Fusion Engineering and Design 191 (2023) 113774

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

Time-sensitive networking for interlock propagation in the IFMIF-DONES facility

Carlos Megías [a,1,*], Jorge Sánchez-Garrido [b,1], Víctor Vázquez [a], Eduardo Ros [a], Mauro Cappelli [c], Javier Díaz [a]

[a] Department of Computer Architecture and Technology, University of Granada, Granada, Spain
[b] Orolia Spain SLU, Granada, Spain
[c] Italian National Agency for New Technologies, Energy and Sustainable Economic Development, Rome, Italy

ARTICLE INFO

Keywords:
Time-sensitive networking
Interlock
Determinism
Convergent networks
Machine protection system
IFMIF-DONES

ABSTRACT

In this study, we have proposed the use of time-sensitive networking (TSN) technologies for the distribution of the interlock signals of the machine protection system of the future IFMIF-DONES particle accelerator, required for implementing the protection mechanisms of the different systems in the facility. Such facilities usually rely on different fieldbus technologies or direct wiring for their transmission, typically leading to complex network infrastructures and interoperability problems. We provide insights of how TSN could simplify the deployment of the interlock network by aggregating all the traffic under the same network infrastructure, whilst guaranteeing the latency and timing constraints. Since TSN is built on top of Ethernet technology, it also benefits from other network services and all its related developments, including redundancy and bandwidth improvements. The main challenge to address is the transmission of the interlock signals with very low latency between devices located in different points of the facility. We have characterized our initial TSN architecture prototype, evaluated the latency and bandwidth obtained with this solution, identified applications to effectively shape the attainable determinism, and found shortcomings and areas of future improvements.

1. Introduction

An interlock can be defined as a protection mechanism to prevent the malfunctioning of part of a system. It generates signals, the interlock signals, to indicate actions that must be performed when a value has been detected out of its allowed range, or to grant permission for another action once a previous process has taken place. These mechanisms can be found in many different areas such as particle accelerators [1,2] or clinical applications [3,4]. In the case of particle accelerators or industrial plants, the interlocks are an indispensable part of what is often called the machine protection system (MPS) or interlock system, which oversees the protection of the systems of the whole facility. In such facilities, there are protection mechanisms involving multiple systems of the facility that can be located far from each other (intersystem interlocks), requiring the reliable exchange of interlock signals with relatively low latency. They are transmitted through a communications infrastructure which may use different technologies depending on their criticality and requirements. These technologies may lead to the utilization of different protocols and the deployment of more than one physical network, which can incur in high maintenance costs, interoperability and scalability issues, eventually resulting in complex communications infrastructures.

In large facilities with many operating dependencies between systems, the control of the systems of the plant is handled by separating them into central and local systems. Similarly, the interlocks are separated between central and local interlocks [5]. The former corresponds to the intersystem interlocks, which are managed by the central MPS of the facility. The central interlocks involve more than one local system and therefore their implementation must be consistent with the interlock signals coming from and sent to every local system. On the other hand, the local interlocks are concerned with the protection of one specific local system and are managed by their respective local MPS. The local MPSs manage all the interlock signals coming from the sensors and trigger the tasks of the corresponding actuators. These latter ones are almost exclusively designed and implemented by the contributors to that specific local system.

* Corresponding author.
E-mail address: narg@ugr.es (C. Megías).
[1] Both authors have contributed equally to the work.

https://doi.org/10.1016/j.fusengdes.2023.113774
Received 13 October 2022; Received in revised form 30 March 2023; Accepted 20 April 2023
Available online 27 April 2023

Most of the interlocks are implemented using programmable logic controllers (PLCs), field-programmable gate arrays (FPGA)-based devices and hard-wired logic (relay-based logics, dedicated logic cards or cables), which receive the signals coming from sensors as inputs and output the control decisions to the actuators [6]. The selection of one technology or another for an interlock depends on its maximum allowed response time. Moreover, the real-time control (and supervision) of PLCs and FPGA-based devices is achieved by using real-time communication protocols typically based on fieldbuses, such as Profinet and EtherCAT [7]. It may be possible that more than one communication protocol could be needed at the same time for the transmission of interlock signals. For the implementation of central interlocks, the transmission of data from the local MPS to the central MPS through this real-time infrastructure must be performed with low latency, so that the response time deadlines are always met. In the case of local interlocks, this issue can be easily addressed since the number of devices involved is much smaller.

This is shown in our contribution with the use of a time-sensitive networking (TSN) solution tailored to the use case of international fusion materials irradiation facility-DEMO oriented neutron source (IFMIF-DONES) which will be presented in detail throughout the rest of the manuscript. Hence, the current section finishes with the presentation of the motivation and use case of our study. After that, Section 2 analyzes the current design of the interlock network of the plant used for this study and proposes an alternative solution based on the deployment of a TSN system. Next, we have evaluated the attainable performance of our proposal with the experimental setup introduced in Section 3. Finally, we present the conclusions and outline the main areas of future research in Section 4.

1.1. Motivation

The integration scenario of this study is in the framework of the future research infrastructure of IFMIF-DONES; and more specifically it analyzes that of its MPS, which is part of the instrumentation and control (I&C) system. The DONES I&C system coordinates the entire operation of the plant using a two-tiered architecture: the central systems, which receive the name of central instrumentation and control system (CICS), and the local systems, which receive the name of local instrumentation control subsystems (LICSs). The design of this facility is currently under development, and it will be used to evaluate the characteristics of different materials for their utilization in the construction of future fusion reactors [8,9]. The LICSs are responsible for a set of application-related subsystems, and the CICS coordinates their operation and interactions. It consists of three main systems: control data access and communication system (CODAC), machine protection system and safety control system (SCS). A complete description of the three systems and their CICS-LICS interactions can be found in [9–11]. The three systems have their corresponding representation at the CICS and at each LICS by means of different hardware components (e.g., controllers). This approach is similar to that present at the international thermonuclear experimental reactor (ITER) [12].

1.2. Use case

We are working on the design of a unified network infrastructure that can aggregate all types of signals needed to implement the central or intersystem interlocks. This design features the emerging TSN technology, which is based on a set of standards enhancing Ethernet systems [13]. Thus, it aims to simplify the deployment of the network infrastructure of the MPS, whilst meeting all its time constraints.

The MPS system of IFMIF-DONES is based on three different architectures according to the maximum response time of the interlock actions, which are:

- Slow architecture, for relaxed response times: more than 300 milliseconds (ms).
- Fast architecture, for tight response times: ranging from a few ms to 300 ms.
- Hardwired architecture, for the most demanding response times: less than 30 microseconds (μs).

These architectures are used to categorize the different central interlocks (and thus the interlock signals required for their logical operations). Three different system protection modules (SPMs), which are specific modules of the MPS in charge of the intersystem interlocks and events management, implement the different architectures. One of the key points for their implementation is the latency in the propagation, which for some signals is required to be low and to also have low jitter. Apart from the interlock signals, there are also some non-critical data that are transferred from the local MPSs to the central MPS for supervision and representation purposes.

With the utilization of TSN technology, all the signals required for the implementation of the central interlocks can be aggregated over the same network infrastructure. This would guarantee high data bandwidth and determinism (bounded end-to-end latency), as well as low latency. Although there are other solutions than can achieve latencies in the order of the microsecond [14], such as EtherCAT or FOUNDATION fieldbus [15], TSN provides higher bandwidth. Moreover, it would solve the interoperability issues that can be found when different fieldbus technologies are used. Additionally, there is documentation available for the integration of Profinet and EtherCAT with TSN [16,17]. These features allow TSN to simplify and reduce the costs of the deployment of the network by reducing wiring and using one single infrastructure and technology for the distribution of all the interlock signals; and hence effectively become a convergent system. Consequently, the focus of this contribution lies in the study of the mechanism for exchanging the interlock signals between the central and the local MPSs to implement the central interlocks.

2. Design

A block diagram of the current architecture of the MPS network infrastructure and the interactions between the CICS and the LICS levels is depicted in Fig. 1. The top part represents the central MPS which includes a server and the SPMs. The central MPS is based on redundancy of the entire system. The server manages the communications with the CODAC system via the CODAC interface module, which serves as a gateway, and monitors the different interlock events and the operation of the local MPS (exchange of non-interlock signals) using the supervision module. The SPMs are exposed to the interlock network to allow the communication with the local MPSs, located in the bottom part of the diagram. The local MPSs can include different types of controllers depending on the response time of their associated interlocks. The actuators and sensors are finally connected to these controllers.

The current architecture of the MPS considers the possibility of two physically separated networks, one for the interlock signals of the slow architecture and another for those of the fast architecture; and the possible deployment of directly wired connections (cables) from some sensors and actuators of the local MPSs to the hardwired architecture at the central MPS. Depending on the technology used for the slow and fast controllers, their respective networks could be merged into one. This architecture serves as the starting point for the implementation of the TSN-based solution for the interlock network infrastructure.

2.1. Time-sensitive networking solution for interlock signals

The utilization of TSN can be considered a great solution for the distribution of the interlock signals thanks to its four main operation principles: 1) time synchronization, 2) bounded end-to-end latency, 3) configuration and management, and 4) reliability and redundancy [18].

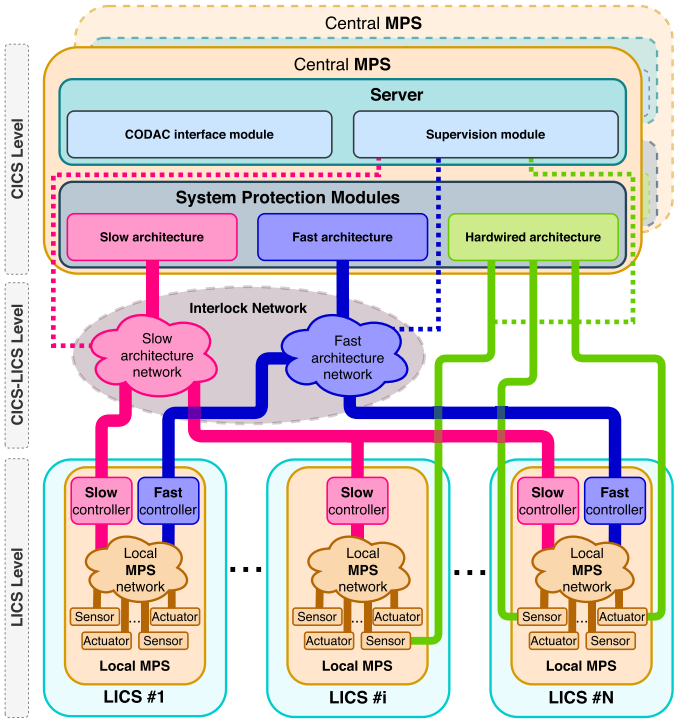


Fig. 1. Block diagram of the current architecture for the machine protection system of IFMIF-DONES. The slow architecture is represented in pink, the fast architecture in dark blue, and the hardwired architecture in light green.

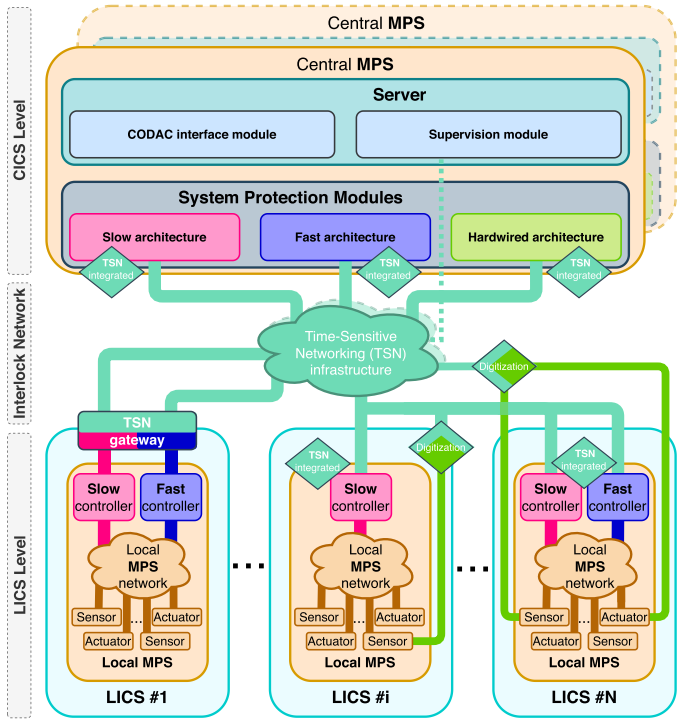


Fig. 2. Block diagram resulting from the utilization of TSN for the network of the machine protection system of IFMIF-DONES. The slow architecture is represented in pink, the fast architecture in dark blue and the hardwired architecture in light green. All parts related to TSN are represented in aquamarine.

The generic precision time protocol (gPTP) profile service derived from the IEEE 1588-2008 standard [19] is used for the distribution of a common time reference to all the nodes (switches and end devices) in the network (802.1AS [20]). The performance of TSN is tied to the synchronization quality of the network. The second principle is achieved by the identification, using virtual local area networks (VLANs) (802.1Q [21]), and prioritized scheduling, by means of a time-aware [traffic] shaper (TAS) component (802.1Qbv [22]), of the different data flows when receiving and transmitting the Ethernet frames. An optimal configuration for each individual node is indispensable to obtain the desired performance (802.1Qcc [23]). Finally, the redundancy and reliability in the communications for data transmission and synchronization distribution is also key for many applications (see Section 2.1.2) [24,25]. Another TSN feature is the capability of interrupting low priority frame transmission when a high priority one arrives. This is known as frame preemption (802.1Qbu [26]) and it increases the determinism. The network architecture resulting from the utilization of TSN for the MPS of IFMIF-DONES is shown in Fig. 2.

In this approach, we leverage the capability of TSN systems to operate as convergent networks capable of aggregating multiple flows with differing bandwidth, end-to-end latency (i.e., deadline), and delivery variation constraints. This would allow for the joint management of the different types of interlock messages (as well as monitoring and non-critical data) that are expected in the MPS network of a particle accelerator through the application of the appropriate cyclic scheduling (gate control lists - GCLs) and forwarding policies of 802.1Q-tagged messages.

The different controllers of the local MPSs may be TSN-compliant (indicated in Fig. 2 as TSN integrated), use a protocol that can be transmitted over TSN (such as Profinet or EtherCAT) or will have to implement protocol gateways to properly interface with the TSN network infrastructure. There are currently commercial solutions (and others under development) that suit this application with TSN support [27,28]. In the case of the hardwired architecture, it is required to digitize (from an analog signal to digital domain and data packet format for TSN, or directly from a digital signal to a data packet format for TSN) the signals to interface the unified network. Nonetheless, the ideal solution for the interfaces of the different architectures at the central MPS is to be fully compliant with TSN.

A first insight of how to perform the prioritization of the different data flows that would traverse the TSN network is shown in Table 1.

2.1.1. *Network traffic control: gate control list*

The prioritization of the data flows configured in the system is mainly considered at the port level, just before the data leaves the network node through the physical link. TSN incorporates the TAS component to schedule the traffic of different priorities awaiting to be sent. This component, whose simplified structure is depicted in Fig. 3(b), uses a gate control list (GCL) to select the traffic priorities that can be transmitted for different intervals of time. Therefore, it can be interpreted as a two-dimensional array, in which the rows represent time intervals (that do not necessarily have the same duration) and each column represents a priority. The matrix is filled with individual bits, used to specify an "*allow*" (1) or "*block*" (0) status to control the transmission of a prioritized flow. An example of a GCL configuration is shown in Fig. 3(a) and its influence on the traffic that is transmitted can be observed in Fig. 3(c).

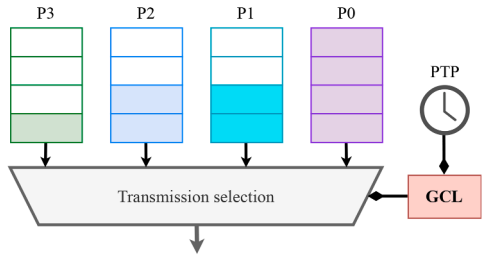
The current implementation affords a maximum of four types of priorities, in ascending order of priorities from right to left. This implies that with a 0b0000 (*0x0*) priority setting, no traffic can be sent. On the other hand, a setting of 0b1111 (*0xf*) means that traffic of all priorities is

Table 1
Example of data flows prioritization of the TSN interlock network.

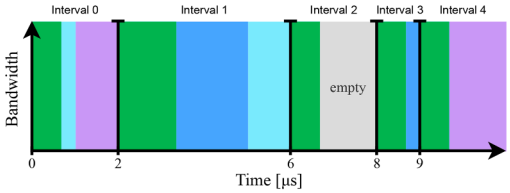
Type of interlock	Priority	Response time
Hardwired	High	$\leq$ 30 µs
Fast	Medium	$\geq$ few ms, $\leq$ 300 ms
Slow	Low	$\geq$ 300 ms
Supervision	Best-effort	$\geq$ 300 ms

Gate Control List (GCL)						
Interval	Duration	P3	P2	P1	P0	Settings
0	2 µs	1	0	1	1	*0xb*
1	4 µs	1	1	1	0	*0xe*
2	2 µs	1	0	0	0	*0x8*
3	1 µs	1	1	0	0	*0xc*
4	2 µs	1	1	1	1	*0xf*

(a) GCL configuration example for the intervals (duration) and gates (priorities - P*k*) control.



(b) Internal structure of the TAS component.



(c) A representation of the resulting data transmission over time (intervals).

Fig. 3. Gate control list (GCL) example.

allowed to be forwarded over the transmission path.

The TAS component multiplexes the different data coming from all the priorities (being stored in dedicated queues) to the port. In the example shown, we have a GCL cycle made up of five intervals for forwarding the traffic held at the different queues of the TAS module. The GCL specifies the duration of each interval and the priorities allowed to transmit their data. Then, a strict priority selection algorithm is used for designating the traffic that will be transmitted if there are multiple queues active during the same interval; i.e., the traffic from the higher priority queue that is allowed in an interval is the first one selected for transmission. Once all its associated data have been transmitted, then the data from the next higher priority queue are selected. This is the approach that we have taken in the experiments presented in the manuscript. Nonetheless, the TAS module is highly flexible and also allows the implementation of complementary transmission mechanisms for flushing the data of select queues, such as the credit-based shaper (CBS) [29], which is particularly suited to multimedia applications. This would potentially allow the use of more advanced shaper modules to fulfill the requirements of diverse applications. This strict prioritization process continues until the end of the time interval and can be observed graphically in Fig. 3(c). Hence, if we take the "*interval 0*" as an example, only the data from priorities P_3, P_1 and P_0 are allowed to be sent. In this case, the first batch of data transmitted belongs to P_3 (green). When its corresponding queue is empty, then the TAS starts transmitting the traffic from P_1 (blue), and then lastly it proceeds to flush the contents of P_0 (purple). An analogous process is repeated for all the intervals in the GCL and, once all of them have been evaluated, the TAS module starts over the execution of the GCL from the first interval ("*interval 0*" in Fig. 3 (a)), thereby repeating the list cyclically.

2.1.2. Redundancy of the system

The robustness of the TSN interlock network is another key element of our proposal. Consequently, the design would rely on the redundant data and timing distribution capabilities of TSN systems to implement a safe and reliable system for the transmission of interlocks. Hence, the reliable distribution of timing in our design would make use of advanced mechanisms that allow the definition of redundant timing paths. These redundant timing paths can be established dynamically through the application of the best master clock algorithm (BMCA), which is built into the implementation of gPTP and allows the designation of fallback timing paths that TSN nodes can fall back on in the event of a failure of the direct communication link with the timing source of the network. Furthermore, we could also explore the latest iteration of the 802.1AS, which takes this concept further by allowing the definition of multiple redundant timing domains within the same system to enhance redundancy.

Redundant data transmissions should also be included in the design of a robust interlock distribution network. In our approach to the design of the interlock system, the use of Ethernet would allow the exploration of multiple alternatives that can be natively supported over Ethernet networks, such as the parallel redundancy protocol (PRP), the high-availability seamless redundancy protocol (HSR [30]), or even the dedicated enhancements of TSN itself for seamless redundancy (802.1CB [31] - frame replication and elimination for reliability [FRER]). Thus, all three foregoing alternatives provide a zero-time switchover and fast recovery capability of the network by establishing redundant data transmission paths. The use of any of the aforementioned techniques could be applicable to the design of an interlock transmission network to provide the expected level of availability and reliability of these types of systems. Nonetheless, we envision that the interlock system of our design will probably be based on the native redundancy of TSN systems (FRER), which would allow for the definition of redundant transmission paths for user-designated data streams over VLAN-tagged frames. Thus, unlike other more complex protocols for replication such as HSR, which would imply the definition of redundant transmission interfaces with specialized hardware for all the traffic in the system, we believe that the use of the extensions for FRER would simplify the design of the network and the management of the redundant flows, as the features for redundant transmission would only be enabled for select user-designated flows. This would in turn allow for the integration of all the interlocks of the accelerator into an Ethernet network that is entirely based on the main standardized mechanisms for TSN systems for traffic forwarding and timing distribution: 802.1AS, 802.1Qbv, 802.Qbu & 802.3br [32], 802.1CB. In addition, the use of advanced configuration methodologies, such as the centralized orchestration paradigm for TSN systems defined in 802.1Qcc, could also be considered to allow the seamless, centralized management of all the network nodes in the TSN-based interlock distribution system.

2.2. Aggregation of additional data

In addition to the unification of all the interlock signals under the same network infrastructure, the utilization of Ethernet-based technologies (such as TSN) for the core of the interlock network allows the introduction of additional network services. One example could be its utilization for the distribution of the synchronization data from protocols such as network time protocol (NTP) [33] or PTP. Moreover, since precise time synchronization is needed for the correct operation of TSN, it may be considered natural to use the interlock network for this purpose, which could provide an accuracy in the order of nanoseconds (or tens of them).

We could also take advantage of the great capability of TSN technology for handling different criticality levels to integrate the data exchanged between the central and local systems of CODAC and SCS seamlessly over the presented TSN infrastructure for the interlock network. These considerations could eventually lead to a fully convergent system for the joint handling of timing, control, data acquisition, facility safety, and machine protection over one fully convergent TSN network that streamlines the management and unifies the deployment of

the critical network infrastructure of scientific facilities for high-energy physics.

3. Experimental proof of concept

The feasibility of our proposal for this application domain (critical and scientific facilities) has been studied in terms of the latency and determinism of critical interlock signals, as these are the most important qualities of the interlock network. To that end, we have performed several experiments to evaluate the latency experienced by interlock signals under different network conditions and GCL configurations. To do so we have deployed the experimental setup of Fig. 4.

The setup consists of a 3-hop switched network (link) of synchronized TSN nodes in a simple daisy chain topology based on customized versions of the WR-Z16 [34]. These nodes are based on the Zynq-7000 SoCs from Xilinx and implement a TSN architecture developed in our research group that is implemented directly on their FPGA logic. In our experiments, we have used this setup to validate the propagation of highly critical interlock messages and to verify the data aggregation capabilities of the TSN system.

The propagation of interlock signals through the network is modelled using a signal generator that produces a square wave at different rates. This effectively simulates a generic digitized output from a sensor in the system associated with the occurrence of the typical kind of event that would trigger the propagation of an interlock message, e.g., exceeding a safety threshold. The square wave signal is in turn fed into a packet generator module in the FPGA of the first node ($Z16_0$) to trigger the transmission of a high-priority Ethernet frame upon receipt of a rising edge. Hence, this setup allows us to use the rate of the square wave to configure the bandwidth of the high-priority data flow. This could be useful for more accurately simulating the digitization process needed for the propagation of hardwired interlock signals. Likewise, the length of the generated packet can also be configured in the node. Once the interlock messages are produced, their propagation time is measured directly as their flight time over the network. To do this, at the moment of the frame generation, the emitting $Z16_0$ node also produces a signal to indicate this phenomenon and transmits it to a Universal Frequency Counter device, which timestamps its arrival time. Then, after propagating the generated packet through the remaining nodes of the network ($Z16_1$-$Z16_2$-$Z16_3$), a reception signal is produced at the receiving node ($Z16_3$) at the other end and fed into the Counter device as well, which calculates the time interval between the two signals. This result is the end-to-end latency of the packet in the network. In addition to the propagation of the interlock signals, other traffic can be configured to share the same network. This can be done using intellectual property (IP) cores implemented on FPGA or by using additional personal computers (PCs) to send and receive data frames. In our tests, we also leveraged the data aggregation capabilities of the TSN system to transmit best-effort traffic between the two laboratory PCs alongside other higher priority flows.

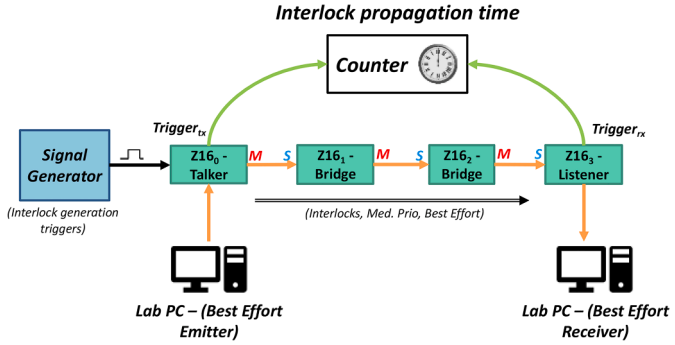


Fig. 4. Block diagram of the experimental setup used for the experiments.

These experiments are combined with a GCL policy, which designates the flows that can be forwarded at a given moment. It is important to remark that the gPTP flow is always allowed through (highest priority). The following settings have also been included in the GCL policies of the experiments:

- 0xc: gPTP and high priority flows are active.
- 0xa: gPTP and medium priority flows are active.
- 0x9: gPTP and low (best-effort) priority flows are active.

3.1. Baseline latency performance

This experiment analyzes the transmission of data packets of different length when no traffic is present in the network. This can be identified with the case when critical packets, associated with a medium to high priority, need to be propagated under ideal network conditions with regular switching and an early version of cut-through (CT) forwarding to reduce the latency. Therefore, this corresponds to the baseline situation, since all the network is dedicated to these critical signals. The GCL can be interpreted as a single fully open time interval, 0b1111 (*0xf*) setting, but without best-effort traffic awaiting to be sent. The results obtained are summarized in Table 2 and represented graphically in Fig. 5.

From these results, we conclude that in order to meet the requirement of 30 μs imposed for the hardwired architecture signals, the length of the frames must be equal or less than 72 bytes (B) (maybe some slightly larger sizes could be admissible). In the case of using regular packet switching, the mean latency and the maximum latency are about 22.3 μs and 25.8 μs. Using the cut-through technique, they improve to 16.3 μs and 20.5 μs, respectively. The peak-to-peak metric (P2P) refers to the difference between the maximum and minimum end-to-end latencies obtained. The 72-B length packets with the regular packet switching mode have been taken as reference for the following experiments related to latency measurements.

3.2. Bandwidth analysis

The actual bandwidth that the current implementation of the system can achieve has been measured using data packets of 1480 bytes at different rates. It has been observed that the desired bandwidth (fixed) and the one measured have a discrepancy of about 5.8% at all bandwidth utilizations, but the design can deliver a performance of up to 90% of the available 1-Gb/s Ethernet interfaces of the TSN nodes.

3.3. Latency analysis under different GCL settings

These experiments aim to analyze the influence that the design of different GCL policies can exert on the end-to-end latency of the more critical flows. To that end, we have simulated the propagation of 72-B interlock messages (P_2-tagged frames) injected at a rate of 1 kHz between the TSN talker and listener nodes in the setup of Fig. 4, where we have iteratively applied the three GCL settings outlined in Table 3 for

Table 2
End-to-end latency obtained for critical packets of different length and switching modes under ideal networks conditions (no traffic) in experiment 3.1.

Size (B)	CT	Max (μs)	Min (μs)	Mean (μs)	P2P (ns)	std (ns)
72	No	25.844	22.140	22.290	3704.033	69.432
	Yes	20.484	16.116	16.266	4368.105	69.612
760	No	54.428	52.476	52.636	1952.056	73.039
	Yes	45.212	42.956	43.116	2256.074	71.705
1480	No	87.108	84.164	84.331	2944.043	71.314
	Yes	73.140	71.036	71.197	2104.048	71.307

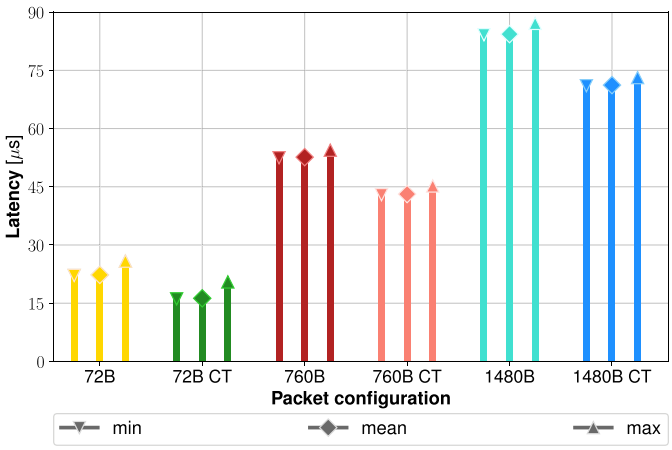


Fig. 5. Latency obtained in terms of minimum, mean and maximum values, for different packet lengths and switching modes (regular and cut-through).

two different scenarios: a baseline test that considers that the system is in idle conditions with no background traffic, and a stress test with interfering best-effort traffic with an approximate rate of 20 Mb/s to determine the effects of clashing best-effort messages on the attainable latency of the critical traffic. The difference between these three GCL configurations lies in the duration of the time intervals. However, the settings of their time intervals are the same: "*interval 0*" has a gate status of *0xc*, with active transmission queues for priorities P_3 and P_2; "*interval 1*" has a gate status of *0xa*, with active transmission queues for priorities P_3 and P_1; and "*interval 2*" has a gate status of *0x9*, with active transmission queues for priorities P_3 and P_0. We have chosen this specific interval structure to show the effects of using progressively shorter interval times for the critical data to effectively reduce their delivery jitter. The results can be examined graphically in Fig. 6, and appear tabulated in Table 4, where we present the outcome of the tests for the idle (*baseline*) and the busy (*stress*) cases of our experiment. Consequently, it can be observed that the main effect of applying different GCL policies with gradually decreasing slot lengths for the forwarding of the critical traffic is to effectively impose a limit on its attainable latency variation. This variation would therefore be directly related to the slot length executed for the current GCL policy and hence shows a way of providing a deterministic data transmission service with tunable performance that can be designed on demand to fulfill the latency variation requirements of different applications and use cases. For instance, the GCL policy of the first iteration of our experiment sets a slot length of 1600 ns, which results in a latency variation of 1752.056 ns (P2P).

This is a promising initial result that speaks to the high configuration capabilities of TSN systems for the handling of interlock messages, although we have also found substantial limitations that will have to be addressed during subsequent stages of the development of our solution. One of them is the handling of interfering traffic, as shown in the results for the busy (*stress*) case of the test, which is affected by large latency swings of up to 5 μs (P2P) caused by a substantial number of critical frames missing their designated slot. Other issues that hold back a tentative deployment of TSN for the transmission of interlocks are related to the existence of minor deviations from the expected linear behavior of the system (e.g., the TAS shaper yields a variation of 1752.056 ns as opposed to the expected 1600 ns in the first iteration) and generic corrections to the logic design to improve stability.

These limitations can be mainly overcome by setting the optimal configuration for the GCL for that specific use case and by improving the implementation of the TSN components. This will allow the TSN capabilities to provide greater bandwidth, similar to that of standard Ethernet, and thus handle these real-time signals, e.g., interlocks, using the same network infrastructure.

3.4. TSN aggregation and integrity validation

This last experiment has the goal of validating the capability of TSN for handling multiple data flows of different criticality, while ensuring the integrity of the most critical ones with respect to those assigned to the lowest priority as a function of the applied GCL settings. To that end,

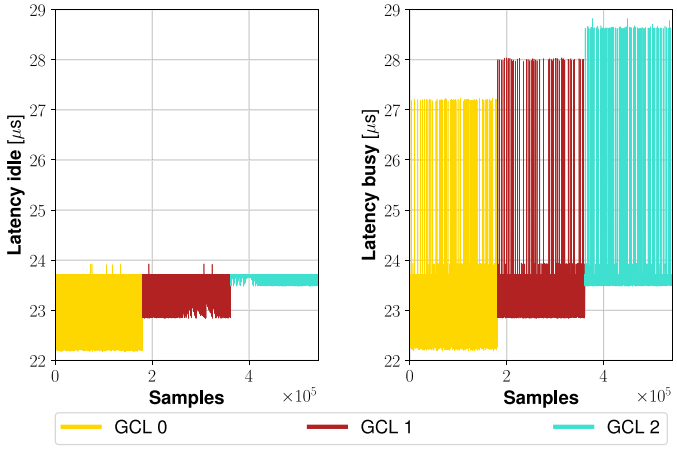


Fig. 6. Behavior of the end-to-end latency variation obtained with the experimental setup as a result of applying different GCL policies for the idle system and stressed system with interfering traffic scenarios.

Table 4
Results of the latency obtained for different GCL configurations and traffic conditions in experiment 3.3.

Traffic	GCL configs.	Max (μs)	Min (μs)	Mean (μs)	P2P (ns)	std (ns)
No	0	23.924	22.172	23.445	1752.056	447.425
	1	23.924	22.844	23.630	1080.063	186.053
	2	23.716	23.484	23.696	232.114	24.291
Yes	0	27.228	22.188	23.447	5040.010	440.653
	1	28.036	22.844	23.625	5192.119	203.081
	2	28.820	23.484	23.691	5336.040	140.302

Table 3
Configurations of the cyclic GCL policies for the different tests of experiment 3.3 to evaluate the latency of critical packets with and without best-effort traffic.

GCL configs.	Cycle time	Interval 0		Interval 1		Interval 2	
		Length (ns)	Settings $\{P_3,P_2,P_1,P_0\}$	Length (ns)	Settings $\{P_3,P_2,P_1,P_0\}$	Length (ns)	Settings $\{P_3,P_2,P_1,P_0\}$
0	4800	1600		1600		1600	
1	4800	800	0b1100 (0xc)	1600	0b1010 (0xa)	2400	0b1001 (0x9)
2	4800	160		1600		3040	

Table 5
Results of the TSN flow aggregation and traffic integrity tests of experiment 3.4 with the applicable GCL settings for each iteration.

GCL configs.	Interval 0		Interval 1		Interval 2		Medium P Traffic (200 kb/s)		Best-Effort Traffic (200 Mb/s)	
	Length (ns)	Settings $\{P_3,P_2,P_1,P_0\}$	Length (ns)	Settings $\{P_3,P_2,P_1,P_0\}$	Length (ns)	Settings $\{P_3,P_2,P_1,P_0\}$	TX (pckts)	Losses (%)	TX (pckts)	Losses (%)
4800	1600	0b1100 (*0xc*)	1600	0b1010 (*0xa*)	1600	0b1001 (*0x9*)	4658	0	99148	0.005
38400	12800		12800		12800		4924	0	107182	1.632
76800	25600		25600		25600		3613	0.028	99712	5.770
85008	28336		28336		28336		3015	0	80913	31.821
90000	30000		30000		30000		3387	0	106912	34.163
153600	51200		51200		51200		3301	0	102647	59.803
307200	102400		102400		102400		3384	0	124252	73.042
3000000	1000000		1000000		1000000		4242	0	114119	96.999

we have applied a GCL policy of three different intervals of the same length, whose duration has been iteratively changed according to what is shown in Table 5.

We have simultaneously transmitted frames of medium priority (P_1-tagged frames of 100 B at 200 kb/s) and best-effort traffic (P_0-tagged frames of 1500 B at 200 Mb/s) and evaluated the forwarding performance of the TSN system in terms of packet losses. The corresponding settings for each iteration of this experiment can be examined in the left half of Table 5. From the results shown in the right part of Table 5, we can conclude that the integrity of the medium priority traffic is maintained at the expense of the best-effort traffic, which suffers severe degradation as the duration of the cycle of the GCL becomes larger since the queues of the TAS shaper have a relatively low capacity of about 4 kB. These results also provide us with valuable insight into how bursts of traffic, e.g., sudden medium-priority monitoring data surges when a safety threshold has been exceeded, can be handled by the system. In these cases, a combination of an optimized GCL policy with the allocation of sufficient buffering capacity at the queues of the TAS module should be able to guarantee the integrity of traffic in cases of fast surges in demand. Specifically, the GCL should be configured to reserve enough bandwidth to handle the worst case of high priority bursts of traffic. As shown in Table 5, the high configurability of the TSN system might allow the user to implement compromise configurations that preserve the critical traffic at the expense of lower-priority messages in resource-constrained designs. Nonetheless, for the case of bursts of critical traffic, the buffering capacity allocation of the architecture and the applicable GCL policy should be designed to accommodate the worst case of traffic demand to guarantee the integrity of the critical data in the network.

4. Conclusion

With this paper, we have proposed a design for the interlock network of IFMIF-DONES using TSN technology. The solution presented leads to a unified network infrastructure, capable of aggregating all types of interlock signals. Additionally, more Ethernet-based network services could use this convergent network, such as generic data and synchronization. This has been shown through different experiments that characterize the end-to-end latency, the bandwidth of the system, the attainable determinism when applying different traffic shaping policies, and the capacity to preserve the integrity of the higher priority flows of the network. Specifically, our solution has obtained latencies of about 22 μs for the forwarding of interlock messages over three hops (16 μs if cut-through is used) under ideal idle conditions. Moreover, the usable bandwidth stays at about 90% of 1 Gb/s, we can effectively apply different traffic shaping policies to reduce latency variation up to 232.114 ns, and we have validated that our system can preserve the integrity of higher priority traffic in the presence of generic data produced at a constant rate of 200 Mb/s. Thus, we have identified that TSN could potentially be considered a robust alternative to support the communications interfaces that are commonly found in particle accelerators.

Overall, this contribution has presented a preliminary study where we have validated our initial prototype of a TSN system that could tentatively be used in this type of scenarios. Furthermore, we have identified areas of improvement that will allow us to deliver a fully functional solution in upcoming future studies, such as latency management and optimization when dealing with highly congested networks.

Lastly, we believe that these results could pave the way to produce higher performing integrations of TSN systems in the future to support the stringent communication needs of major scientific facilities, such as particle accelerators or other applications for high-energy physics.

Author contribution

Antonio Javier Diaz Alonso reports a relationship with Seven Solutions S.L. that includes: board membership, consulting or advisory, and equity or stocks. Eduardo Ros Vidal reports a relationship with Seven Solutions S.L. that includes: board membership, consulting or advisory, and equity or stocks. Jorge Sánchez Garrido reports a relationship with Orolia Spain SLU: full-time employment.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Data availability

Data will be made available on request.

Acknowledgments

This work was supported partially by the Amiga-7 Grant (RTI2018-096228-B-C32), partially by the Programa Operativo FEDER 2014-2020/Junta de Andalucía SINPA Grant (SINPA B-TIC-445-UGR18), partially by the EU DAIS Project (No. 101007273-2 within the ECSEL Calls), partially by the Formación de Profesorado Universitario (FPU) Ph.D. Programme Grants: FPU20/01857 and FPU20/05842, and partially by the Misiones CDTI 2021 framework (DONES-EVO – Grant No MIG-20211006). This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

References

[1] H. Boukabache, M. Pangallo, G. Ducos, N. Cardines, A. Bellotta, C. Toner, D. Perrin, D. Forkel-Wirth, Towards a novel modular architecture for cern radiation monitoring, Radiat. Prot. Dosimetry 173 (2017), https://doi.org/10.1093/rpd/ncw308.

[2] D. Makowski, A. Mielczarek, P. Perek, A. Napieralski, L. Butkowski, J. Branlard, M. Fenner, H. Schlarb, B. Yang, High-Speed Data Processing Module for LLRF, IEEE Trans. Nucl. Sci. 62 (2015), https://doi.org/10.1109/TNS.2015.2416120.

[3] S. Mori, Y. Sakata, R. Hirai, W. Furuichi, K. Shimabukuro, R. Kohno, W.S. Koom, S. Kasai, K. Okaya, Y. Iseki, Commissioning of a fluoroscopic-based real-time markerless tumor tracking system in a superconducting rotating gantry for carbon-ion pencil beam scanning treatment, Med. Phys. 46 (2019), https://doi.org/10.1002/mp.13403.

[4] S. Giordanengo, M.A. Garella, F. Marchetto, F. Bourhaleb, M. Ciocca, A. Mirandola, V. Monaco, M.A. Hosseini, C. Peroni, R. Sacchi, R. Cirio, M. Donetti, The CNAO dose delivery system for modulated scanning ion beam radiotherapy, Med. Phys. 42 (2015), https://doi.org/10.1118/1.4903276.

[5] J.L. Fernández-Hernando, D. Carrillo, G. Ciusa, Y. Liu, I. Prieto-Díaz, R. Pedica, S. Sayas, J. Soni, A. Vergara, The ITER interlock system, Fusion Eng. Design 129 (2018) 104–108, https://doi.org/10.1016/j.fusengdes.2018.02.059.

[6] T. Hakulinen, F. Havart, P. Ninin, F. Valentini CERN, Building an interlock: comparison of technologies for constructing safety interlocks, (2015).

[7] IEC 61158-1:2019 Industrial communication networks - Fieldbus specifications - Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series, (n.d.). https://webstore.iec.ch/publication/59890 (accessed October 11, 2022).

[8] D. Stork, R. Heidinger, T. Muroga, S.J. Zinkle, A. Moeslang, M. Porton, J. L. Boutard, S. Gonzalez, A. Ibarra, Towards a programme of testing and qualification for structural and plasma-facing materials in “fusion neutron” environments, Nuclear Fusion 57 (2017), https://doi.org/10.1088/1741-4326/aa60af.

[9] M. Cappelli, C. Centioli, C. Neri, C. Monti, A. Ibarra, IFMIF-DONES Central instrumentation and control systems: general overview, Fusion Eng. Design 146 (2019) 2682–2686, https://doi.org/10.1016/j.fusengdes.2019.04.084.

[10] M. Cappelli, A. Bagnasco, J. Diaz, J. Sousa, F. Ambi, A. Campedrer, D. Liuzza, B. Carvalho, A. Ibarra, Status of the engineering design of the IFMIF-DONES Central Instrumentation and Control Systems, Fusion Eng. Design 170 (2021), https://doi.org/10.1016/j.fusengdes.2021.112674.

[11] M. Cappelli, A. Bagnasco, A. Ibarra, Preliminary engineering design of the central instrumentation and control systems for the IFMIF-DONES plant, (2019). https://doi.org/10.18429/JACoW-ICALEPCS2019-FRAPP02.

[12] W. Davis, A. Wallander, I. Yonekawa, Current status of ITER I&C system as integration begins, Fusion Eng. Design 112 (2016) 788–795, https://doi.org/10.1016/J.FUSENGDES.2016.04.017.

[13] Time-Sensitive Networking (TSN) Task Group, (n.d.). https://1.ieee802.org/tsn/ (accessed May 30, 2021).

[14] er-soft, Fieldbus Comparison Chart, 2022 n.d. http://www.er-soft.com/files/media/files/ER-Soft–Fieldbus–Comparison–Chart.pdf. accessed October 11

[15] FOUNDATION Fieldbus System Engineering Guidelines, (n.d.). https://www.fieldcommgroup.org/sites/default/files/imce_files/technology/documents/Ff_system_engineering_guidelines_version_3.2.1.pdf (accessed October 11, 2022).

[16] Roadmap of PROFINET over TSN, English. PROFIBUS Nutzerorganisation e.V. (2022) n.d. https://www.profibus.com/technology/industrie-40/profinet-over-tsn. accessed October 1

[17] EtherCAT TSN Communication Profile – project site. English. EtherCAT Technology Group, (n.d.). https://www.ethercat.org/en/downloads/downloads_254672A7CED54910B655F565B974F5AD.htm (accessed October 1, 2022).

[18] J. Sanchez-garrido, B. Aparicio, J. Gabriel Ramírez Rafael Rodriguez, E. Ros Javier Diaz, J. Sanchez-Garrido, E. Ros, J. Gabriel Ramírez, R. Rodriguez, Implementation of a Time-Sensitive Networking (TSN) Ethernet Bus for Microlaunchers; Implementation of a Time-Sensitive Networking (TSN) Ethernet Bus for Microlaunchers, IEEE Trans. Aerosp. Electron. Syst.. 57 (2021). https://doi.org/10.1109/TAES.2021.3061806.

[19] IEEE, 1588-2019 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems | IEEE Standard | IEEE Xplore, (n.d.). https://ieeexplore.ieee.org/document/9120376 (accessed May 31, 2021).

[20] IEEE, 802.1AS-2020 - IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications, IEEE Standard | IEEE Xplore (2021) n.d. https://ieeexplore.ieee.org/document/9121845. accessed May 27

[21] IEEE, 802.1Q-2018 - IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks - Redline | IEEE Standard, IEEE Xplore, 2021 (n.d.), https://ieeexplore.ieee.org/document/8686439. accessed May 31.

[22] IEEE, 802.1Qbv-2015 - IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic | IEEE Standard, IEEE Xplore, 2021 n.d. https://ieeexplore.ieee.org/document/8613095. accessed May 26

[23] IEEE, 802.1Qcc-2018 - IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks – Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements | IEEE Standard, IEEE Xplore, 2021 (n.d.), https://ieeexplore.ieee.org/document/8514112. accessed May 31.

[24] IEC/IEEE, 60802 TSN Profile for Industrial Automation, 2022 n.d. https://1.ieee802.org/tsn/iec-ieee-60802/. accessed October 11

[25] P802.1DP, TSN for Aerospace Onboard Ethernet Communications |, 2022 n.d. https://1.ieee802.org/tsn/802-1dp/. accessed October 11

[26] IEEE, 802.1Qbu-2016 - IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks – Amendment 26: Frame Preemption | IEEE Standard, IEEE Xplore, 2021 n.d. https://ieeexplore.ieee.org/document/7553415. accessed May 31

[27] Time Sensitive Networking (TSN) Frequently Asked Questions. National Instruments., (n.d.). https://www.ni.com/es-es/innovations/white-papers/18/time-sensitive-networking–tsn–frequently-asked-questions.html (accessed October 6, 2022).

[28] TSN – Time-Sensitive Networking. Siemens, (n.d.). https://new.siemens.com/global/en/products/automation/industrial-communication/industrial-ethernet/tsn.html (accessed October 6, 2022).

[29] IEEE, 802.1Qav-2009 - IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams | IEEE Standard, IEEE Xplore, 2023 n.d. https://ieeexplore.ieee.org/document/5375704. accessed March 29

[30] H. Heine, O. Kleineberg, The high-availability Seamless redundancy protocol (HSR): Robust fault-tolerant networking and loop prevention through duplicate discard, in: IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS, 2012, pp. 213–222, https://doi.org/10.1109/WFCS.2012.6242569.

[31] IEEE, 802.1CB-2017 - IEEE Standard for Local and metropolitan area networks–Frame Replication and Elimination for Reliability | IEEE Standard, IEEE Xplore, 2021 n.d. https://ieeexplore.ieee.org/document/8091139. accessed May 31

[32] 802.3br-2016 - IEEE standard for ethernet. Amendment 5, Specification and management parameters for interspersing express traffic., (2016) 148.

[33] J. Martin, J. Burbank, W. Kasch, P.D.L. Mills, Network Time Protocol Version 4: Protocol and Algorithms Specification, (2010). https://doi.org/10.17487/RFC5905.

[34] WR-Z16, The reliable precise time fan-out for White Rabbit distribution on 1G Ethernet-based networks, Orolia Spain SLU (2022) n.d. https://sevensols.com/wr-z16/. accessed October 11