



**HAL**  
open science

## Comparative study of Ethernet technologies for next-generation satellite on-board networks

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

### ► To cite this version:

Pierre-Julien Chaine, Marc Boyer, Claire Pagetti, Franck Wartel. Comparative study of Ethernet technologies for next-generation satellite on-board networks. DASC 2021, Oct 2021, San Antonio, United States. 10.1109/DASC52595.2021.9594399 . hal-03463255

**HAL Id: hal-03463255**

**<https://hal.science/hal-03463255>**

Submitted on 2 Dec 2021

**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.

# Comparative study of Ethernet technologies for next-generation satellite on-board networks

Pierre-Julien CHAINE  
*Airbus Defence and 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 and Space*  
Toulouse, FRANCE  
franck.wartel@airbus.com

**Abstract**—The spacecraft industry is facing a new challenge: new missions and customers require always more on-board performance. The current satellite network technologies will not be able to handle this increasing demand for long which leads the spacecraft industry to consider an upgrade of their satellite on-board networks. One opportunity has appeared through the use of Ethernet technologies to benefit from COTS components of a mass market. In this paper we discuss the suitability of several Ethernet technologies/standards with respect to the requirements of a satellite network in a qualitative approach. We will compare Ethernet, ARINC 664, TTEthernet and Time Sensitive Networking under three properties i.e. their quality of service capabilities, their synchronization capabilities and their fault tolerance capabilities.

**Index Terms**—Embedded Networks, Real-Time Networks, Satellite, TSN, ARINC 664, TTEthernet, Ethernet

## I. SATELLITE CONTEXT

### A. Context

In accordance with the ever-expanding volume of data generated and handled by ground-level equipments (telephone, cars, scientific instruments, etc.), satellites must be capable of producing and transmitting massive amounts of data in order to meet their users' requirements. This entails both instruments (telescopes, cameras, transponders, etc.) and the network carrying them (on-board and board to ground) must now, more than ever, improve in performance. Whereas this issue may seem clear-cut for generating data, the problem of conveying it remains complex. Thus, the scope of this paper will remain focused on embedded networks.

### B. Current Satellite On-Board Network

In a generic satellite architecture, the on-board network is typically "composed" of two networks: *platform* and *payload*. Each of these networks fulfils diverging and sometimes contrasting needs.

On the one hand, the *platform network* is in charge of conveying all the necessary information used to guarantee the nominal behaviour of the satellite. It transmits data from sensors (position, magnetic field, temperature, etc.) 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 [1] or CAN [2] 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 telescopes, telemeters from weather sensor or IoT (*Internet of Things*) data. 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 [3].

### C. Reason 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 i.e. 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 use, development and updates are quite expensive, in particular in terms of non-recurring costs. Using a technology based on COTS - *Commercial-off-the-shelves* - components, or IPs - *Semiconductor Intellectual Property Cores* (instantiated into specific space oriented hardware), shared, for some parts or as a whole, with other industrial sectors (automotive, industrial automation, aeronautics, etc.) could help lower the overall cost of the satellite network. Having a wide-spread technology could also facilitate the interaction between the spacecraft industry and the academic world. This is the reason why we focus on Ethernet-based technologies.

Moreover, adding more mechanisms at network level (ISO Level 2) could ease the integration of an increasing number of equipments on-board and reduce the development effort to be done at application level. Indeed, design would be simplified as the network would be seen as a black-box from an application point of view, with one or more Service Access Points to interact with it. The network would hence provide properties (introduced in the following sections) guaranteed to the application.

The expected properties and requirements of this future on-board network are the subject of the next section.

## II. PROBLEMATIC

### A. Identification of expected properties of the future network

In order to compare different candidates for the upgrade of the satellite network, we identify, in this paper, three "properties" that the future network should have. We remind the reader that these properties shall be provided at ISO level 2 i.e. MAC level. Later, we will analyse the capabilities of each pre-selected technologies with respect to these properties. For each property, we define criteria that will be used to determine whether a property is satisfied or not. Let us now introduce the properties and associated criteria.

#### Property 1: Mixed Quality of Service - Mixed QoS

Capability of the network system to convey, on the same equipment, several flows with different characteristics (e.g. data rate, min/max packet size, deadline, jitter).

For instance, a network satisfying Prop. 1 shall be able to convey, with the same equipments low data rate with low jitter and high data rate traffic while operating at 1Gbits/s.

##### Criteria for Property 1:

- Determinism capability, with user defined values for latency and jitter,
- Maximum data rate.

#### Property 2: Time Management

Capability of the network system to manage time, i.e. ensuring either a global common clock of all network elements or at least applicative time distribution.

##### Criteria for Property 2:

- Time synchronisation capability at MAC level,
- Time management algorithms' robustness,
- Interaction with higher layer capabilities (Service Access Points, possibility of synchronization with applications, use of time to trigger actions at application level, etc.).

Before defining the third *property*, let us first explain that we consider a faulty behaviour as either incorrect, lost, out of time constraints or out of traffic contracts.

#### Property 3: Fault Tolerant Operations

Capability of the network system to operate in a faulty context by preventing faults, by detecting, isolating and recovering from certain faults and by generating failures report/indicators for higher level fault management in case fault cannot be dealt locally, in a seamless manner.

##### Criteria for Property 3:

- Error detection capabilities,
- Error reporting capabilities,
- Redundancy capabilities,
- Fault Containment capabilities (ensured by traffic segregation).

Thanks to these three properties, we expect to answer to the following question: *Is the considered network technology capable of providing, at the same time, Mixed QoS, Time Management and Fault Tolerant Operations capabilities for the upgrade of a satellite network ?*

The real challenge is to find a technology that is capable of satisfying all the properties at the same time, forming a

*unified network*. This is usually not the case in the current implementations in space where one bus is used for hard real time, highly critical traffic (usually MIL-STD-1553 or CAN) and another bus is used for payload traffic (usually SpaceWire).

### B. Motivating Example

Let us consider a motivating example that we will use as support for the overview of the technologies' capabilities.

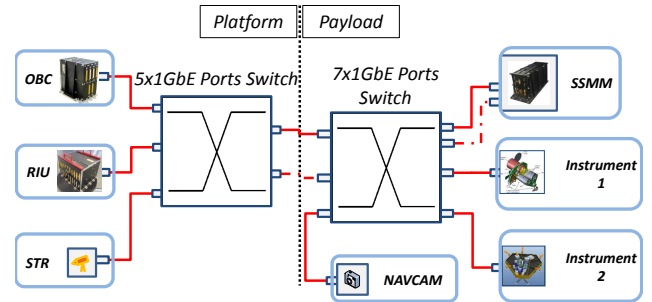


Fig. 1. Motivating network example

This motivating example comes from the use case consolidated after the requirement analysis in [4]. The network system is smaller (fewer end-points) but we consider it to be representative enough to illustrate this work.

The network of Fig. 1 has 7 end-points (ES), 2  $10^*1Gbits/s$  switches (SW), all the devices 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. The main device in the network is OBC - *On-Board Computer*. It has several applications running concurrently. They send periodic messages to the each end-point of the *platform*. All the communication exchanges in the platform are *transaction-like* i.e. one application sends a request to one device and the device immediately answers to this request. On the *Payload* side, instruments send messages to the SSMM (*Solid State Mass Memory*). Finally, the NAVCAM (*Navigation Camera*) sends periodic messages to the OBC. Both instruments and NAVCAM do not use transactions to communicate, messages are sent when needed. We characterize the previously introduced exchanges with flows in Table I. This set of flows, denoted  $\mathbb{F}$ , is just an extract of all the flows of the complete use case but is still representative of the performance requirements of this network. We let the reader refer to [4] for a more

TABLE I  
TRAFFIC CHARACTERISTICS AND REQUIREMENTS

Name	Type	Data Rate	Max. Latency	Max. Jitter
g	OBC to Platform	4bits/s	1ms	1 $\mu$ s
h	OBC to Platform	4kbits/s	1ms	10 $\mu$ s
i	Platform to OBC	4kbits/s	1ms	NA
j	Instr. to SSMM	65Mbits/s	NA	NA
k	NAVCAM to OBC	250Mbits/s	NA	NA

complete use case description.

Applying the three properties to our use case, we expect that:

- In this network,  $g$ ,  $h$ ,  $i$ ,  $j$  and  $k$  shall meet their latency and jitter constraints (Property 1).
- There shall be a global common clock (Property 2).
- All frames shall respect their traffic contracts, arrive in their time constraints, no frames shall be lost and frame integrity shall be guaranteed (Property 3).

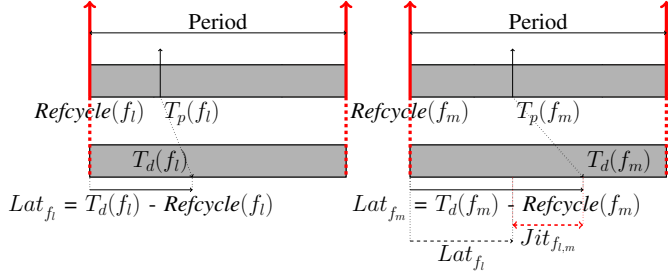


Fig. 2. Latency and Jitter concept in our use case

Before analysing the technologies capabilities, let us define precisely the jitter and latency concept of our use case.

**Definition 1 (Production/Delivery Dates):** Let  $f_l$  the  $l$ -th message of flow  $f$ , let  $T_p(f_l)$  (resp.  $T_d(f_l)$ ) denote the production date (resp. delivery date) of  $f_l$  at application level (ISO L7).

**Definition 2 (Emission/Reception Dates):** Let  $T_e(f_l)$  (resp.  $T_r(f_l)$ ) denote the emission date (resp. reception date) of  $f_l$  at physical level (ISO L1).

**Definition 3 (Deposit Date):** Let  $T_{SAP}(f_l)$  denote the deposit date of  $f_l$  i.e. the date at which the application deposits  $f_l$  in the L2 service access point of the appropriate output port of the emitting end station.

Immediately,  $T_p(f_l) < T_{SAP}(f_l) < T_e(f_l) < T_r(f_l) < T_d(f_l)$ .

**Definition 4 (Reference Date):** For each frame of a flow, we define a production window which lower bound is entitled *reference date* denoted  $Refcycle(f_l)$ . Therefore,  $\forall f \in \mathbb{F}, Refcycle(f_l) \leq T_p(f_l)$ .

**Definition 5 (Latency):** The latency of a message is defined as the age of that message compared to its reference date. Thus,  $\forall f \in \mathbb{F}, \forall l \in \mathbb{N}, Lat_{f_l} = |T_d(f_l) - Refcycle(f_l)|$ . The overall latency of a flow is defined by  $\forall f \in \mathbb{F}, Lat_f = \max_l Lat_{f_l}$ .

This definition of latency differs from classical network latency, which is defined as the duration between the emission and reception dates of  $f_l$  i.e.  $T_d(f_l) - T_p(f_l)$ . In particular, our definition do not depend on the production and emission dates of messages but only on their delivery dates.

**Definition 6 (Jitter):** In our setting, the *jitter* is understood as the latency variability. It is sometimes referred as *reception jitter*. It is first defined for two messages as:  $\forall f \in \mathbb{F}, \forall l, m \in \mathbb{N}, Jit_{f_l, m} = |Lat_{f_l} - Lat_{f_m}|$ . The overall jitter of a flow is defined by  $\forall f \in \mathbb{F}, Jit_f = \max_{l, m} Jit_{f_l, m}$ .

We illustrate latency and jitter definitions in Fig. 2. In-depth description of these concepts is available in [4].

### C. Technologies of interest

Several technologies were previously identified, in internal projects, as candidates for an upgrade of the satellite on-board network, namely Ethernet, ARINC 664, TTEthernet, SpaceFibre and Time Sensitive Networking. They were mainly selected because, at first glance, they could provide a high throughput network with QoS mechanisms available. Other technologies, such as 1553, CAN or SpaceWire are *de facto* discarded from this study since their maximum throughput is too low (less than 1Gbits/s). SpaceFibre [5] is considered out of scope of this paper being not based on Ethernet, its suitability will be addressed in future work. Before discussing their capabilities w.r.t. the satellite network requirements, let us briefly introduce them.

- **Ethernet:** *Full Duplex Switched Ethernet* or *Ethernet* is an ISO Level 2 (Data Link) technology based mainly on IEEE 802.3 [6] and 802.1Q [7]. In this paper, we name *Ethernet* the technology defined in 802.1Q-2008<sup>1</sup>. The network is composed of switches and end-stations that exchange Ethernet frames (format defined in [6]). Ethernet is spread worldwide as it is the standard networking technology used at home and in ISP core networks.
- **ARINC 664:** *ARINC 664* [8] defines an avionic bus with a "deterministic" Ethernet protocol. It is, in particular, used at Boeing, and at Airbus under the name AFDX (Avionics Full Duplex Switched Ethernet). It extends Ethernet(802.1Q-2008) with determinism and fault tolerant capabilities.
- **TT-Ethernet:** *TT-Ethernet* is a 100Mbps/1Gbps scalable networking technology designed for industrial automation and aerospace applications, standardized by SAE under the reference AS6802 [9]. TTEthernet extends the ARINC 664 standard. It supports mixed quality of service with both synchronous (time-triggered) and asynchronous communications schemes, with the help of a fault-tolerant synchronization strategy.
- **Time Sensitive Networking:** *Time Sensitive Networking* or *TSN* [10] is a technology based on Ethernet developed and promoted by the IEEE TSN Working Group (former AVB - Audio Video Bridging - Working Group) since 2012. It aims at providing many QoS capabilities for conveying traffic of different criticality. It has received attention in various industry verticals such as Automotive, Industrial Automation, 5G and Aerospace.

### D. Related Works

The question of choosing a unified Ethernet based network is a challenge on which we have been working for some time. We have introduced in [11] the need for an upgrade of the satellite network and identified several challenges on the use of TSN for space applications. In [4], we formally explained some generic network performance requirements and instantiate them on a simple satellite use case. The goal

<sup>1</sup>2014 and 2018 versions are considered beta versions of TSN

of this current paper is to discuss which technologies are good candidates for next-generation satellite on-board networks.

The closest work is the paper [12] which proposed in 2016 a first analysis, comparing ARINC 664, TTEthernet and Ethernet for Space, a tailoring of ARINC 664 for space applications. We add more technologies to this comparison, namely Ethernet and TSN, and provide a more in-depth analysis of properties for each technology. [13] introduces several requirements oriented towards space robotics needs and provides a high level analysis of the capabilities of SpaceWire/SpaceFibre w.r.t. to said requirements. The authors conclude that SpaceWire/SpaceFibre could be a potential candidate for the upgrade of the satellite network. Nevertheless, SpaceFibre is out of scope of this paper.

Outside of the space domain, the race for evolving from a real-time bus towards a faster, better and more common technology is also raging, in particular in the automotive and factory automation domain, but also in the avionic domain. Most of the work presented hereafter uses simulation to compute temporal attributes but in the following papers, the definition of latency and jitter is not identical to ours. We introduce them as a first step towards a potential future work on TSN performance evaluation via simulation.

[14] proposed a similar approach to ours, applied to the automotive industry. Their analysis targeted the evaluation of the use of AFDX, TTEthernet, EtherCAT and AVB in an automotive context. They consider Property 1 and 3 but also address physical layer, system start-up and costs aspects. We consider these aspects out of scope of our comparison. We complete this work with Property 2 and further analysis for Property 3 on a space use case.

TSN, the successor of AVB, can reproduce, via configuration, the behaviour of AVB. Hence, the results observed in AVB are transposable to TSN when configured in AVB mode. [15] proposed a comparison between AFDX (ARINC 664) and AVB in an avionic context. The authors then, based on network calculus, declare the latency determinism of AVB as suitable for aircraft avionics' network requirements. Their paper also states the new mass market controllers and switches developed for AVB (now TSN) could be a cost-effective alternative for COTS-based low criticality systems. We share their view on the performance and cost analysis. However, we did not discuss AVB in this comparison since, for our specific application, it will not match our needs in terms of jitter. [16] proposed a competitive performance evaluation of AVB and TTEthernet. The evaluation is performed on an automotive use case via simulation in OMNET++. Results of the evaluation show comparable performances for both technologies in terms of latencies. However, the authors identified that background traffic had a significant impact in AVB and planned to make more evaluation while adding background traffic to quantify its impact on performances. [17] also do simulative assessment of the performances of AVB and TTEthernet in an automotive context that leads them to the same conclusions than [16]. [18] compares the performances of AVB and Ethernet in an automotive use case at 100Mbit/s. In our analysis, we compare

the performances of TSN and Ethernet at 1Gbits/s.

[19] proposed a comparison between TTEthernet and TSN. The comparison is very thorough and technical. Several properties, similar to ours are used for the comparison i.e. *synchronization, bandwidth allocation, traffic shaping and traffic scheduling and redundancy*. The authors compute worst case delays in a specific TSN configuration based on network calculus. We redo the qualitative protocol comparison with our specific definition of latency and our use case and add a more detailed redundancy analysis. Plus, as mentioned before, we cannot exploit their worst case delays results as our definitions of latency and jitter differ (see II-B).

### III. TECHNOLOGIES OF INTEREST CAPABILITIES W.R.T. SPACE REQUIREMENTS

#### A. Ethernet

Let us now discuss the compatibility of Ethernet with the three properties of section II-A.

1) *Property 1*: Ethernet, with respect to Quality of Service, is fairly simple. Only one mechanism is available: Static Priority. Static Priority relies on the 802.1Q optional tag of the Ethernet frame. This tag contains a 3 bit field called *Priority*, that can be used to define, at MAC level, a priority between frames on 8 ( $2^3$ ) levels. This allows to define a priority for messages or flows and decide, in case of medium access conflict, which *Ethernet Frame* shall be emitted first. Static priority does not provide any guarantee on delays as far as there are no traffic contracts in place. Since Ethernet does not provide this kind of capabilities, traffic contracts are left to be dealt with at application level.

Let us roughly analyse the different parts that affect latency.

$$Lat_{f_l} = AppEmOffset + \sum_{link \in Path(f)} \Delta_{link}$$

$$Where \Delta_{link} = HPB_{link} + SPB_{link} + LPB_{link} + \tau_{emit} + \tau_{propag} \quad (1)$$

*AppEmOffset* represents the duration between the reference date of  $f_l$  and its deposit in the queue of the emitting end-system i.e.  $T_{SAP}(f_l) - Refcycle(f_l)$ .  $HPB_{link}$ ,  $SPB_{link}$  and  $LPB_{link}$  represent the delays induced per hop, in the path of  $f_l$ , by higher, same and lower priority traffic.  $\tau_{emit}$  and  $\tau_{propag}$  represent the duration of emission and propagation of  $f_l$  on a link.

We distinguish two situations:

- 1) The application has a scheduler that prevent any contention for medium access,
- 2) The application does not have a scheduler.

With the scheduler, there is only one frame accessing the medium at a time. For *OBC* to *Platform* traffic, there is no congestion on links since the links are only used for *Command and Control (C&C)* traffic and frames are sent one at a time in our use case. In this situation, we can consider that, on each hop of the frame on its path  $HPB_{link} = SPB_{link} = LPB_{link} = 0$  and  $\tau_{propag} = Constant, \tau_{emit} = Constant$ . This means that latency, for Ethernet Static Priority with applicative scheduler,

is reduced to  $AppEmOffset$  (variable) and  $\tau_{propag} + \tau_{emit}$  (constant). Hence ensuring a low jitter for a *OBC* to *Platform* flow consists in ensuring a low jitter emission of the frame at applicative level. In our industrial context, implementing a low jitter emission scheme at application level is not targeted.

Without scheduler, equation  $HPB_{link} = SPB_{link} = LPB_{link} = 0$  is not valid anymore. Therefore there are even more parameters to take into account for a low jitter flow since the lower priority blocking adds up to an additional 12,304 microseconds (see Table II) to the frame's latency per hop. In this situation, the jitter constraint for flows  $g$  and  $h$  is not satisfied.

TABLE II  
USUAL FRAME EMISSION DELAY AT 100MBITS/S AND 1GBITS/S

	100Mbits/s	1Gbits/s
Max. size frame (1518 + 20 bytes) <sup>2</sup>	123,04 $\mu$ s	12,304 $\mu$ s
Min. size frame (64 + 20 bytes)	6,72 $\mu$ s	0,672 $\mu$ s

To conclude on Property 1, Ethernet is not deemed suitable.

2) *Property 2*: At ISO level 2, Ethernet alone does not provide any mechanisms for time management. However, the use of higher level protocols over Ethernet for time distribution and/or time synchronization is very common. These protocols often require a lower level layer support, at either MAC or PHY level. Hardware supporting these protocols shall be able to timestamp frames in emission and reception and retrieve these timing informations at higher layer for these protocols to use. The most mainstream synchronization protocol over Ethernet (but not at level 2) is PTP - *Precision Time Protocol* [20].

PTP (i.e. IEEE 1588) is quite a simple protocol, one time master device (entitled *grandmaster*) distributes time with broadcast messages. Devices in the system measure propagation delays with their peers or with the time master and use this information to correct the time received from the master. If several potential grandmasters exist in the network, IEEE 1588 introduces the *Best Master Clock Algorithm - BMCA*, which chooses which device among potential masters is going to be grandmaster. This algorithm allows for faster recovery than PTP since the BMCA is run permanently and triggered if the current grandmaster is not functioning anymore in which case a new master can hence be automatically elected.

PTP traffic is travelling in the same network than user data. As per recommendation of the standard, synchronization traffic is not best effort and shall be handled consequently. In Ethernet, this means that synchronization traffic shall be given a relatively high priority. Further analysis will be required to see if synchronization traffic could share the same priority and resources than the OBC to Platform traffic i.e. if both flows do not impact each other behaviour/performances and conclude on the priority mapping for PTP traffic. Plus, a network with this implementation might not be very flexible and it may require analysis to be re-run if the user traffic is changed and verify whether the synchronization quality is still satisfactory

<sup>2</sup>The 20 bytes correspond to the sizes of Preamble, Frame Delimiter and Inter-Frame Gap, commonly denoted SFD + IFG

and synchronization traffic does not impact the performance of other traffic.

Anyway, PTP is not a level 2 service and no interface are specified for interaction with higher level, although it may have been implemented by some manufacturers in, for instance, Linux PTP project (to benefit from Linux APIs).

We conclude that Ethernet+PTP, apart from manufacturer specific implementations, might not satisfy Property 2.

3) *Property 3*: Regarding the third property, Ethernet only offers error detection. At ISO level 2, there are absolutely no mechanisms for redundancy. For error detection, the only available tool is the Ethernet frame CRC i.e. the *FCS - Frame Checking Sequence*. This 16 bits field allows checking the integrity of the content of the Ethernet frame. A CRC error usually leads to the erroneous frame being dropped, which coincides with the fault isolation criteria. This CRC will help detecting and preventing the incorrect faulty behaviour. Ethernet, however, is widespread and hence is used as MAC layer for several higher layer protocols like, for instance, UDP/IP. Although Ethernet does not provide any redundancy mechanisms or more error handling mechanisms, they could be provided with these higher level protocols. However, this option is out of scope of this study.

In addition to the Ethernet frame CRC, Ethernet devices usually hold counters, called MIB - *Management Information Base* counters, that describe the behaviour of the device such as number of received frames, number of emitted frames, number of CRC errors, etc. These MIBs are, for most parts, standardized by IEEE (e.g. MIB section in IEEE 802.3 [6] and IEEE 802.1Q [7]) or IETF (e.g. RFC 3635). The information included in these MIBs, updated in every device, could be gathered by a higher layer entity (with the help of SNMP protocol for instance) and serve as error detection mechanism. MIBs will help detect incorrect, lost and out of traffic contracts faulty behaviours.

Although Ethernet has some error detection mechanisms, there is no real mean to prevent faults, in particular, lost faulty behaviour (like single points of failure). Moreover, there are no mechanisms available to detect and or prevent the out of time constraints faulty behaviour.

Regarding fault containment capabilities, Ethernet provides weak traffic segregation through Static Priority. In fact, if a device sends too many messages (ex: babbling idiot) which corresponds to not in traffic contracts faulty behaviour, it will definitely affect the available bandwidth of the other flows and might lead to buffer overflow. Moreover, this faulty behaviour will propagate downstream.

According to the capabilities introduced above, Ethernet technology is not effective enough w.r.t Property 3.

## B. ARINC 664

Let us now do the same work for ARINC 664.

1) *Property 1*: Ethernet did not provide any QoS guarantee by itself. ARINC 664 extends Ethernet with determinism. In fact, in addition to Static Priority (reduced to only two level and only present in switches), the concept of VL - *Virtual Link*

is introduced in ARINC 664. It is indeed a reserved bandwidth for a specific traffic on a static route. It is characterized by two parameters, the maximum frame size and the so-called BAG - *Bandwidth Allocation Gap* - i.e. the minimum time between two frames' emission in the same VL. This VL concept allows guaranteeing interesting temporal properties i.e. bounded latency. Latencies can be determined with more or less pessimism through, for instance, network calculus. Typical latencies are in the range of 1 to 10 milliseconds [21].

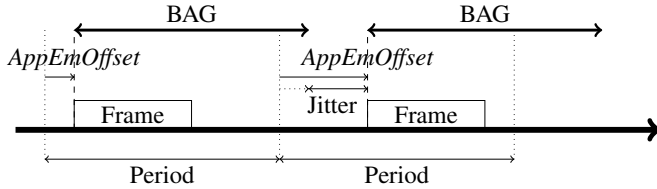


Fig. 3. BAG concept in AFDX and *AppEmOffset*

In ARINC 664 there are only two levels of priority in switches and no priority in end-points, but everything that was explained for Ethernet (and 8 levels of priority) remains true for ARINC 664. Latency is now guaranteed by the network since there are traffic contracts in place (i.e. VLs). But ARINC 664 traffic shaping is asynchronous, hence the low jitter problem present in Ethernet is the same in ARINC 664. Even if a flow respects its BAG constraint, it can have a jitter at emission, as shown in Fig. 3. Therefore, the low jitter on reception problem remains a low jitter on emission problem with ARINC 664.

The video traffic coming from the navigation camera (flow  $k$ ) could either be fitted in a specific VL or be standard Ethernet traffic [22].

So far, ARINC 664 does not seem suitable with respect to Property 1.

2) *Property 2*: Regarding synchronization, ARINC 664 is identical to Ethernet and our conclusion is almost the same. The synchronization traffic will be given a specific VL. It will hence be provided with a guaranteed reserved bandwidth and ARINC 664 fault tolerant operation mechanisms will prevent any impact from synchronization traffic to user traffic and respectively.

3) *Property 3*: ARINC 664 offers another major improvement from Ethernet: it enhances Ethernet with fault tolerance and traffic policing capabilities.

In addition to CRC and MIB counters, ARINC 664 uses an end-to-end redundancy protocol at level 2. This redundancy protocol will help preventing the lost faulty behaviour. It can be seamless (configurable), meaning that the application level is not aware that a redundancy protocol was running at level 2 and does not need this information to work nominally. Let us rapidly describe this redundancy protocol and what it offers. Redundancy in ARINC 664 allows the duplication of messages of a flow at the emitting end-point and their reassembly (elimination of the duplicate) at the receiving end-point. Duplicates travel on the same path but on two different channels i.e. two different cables. Duplicates are

identified by a two bytes *sequence number* appended to the Ethernet frame right before the frame's FCS. The sequence number is used for two functions i.e. *Integrity Checking* and *Redundancy Management*. Integrity Checking is a fault detection and isolation mechanism that helps to detect and eliminate faulty duplicates (incorrect faulty behaviour) as well as lost messages (lost faulty behaviour) whereas Redundancy Management eliminates duplicated messages once the nominal message has been received (lost faulty behaviour).

ARINC 664 switches have a traffic policing feature. It allows the switch to verify that every VL respects its traffic contract. If not, flows that exceeds their contract are dropped. This will help preventing the out of traffic contract faulty behaviour, but also fault containment. In fact, if a device is emitting more than it should or emitting with incorrect addressing (wrong destination address), its traffic will immediately be eliminated at the next hop and will not impact the nominal behaviour of other VLs. These new mechanisms introduced in ARINC 664 are also represented in the MIB counters. Although traffic policing offers a good handling of out of traffic contracts faulty behaviour, it does not detect or prevent the out of time constraints faulty behaviour.

Regarding fault containment, VL and traffic policing offer a traffic segregation of all the different flows. Any fault occurring in one VL will not affect the nominal behaviour of the other VLs.

According to the previous mechanisms, ARINC 664 is deemed unsuitable w.r.t. to Property 3.

### C. TTEthernet

Let us now shift our focus on TTEthernet.

1) *Property 1*: TTEthernet extends ARINC 664 even further. In addition to BE -*Best Effort* (standard Ethernet) and RC-*Rate Constrained* (ARINC 664) traffic, TTEthernet introduces a third traffic type entitled TT - *Time Triggered*. Time Triggered traffic is sent in a time triggered manner. Each TTEthernet device (end-point) has a transmit schedule per flow. This schedule allows flows to achieve constant communication latency and low jitter. In fact, this new traffic solves the jitter issue that Ethernet and ARINC 664 faced. Since TT traffic achieves constant communication latency and the schedule (especially in switches) ensures there's no blocking from other frames, equation (1) leads to the same conclusion than Ethernet i.e. the low jitter on reception problem roots to a low jitter on emission problem. However, thanks to our definition of latency, the per-flow schedule solves the problem. We illustrate how time-triggered schedule cancels application emission jitter in Fig. 4. In fact, even if the production date of frames changes between periods in the figure, the emission date remains constant thanks to TTE schedule.

Since the schedule gives a fixed emission date on the medium, whatever the jitter occurring at application level (under conditions of course, like message being emitted before its schedule), it will be masked by the constant offset of the flow's schedule, resulting in a low jitter message emission. The only elements taking part into jitter in this situation are

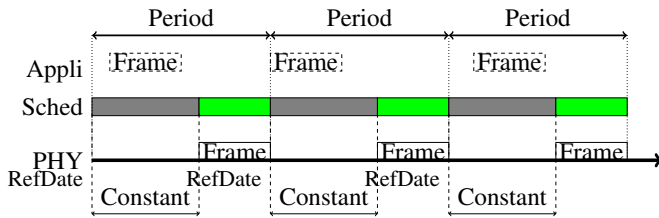


Fig. 4. TT Schedules cancels application emission jitter

clock precision and TTEthernet constraint stating that in any schedule, there shall be space to fit a PCF - *Protocol Control Frame* - frame. The size of this PCF is 84 bytes (including SFD and IFG), which means that it can lead to a jitter of at most 672 nanoseconds (cf. Fig. II). This jitter is compatible with the jitter constraint of flows  $g$  and  $h$ .

According to our analysis, TTEthernet appears to be a relevant candidate with respect to Property 1.

2) *Property 2*: TTEthernet proposes a new protocol to establish and maintain a global time throughout the network. It is realized by synchronization of local clocks within the network nodes (end-point and switches). It works in a similar manner than IEEE 1588 (used in Ethernet cf. III-A2). However, it introduces a new mechanism targeted for fault tolerance i.e. clock redundancy. In fact, in TTEthernet, there can be several synchronization masters (instead of one in PTP). These *masters* send their time through PCF frames in the network. Specific devices entitled compression masters gather PCFs coming from masters and make a correction that is then distributed through a new PCF towards synchronization clients (devices that synchronize with compression masters). The PCF is also sent back to synchronization masters for fault detection.

Regarding network resources, every slot (for the TT traffic) is configured so that both the frame of this slot and a PCF frame can be emitted. Hence, there is a resource reservation of 84 bytes (frame size + SFD + IFG) per slot for time synchronization.

Synchronization is critical in TTEthernet. Without synchronization, the proper temporal property obtained for TT flows (in time slots), i.e. very low jitter and constant network latency cannot be guaranteed anymore.

To conclude, we consider TTEthernet relevant w.r.t. Property 2.

3) *Property 3*: Regarding fault tolerance, TTEthernet provides both redundancy and policing capabilities inherited from ARINC 664 (cf. Annex D of SAE AS6802 [9]). TTEthernet network takes back the redundancy management and the integrity checking of ARINC 664 for TT and RC traffic. TTEthernet switches can also implement a *central bus guardian* function. It basically provides a policing mechanism to check that traffic contracts (of RC traffic) and schedule (of TT traffic) are respected. This function serves the identification, elimination and containment of the out of traffic contracts and out of time constraints faulty behaviours. In fact, it can for instance prevent the propagation of faults coming from a babbling idiot device in the network.

In addition to ARINC 664 capabilities regarding fault tolerant operations, TTEthernet offers the possibility for synchronization mechanism to work with multiple masters. In this situation, a single master failure will not affect synchronization and hence not affect the TT flows temporal behaviours. For as much, in case of multiple active synchronization masters, there will not be any fail-over time in case of grandmaster failure. This last point is a major improvement compared to IEEE 1588.

We conclude that TTEthernet is suitable w.r.t. Property 3.

#### D. Time Sensitive Networking

Finally, let us introduce Time Sensitive Networking capabilities.

1) *Property 1*: Time Sensitive Networking (or TSN) extends Ethernet with powerful traffic shaping capabilities and medium access enhancement.

In fact, TSN introduces a new mechanism entitled *Frame Preemption*, introduced in [23]. Frame Preemption helps, depending on configuration, to solve the Static Priority jitter introduced by lower priority blocking. In fact 802.1Qbu and 802.3br allow purposely tagged frames (*express*) to suspend the transmission of other frames (*preemptable*) for their own transmission on a point-to-point link, defining a frame fragmentation similar to IP fragmentation. Using Frame Preemption, the lower priority blocking jitter is reduced to the necessary time to transmit, in the worst case, a 143 bytes long frame (see [24]). At 1Gbit/s, this will lead to a  $1,144\mu\text{s}$  jitter.

Then, TSN offers the opportunity, for every TSN device, to have up to 8 traffic shapers associated to up to eight transmission queues per port. The most notable shapers are *Time Aware Shaper - TAS*, *Credit Based Shaper - CBS* and *Enhanced Transmission Selection - ETS*. Other shapers, like *Asynchronous Traffic Shaper - ATS* are still being standardized (and will not be introduced in this paper). Each of these shapers have rules that define whether a frame in its queue can be available for transmission. Several frames available for transmission in several queues are then emitted in priority order, using static priority.

Time Aware Shaper allows defining time windows in which frames can be emitted, almost in a similar manner than TTEthernet. However, contrary to TTEthernet, the time schedule is not applied to flows (named *Streams* in TSN) but to emission queues. This means that the good temporal properties obtained in TTEthernet are not immediately achievable here. In particular, in the network, there may be more than 8 streams, meaning that several streams would have to share the same queue hence risking additional communication delay from non-exclusive resources.

In addition, TSN inherits mechanisms introduced in AVB. One of them is *Credit Based Shaper (CBS)*. CBS was introduced in AVB to contain the starvation problem of Static Priority. It defines a rule to allocate a bandwidth to a queue (ex: 4Mbits/s for queue 1) based on a credit that evolves when frames are enqueued or dequeued. In fact, in Static Priority, if the high priority queues have a lot of messages to emit,



the lower priority queues will not get a chance to emit their frames. With CBS, each queue can emit up to their allocated bandwidth.

Finally, Enhanced Transmission Selection defines rules for sharing the available bandwidth between queues of an output port. One algorithm for ETS can be Round Robin or Weighted Round Robin.

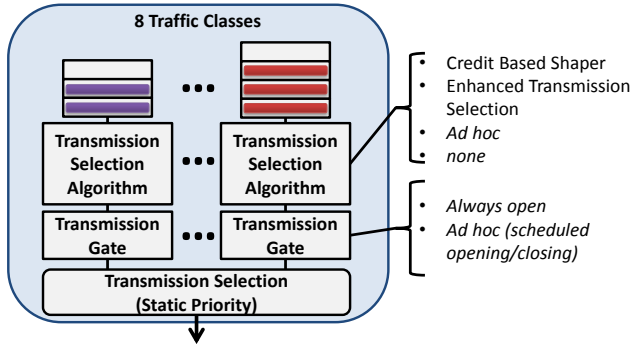


Fig. 5. TSN Traffic Shaping Capabilities

The traffic shaping capabilities of TSN are huge thanks to the high number of mechanisms available. It is also quite flexible, many configurations can exist by choosing which mechanisms are activated or not. Some notable configurations are full TT (only time aware shaper), full SP (only static priority), ABV (a mix of CBS and Static Priority) and TT-CBS-BE (mix of time aware shaper, CBS and static priority). Fig. 5 summarizes an output port traffic shaping capabilities. Selecting one mechanism for each box in the figure among the proposed mechanisms listed on the right of the figure will lead to different configurations. Nevertheless, this flexibility comes with a price: configuring such a network to ensure latency and jitter requirements might not be very easy. One solution to do so would be to reproduce TTEthernet with TSN mechanism Time Aware Shaper. This work is presented in [25], [26] and [27]. It helps achieving a per-flow schedule in the TSN network. This solution however might not be very scalable but still, considering the size of the satellite network it might not be an issue. Nevertheless, in general, some analysis, with for instance network calculus ([28], [29], [30]) will be required to find good configurations that meet the network requirements.

We hence declare TSN suitable w.r.t. Property 1.

2) *Property 2*: Inherited from Ethernet, Time Sensitive Networking supports several synchronization protocols. In particular, it supports IEEE 1588 (PTP, introduced in III-A2) but also IEEE 802.1AS [31] also called gPTP. Like for Ethernet and ARINC 664, 802.1AS is not entirely a MAC level protocol.

gPTP is, for most part, fairly similar or identical to PTP. It contains in fact a profile of 1588 for use in TSN networks. We will not explain how the synchronization is established. Instead, let us focus on the two main events with gPTP : the possibility for a network to have several synchronization masters (called *grandmasters*) and the specification of interfaces and primitives for "time sensitive applications".

Like TTEthernet synchronization, TSN synchronization protocol offers the possibility to have several masters in the network for availability purposes. However, it does not work quite like TTEthernet. In fact, 802.1AS standard does not provide any consolidation strategy for several grandmasters running concurrently (realized with compression masters in TTEthernet), it is left to be developed at application level.

gPTP does not provide any fault tolerance mechanisms apart from BMCA (cf. §7.2.4.3 of [31]) which covers a grandmaster not working.

gPTP also introduces application interfaces (cf. Clause 9 of [31]) for use with time sensitive applications. These interfaces are "model of behavior and not application program interfaces". Five interfaces are described in the standard but it is stated that others can exist. Hopefully, this will help facilitate the port of *time sensitive applications* on different TSN products from potentially different manufacturers.

In gPTP, like in PTP and in TTEthernet, the synchronization traffic travels in the same network than user data. The problems identified in Ethernet for PTP in terms of resources reservation or resources sharing are identical with 802.1AS. However, [32] shows that the synchronization process based on 802.1AS is not affected by high network load.

To conclude, we consider TSN relevant w.r.t. Property 2.

3) *Property 3*: Time Sensitive Networking offers a wide range of mechanisms for Fault Tolerant Operations. Since it extends Ethernet, all elements introduced in Ethernet (CRC and MIBs) are still true for TSN.

Redundancy in TSN is fairly similar than in ARINC 664. Frames are duplicated and then reassembled using a sequence number. In TSN this sequence number is located in an Ethernet optional field (or tag). Several protocols are available for redundancy such as RTAG, PRP or HSR tags. TSN can deliver frames out of order whereas ARINC 664 provides in order delivery guarantee. However, TSN improves ARINC 664 redundancy by adding the opportunity to have more than two duplicates. It also offers the ability to specify the path of every duplicated flow, meaning that nominal and redundant flows does not have to travel on the same path using to different channels. In addition, in TSN it is possible to do redundancy on only a fragment of the path of the flow instead of only end-to-end. As for ARINC 664 and TTEthernet, redundancy in TSN will help detecting and prevent the incorrect and lost fault behaviour.

In terms of traffic policing, IEEE 802.1Qci - *Per Stream Filtering and Policing* [33] offers multiple mechanisms. One serves at detecting any temporal error in the reception of frames, this is particularly useful when TSN is configured with time triggered traffic. Another one serves at ensuring, per flow or per queue, the compliance of the traffic with the traffic contracts in place. Indeed, *Per Stream Filtering and Policing* will help detect and prevent the out of contract faulty behaviours.

Regarding segregation capabilities in TSN, for traffic shaping, it behaves like Ethernet i.e. the segregation unit is a queue (through Static Priority). However, for traffic policing, the

segregation unit is not a queue but a flow since the traffic policing functions are applied per stream (in *Per Stream Filtering and Policing*). This difference of granularity between traffic shaping and traffic policing might increase the complexity of the segregation analysis for fault containment purposes.

In consideration of the above analysis, we declare TSN suitable w.r.t. Property 3.

#### IV. ANALYSIS

Now that each technology have been properly presented, let us compare the suitability of each technologies w.r.t. to Property 1, 2 and 3.

##### A. Property 1 - Mixed QoS

With respect to the previous presentation of the technologies and Table III, Ethernet and ARINC 664 can immediately be eliminated from the list of candidates since both do not allow for low jitter transmission of messages without strong application support.

TTEthernet seems fitting for this kind of transmission. TSN however is still a little bit uncertain. In fact, it looks like low jitter communication would be possible without application support but configuration doing so still need to be consolidated.

However, for TSN and TTEthernet, it is important to underline

TABLE III  
COMPLIANCE TO PROPERTY 1 - MIXED QoS

Criteria	Ethernet	ARINC 664	TTEthernet	TSN
Data Rate	✓	✓	✓	✓
Latency	✗	✓	✓	✓
Jitter	✗	✗	✓	✓
Suitability	✗	✗	✓	✓

that the very low jitter communication can only be guaranteed if network synchronization is working correctly. The time slotting mechanisms of TSN and TTEthernet are strongly relying on synchronization. Since current satellite networks technologies do not rely on a synchronization protocol to work nominally, the use of TTEthernet and/or TSN may increase the complexity of the validation phase of the network. In particular, the fault tolerance aspect of the synchronization itself will have to be analysed thoroughly.

##### B. Property 2 - Time Management

In terms of synchronization support, all four technologies are fairly similar (cf. Table IV). It is not fully satisfactory w.r.t. Property 2. In fact, all synchronization protocols introduced in this paper require application support to work correctly. For instance, in PTP, the protocol specifies the messages exchanges at ISO level 2 and how they are exchanged but does not specify how time information obtained from PTP frames are used to correct time. This correction will be done at application level. Again, TTEthernet and Time Sensitive Networking will be preferred to Ethernet and ARINC 664 thanks to the fault tolerance capabilities or predispositions promoted in the standard.

TABLE IV  
COMPLIANCE TO PROPERTY 2 - TIME MANAGEMENT

Criteria	Ethernet	ARINC 664	TTEthernet	TSN
At Layer 2	✗	✗	✓	✓
Robustness	✗	✗	✓	✓
Higher Layer Interaction	✗	✗	✗	✓
Suitability	✗	✗	✓	✓

##### C. Property 3 - Fault Tolerant Operations

Finally, regarding fault tolerant operations, apart from Ethernet and ARINC 664, all technologies seem to match with Property 3 (cf. Table V). The redundancy, fault identification, and fault containment they provide are almost identical and seamless. TSN could be preferred for the flexibility it offers over the redundancy mechanisms (number of duplicates, not mandatory to be end-to-end) but the lack of in order delivery guarantee (in a general manner) will force to have some potentially unwanted higher layer support.

TABLE V  
COMPLIANCE TO PROPERTY 3 - FAULT TOLERANT OPERATIONS

Criteria	Ethernet	ARINC 664	TTEthernet	TSN
Error Detection	✗	✗	✓	✓
Error Reporting	✓	✓	✓	✓
Redundancy	✗	✓	✓	✓
Fault Containment	✗	✗	✓	✓
Suitability	✗	✗	✓	✓

##### D. Third-party arguments for the selection of an upgrade candidate

According to the previous analysis, it seems like Time Sensitive Networking and TTEthernet would both be good candidates for a future unified satellite on-board networks. However, apart from performance, several arguments shall also be taken into consideration between choosing between one or the other.

First, satellite components have stringent hardware/software constraints, not in certification like in aerospace, but more in radiation, temperature, SEU -*Single Event Upset*- tolerance, etc. This means that the satellite network manufacturer has to either buy end-points and switches designed for space or buy IPs that would be instantiated into space-hardened components. On the one hand, TTEthernet, through TTTech, has already been implemented into several space projects both in Europe and in the USA and is even being standardized for space use by ESA in an ECSS standard (*European Cooperation for Space Standardization*). It would hence be possible to obtain space-oriented TTEthernet components. On the other hand, TSN, for the past years, has gained increasing interest from the automotive industry and automation industry. The TSN devices that would be available on the market would not completely fulfil the space requirements, especially in term of radiation tolerance. It would however be possible either to buy IPs and instantiate them into space-hardened components or buy the entire COTS and do a radiation tolerance evaluation.

Then, the space community is hoping that the use of COTS components from a widespread technology, shared with other industry verticals would help reducing the overall cost of design, purchase of devices and software development. One drawback of using TTEthernet instead of TSN would be that it is only produced and maintained by very limited manufacturers whereas TSN has already dozens of manufacturers working on it. Nevertheless, the products currently advertised by TSN automotive manufacturers might not exactly fit the space needs in terms of performance or environment tolerance and might require further work before being used in space systems; which in the end would lead to an increase of costs. However, the impact on non-recurring cost would be significant enough to make the use of TSN worth. That is why the definition of a profile (like the TSN Automotive Profile but for space) would be a very good starting point to give space and aerospace an identity towards TSN components manufacturers.

## V. CONCLUSION AND FUTURE WORK

The choice of a future satellite on-board network is clearly not made yet. TTEthernet and Time Sensitive Networking are two very good candidates for it in terms of pure network performance requirements.

This paper purposely limited the candidates to Ethernet technologies. However, other technologies are potentially good candidates for it in terms of performance like SpaceFibre, the successor of SpaceWire. It would nonetheless lead again space into a niche market with a very limited manufacturer for its satellite networks.

In future work, we will focus on TSN and especially on the configuration aspects for a space context.

## REFERENCES

- [1] Department of Defense – DoD, “MIL-STD-1553B – Aircraft Internal Time Division Command/Response Multiplex Data Bus,” Tech. Rep., 1978.
- [2] ISO 11898-2, “Road vehicles – Controller area network (CAN),” Tech. Rep., 2016.
- [3] European Space Agency, “SpaceWire-Links, nodes, routers and network,” ESA, Tech. Rep. ECSS-E-ST-50-12C, 2018.
- [4] P.-J. Chaîne, M. Boyer, C. Pagetti, and F. Wartel, “Formal specification of satellite on-board networks requirements,” 2020, working paper or preprint.
- [5] European Space Agency, “SpaceFibre – Very high-speed serial link,” ESA, Tech. Rep. ECSS-E-ST-50-11C, 2019.
- [6] IEEE, “IEEE Standard for Ethernet,” Tech. Rep. IEEE 802.3, 2013-2015-2018.
- [7] —, “IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks,” Tech. Rep. IEEE 802.1Q, 2008-2014-2018.
- [8] Aeronautical Radio Incorporated, “ARINC Report 664P7-1 Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network,” Tech. Rep. ARINC 664P7.
- [9] SAE, “AS6802 - Time-Triggered Ethernet,” Tech. Rep. SAE AS6802.
- [10] Janos Farkas, “Introduction to 802.1, Focus on the Time-Sensitive Network Task Group,” 2018.
- [11] P.-J. Chaîne, M. Boyer, C. Pagetti, and F. Wartel, “TSN Support for Quality of Service in Space,” in *10th European Congress on Embedded Real Time Software and Systems (ERTS 2020)*, Toulouse, France, 2020.
- [12] C. Pruvost, T. Planche, O. Notebaert, A. Rossignol, H.-J. Herpel, and A. Schuttauf, “Ethernet for space: an enabler for next-generation avionics,” in *2016 Data System In Aerospace (DASIA)*, 05 2016.
- [13] O. Notebaert, G. Montano, T. Planche, C. Pruvost, F. Wartel, A. Schuttauf, H.-J. Herpel, C. Honvault, and D. Jameux, “Towards spacewire-2: Space robotics needs: Spacewire missions and applications, long paper,” 10 2016, pp. 1–9.
- [14] R. Cummings, K. Richter, R. Ernst, J. Diemer, and A. Ghosal, “Exploring use of ethernet for in-vehicle control applications: Afdx, ttethernet, ethercat, and avb,” *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, vol. 5, pp. 72–88, 05 2012.
- [15] S. Schneelee and F. Geyer, “Comparison of ieee avb and afdx,” in *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*. IEEE, 2012, pp. 7A1–1.
- [16] T. Steinbach, H. Lim, F. Korf, T. C. Schmidt, D. Herrscher, and A. Wolisz, “Tomorrow’s in-car interconnect? a competitive evaluation of ieee 802.1 avb and time-triggered ethernet (as6802),” in *2012 IEEE Vehicular Technology Conference (VTC Fall)*. IEEE, 2012, pp. 1–5.
- [17] G. Alderisi, A. Caltabiano, G. Vasta, G. Iannizzotto, T. Steinbach, and L. L. Bello, “Simulative assessments of ieee 802.1 ethernet avb and time-triggered ethernet for advanced driver assistance systems and in-car infotainment,” in *2012 IEEE Vehicular Networking Conference (VNC)*, 2012, pp. 187–194.
- [18] H. Lim, D. Herrscher, and F. Chaari, “Performance comparison of ieee 802.1q and ieee 802.1 avb in an ethernet-based in-vehicle network,” in *2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT)*, vol. 1, 2012, pp. 1–6.
- [19] L. Zhao, F. He, E. Li, and J. Lu, “Comparison of time sensitive networking (tsn) and ttethernet,” in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. IEEE, 2018, pp. 1–7.
- [20] IEEE, “IEEE 1588v2 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,” Tech. Rep. IEEE 1588v2, 2008.
- [21] M. Boyer, N. Navet, and M. Fumey, “Experimental assessment of timing verification techniques for AFDX,” in *Proc. of the 6th Int. Congress on Embedded Real Time Software and Systems*, Toulouse, France, 2012.
- [22] O. Hotescu, “Vers la convergence de réseaux dans l’avionique,” Ph.D. dissertation, Toulouse INP, 2020.
- [23] IEEE, “Ieee 802.3br-2016, ieee standard for ethernet, amendment 5: Specification and management parameters for interspersing express traffic and ieee 802.1qbu, ieee standard for frame preemption,” Tech. Rep., 2016.
- [24] D. Thiele and R. 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*, 2016, pp. 1–9.
- [25] S. S. Craciunas and R. S. Oliver, “Smt-based task- and network-level static schedule generation for time-triggered networked systems,” in *Proceedings of the 22nd International Conference on Real-Time Networks and Systems*, ser. RTNS ’14, 2014, p. 45–54.
- [26] S. S. Craciunas, R. S. Oliver, M. Chmelfik, and W. Steiner, “Scheduling real-time communication in ieee 802.1qbv time sensitive networks,” in *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, ser. RTNS ’16, 2016.
- [27] S. S. Craciunas and R. S. Oliver, “Combined task- and network-level scheduling for distributed time-triggered systems,” *Real-Time Syst.*, vol. 52, no. 2, p. 161–200, 2016.
- [28] J. Migge, J. Villanueva, N. Navet, and M. Boyer, “Insights on the Performance and Configuration of AVB and TSN in Automotive Ethernet Networks,” in *9th European Congress on Embedded Real Time Software and Systems (ERTS 2018)*, 2018.
- [29] L. Maile, K. Hielscher, and R. German, “Network calculus results for tsn: An introduction,” in *2020 Information Communication Technologies Conference (ICTC)*, 2020, pp. 131–140.
- [30] L. Zhao, P. Pop, Z. Zheng, H. Daigmorte, and M. Boyer, “Latency analysis of multiple classes of avb traffic in tsn with standard credit behavior using network calculus,” *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2020.
- [31] IEEE, “Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks,” Tech. Rep. IEEE 802.1AS, 2011-2020.
- [32] H. Lim, D. Herrscher, L. Völker, and M. J. Walfi, “Ieee 802.1as time synchronization in a switched ethernet based in-car network,” in *2011 IEEE Vehicular Networking Conference (VNC)*, 2011, pp. 147–154.
- [33] IEEE, “IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks—Amendment 28: Per-Stream Filtering and Policing,” Tech. Rep. IEEE 802.1Qci.