



HAL
open science

TSN Support for Quality of Service in Space

Pierre-Julien Chaine, Marc Boyer, Claire Pagetti, Franck Wartel

► **To cite this version:**

Pierre-Julien Chaine, Marc Boyer, Claire Pagetti, Franck Wartel. TSN Support for Quality of Service in Space. 10th European Congress on Embedded Real Time Software and Systems (ERTS 2020), Jan 2020, Toulouse, France. hal-02441327

HAL Id: hal-02441327

<https://hal.science/hal-02441327>

Submitted on 15 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TSN Support for Quality of Service in Space

Pierre-Julien CHAINE
Airbus Defence & Space
Toulouse, FRANCE
pierre-julien.chaine@airbus.com

Marc BOYER
ONERA
Toulouse, FRANCE
marc.boyer@onera.fr

Claire PAGETTI
ONERA
Toulouse, FRANCE
claire.pagetti@onera.fr

Franck WARTEL
Airbus Defence & Space
Toulouse, FRANCE
franck.wartel@airbus.com

Abstract—The European spacecraft industry has developed guidelines for generic satellite development known as SAVOIR (Space AVionics Open Interface Architecture). While the current satellites on-board networks implementations are compliant with this standard, their evolution opportunities are strongly limited. New missions and new clients are always more demanding on performance on-board, leading to the conclusion that the satellite embedded network must be upgraded. One opportunity appears with Time Sensitive Networking, an IEEE Ethernet technology capable of supporting both real-time and high-bandwidth traffic. The goal of this paper is to discuss, in a qualitative study, how TSN protocols can help to integrate Quality of Service in new generation satellites.

Index Terms—Time Sensitive Network (TSN), Embedded Networks, Satellites, SAVOIR-OSRA, Ethernet

I. SATELLITE ARCHITECTURE OVERVIEW

A. Introduction

For a long time, agencies and space companies, at prime and supplier levels, have raised the need of increasing the level of reuse and standardization in spacecraft avionics systems in order to improve efficiency and reduce development costs. This has led to studies and initiatives which are now federated under the Space Avionics Open Interface Architecture (SAVOIR) [Eur16] initiative through different working groups.

SAVOIR is, in a way, a reference to which any satellite manufacturer can rely on when designing a satellite. Plus, it does not contain any industrial's proprietary information and therefore is a good support for discussions and publications dealing with satellite architectures. SAVOIR vocabulary and reference architecture will be the terminology used in the paper.

B. SAVOIR reference architecture

The SAVOIR framework defines the *SAVOIR Functional Reference Architecture* based on the needs of all kinds of *missions* (scientific, telecommunication, earth observation, etc.). For the SAVOIR Advisory Group, the term *avionic* is defined as a set of key elements of an on-board system, e.g. hardwares, softwares and interfaces of the satellite platform, in charge on ensuring the good behaviour of the satellite and handling payload data. The SAVOIR Functional Reference Architecture suggests a physical implementation for the electronic units of the platform; the On-Board Computer (OBC) architecture and the payload units. It aims at defining standard building blocks and their associated functions. It focuses on data

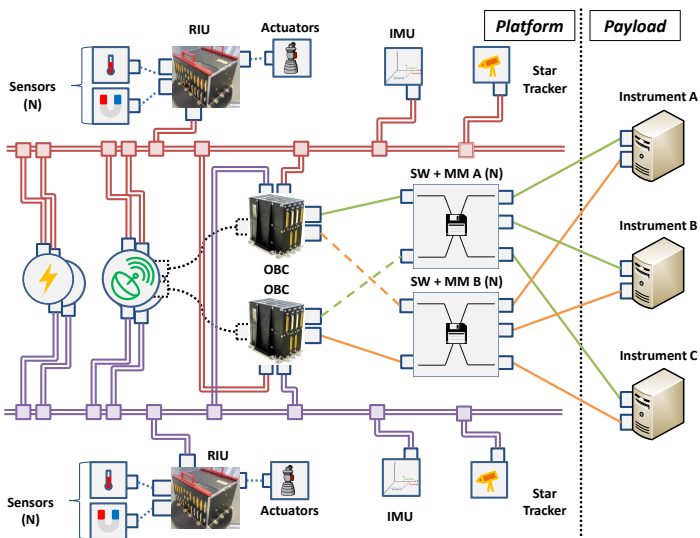


Figure 1. Traditional Satellite Network Topology

management and communications means, but also considers ECSS (*European Cooperation for Space Standardization*) compliant interfaces interconnecting the blocks. In order to be relevant to a sizeable range of mission, generic specifications, considered as a common-core of avionic specifications have been gathered in several SAVOIR documents. An example of such a requirement is *the presence of an attitude and orbit control function*.

C. Current Satellite Architecture

Our representative generic satellite network topology shown in Fig. 1 is compliant with SAVOIR reference architecture: it contains several equipments corresponding to different functions interconnected with communication links.

First of all, this architecture is split in two parts: **Payload** and **Platform**. On the one hand, the *payload* is the purpose of the satellite, it is the part of the satellite that generates added value for our clients, it is specific to each and every mission. Generally, the payload is composed of instruments such as antennas or transponders for a telecommunication satellite, telescopes or cameras for an Earth observation satellite, or any form of scientific instruments for a scientific mission (radiation sensors, magneto-sensors, etc.).

On the other hand, the *platform* is what allows the satellite to achieve its mission. If this part of the satellite does not work properly, the satellite is useless or might even be lost. The platform is composed of all the systems and subsystems that ensure a nominal behaviour of the satellite. Some notable systems are the Power Supply system, the Attitude and Orbit Control Subsystem (AOCS), the control and monitoring of the payload status, the telecommunication with ground stations system, etc.

The architecture itself has one (duplicated) *On-Board Computer* or *OBC*. This OBC is the master that manages all the platform equipments. Using a platform bus, it handles a polling mechanism to all the sensors or actuators, being eventually gathered in a *Remote Interface Unit - RIU* - also called *Data Concentrator*. OBC N, for *Nominal*, (and OBC R, for *Redundant*) is also hosting AOCS functions. To do so, it gathers information from several sensors, including one or several Star Trackers and Inertial Measurement Units, processes and exploits them in order to control the propulsion system, or the wheels and magnetorquers of the satellite. OBC is also connected to a data storage system, usually a solid state mass memory, mainly used for storing payload data, which is connected to the instruments through a payload network. Finally, it is generally in charge of routing both platform and payload telecommand and telemetry over command & control and mission datalinks i.e. between the communication subsystem and the equipments of the satellite.

In order to avoid single points of failure (for reliability and availability purposes), the network itself, platform and payload, is mirrored. There are always 2 distinct links from one device to one other. On the platform side, as well as on the payload side, there is only 1 active bus at any given time, resulting in a cold redundancy scheme. On the payload side, there is also only one active payload switch. This active switch is constantly monitored in order to be able to trigger the other switch in case of failure.

D. Current Satellite Network

Let us now focus a little bit closer on the communication links. As mentioned in the previous section, there are two interconnected networks in our generic satellite architecture: the platform and the payload networks.

On the one hand, the *platform network* is in charge of conveying all the necessary information in order to guarantee the nominal behaviour of the satellite. It transmits data from sensors (position, magnetic field, temperature) as well as, among others, flight control commands. This kind of traffic, often described as *time critical traffic*, requires bounded latency and low jitter communications. However, due to the small size and small volume of messages, a low data rate is enough to achieve the platform needs. In general, the platform network is implemented using a dual MIL-STD-1553 bus [Dep78] or CAN [ISO16] bus.

On the other hand, the *payload network* requires a very high data rate in order to convey the huge amount of raw data generated by the payload instruments such as pictures

from optical instruments (GAIA, SPOT), telemeters from a weather sensor or IoT (*Internet of Things*) data. However, the constraints are less stringent for a payload network: a delay in the packet communication path will not impact the nominal behaviour of the satellite. The payload network is based in general on SpaceWire [Eur18].

II. OPPORTUNITY FOR A UNIFIED TSN NETWORK

A. Reasons for a change

Although the actual architecture works perfectly fine, it has started to show its limits: new instruments and more generally new equipments are capable of generating gigabits of data that the network cannot handle in its current version (100Mbits/s on a SpaceWire network). Using a gigabit-capable network could allow satellite users to access this huge amount of raw data. Furthermore, SpaceWire bus is only used in the spacecraft industry, thus its development and update are quite expensive. Using a technology based on Ethernet and COTS - *Commercial-off-the-shelves* - components shared, for some parts or as a whole, with other industrial sectors (automotive, industrial automation, aeronautics, etc.) could help lower the overall price of the satellite network. Having a wide-spread technology could also facilitate the interaction between the spacecraft industry and the academic world. Finally, adding *Quality of Service* mechanisms provided by TSN could ease the integration of an increasing number of equipments on-board.

Thus, an Ethernet (and even a TSN)-based network is a path to investigate in order to define the upgrade of the satellite on-board network. However, the behaviour of Ethernet networks is often complex and non deterministic in general. Plus, the hardware available in the market has not been qualified for a space environment yet. Finally, the current platform sensors do not have any Ethernet interface, there is hence a risk of redesigning some "legacy" devices. The interest of TSN over the current technologies (1553 and SpaceWire) really needs to be discussed.

B. Challenge

To determine whether a unified TSN-based network could be a viable architecture for the next generation of satellite, independently from the already identified possible limitations (II-A), we need to answer the following question:

Is it possible to satisfy both *Platform* and *Payload* network requirements using this technology?

In order to tackle this open question, we propose to decompose it as a series of intermediate steps:

- 1) What are the requirements of the satellite network ?
- 2) What are the protocols/features offered by TSN ?
- 3) What are the valid topologies for the future on-board network?
- 4) Which combination of TSN protocols/features satisfies the satellite requirements ?
- 5) What is the minimum subset of TSN protocols/features satisfying these requirements ?

Class	Freq. scale (Hz)	Data Rate scale	Timestamp	Max. Jitter	Max. Latency	Level of determinism	QoS Level
	Min-Max	Min-Max	Mandatory/Optional	ms	ms	Low/Medium/High	0/1/2
1	0.1-1	100bits-10kbits/s	Optional	10	10	Medium	0/1
2-a	8-10	<1Mbits/s	Optional	5-10	10	High	2
2-b	8-10	<1Mbits/s	Mandatory	5-10	10	Medium	1/2
3	8-10	<250kbits/s	Optional	10	10	High	2
4	0.1-1	>100Mbits/s	Optional	<100	<100	Low/Medium	0/1
5-a	10-1000	<3Mbits/s	Optional	0.5-1	0.5	High	2
5-b	10-1000	<3Mbits/s	Mandatory	0.5-1	0.5	Medium	1/2
6	1-10	>100Mbits	Mandatory	2	10	High	1/2
7	1-10	100bits-1kbits/s	Optional	1	2	High	0/1

Figure 2. OSRA-NET Traffic Classes

We provide some answers in the next sections.

III. ON-BOARD NETWORK: CURRENT AND FUTURE REQUIREMENTS

This section introduces the requirements that a future unified network would have to fulfil. The specification of the requirements that we will use for the rest of the study will be based on two elements: the current (refined) and future (expected) requirements and OSRA-NET, an ESA-funded network specification.

A. Challenge

One may think that describing the current and future requirements would be easy, however today, these requirements are not explicitly formulated and thus they need to be refined. Indeed, in the current on-board network specification, many of them suffer from *over-specification* i.e. being more inherited from the currently deployed network technology than driven by the satellite architecture itself. Hence, the first job will consist in identifying this *over-specification*.

Once this has been done, the network requirements will be classified into two categories: *performance & real-time properties* requirements on the one hand and *FDIR - Fault Detection Isolation and Recovery* requirements on the other hand. The rest of the paper will deal with the first category of requirements only.

B. OSRA-NET: SAVOIR Network Specification

A specific SAVOIR document focuses on the network aspects of the functional architecture: OSRA-NET [Eur17]. The goal of this *On-Board Software Reference Architecture Network Communication Specification* (OSRA-NET) is to provide generic requirements related to the on-board communications needs of a satellite. It has been built by taking into account the foreseen needs for the on-board application of future missions.

Definition 1 (End-point, End-system). *Component of network that needs to transmit or receive data. It is called end-point because this component will not forward the messages it receives to another network element (by opposition to a switch). It is also called End-System (ES) in the Airbus terminology and End-Station in the IEEE terminology.*

OSRA-NET defines one communication system as the union of a number of *end points* and the *network* (with a defined topology: bus or network). This network is designed to offer a set of properties that can be guaranteed. These properties are referred to as *Quality of Service (QoS)*. Those properties group: bandwidth, communication delays, jitter (understood as communication delays variation), packet loss statistics, data stream segregation, priority management, etc. Among them, OSRA-NET defines an important property: the *QoS Level* that defines the retransmission strategy for a data exchange in case of packet loss. In fact, three QoS Levels are defined for data transfers in OSRA-NET, they will not be directly used in this paper but this knowledge will be helpful when specifying the data flows requirements between all the devices in the network. QoS level 0 can be likened to a *Best Effort* quality of service. In a communication happening at QoS 0, there is no retransmission whatsoever of a missing message on the receiving side. It is also called *QoS 0 - At Most Once* level i.e. a message cannot be received more than once by the end point. At *QoS 1 - At least Once*, a re-emission is possible in case of unsuccessful transmission or communication time-out (at application level). This QoS is supported at network level by the use of duplicates and at higher level by acknowledgements. Finally, at QoS 2, a retransmission is only possible in the case of an unsuccessful transmission (and not in case of communication time-out). In this QoS level, called *QoS 2 - Exactly once*, a message cannot be received more or less than once by the end point. It is generally used for very critical operations.

Definition 2 (QoS). *The Quality of Service for a data exchange is a set of properties that can be guaranteed during the exchange. For instance, for a given data exchange, the Quality of Service can be to have a maximum jitter of 1 microsecond for all messages of the exchange.*

It is important to emphasize that even if this QoS specification is only relevant at network level, it cannot be guaranteed if the other elements of the system do not fit this QoS requirement. For instance, having a network operating at 1Gbits per second but an end point capable of exploiting only

Requirements	Current network capabilities		Future network expected capabilities
	Platform - 1553	Payload - SpaceWire	Future unified satellite network
Number of end-systems	5-8	1-15	Req. 1
Data Rate	Low (1Mbit/s)	High (1Gbits/s)	Req. 2
Perf. Requirements	Low Latency(<10ms)	High average throughput	Req. 3
	No Jitter (<10 μ s)		
Transaction size	2-64bytes	Unlimited (typically 2048bytes)	Req. 4
Behaviour	Simple	Simple	Req. 5

Figure 3. Platform and Payload Network Performance Requirements

500Mbps per second of data will create a bottleneck and make it impossible to fulfil a 1Gbits/sec requirement at system level while it would still be fulfilled at network level.

OSRA-NET also defines a set of seven communication classes that are identified through a combination of communication specifications (frequency, message size, level of determinism, time stamp, data rate, etc.) and QoS properties (e.g. QoS level). These seven communication classes are deeply detailed in [Eur17] and are introduced in the table of Fig. 2. In Fig. 2, the communication specifications are in dark grey and the QoS properties in light grey.

Definition 3 (OSRA-NET traffic class). *An OSRA-NET traffic class is a shared set of characteristics associated to data exchange. These characteristics include the specification (such as the frequency of data exchanges, data rate) and the expected QoS properties (such as QoS level, maximum jitter, maximum end-to-end latency and level of determinism).*

C. Application of OSRA-NET to our satellite architecture

The traffic classes introduced above may not match perfectly the exact industrial requirement. However, we want to stay as close as possible to a generic satellite network specification and architecture, hence when working with a traffic flow, we will assimilate it to the closest OSRA-NET traffic class even if some values (jitter, latency, etc.) may be different. In a way, OSRA-NET will help us define "traffic families" more than an exact traffic flow specification.

Among all OSRA-NET requirements, we have selected two of them that summarize well our network requirements:

OSRA-Requirement 1. *An OSRA-NET compliant network shall be able to transfer time-critical data with timing requirements and non-time-critical data, concurrently or separately, while providing a time distribution service.*

OSRA-Requirement 2. *An OSRA-NET compliant network shall support cold, warm or hot redundancy and provide error detection at data link, network or transport layer. This requirement will not be discussed in this paper.*

D. Performance & real-time properties requirements

From our experience, we have also listed a series of requirements that the current and future networks should satisfy in terms of performance and real-time properties.

Requirement 1. *The future unified satellite network shall be capable of having up to 25 end-points.*

This is to support all the sensors and actuators of the platform plus more or less instruments depending on the mission type (Earth Observation, Telecommunication, Space exploration, etc.).

Requirement 2. *The future unified satellite network shall be capable of handling low speed traffic (around 1Mbits per second) and high speed traffic (around 1Gbits per second) concurrently or separately.*

This is to handle the *platform* and *payload* traffic.

Requirement 3. *The future unified satellite network shall be capable of handling low latency, low/very low jitter traffic and high average throughput traffic, concurrently or separately.*

Again, this is to handle the time critical messages of the platform with the huge volume of data of the payload.

Requirement 4. *The future unified satellite network shall be capable of handling small and large frame payload.*

Messages of the *platform* are generally small whereas messages in the payload are somewhat big.

Requirement 5. *The future unified satellite network shall have a simple behaviour, that is easy to analyse and configure.*

The rationale for this requirement is quite obvious: the simpler the system is, the easier it is to validate and certify.

The table in Fig 3 gives an overview of the very first *performance & real-time properties* requirements that we have identified for the satellite network. It is still a work in progress: the current requirements may be clarified and new ones may be added to the list. The goal is now to find whether TSN is indeed capable of satisfying the previous requirements.

E. Future work

Currently, we have on one hand generic traffic classes with network performance requirements and on the other hand some high level requirements. The remaining job is to instantiate these requirements on a specific topology in order to perform a network performance analysis. This will help enlighten *bottle-necks* or difficult points in the network that will need a specific, in detail, analysis. To do such instantiation, we

have already identified all the messages exchanged between all end-points and defined an encapsulation strategy. We are now currently struggling in assigning each message to a traffic class. Hopefully, this should be solved by the end of the year.

IV. TSN: A SOLUTION FOR A UNIFIED NETWORK?

A. Compliance of TSN w.r.t. the network performance requirements

This section is dedicated to showing how TSN protocols and associated mechanisms could solve the previously introduced requirements. The table of Fig. 4 summarizes what we will discuss in this section.

Requirements	TSN Capabilities
Req. 1	✓
Req. 2	✓
Req. 3	?
Req. 4	✓
Req. 5	?

Figure 4. TSN capabilities with respect to the satellite network performance requirements

Satisfying the first requirement with TSN is really straightforward: the number of supported end-points in TSN is theoretically limited by the number of available MAC addresses in Ethernet, which is way higher than what we are requesting for a space use. In reality, it will be more likely less than the number of available MAC addresses because switch manufacturers may have limitations in the number of streams that can be supported in their devices. However, our system is pretty small compared to other industrial applications. Thus we believe that we will not be affected by this limitation and hence that Req. 1 is satisfied using TSN.

The second requirement deals with the throughput available in the future network. The requirement is at least 1Gbits/s. Ethernet TSN can support this data rate and even more as some manufacturers already showcase 10Gbits/s capable switches. As a consequence, we consider that Req. 2 is satisfied using TSN.

The third requirements states that the future unified on-board network shall be capable of handling both low latency, low jitter traffic and high throughput traffic. This is the million dollar question for all network systems. We will discuss the capabilities of Time Sensitive Networking with respect to Req. 3 in section V.

The fourth requirement requires that TSN shall be able to handle large and small payloads (in the sense of size of data transfers). This is something possible with Ethernet TSN: an Ethernet frame is capable of handling a data payload between 46 and 1500 bytes (jumbo frames are not in the perimeter of this paper). Hence directly, Ethernet TSN is a solution for data payloads between 46 and 1500 bytes. For payload bigger than 1500 bytes, we will have to implement a fragmentation strategy at higher ISO level (Network or Transport); one solution for that could be to use IP fragmentation. For the

smallest payloads (under 46 bytes), we will still be able to use Ethernet but some padding will be added to the data payload in order to reach the minimum Ethernet payload size (46 bytes). This may add a significant overhead (62 bytes) to the data (2 bytes) we want to send (42 bytes for padding + 20 bytes for the Ethernet header). However, our calculations lead us to the following result: considering a 1Gbits/s physical link for TSN, it takes less time to transfer those 64 bytes than to transfer the 2 bytes on a legacy low speed bus. As a consequence, we consider that Req. 4 is satisfied. The reader may comment that we could consider grouping some messages in order to reduce this overhead. This is a point that we might consider in the future, but for now, we would like to reduce as much as possible the impact on the legacy equipments i.e. do not modify the software as much as possible. That is why we will keep these 2 bytes data payloads in this study.

The fifth network performance requirement says that the future unified on-board network must have a simple behaviour that is easy to analyse. At first glance, this is not guaranteed at all with TSN. IEEE Time Sensitive Networking is a wide "standard", composed by more than 20 published protocols and still others are on their way, along with some revisions. Having a system implementing all TSN protocols will, for sure, be extremely complex to analyse and certify. If a lot of complex protocols are required for a single and simple performance requirement, this might be a show-stopper for TSN. That is why, in the challenge introduced in section II-B, we decided to search the minimum subset of TSN protocols that satisfies all the network requirements. We hope that the fewer protocols involved, the easier the analysis will be. The rationale for this "complexity" requirement will not be further discussed in this paper.

B. Which topology for a Unified TSN-based Network ?

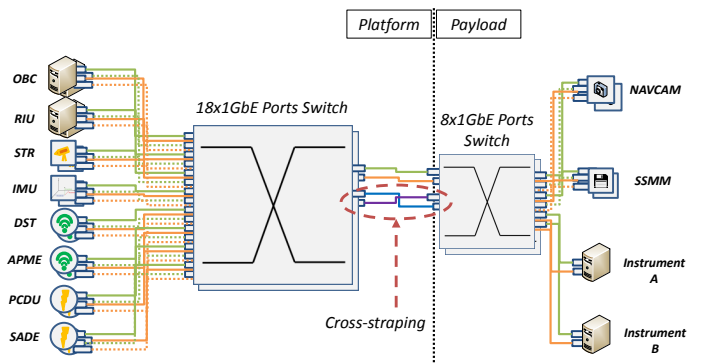


Figure 5. Possible network topology using TSN

Any new topology must be compliant with SAVOIR requirements. The question of how to organize the switches and links is still open at this stage and will be explored in the next year. For instance there could be two independent networks : one for Platform and one for Payload; or only one network

conveying both *Platform* and *Payload* messages. One example of such network is given in Fig 5.

Definition 4 (Topology). A topology is a network organization of all the devices of the satellite, including End-Systems and switches.

The trade-off between topologies is complex because it is not limited only to matching network device capabilities with network requirements. It must also take into account the number of devices in the final topology, their weight, their power consumption, their thermal properties, and other parameters. The goal here would be to find a topology that is at least equivalent if not better, with respect to the parameters cited above, than the current topology.

C. Overview of TSN protocols

For the last years, an IEEE Ethernet technology capable of supporting both real-time and high-bandwidth traffic has been defined in the working group TSN (Time Sensitive Networking) [IEE16], continuing the work of the former AVB (Audio Video Broadcasting) working group. Founded in 2012, the working group has already published a dozen of amendments to the 802.1 standard family in order to ensure a behaviour that is simultaneously real-time, adaptive and flexible, mixing synchronous and asynchronous approaches. Fig. 6 summarizes the available TSN protocols/features. This list is not exhaustive as some protocols are still under development today, however, it seems to be the TSN core protocols.

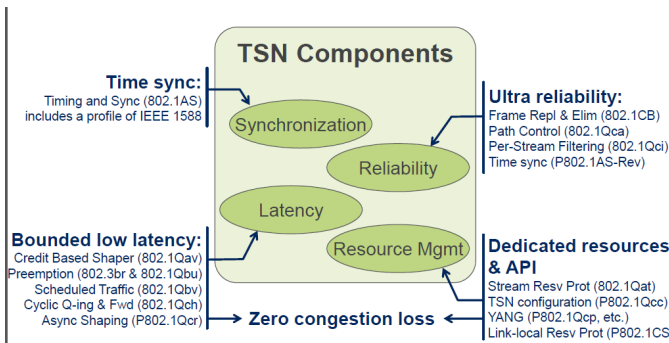


Figure 6. TSN available services, excerpt from [Jan18]

From all available TSN addenda, five main directions/families have been identified. The first offers a network level synchronization service and it groups 802.1AS, 802.1AS-rev. 802.1AS defines synchronization and time distribution protocols for a TSN network. 802.1AS-rev defines upgrades to 802.1AS, mainly specifying a redundancy protocol for the synchronization service.

The second family concerns *Ultra Reliability* (802.1CB, 802.1Qca, 802.1Qci, 802.1As-Rev). It aims at preventing, as much as possible, the loss of a frame a application level by duplicating frames and/or by controlling that bandwidth reservation is respected by all streams. 802.1CB is a protocol used to support seamless network redundancy. 802.1Qca

defines explicit path control and bandwidth and resource reservation protocol. 802.1Qci defines ingress policing strategies for TSN switches and end-points.

The third family is the *Bounded Low Latency* (802.1Qav, 802.1Qbu, 802.1Qbv, 802.1Qch, 802.1Qcr). It aims at providing protocols that a bounded end-to-end (network) latency for specific streams in the TSN network. This can be made possible through bandwidth reservation, traffic shaping (synchronous and asynchronous) and preemption strategies. 802.1Qav defines traffic shaping strategies for TSN switches and end-points. 802.1Qbu defines a preemption protocol at ISO Layer 2 (Ethernet frame level). 802.1Qbv refines and upgrades 802.1Qav. 802.1Qch is a combination of other TSN protocols, aiming at building a TSN network with fixed latency and jitter. 802.1Qcr defines an asynchronous traffic shaping strategy for TSN switches and end-points.

The fourth family is *Dedicated Resources and API* (802.1Qat, 802.1Qcc, 802.1Qcp). It defines resource management protocols as well as configuration strategies for a TSN network. 802.1Qat defines a resource reservation protocol for TSN. This can be done statically or dynamically. 802.1Qcc refines and upgrades 802.1Qat, it also defines a configuration protocol for TSN. 802.1Qcp defines a standardized model (YANG model) used to describe a TSN network, the capabilities of its devices, and potentially its configuration.

The last family is *Zero Congestion Loss* (802.1Qav, 802.1Qbu, 802.1Qbv, 802.1Qch, 802.1Qcr, 802.1Qat, 802.1Qcc, 802.1Qcp). It aims at guaranteeing that no frames are lost due to congestion i.e. buffer overflow in switches. There are not much more details on this family as its protocols have already been introduced in other families.

D. Definitions

Let us introduce some vocabulary and notations, useful for the remainder of the paper.

Definition 5 (Protocol). A protocol refers to a standard or a substandard. For instance, 802.1Qbv is a protocol.

Definition 6 (Mechanism). A mechanism (or feature) is a part of a protocol. For instance, Time Aware Shaper (TAS) is a mechanism offered by 802.1Qbv protocol. It is represented in Fig. 7 by the block Transmission Gate.

Definition 7 (Parameter). A parameter is a variable in a specific mechanism of a specific protocol that can have several values. For instance, the Transmission Selection Mechanism of 802.1Qbv protocol have one parameter "Preemption" that can take the value Preemptive or Non-Preemptive. This will lead to two different 802.1Qbv configuration.

Definition 8 (Configuration). A configuration consists of a selection of parameters' values for a specific protocol. Ex.

specifying all parameters for Transmission Selection Algorithm, Transmission Gate and Transmission Selection for each network element is a configuration for TSN protocol 802.1Qbv.

Definition 9 (Streams). A stream is multicast data transmission i.e. a unidirectional data transmission between one sender and one or several receivers.

V. MIXING DIFFERENT TRAFFIC WITH TSN-802.1QBV

This section will enlighten how one or several configuration of TSN protocol IEEE 802.1Qbv - *Enhancement for Scheduled Traffic* could satisfy Req. 3. In order to understand how to validate the previous statement, let us first give an overview of 802.1Qbv before applying it to a simple network example.

A. Overview of TSN 802.1Qbv protocol

IEEE 802.1Qbv - Amendment 25: *Enhancement for Scheduled Traffic* is a protocol that provides enhancement to the forwarding process of bridges and end-points. Fig. 7 shows a TSN Qbv-compliant switch output port. This port is composed of four elements: Queues, Transmission Selection Algorithms (TSA), Transmission Gates and Transmission Selection. Finding a configuration for a Qbv port is equivalent to finding a configuration for each of the four elements listed above.

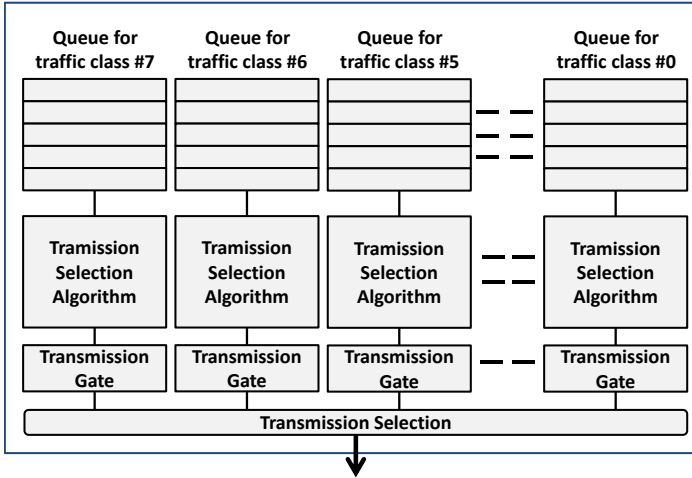


Figure 7. Switch Qbv Output Port

Let us consider a motivating example and try to find the right Qbv configuration (i.e. picking a value for each of the four elements) for it.

The network of Fig. 8 has 3 end-points (ES), 1 Switch (SW) the output port of which uses TSN-802.1Qbv protocol, all of them are 1Gbits/s capable. The physical medium is also capable of conveying a 1 Gbits/s traffic load. The nature of the physical medium (optical or copper) is outside of the scope of this study. ES1 sends video traffic to ES3 and ES2 sends C&C traffic to ES3. The characteristics and requirements of both flows are listed in the table in Fig. 9. The priority field correspond to OSRA-NET *priorities management* Quality of Service property. However, it did not appear in the definition

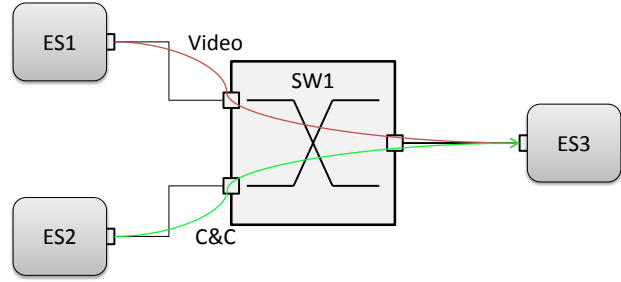


Figure 8. Motivating network example

Type	C&C	Video
OSRA-Net Traffic Type Family	OSRA-4	OSRA-5a
Size	512bits	10Mbits
Frequency	100Hz	8Hz
Max. Latency requirement	1ms	100ms
Max. Jitter requirement	2 μ s	100ms
Priority	High	Low

Figure 9. Traffic characteristics and requirements

of the OSRA-NET traffic classes. In fact it is used, by the system designer, in order to define a priority order between streams. In this motivating example, we specify that the C&C stream is more important than the Video stream.

B. Associating the traffic classes to the traffic types

First of all, a port can have between 1 and 8 output queues (implementation dependent), associated to 1-8 traffic classes.

Definition 10 (TSN traffic type, Qbv traffic class). A difference shall be made between TSN traffic type :

- TSN traffic types correspond to the traffic specification of the TSN network. In the OSRA-NET vocabulary, TSN traffic types correspond to OSRA-NET traffic classes.
- Qbv traffic classes corresponds to the 8 available queues of a specific 802.1Qbv output port (End-point or Switch).

Definition 11 (Time critical traffic). In this paper, time critical traffic is understood as traffic with real time constraints i.e. bounded low latency and low jitter. By contrast, best effort traffic barely has any constraints on latency nor jitter.

These queues are ordered by priority i.e. priority of queue 0 is higher than priority of queue 7. The allocation challenge is to find how to associate traffic types or messages to one specific queue i.e. finding a traffic type to traffic class mapping. For instance, one possible mapping could be to have all time critical messages in one queue and the rest of the traffic in the other queues. However, one might then wonder if this mapping is the best one ? (in terms of memory usage, waiting time, etc.). It is all the more difficult that this mapping can change on the path of a specific stream (of a certain traffic type).

For the motivation example, we assign the *high* priority queue (i.e. queue 0) to C&C and *low* to Video. There is already some literature existing on this subject, for instance, [Voi19]

proposes a Tabu-search algorithm to assign streams (classified in Hard Real Time, Soft Real Time and Non Critical) into Qbv traffic classes.

C. Configuring the output port

For the motivating example, we need to configure the following Qbv elements : *Transmission Selection Algorithm (TSA)*, *Transmission Gates* and *Preemption*. The parameters and their possible values are summarized in Fig 10, a schematic representation of SW1 output port.

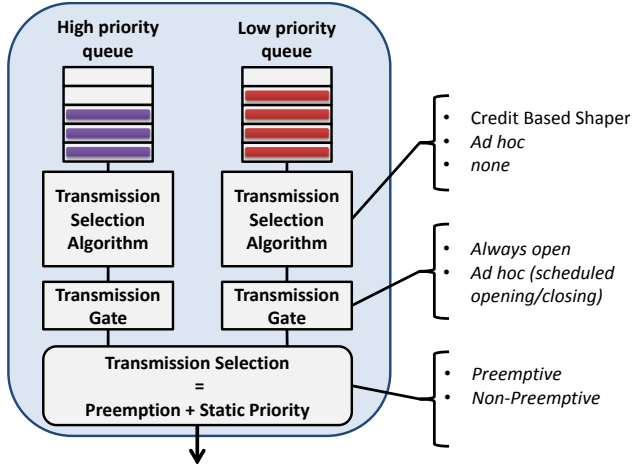


Figure 10. SW1 Output port schematic view

1) *Configuring the TSA algorithms*: Each queue within an output port can have a specific transmission selection algorithm. These algorithms decide when a queue is allowed to try to access the medium in order to send its frames. Example of these shapers can be Credit Based Shaper (IEEE 802.1Qav), Asynchronous Traffic Shaper (IEEE 802.1Qcr), Round Robin, and user defined (*ad hoc*) algorithm. The goal of these shapers is to define a formal way to share the bandwidth between all the queues of the output port. The challenge here is to find which TSA to associate to which queues. A well known and studied configuration is the TT-CBS-BE configuration where the highest priority queue has no TSA, it uses transmission gates to guarantee a certain bandwidth for its queue (see after), the two following use Credit-Based Shaper and the rest share the bandwidth using strict priority. In particular, the modelling and performance analysis of this configuration has been addressed in [DBZ18].

2) *Configuring the transmission gates*: The second element is the transmission gate. Again, each queue within an output port is associated with a specific transmission gate. As for the Transmission Selection Algorithm, the transmission gate can be used to define when a specific queue is granted the right to try to access the medium. These gates can be configured individually with a specific schedule consisting in closing and opening sequences. Basically, these gates will be, in most cases, used to create a "time-triggered" (TT) queue and ensure that no other queues are granted access to the medium while

the TT queue is emitting ("*exclusive gating*"). Finding the right schedule for the gate (i.e. finding a *Gate Control List*) is a complex open problem. It has already been addressed in [COCS16]

3) *Configuring the transmission selection block*: Finally, the last element is the transmission selection block. This block is the final element that decides which queue is, in the end, granted access to the medium. It works with static priority. From all the queues that are available for transmission (i.e. their gate is open and their transmission selection algorithm allows them to try to access the medium), this block chooses the queue of highest priority first. This element is coupled with another configurable TSN protocol, IEEE 802.1Qbu - *Frame Preemption* (and 802.3br) for improved performances [TE16].

VI. CONFIGURATION ANALYSIS FOR THE MOTIVATING EXAMPLE

Already, with only 3 mechanisms and their possible values, many 802.1Qbv configurations exist and each of them should be analysed to see if Video and C&C requirements are met. In the sequel, we consider and analyse 3 configurations, detailed in Fig 11.

A. First configuration: Static Priority

In this first configuration, the TSN port has no transmission selection algorithm configured, no gate control list configured, and no frame preemption. In that case, this port is equivalent to a port with two queues arbitrated using static priority. In fact, it behaves like a standard Ethernet switch port would i.e. without any TSN or AVB protocols.

As specified in Fig. 3, for this example, C&C traffic has a higher priority than Video traffic. In terms of delay:

- The best-case scenario for this configuration is the following: a C&C frames arrives in the switch, no other packet is being processed or stored by the switch and the frame is "*immediately*" transmitted to ES3. In this best-case, the delay induced by SW1 is its technological latency. Let us assume that this latency is not significant when compared to the frame's delay requirements.
- The worst-case scenario is as follows: if a C&C frame arrives just after the beginning of the transmission of a Video frame of 1518 bytes (max. length for an Ethernet frame without *inter frame gap* and *start frame delimiter*), the C&C frame will have to wait for that frame to be transmitted before being granted access to the medium. The waiting time is obtained by computing the formula below:

$$\frac{MessageSize}{LinkSpeed} = \frac{1518 * 8}{1.10^9} = 12.144\mu s$$

In this configuration, the jitter induced by the switch, understood as delay variability, for a C&C frame, has a value of 12.144 μs , which is higher than the maximum admissible jitter for this class of traffic. Obviously, this is only one part of the delay analysis that should be realised *End-to-End* on

Configuration	1 (see .VI-A)	2 (see .VI-B)	3 (see .VI-C)
TSA	<i>none</i>	<i>none</i>	<i>none</i>
Transmission Gates	<i>always open</i>	<i>always open</i>	<i>Ad Hoc</i>
Preemption	<i>non-preemptive</i>	<i>preemptive</i>	<i>preemptive</i>

Figure 11. 802.1Qbv summary

this network example for this configuration. Nevertheless, if the jitter is already higher than the requirements in the switch only, the rest of the analysis does not matter: this configuration is declared not valid and Req. 3 is not satisfied.

B. Second configuration: Static Priority + Frame Preemption

In the first configuration, the length of the Video frame, in the worst case, was too big; so was the C&C frames' jitter. One solution to reduce this jitter would be to reduce the size of the Video frame. However, this will lead to a significant overhead hence a reduction of the usable bandwidth in the network. We do not consider this path as a good idea to reduce the jitter.

Fortunately, TSN introduced 802.1Qbv (with the help of 802.3br): a protocol dedicated to *Frame Preemption*. Thanks to this feature, an Ethernet frame of lower priority can now be preempted by a higher priority frame. With this new protocol, the length of the shortest non-preemptable frame drops down to 143 bytes with a small impact on the available bandwidth [TE16]. By re-applying (VI-A) with this new frame size, the delay is reduced to $1.144\mu s$.

This result looks appealing but again, this jitter is only the jitter induced by the switch and the end-to-end analysis must be realised to validate that is it compatible with with C&C traffic class jitter requirement. Let us assume that this is the case. Considering that C&C traffic has its constraints satisfied, it is necessary to check that Video frames still have enough bandwidth. C&C traffic uses less than 1% of the available bandwidth and Video requires 1%. Hence there should be enough bandwidth available for Video frames and Video constraints are fulfilled (formal analysis pending). As a result, this configuration is a valid configuration with respect to the initial goal and Req. 3 is satisfied.

C. Third configuration: Use of gate control lists

The previous section showcased a first Qbv configuration that seems to satisfying the network performance requirements of our motivating example. Let us now slightly modify our motivating example (changes shown in grey in 12). In this new network specification, the *Command&Control* frames now require an End-to-End jitter of $1\mu s$.

With this new requirement, the solution presented in section VI-B is not valid any-more. The jitter, at switch level is greater than the End-to-End required jitter. There is no possibility to reduce even more the size of the smallest non-preemptable frame (because it would lead to not respecting Ethernet minimum frame size). We must then explore a new direction to try to even more reduce the jitter. From there, two paths are possible: using Transmission Selection Algorithm or

Type	C&C	Video
OSRA-Net Traffic Type Family	OSRA-4	OSRA-5a
Size	512bits	10Mbits
Frequency	100Hz	8Hz
Max. Latency requirement	1ms	100ms
Max. Jitter requirement	$1\mu s$	100ms
Priority	High	Low

Figure 12. Traffic characteristics and requirements

using Transmission Gates. In this paper, we will introduce the second path.

For this new possible configuration, we will still do not use any Transmission Selection Algorithm and we will keep preemption. In addition, we will use a Gate Control List. The goal now is to find a gate schedule i.e. a sequence of gate closing and opening for both queues that allows to satisfy the jitter requirements while guaranteeing enough bandwidth for the Video traffic.

Learning from the last two configurations, jitter appeared because of the interaction between frames from different traffic classes. As a consequence, the first thing that we can do to configure the queues to work in "*exclusive gating*". This means that when the gate for one queue is open, the other one is always closed. This will prevent the interaction between traffic classes.

The only thing left, which is not easy, is to compute a schedule of gate for our streams. A gate control list has several configurable parameters:

- **Time Cycle**, the duration of the whole sequence of gate actions
- **Number of states**, the number of gate states in the Time Cycle
- **Time interval**, the duration of a gate state in the Time Cycle
- **State**, the status of the gate, it can be Open (o) or Closed (C)

Again, finding a configuration for the Gate Control List corresponds to choosing values to each of these parameters.

With this motivating example, let us take a very simple and naive configuration for this protocol (we will not justify this configuration in the paper). In a Time Cycle of $100\mu s$, there will be 10 states of $10\mu s$ each. During the first four states, the gate for C&C will be open, and the gate for Video will be closed. During the following six states, the gate for C&C will be closed and the gate for Video will be open. This configuration is summarized by the drawing below (Fig. 13).

With this *Gate Control List* configuration, if a C&C frame arrives in the switch output port right at the time when its associated gate opens, then the frame will be immediately given access to the medium. In that situation, the jitter induced by the switch is minimal and fits our requirements. However, using this configuration is also very risky. In fact, if a C&C frame arrives just after its associated gate closes, it will have to wait $60\mu\text{s}$ before being granted access to the medium which does not match the network requirements at all !

In order to avoid this problem, two things must be put in place. First of all, all the devices along the "time-triggered" path in the network must be synchronized in order to ensure that the gate will open at the right time i.e. at the time when the C&C frame is supposed to arrive at the output port of the device (will have to be computed *a priori*). The second thing that must be realized is to set up a software-network synchronization. The reason for this need is that the End-to-End jitter requirement considered for the satellite is not at network level but at application level. Therefore, this means that the variability of the transmission time must be bounded from the sending to the receiving application and not end-point. To ensure that when an application emits a packet, it arrives at the end-point output port right when its associated gate is open, this synchronization must be defined.

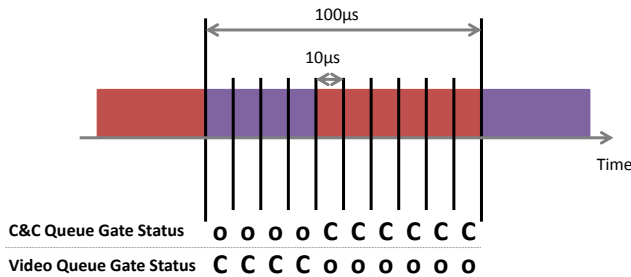


Figure 13. GCL-Configuration

If both of those points are set, then this configuration of TSN will:

- First, be compliant with Req. 3 and OSRA-Req. 1. We did not discuss the time distribution requirement but we assume that if the network is synchronized, then time is distributed across the network.
- Second, provide a "zero jitter" communication path for the C&C frames.

We did not analyse in this last configuration the requirements of the Video traffic, but the configuration was thought to provide enough bandwidth for the Video traffic requirement.

D. Way Forward

The last section gave a hint of the level of complexity associated with the upgrade of the satellite on-board network using TSN, only focusing on a very basic motivating example with two traffic classes and a very low load of the network. With this example, Standard Ethernet (using static priority) is not enough for a future satellite network. One simple

asynchronous solution relies on TSN Frame Preemption and offer a minimal jitter of $1.144\mu\text{s}$ at low cost on a 1Gbits/s network. For applications demanding an even higher jitter, one possible solution relies on using 802.1Qbv synchronous feature i.e. configure a schedule for the transmission gates. The analysis is still pending but it could potentially help building a "zero-jitter" path under for the C&C flow under the conditions explained before. Going to a synchronous TSN configuration however implies using time distribution and time synchronization services at both network and application layer in the current architecture. Although it does not look like a big issue, we fear it might heavily impact the FDIR analysis during satellite design.

What is left to do is:

- 1) Generalize the configuration search process to all the satellite requirements and all the TSN protocols, instantiated on a pre-determined topology like Fig. 5. This could lead to the development of a configuration software, similar to the one being developed by switch manufacturers.
- 2) Define a way to choose between several possible configurations. The last section of this paper will propose a methodology for how to choose between configurations.

VII. CONCLUSION

In this paper, we have introduced generic satellite network requirements with the help of OSRA-NET, ESA's network specification. We have then identified TSN as a good opportunity for the satellite network upgrade. In the last sections, we have started to show how TSN could indeed satisfy the satellite network requirements. Finally, we underlined the configuration challenge of TSN. It is a complex problem and we conjecture it to be NP-Hard. Thus scalability will be a huge issue. Once some configurations have been computed, one must be selected to be part of the final network implementation. How to choose one configuration from one another is an interesting question, specific to every vertical i.e. every field of application of TSN. Our future work will consist in finishing the network requirements definition while going further by formally tackling the configuration challenge of TSN. We will also evaluate TSN-ready hardware components in order to estimate the gap between standard TSN and its implementations.

REFERENCES

- [COCS16] Silviu S. Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried 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*, 2016.
- [DBZ18] Hugo Daigormte, Marc Boyer, and Luxi Zhao. Modelling in network calculus a TSN architecture mixing Time-Triggered, Credit Based Shaper and Best-Effort queues. working paper or preprint, 2018.
- [Dep78] Department of Defense – DoD. MIL-STD-1553B – Aircraft Internal Time Division Command/Response Multiplex Data Bus. Technical report, 1978.
- [Eur16] European Space Agency. SAVOIR Functional Reference Architecture. Technical Report Technical Note 001, ESA, 2016. <https://essr.esa.int/project/savoir>.

- [Eur17] European Space Agency. OSRA Communication Network Specification. Technical Report Technical Note 003, ESA, 2017. <https://essr.esa.int/project/savoir>.
- [Eur18] European Space Agency. Spacewire-Links, nodes, routers and network. Technical Report ECSS-E-ST-50-12C, ESA, 2018.
- [IEE16] IEEE. Time Sensitive Networking Task Group. Technical report, 2016. <https://1.ieee802.org/tsn/>.
- [ISO16] ISO 11898-2. Road vehicles – Controller area network (CAN). Technical report, 2016.
- [Jan18] Janos Farkas. Introduction to 802.1, Focus on the Time-Sensitive Network Task Group. 2018.
- [TE16] Daniel Thiele and Rolf Ernst. Formal worst-case performance analysis of time-sensitive ethernet with frame preemption. In *21st IEEE International Conference on Emerging Technologies and Factory Automation, ETFA'16*, pages 1–9, 2016.
- [Voi19] Voica Gavrilut and Paul Pop and Soheil Samii. Traffic Type Assignment for TSN-based Mixed-Criticality Cyber-Physical System. https://orbit.dtu.dk/files/163299831/phd500_Gavrilut_M.pdf, 2019.