle L_{TAS}. For example, in all our simulation stream A1 was transmitted before being blocked by the second Protected Window of the TAS, given the fact that the difference between the observed delay and the theoretical delay equals 14μs, which is exactly the size of a Protected Window. Stream B6 on the other hand, suffers a high amount of blocking from AVB_A streams, pushing it to also be blocked by the second Protected

Stream	C_i	TSN Analysis	TSN Simulation
A1	1	138	124
A2	2	137	122
A3	3	137	121
A4	4	137	124
A5	5	137	126
A6	6	136	127
A7	7	136	124
A8	8	136	126
A9	9	136	130
A10	10	135	127
A11	11	135	130
A12	12	135	130
B1	1	201	181
B2	2	197	185
B3	3	193	175
B4	4	198	172
B5	5	185	171
B6	6	181	181

Table 6: Delays on switch SW1 for the extended experiment

Window of the TAS, leading to a large transmission delay, which is correctly computed by our analysis.

6.2. The case of multi-packet video streams

We now present a set of experiments that assess the efficacy of the analysis introduced in Section 5. As this analysis is a local one, we focus on the accuracy of the analysis applied on Stream Sets passing through a single switch. In Subsection 6.2.1 we take a look at the accuracy of the analysis and discuss the assumption that the bandwidth reserved by the TAS for Guard Bands is not available for AVB transmissions. In Subsection 6.2.2 we take a closer look at the local feasibility of randomly generated Stream Sets and how it varies when some parameters of the Stream Sets are varied.

Throughout all the experiments we consider that the deadline of an I-frame is equal to the period of the stream, that is, the frame needs to finish transmission before the next frame of the same flow is ready to be transmitted. We also fix the bandwidth of the egress port to 1 Gbps.

In order to assess the worst-case delay of a video frame, our local analysis takes into account the synchronous arrival of all I-frames as well as all AVB_A and BE frames, hence this is the scenario that we will be considering. This is analogous to the case when P-frames have the same size as I-frames and virtually there is no difference between P-frames and I-frames (the worst video traffic load case). We leave as future work the analysis of the general (average) case when there are both I-frames and P-frames of different sizes in the link, leading to smaller delay.

6.2.1. Pessimism and Guard-Band Impact

For this experiment we fix the number of AVB_A flows to two and the number of BE flows also to two. The number of video flows (AVB_B) is varied. The *idleSlopes* and *sendSlopes* are fixed through the experiment at $\alpha_A^+ = 40\% \times BW = 400Mbps$, $\alpha_A^- = 60\% \times BW$, $\alpha_B^+ = 60\% \times BW = 600Mbps$, $\alpha_B^- = 40\% \times BW$. We note that according to the IEEE 802.1Qbv standard [1] the *idleSlopes* of AVB streams are first set based on the class utilization and as

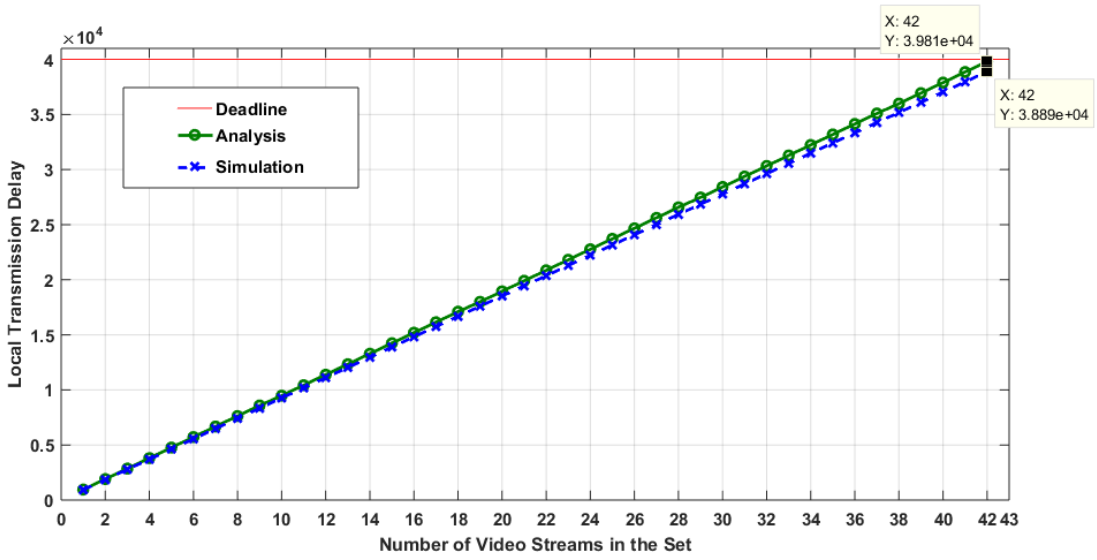


Figure 10: The increase Transmission Delay with respect to the number of video streams in the switch in the case of five Protected Windows.

if there wouldn't be any control data traffic and subsequently the *idleSlopes* is scaled up by a factor proportional to the amount of Bandwidth available to the class in a TAS cycle ([1, Section 8.6.8.2, page 19]). For this experiment we do not scale the *idleSlopes* for each point on the X-axis as it suffices to know that the Bandwidth reservations for each AVB class is enough to accommodate the traffic with bounded delays.

The parameters for AVB_A and BE flows are given in Table 1, while the parameters of AVB_B flows are $C_I = 500\mu$s, $B_I = 50$, $T_B = 40000$. This means that the payload of the I-frame is split into 50 packets, each requiring $C_v = 10\mu$s to be transmitted. All video frames have the same parameters.

The TAS has a cycle of length $L_{TAS} = 500\mu$s, with five equally spaced Protected Windows. Each PW is composed of a Guard Band of length $L_{GB} = 10\mu$s and a CDT slot of length $L_{CDT} = 2\mu$s.

The number of video flows (AVB_B) is varied between 1 and 42, with 42 being the maximal number at which the Stream-Set is still feasible (both in theorey and in simulation) with the given *idleSlopes* and *sendSlopes*, i.e. with 42 AVB_B streams in the set, Equation (26) holds true, while with 43 or more it no longer holds true.

Figure 10 presents the increase in transmission delay of video frames as the number of Video flows in the stream set is incremented starting from a single one and up to 42 flows. If one more video flow would be added then the link would be saturated, video frames would not be able to be transmitted within their deadlines and the system would be unfeasible. Hence a total of 42 Stream-Sets are represented in Figure 10, each one with two AVB_A flows, two BE flows and a different number of Video flows in the range [1, 42]. For example, the Stream-Set on position 20 on the X-axis is composed of 2 × AVB_A flows, 20 × Video flows, 2 × BE flows and 5 × Protected Windows. As the link becomes more saturated and the transmission delays larger, the pessimism of the analysis also increases. This is due to the fact that the delays being larger, more TAS cycles (hence more Protected Windows) impact the transmission of a frame. The analysis considers that all Protected Windows that can impact a frame will indeed delay it, which is not necessarily the case, leading to the gap between the two curves (Analysis and

Simulation) in Figure 10.

Let us take a closer look at the case when there are 42 Video streams in the set (i.e. the point of saturation). It has a Theoretical Worst case Delay of 39808μs and we observed a maximal delay during simulation of 38891μs, meaning that there is a difference of 917μs between Analysis and Simulation. During the transmission of a video frame it can potentially be impacted by at most $\frac{40000}{500} = 80$ TAS cycles. As each TAS cycle has 5 protected windows of length $L_{PW} = 10 + 2 = 12$, it means that a video frames can suffer an extra delay (with respect to the AVB case) equal to $80 \times 5 \times 12 = 4800\mu$s. This value represents 12% of the delay of the frame, but it also means that 12% of the link Bandwidth is reserved for Protected Windows and is potentially not available for AVB traffic. Out of this value 4000μs is reserved for Guard Bands (hence part of it might be used by AVB traffic) and 800μs is reserved for CDT-slots.

As the difference between Analysis and Simulation in only 917μs and not the entire 4800μs that are reserved for Protected Windows, it means that AVB traffic takes advantage of 917μs out of the 4000μs given to the Guard Bands. This still leaves $4000 - 917 = 3883\mu$s that is wasted Bandwidth.

This shows that, even though our analysis has some pessimism, it is not as much as we initially expected as (in this experiment) the AVB traffic only manages to access about a quarter of the Bandwidth reserved for Guard Bands, while the other three quarters are wasted and simply represent blocking times for the AVB traffic.

6.2.2. Stream-Set Feasibility

In this experiment we further investigate the pessimism of our analysis, while also looking at how the feasibility of synthetically generated Stream-Sets varies with respect to given parameters.

Since our analysis is a sufficient one (not a necessary one), we have devised a series of experiments to assess its tightness with respect to (i) the percentage of Bandwidth reserved for Protected Windows, (ii) the number of video streams in the set and (iii) the number of audio streams in the set.

For these experiments we have generated a total of 4924800 Stream-Sets in the following way:

- The number of AVB_A streams is varied between 1 and 40 per Set. Each AVB_A stream has a transmission time of 3μs and a period of 125μs.
- The number of AVB_B (Video) streams is varied between 1 and 80 per Set. Each AVB_B stream has a transmission time of 500μs and a period of 40000μs and each frame is split in 50 packets. We consider that all the frames generated by the video streams are I-frames (i.e. there are no P-frames) in order to analyze the worst-case scenario that may occur in the switch.
- The number of AVB_BE streams is fixed to 10 for all generated Stream-Sets. Each AVB_BE stream has a transmission time of 3μs and a period of 125μs.
- The number of Protected Windows in the TAS scheduling table is varied between 2 and 20.
- The cumulative percentage of Bandwidth reserved for Protected Windows is varied between 10% amd 90% in increments of 1%.
- The *sendSlopes* α_A^+ and α_B^+ are computed using Equation (26) once the number and types of streams in the set has been decided.

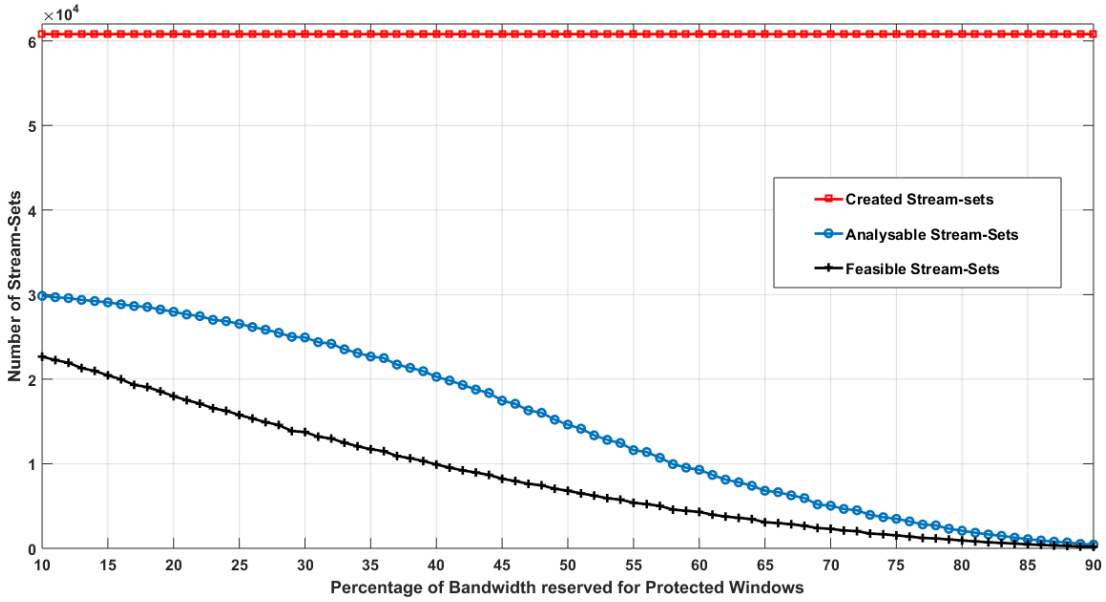


Figure 11: Number of feasible Stream-Sets, number of Analyzable Stream-Sets and number of Created Stream-Sets varying the percentage of Bandwidth reserved for Protected Windows.

Once a Stream-Set is created, the Utilisations U_A and U_B of classes AVB_A and AVB_B respectively are computed. If either $U_A > 1$ or $U_B > 1$ then the Stream-Set is marked as unfeasible without being analyzed. We note that if this is the case the *sendSlopes* α_A^+ and α_A^+ are not generated either as they are not defined for utilizations larger than 1. If both $U_A \leq 1$ and $U_B \leq 1$ then the Stream-Set is analyzed and its feasibility is assessed.

Figure 11 shows how feasibility decreases when the percentage of Bandwidth reserved for Protected Windows increases. The upper curve (red) represents the number (in this case 60800) of Stream-Sets created for each point on the X-axis. The middle curve (blue) represents the number of Stream-Sets that were analyzed. The bottom curve (black) represents the number of Stream-Sets that are deemed feasible using our analysis. This means that the last package of a video frame finishes transmission on the link within a deadline of 40000μs, before the next video frame of the same stream is available for transmission.

The structure (and colour coding) of Figure 11 is also used for Figure 12 and Figure 13.

The difference between the bottom (black) and the middle (blue) curve can be explained in several ways:

- The analysis is pessimistic in the sense that it considers that Guard Bands are completely unusable bandwidth, which is not the case. Furthermore the analysis considers that all Protected Windows that can interfere with the transmission of a frame will delay it, but this may not be the case either as the frame may already have finished transmission by the time a new Protected Window is instantiated. For example, every time the Guard Band is used by an AVB packet the delay of all subsequent packets queued in the egress port is decreased with respect to the analysis and this further accentuates the pessimistic assumption that all Protected Windows will interfere with the transmission of a given packet.

- The check $U_A \leq 1$ and $U_B \leq 1$ is performed without knowledge of the amount of Bandwidth that is reserved for Protected Windows, i.e. as if 100% of the link Bandwidth would

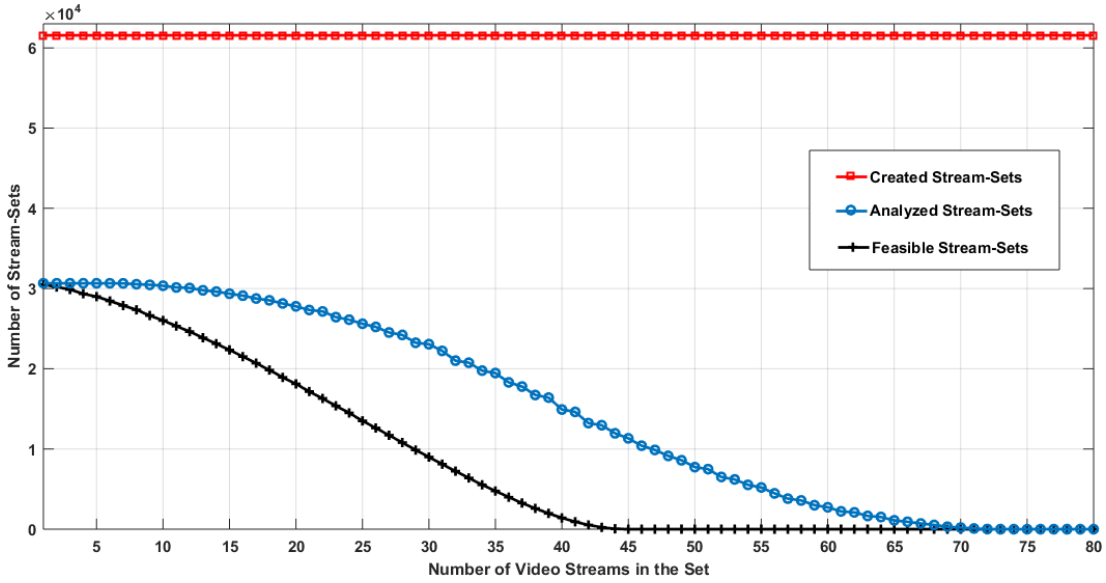


Figure 12: Number of feasible Stream-Sets, number of Analyzable Stream-Sets and number of Created Stream-Sets varying the number of Video Streams in the egress port.

be available for the AVB traffic. This means that the actual bandwidth available for the AVB traffic may not be enough to make the Stream-Set feasible, further increasing the distance between the two curves under consideration in the plot.

- The proposed analysis is a sufficient one (but not a necessary one), which means that in the gap between the two curves there are some Stream-Sets the are in fact feasible.

Figure 12 presents the effect that increasing the number of Video streams in the set has on the feasibility of the entire Stream-Set. In this case, 61560 Stream-Sets were created for each of the 80 points on the X-axis. As it was to be expected, the feasibility decreases when we increase the number of Video streams as the link gets saturated and it is more and more difficult to obtain the Bandwidth necessary to transmit all the video frames within their deadlines. The gap between the curve representing the Analyzed Stream-Sets and the Feasible Stream-Sets can be explained in the same way as in Figure 11.

Figure 13 shows how the feasibility of the Stream-Set decreases when the number of AVB_A streams in the set increases. In this case, 123120 Stream-Sets were created for each of the 40 points on the X-axis.

We note that the number of Protected Windows in a TAS cycle does not affect the theoretical Worst-Case delay of a frame as the analysis does not take into account this information, just the total amount of Bandwidth that is reserved for Protected Windows, independent of how this Bandwidth is split.

7. Concluding remarks

In this paper we have investigated the effects of time aware shaping on the transmission of CBS shaped AVB flows in TSN switches. We have presented a description of the mechanisms involved and how they interact and influence the scheduling of these flows. Based on the eligible interval approach, we proposed an analysis for computing the local delays of AVB flows

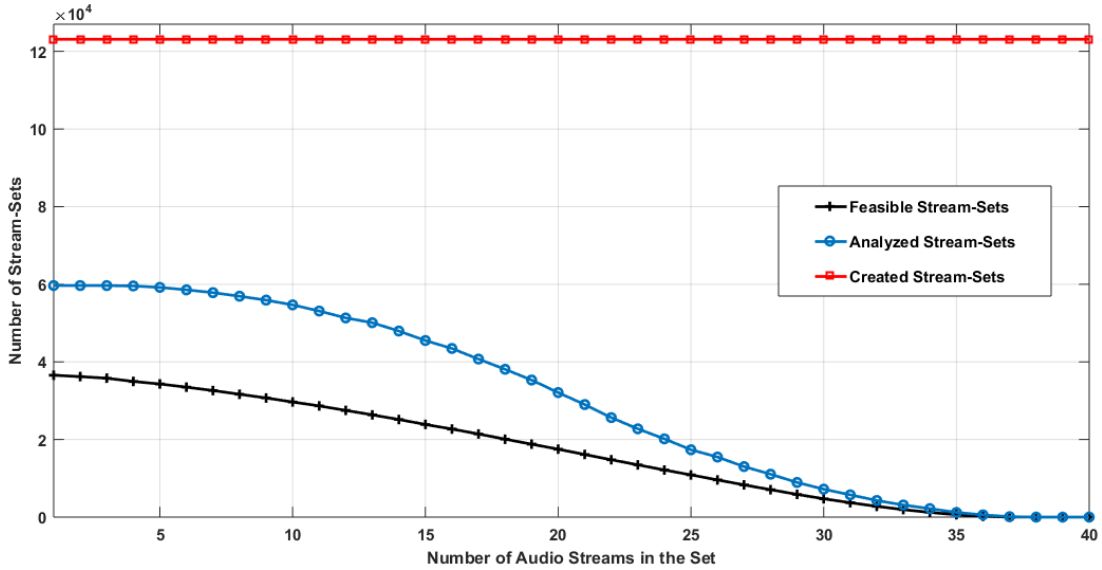


Figure 13: Number of feasible Stream-Sets, number of Analyzable Stream-Sets and number of Created Stream-Sets varying the number of Audio Streams in the egress port.

and an utilization based feasibility condition. We take an incremental approach to building and presenting the analysis, starting from the simplest case where there is a single protected window that can block the AVB packets under analysis. We then generalize the analysis to the case when there are multiple protected windows per TAS cycle and we further extend the analysis to the case when large (generally video) frames are split in multiple packets which are transmitted over several TAS cycles.

The obtained results include a formally proved tight local worst case delay bound of CBS shaped AVB streams under TAS influence, the rules of the CBS credit allocation for guaranteeing bounded delays of AVB streams, and a way to apply them to analyzing large size video frame transmission delays.

Our experiments show that the analysis is safe for all cases that we have regarded and the tightness varies depending on the load of the switch and the complexity of the TAS scheduling table, with more protected windows in the TAS cycle inducing more pessimism in the analysis.

How to extend this approach for further reducing the pessimism when applying it to multi-hop end-to-end delay analysis is our future work.

Acknowledgements

This work was partially funded by the EUROSTARS RETINA Project. The authors would like to also thank Mathias Johanson from ALKIT Communications for his inputs regarding the automotive video model adopted in Section 5.

Bibliography

[1] IEEE, 802.1qbv standard for local and metropolitan area networks, bridges and bridged networks, amendment 25: enhancement for scheduled traffic (2015).

[2] IEEE, 802.1qav forwarding and queuing enhancements for time-sensitive streams (2009).

[3] S. Thangamuthu, N. Concer, P. J. L. Cuijpers, J. J. Lukkien, Analysis of ethernet-switch traffic shapers for in-vehicle networking applications, in: 2015 Design, Automation Test in Europe Conference Exhibition (DATE), 2015, pp. 55–60. doi:10.7873/DATE.2015.0045.

[4] F. Dürr, N. G. Nayak, No-wait packet scheduling for ieee time-sensitive networks (tsn), in: Proceedings of the 24th International Conference on Real-Time Networks and Systems, RTNS '16, ACM, New York, NY, USA, 2016, pp. 203–212. doi:10.1145/2997465.2997494.
URL http://doi.acm.org/10.1145/2997465.2997494

[5] S. S. Craciunas, R. S. Oliver, M. Chmelík, W. Steiner, Scheduling real-time communication in ieee 802.1qbv time sensitive networks, in: Proceedings of the 24th International Conference on Real-Time Networks and Systems, RTNS '16, ACM, New York, NY, USA, 2016, pp. 183–192. doi:10.1145/2997465.2997470.
URL http://doi.acm.org/10.1145/2997465.2997470

[6] R. Serna Oliver, S. S. Craciunas, W. Steiner, Ieee 802.1qbv gate control list synthesis using array theory encoding, in: Proc. Real-Time and Embedded Technology and Applications Symposium (RTAS), IEEE, 2018.

[7] J. A. R. De Azua, M. Boyer, Complete modelling of avb in network calculus framework, in: Proceedings of the 22nd International Conference on Real-Time Networks and Systems, RTNS '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 55–64. doi:10.1145/2659787.2659810.
URL https://doi.org/10.1145/2659787.2659810

[8] U. D. Bordoloi, A. Aminifar, P. Eles, Z. Peng, Schedulability analysis of ethernet avb switches, in: 2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications, 2014, pp. 1–10. doi:10.1109/RTCSA.2014.6910530.

[9] X. Li, L. George, Deterministic delay analysis of avb switched ethernet networks using an extended trajectory approach, Real-Time Systems 53 (1) (2017) 121–186. doi:10.1007/s11241-016-9260-5.
URL http://dx.doi.org/10.1007/s11241-016-9260-5

[10] J. Cao, P. J. Cuijpers, R. J. Bril, J. J. Lukkien, Independent yet tight wcrt analysis for individual priority classes in ethernet avb, in: Proceedings of the 24th International Conference on Real-Time Networks and Systems, RTNS '16, ACM, New York, NY, USA, 2016, pp. 55–64. doi:10.1145/2997465.2997493.
URL http://doi.acm.org/10.1145/2997465.2997493

[11] J. Cao, P. J. L. Cuijpers, R. J. Bril, J. J. Lukkien, Tight worst-case response-time analysis for ethernet avb using eligible intervals, in: 2016 IEEE World Conference on Factory Communication Systems (WFCS), 2016, pp. 1–8. doi:10.1109/WFCS.2016.7496507.

[12] D. Thiele, R. Ernst, J. Diemer, Formal worst-case timing analysis of ethernet tsn's time-aware and peristaltic shapers, in: 2015 IEEE Vehicular Networking Conference (VNC), 2015, pp. 251–258. doi:10.1109/VNC.2015.7385584.

[13] D. Thiele, R. Ernst, Formal worst-case timing analysis of ethernet tsn's burst-limiting shaper, in: 2016 Design, Automation Test in Europe Conference Exhibition (DATE), 2016, pp. 187–192.

[14] M. Ashjaei, G. Patti, M. Behnam, T. Nolte, G. Alderisi, L. Lo Bello, Schedulability analysis of ethernet audio video bridging networks with scheduled traffic support, Real-Time Systems (2017) 1–52doi:10.1007/s11241-017-9268-5.
URL http://dx.doi.org/10.1007/s11241-017-9268-5

[15] D. Maxim, Y.-Q. Song, Delay Analysis of AVB traffic in Time-Sensitive Networks (TSN), in: RTNS 2017 - International Conference on Real-Time Networks and Systems, Grenoble, France, 2017, p. 10. doi:10.1145/3139258.3139283.
URL https://hal.inria.fr/hal-01614677

[16] L. Zhao, P. Pop, Z. Zheng, Q. Li, Timing analysis of avb traffic in tsn networks using network calculus, in: 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2018, pp. 25–36. doi:10.1109/RTAS.2018.00009.

[17] L. Zhao, P. H. M. Pop, Z. Zheng, H. Daigmorte, M. Boyer, Improving worst-case end-to-end delay analysis of multiple classes of avb traffic in tsn networks using network calculus, in: DTU Compute Technical Report, 2018.

[18] J. Cao, P. J. L. Cuijpers, R. J. Bril, J. J. Lukkien, Independent wcrt analysis for individual priority classes in ethernet avb, Real-Time Systems 54 (4) (2018) 861–911.

[19] J. Migge, J. Villanueva, N. Navet, M. Boyer, Insights on the Performance and Configuration of AVB and TSN in Automotive Ethernet Networks, in: 9th European Congress on Embedded Real Time Software and Systems (ERTS 2018), Toulouse, France, 2018.
URL https://hal.archives-ouvertes.fr/hal-01746132

[20] IEEE, 802.1as standard for local and metropolitan area networks-timing and synchronization for time-sensitive applications in bridged local area networks (2011).

[21] J. Cao, M. Ashjaei, P. J. L. Cuijpers, R. J. Bril, J. J. Lukkien, An independent yet efficient analysis of bandwidth reservation for credit-based shaping in ethernet tsn, in: 2018 IEEE World Conference on Factory Communication Systems (WFCS), 2018, pp. 1–10.

[22] M. Boyer, H. Daigmorte, Impact on credit freeze before gate closing in cbs and gcl integration into tsn, in: Proceedings of the 27th International Conference on Real-Time Networks and Systems, RTNS '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 80–89. doi:10.1145/3356401.3356412.
URL https://doi.org/10.1145/3356401.3356412

[23] V. Edpalm, A. Martins, K.-E. Årzén, M. Maggio, Camera Networks Dimensioning and Scheduling with Quasi Worst-Case Transmission Time, in: the Proceedings of the 30th Euromicro Conference on Real-Time Systems (ECRTS 2018), 2018.

[24] ITU-T, Advanced video coding for generic audiovisual services, https://www.itu.int/rec/t-rec-h.264-201704-i/en (2017).

[25] D. Maxim, L. Cucu-Grosjean, Response time analysis for fixed-priority tasks with multiple probabilistic parameters, in: RTSS, 2013, pp. 224–235.

[26] D. Maxim, A. Bertout, Analysis and simulation tools for probabilistic real-time systems, in: Proceedings of the 8th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS), 2017.

[27] N. Benammar, H. Bauer, F. Ridouard, P. Richard, Timing analysis of AVB ethernet network using the forward end-to-end delay analysis, in: Y. Ouhammou, F. Ridouard, E. Grolleau, M. Jan, M. Behnam (Eds.), Proceedings of the 26th International Conference on Real-Time Networks and Systems, RTNS 2018, Chasseneuil-du-Poitou, France, October 10-12, 2018, ACM, 2018, pp. 223–233. doi:10.1145/3273905.3273922.
URL https://doi.org/10.1145/3273905.3273922

[28] L. Zhao, P. Pop, Z. Zheng, H. Daigmorte, M. Boyer, Latency analysis of multiple classes of AVB traffic in TSN with standard credit behavior using network calculus, IEEE Trans. Ind. Electron. 68 (10) (2021) 10291–10302. doi:10.1109/TIE.2020.3021638.
URL https://doi.org/10.1109/TIE.2020.3021638