

Implementation of a time-sensitive networking (TSN) Ethernet bus for microlaunchers

J. Sanchez-Garrido, B. Aparicio, J.G. Ramírez, R. Rodriguez, M. Melara,
L. Cercós, E. Ros, and J. Diaz

Abstract

The design of the aerospace systems for future aircraft requires the identification of new, suitable communication infrastructures that can overcome the limitations that often come with the use of the legacy, albeit well-proven, protocols that are routinely integrated in aerospace. This allows us to overcome the bandwidth constraints, large deployment costs or the lack in flexibility of other alternatives, like Spacewire, or legacy systems such as the MIL-STD-1553B bus. These protocols can be replaced with new technologies that can fulfill the greater real-time and interconnectivity demands of advanced scientific probes or manned spacecraft. The advent of the new microlauncher systems has all but confirmed this trend. In this context, we describe the design and implementation of a time-sensitive networking (TSN) bus for the avionics of the Miura 1 suborbital microlauncher. TSN represents an appropriate interface for this type of platform given its ability to provide the determinism and reliability expected in space-grade systems in combination with the higher data rates (Gigabit Ethernet) and greater flexibility of standard Ethernet. This has resulted in a TSN platform developed by Seven Solutions S.L. based on the commercially available Zynq-7000 devices from Xilinx. Thus, our design features a light-footprint field-programmable gate array (FPGA) architecture powered by a real-time executive for multiprocessor systems (RTEMS) operating system, which is currently pending its certification from the European Space Agency (ESA) for space applications. All these elements have been successfully integrated and validated for the avionics of the Miura 1 sounding rocket, which represents an illustrative case that verifies their applicability to similar scenarios.

J. Sanchez-Garrido, E. Ros, and J. Diaz are with the Computer Architecture and Technology Department at the University of Granada, Granada 18071, Spain.

B. Aparicio is with the Institute of Astrophysics of Andalusia (IAA-CSIC), Granada, 18008, Spain.

J.G. Ramírez and R. Rodriguez are with the Research and Development Department of Seven Solutions S.L., Granada, 18014, Spain.

M. Melara and L. Cercós are with the Avionics and On-Board Software Division of the Space Segment and Robotics Business Unit of GMV Aerospace and Defense SAU, Tres Cantos, 28760, Spain.

Index Terms

Aerospace buses, Microlauncher, RTEMS, Time-Sensitive Networking.

I. INTRODUCTION

Microlaunchers are versatile vehicles that have gained notoriety in the context of the so-called “*New Space*” endeavors [1]. These are launch vehicles that are usually meant to deploy relatively reduced payloads of up to 300 kg in some cases to the lower Earth orbits, i.e., the low Earth orbit or the sun-synchronous orbit, reaching moderate heights of up to ~ 700 km above the Earth’s surface. Hence, these platforms were developed with the goal of providing direct, convenient, and affordable access to space for diverse business opportunities, corporations, or government institutions, as an alternative to traditional launch platforms like Ariane [2]. This approach places new constraints, demands, and novel design parameters in the development cycle of these vehicles that differ from those typically found in a traditional space mission. Given the size of their payloads, these platforms do not need to be as powerful or large as traditional launchers. Instead, agile development cycles, the reusability of the launching vehicles themselves for multiple missions in some cases, or the feasibility of a fast production often drive the design efforts of these projects. The main consequence of this shift in paradigm is that manufacturers are trying to deploy commercially available off-the-shelf (COTS) components in order to reduce development effort and boost interoperability with standards originating from industrial applications or other domains. Furthermore, the use of commercial devices usually has the advantage of providing greater performance than their space-grade counterparts. That is the case of the Myriade [3] computing system, which was developed with specific design methodologies for aerospace and was integrated in the Demeter microsatellite [4]. Other notable examples are the computing systems of the Iridium NEXT satellites [5]. This trend also includes the avionics bus subsystems that are built into these platforms. This new paradigm has already been trialed in well-known projects, such as the VEGA [6], [7] launchers, although they did not use commercially available components, or the Ariane 6 launch vehicle, which made use of a customized version for aerospace of the COTS-based time-triggered protocol from TTTech [8], [9].

Typical bus implementations for the avionics subsystems of space launchers or satellites are normally based on protocols that have become de-facto aerospace industry standards, such as the well-known MIL-STD-1553B [10] or Spacewire [11], with the latter having a more widespread adoption nowadays. These buses can provide the determinism and reliability that are expected in aerospace applications for managing their processes and communications but come at the expense of requiring specialized, vendor-locked equipment that is not openly interoperable with that of other providers.

As a result, the companies driving the design of microlauncher platforms are overwhelmingly opting for communication buses that are based on open standard interfaces, and preferably Ethernet-based technologies. Ethernet has the advantage of defining a non-centralized type of network whose nodes implement an open link-layer protocol that is supported by the broad majority of vendors, thereby simplifying integration and interoperability. Its original best-effort (*BE*) nature made it unsuitable for its use in avionics systems. Nonetheless, this picture changed with the development of a number of enhancements over Ethernet implementing the determinism and reliability sought in avionics. These enhancements, taken as a whole, are known as time-sensitive networks (TSN) [12]. This approach allows the application of Ethernet-based interfaces to replace the legacy fieldbuses of multiple environments that range from industrial automation, automotive systems, and even through to avionics networks.

Consequently, this publication examines the use of a TSN system to support the deployment of the avionics network of the Miura 1 [13] microlauncher from GMV [14] and PLD [15]. This represents the type of next generation spacecraft that can benefit from integrating open-standard, COTS components to achieve faster development times and overall lower mission costs. Hence, Section II presents a review of different candidate fieldbus technologies that can be considered for these applications alongside the rationale for selecting TSN. Next, Section III introduces the use case of the Miura 1 sounding rocket and the architecture of its TSN-based avionics systems. Section IV then follows with an operational validation of the performance of the TSN-based avionics network that confirms the ability of the system to deliver the deterministic and reliable performance required for aerospace. Lastly, Section V discusses the main conclusions of our development effort and highlights its main results, such as the use of a purely COTS-based solution, its moderate footprint for a field-programmable gate array (FPGA), or the prospective certification from the European Space Agency (ESA) of its software environment.

II. REVIEW OF APPLICABLE FIELDBUSES FOR AEROSPACE AND THE OPEN, INTEROPERABLE ALTERNATIVE OF TSN

There are several comprehensive studies that have analyzed the prevalent landscape amongst fieldbus communications protocols, such as the report in [16] from NASA. Even though this report dates back to 2006, its conclusions can still be considered a relevant portrayal of the main types of interfaces of next generation spacecraft. Thus, it is stated in the report that these avionics implementations should be modular and able to accommodate systems with distributed physical and logical functionalities. Moreover, given the mission-critical and safety-critical requirements that are placed on space missions, they are expected to provide a service with guaranteed determinism and reliability. Furthermore, in the context of the reusability, lower mission costs, and optimized development effort of the new space vehicles, the development of new communication architectures will most likely be driven by the adoption of COTS solutions and components. Hence, in order to give the reader a general overview, the most relevant buses from [16] are now described.

A. *Review of Fieldbus Technologies Applicable to Space*

This point introduces some of the main communication technologies that can be applied for the avionics of the new generation space vehicles.

1) **MIL-STD-1553B**: This data bus is one of the earliest implementations of an interface for the communications of the different subsystems of aircraft avionics. It is a military standard initially drafted in 1976 [10] that has subsequently undergone several revisions, with its latest version appearing in 2006. It uses time-division multiple access (TDMA) and redundant transmissions to handle communications from the different nodes, which can take on the roles of either a bus controller (BC) or a remote terminal (RT). RTs are the units originating or receiving the data, whereas BCs coordinate bus access and the transmissions from the RTs. This protocol has been broadly used for aerospace applications such as the International Space Station. However, its modest bandwidth (~ 1 Mb/s) usually makes its use in combination with faster buses necessary, thus restricting its application to safety-related aspects. The literature shows that circumventing this limitation is feasible, but usually through custom solutions that lack interoperability, such as the 100-Mb/s enhancement in [17]. This limitation impacted the Exomars mission in 2016 [18], which had to utilize Spacewire for science data when it was not sending commands or telemetry

messages. Moreover, its high cost components and a lack of a broad market of suppliers further hinder its adoption as a suitable technology for the next generation spacecraft.

2) ***Spacewire***: Spacewire [11] was developed by ESA as a communication interface for satellites and spacecraft. It defines switched networks that can encompass large extensions with cascaded topologies. Congestion losses are avoided with the use of credit-based traffic shapers, thereby attaining rates of up to ~ 200 Mb/s over twisted pair copper interfaces. It can deliver deterministic transmissions, but its dependence on specialized, high-cost equipment prevents its application to the new space vehicle platforms, which lean towards COTS components. Indeed, Spacewire has been widely used for both past and current projects, such as Exomars in 2016 and the PLATO 2.0 mission. The latter was a deep-space mission supported with Spacewire [19] requiring high reliability, which involved the design of a complex redundant network. This ultimately limited the number of instruments (telescopes) aboard the vehicle as a result of its increased weight and cost.

3) ***Avionics full-duplex switched (AFDX) Ethernet***: AFDX Ethernet [20] is a bus specification developed by the Airbus Group that was built on the foundation of 802.3 (Ethernet) networks. It provides customized versions of the Ethernet link layer with medium access control (MAC) mechanisms, and tailored versions of the internet protocol and the user datagram protocol (UDP) which were adapted from commercial implementations. It can support transmission rates of up to 100 Mb/s with bounded end-to-end latency below $500 \mu\text{s}$. Additionally, a more detailed study of the performance of its switching elements is available in [21]. Transmissions are forwarded using virtual links (VLs) configured on a switch-by-switch basis with a redundant network topology. Its widespread adoption for the new aerospace vehicles is hampered by its cost and the lack of a substantial COTS market.

4) ***The time-triggered Ethernet protocol***: The time-triggered Ethernet (“*TTEthernet*”) protocol was first devised by the University of Vienna in 1993 and was initially targeted to automotive applications. It was subsequently standardized in the SAE AS6802 specification [22] and commercialized by TTTech as a hardware implementation allowing deterministic forwarding of mixed-criticality data according to a user-driven configuration for each network switch. For aerospace domains, it is supplied in the form of a specialized application-specific circuit (ASIC) [23]. Data redundancy, fault-tolerant synchronization and Gigabit Ethernet are supported, but there is a lack of interoperability with other vendors as the protocol has only been fully implemented by TTTech so far.

B. The Open Alternative of Time-Sensitive Networking

Time-sensitive networking stands out against the foregoing alternatives as it consists of a number of enhancements over regular Ethernet networks brought forward by the IEEE standardization committees (TSN Working Group [12]). Hence, it is an open specification supported by multiple vendors that is designed along four main pillars: *a)* system-wide synchronization, *b)* bounded end-to-end latency, *c)* system management and traffic class identification, and *d)* reliability and redundancy.

- a) The **system-wide synchronization** supplies a common network time base to synchronize the operation of the system nodes through a particularized implementation of the precision time protocol (PTP) [24]. This is known as gPTP timing and is described in the 802.1AS [25] specification.
- b) The **bounded end-to-end latency** can be guaranteed with the TSN traffic shapers, such as the time-aware traffic shaper (TAS) defined in 802.1Qbv [26] which allows the cyclic forwarding of TSN data flows (*streams*) during the time slots specified in its gate control list (*GCL*) schedule. Furthermore, delivery jitter can be reduced with the implementation of the frame preemption enhancement (802.1Qbu [27] & 802.3br [28]) of the Ethernet MAC to prioritize critical messages over the rest of the traffic.
- c) The **system-wide configuration and management** is essential to direct the overall effects and operation of the TSN network. As a rule of thumb, traffic identification rules for VLAN encapsulation (802.1Q [29]) of TSN streams, routing and topology information, the activation of redundant paths, or GCL definitions have to be supplied. This is usually achieved with resource reservation protocols (802.1Qcc [30] or 802.1BA [31]).
- d) The **redundant transmissions**, which are often required in avionics networks, are also supported in TSN systems with the use of the 802.1CB [32] component. This allows the transmission of duplicate messages over different physical paths as an enhanced layer of protection for highly critical data. Although we have located several approaches for building redundant avionics systems, such as the WDM-based architecture found in [33], this modular component of TSN provides a more convenient, standard-based mechanism for building redundant networks.

TSN systems can be supported over multiple variants of the Ethernet physical layer, such as 10/100Base-T, unshielded twisted pair (e.g., Broadcom's BroadR-Reach transceiver chip

[34]) for automotive applications, or even standard 1G/10G Ethernet. Most importantly, TSN functionalities are built on top of the ordinary Ethernet specification. This guarantees critical data delivery and backward compatibility with non-TSN compliant Ethernet equipment. This has led to a landscape with multiple vendors of interoperable, COTS solutions [35] (e.g., Cisco [36], National Instruments [37], Belden [38], ...) for the automotive, industrial, or avionics domains. Thus, the use of TSN stands out as a very valid alternative for the new space vehicles. We have confirmed this approach with the implementation of the specific use case of the Miura 1 microlauncher, which will be presented in detail in Section III.

C. Review of Network Topologies for Aerospace

We briefly describe some network designs for space to better frame our proposal within this field. Avionics systems usually adopt redundant network topologies, as can be seen in most system-level designs using fieldbuses for space. The architecture of past space missions can also show practical use cases with successful design approaches for supporting the avionics or the instrumentation of diverse types of space vehicles. Most of these communication systems usually adhere to three main types of topologies: ***bus line, star, or ring network layouts***. Moreover, they are usually enhanced with redundant interfaces to increase the robustness of their data transmissions.

Detailed descriptions of the avionics systems of different missions are not usually disclosed, and rather it is the instrumentation system that is more frequently documented publicly. The orbiter of the Exomars mission (2016) is a good example. It implemented a redundant ***bus line*** topology with the MIL-STD-1553B bus for sending telecommands and telemetry data, as evidenced by its interface with the on-board NOMAD instrument [39]. Contrary to this, other vehicles have used alternative approaches with more versatile protocols. This was the case of the PLATO mission for locating exoplanets. It used a system-wide approach to redundancy that encompassed its communication interfaces and the computing modules. Its instrumentation network was supported with redundant Spacewire links using a ***star*** topology [40].

Ethernet-based interfaces are another important trend in aerospace designs. The SUNRISE [41] mission, which consisted of a “balloon-borne” telescope in the stratosphere for observing the Sun, supported its data acquisition network with a simple Ethernet link. Other projects, such as the Ariane 6 [8], use *TTEthernet* for supporting fully redundant avionics networks with duplicated computing and traffic-switching elements [42]. AFDX is another well-known protocol

for avionics that borrows heavily from the Ethernet specification. It could almost be considered a precursor to TSN. AFDX is usually deployed to commercial aircraft using a layout of *multiple redundant rings* [43]. It is an inherently robust protocol that can be used for building fully redundant computing designs [44].

III. THE CASE OF THE MIURA 1 SOUNDING ROCKET. SYSTEM ARCHITECTURE

The Miura 1 sounding rocket is a specific case of a microlauncher vehicle that belongs with the new generation of space platforms. It was designed as a suborbital vehicle to carry out a mission for demonstrating and prototyping the new technologies and design methodologies that will eventually be integrated in the larger Miura 5 mission, which will be able to inject reduced payloads of up to 300 kg to the low Earth orbit (~ 400 km). Consequently, the Miura 1 has been designed to follow the general principles expected in microlauncher vehicles in terms of affordability and shorter mission time spans. These capabilities place important constraints on its aeronautic and on-board systems design.

This publication focuses on the implementation of the on-board avionics platform for Miura, which we introduced in [45]. Its overall requirements are summarized in the following points. Moreover, we have used a preliminary version of this research to illustrate the results of a broader study on time-sensitive networking [46].

A. Requirements of the Miura Avionics Platform

The focus on COTS and reusability in this project was a main driver of the design process. Since redundant systems and the handling of critical data are also needed in a space application, a TSN platform was a clear-cut choice from the outset. An overview of the requirements is outlined in Table I. These requirements translate into the specifications detailed in Section III-C.

B. The Avionics Use Case with TSN for the Miura Vehicle

Besides the generic requirements described in Table I, the Miura 1 launcher will need to implement a redundant network with a ring-style topology for handling three different classes of traffic that are commonly found in avionics applications: command and control (CC) traffic, housekeeping telemetry (*Tel*) data, and best-effort flows. The first two classes are usually high priority data requiring the use of redundancy protection features, whereas the latter class usually

TABLE I: Summary of the generic requirements for the avionics systems of the Miura 1 microlauncher.

| Communication Bus |
|--|
| <ul style="list-style-type: none"> – Open, standard-based. – Gigabit Ethernet, twisted pair. – TSN system, capable of handling multiple priorities, critical flows, and preempting traffic. – Up to 40 m system with 15 hops. Maximum distance between nodes of 10 m. – Determinism better than 50 μs over 10 hops (512 B over 1 Gb/s). – Support for redundant ring configurations with reliable, fault-tolerant network topologies. |
| Network Nodes |
| <ul style="list-style-type: none"> – Custom board design. – Implementation of two node subtypes: <ul style="list-style-type: none"> – a) 4-Port Main Board. – b) 2-Port Secondary Board. |
| Software Environment |
| <ul style="list-style-type: none"> – RTEMS v5 real-time operating system. – Real-time user-level tasks. – Static memory paradigm. – Application programming interfaces (APIs) for configuration. – Custom RTEMS Ethernet drivers. |
| Embedded Platform |
| <ul style="list-style-type: none"> – COTS sensors/actuators. – Preferable use of programmable logic devices coupled with feature-rich microprocessors. – Moderate resource usage to allow for integration with other mission components. – Suborbital vehicle with relaxed radiation hardening requirements. |

corresponds to best-effort monitoring or video feeds. Thus, we have materialized the foregoing specifications into the TSN system topology shown in Fig. 1.

As can be observed in the network topology, the TSN system proposed for the Miura 1 microlauncher is implemented using a redundant, ring-style system with different types of nodes.

- The **on-board computer (OBC)** handles most flows in the system. It emits and receives

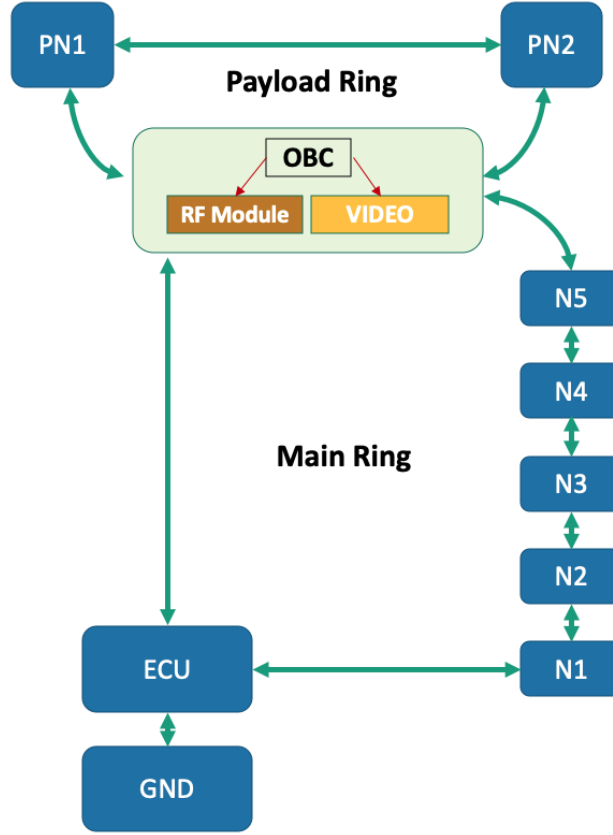


Fig. 1: Topology of the TSN network deployed to implement the avionics system of the Miura 1 microlauncher.

control and telemetry data, and operates as a sink for all the video feeds originating from the launcher. It is implemented with the four-port Main Board node. Furthermore, it also incorporates radio frequency (RF) interfaces for forwarding data to ground control stations.

- The **engine control unit (ECU)** handles telemetry and control flows. It also operates as a bridge between the ground node and the system avionics. It is built using the four-port Main Board.
- The **sensor and actuator boards (N_i nodes)** operate as gateways between the TSN avionics network domain and the specific sensors and actuators involved in the processes steered by the system. Hence, these nodes usually operate as TSN *listener* or *talker* devices handling commands, telemetry or, occasionally, best-effort video feeds. They are placed on the main ring, implemented with the dual-port Secondary Board nodes and linked in a daisy-chain. This translates into fewer four-port Main Board bridges.
- The **payload nodes (PN_k)** also operate as TSN *listeners* or *talkers* for handling commands

or telemetry messages bound for specific payload modules of the launcher. They are located on the separate payload ring.

- The **ground (GND)** node is a third-party controller from the IC-317x series (National Instruments [47]), which provides a stable clock synchronization source before the start of the mission. At this point, its link with the avionics system is severed.

Hence, the avionics nodes of Miura need to handle three main types of traffic. We briefly described these flows in the points below (a more detailed description is available in [48]).

- Critical command and control flows normally originate from the sensor/actuator nodes or the payload nodes and are routed towards the OBC and ECU units, where they can be forwarded to a ground station or be processed as part of a specific control loop. They have relatively reduced size ([50-100] B) and are emitted either periodically (10 messages per second) or sporadically, as in the case of alarm/one-time conditions. Their high priority implies that they normally require the use of redundant paths (802.1CB) and express queues in preemption-capable traffic shapers (802.1Qbu & 802.3Qbr).
- Medium priority housekeeping data is emitted from all the nodes in the system and forwarded towards the OBC. These messages carry telemetry data frames with modest sizes ([63-308] B) that are emitted at rates of up to ~ 246.4 kb/s. Delivery of telemetry data is guaranteed by the use of redundant network routes (802.1CB).
- The best-effort flows are the low priority streams originating from several sensor/actuator nodes (N_i). They consist of different video feeds emitted at a constrained rate of 8 Mb/s using 1500-B frames.

This is the typical architecture of the communication system of a launch vehicle. However, alternative designs for the system topology, traffic class assignment or even the software/hardware partitioning of our nodes could also be found through advanced simulation and modeling methods, such as those presented in the study in [49].

We have proposed a topology (Fig. 1) that can be considered a middle ground amongst the previous alternatives in the review from Section II-C. Thus, rather than building a fully redundant system, we have a “*lightweight*” approach with only two redundant rings, where we only replicate the communication interfaces and several user-defined flows. Other systems, such as AFDX, feature a more comprehensive methodology that also replicates the computing elements with a more complex ring topology. Nonetheless, our interoperable, COTS-based solution can provide

a competitive robustness level compared to traditional space fieldbuses.

C. Architecture of the TSN-capable Avionics Node

We have supported the preceding use case and its stated requirements with a TSN node design which is based on the Z-7030 device from the Zynq-7000 family [50] from Xilinx. These devices bundle together a dual-core ARM processor with a separate section of programmable FPGA logic (PL). The former can be used for running software components such as an operating system (OS), whereas the latter can be used for implementing custom coprocessors or Intellectual Property (IP) cores that interface with the ARM processor.

These nodes are implemented as an embedded platform – the Main node – with the architecture shown in Fig. 2, which shows a clear partition between its software modules and its FPGA resources. The reduced design of the Secondary node – a lightweight firmware variant of the Main node – is also highlighted in the figure. This platform has been developed on the foundation of a previous node designed for Smart Grid applications [51].

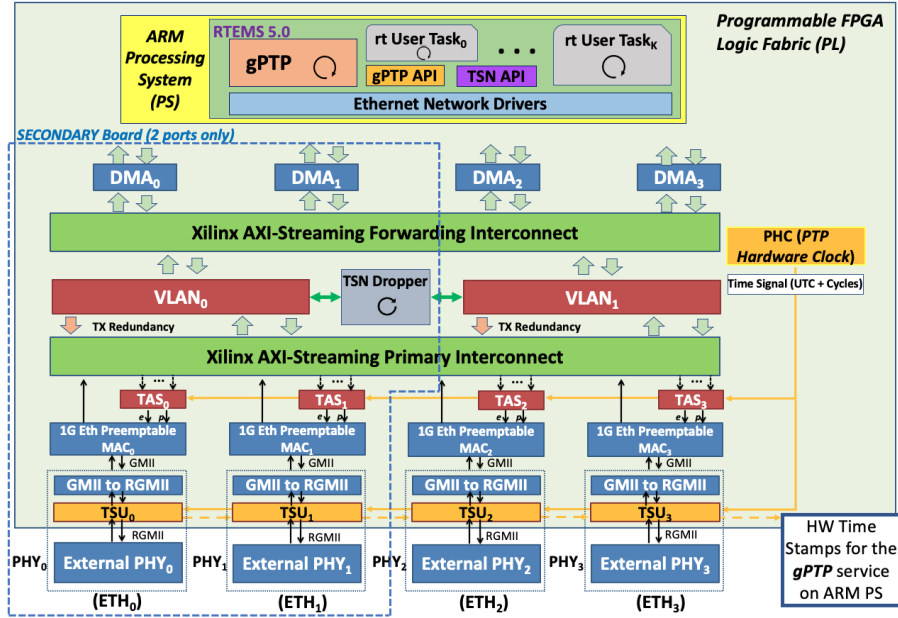


Fig. 2: General Miura 1 avionics nodes architecture diagram highlighting its RTEMS OS software elements on the ARM PS and its FPGA subcomponents for TSN networking.

1) **TSN Implementation on FPGA Logic:** The main subsystems of our TSN nodes can be broken down into three major groups.

a) *The Ethernet subsystem:* This comprises the blocks tasked with supporting a basic 1-Gb/s Ethernet communication service. These include several off-the-shelf and custom components: The Xilinx DMA [52] for forwarding data between the ARM processing system (PS) and the Ethernet ports, a preemptable Ethernet MAC (802.3br) adapted from the design in [53], RGMII/GMII bus infrastructure, and an external transceiver (PHY) chip.

b) *The timing subsystem:* This ensures the deterministic operation of the TSN node by providing and maintaining the common time reference required for synchronizing the TSN traffic shapers. In the FPGA logic, it is supported by the PTP hardware clock (PHC), which contains an internal system time representation disciplined by the gPTP software on the ARM PS, and the time-stamping units (TSUs), which generate hardware time stamps for estimating path delays in the gPTP protocol.

c) *The TSN and switching subsystem:* This implements the main functionality of the avionics nodes: deterministic transmissions with the capability of message forwarding amongst multiple ports. These elements include the main data switching interconnects (“*Primary*” and “*Forwarding*”); time-aware traffic shapers instantiated on a per-port basis (802.1Qbv) capable of supporting traffic preemption (802.1Qbu); VLAN modules (802.1Q) for traffic identification, flow routing, and redundant transmissions; and a Dropper Module for discarding redundant duplicates of messages (802.1CB).

2) **FPGA Resource Footprint:** Our TSN design will have to coexist with specific communication and flight mission IP cores, and therefore represents a tradeoff between its deterministic performance and its expected moderate use of resources. Hence, moderate FPGA footprint below 60% overall usage was a requirement of this design. The main consumers of resources are the TSN TAS traffic shapers, the VLAN modules, and the packet forwarding stages. These interconnect modules demand large amounts of block RAM (BRAM) primitives and general look-up table (LUT) logic for implementing buffering elements, finite state machines and traffic identification engines.

Consequently, we have customized our TSN implementation for making sensible usage of resources with several landmark architectural features, such as the use of shared VLAN modules between every two ports, a reduced number of VLAN tagging rules (16 rules per VLAN module) for identifying traffic, or the use of four different priority levels and modest buffer sizes (4 kB) for the queues of the TAS. Additionally, arithmetic computations and comparisons were offloaded on digital signal processing (DSP) primitives and no internal transceivers (GTPE_channel) were

needed since an external PHY chip was used.

This has resulted in an architecture that can be fitted in the Z-7030 device whilst also allowing for the integration of other mission-related IP cores. Thus, the resulting footprint is lower than what is found in alternative FPGA designs of TSN, such as those of Xilinx [54], which can only be targeted to larger Zynq-7000 or UltraScale devices to attain comparable features. Table II shows the moderate footprint in a Z-7030 device of this approach in terms of LUT fabric logic usage, both in the four-port Main node ($\sim 60\%$) and the dual-port Secondary node ($\sim 45\%$). Furthermore, both node designs keep their consumption of BRAM and DSP primitives below reasonable thresholds as to allow their integration with other third-party modules: BRAM usage is below $\sim 52\%$ for the Main node and $\sim 34\%$ for the Secondary, whereas both versions maintain their DSP consumption below $\sim 13\%$. The high-speed transceivers (GTPE_channels) remain unused in both cases as our design relies on external transceivers.

TABLE II: FPGA footprint of the TSN implementation of the avionics nodes for each type of resource in the Z-7030 device.

| Module | Slice LUTs | | Slice Registers | | Multiplexers | | BRAM | | DSPs | |
|--|--------------|---------------|-----------------|---------------|--------------|---------------|--------------|---------------|-----------|---------------|
| <i>Ethernet Subsystem (1 Port)</i> | 3771 | 4.80% | 5761 | 3.66% | 13 | 0.02% | 8 | 3.02% | 0 | 0% |
| <i>Timing Subsystem (1 Port)</i> | 2443 | 3.11% | 3175 | 2.02% | 2 | 0.003% | 2.50 | 0.94% | 0 | 0% |
| <i>TSN Subsystem (1 Port)</i> | 6887 | 8.76% | 9410 | 5.99% | 160 | 0.27% | 23.5 | 8.87% | 42 | 10.50% |
| <i>Switching Interconnects (a)</i> | 1796 | 2.28% | 1798 | 1.14% | 0 | 0% | 9 | 3.40% | 0 | 0% |
| <i>Switching Interconnects (b)</i> | 545 | 0.69% | 767 | 0.49% | 0 | 0% | 4.5 | 1.70% | 0 | 0% |
| <i>AMBA AXI Bus Infrastructure (a)</i> | 6150 | 7.82% | 5221 | 3.32% | 0 | 0% | 0 | 0% | 0 | 0% |
| <i>AMBA AXI Bus Infrastructure (b)</i> | 3488 | 4.44% | 3110 | 1.98% | 0 | 0% | 0 | 0% | 0 | 0% |
| Implementation Totals | | | | | | | | | | |
| (a) Main: 4 Ethernet Ports | 46408 | 59.04% | 62587 | 39.81% | 566 | 0.960% | 136.5 | 51.51% | 50 | 12.5% |
| (b) Secondary: 2 Ethernet Ports | 34649 | 44.08% | 48643 | 30.94% | 313 | 0.53% | 88.5 | 33.40% | 42 | 10.50% |

3) *The Software Environment*: The software components of the avionics nodes are executed on the ARM PS of the Zynq-7000 Z-7030 device, as shown in Fig. 2. This type of application requires the use of a real-time (rt) OS, and hence the real-time executive for multiprocessor systems – version 5 (RTEMS v5) [55] was selected as it is commonly used in this domain. The RTEMS OS was adapted to this platform with a static memory paradigm for enhanced execution safety and efficiency. Furthermore, it serves as the framework for supporting the main features of the system: a gPTP implementation with a cyclic executive service adapted from [56], custom TSN Ethernet drivers ported from the Xilinx Linux repositories, configuration utilities (APIs) for TSN and the gPTP protocol, and support for real-time user-level tasks. Ultimately, this development led to a robust avionics software platform which is under consideration for an ESA certification for space flight. It features, as its main novelty, a pioneering case of a

successful integration of the full IEEE 802.1AS timing stack from TSN for aerospace with an RTEMS OS environment.

4) **The Embedded Platform:** The avionics nodes were built as an embedded platform on a purpose-built printed circuit board (PCB) using a 160 mm by 160 mm form-factor for a streamlined integration in the structure of the Miura launcher. Several interfaces are prominently featured in the board design to interact with the main sensing and controlling elements of the microlauncher, such as a CAN interface, a low-pin count connector over an FPGA mezzanine card (FMC) interface, 20 generic I/O pins (GPIOs), and four Gigabit Ethernet ports. Furthermore, the platform was custom-built for performing a suborbital flight, which demands lower radiation resistance levels for the electronic components of the system [57]. Hence, an automotive-grade variant of the Z-7030 device which could withstand the thermal and mechanical stresses of a suborbital flight was selected. A picture of the embedded platform can be seen in Fig. 3, which supports both the Main and Secondary firmware variants of the avionics nodes.

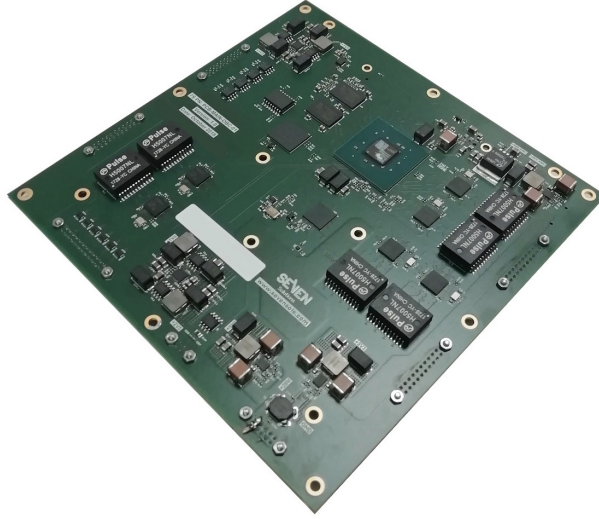


Fig. 3: The embedded avionics node of the Miura launcher.

IV. SYSTEM CHARACTERIZATION AND PERFORMANCE EVALUATION

We have characterized the operation of our avionics nodes through several test benches studying the performance of the TSN system under network scenarios comparable to those expected during the Miura 1 mission. These tests assess the performance of the system with two incremental approaches. Firstly, we will evaluate its determinism, timing distribution accuracy,

and transport reliability with unitary tests. Next, a comprehensive system integration test will show its ability to handle a TSN configuration scenario that simulates the environment of an actual mission.

A. Timing Distribution over gPTP

The synchronous operation of the TSN system is guaranteed by the action of the gPTP timing distribution service, which allows all the shapers in the network to forward traffic deterministically in a coordinated manner. Its use in an avionics environment requires a high level of accuracy, support for ring topologies, and a certain degree of fault tolerance. These aspects are studied using the laboratory setup of Fig. 4a. The effects of temperature variations are examined in the experiment depicted in Fig. 5a.

1) *Synchronization Accuracy with a Ring Network Topology:* Ring-style networks are commonly found in critical avionics systems, which require additional safeguards in the form of redundant systems. The gPTP protocol, through the use of the best master clock algorithm (BMCA), can support redundant timing paths and automatically adjust the role of each port (*Master, Slave, or Passive*) to adapt to a certain topology.

Hence, this test has a twofold goal. On the one hand, the action of the BMCA is verified by supplying the Main nodes in Fig. 4a with the appropriate configuration to ensure that nodes ID54 and ID48 operate as timing slaves of ID8. On the other hand, once the system is configured, the accuracy of the gPTP synchronization between the master (ID8) and the slave nodes (ID54 and ID48) is measured.

The action of the BMCA is verified by examining the system log files of the corresponding nodes, whereas the timing accuracy is determined by monitoring the synchronization jitter (slave-to-master time difference based on PHC values) reported by the gPTP service for approximately two hours. The results can be examined in Figs. 4b and 4c for the pairs ID8-ID54 and ID8-ID48, respectively, and show that ring topologies can be supported with accuracy levels below 100 ns. This level of performance is to be expected for general commercial and industrial-grade PTP implementations, as seen in [58] or our previous study for electrical substations in [51]. We have shown that a commercial-grade gPTP service could effectively supply the common time base of the avionics system with a sufficient level of accuracy to guarantee that the TAS shapers can be synchronized within a range of 50 μ s over 15 hops (see Table I). Furthermore, to the best

of our knowledge, this is one of the first applications of the gPTP timing service of TSN to an aerospace system with an RTEMS OS.

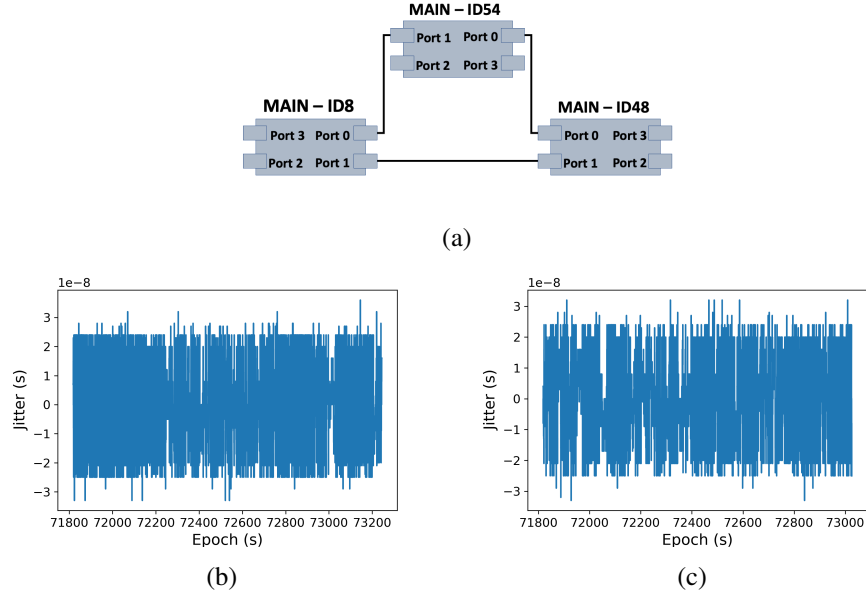


Fig. 4: Assessment of the gPTP synchronization efficiency using a ring topology configuration. **Fig. 4a** shows the experimental setup, whereas **Figs. 4b and 4c** show the attained accuracy between the master device *ID8* and the slave devices *ID54* and *ID48*, respectively.

2) *Short Loss of gPTP Protocol Messages*: The resilience of the gPTP service was tested by purposefully degrading the timing service implementation. This was achieved by instantiating a simple packet filter in the gPTP socket binding mechanism of the master node that could selectively discard protocol frames at specific rates. Hence, “peer delay request” (*Pdelay_Req*) and “synchronization” (*Sync*) messages will be discarded, as concluded in a preliminary analysis, to cause this degraded mode. Thus, the test measured the performance of the degraded gPTP between the nodes *ID8* and *ID48* for 20 minutes and found no significant loss of accuracy for degradation rates of up to 25% (Table III).

An assessment of the resilience of the protocol is paramount to validate its application for an aerospace environment. Transmission errors over the physical links of the network might be generated by the vibration or radiation stresses of the system. The transmission of critical data would remain relatively safe with redundant transmissions, but the overall quality of the timing synchronization could be affected. Our implementation of gPTP timing applies the necessary phase corrections upon the reception of the *Sync* and *Pdelay_Req* messages. Hence, the occasional loss of one of these messages may have an impact on the synchronization. Thus,

the experiment did indeed show that the synchronization could be affected, although to varying degrees depending on the type of message that was lost. *Pdelay_Req* messages are exchanged more frequently than the *Sync* frames. This produced the results of Table III, where we can observe that dropping up to 25% of the former produced a barely noticeable effect on the attainable performance of the synchronization service. In contrast, comparable message losses for the *Sync* messages were much more noticeable: the peak-to-peak synchronization offset at a 25% discarding rate could grow twice as large – up to 157 ns – as that of the initial degradation rate of 12.5%, which yielded a 77-ns peak-to-peak offset. Thus, it can be seen that the occasional drop of a *Pdelay_Req* message is not as dramatic as discarding *Sync* messages. Nonetheless, in general terms, we have verified that gPTP can withstand message losses while maintaining its accuracy of hundreds of nanoseconds even under substantial degradation rates.

TABLE III: Effects of different types of message losses on the accuracy of the gPTP protocol.

| Message Type | Loss Rate (%) | Mean (ns) | Std.Dev. (ns) | Peak-to-Peak (ns) |
|-------------------|---------------|-----------|---------------|-------------------|
| <i>Pdelay_Req</i> | 12.5 | -0.40 | 10.37 | 69 |
| | 25 | -0.26 | 10.11 | 65 |
| <i>Sync</i> | 12.5 | 0.15 | 10.93 | 77 |
| | 25 | 0.17 | 12.03 | 157 |

3) *Effects of Temperature Variation on gPTP Synchronization:* It is acknowledged that sub-orbital flights have less stringent demands on the electronic components of the mission, with accordingly lower levels of thermal stress. That is one of the reasons why we selected automotive-grade components for building the embedded avionics of the Miura 1. Nonetheless, the underlying synchronization of the system could be highly sensitive to temperature variations and, even though the nodes will be kept in enclosures maintained at relatively constant temperature, the robustness of the system had to be verified in the case that certain events in the mission such as the booster ignition could produce large temperature swings.

Hence, the action of the frequency and offset compensation mechanisms of gPTP timing was studied with the thermal chamber experiment depicted in Fig. 5a. The experiment aimed to simulate the conditions that the electronics of the launcher will face. We placed one of our Main boards, which is designated as ID48 in Fig. 5a, inside the thermal chamber for this study. The Slave Board 48 will be disciplined to the external timing source supplied by the master at ID8. During the experiment, the impact of temperature variations will be examined by monitoring the evolution of the master-to-slave jitter between the nodes. We can achieve this by forwarding the

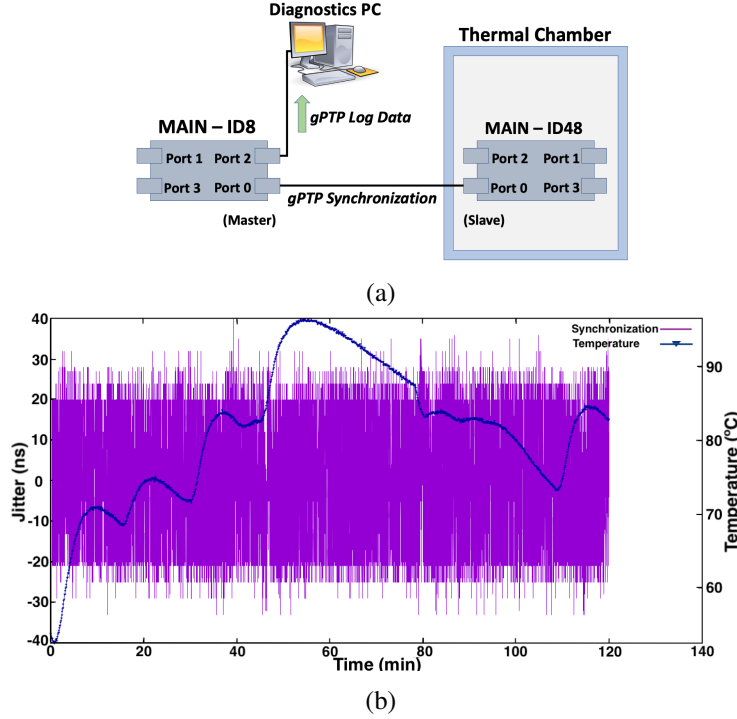


Fig. 5: Robustness of the gPTP synchronization in the thermal chamber experiments. **Fig. 5a** shows the experimental setup and **Fig. 5b** contains the test results.

log files of the operation of the gPTP service to a diagnostics PC over a TSN stream for further analysis and inspection.

The experiment consisted of a 2-hour test, which subjected the Slave Board 48 to a temperature curve that applied stepwise increments until the system reached the point of ~ 100 °C, after which the temperature gradually tapered off. We verified upon examining the log files that the synchronization accuracy of the system did not seem to be affected and remained below 100 ns, as shown in Fig. 5b. This shows that the gPTP implementation that we have supplied can withstand slow temperature changes without any significant loss of accuracy. This enhanced resilience is required for its deployment in an industrial or aerospace environment.

B. Verification of System-level Determinism of the TSN Network for an Avionics System

The avionics of the Miura launcher is supported by a TSN network that can guarantee the deterministic forwarding and delivery of time-critical flows. We verified this claim by using a multistage approach that starts by measuring the overall system-level determinism of the network and then moves on to assessing the effects of the traffic shaping methods in the TSN system.

1) *Evaluation of Forwarding Efficiency of the TSN Nodes:* The forwarding efficiency of the underlying FPGA switching fabric is the initial aspect considered for assessing the overall system-level determinism of the nodes. Hence, this test consists of a simple study intended to model the determinism of the hardware pipeline used for forwarding traffic under the assumption that the entire communication channel is idle, i.e., without any other competing traffic present other than the underlying gPTP for synchronizing the nodes, so that it can therefore be fully devoted to transmitting probe frames.

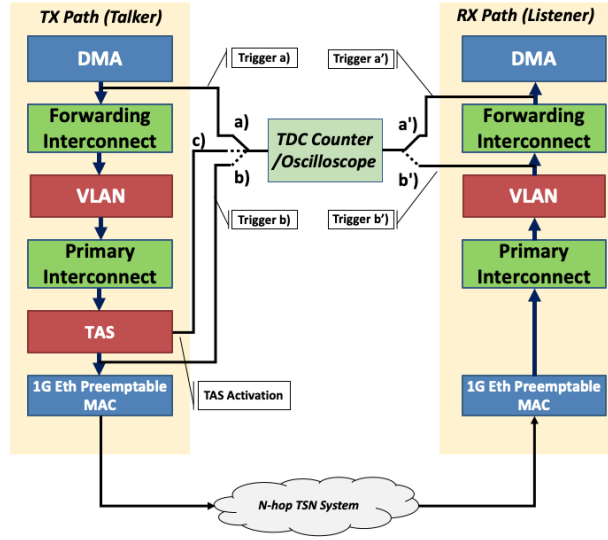


Fig. 6: Diagram of the laboratory setup for studying the attainable determinism of the TSN avionics network. The diagram shows the probing points at generic TSN talkers and listeners of the signals considered for different experiments.

Thus, this test provides an estimation of the attainable determinism of the network by injecting up to one million probe frames between two Main nodes linked together in a daisy-chain (a one-hop network). The nodes were fitted with a diagnostics core that could be connected to specific sections of the internal data bus to produce an analog trigger upon detection of an Ethernet frame, as shown in Fig. 6. Next, the time of flight of the message can be measured with a simple frequency counter [59]. We use the frequency counter as a time-to-digital converter (TDC), which records time stamps of the triggers produced at the egress (a) and ingress (a') points of the probe message. We can then use these time stamps to calculate the time difference between the generation of the triggers at the probing points and, thus, derive the propagation time through the network.

As a result, this test provides an estimation of the expected packet delay variation (PDV) of the system under ideal conditions. For 500-B probe frames, the average single-hop delay was pegged to $17.49 \mu\text{s}$ with a PDV of $\sim 301 \text{ ns}$, as observed in Fig. 7. The figure depicts the generic shape of the attainable latency distribution under ideal conditions with our current implementation. We characterized this distribution by plotting the histogram of the end-to-end latency alongside its corresponding kernel density estimate. This allowed us to observe that it does not conform to the shape of a normal distribution: its average value is centered about $17.49 \mu\text{s}$ with a standard deviation (σ) of 12.23 ns , but it does exhibit a long tail. This indicates that most values fall within a close range (2σ) of the latency average of $17.49 \mu\text{s}$. However, our measurements with the frequency counter showed that the data distribution is bounded between 17470.47 ns and 17771.44 ns , thereby yielding a peak-to-peak value of 300.97 ns and producing the tail that we observe in the figure. Given the shape of the distribution, only a minority of the frames will experience an increased delay in their transmission, but it is a sign of the limitations imposed by the tradeoffs of the architecture. Hence, the use of reduced buffer sizes and interconnects with internal arbitration mechanisms to conserve FPGA resources may be responsible for this disparity between the standard deviation and the PDV. Our experiment aims to depict the behavior of our network when it is idle, thus this extended tail can be attributed to interconnect contention issues with the gPTP synchronization messages. As a result, in the event of contention, our crossbar interconnects would have to arbitrate between the messages for the experiment and those for gPTP. This increases the packet delay variation up to the $\sim 301 \text{ ns}$ that we determined empirically. This is also a reasonable value that follows the expected behavior of our architecture (see Section IV-C1). Thus, as implied in the expression (1), we can expect the arbitration time associated with the forwarding of a 60-B gPTP frame to be around 300 ns .

2) *Effects of Traffic Shaping and Flow Aggregation:* Since the use of a fully dedicated channel is impractical for a viable application, TSN systems may have different flows share the same physical link using the time-based multiplexing approach implemented through the time-aware traffic shapers. This methodology allows the reservation of the communication channel for different TSN flows during their designated time slots. This is shown in the test results depicted in the oscilloscope captures of Fig. 8.

This test, whose results are shown in the figure, monitors the transmission of TSN messages between two daisy-chained devices. The messages are transmitted from the ARM PS of the talker node (*a*)), forwarded when their TAS queue is active (*c*)), and received at the ARM PS

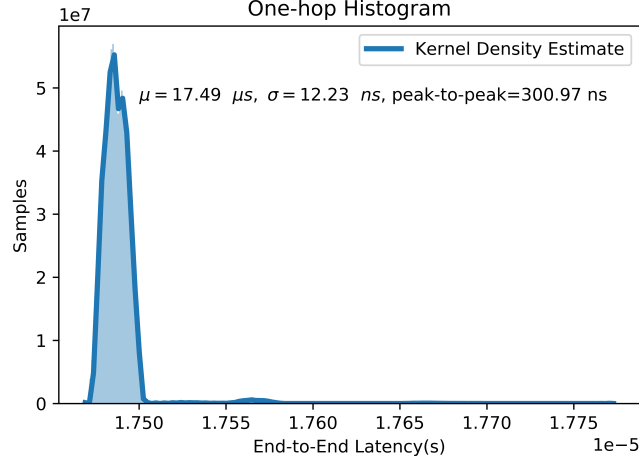


Fig. 7: Reference end-to-end latency graph of a single-hop TSN system, showing that the attainable determinism under ideal conditions sits around $17.49 \mu s$ with a PDV of 300.97 ns and a standard deviation of 12.23 ns.

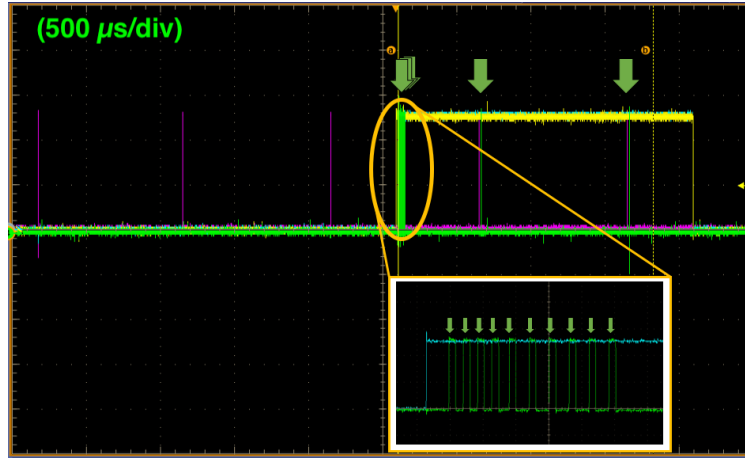


Fig. 8: Oscilloscope capture showing the application of a simple shaping policy to ensure the reception of TSN frames within a specific time slot. The arrows indicate the reception of TSN messages at the listener node.

of the listener node (a')). The action of the TAS shaping policy is noticeable in Fig. 8, as the packets produced at the talker (magenta trace - a)) can only be received at the listener (green trace - a')) during their corresponding slot (yellow trace - c)). Otherwise, they are held at their shaper queue and released, potentially as a burst, at the start of the slot. This latter effect, shown in the magnified part of Fig. 8, is produced by the lack of synchronization between the emitter of the probe frames in the test and the TAS traffic-shaping cycle (the blue trace in the magnified section - c)).

C. Evaluation of Determinism in a Demonstration Avionics Setup

The characterization phase concludes with the evaluation of the attainable determinism of our TSN solution in a simplified version of the Miura avionics network from Fig. 2. The test evaluates the performance of the system when handling flows representative of a typical suborbital flight with the setup depicted in the diagram of Fig. 9. This is a representative test that we also demonstrated as part of our presentation in [45]. It features a ring network topology that will be used for routing the most critical flows redundantly. This enhances the robustness of the system to link failures and is also in line with the requirements from Table I. Furthermore, in avionics systems, resilience and redundant networks are an essential part of this requirement. Hence, any realistic test bench for avionics should at least feature some form of this topology to derive meaningful results. These considerations led to the design of our test bench from Fig. 9.

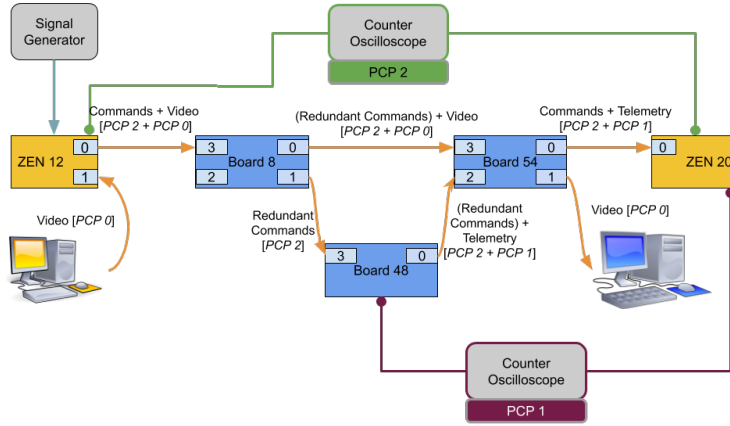


Fig. 9: Diagram of the simplified test bench for simulating a traffic scenario representative of a suborbital flight mission.

The test uses three Main nodes (Boards 8, 48 and 54) to create a ring topology, and two TSN end-points for injecting and receiving time-critical flows based on the dual-port ZEN Board from Seven Solutions S.L. [60], which the authors have used as a development platform in a previous project [51]. Moreover, our demonstration system from Fig. 9 will also feature the typical flows of an avionics environment so that its results are meaningful in the context of aerospace networks. Hence, the system will handle TSN flows for command and control messages, payload telemetry, bulk best-effort video, and gPTP synchronization messages, as indicated in Table IV.

The actual experimental setup can be examined in Fig. 10, where we have also indicated the corresponding designation for each element. Thus, the frequency counter is visible in the

TABLE IV: Outline of the different traffic classes handled in the avionics TSN demonstrator and their corresponding traffic profile characterization.

| Traffic Class | Priority Level | Traffic Profile | Critical | Redundant |
|-------------------------------|----------------|------------------------------------|----------|---------------|
| <i>gPTP</i> (Synchronization) | 3 | 60-B <i>gPTP</i> protocol messages | <i>e</i> | (BMCA) |
| <i>Command and Control</i> | VLAN PCP 2 | 400 B at variable rate | <i>e</i> | Yes (802.1CB) |
| <i>Payload Telemetry</i> | VLAN PCP 1 | 500 B at 1 frame/ms | <i>e</i> | - |
| <i>Bulk (Video)</i> | VLAN PCP 0 | 750 B at 20 Mb/s | <i>p</i> | - |

upper left (the TDC) alongside the signal generator. The former will be used for calculating the message flight time over the network and for measuring the integrity of the transmission. The latter produces periodic analog triggers that are injected into the ZEN talker (*ZEN 12* in Fig. 9) to simulate the production of the critical messages of the test. The redundant TSN ring can be seen in the center of the image (Boards 8, 54, and 48) and is made up of four-port Main boards that were enclosed in temporary casing during the tests. The critical and telemetry flows are received at the ZEN listener (*ZEN 20* in Fig. 9), which also forwards the best-effort video to the receiver PC (in the lower right of the image). The emitter PC, visible in upper right corner, injects the video feed into the TSN network through one of the ports of the ZEN talker. Thus, we can simulate the transmission of a low-priority video feed alongside the typical traffic of a space mission with this setup, which can ensure its deterministic and robust delivery with the configuration outlined in the following section.

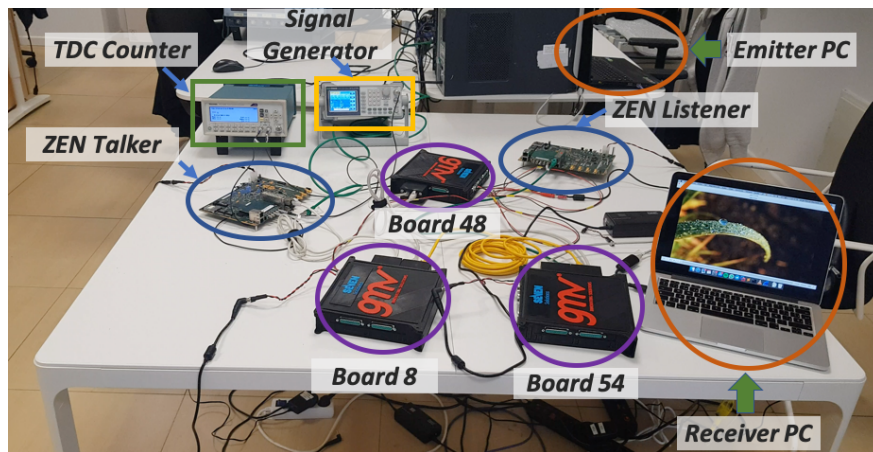


Fig. 10: Picture of the laboratory setup for building the avionics demonstrator test bench.

1) *Configuration of the Demonstration Setup:* We have configured our test bench from Fig. 9 so that commands and control messages are emitted from the TSN ZEN talker device and routed redundantly to the ZEN listener at the other end of the network. These messages are produced upon reception of analog triggers into the ZEN talker from a signal generator with a variable rate. Likewise, bulk video is injected through one of the available Ethernet ports of the ZEN talker and sent to a receiving PC attached to the ZEN listener. Payload telemetry is emitted from the Main Board 48 and delivered to the ZEN listener. Additionally, gPTP messages are exchanged amongst all the nodes on a point-to-point basis. Synchronization, commands and control, and telemetry are deemed time-critical flows requiring non-preemptable, express (e) forwarding and redundancy, which is either implemented through the features of 802.1CB for TSN flows or by the action of the BMCA for synchronization. Bulk video is usually low priority and is hence preemptable (p).

The foregoing flows are assigned to the TAS queue indicated in their VLAN priority code point (PCP), except for the gPTP messages, which are always assigned to the higher priority queue ($Q3$). The traffic shaping policy applied to the entire network can be examined in Table V, which describes the slots designated for forwarding the different flows, their associated queue (Qk) settings, and the corresponding relative offsets (“*base time*”) of the shapers in each node to compensate for forwarding delays in the network. We determined our routing and forwarding settings analytically to adapt to the bandwidth and traffic requirements of our experimental setup. More complex topologies with aperiodic flows, i.e., sporadic data feeds, could benefit from the use of different algorithms for estimating these parameters, as shown in [61].

TABLE V: Traffic shaping policy of a 1.012 ms cycle applied to all the nodes in the demonstration TSN setup indicating slot contents, queue open/closed (O/c) and preemption (e/p) status, and the *base time* for each node.

| Interval (μs) | $Q3(e)$ | $Q2(e)$ | $Q1(e)$ | $Q0(p)$ | Slot Contents |
|-----------------------|---------|---------|----------|----------|-------------------|
| 28 | O | O | O | O | gPTP, CC, Tel, BE |
| 478 | | | c | c | gPTP, CC |
| 28 | | | O | O | gPTP, CC, Tel, BE |
| 478 | | | c | c | gPTP, CC |
| Node ID | ZEN 12 | Board 8 | Board 48 | Board 54 | ZEN 30 |
| Base Time (μs) | 0 | 10.5 | 21 | 21 | - |

Thus, in the case of our avionics demonstrator, we designed the simple GCL policy of Table

V that could allow us to showcase one of the main features of TSN systems: the aggregation of different types of traffic while ensuring their deterministic delivery and the preservation of the integrity of the higher priority flows. The cycle time is a fundamental parameter for the design of the scheduling policy, as it sets the minimum rate for serving the different queues of the TAS modules. We designated a cycle time of 1.012 ms in accordance with the characteristics of the traffic classes that we described in the Table IV. In the table, since the telemetry traffic was the only flow with a predetermined periodicity of 1 message per millisecond, we took it as the reference for setting the base cycle time of the GCL policy.

Next, we designed the slots within the cycle. As the telemetry messages have a 500-B frame size, we can then estimate their transmission time through the TAS module with the implementation-dependent expression in (1), where N_B is the frame size in bytes, and t_{clk} is the 20-ns clock cycle of the main system bus. This yields a transmission time of $\sim 2.5\mu s$. Hence, designating a slot of $28\mu s$ every $478\mu s$ should ensure that no telemetry messages are lost to congestion, as its assigned TAS queue would only be able to accommodate up to 8 telemetry frames when it is idle. We designed our GCL to ensure that the critical messages are always forwarded, even at their variable generation rate. These are the higher priority messages of the system besides the gPTP synchronization and are also designated as express (e) for reducing their PDV. Furthermore, they are assigned two reserved slots of $478\mu s$. As for the video messages, their transmission would take $3.75\mu s$ according to (1). If they were transmitted at a constant rate, a new frame would be received at the corresponding video queue of the TAS every $300\mu s$. Consequently, it can be seen that this policy would also be able to handle the video flow with the two $28\mu s$ slots that we have provisioned for the best-effort traffic. Even though the slots are evenly spaced $478\mu s$ apart, the probability of video congestion losses is relatively low as the TAS video queue ($Q0$) can hold up to 5 video packets when it is inactive.

$$t_{x,TAS} = \frac{N_B}{4} \cdot t_{clk} \quad (1)$$

Actual video flows may also exhibit the occasional burst of frames. In these cases, it is expected that video congestion losses will occur. This would demonstrate how TSN can deliver critical flows deterministically at the expense of the best-effort traffic. In these cases, the TAS would apply a strict priority selection algorithm in combination with the enhancements of frame

preemption to ensure that the critical messages are forwarded first.

2) *Experimental Results:* This experiment has allowed the evaluation of the achievable determinism and reliability of the Miura avionics in a test-bed that simulates the microlauncher network supporting a typical flight.

Thus, we have assessed the overall system determinism by measuring the end-to-end latency of both the *CC* and *Tel* messages to estimate their corresponding PDVs. Hence, their flight time through the network has been measured as the difference between the time at the egress paths (*b*)) of the ZEN talker and Board 48 for the *CC* and *Tel* flows, respectively, and that at the ingress path (*b'*)) of the ZEN listener, which is the designated end-point for both flows. Moreover, even though they are both designated as express, non-preemptable traffic, they have different associated levels of priority for the TSN traffic shapers. This results in the end-to-end latency values obtained in Fig. 11, where we could observe that the time-driven schedule of the shapers avoids interference amongst the *CC* flow, the *Tel* traffic injected at Board 48, and the background video traffic.

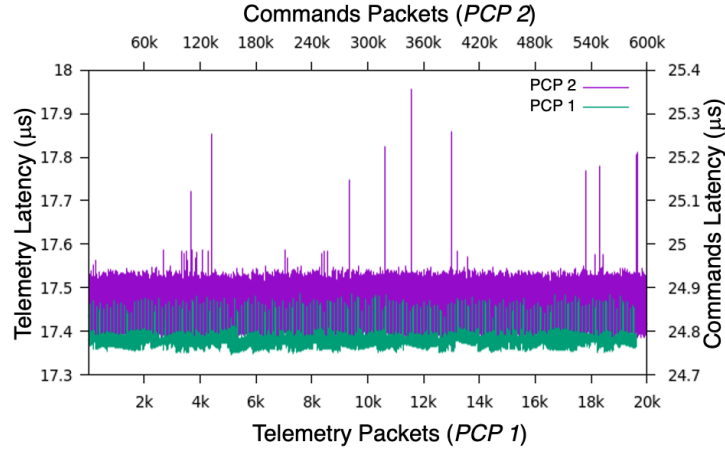


Fig. 11: End-to-end latency for the TSN frames of the time-critical flows injected during the experiment. PDV and latency remain stable as interference between *CC* and *Tel* messages is averted by the traffic shapers.

As a result, the *CC* flow attained an end-to-end latency centered about $\sim 24.85 \mu s$, with a PDV of 600 ns, whereas the *Tel* flow, which takes a shorter path through the network, achieved a PDV of 141 ns and a lower latency of $\sim 17.4 \mu s$. However, it should be noted that the PDV of *CC* traffic remains below ~ 150 ns most of the time, with the occasional peak of up to 600 ns. This behavior has been attributed to the larger generation rate of *CC* messages, hence making

interference with higher priority gPTP messages more likely. This effect is a manifestation of the tradeoffs in our design to reduce resource usage, and it could be improved with an enhanced architecture with larger switching and buffering elements.

Lastly, we studied the system reliability by characterizing the *CC* TSN flow forwarded during the experiment. As *CC* traffic has been deemed time-critical, its transmission was protected by the use of a redundancy scheme with the 802.1CB mechanism. The number of *CC* frames at the egress path (b) of the ZEN talker, and those at the ingress path (b') of the ZEN listener after discarding duplicates at Board 54 were totalized with a counter instrument to detect packet losses along the redundant network topology or when one of its redundant links (direct path or fallback path through Board 48) was severed. We thus verified that the integrity of the flow was maintained since no packet losses were detected in either case: the counter totalization function indicated that the difference between the number of transmitted and received *CC* frames remained constant at zero throughout the experiment. We also made a qualitative assessment of the video feed, which remained operational during the tests and could be played back at the receiver PC attached to Board 54, except when we severed its transmission route during the verification of the redundancy feature. Moreover, the overall robustness of the system could be enhanced in future upgrades with the use of preprogrammed reconfiguration strategies, as proposed in [62], which could activate different redundant paths in response to different fault events when more complex topologies are used.

V. CONCLUSION

Our research has shown the development of a TSN platform customized to meet the resource-constrained scenario of the Miura 1 microlauncher. Thus, the feasibility of applying a TSN system for building the avionics network of the microlauncher has been confirmed through a series of experiments. This design methodology falls into the category of the so-called COTS solutions that have become the predominant trend in these types of vehicles.

This resource-optimized TSN bus can thus provide the determinism and reliability required to forward the time-critical flows normally handled in avionics systems by using the TSN enhancements of time-driven forwarding schedules, frame preemption, and redundancy. It is consequently a promising solution as compared to other specialized space buses and interfaces, such as Spacewire or MIL-STD-1553B, given the comparable levels of determinism and reliability of our solution. Furthermore, provided that our design is supported with a COTS platform

using standard-based Ethernet physical interfaces, it can also achieve a far superior bandwidth of up to 1 Gb/s. In addition, as indicated in [63], it represents a comprehensive platform for testing, simulating, and verifying the integration of a complete avionics system.

Thus, our solution achieved enhanced levels of performance and interoperability. Its architecture based on the automotive-grade Z-7030 device is also highly flexible and robust enough to withstand the thermal, mechanical and radiation stresses of a suborbital flight. It provides an FPGA framework for supporting its TSN communications and an ARM microprocessor for executing the RTEMS OS featuring a gPTP synchronization service. Furthermore, at the time of writing this article, this has been one of the first cases of a successful integration of gPTP for TSN in an RTEMS-based system. Our proposed design can deliver critical flows with PDV values below 1 μ s. Furthermore, given the flexibility of the platform, this figure could be further improved in future projects with more relaxed constraints in terms of resources by implementing larger buffers, faster packet processing stages, or more efficient switching interconnects, like the FPGA router design in [64].

In consequence, we have shown a viable application of a TSN system to support the avionics of the Miura 1 microlauncher with a solution based on commercially available components that is highly flexible, interoperable, and easily upgradeable. Furthermore, the underlying determinism of its TSN network is complemented with an RTEMS implementation, currently under consideration for an ESA certification, which also supports real-time tasks and features a novel integration of gPTP synchronization for aerospace. Hence, our solution can deliver the deterministic performance required for the avionics of Miura and could be considered a plausible COTS-based alternative for similar microlauncher platforms.

ACKNOWLEDGMENT

This study has been partially supported by the Amiga-7 (RTI2018-096228-B-C3) grant and it was conducted in the context of the Gigabit Ethernet TSN Deterministic Network (GETDEN) project of the European Space Agency. The authors would also like to thank Emanuele Di Sotto, from GMV Aerospace and Defense SAU, for his support and supervision of this project; and Luis Medina Valdés, Jorge Manuel Machado Cano, and Marco Fuentes García, from Seven Solutions S.L., for their contribution to the development of the TSN nodes for the avionics system of the Miura 1 microlauncher.

REFERENCES

- [1] M. Tugnoli, M. Sarret, and M. Aliberti, "Overview on Micro Launchers," in *European Access to Space: Business and Policy Perspectives on Micro Launchers*, Springer, Cham, 2019, ch. 2, pp. 5–28.
- [2] "Enabling and Support: Ariane 5," ESA (European Space Agency). Accessed: May 24, 2020. [Online]. Available: http://www.esa.int/Enabling_Support/Space_Transportation/Launch_vehicles/Ariane_5
- [3] V. Dubourg, J. L. Carayon, P. Danto, and G. Galea, "An innovative onboard computer for CNES microsattellites," in *Proc. 21st DASC Conf.*, Irvine, CA, USA, Oct. 27-31, 2002, vol. 2, pp. 9B2–9B2.
- [4] T. Cussac, M.A. Clair, P. Ultré-Guerard, F. Buisson, G. Lassalle-Balier, M. Ledu, C. Elisabelar, X. Passot, and N. Rey, "The Demeter microsatellite and ground segment," *Planet. Space Sci.*, vol. 54, no. 5, pp. 413–427, 2006.
- [5] P. Murray, T. Randolph, D. Van Buren, D. Anderson, and I. Troxel. "High performance, high volume reconfigurable processor architecture," in *2012 IEEE Aerosp. Conf.*, Big Sky, MT, USA, 2012, pp. 1–8.
- [6] "Enabling and Support: VEGA," ESA (European Space Agency). Accessed: May 24, 2020. [Online]. Available: http://www.esa.int/Enabling_Support/Space_Transportation/Launch_vehicles/Vega
- [7] "VEGA (Maiden Flight of new ESA Launcher and selected Payloads)," Earth Observation Portal (European Space Agency). Accessed: May 24, 2020. [Online]. Available: <https://directory.eoportal.org/web/eoportal/satellite-missions/content/-/article/vega>
- [8] R. Clavier, P. Sautereau, and J.F. Dufour, "TTEthernet, a promising candidate for Ariane 6," in *Proc. DASIA*, Warsaw, Poland, June 3-4, 2014, vol. 725, p. 34.
- [9] "Hi-Rel Solutions for Space Launch Vehicles," TTTech. Accessed: July 9, 2020. [Online]. Available: <https://www.tttech.com/markets/space/projects-references/ariane-6/>
- [10] Aircraft Internal Time Division Command/Response Multiplex Data B, MIL-STD-1553B, United States Department of Defense, Sept. 1978.
- [11] SpaceWire–Links, Nodes, Routers, and Networks, ECSS-E-ST-50-12C Rev.1, European Cooperation for Space Standardization (European Space Agency), May 2019.
- [12] *Time-Sensitive Networking (TSN) Task Group*, Accessed: Dec. 17, 2019. [Online]. Available: <https://1.ieee802.org/tsn/>
- [13] PLD Space. *Miura 1 information site*. Accessed: July 16, 2020. [Online]. Available: <https://www.pldspace.com/en/miura-1>
- [14] GMV – *Official Company Site*. Accessed: July 16, 2020. [Online]. Available: <https://www.gmv.com/en/>
- [15] PLD Space, *Official Company Site*. Accessed: July 16, 2020. [Online]. Available: <https://pldspace.com/en/>
- [16] D.A. Gwaltney and J.M. Briscoe, "Comparison of Communication Architectures for Spacecraft Modular Avionics Systems," NASA, Marshall Space Flight Center, Huntsville, AL, USA, Tech. Rep. NASA/TM-2006-214431, June 1, 2006. [Online]. Available: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20060050129.pdf>
- [17] P. Pendyala and V. S. R. Pasupureddi, "100-Mb/s enhanced data rate MIL-STD-1553B controller in 65-nm CMOS technology," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 6, pp. 2917–2929, 2016.
- [18] Steve Parkes. "ESA ExoMars," in *SpaceWire User's Manual*, STAR-Dundee Ltd., 2012, ch. 2, sec. 5.2, pp. 36–37. [Online]. Available: https://www.star-dundee.com/wp-content/star_uploads/2019/05/SpaceWire-Users-Guide.pdf
- [19] J.M. Mas-Hesse, R. Urquí-O'Callaghan, J.C. Suárez, J. Cabrera, H. Deeg, and A. Balado, "The PLATO 2.0 mission. Spanish contribution," in *Proc. 11th Scientific Meeting Spanish Astronomical Soc.*, Teruel, Spain, Sept. 8-12, 2014, pp. 755–763. [Online] Available: <https://www.sea-astronomia.es/sites/default/files/archivos/proceedings11/instrumentacion/mashessejm/mashessejm.pdf>
- [20] Aircraft Data Network Part 7 Avionics Full-Duplex Switched Ethernet (AFDX) Network, ARINC Specification 664, Part 7, Aeronautical Radio Inc., June 2006.

- [21] C. Suthaputchakun, Z. Sun, C. Kavadias, and P. Ricco, "Performance analysis of AFDX switch for space onboard data networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 4, pp. 1714–1727, 2016.
- [22] Time-Triggered Ethernet, AS6802, SAE International, Nov. 2016. [Online]. Available: <https://www.sae.org/standards/content/as6802/>
- [23] "Reaching for the sky with certified and safe solutions for the aerospace market," TTTech Group. Accessed: May 25, 2020. [Online]. Available: [www://https.tttech.com/wp-content/uploads/TTTech_Aerospace-Core-Brochure-1.pdf](http://www.tttech.com/wp-content/uploads/TTTech_Aerospace-Core-Brochure-1.pdf)
- [24] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IEEE 1588-2008, 2008.
- [25] IEEE Standard for Local and Metropolitan Area Networks – Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks, IEEE 802.1AS-2011, 2011.
- [26] IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic, IEEE 802.1Qbv-2015, 2015.
- [27] IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks – Amendment 26: Frame Preemption, IEEE 802.1Qbu-2016, 2016.
- [28] IEEE Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic, IEEE 802.3br-2016, 2016.
- [29] IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks, IEEE 802.1Q-2018, 2018.
- [30] IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks – Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements, IEEE 802.1Qcc-2018, 2018.
- [31] IEEE Standard for Local and metropolitan area networks–Audio Video Bridging (AVB) Systems, IEEE 802.1BA-2011, 2011.
- [32] IEEE Standard for Local and metropolitan area networks–Frame Replication and Elimination for Reliability IEEE 802.1CB-2017, 2017.
- [33] R. Wang, R. J. Black, B. Moslehi, A. R. Behbahani, and B. Mukherjee, "Optical Control Network for Avionics Applications Using a WDM Packet Ring," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 1, pp. 637–648, 2014.
- [34] Octal 10/100BASE-TX Ethernet BroadR-Reach Transceiver, Broadcom. Accessed: July 17, 2020. [Online]. Available: <https://docs.broadcom.com/doc/1211168574381>
- [35] Avnu Certification Testing Services website, Interoperability Laboratory – University of New Hampshire. Accessed: July 17, 2020. [Online]. Available: <https://www.iol.unh.edu/testing/switching/avnu>
- [36] Cisco Industrial Ethernet 4000 Series Switches, Cisco Systems, Inc. Accessed: July 17, 2020. [Online]. Available: <https://www.cisco.com/c/en/us/products/collateral/switches/industrial-ethernet-4000-series-switches/datasheet-c78-733058.html>
- [37] TSN-capable CompactRIO Controllers, National Instruments. Accessed: July 17, 2020. [Online]. Available: <http://www.ni.com/pdf/product-flyers/compactrio-controller.pdf>
- [38] Hirschmann BOBCAT Next-Generation Managed Switches, Belden Inc. Accessed: July 18, 2020. [Online]. Available: <https://www.belden.com/products/industrial/networking/managed-switches/bobcat>
- [39] M.C. Pastor-Morales *et al.*, "SINBAD flight software, the on-board software of NOMAD in ExoMars 2016," in *Software and Cyberinfrastructure for Astronomy IV*, in *SPIE*, vol. 9913, pp. 1261 – 1274, 2016.
- [40] M. Focardi *et al.*, "The design of the Instrument Control Unit and its role within the Data Processing System of the ESA PLATO Mission," in *Space Telescopes Instrum. 2018: Opt., Infrared, Millimeter Wave*, in *SPIE*, vol. 10698, pp. 1334 – 1345, 2018.
- [41] P. Barthol *et al.*, "The Sunrise Mission," *Sol. Phys.*, vol. 268, no. 1, pp. 1–34, Nov. 2011.

- [42] A. Petersén, “Implementation aspects of TTEthernet Interfaces,” in *7th ESA Workshop ADCSS*, Noordwijk, The Netherlands, Oct. 22-24, 2013. [Online]. Available: https://indico.esa.int/event/22/contributions/2011/attachments/1698/1990/6.ADCSS_2013_-_Implementation_aspects_of_TTEthernet_Interfaces_P-FLPP-HO_-_1108586_-_RSE_-_1_-_1.pdf.
- [43] A. Sheikh, O. Brun, M. Chéramy, and P.E. Hladik, “Optimal design of virtual links in AFDX networks,” *Real-Time Syst.*, vol. 49, no. 3, pp. 308–336, 2013.
- [44] R.L. Alena, J.P. Ossentfort, K.I. Laws, A. Goforth, and F. Figueroa, “Communications for Integrated Modular Avionics,” in *IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 3-10, 2007, pp. 1–18.
- [45] L. Medina, M. Melara, and L. Cercós, “An IPCORE for Deterministic Ethernet via Time Sensitive Networking (TSN) light implementation: challenges and opportunities,” in *13th ESA Workshop ADCSS*, Noordwijk, The Netherlands, Nov. 12-14, 2019. [Online]. Available: https://indico.esa.int/event/323/contributions/5043/attachments/3745/5201/12.30_-_An_IPCORE_for_Deterministic_Ethernet_via_TSN_....pdf.
- [46] J. Sanchez-Garrido, “Time-sensitive networks based on ultra-accurate synchronization mechanisms,” Ph.D. dissertation, Dept. Comp. Arch. Tech., Univ. Granada, Granada, Spain, (submitted in Nov. 2020).
- [47] IC-317x Industrial Controller with Reconfigurable I/O. User Manual, National Instruments. Accessed: May 27, 2020. [Online]. Available: <http://www.ni.com/pdf/manuals/375285b.pdf>
- [48] M. Melara, C. Domínguez, L. Cercós, G. Ramírez, R. Rodríguez, E. González, J. Bru, and J.P. Préaud, “MIURA 1: Data Handling System,” presented at *Int. Space Syst. Conf. DASIA*, Torremolinos, Spain, June 4-6, 2019.
- [49] B. Annighöfer and E. Kleemann, “Large-Scale Model-Based Avionics Architecture Optimization Methods and Case Study,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 6, pp. 3424–3441, 2019.
- [50] Zynq-7000 SoC Data Sheet: Overview, Xilinx, July, 2018. Accessed: July 18, 2020. [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf.
- [51] J. Sanchez-Garrido, A. Jurado, L. Medina, R. Rodriguez, E. Ros, and J. Diaz, “Digital Electrical Substation Communications Based on Deterministic Time-Sensitive Networking Over Ethernet,” *IEEE Access*, vol. 8, pp. 93621–93634, May, 2020.
- [52] Xilinx DMA Controller Documentation, Xilinx. Accessed: July 18, 2020. [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/axi_dma/v7_1/pg021_axi_dma.pdf
- [53] Alex Forencich. *Verilog Ethernet Components. A collection of 1G, 10G, 25G Packet Processing Data Paths*. Accessed: Jan. 8, 2020. [Online]. Available: <https://gitlab.com/alex.forencich/verilog-ethernet>
- [54] 100M/1G TSN Subsystem, Xilinx. Accessed: July 9, 2020. [Online]. Available: <https://www.xilinx.com/products/intellectual-property/1gtsn.html>
- [55] The RTEMS Project. *RTEMS Real-Time Operating System (RTOS)*. Accessed: Jan. 8, 2020. [Online]. Available: <https://www.rtems.org/>
- [56] Avnu Alliance. *OpenAvnu Project: an Avnu sponsored repository for Audio/Video Bridging and Time Sensitive Networking technology*. Accessed: July 18, 2020. [Online]. Available: <http://avnu.github.io/OpenAvnu/>
- [57] M. Amrbar, F. Irom, S. M. Guertin, and G. Allen, “Heavy Ion Single Event Effects Measurements of Xilinx Zynq-7000 FPGA,” in *2015 IEEE REDW*, Boston, MA, USA, July 13-17, 2015, pp. 1–4.
- [58] N. Moreira, J. Lázaro, U. Bidarte, J. Jimenez, and A. Astarloa, “On the Utilization of System-on-Chip Platforms to Achieve Nanosecond Synchronization Accuracies in Substation Automation Systems,” *IEEE Trans. Smart Grid*, vol. 8, pp. 1932–1942, July, 2017.
- [59] 53230A RF Counter. Brochure for the 53200 Series of RF and Universal Frequency Counter/Timers, Keysight. Accessed: July 18, 2020. [Online]. Available: <https://www.keysight.com/us/en/assets/7018-02654/technical-overviews/5990-6339.pdf>
- [60] WR-ZEN TP Product Site, Seven Solutions. Accessed: July 18, 2020. [Online]. Available: <https://sevensols.com/index.php/products/wr-zen-tp/>

- [61] M. Hu, J. Luo, Y. Wang, and B. Veeravalli, "Scheduling periodic task graphs for safety-critical time-triggered avionic systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 3, pp. 2294–2304, 2015.
- [62] A. A. da Fontoura, F. A. M. do Nascimento, S. Nadjm-Tehrani, and E. P. de Freitas, "Timing Assurance of Avionic Reconfiguration Schemes Using Formal Analysis," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 1, pp. 95–106, 2020.
- [63] R. B. Atitallah, V. Viswanathan, N. Belanger, and J. Dekeyser, "FPGA-Centric Design Process for Avionic Simulation and Test," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 3, pp. 1047–1065, 2018.
- [64] S. Janković, A. Smiljanić, M. Vesović, H. Redžović, M. Bežulj, A. Radošević, and S. Moro, "High-capacity FPGA Router for Satellite Backbone Network," *IEEE Trans. Aerosp. Electron. Syst.*, 2019, 10.1109/TAES.2019.2951187.

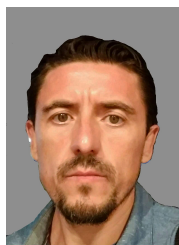


industry.

J. Sanchez-Garrido. Jorge Sánchez Garrido received his B.Sc. in Telecommunications Engineering in 2013 and his Master's Degree in Data Science and Computer Architecture in 2016, both from the University of Granada (Spain). He previously worked as a consultant at Deloitte Spain in Madrid (2014-2015), and then started a Ph.D. Programme on Time-Sensitive Networking in 2015 at the University of Granada with interests in White Rabbit timing, deterministic networks, Smart Grid, and FPGA-based systems design. He has been an active researcher in the EU FITOPTIVIS project and in several transfer projects into the



B. Aparicio. Beatriz Aparicio received her electronic engineering B.Sc. in 2002 from the University of Granada (Spain). She joined the Institute of Astrophysics of Andalusia in 2003, where she has participated in ESA/NASA missions as senior FPGA architect and contributed to the design of the NOMAD instrument (Exomars), the SOPhi instrument (Solar Orbiter) or the IMAX magnetograph (Sunrise). Her main interests are aerospace systems, fault-tolerance techniques, FPGAs, instrument control and deterministic communication buses.



J.G. Ramírez. José Gabriel Ramírez obtained his degree in Electronics Engineering in September 1999 at the University of Granada (Spain). From May 2000 through September 2007, he worked as a Hardware Engineer at Tecnobit S.L. on projects related to infrared cameras for military applications (F18, Eurofighter). Since October 2007, he has been working at Seven Solutions as a Hardware Engineer with the Research and Development division, where he is also the Project Manager and the Head of the Hardware Department. He has worked on numerous design projects for scientific facilities or institutions, such as CERN, GSI, CTA, SKA, IFMIF-EVEDA, SARAF, the University of Granada, or the Institute of Astrophysics of Andalusia.



R. Rodriguez. Rafael Rodríguez became a Computer Engineer in 2005 and worked as a researcher on different EU projects through 2010, while gaining experience in FPGA computer systems. Cofounder and CTO of Seven Solutions, where he was appointed senior Engineer in 2011. He has been Involved in system designs for projects such as ExoMars, particle accelerators (CERN), radioastronomy (SKA), Metrology Institutes (UK). Specialized in timing and open embedded platforms.



M. Melara. Mariasole Melara is a Telecommunication Engineer graduated at the University of Naples Federico II in Italy who has always worked in the aerospace sector. She previously worked at ELV/AVIO (2009-2014), where she was technical responsible of all telemetry systems and power systems (batteries) for VEGA launcher and Avionics Design Authority for First and Second flight during the launch campaigns in French Guyana. At OHB (2014-2016), she was appointed as the Electrical Architect of the Exomars Carrier Module, for which OHB is the Prime Contractor. She is currently the Head of the Avionics and On-Board Software Division in the Space Segment and Robotics Business Unit of GMV and she is also the project manager of GMV avionics system for microlaunchers.



L. Cercós. Lorenzo Cercós received his M.Sc. in Physics from the Complutensian University of Madrid in 2009. He has been the Head of Hardware at the Space Segment and Robotics Business Unit of GMV Aerospace and Defense SAU since 2012. Previously, he had been managing the Attitude Control Subsystem of small satellites at the National Institute of Aerospace Technology of Spain for three years. Currently, he is technical leader of the avionics for the Miura 1 sounding rocket and project manager of the Gigabit Ethernet TSN Deterministic Network (GETDEN) ESA project.



E. Ros. Eduardo Ros received his M.Sc. and Ph.D. degrees in Physics from the U. Granada (Spain) in 1992 and 1997 respectively. He is currently Full Professor in the Dept. of Computer Architecture and Technology of U. Granada. He has participated in many international projects as PI (such as SENSOPAC, FITOPTIVIS, etc). He has authored more than 90 publications in his main areas of interest, including spiking neural networks, White Rabbit timing, or computational neuroscience.



J. Diaz. Javier Díaz received his Ph.D. in electronics in 2006 from the University of Granada, where he combines his academic work as Full Professor with transfer activities as cofounder of Seven Solutions (www.sevensols.com). His main interests are image processing, safety-critical systems, time synchronization and frequency distribution. He collaborates with multiple facilities (CERN, CTA, ...) researching on deterministic communications and White-Rabbit technology.

LIST OF FIGURE CAPTIONS

As indicated in the instructions for the submission, we provide a list of the contents of the figure captions below.

- Fig. 1.** Topology of the TSN network deployed to implement the avionics system of the Miura 1 microlauncher.
- Fig. 2.** General Miura 1 avionics nodes architecture diagram highlighting its RTEMS OS software elements on the ARM PS and its FPGA subcomponents for TSN networking.
- Fig. 3.** The embedded avionics node of the Miura launcher.
- Fig. 4.** Assessment of the gPTP synchronization efficiency using a ring topology configuration. *Fig. 4a* shows the experimental setup, whereas *Figs. 4b and 4c* show the attained accuracy between the master device *ID8* and the slave devices *ID54* and *ID48*, respectively.
- Fig. 5.** Robustness of the gPTP synchronization in the thermal chamber experiments. *Fig. 5a* shows the experimental setup and *Fig. 5b* contains the test results.
- Fig. 6.** Diagram of the laboratory setup for studying the attainable determinism of the TSN avionics network. The diagram shows the probing points at generic TSN talkers and listeners of the signals considered for different experiments.
- Fig. 7.** Reference end-to-end latency graph of a single-hop TSN system, showing that the attainable determinism under ideal conditions sits around $17.49 \mu\text{s}$ with a PDV of 300.97 ns and a standard deviation of 12.23 ns .
- Fig. 8.** Oscilloscope capture showing the application of a simple shaping policy to ensure the reception of TSN frames within a specific time slot. The arrows indicate the reception of TSN messages at the listener node.
- Fig. 9.** Diagram of the simplified test bench for simulating a traffic scenario representative of a suborbital flight mission.
- Fig. 10.** Picture of the laboratory setup for building the avionics demonstrator test bench.
- Fig. 11.** End-to-end latency for the TSN frames of the time-critical flows injected during the experiment. PDV and latency remain stable as interference between *CC* and *Tel* messages is averted by the traffic shapers.