

Do We Really Need TSN in Next-Generation Helicopters? Insights From a Case-Study

Cédric Mauclair
Airbus Helicopters
Marignane, France
cedric.mauclair@airbus.com

Marina Gutiérrez, Jörn Migge
RealTime-at-Work (RTaW)
Nancy, France
{marina.gutierrez,
jorn.migge}@realtimeatwork.com

Nicolas Navet
Department of Computer Science
University of Luxembourg
& Cognifyer SàrL
Esch-sur-Alzette, Luxembourg
nicolas.navet@uni.lu

Abstract— As Ethernet rapidly replaces legacy networks as the core high-speed network in helicopter’s avionics and mission systems, we ask in this paper the question of the technical benefits of migrating to Ethernet Time-Sensitive-Networking (TSN). Indeed, TSN has become a rich toolbox of mechanisms and protocols to address Quality-of-Service (QoS) requirements pertaining to timing and reliability. TSN is quickly becoming the prominent technology for wired high-speed communications in a variety of application domains like automotive, industry 4.0 and telecom. In this context, this work explores the use of TSN timing QoS mechanisms for helicopter’s avionics and mission systems on a case-study representative of the communication requirements of next-generation systems. This study aims to provide quantified insights into what can be expected from TSN in terms of timing, memory usage and extensibility.

Keywords—*Helicopters networks, Ethernet, Time-Sensitive Networking, IEEE802.1.*

I. INTRODUCTION

Context of the study. Real-time communication with Quality-of-Service (QoS) guarantees is essential for some aerospace embedded systems like found in helicopters. In such systems, just like in many other applications domains, Ethernet is progressively becoming the prominent technology for high-speed wired communications. One of the reasons is that Ethernet has been constantly evolving and adapting to successfully address the needs of the new systems being developed. In particular, the IEEE 802.1 TSN TG (Time Sensitive Networking Technical Group), started in 2012, develops technologies to address QoS requirements pertaining to timing and reliability.

Main drivers for the adoption of Ethernet and TSN in helicopters. Ethernet, and specifically TSN networks, are seen as offering solutions to handle an increase of traffic and subscribers’ number with mixed criticality traffic like safety-critical functions, time-sensitive control algorithms distributed over the network or QoS-sensitive audio/video over IP for example. Additionally, standard Ethernet based solutions are expected to reduce costs of network devices, wiring harnesses and test facilities compared to aerospace specific solutions like Arinc 664P7.

Contributions of the paper. This work considers two basic requirements: ensuring that, in any possible circumstances, 1) communication latencies are below deadlines and 2) that the memory needed in hardware to store packets does not exceed the available memory.

The first objective of this work is to provide a review of the TSN standards that are relevant to helicopters’ avionics and mission networks, with a focus on timing. The second objective is to evaluate, on a network that is realistic with respect to a subset of next-generation’s systems, the efficiency of core TSN timing QoS mechanisms in terms of the maximum number of streams they can successfully schedule. This experimental assessment is conducted by performance evaluation using simulation and worst-case analysis, as well as by sensitivity analysis. The results of the experiments highlight several non-obvious behaviours of TSN timing QoS mechanisms, such as the limited efficiency of shapers in certain scenarios. Finally, we discuss the issue of selecting and configuring TSN mechanisms and give our views about the interest that TSN possesses for helicopters at this time.

Organisation of the paper. The remainder of this paper is organised as follows. Section 2 presents the landscape of the TSN standardisation and the main TSN standards that are relevant to helicopters. In Section 3, we describe the characteristics of the case-study that is used in the experiments of Section 4. Finally, Section 5 concludes with a summary and recommendations.

II. IEEE 802.1 STANDARDS

The IEEE 802.1 Working Group (WG) develops standards for switched Ethernet networks. The mechanisms and protocols defined in those standards rely on the services offered by the data link layer and they provide features like time synchronization, network configuration and management and QoS.

Before going into more details, it is worth pointing out that there are two different types of TSN standardization projects. First, there are *base standards*, whose short names are written all in upper-case letters (IEEE 802.1Q [1], IEEE 802.1CB [3] and IEEE 802.1AS [2]) and which undergo revisions every few years. Second, the ones adding lower-case letters to the base standard, are *amendments*, which, once finalized, are incorporated into a base standard in its next revision (e.g., IEEE 802.1Qcr published in September 2019 will be merged into IEEE 802.1Q-2021).

The 802.1 WG has been active since 1990 and throughout its history the focus of the group has evolved. From the early projects defining the basics of MAC bridges (also called “switches”) and traffic classes (IEEE 802.1D and IEEE 802.1p respectively, both incorporated into IEEE 802.1Q). To the creation of the Audio Video Bridging (AVB) Task Group (TG) in 2005 targeting the professional audio video market, that introduced a time synchronization protocol (IEEE 802.1AS-2011), the Stream Reservation Protocol (SRP, IEEE 802.1Qat) and the Credit Based Shaper (IEEE 802.1Qav). And finally, the renaming of AVB TG to the current Time-Sensitive Networking (TSN) TG in 2012, to expand the scope of application to the Automotive, Industrial Automation and Aerospace fields.

To this day the TSN TG continues developing mechanisms to provide a deterministic service, with bounded low latency (IEEE 802.1Qbv, IEEE 802.1Qbu, IEEE 802.1Qcr), reliability (IEEE 802.CB, IEEE 802.1Qci) and improved time-synchronization (IEEE 802.1AS-2020). In fact, with so many different mechanisms and features it is difficult to understand what TSN can bring to an application. To solve that problem the TSN TG is developing another type of projects called profiles. In essence, a TSN profile selects mechanisms and configuration strategies to satisfy the requirements of a given application. The first profile was IEEE 802.1BA, for audio video bridging, defining what is known as the AVB standards. Then the second TSN profile was IEEE 802.1CM for telecommunication fronthaul networks. The IEEE 60802 Industrial Automation profile, the IEEE 802.1DG Automotive profile and the IEEE 802.1DP Aerospace profile are currently under active development.

In this paper we are interested in the real-time requirements of helicopter networks, thus in the next section we analyze in detail the TSN mechanisms that provide timing QoS. Additionally, we present a first look at the TSN Aerospace Profile.

A. 802.1 Timing QoS

One of the most important requirements in real-time communication systems is to have guaranteed bounded latencies. We refer to latency as the time that a frame takes to cross the network from the sender to the receiver(s). In a network with real-time requirements, like in helicopters, it is common that streams have a maximum allowed latency, *i.e.* they are subject to deadline constraints.

1) Priorities

The first and simplest method to guarantee latencies is to assign priorities to the different streams and use the priorities to arbitrate between frames ready to be sent on the same egress port. In 802.1 that is realized through traffic classes, up to eight in a TSN network, and thus up to eight priority levels can be used. Traffic classes represent the egress queues implemented in a port of a switch or a network interface. Frames are assigned to different egress queues or traffic classes according to their priority. It is important to notice that the allocation of priorities to frames is a design decision that can be made both manually and using an algorithm [14], but such allocation shall be verified offline to ensure that it guarantees the desired latencies, and any other timing constraints like jitters and throughput.

In addition to being highly effective to meet deadline constraints at the high priority levels, traffic classification ensures that streams of different criticality levels do not share the same waiting queues. This guarantees that critical frames will not be discarded because a lower importance stream does not meet its traffic specification.

There are still however two open issues with using just priorities: high priority traffic can be blocked by other high priority traffic and low priority traffic can be blocked by high priority traffic. The latter problem can be alleviated using a traffic shaping mechanism like the Credit Based Shaper (CBS).

2) Credit Based Shaper

High priority traffic can, by definition, block low priority traffic. However, even low priority traffic may have performance requirements, typically throughput or “soft deadlines” (*i.e.*, deadlines that can be occasionally missed). Using just priorities, a burst of high priority traffic (e.g., a video frame being transmitted) or a sub-optimal configuration of the priorities can cause low priority traffic to “starve”, that is to be prevented access to the network for a long period of time. The Credit Based Shaper (CBS, defined in IEEE 802.1Qav) helps alleviate this issue. The basic idea is to associate a “transmission credit” to the egress queues / traffic classes. To be allowed to send a frame of a given traffic class, the credit value associated with that class must be positive. Once the transmission of a frame starts, the credit of the class is reduced at a rate called the “Send Slope”, which is a parameter specific to a port and a class. Once the credit of a traffic class becomes negative, then no frame of that traffic class can be transmitted anymore. While frames belonging to other traffic classes are being sent, the credit of a class is progressively replenished at a per-port per-class rate called the “Idle Slope”. By applying the CBS to one or two of the high priority queues, it is in many cases possible to prevent the starvation of lower priority traffic classes.

The other problem of using just priorities is the blocking of high priority streams caused by other high priority streams. Assuming that, as explained before, the priorities allocation allows to meet the timing requirements, this is only a problem if new traffic is added to the network, or if some traffic does not behave as expected. In such cases, high priority traffic is not protected and may suffer unexpected delays, resulting in longer latencies and missed deadlines. To solve this problem 802.1 defines mechanisms to allocate and reserve bandwidth for specific streams, like the Stream Reservation Protocol (SRP), that works on a per-hop basis, or the Scheduled Traffic (defined in IEEE 802.1Qbv). Next, we describe how the scheduled traffic works, as this mechanism offers a solution to schedule streams with stringent deadline constraints.

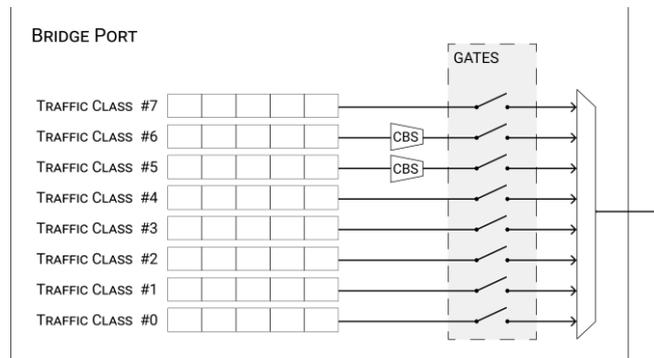


Fig. 1. Scheduling on an egress port as defined in IEEE 802.1Q. The optional QoS mechanisms are applied in a hierarchical manner: first priorities, then shaping according to the Credit Based Shaper, then the Time-Aware-Shaper, then frame preemption.

3) Scheduled Traffic

For applications with strict timing requirements, the TSN TG added in IEEE 802.1Qbv support for Scheduled Traffic, with a mechanism called the Time Aware Shaper (TAS). TAS allows managing egress queues so that frames are sent in a time-triggered manner. This is realized through logical gates associated with each egress queue. Traffic of a given class can only be transmitted if the associated gate is open. The opening and closing of the gates is controlled by a time-triggered schedule stored in the Gate Control List (GCL). The mechanism itself is quite simple but finding a right schedule is where the complexity lies. Indeed, constructing a schedule is a difficult algorithmic problem (*i.e.*, NP complete) that typically is solved using heuristics or SMT solvers [15]. The advantage of using Scheduled Traffic is twofold: first it can guarantee low latencies and low jitters and second, the traffic classes can use exclusive forwarding windows, *i.e.*, periods of time in which only the gate corresponding to a single traffic class is open. The traffic class is thus protected and the timing behavior for the packets of that class will not change even if new traffic is added to the network in other priority classes. This way of using TAS is called “exclusive gating” and, to the best of our knowledge, it is the most common way of configuring TAS at this time.

4) *Frame Preemption*

The use of exclusive windows brings a potential waste of bandwidth if the windows are not used. Additionally, an implementation of Scheduled Traffic must enforce that the transmission of a frame has ended before the gate of that traffic class is closed. This means that a buffer, called “guard band”, must be provisioned before the gate closing events. That can result in up to 1522 bytes (maximum frame size) of wasted bandwidth. To improve bandwidth utilization by reducing the guard band, Frame Preemption (introduced in IEEE 802.1Qbu) can be used in conjunction with Scheduled Traffic. When Frame Preemption is enabled, frames of the traffic classes marked as “express” can interrupt or preempt the transmission of frames of the traffic classes marked as “preemptible”. Besides improving bandwidth utilization, Frame Preemption can potentially reduce latencies of less critical traffic.

Frame preemption can also be used without Scheduled Traffic. It can actually offer an alternative to time-triggered communication with Scheduled Traffic for applications with strong deadlines. At the expense of a limited overhead (24 bytes per preemption), preemption allows to shorten the latencies of the critical streams by reducing what is commonly referred to as the “blocking factor”. The blocking factor is the maximal interference occurring at each hop due to a lower priority frame (up to about 123us per switch at 100Mbit/s). With preemption, the blocking factor is reduced by more than a factor 10 (see [16]).

B. *TSN aerospace profile*

IEEE 802 and SAE Avionics Networks AS-1 A2 are currently working together to define a TSN profile for aerospace onboard Ethernet communications. The project, IEEE P802.1DP / SAE AS 6675, has yet to publish its first draft, so it is too early to know the details of the specification. So far, the work has been focused on the collection of use-cases [4][5] and the expectation is that the TSN profile can be applied to both military and passenger aircrafts, helicopters and even satellites. For these use-cases, the joint project is looking into mechanisms to provide time-synchronization (IEEE 802.1AS), traffic shaping (IEEE 802.1Qav and IEEE 802.1Qbv), redundancy (IEEE 802.1CB), filtering and policing (IEEE 802.1Qci), along with configuration and management (IEEE 802.1Qcc). Even though at the end of the project there will be a single specification, it may contain more than one profile. For example, as suggested in [6], it could have a synchronous profile (*i.e.*, using Scheduled Traffic) and an asynchronous profile, in the spirit of AFDX.

III. REALISTIC TSN-BASED COMMUNICATION ARCHITECTURE

Our case-study is a prototype TSN-based communication architecture provided by Airbus Helicopters. The architecture comprises two Ethernet sub-networks interconnected at five end-systems (see Fig. 2). The upper part is mostly devoted to core avionics functions. It hosts most of the critical functions for the safety of the helicopter and thus most of the critical streams; it is the focus in the rest of this work. The lower part is mostly related to mission aspects. While some streams in this network are of great importance for customers to accomplish their mission, they have lower impacts on the safety of the aircraft. In the following, the term *traffic flow* and *traffic stream* refer to a sequence of frames sent from one sender to one or several receivers (*i.e.*, unicast or multicast flows).

A. *Network topology*

The upper network that is shown in Fig. 2 is a switched Ethernet network comprising two switches, 40 full-duplex links and 32 end-nodes. The data transmission rate is mixed: 100Mbit/s on 30 links and 1Gbit/s on 10 links. In terms of topology, it should be noted that not all links are connected to a switch, there are also direct point-to-point connections between end-systems.

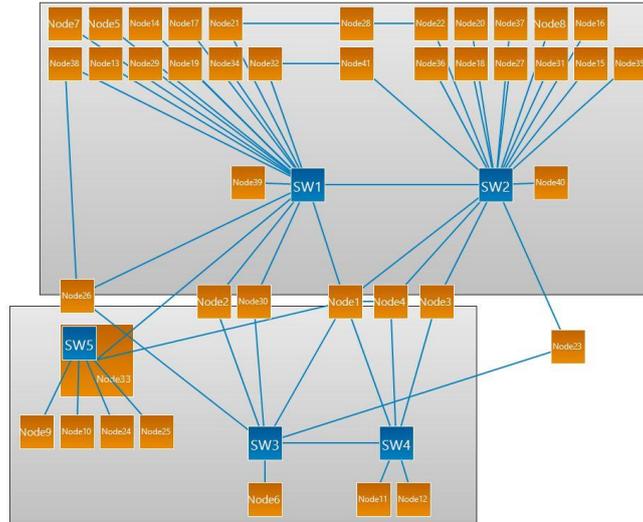


Fig. 2. The communication architecture comprises two sub-networks interconnected at several end-systems. The names of the end-systems and switches have been anonymized (RTaW-Pegase screenshot).

B. Traffic characteristics

The traffic is composed of 6 types of streams, distinguished based on their functional domain. The characteristics of the traffic are summarized in Table 1. The routing in the network is static, as it is the norm in critical systems.

TABLE I. CHARACTERISTICS OF THE 6 TYPES OF TRAFFIC. THE PERFORMANCE REQUIREMENT IS EITHER TO MEET TIMING CONSTRAINTS OR THROUGHPUT CONSTRAINTS. THE FRAME SIZES INDICATED ARE DATA PAYLOAD ONLY.

Audio streams	<ul style="list-style-type: none"> – 214 byte frames, unicast – 20ms period and deadline constraint – TSN priority: 7 (highest) – proportion (# of streams): 10.8%
Command & Control (C&C)	<ul style="list-style-type: none"> – 1480 byte frames, unicast – 20 and 40ms periods and deadlines – TSN priority: 6 – proportion: 19.1%
Video (external)	<ul style="list-style-type: none"> – 800 byte frames, unicast – burst of 6*800byte frames every 40ms, deadline is 40ms (last packet) – TSN priority: 5 – proportion: 1.1%
Data	<ul style="list-style-type: none"> – from 500 to 1480 byte frames, unicast – bursty (segmentation in 7 packets maximum) and non-bursty transmissions, from 20 to 200ms periods, deadlines equal to periods – TSN priority: 4 – proportion: 65.4%

Video (internal)	<ul style="list-style-type: none"> – 1050 to 1472 byte frames, unicast – bursty (segmentation in 12 packets maximum), 20 and 40 ms periods, deadlines equal to periods – TSN priority: 3 – proportion: 3.6%
TFTP	<ul style="list-style-type: none"> – 6 TFTP streams – Transfer of 10KB (20 packets) every second per stream – Acknowledgment delay (<i>i.e.</i>, elaboration time): 50m, each packet is acknowledged individually. – Throughput constraints: 0.01 Mbit/s – TSN priority: 2 – proportion: fixed number of streams whatever load level

In Table 1 the priorities are assigned based on the timing constraints and criticality of the functional domain to which the streams belong. This is a reasonable baseline priority allocation, which is referred to as “User Priorities” in the following.

We assume that the first packet of a stream is released with a certain transmission *offset* with respect to the start-up time of the end-system. Subsequent packets are released exactly one period apart. The offsets of all streams have been generated at random with a granularity of 5ms. Without offsets, all packets would sometimes be released at the same time (e.g. a time 0), which is not realistic as each software task typically does not have a dedicated CPU and thus packets will not all be released at the same time.

In the performance evaluation done in Section 4, the amount of traffic is progressively increased according to the proportions indicated in Table 1. Only the number of TFTP streams will remain constant whatever the network load.

IV. EFFICIENCY OF TSN SCHEDULING SOLUTIONS

In this section, we assess the relative ability of different TSN mechanisms to meet timing constraints when the network load grows. We also quantify the memory usage with the different scheduling solutions for a specific application comprising 1000 streams. All experiments have been conducted with the software RTaW-Pegase v3.9.9 (see [7]).

A. Overload analysis

A network configuration is *overloaded* if at least one of its links has a load which is greater than 100%. Such configurations are not acceptable since packets will necessarily be lost as the memory space of the network devices is not infinite.

An overload analysis is a coarse-grained evaluation, independent of the TSN QoS mechanisms, of the amount of traffic that can be supported by a given network topology. It consists in measuring the percentage of overloaded network configurations when the number of streams increases, as shown in Fig. 3. The overload analysis provides an upper bound on the network capacity as the TSN scheduling mechanisms selected may not be able to successfully schedule all the configurations that are non-overloaded.

We observe on Fig. 3 that overloaded networks only appear beyond 6000 streams, which suggests, given that deadlines are equal to periods, that the network is likely to support an important load provided an appropriate TSN scheduling solution is found. However, above 7000 flows the curve increases sharply, which sets an upper-bound on what can be expected from the network.

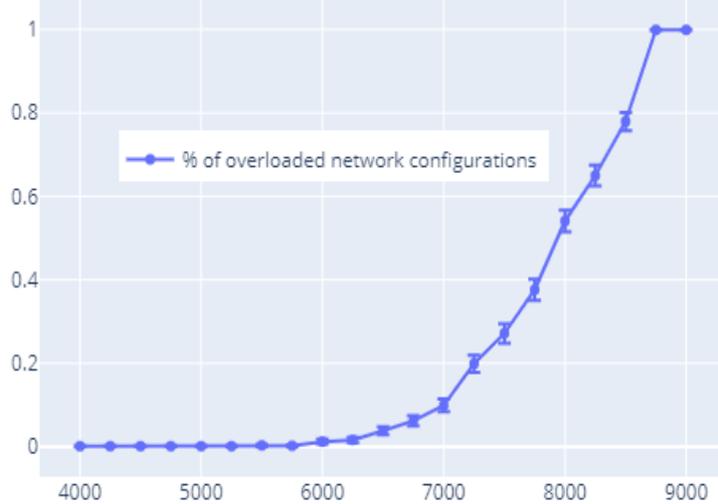


Fig. 3. Percentage of overloaded network configurations when the network load grows. The x axis is the number of flows from 0 to 9000, and the y axis the percentage of overloaded configurations. The confidence intervals of the points are shown with vertical bars. Each point is the result of 1000 experiments.

B. Priority-based TSN scheduling solutions

We compare in this paragraph the relative efficiency of different scheduling solutions, all exclusively relying on the use of priorities:

- FIFO: all streams are sharing the same priority level and thus the same waiting queue on the egress ports of the switches and end-systems.
- “Users-Priorities”: priorities are assigned based on the stream functional domain, as indicated in Table 1.
- “Concise-Priorities”: TFTP streams are assigned the lowest priority level, while the rest of the priorities are assigned by the *Concise-Priorities* (CP) algorithm available in RTaW-Pegase. CP is a variant of the Optimal Priority Assignment (OPA) algorithm initially proposed for mono-processor systems [14], which, albeit non-optimal with the schedulability analysis used in this study, performs very well in our experience.

Fig. 4 shows data obtained from a *Topology-Stress-Test*[®] (TST) execution in RTaW-Pegase. TST is a design-space-exploration function that estimates, by repeated experiments on artificial yet realistic network configurations, the probability that an application comprising a certain number of streams can be successfully scheduled with a given TSN scheduling solution. TST works by iteratively creating configurations of increasing size, configuring them, and then evaluating by schedulability analysis and simulation whether the application’s timing constraints are met (deadlines and throughput). A main purpose of TST is to perform sensitivity analysis, that is to progressively increase the network load and measure the probability that the network will still be able to successfully meet all the throughput and deadline constraints of the streams making up an application (see [8], [9] and [10] for the use of TST in automotive applications).

A first takeaway of this experiment is that “User priorities” is outperformed by FIFO. User-Priorities’ success percentage drops below 70% when the load is larger than 2000 streams, versus 3000 streams for FIFO. On the other hand, Concise Priorities’ success rate is still 71% for 7500 streams. This shows that the way priorities are assigned is key in terms of meeting timing constraints, a sub-optimal allocation being possibly worse than the use of a single priority class.

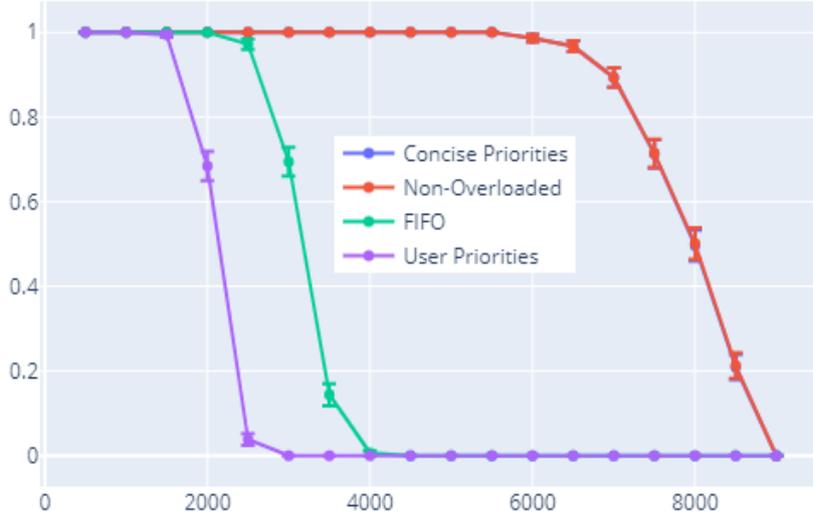


Fig. 4. Probability that a given number of streams will be successfully scheduled by different scheduling solutions relying exclusively on the use of priorities. The x axis is the number of flows from 0 to 9000, and the y axis the probability to meet all timing constraints. The confidence intervals of the points are shown with vertical bars. Each point is calculated based on 1000 experiments. It is noteworthy that the solution without the use of distinct priority levels (FIFO) outperforms “User Priorities”, which uses 6 priority levels (see Table 1). Concise Priorities, *i.e.* 8 priority allocated by an algorithm, is almost superposed with the curve of the “Non-Overloaded” network configurations, which means that it performs almost optimally for our case-study.

In our specific case-study, the success rate for User-Priorities drops quickly because of video streams at priority level 3 with 20ms latency constraints which experience far more interferences with User Priorities than with a single priority level (FIFO). Indeed, in the latter case, only the packets arrived before delay the packets of these video streams.

The second takeaway is that Concise-Priorities (CP) performs near optimally here as almost all the network configurations (e.g., 99% at a load of 8500 streams) which are not overloaded are successfully scheduled by CP. It implies that, for the network topology and traffic of the use-case considered, there is no gain in terms of meeting timing constraints in introducing additional mechanisms like shaping, time-triggered transmissions, or frame pre-emption. However, these mechanisms have an influence on the memory used in the networks as shown in §IV.D.

C. Shaping and preemption

Traffic Shaping according to the Credit-Based-Shaper (CBS) applied at a certain priority level helps improve the timing performance at lower priority levels. Shaping is essentially meant to reduce the interferences brought by higher- or same-priority bursty streams. Here, CBS applied on Video and Data streams (see Table 1) does not improve the throughput of lower-priority TFTP streams. The reason is that most of the delays of the TFTP transactions come from the 50ms it takes for the receiver to generate an acknowledgement (see Table 1). Once received, the acknowledgment triggers the transmission of the next data packet. The communication delays being more than one order of magnitude lower than the acknowledgment delay, the gain of using shaping is negligible.

Preemption and TAS essentially help reduce the interference caused by lower-priority streams (*i.e.*, one maximum size frame at each hop). In our system, neither are useful since constraints violations at top priority levels come from the interference of same priority streams (19.1% of the streams share the top priority levels, see Table 1), not lower-priority streams. What can be observed is that beyond a certain load level, many top-priority streams violate their constraints simultaneously.

CBS, Preemption and TAS being of no use in our context is an illustration that the efficiency of the different TSN QoS mechanisms depends on 1) how demanding the performance requirements are and 2) the amount of each kind of interferences among the streams making up the application. Indeed, timing interferences and thus delays are caused by higher priority, same priority and lower-priority streams, each TSN QoS mechanism allowing to control one or several types of these interferences.

D. Memory

For a scheduling solution to be feasible, it must be guaranteed that the memory available in the network devices will not be exceeded, otherwise packets would be lost. Depending on the hardware components, memory can be as much a constraint as timing in certain systems.

Fig. 5 shows the maximum memory usage measured during simulations on the egress ports of the end-systems and switches for three TSN scheduling solutions: FIFO without pre-shaping, Concise-Priorities with and without pre-shaping on all segmented messages (Video and Data, see Table 1). *Pre-shaping* is a shaping strategy that is not part of the TSN specifications per se but that can be used with TSN. It consists in inserting idle times, *i.e.* pauses, between the times at which successive frames of a segmented message, typically a camera frame, are queued for transmission. The other characteristics of the traffic remain unchanged. This strategy, applied on the sending nodes on a per flow basis, can be implemented in software. Intuitively, pre-shaping allows lower or same priority frames that cross the path of pre-shaped streams to be transmitted sooner, taking advantage of the inserted idle times. In some cases, pre-shaping can provide a simple alternative to CBS (see [13]). Pre-shaping can also be used on top of CBS, to enforce Class Measurement Intervals (CMI) and shape the flow of packets passed down the communication stack, reducing thus the memory needed in the hardware that implements CBS.

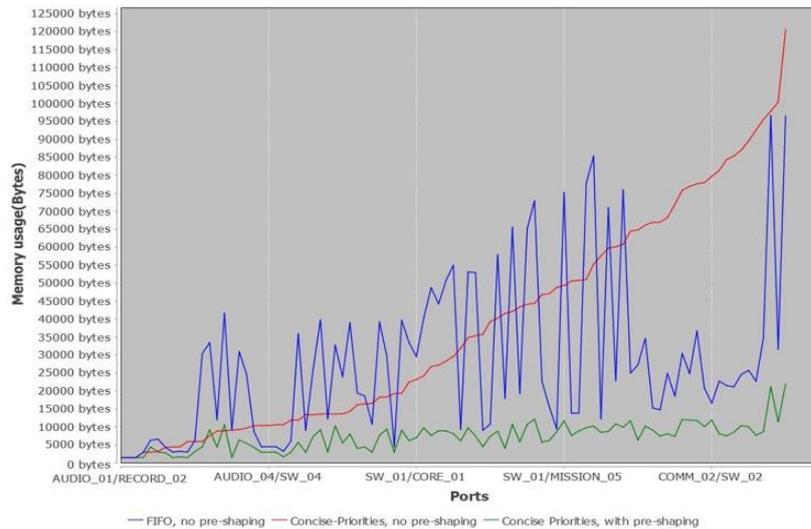


Fig. 5. Maximum memory usage (Bytes) in the output ports of the switches and end-systems with 8 priority levels allocated by Concise Priorities (CP) algorithm with and without pre-shaping (resp. green and red curve), and a single priority level (FIFO) without pre-shaping (blue curve). Pre-shaping with CP reduces the memory usage by 80.7% on average. Results obtained by simulation (RTaW-Pegase screenshot, ports are sorted so that the red curve is monotonically increasing).

As can be seen on Fig. 5, shaping is effective at reducing memory usage with an average gain of more than 80% (average over all egress ports) when used with Concise Priorities. It should be noted that, without shaping, the use of priorities does not reduce memory usage. For instance, FIFO requires on average 25.7% less memory than CP and 27.8% less memory than User Priorities. In this medium-size 1000-stream system, the total memory needed per switch with CP is up to 568KB without shaping and 168KB with shaping.

It should be noted that, in this experiment, we do not use the standardized CBS mechanism that is widely supported by commercial switches and interfaces, although CBS is known to be efficient at reducing memory usage too (see, for instance, [11] and [12]). CBS, which is per-class strategy, is an efficient solution here for User Priorities but not for CP, where all types of streams, both segmented and non-segmented messages, are mixed in the same traffic classes. As a result, non-segmented messages in CBS-shaped classes experience larger delays, which is detrimental in terms of schedulability. A solution would be to isolate the non-segmented messages into a dedicated traffic class at the immediate higher priority level, which is not required for pre-shaping that applies only on segmented messages. However, using one additional priority level per CBS-shaped class may not be feasible at high-load levels, as the eight priority levels may have been already utilized.

V. DISCUSSION AND CONCLUSIONS

TSN has become a large toolbox of 20+ standards to choose from. While IEEE 802.1CB and 802.1Qci are natural choices for dependability, the designer has many more possibilities when it comes to timing QoS. The selection and configuration of a suitable TSN scheduling solution is not straightforward, as many mechanisms are available, most of them being complex to configure (CBS, TAS, ATS) with a limited return of experience at this time. In addition, for maximal efficiency, QoS mechanisms should be used in a combined manner, which makes the configuration problem harder. For instance, priorities are typically used with some form of shaping, targeted at reducing the interferences of bursty traffic (CBS, pre-shaping) or reducing the interference of lower-priority traffic (TAS, pre-emption). Finally, it is permitted in TSN that only a selected subset of the end-nodes and switches is equipped with a certain mechanism, allowing for cost-effective implementation. The main promise of TSN is to transmit mixed-criticality traffic on a shared network, which often require different QoS mechanisms, possibly not supported uniformly all across the network. For instance, access links may not need the same mechanisms as backbone links. Also, a subset of the network supporting control loops may need extra mechanisms like pre-emption or TAS to guarantee the latencies needed by the control loops. Of course, many other concerns play a role in the selection of the TSN mechanisms: price and availability of components, risk of using a new technology, time-to-market, overall impact on the existing design and development process (e.g., with time-triggered communications), etc.

From a technical standpoint, progresses are made in the configuration of TSN protocols. In the academia, a quickly increasing body of theoretical results is available (e.g., see [15] for TAS). COTS tools like RTaW-Pegase now possesses functions to identify all combinations of mechanisms possible from a performance standpoint [9]. Upcoming TSN profiles, and specifically IEEE P802.1DP Aerospace profile, will also provide some help to designers by defining relevant subsets of mechanisms for an application domain and guidelines in using them.

In this work, on a case-study that is realistic with respect to some aspects of next-generation helicopter high-speed networks, we have made several observations:

- Provided that the adequate timing QoS mechanisms are used with the proper configuration strategy, the total network capacity in terms of streams that can be successfully scheduled is large, up to 7000 for our specific case-study,
- Depending on the characteristics of the traffic, and the performance requirements, shaping with CBS, pre-emption or TAS may not bring any significant performance boost. With the topology and the traffic of the case-study, priorities is the only mechanism needed from the timing perspective. It should be noted that priorities, developed in the 802.1p working group, largely pre-dates TSN as it has been standardized in 1998. This would allow an easy first-step solution towards the yet to come Aerospace profile.
- Memory usage in the network devices depends importantly on the TSN or pre-shaping mechanisms used, up to 80% difference in memory usage in our experiments. The need to use shaping may arise in some cases from memory constraints and not from timing constraints. Shaping can be implemented in software (pre-shaping), in hardware (CBS) or both (CBS with the use of CMI). When the implementation involves software, the cost in CPU cycles should be considered. Timing, memory, and CPU usage are constraints that should be considered in the selection of TSN mechanisms.

This study showed that TSN may not be strictly necessary yet to support realistic network configurations, even quite highly-loaded ones, when focusing mainly on timing properties. However, the experiments conducted in this study suggest that memory in switches may become the limiting factor more quickly than timing requirements for the types of streams considered. However, other features of TSN will most likely be beneficial for other types of traffic: like Scheduled Traffic for quick control loops, gPTP for metrology or to replace highly-precise time distribution (gPTP is simpler to implement than classic 1588 PTP). As previously mentioned, non-technical aspects also need to be considered like weight of redundant equipments, costs and certification efforts.

REFERENCES

- [1] Institute of Electrical and Electronics Engineers, “IEEE Std 802.1Q - Bridges and Bridged Networks”, 2018.
- [2] Institute of Electrical and Electronics Engineers, “IEEE Std 802.1AS - Timing and Synchronization for Time-Sensitive Applications”, 2020.
- [3] Institute of Electrical and Electronics Engineers, “IEEE Std 802.1CB – Frame Replication and Elimination for Reliability”, 2017.
- [4] W. Fischer, “TSN Use-Case: Commercial Aircraft Cabin”, <https://www.ieee802.org/1/files/public/docs2021/dp-fischer-aircraft-cabin-use-case-0321-v01.pdf>. Accessed July 2021.
- [5] F. Wartel, “Satellite On-board Network Use Case”, <https://www.ieee802.org/1/files/public/docs2021/dp-wartel-satellite-onboard-network-use-case-0526-v01.pdf>. Accessed July 2021.
- [6] A. Jabbar, “Two Profile Approach for IEEE/SAE 802.1DP”, <https://www.ieee802.org/1/files/public/docs2021/dp-Jabbar-two-profile-approach-0521-v01.pdf>. Accessed July 2021.
- [7] RealTime-at-Work, “RTaW-Pegase: Modeling, Simulation and automated Configuration of real-time communication architectures”. <https://www.realtimetowork.com/software/rtaw-pegase>. Accessed July 2021.
- [8] O. Creighton, N. Navet, P. Keller, J. Migge, “Towards Computer-Aided, Iterative TSN-and Ethernet based EE Architecture Design”, 2020 IEEE Standards Association (IEEE-SA) Ethernet & IP @ Automotive Technology Day, Munich, September 14-18, 2020.
- [9] N. Navet, H. H. Bengtsson, J. Migge, “Early-stage Bottleneck Identification and Removal in TSN Networks“, Automotive Ethernet Congress, February 12-13, 2020.
- [10] N. Navet, J. Villanueva, J. Migge, “Early-stage topological and technological choices for TSN-based communication architectures”, 2019 IEEE Standards Association (IEEE-SA) Ethernet & IP @ Automotive Technology Day, Detroit, USA, September 24-25, 2019.
- [11] J. Villanueva, N. Navet, J. Migge, “QoS-Predictable SOA on TSN: Insights from a Case-Study”, Automotive Ethernet Congress, February 9-11, 2021.
- [12] J. Migge, J. Villanueva, N. Navet, M. Boyer, “Insights on the performance and configuration of AVB and TSN in automotive networks”, Proc. Embedded Real-Time Software and Systems (ERTS 2018), Toulouse, France, January 31-February 2, 2018.
- [13] N. Navet, J. Migge, J. Villanueva, M. Boyer, “Pre-shaping bursty transmissions under IEEE802.1Q as a simple and efficient QoS mechanism”, SAE International Journal of Passenger Cars—Electronic and Electrical Systems, 11(3):2018.
- [14] N.C. Audsley, “On priority assignment in fixed priority scheduling”, Inform. Process. Lett. 79, 1, pp 39–44, 2001.
- [15] R. Serna Oliver, S. S. Craciunas, and W. Steiner, “IEEE 802.1Qbv Gate Control List Synthesis using Array Theory Encoding,” in Proc. IEEE Real-Time Application Symposium (RTAS), 2018.
- [16] D. Thiele, R. Ernst, “Formal worst-case performance analysis of time-sensitive Ethernet with frame preemption”, in Proc. IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), 2016.