processes

MDPI

Article

An Effective Communication Prototype for Time-Critical IIoT Manufacturing Factories Using Zero-Loss Redundancy Protocols, Time-Sensitive Networking, and Edge-Computing in an Industry 4.0 Environment

Kahiomba Sonia Kiangala [1,†] and Zenghui Wang [2,*,†]

[1] College of Science, Engineering and Technology (CSET), University of South Africa, Johannesburg 1710, South Africa; 60988568@mylife.unisa.ac.za
[2] Department of Electrical and Mining Engineering, University of South Africa, Johannesburg 1710, South Africa
* Correspondence: wangz@unisa.ac.za
† These authors contributed equally to this work.

Abstract: The Industrial Internet of Things (IIoT), the implementation of IoT in the industrial sector, requires a deterministic, real-time, and low-latency communication response for its time-critical applications. A delayed response in such applications could be life-threatening or result in significant losses for manufacturing plants. Although several measures in the likes of predictive maintenance are being put in place to prevent errors and guarantee high network availability, unforeseen failures of physical components are almost inevitable. Our research contribution is to design an efficient communication prototype, entirely based on internet protocol (IP) that combines state-of-the-art communication computing technologies principles to deliver a more stable industrial communication network. We use time-sensitive networking (TSN) and edge computing to increase the determinism of IIoT networks, and we reduce latency with zero-loss redundancy protocols that ensure the sustainability of IIoT networks with smooth recovery in case of unplanned outages. Combining these technologies altogether brings more effectiveness to communication networks than implementing standalone systems. Our study results develop two experimental IP-based industrial network communication prototypes in an intra-domain transmission scenario: the first one is based on the parallel zero-loss redundancy protocol (PRP) and the second one using the high-availability seamless zero-loss redundancy protocol (HSR). We also highlight the benefits of utilizing our communication prototypes to build robust industrial IP communication networks with high network availability and low latency as opposed to conventional communication networks running on seldom redundancy protocols such as Media Redundancy Protocol (MRP) or Rapid Spanning Tree Protocol (RSTP) with single-point of failure and delayed recovery time. While our two network communication prototypes—HSR and PRP—offer zero-loss recovery time in case of a single network failure, our PRP communication prototype goes a step further by providing an effective redundancy scheme against multiple link failures.

Keywords: Industrial Internet of Things (IIoT); time-critical applications; edge computing; time-sensitive networking (TSN); zero-loss redundancy protocols

Citation: Kiangala, K.S.; Wang, Z. An Effective Communication Prototype for Time-Critical IIoT Manufacturing Factories Using Zero-Loss Redundancy Protocols, Time-Sensitive Networking, and Edge-Computing in an Industry 4.0 Environment. *Processes* **2021**, *9*, 2084. https://doi.org/10.3390/pr9112084

Academic Editor: José Barbosa

Received: 12 October 2021
Accepted: 9 November 2021
Published: 21 November 2021

1. Introduction

The manufacturing industry is diving deep into the current industrial revolution, Industry 4.0 (I40). It represents a new manufacturing age where production processes and systems integrate information technologies (IT) techniques to produce advanced automation systems and self-optimization for smart manufacturing. Smart manufacturing promotes intensive interconnectivity between all manufacturing operations stakeholders

and machinery, resulting in high-volume data exchange. As most manufacturing equipment has relatively small storage and computing abilities, another efficient mechanism should be implemented within the network to process the generated data without hindering the overall system's performance [1]. The cloud-based services for data analytics and repository have been an effective solution to the said limitation bringing several benefits such as condition monitoring, energy optimization, and predictive maintenance [2]. The Internet of Things (IoT) paradigm is mostly the inspiration behind the concept of smart manufacturing widely implemented in the IT world with various applications impacting our daily lives [3,4]. IoT devices are utilized in the health sector to provide health condition monitoring [5] and first health care services reporting [6]. Smart homes are also IoT products; they allow individuals to control and monitor different activities such as adjusting the lights, observing all motions around their homes, and much more [7]. Most IoT devices have boosted intelligence capabilities through their modern communication interfaces, advanced sensing apparatuses, and data analysis features.

When applying the core concept of IoT in the industrial sector, a new standard is born for the industry called the Industrial Internet of Things (IIoT) [8,9]. As its parent root, the IoT, the IIoT offers various improvements for the industrial world, using advanced technologies such as enhanced automation through cloud computing for energy management, manufacturing, and transportation [10]. The IIoT increases the intelligence of traditional industrial production processes and their flexibility, which produces an improved overall system with lower production and maintenance costs in the long run. Data processing and analysis is a crucial component of IIoT. The data present in the network originates from the exchange of information between IIoT components. The data processing occurs in cloud servers, which offer better computational features than IIoT devices themselves [11]. There are various communication protocols for IIoT networks such as standard protocols: message queuing and telemetry transport (MQTT), advanced message queuing protocol version 1.0 (AMQP 1.0), the constrained application protocol (CoAP), radio frequency identification (RFID), and TCP/IP protocols: internet, Modbus TCP, intranet, Wi-Fi [12]. Our research focuses on IIoT communication networks solely based on the Internet Protocol (IP), requiring a uniform standard across all the communication layers [13]. An automated IIoT communication network has several characteristics: a low communication latency, a high network availability, a high amount of data, direct communication between sensors and cloud [14], and plug-and-play functions for modular components. Successfully achieving all these characteristics is only possible through implementing a proper communication system with suitable protocols across the whole communication chain, from higher-end devices like cloud servers to field devices. IIoT devices usually require real-time and deterministic communication with a very low tolerance for latency. IIoT applications involve time-critical systems such as motion control solutions that do not tolerate communication delays [15]. The institute of electrical and electronics engineers (IEEE) is currently addressing this challenge by implementing a new Ethernet technology that combines several standards called time-sensitive networking (TSN) [16]. TSN aims to reach guaranteed data communication with low delay and the least latency in Ethernet communication networks. TSN affects the Ethernet frame transmission and requires a time synchronization all over the network [17].

With the various promises of TSN to improve the quality of IIoT networks, several studies have been conducted to gain the most out of TSN's future implementation. Q. Yu and M. Gu [18] developed a TSN framework for multicast time-critical networks in which, through three phases: preprocessing, schedule synthesis, and post-processing, the system can deal with dynamic data transmission and reduce execution time. M. Vlk et al. (2020) [19] enhanced switch hardware to handle TSN requirements of a strict schedulability for traffic and improved the throughput of time-triggered data. This work maintained the determinism of the TSN standards. H. Zhu et al. (2020) [20] implemented an algorithm to improve the reliability and the accuracy of the clock synchronization system in TSN. Collisions between packets could cause inaccuracies in time slot-based synchronization,

thus the need for an improved system to correct any abnormalities in actual frame transmission. X. Jin et al. (2021) [21] applied joint algorithms to improve the performance of real-time no-wait scheduling due to inappropriate packet fragmentation. Their joint algorithm can improve the performance of the schedulability by 50%. X. Jin et al. (2020) [22] introduced another interesting TSN framework to increase the number of real-time flows an off-the-shelf TSN switch can handle. The default number of flows depends on the size of the schedule table, which is limited to 1024. By relaxing the scheduling rules and dividing the satisfiability modulo theories (SMT) into multiple optimization theories (MOT), they increased the data size a TSN switch could manage. Q. Yu and M. Gu (2020) [18] implemented a method to calculate the end-to-end latency and the worst-case delay between different network nodes. The worst-case delay calculation in a TSN application is critical to determine the appropriate clock synchronization mechanism and the best time-based transmission selection.

Due to the massive amount of data traffic generated by additional IIoT devices and expanding networks, there is also a challenge in IIoT networks: the high bandwidth utilization. Especially when the large sets of information need to travel back and forth from the manufacturing sites to the cloud servers located far from the end-devices, the communication bandwidth and the real-time response are negatively affected. The concept of edge computing [23] is another exciting technology introduced in manufacturing plants to deal with this latest issue. Edge computing provides adequate computational and storage support for IIoT applications at the edge of the network, closer to the field devices, reducing bandwidth, and response time. The edge servers are bridges between the IIoT devices and cloud servers [24,25]. In the area of edge computing, some research has been done to create efficient IIoT factories. F. Prinz et al. (2018) [15] designed a network architecture that incorporates cloud computing and edge computing services in a single factory, providing details on how these two paradigms would work together to produce better results. Q. Qi and F. Tao (2019) [26] built an intelligent framework for IIoT network called IIoT learning by combining edge computing services with some wireless industrial network technologies such as low-power wide-area network (LPWAN). They also integrated smart gateways and sensors accessible via wireless to learn and discover information from various network branches. Pustokhina I.V. et al. (2020) [25] implemented the concept of edge computing to improve deep neural network methods in the analysis and diagnosis of the Internet of Medical things (IoMT) in the health sector. Liao H. et al. (2020) [27] developed an effective manufacturing scheduling system for a smart factory using edge computing. The main advantage of edge computing in the manufacturing scheduling system is to reduce the response time using several edge servers, especially for extensive schedules between pieces of machinery. Gong C. et al. (2020) [28] developed a specialized platform that combines the benefits of Artificial Intelligence (AI) and edge computing to enable IoT functions and tasks such as Quality of Experience (QoE). Carvalho A. et al. (2019) [29] provided interesting insights on the implementation of edge computing to improve the accuracy of machine learning (ML) techniques and artificial intelligence (AI) applications in areas such as face recognition, augmented reality, and reinforcement learning. Chen Y. et al. (2020) [30] introduced a protocol for data transmission in edge computing systems to reserve channels when transmitting information and avoid collisions. The protocol is built based on the MAC layer. It is known as the channel reserved MAC (chRMAC) protocol. This protocol aims to reduce the latency due to packet transmission clashes at the edge computing level.

Network availability remains essential for the operational technology (OT) environment. While TSN and edge computing tends to improve the communication requirement of IIoT applications in the software side of the network, the physical network part remains exposed to unforeseen errors. Faults due to cables and network switch failures, incorrect cabling disconnections are unpredictable and can cause unacceptable network downtime. Network redundancy protocols have responded to physical network errors providing backup transmission channels with the least recovery time whenever faults occur. The redundancy protocol depends on the network topology. Some of the most popular ones

are spanning tree protocol (STP) [31,32], Media Redundancy Protocol (MRP) [33,34], Parallel Redundancy Protocol (PRP) [10], and high-availability seamless redundancy protocol (HSRP) [35]. Some studies have also been conducted in network redundancy to develop more secure communication networks by incorporating various protocols. Wylian S.F. (2020) [36] implements the multiple spanning tree protocol as a protection scheme for link failures in a communication network using virtual local area networks (VLANs). Whenever a fault occurs, the spanning tree protocol reconfigures the data path to available routes. Willis P. et al. (2020) [37] developed an improved protection scheme for a mesh topology network using the spanning tree protocol principle called meshed tree protocol. Their work intends to produce faster recovery time in mesh networks in case of link failures and avoid network loops. Giorgetti A. et al. (2013) [33] tested the performance of the MRP in a ring topology when a network failure happened and compared its operations to RSTP. Xu B. et al. (2021) [38] implemented PRP to enhance a Gas plant's stability and reliability. They created two redundant networks receiving duplicated data from the PRP device to ensure data delivery in case of link failure.

1.1. Motivation of the Study

Network availability is an essential component of the operational technology (OT) of every industrial manufacturing organization. It ensures that field devices and machinery communicate effectively with the least latency. Most applications in OT are time-critical, and a delayed response in them could easily result in significant incidents on pieces of machinery or humans and very high production losses. Under the current trend of automation, I40, characterized by new technologies such as IIoT, where a considerable amount of data is constantly shared between production stakeholders, the need for a reliable communication network becomes even more critical. Our study proposes an effective communication prototype for IP-based industrial manufacturing networks running time-critical applications that ensures low-latency responses between network nodes. On the one hand, our prototype covers the software side of the network by implementing communication technologies like TSN and edge computing developed to bring more determinism into industrial communication networks. On the other hand, we apply zero-loss redundancy protocols to the network's physical layer to prevent communication delays due to failures of physical components and long recovery times.

1.2. Contribution, Assumptions and Limitations of the Study

Our primary research contribution is the design of efficient IP-based industrial communication network prototypes based on the following.

- Two zero-loss redundancy protocols operation principle: PRP, suitable for protection against multiple points of failures, and HSR to palliate to IP-based IIoT networks' unforeseen link failures, therefore reducing the risks of communication delays and downtimes.
- TSN and edge computing concepts benefits to diminish communication latency risks when dealing with critical data transmission.

We make the following assumptions in this study:

- All network components (switches) are TSN-capable devices.
- We assume that inter-domain transmissions for the edge computing are done in the background.

We can summarize our study's limitations as follows.

- Our research intends to develop experimental industrial network communication prototypes solely built on theoretical concepts such as TSN, edge computing, and zero-loss redundancy protocols. We do not conduct deep simulations on these technologies but incorporate their benefits to design a robust and reliable industrial communication network. Our communication prototypes are helpful to plan robust network design before the actual deployment on physical network devices. The concept of TSN

is currently not finalized yet. Deep tests on this technology should happen once completed.
- Our research only focuses on IP-based industrial networks communication.
- Our research is limited to intra-domain transmission.

1.3. Previous Works Gaps Summary

Most previous works done on TSN and edge computing applications to improve IIoT networks' performance have not considered empowering the physical layer of their network devices. Although most of these applications achieve outstanding results regarding frame transmissions and response time, which covers the software area of the network, they did not make provision for system responses in case of physical component failures such as cabling or switches. In real-life networks, these types of faults are usually unforeseen and almost inevitable. Most previous research conducted on network redundancy protocol applications on the physical side of networks has not considered improving the data transmission (software part) by integrating state-of-the-art technologies that enhance frames transmissions or improve node communications. Our research is a response to this gap by designing a communication prototype that addresses network improvement on its software side by implementing technologies like TSN and edge computing; on the physical network side by applying zero-loss redundancy protocols such as PRP and HSR to reduce the risks of network downtime in case of errors at the physical layer.

2. Background and Theory

2.1. Edge Computing

Under the fourth industrial revolution (4IR), production systems and manufacturing processes are intended to be self-optimizing, very responsive, intelligent, and interconnected via a combination with improved manufacturing methods and IIoT [39]. Manufacturing plants and factories will have several machine-type devices (MTDs), carrying out operational chores like billing, monitoring, or protection [40,41]. MTDs are devices capable of making decisions and operating without human intervention. They have an application section, a networking connection, and sensors to make them autonomous [42]. In the transition to an era of IIoT, some legacy hardware and controllers indispensable to factories' operations can be adapted to MTDs by by assigning them to some external devices and software. A good illustration is the use of sensors connected to a programmable logic controller (PLC) to trigger production processes. IoT and IIoT are two neighboring concepts but have different critical requirements as operating in two different spectrums. Some key differentiators between these two notions are the high communication bandwidth required in IIoT applications to transmit big-data in real-time, with reliable connectivity, low jitter, low cost, and low latency that will result in efficient and stable engineering systems [43]. IIoT systems usually deal with critical applications for which uncontrolled transmission delays can generate unsafe conditions for human beings or economic instability. The effective responses and decisions of IIoT applications depend primarily on data analytics, processed at a cloud platform, whose feedback needs to be reliable and timely [44].

As per the work in [1], a summary of some of the critical implications of MTDs and IIoT devices in smart factory networks is listed below.

- Big IIoT data: IIoT devices create massive data which are collected, processed, and stored in the smart factory network. These data can be directly collected from an MTD or an IIoT device; in this case, they are called raw data. Other data present in the network are the raw data processing product to make production processes decisions, take actions, and send information back to IIoT devices.
- Ultra-low-latency response: Most IIoT applications require real-time responses and extreme low-latency for precise decision-making from continuously monitoring IIoT devices data. Individual MTDs and IIoT devices are unable to achieve, on their own, the analysis and the processing of all smart factory data at ultra-low latency.

- Reliable medium: The data processing and collection in an IIoT environment needs to be continuous and uninterrupted to ensure that the production processes deliver excellent results in terms of quality and quantity. Therefore, it is imperative to have a reliable medium through which data is transmitted and processed without unnecessary failure.

In traditional manufacturing systems, utilizing cloud computing technologies, the remote cloud facilities performing data analytics are at very distant remote locations far from end-devices and MTDs. This long-distance cause numerous disadvantages, such as network congestion, non-reliable connection, and unacceptable latencies [45,46]. Edge computing addresses some of these issues by enabling data processing, analytics, and intelligent services for critical data closer to the manufacturing shop floor. It offers networking, computing, and storage abilities for IIoT applications that create agile connections, responsive cloud computing services, data analytics at edge nodes, and privacy strategies [24,47]. Edge computing eases the fulfillment of IIoT promises by moving away from massive computational operations from limited MTDs and far-away clouds to powerful and closer edge facilities. Implementing an edge computing solution supports future-proofed techniques that aim to accommodate a rapidly growing industrial environment [27]. Edge computing therefore becomes an essential technology for speedy real-time control of big data in IIoT [28,48].

Figure 1 is a graphical representation of the edge computing concept illustrating field devices in a manufacturing environment, edge servers closer to field devices, and a cloud server located at the architecture's upper end. It is worth mentioning that the edge servers represented in Figure 1 are under a single domain administration. They all share the same network domain. It is a basic scenario of the edge computing principle. Much more complex scenarios exist where the edge servers are under multi-domain administrations [24].

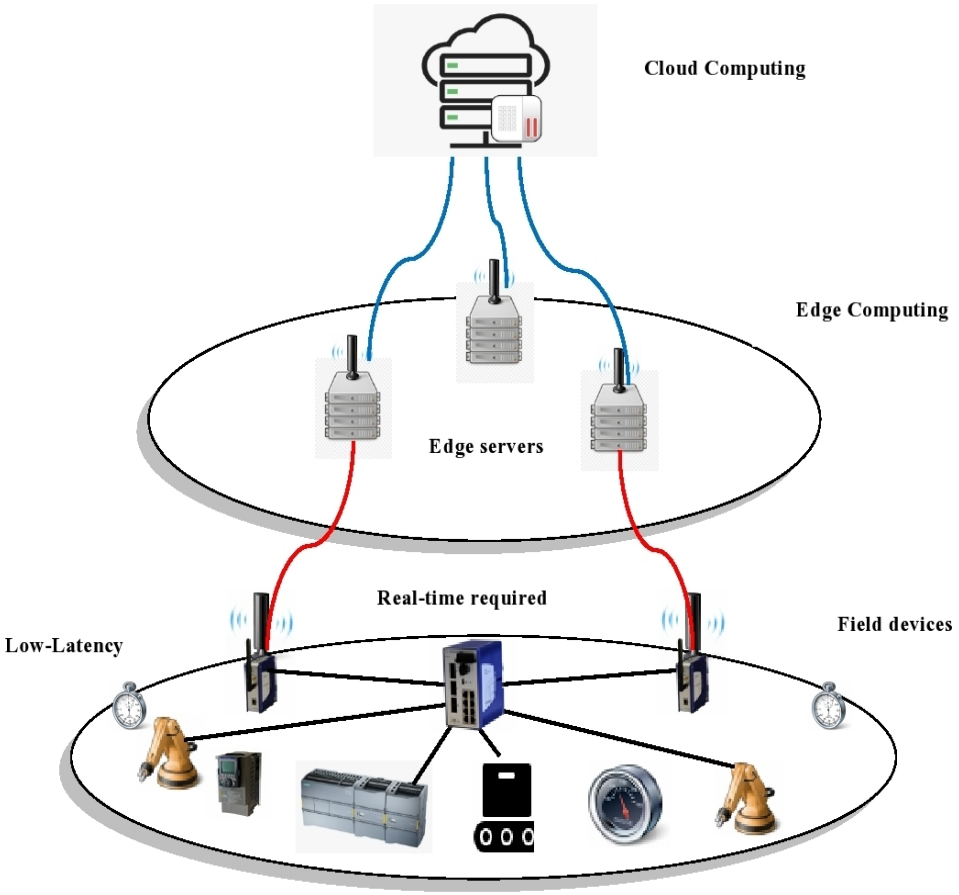

Figure 1. Edge computing architecture.

2.2. Ethernet and Switches

Ethernet is one of the most straightforward networking protocols used to link two or more endpoints at the second layer of the Open Systems Interconnection (OSI) model, the data link layer [49]. Since the 1970s, Ethernet, also known as the IEEE802.3 code, successfully implemented local area networks (LANs) in the office environment. Around the year 2000, several organizations introduced optimizations technologies to the Ethernet protocol to accommodate its use in automation applications requiring more determinism and real-time response than office-based ones [15]. However, the Ethernet optimized solutions generated are all proprietary and cannot allow direct interoperability. They were designed for the same applications but used very different features and details [17].

The Ethernet protocol transmits information through so-called "Ethernet frames". A standard Ethernet frame (Ethernet version 2(v2) and Ethernet IEEE 802.3), whose graphical representation is in Figures 2 and 3, contains various parts with different roles ensuring that the information sent from one end-point is smoothly transmitted to the designated end-point. The main parts of a standard Ethernet frame are as follows:

- The preamble is the first section of the Ethernet frame that synchronizes the recipient of the Ethernet frame. It is usually a sequence of '1 s' and '0 s' in 7 bytes.
- The Start of Frame Delimiter (SFD): This section marks the beginning of a frame as a sequence of '1 s' and '0 s' bits in 1 byte.
- The Destination and The Source address: These two sections save the physical or MAC (media access control) address of each end-device from which and to which the frame is going.
- The type: This section is only available in Ethernet V2.0 frames and indicates the protocol used in the Ethernet frame: IP or UDP.
- The Length: This part is only available in Ethernet IEEE 802.3 frame and indicates the size of the data field.
- The protocol data unit (PDU): This section contains the data is transmitted from one node to another.
- The Frame Checking Sequence (FCS): This part offers a checksum to check errors in the Ethernet frame. Its size is 4 bytes.
- The Inter Frame Gap (IFG): This is a 12 bytes section to mark the minimum space between two frames following each other.

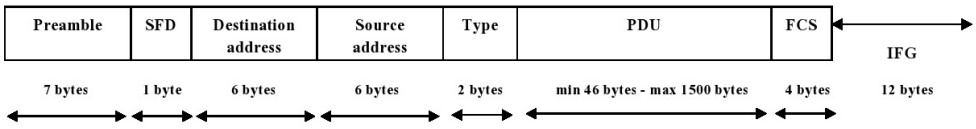

Figure 2. Ethernet v2 frame [50].

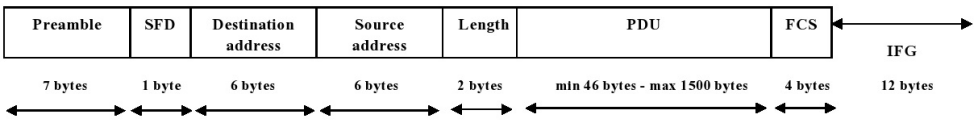

Figure 3. Ethernet IEEE 802.3 frame [50].

The Ethernet frames are transmitted via some layer 2 (of the OSI model) devices called switches. A switch can send frames, analyze incoming ones through the source address, and detect low error via the checksum. The switch uses a forwarding table or an address table to learn the addresses of nodes connected to each port. This process occurs during online operations. The switch can forward frames directly to corresponding nodes by learning peers' addresses without sending them to all ports (causing unnecessary bandwidth reduction). When a new device connects to the switch or does not have the address stored in the forwarding database, the switch floods the frames to every port until it saves the destination address.

Figures 4 and 5 are illustrations of switches forwarding frames between them. They send frames serially from an input port (ingress) to an output port (egress port). If the egress ports are different, frames can be transferred in parallel. When switches have several frames received at once for the same egress port, they store them in their memory until the egress port is available to receive new frames [39].

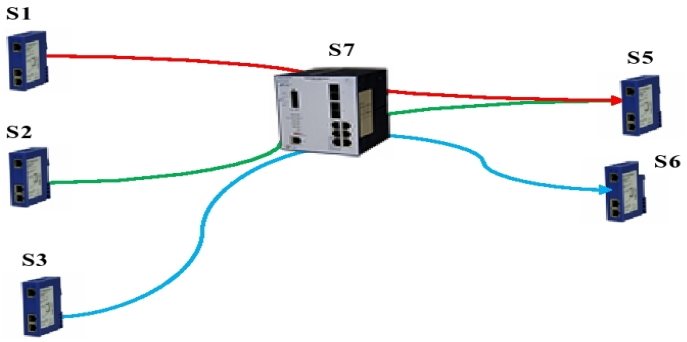

Figure 4. Switching frame sequence part 1.

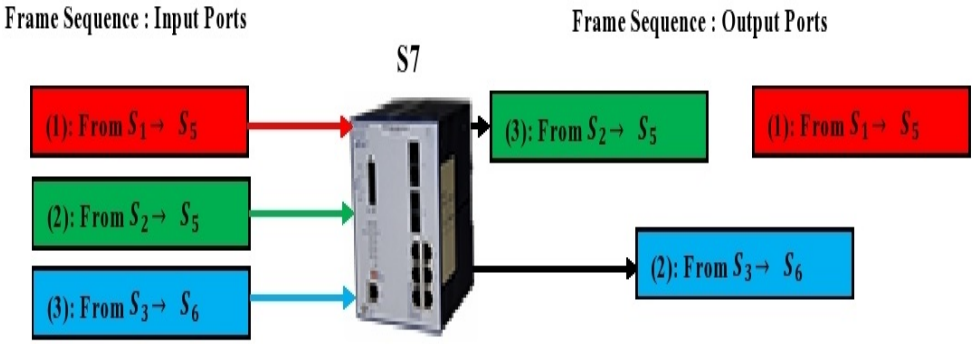

Figure 5. Switching frame sequence part 2.

We summarize frames transmission data paths in Figure 2 as

$$p_1 = S_1 \rightarrow S_7, S_7 \rightarrow S_5 \quad (1)$$

$$p_2 = S_2 \rightarrow S_7, S_7 \rightarrow S_5 \quad (2)$$

$$p_3 = S_3 \rightarrow S_7, S_7 \rightarrow S_6 \quad (3)$$

where p_1 is the first data path identification.

Frames processing and forwarding from one switch to another are unfortunately not instantaneous. Many delays arise when transmitting Ethernet frames. Lee K.C. et al. (2006) [42] examine a few cases of delays in Ethernet communication:

- Assuming that, in a network with two switches only, a frame travels directly from a switch to another without waiting in the source switch memory, we present a mathematical expression for the minimum communication frame delay as

$$\delta_{cmin} = \delta_{src} + \delta_{dst} + 2(\delta_{frm} + \delta_{cbl}) \quad (4)$$

where δ_{src} is the frame processing delay from the source node or switch; δ_{dst} is the processing delay at the destination node; δ_{frm} is the delay generated by the frame transmission. We defined δ_{frm} in (5). δ_{cbl} is the delay issued by the electrical signal traveling through the physical medium (the copper cable or the fiber cable for long distances). At a worst-case scenario, frames are assumed to be transmitted at about $\frac{2}{3}$ the speed of light via a cable. We consider that the length of cables used between the

transmitting node and the two switches is the same. The worst-case delay of δ_{cbl} is therefore approximated to about 5 µs per kilometer or 0.1 µs for 20 m.

$$\delta_{frm} = \frac{\eta}{x} \tag{5}$$

where η is the size of the transmitted frame in bits and x is the data rate in bits per seconds.

- Assuming that the frame will be stored for few times in one of the switches before the transmission to the end-point, the frame communication delay can be expressed as

$$\delta_c = \delta_{cmin} + \delta_{mry} \tag{6}$$

$$\delta_{mry} = \sum_{n=1}^{F_m} IFG + \max(S_n + S_{hd})\frac{1}{x} \tag{7}$$

where F_m is the number of frames waiting in the switch memory, IFG is the inter frame gap, S_n is the data size of the n^{th} frame in the queue, S_{hd} represents the overhead of the frame and x is the data transmission rate in bits per seconds.
The above delay calculations are only applicable for the store and forward switching method that considers the overall size of the frame before its transmission unlike other technologies such as cut-through. As per Gutiérrez C.S.V. et al. (2018) [51], a simplified expression of the delay from an end-point Y to another end-point Z in a network with k number of switches is presented in (8).

$$\delta_{YZ} = \delta_{t1} + \sum_{p=1}^{k}(\delta_{lnkp}) + \sum_{p=1}^{k-1}(\delta_{swtp}) \tag{8}$$

where δ_{t1} is delay to transmit all frames into the link, δ_{lnk} is the delay a frame encounters to travel on each link based on the data rate of the link, and δ_{swt} is the processing delay for a frame to be forwarded from a switch ingress port to its egress port.

2.3. Time-Sensitive Networking (TSN)

Many industrial and automation systems rely on their factory networks' sound operations to transmit safety and time-critical information that commands physical processes. Therefore, these networks must ensure prompt and guaranteed delivery of messages [39]. Ethernet is currently one of the protocols intensively exploited for use in Industrial real-time applications. As previously mentioned, the original creation of the Ethernet had no real-time communication capabilities, but over the years, several methods adjusted its functioning and made its utilization possible within time-critical industrial applications [52]. When adopting Ethernet's use in industrial networks, the main aim was to meet the minimal delay restriction. In this adjustment process, Ethernet's most significant challenges were that delays of Ethernet frames are not deterministic. Various research has been done to illustrate the impact of Ethernet frame delays in industrial applications based on different network configurations [42,53]. As a response to the said research, several approaches have been introduced, resulting in the so-called Industrial Ethernet [52].

TSN is the new Ethernet standard introduced by IEEE to address some of the current Industrial Ethernet shortcomings and meet the IIoT communication requirements in the era of Industry 4.0 [54]. TSN resolves non-reliable communication and high real-time by applying fundamental technologies such as traffic shaping [55], bandwidth reservation, and precise clock synchronization [56,57]. TSN is a set of standards established by the IEEE to warrant low-latency and deterministic communication in Ethernet-compliant networks. Therefore, TSN intends to become a vendor-independent and a standardized networking technology that replaces current exclusive Industrial Ethernet protocols and unifies the traditional automation pyramid [15]. Figure 6 illustrates this concept.

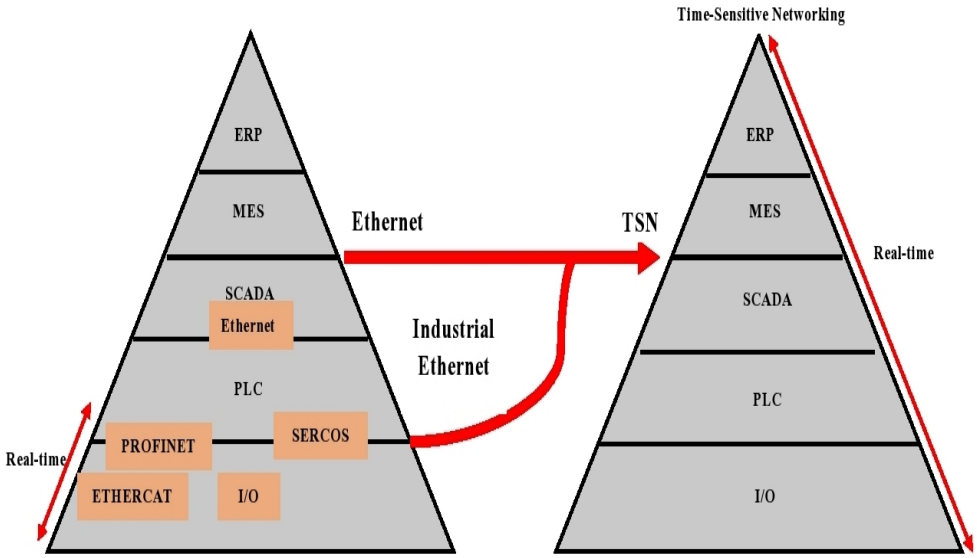

Figure 6. TSN implementation on the automation pyramid [58].

Industrial and automation factories use different proprietary industrial Ethernet protocols such as Profinet, EthernetCAT, Modbus-TCP, Ethernet/IP, and Sercos to ensure real-time communication of their processes. Usually, on the IT side of these factories, they use the standard Ethernet protocol for centralized communication. It becomes compulsory to utilize some gateways at the PLC level to establish communication between the IT and the field level (PLC, sensors, and motors). TSN offers a connection between Industrial Ethernet protocols and standard Ethernet, as displayed in Figure 6 as a standardized communication protocol for the overall automation system [58]. TSN is the name given to the IEEE 802.1 task group (TG) appointed to create several standards that enable network communication to meet low delay variation, low loss, and assured data carriage with very low latency. The TSN TG has finalized a number of those standards, as displayed in Figure 7, but is still working on releasing more in the future [17]. The advantages of TSN depend on the implementation of different tools. As per the work in [39], the traffic shaper is the most appropriate of all these tools. Some ways of reaching low latency with TSN are (i) to give higher priority to time-critical frames so that they stop the transmission of lower priority frames (802.1Qbu and 802.3br), (ii) to choose devoted time slots or windows for the transmission of critical traffic (802.1Qbv), (iii) to split traffic of a switch forwarding mechanism between real-time and non-real-time data regularly (802.1Qch), and (iv) reserve and restrict a network bandwidth to the advantage of time-critical traffic (802.1Qav) [17]. The TSN concept is also incorporated in communication technologies like Wi-Fi to develop transmission systems with extremely low latency [55]. The TSN standards present several research opportunities to be explored for automation applications to improve their performances and operations [59].

Some of the key features of TSN are as follows:

1. Time Synchronization: It is a crucial feature of TSN networks as it allows devices, by clock synchronization, to consistently exchange data at specific time slots.
2. Scheduling and Traffic Shaping: This feature enables different traffic classes and types to operate in one network. The classes have individual bandwidth requirements, priorities, and delays.
3. Stream Management and Fault Tolerance: This feature is about registering, identifying, and managing data paths before the communication begins. It allows us to keep a close look into each communication stream's time accuracy and conduct [57].

The TSN standardization is still ongoing. Some of its standards in the entertainment industry, industrial automation, and automotive communication have already been tested and exhibited outstanding results in determinism instead of legacy standards [60].

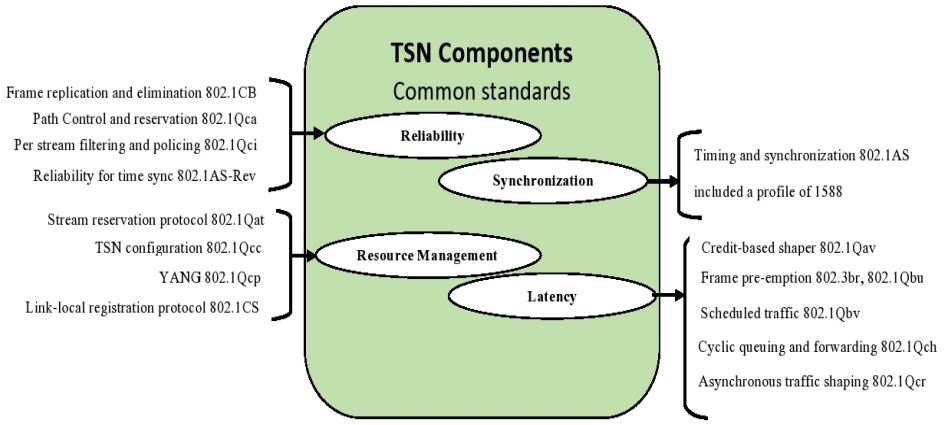

Figure 7. TSN common finalized standards [61].

2.4. Network Redundancy

Network communication plays an essential role in an industrial manufacturing environment. It allows end devices to exchange information and communicate with higher-end devices like controllers and gateways. There are various network topologies from which end-nodes can be connected [60]:

- A bus or line topology: In this network topology, network devices like switches are connected one after another in a line. Figure 8 is an illustration of the bus topology.
- A ring topology: The ring topology is very popular in industrial networks. It can be defined as a bus topology from which the first and last device is connected. A ring topology is presented in Figure 9.
- A mesh topology: In the mesh topology, devices are interconnected through several connections coming back and forth from one device to another. A graphical representation of the mesh topology is presented in Figure 10.
- A star topology: In a star topology, devices are connected to each other via a single switch. It is the preferred topology for the office environment. Figure 11 an illustration of this topology.

Figure 8. Bus network topology.

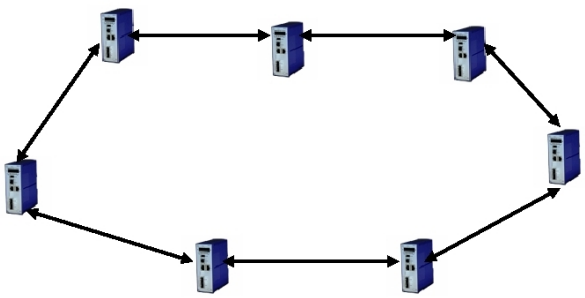

Figure 9. Ring network topology.

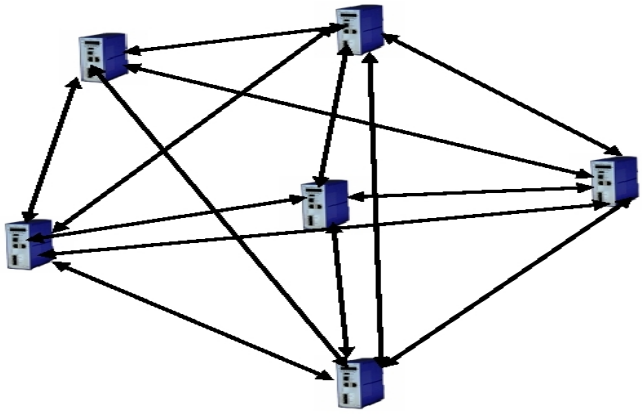

Figure 10. Mesh network topology.

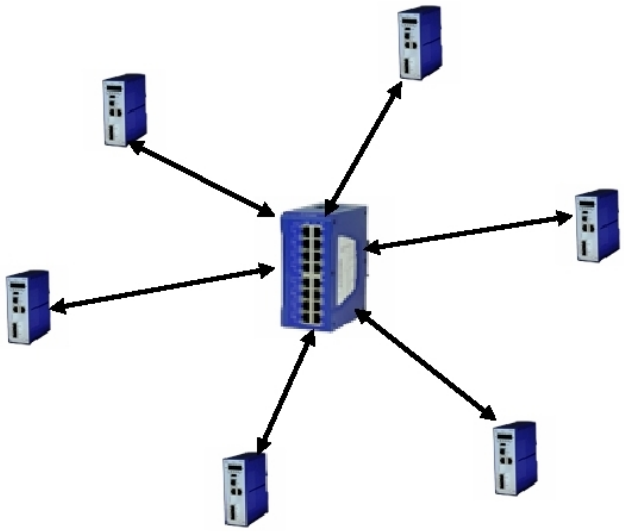

Figure 11. Star network topology.

It is worth mentioning that implementing a ring or mesh topology is impossible without the activation of specialized protocols in network switches to avoid network loops generated by multiple connections between switches. One of the underlying protocols used in this regard is the spanning tree protocol (STP) or the rapid spanning tree (RSTP) [62].

3. Some Industrial Network Redundancy Protocols in Ip-Based Communication Networks

In an industrial networking environment, despite all efforts in place for frequent maintenance on networking equipment, unforeseen software or hardware faults such as cabling failures, disconnection of wrong cables, switch systems, and software crashes can cause the whole network to go down. Network failures produce delays that are unacceptable for critical industrial manufacturing processes. Therefore, several redundancy mechanisms and protocols were created to reduce delays whenever said unpredicted faults occur. The choice of the redundancy mechanism depends solely on the network topology implemented [15].

3.1. Standard Redundancy Protocols

Two of the current popular redundancy protocols in IP-based industrial communication networks are as follows:

- Rapid Spanning Tree Protocol (RSTP): RSTP is an improved version of the spanning tree protocol (STP) implemented for loop prevention in Ethernet networks [37]. RSTP

can be used in various network topologies. It also offers a higher number of network participants than the original STP (of up to forty switches for a ring topology) and a better recovery time of a minimum of one second. However, the recovery time of RSTP, which depends on the positioning of the network switch, can increase to more than one second and is not enough to offer deterministic behavior to critical industrial applications that require highly ultra-low latency.

The STP has another improved variant suitable for implementation in communication networks using virtual local area networks (VLANs). This variant is called Multiple Spanning Tree Protocol (MSTP) [32,36]. It permits the implementation of various spanning trees in different VLANs. The STP and RSTP configurations in industrial network switches are relatively straightforward and differ slightly from one vendor to another [63].

- Media Redundancy Protocol (MRP): MRP is a ring redundancy protocol that provides high availability for Ethernet networks when it comes to network recovery [64]. The protocol supports up to fifty network participants and a recovery time of 500ms for a worst-case scenario. One of the links (redundant link) remains blocked in an MRP ring until there is a fault in the network. Each MRP ring has a switch configured as a manager that monitors the network state and detects failure to activate the redundant link [33]. Although MRP offers perfect recovery time for most industrial network applications, it is still insufficient to satisfy some time-critical applications like specific IIoT applications that do not tolerate any downtime. For such applications, zero-loss redundancy protocols were introduced [65].

Note: In a ring topology, MRP and RSTP can only protect the network against a single link failure [38,65]. In other words, when more than one link fails in the network, there is no guarantee of sound data transmission to all network participants.

3.2. Zero-Loss Redundancy Protocols for IP-Based Industrial Networks

In this research, we focus on two zero-loss redundancy protocols operational principles: PRP and HSR.

- Parallel Redundancy Protocol (PRP): PRP is a protocol (in the International Electrotechnical Commission-IEC 62439-3 standard) developed to achieve zero loss recovery time whenever a failure occurs in a network. A protocol is applied closer to the end-devices while the remaining network switches can use different other protocols. The operational principle of PRP is to send duplicated frames from a PRP capable end-device, called Double-Attached Node supporting PRP (DAN P), or from a Redundancy Box (RedBox), which is a switch that supports the PRP protocol, to the corresponding PRP device which accepts the first frame arriving and discards the second one. The implementation of PRP depends on two independent networks (Local Area Network-LAN A and Local Area Network-LAN B) through which the transmitted frame travels until it reaches its destination. The two independent networks can have a different redundancy protocol, such as RSTP, MRP, or none (single switches connected). By sending duplicated frames in independent networks, the PRP protocol ensures that a frame will always reach the destination at the source speed even if a physical failure happens in one of the networks [66]. Figure 12 is an illustration of the operation in a PRP network [66].

Depending on the network topology through which the PRP is running, it has the advantage of preserving the communication from multiple link failures. It is only effective when the faulty transmission links are not directly connected to one of the critical end-devices in communication (any one of the two network switches in Figure 12).

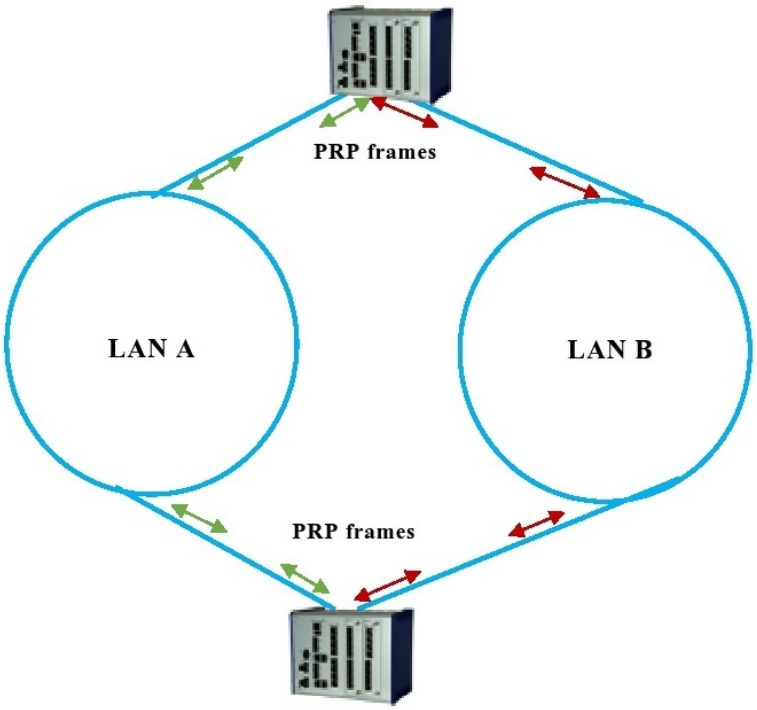

Figure 12. PRP network topology.

- High-Availability Seamless Redundancy Protocol (HSRP): Like PRP, HSR is a zero-loss redundancy protocol defined by the IEC 62439-3 standard. It also consists of sending duplicated frames through the network to ensure prompt delivery to the destination in hardware or software failure in network devices. The HSR protocol is designed for use in a ring topology only with a maximum of 512 participants. In an HSR network, all ring devices need to support the HSR protocol to forward duplicated frames and discard the one arriving second. The HSR capable devices are called Double-Attached Node supporting HSR (DAN H) [66,67]. Figure 13 is a graphical representation of operations in an HSR network [66].

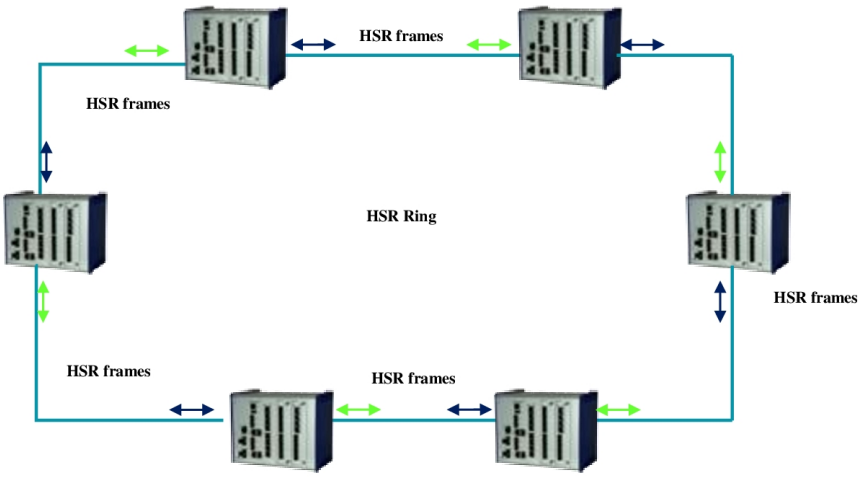

Figure 13. HSR network topology.

Our research integrates the advantages of these two zero-loss redundancy protocols: PRP and HSR, to design a capable Ethernet (IP-based) communication prototype for IIoT

time-critical applications. We summarize in Table 1 the different redundancy protocols explored in this section and some of their critical features to consider for network design.

Table 1. Redundancy protocols features summary.

Redundancy Protocol	Network Topology	Max. Number of Switches	Recovery time	Protection limitation
RSTP	Ring	40	≥2 s (Depends on network size)	Single link failure
RSTP	Mesh, Start, Any other	Infinite	≥2 s	Single link failure (Except for mesh)
MRP	Ring	50	500 ms	Single link failure
PRP	Double networks	Infinite	0 ms	Multiple link failure
HSR	Ring	512	0 ms	Single link failure

4. Results and Discussions

4.1. The Effective Network Communication Prototypes Design

The design of our communication prototype combines some of the best physical and software methods that make a reliable industrial communication system. On the physical side of the prototype, we implement zero-loss recovery redundancy protocols: PRP and HSR to reduce the risk of communication delays due to hardware failure on physical components such as cables or switches. Figure 14 represents our communication prototype using the PRP protocol, and Figure 15 represents the communication prototype using the HSR protocol. High-availability redundancy protocols are essential for IIoT time-critical applications which do not tolerate communication downtime. They offer a solution to unforeseen link and physical device failures that are almost inevitable. PRP and HSR capable equipment (switches) forward duplicated frames in the network from two different ports in our design. This kind of transmission maximizes network availability in the event of a failure.

The PRP communication prototype consists of two PRP-capable devices (A and B) that send and receive the duplicated frames from two independent MRP rings (MRP ring 1 and MRP ring 2). The MRP rings have three switches each. The ring manager (RM) controls redundancy operations. If one side of the ring is ever affected, the other independent ring takes over without delay. The PRP communication prototype supports network protection against multiple link failures by transmitting frames via two independent rings topologies, unlike a standalone MRP or RSTP ring that only protects against a single point of failure. However, as previously mentioned, the multiple link protection is only applicable to any network link except those directly corrected to the PRP-capable devices. Table 2 is a summary of switches functions and attributes of the PRP network prototype in Figure 14.

The HSR communication prototype lies in a ring topology made of HSR-capable devices that send and receive duplicated frames in two different ports. Although the HSR communication prototype is also a single point of failure protection scheme, it usually needs less network infrastructure (cabling, switches) than a PRP network.

On the software side of the prototype, we integrate the concept of edge computing to lessen the network latency and network bandwidth utilization due to the amount of data sent for advanced processing directly to a faraway cloud by several devices (controllers, IIoT devices, and field devices). An edge server closer to the factory network can process advanced data functions from the network devices and communicate back to them. The edge server interacts with the cloud at a higher level at non-peak activity hours (to spare the bandwidth use) or non-time-critical responses. In a communication network, not all frames have the same priority. Our communication prototype uses TSN-capable synchronized switches to guarantee delay-free communication for time-critical data. An example of time-critical frame transmission with and without TSN capabilities is presented by FU S. et al. (2018) [57] for two consecutive transmission cycles. TSN offers several mechanisms such as the IEEE 802.1Qbv Guard Bands Mechanism and the IEEE 802.1Qbu Frame Pre-emption to protect time-critical priority windows from the intrusion of any other frame transmission. In this specific case, the frame's diffusion about to jump to a time-critical frame is paused by the guard band and retransmitted after the time-critical window. In

this case, time-critical frames will not experience any delay before being forwarded. Our communication prototypes integrate TSN-capable switches to avoid unnecessary delays for time-critical frame transmission.

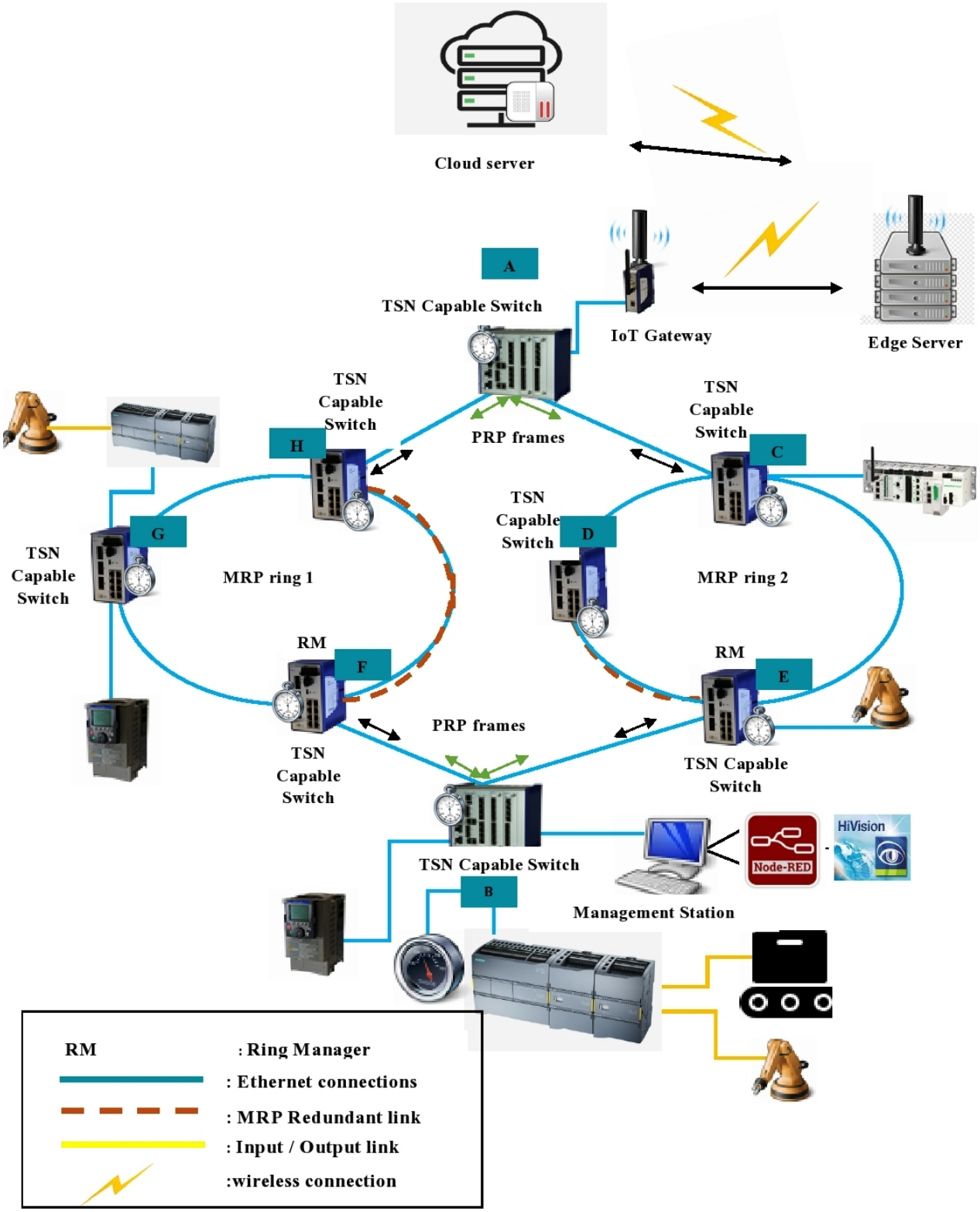

Figure 14. Network communication prototype using PRP.

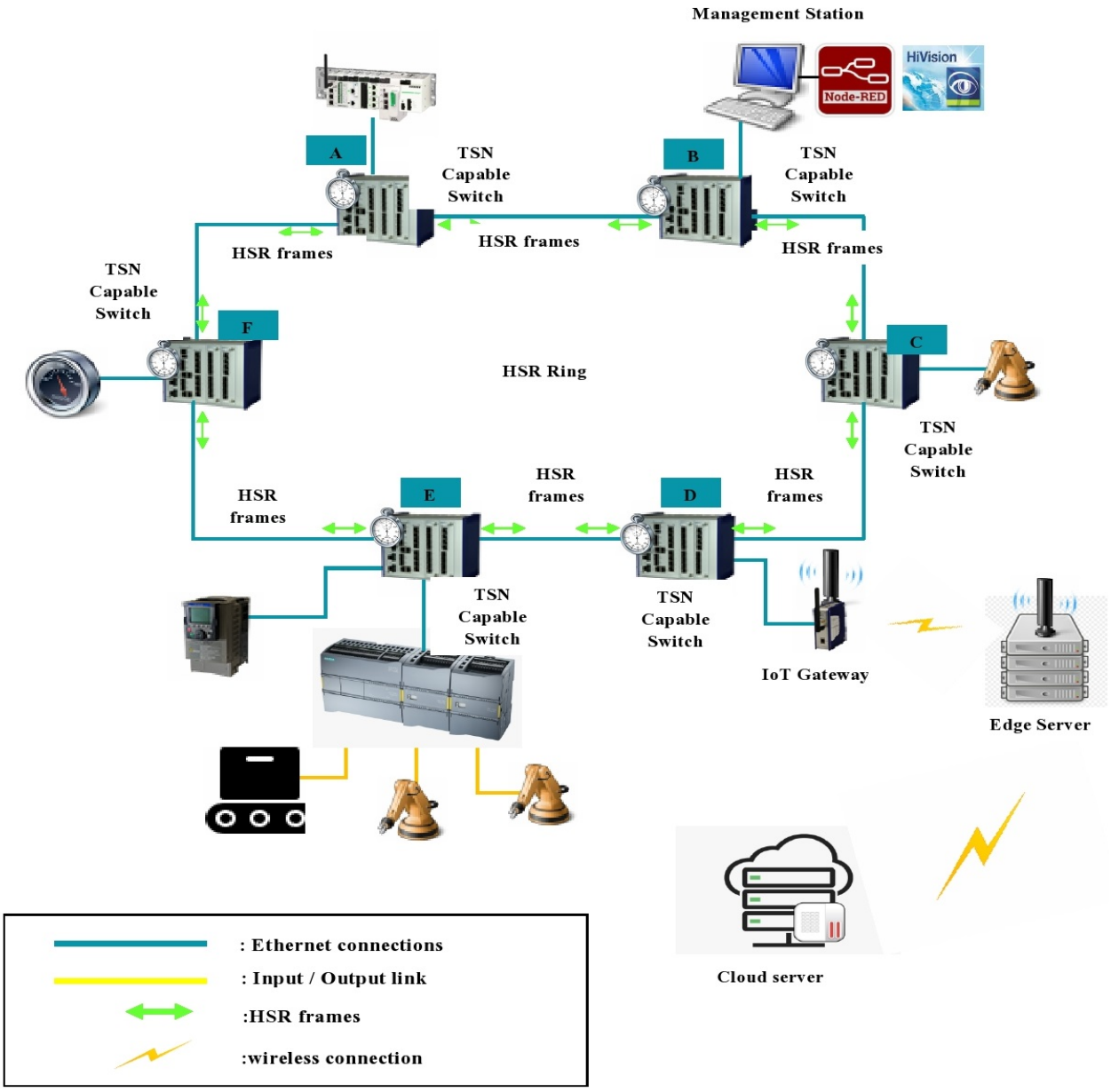

Figure 15. Network communication prototype using HSR.

Table 2. Switches attributes in PRP network prototype.

Switches	Redundancy Protocol	Redundancy Protocol
A	PRP	TSN capable
B	PRP	TSN capable
C	MRP (ring 2)	TSN capable
D	MRP (ring 2)	TSN capable
E	MRP (ring 2)	TSN capable, Ring Manager
F	MRP (ring 1)	TSN capable, Ring Manager
G	MRP (ring 1)	TSN capable
H	MRP (ring 1)	TSN capable

4.2. Frame Transmission Time in TSN-Capable versus Non-TSN-Capable Network Switches

The network communication prototype we designed in the previous section uses TSN-capable switches to exchange data. It means that all critical messages will always have priority when transmitted through switches supporting TSN. We provided more details on the TSN frame transmission in the previous section. The TSN standardization process is still under development and not fully distributed in the market yet. However, some simulations demonstrating the operation of TSN communication have been conducted via the Omnet++ open-source network simulator [68]. Suljic H. and Muminovic M. (2019) [68] developed a performance study and analysis of time-sensitive networking, where they tested up to five scenarios using the Omnet++ network simulator. In our study, we have not conducted any simulation for the TSN concept using Omnet++. Our research's purpose is mainly to include the TSN operational principle into a robust communication network infrastructure for high availability and low latency. Note that the primary role of the TSN standards is to ensure that time-critical frames travel smoothly from source to destination without being impacted by the transmission of other frames with less priority. TSN does not erase the delay caused by frames traveling over the cable medium or the transmission delay due to the transmission speed.

We display in Figure 16 an RSTP ring network monitored by the Hirschmann Industrial Hivision network managed software (implemented as a standalone protection scheme or redundancy protocol). The dotted lines between switch 172.16.4.3 and switch 172.16.4.4 represent the redundant link blocked in regular operation to avoid loops. We use the mpingLCD tool to measure the recovery time whenever a fault occurs. We set up the mpingLCD tool to monitor the recovery time between the PC 172.16.4.205 and the switch 172.16.4.6. In Figure 17, a fault occurred in the RSTP ring, the redundant link is now active (a solid line), and data are retransmitted. The recovery time measured is approximately 1 s:940 ms. We only tested the recovery time on a single link failure. As previously explained, a single link failure will have no impact (no communication delays) on our communication prototypes because of the zero-loss redundancy protocols implemented (PRP and HSR). The recovery time in the RSTP network is directly proportional to the network size. The RSTP ring network is relatively small; therefore, the recovery time is negligible depending on the application. The time is suitable for non-time-critical applications.

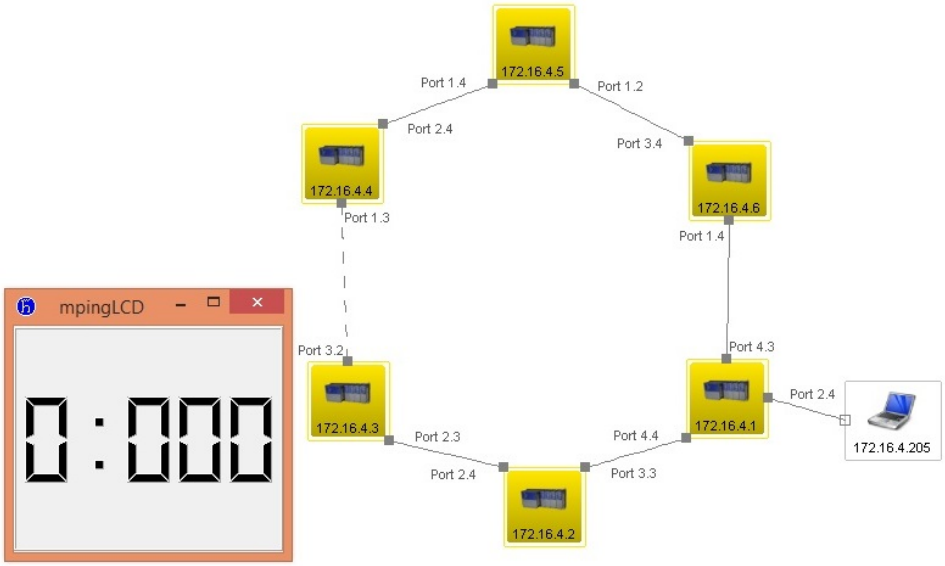

Figure 16. RSTP ring network with no cable failure.

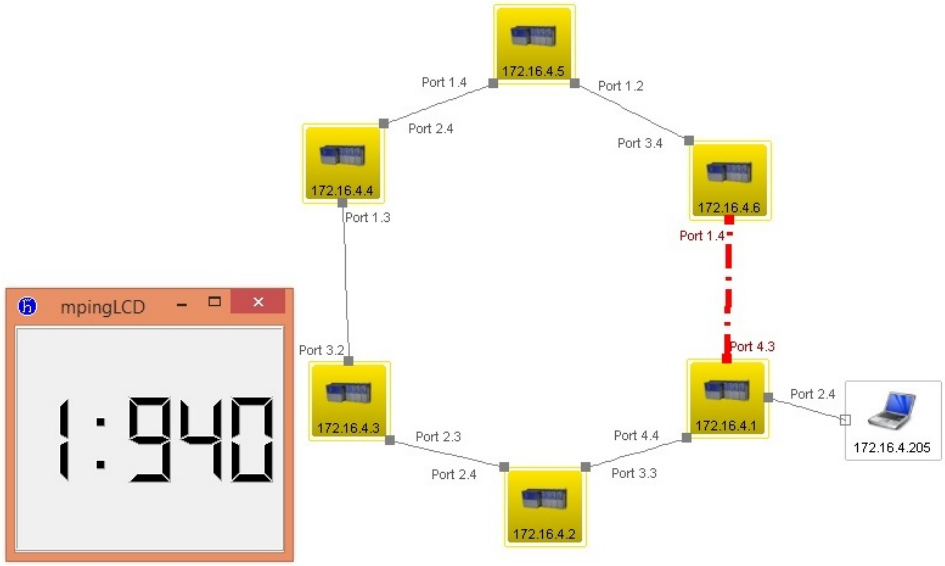

Figure 17. RSTP ring network with one cable (link) failure.

We present an MRP ring network (implemented as a standalone protection scheme or redundancy protocol) in Figure 18 monitored using the Hirschmann Industrial Hivision software. As for the RSTP network, the dotted lines are an indication of the redundant link. In MRP rings, the dotted lines are always next to the RM. In this network, the switch with IP address 172.16.4.1 is the RM. When one of the cables or one switch is faulty, the redundant link becomes active, and the frame transmission goes through it. Figure 19 displays the MRP ring network with a broken link and a recovery time of 40 ms for a relatively small network.

From Figure 19 in the MRP ring, the link data rate between switches is x = 100 Mbps (known as the fast Ethernet data rate). For a frame of size η, in bits, traveling from switch with IP address: 172.16.4.1 to switch 172.16.4.6 through switches 172.16.4.2, 172.16.4.3, 172.16.4.4, and 172.16.4.5, the frame transmission delay δ_{frm}, that is the transmission time for frames traveling through TSN-capable switches, can be estimated as

$$\delta_{frm} = \frac{\eta}{100\times10^6} + \frac{\eta}{100\times10^6} + \frac{\eta}{100\times10^6} + \frac{\eta}{100\times10^6} + \frac{\eta}{100\times10^6}$$

$$\delta_{frm} = 5\frac{\eta}{100\times10^6} = \frac{\eta}{20}\mu s$$

If the five network switches through which the frame η traveled had different link data rates ($x_1, x_2, x_3, x_4, \; and \; x_5$), the transmission delay would have be calculated by

$$\delta_{frm} = \frac{\eta}{x_1} + \frac{\eta}{x_2} + \frac{\eta}{x_3} + \frac{\eta}{x_4} + \frac{\eta}{x_5}$$

$$\delta_{frm} = \eta(\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \frac{1}{x_4} + \frac{1}{x_5})$$

The transmission time of the same frame size over a non-TSN capable switches network would depend on many other external factors such as the number of frames available in every switches' buffer while the time-critical frame is transmitted. It will, therefore, be approximately

$$\delta_{frm} = \frac{\eta}{100\times10^6} + \frac{\eta}{100\times10^6} + \frac{\eta}{100\times10^6} + \frac{\eta}{100\times10^6} + \frac{\eta}{100\times10^6} + \delta_{mry1} + \delta_{mry2} + \delta_{mry3} + \delta_{mry4} + \delta_{mry5}$$

$$\delta_{frm} = 5\frac{\eta}{100\times10^6} + \delta_{mry1} + \delta_{mry2} + \delta_{mry3} + \delta_{mry4} + \delta_{mry5}$$

$$\delta_{frm} = \frac{\eta}{20}\mu s + \delta_{mry1} + \delta_{mry2} + \delta_{mry3} + \delta_{mry4} + \delta_{mry5}$$

where δ_{mry} is the delay of frames in each switch memory defined in (7).

The worst-case delay could happen if, in Figure 19, while the transmission started, another physical failure occurred in the network. In this case, the delay depends on the time needed to reconnect at least one of the links. Our proposed communication prototypes have the benefits of implementing zero-loss redundancy protocols that avoid any recovery time. The prototype built on PRP offers better flexibility for better chances of having smooth communication for more than one physical failure.

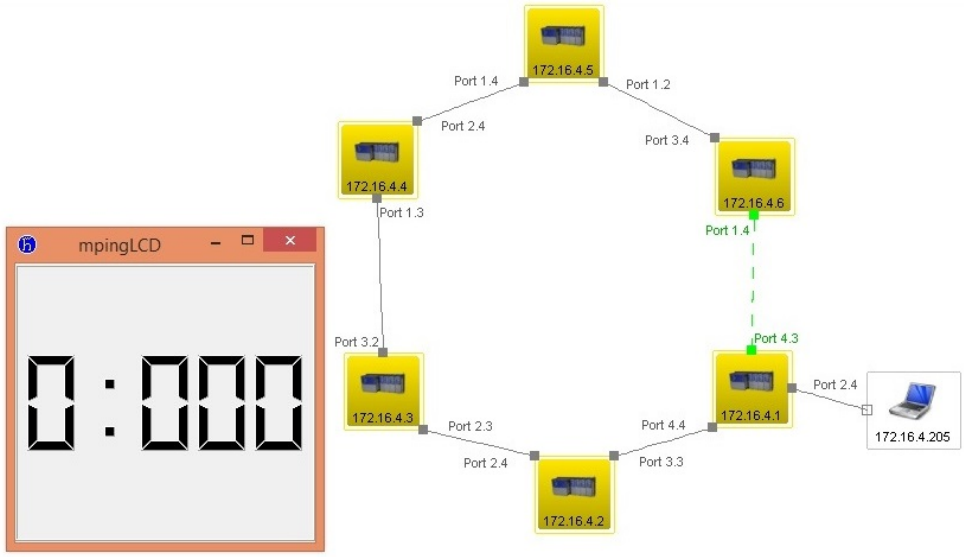

Figure 18. MRP ring network with no cable failure.

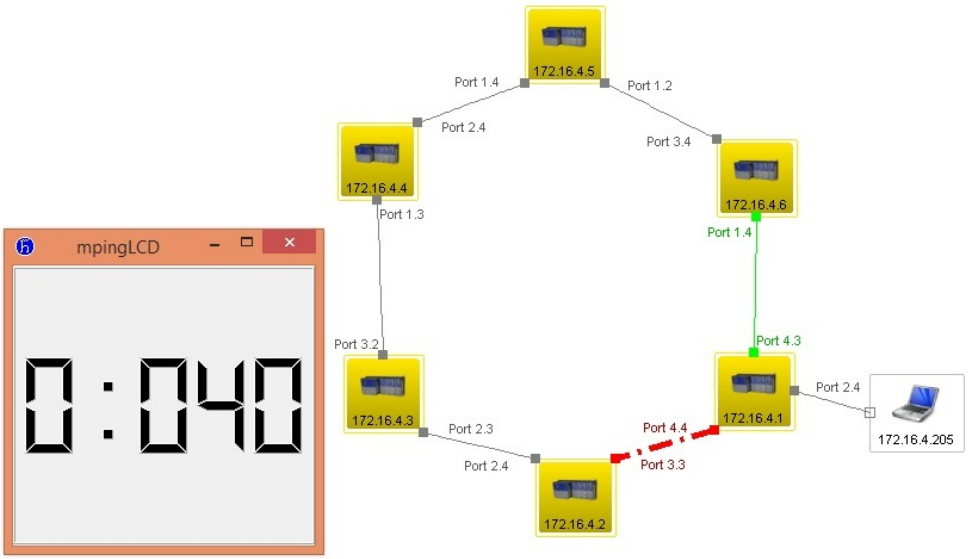

Figure 19. MRP ring network with one cable (link) failure.

4.3. Our Proposed Network Communication Prototypes versus Standalone Protection Schemes (RSTP and MRP)

As per the above results and discussions, Table 3 highlights our proposed network communication prototypes benefits and shortcomings compared to the two standalone redundancy protocols explored in this research: RSTP and MRP.

Table 3. Proposed network communication prototypes versus standalone redundancy protocols.

Network Protection Schemes	Advantages	Disadvantages
PRP-based prototype	Multi-link failures, 0 ms recovery time, Low communication latency with TSN and Edge computing technology	Requires a high number of infrastructure components (cabling and switches) and specialized switches supporting PRP technology
HSR-based prototype	0 ms recovery time, Low communication latency with TSN and Edge computing technology, Low number of infrastructure components needed (simple ring topology)	Single-link failure, requires specialized switches supporting HSR technology in the entire ring
MRP	Low number of infrastructure components needed (simple ring topology), Standard redundancy protocol available in most industrial network switches	Single-link failure, Delayed recovery time, Communication latency
RSTP	Low number of infrastructure components needed (simple ring topology), Standard redundancy protocol available in most industrial network switches, Relatively easy to configure	Single-link failure, Delayed recovery time, Communication latency

From Table 3, we notice that the standalone redundancy protocols RSTP and MRP are not suitable for time-critical applications because of the delayed recovery time and the communication latency that are unacceptable in these applications. Other studies that integrate zero-loss redundancy protocols such as Xu, B. et al. (2021) [38] do not offer a solution to prevent the low-latency communication due to high data volume (especially in an IIoT environment). Our proposed network communication prototypes combine zero-loss redundancy protocols, TSN, and edge computing to palliate these shortcomings and offer more reliable industrial communication networks.

5. Conclusions

In this research, we designed two effective IP-based network communication prototypes to solve the demanding requirements of a highly stable and reliable network for IIoT time-critical applications. We integrated the operational principles of zero-loss redundancy protocols PRP and HSR to create robust protection against network downtime due to link and network devices failures. Our PRP-based communication prototype, in particular, offers network protection against multiple link failures. The results section compares our proposed prototype features to two available standalone redundancy protocols: MRP and RSTP. Although both existing protocols appear easy to implement in network switches and require less network infrastructure, they cannot meet zero-loss recovery time during link failures and are therefore unfit for IIoT time-critical applications. Furthermore, these two standalone redundancy protocols are only suitable for a single point of failure, unlike our PRP-based prototype. Our proposed solution goes a step further by integrating current state-of-the-art communication technologies like TSN and edge computing to reduce communication latency risks during data transmission. The result section also demonstrates the importance of implementing TSN-capable switches in a communication network by estimating the frame transmission time with and without TSN capabilities. The use of TSN in network switches lessens the impact of unnecessary delays due to external factors such as additional frame storage time in switches buffers. While most previous researches offer solution enhancement on either the physical network segment (redundancy protection schemes) or its software segment (data transmission improved systems), the combination of zero-loss redundancy protocols with TSN and edge computing suggested by our communication prototypes creates an effective and highly reliable communication prototype.

For future works, we expect to investigate detailed configurations and platforms required to include the transmission of legacy hardware data present in an advanced IIoT environment without compromising the stability of the network. We want to explore different scenarios and details on implementing our IP-based network communication prototypes in inter-domain transmissions. We also intend to provide a more in-depth approach to implementing TSN priority windows on all network devices and network monitoring software. The in-depth use of dedicated simulators for each concept implemented in

the proposed communication prototype design (PRP, HSR, TSN, and edge computing) is interesting for future work.

Author Contributions: Conceptualization, K.S.K. and Z.W.; methodology, K.S.K. and Z.W.; software, K.S.K.; validation, K.S.K. and Z.W.; formal analysis, K.S.K. and Z.W.; investigation, K.S.K. and Z.W.; resources, K.S.K.; data curation, K.S.K.; writing—original draft preparation, K.S.K.; writing—review and editing, Z.W.; visualization, K.S.K.; supervision, Z.W.; project administration, K.S.K.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported partially by South African National Research Foundation Grants (No. 112108, 132797 and 137951) and Tertiary Education Support Program (TESP) of South African ESKOM.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following symbols are used in this manuscript:

δ_{cmin}	Minimum communication frame delay
δ_{src}	Frame processing delay from the source node or switch
δ_{dst}	Processing delay at the destination node
δ_{frm}	Delay generated by the frame transmission
δ_{cbl}	Delay caused by the electrical signal traveling through the physical medium
η	Size of the transmitted frame in bits
x	Link data rate in bits per seconds
δ_c	Overall frame communication delay when stored before transmission
δ_{mry}	Frame delay in switch memory
F_m	Number of frames waiting in the switch memory
IFG	Inter frame gap
S_n	Data size of the n^{th} frame in the queue
S_{hd}	Overhead of the frame
δ_{YZ}	Frame delay from point Y to point Z
δ_{t1}	Delay to transmit all frames into the link
δ_{lnkp}	Delay a frame encounters to travel on each link based on its data rate
δ_{swtp}	Frame processing delay from a switch ingress port to its egress port

References

1. Dao, N.; Lee, Y.; Cho, S.; Kim, E.; Chung, K.; Keum, C. Multi-tier multi-access edge computing: The role for the fourth industrial revolution. In Proceedings of the 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 18–20 October 2017; pp. 1280–1282.
2. Al-Gumaei, K.; Schuba, K.; Friesen, A.; Heymann, S.; Pieper, C.; Pethig, F.; Schriegel, S. A survey of Internet of Things and big data integrated solutions for industrie 4.0. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018; pp. 1417–1424.
3. Bedi, G.; Venayagamoorthy, K.; Singh, R.; Brooks, R.R.; Wang, K.C. Review of Internet of Things (IoT) in electric power and energy systems. *IEEE Internet Things J.* **2018**, *5*, 847–870. [CrossRef]
4. Miraz, M.H.; Ali, M.; Excell, P.S.; Picking, R. *A Review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT)*; Internet Technologies and Applications (ITA): Wrexham, UK, 2015; pp. 219–224.
5. Yang, G.; Xie, L.; Mantysalo, M.; Zhou, X.L.; Pang, Z.B.; Xu, L.D.; Kao-Walter, S.; Chen, Q.; Zheng, L.R. A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2180–2191. [CrossRef]
6. Pasluosta, C.F.; Gassner, H.; Winkler, J.; Klucken, J.; Eskofier, B.M. An emerging era in the management of Parkinson disease: Wearable technologies and the internet of things. *IEEE J. Biomed. Health Inform.* **2015**, *19*, 1873–1881. [CrossRef] [PubMed]
7. Lin, J.; Yu, W.; Zhang, N.; Yang, X.Y.; Zhang, H.L.; Zhao, W. AA survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet Things J.* **2017**, *4*, 1125–1142. [CrossRef]

8. Schriegel, S.; Kobzan, T.; Jasperneite, J. Investigation on a distributed SDN control plane architecture for heterogeneous time sensitive networks. In Proceedings of the 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), Imperia, Italy, 13–15 June 2018; pp. 1–10.
9. Kharb, S.; Singhrova, A. Fuzzy based priority aware scheduling technique for dense industrial iot networks. *J. Netw. Comput. Appl.* **2019**, *125*, 17–27. [CrossRef]
10. Perera, C.; Liu, C.H.; Jayawardena, S.; Chen, M. A survey on Internet of Things from industrial market perspective. *IEEE Access* **2014**, *2*, 1660–1679. [CrossRef]
11. Rezaeibagha, F.; Mu, Y.; Huang, X.; Yang, W.; Huang, K. Fully Secure Lightweight Certificateless Signature Scheme for IIoT. *IEEE Access* **2019**, *7*, 144433–144443. [CrossRef]
12. Jaloudi, S. Communication Protocols of an Industrial Internet of Things Environment: A Comparative Study. *Future Internet* **2019**, *11*, 66. [CrossRef]
13. Heymann, S.; Stojanovci, L.; Watson, K.; Nam, S.; Song, B.; Gschossmann, H.; Schriegel, S.; Jasperneite, J. Cloud-based plug and work architecture of the IIC testbed smart factory Web. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018; pp. 187–194.
14. Kobzan, T.; Schriegel, S.; Althoff, S.; Boschmann, A.; Otto, J.; Jasperneite, J. Secure and time-sensitive communication for remote process control and monitoring. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018; pp. 1105–1108.
15. Prinz, F.; Schoeffler, M.; Lechler, A.; Verl, A. End-to-end Redundancy between Real-time I4.0 Components based on Time-Sensitive Networking. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018; pp. 1083–1086.
16. Ergenç, D.; Fischer, M. On the Reliability of IEEE 802.1CB FRER. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications, Virtual Conference, 10–13 May 2021; pp. 1–10.
17. Bruckner, D.; Stănică, M.P.; Blair, R.; Schriegel, S.; Kehrer, S.; Seewald, M.; Sauter, T. An Introduction to OPC UA TSN for Industrial Communication Systems. *Proc. IEEE* **2019**, *107*, 1121–1131. [CrossRef]
18. Yu, Q.; Gu, M. Adaptive group routing and scheduling in multicast time-sensitive networks. *IEEE Access* **2020**, *8*, 37855–37865. [CrossRef]
19. Vlk, M.; Hanzálek, Z.; Brejchová, K.; Tang, S.; Bhattacharjee, S.; Fu, S. Enhancing Schedulability and Throughput of Time-Triggered Traffic in IEEE 802.1Qbv Time-Sensitive Networks. *IEEE Trans. Commun.* **2020**, *68*, 7023–7038. [CrossRef]
20. Zhu, H.; Liu, K.; Yan, Y.; Zhang, H.; Huang, T. Measures to Improve the Accuracy and Reliability of Clock Synchronization in Time-Sensitive Networking. *IEEE Access* **2020**, *8*, 192368–192378. [CrossRef]
21. Jin, X.; Xia, C.; Guan, N.; Zeng, P. Joint Algorithm of Message Fragmentation and No-Wait Scheduling for Time-Sensitive Networks. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 478–490. [CrossRef]
22. Jin, X.; Xia, C.; Guan, N.; Xu, C.; Li, D.; Yin, Y.; Zeng, P. Real-time scheduling of massive data in time sensitive networks with a limited number of schedule entries. *IEEE Access* **2020**, *8*, 6751–6767. [CrossRef]
23. Cao, K.; Liu, Y.; Meng, G.; Sun, Q. An overview on edge computing research. *IEEE Access* **2020**, *8*, 85714–85728. [CrossRef]
24. Chen, B.; Wan, J.; Celesti, A.; Li, D.; Abbas, H.; Zhang, Q. Edge Computing in IoT-Based Manufacturing. *IEEE Commun. Mag.* **2018**, *9*, 103–109. [CrossRef]
25. Pustokhina, I.V.; Pustokhin, D.A.; Gupta, D.; Khanna, A.; Shankar, K.; Nguyen, G.N. An Effective Training Scheme for Deep Neural Network in Edge Computing Enabled Internet of Medical Things (IoMT) Systems. *IEEE Access* **2020**, *8*, 107112–107123. [CrossRef]
26. Qi, Q.; Tao, F. A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing. *IEEE Access* **2019**, *7*, 86769–86777. [CrossRef]
27. Liao, H.; Zhou, Z.; Zhao, X.; Zhang, L.; Mumtaz, S.; Jolfaei, A.; Ahmed, S.H.; Bashir, A.K. Learning-Based Context-Aware Resource Allocation for Edge-Computing-Empowered Industrial IoT. *IEEE Internet Things J.* **2020**, *7*, 4260–4277. [CrossRef]
28. Gong, C.; Lin, F.; Gong, X.; Lu, Y. Intelligent Cooperative Edge Computing in Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 9372–9382. [CrossRef]
29. Carvalho, A.; O'Mahony, N.; Krpalkova, L.; Campbell, S.; Walsh, J.; Doody, P. Edge computing applied to industrial machines. *Procedia Manuf.* **2019**, *38*, 178–185. [CrossRef]
30. Chen, Y.; Sun, Y.; Lu, N.; Wang, B. Channel-reserved medium access control for edge computing based IoT. *J. Netw. Comput. Appl.* **2020**, *150*, 102500. [CrossRef]
31. Roig, P.J.; Alcaraz, S.; Gilly, K. Formal Specification of Spanning Tree Protocol Using ACP. *Elektron. Elektrotechnika* **2017**, *23*, 84–91.
32. Longo, E.; Redondi, A.E.C.; Cesana, M.; Arcia-Moret, A.; Manzoni, P. MQTT-ST: A Spanning Tree Protocol for Distributed MQTT Brokers. In Proceedings of the ICC 2020–2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
33. Giorgetti, A.; Cugini, F.; Paolucci, F.; Valcarenghi, L.; Pistone, A.; Castoldi, P. Performance analysis of media redundancy protocol (MRP). *IEEE Trans. Ind. Inform.* **2013**, *9*, 218–227. [CrossRef]
34. Naukkarinen, H. Ethernet Technology in Safety Automation. Bachelor's Thesis, Metropolia University of Applied Sciences, Vantaa, Finland, 2020.

35. Musaddiq, A.; Zikria, Y.B.; Hahm, O.; Yu, H.; Bashir, A.K.; Kim, S.W. A survey on resource management in IoT operating systems. *IEEE Access* **2018**, *6*, 8459–8482. [CrossRef]
36. Wylian, S.F. Multiple Spanning-Tree (MST) to Improve Enterprise Network Security. Available online: https://elar.urfu.ru/bitstream/10995/84223/1/978-5-91256-486-4_2020_083.pdf (accessed on 29 October 2021).
37. Willis, P.; Shenoy, N.; Pan, Y.; Hamilton, J. Root Redundancy in Meshed Tree Bridged Networks. In Proceedings of the IEEE 45th Conference on Local Computer Networks (LCN), Sydney, Australia, 16–19 November 2020; pp. 297–308.
38. Xu, B.; Gao, J.; Bosley, B.; Garcia, J.; Clark, T. Fast Load Shedding Scheme for Enhancing Reliability and Stability of Expanded Liquified Gas Plant. In Proceedings of the 74th Conference for Protective Relay Engineers (CPRE), College Station, TX, USA, 22–25 March 2021; pp. 1–7.
39. Lo Bello, L.; Steiner, W. A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems. *Proc. IEEE* **2019**, *6*, 1094–1120. [CrossRef]
40. Ali, A.; Feng, L.; Bashir, A.K.; El-Sappagh, S.; Ahmed, S.H.; Iqbal, M.; Raja, G. Quality of service provisioning for heterogeneous services in cognitive radio-enabled internet of things. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 328–342. [CrossRef]
41. Muhammad, G.; Alhamid, M.F.; Alsulaiman, M.; Gupta, B. Edge computing with cloud for voice disorder assessment and treatment. *IEEE Commun. Mag.* **2018**, *56*, 60–65. [CrossRef]
42. Lee, K.C.; Lee, S.; Lee, M.H. Worst Case Communication Delay of Real-Time Industrial Switched Ethernet with Multiple Levels. *IEEE Trans. Ind. Electron.* **2006**, *53*, 1669–1676. [CrossRef]
43. Karachalios, K. *A Vision for the Next Wave of Connectedness*; IEEE GSMA PSMC: Barcelona, Spain, 2017.
44. Oyekanlu, E. Predictive edge computing for time series of industrial IoT and large scale critical infrastructure based on open-source software analytic of big data. In Proceedings of the 2017 IEEE International Conference on Big Data, Boston, MA, USA, 11–14 December 2017; pp. 1663–1669.
45. Islam, M.T.; Taha, A.E.M.; Akl, S. A survey of access management techniques in machine type communications. *IEEE Commun. Mag.* **2014**, *52*, 74–81. [CrossRef]
46. Anagnostopoulos, C.; Kolomvatsos, K. An intelligent, time-optimized monitoring scheme for edge nodes. *J. Netw. Comput. Appl.* **2019**, *148*, 102458. [CrossRef]
47. Ray, P.P.; Dash, D.; De, D. Edge computing for Internet of Things: A survey, e-healthcare case study and future direction. *J. Netw. Comput. Appl.* **2019**, *140*, 1–22. [CrossRef]
48. Raileanu, S.; Borangiu, T.; Morariu, O.; Iacob, I. Edge Computing in Industrial IoT Framework for Cloud-based Manufacturing Control. In Proceedings of the 2018 22nd International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 10–12 October 2018; pp. 261–266.
49. Huynh, M.; Goose, S.; Mohapatra, P.; Liao, R. RRR: Rapid Ring Recovery Submillisecond Decentralized Recovery for Ethernet Ring. *IEEE Trans. Comput.* **2011**, *60*, 1561–1570. [CrossRef]
50. Nigel, B. *PRfc 2544 Testing of Ethernet Services in Telecom Networks—White Paper*; Agilent Technologies: Santa Clara, CA, USA, 2004. Available online: http://literature.cdn.keysight.com/litweb/pdf/5989-1927EN.pdf (accessed on 10 November 2021).
51. Gutiérrez, C.S.V.; Juan, L.U.S.; Ugarte, I.Z.; Vilches, V.M. Time sensitive networking for robotics. *arXiv* **2018**, arXiv:1804.07643.
52. Maestro, J.A.; Reviriego, P. Energy Efficiency in Industrial Ethernet: The Case of Powerlink. *IEEE Trans. Ind. Electron.* **2010**, *57*, 2896–2903. [CrossRef]
53. Skeie, T.; Johannessen, S.; Holmeide, O. Timeliness of real-time IP communication in switched industrial Ethernet networks. *IEEE Trans. Ind. Inform.* **2006**, *2*, 25–39. [CrossRef]
54. Farkas, J.; Bello, L.L.; Gunther, C. TTime-sensitive networking standards. *IEEE Commun. Stand. Mag.* **2018**, *2*, 20–21. [CrossRef]
55. Adame, T.; Carrascosa, M.; Bellalta, B. Time-Sensitive Networking in IEEE 802.11be: On the Way to Low-latency WiFi 7. *arXiv* **2020**, arXiv:1912.06086.
56. Tian, S.; Hu, Y. The Role of OPC UA TSN in IT and OT Convergence. In Proceedings of the 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22–24 November 2019; pp. 2272–2276.
57. Fu, S.; Zhang, H.; Chen, J. Time-sensitive networking technology overview and performance analysis. *ZTE Commun.* **2018**, *16*, 57–64.
58. Prinz, F.; Schoeffler, M.; Lechler, A.; Verl, A. A Dynamic real-time orchestration of i4.0 components based on time-sensitive networking. *Procedia CIRP* **2018**, *72*, 910–915. [CrossRef]
59. Ashjaei, M.; Lo Bello, L.; Daneshtalab, M.; Patti, G.; Saponara, S.; Mubeen, S. Time-Sensitive Networking in automotive embedded systems: State of the art and research opportunities. *J. Syst. Archit.* **2021**, *117*, 102137. [CrossRef]
60. Vitturi, S.; Zunino, C.; Sauter, T. Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G. *Proc. IEEE* **2019**, *107*, 944–961. [CrossRef]
61. Farkas, J. TSN Basic Concepts. DetNet—TSN workshop 2018. Available online: https://www.ieee802.org/1/files/public/docs2018/detnet-tsn-farkas-tsn-basic-concepts-1118-v01.pdf (accessed on 10 November 2021).
62. Wu, F.; Tian, A. rXstp: A Topology Discovery Mechanism Based on Rapid Spanning Tree for SDN In-Band Control. In Proceedings of the International Conference on Communications, Information System and Computer Engineering (CISCE), Beijing, China, 14–16 May 2021; pp. 703–706.
63. Lindstrom, H. Migration to P4-Programmable Switches and Implementation of the Rapid Spanning Tree Protocol. Master's Thesis, Linköping University, Linköping, Sweden, 2020.

64. Peón, P.G.; Steiner, W.; Uhlemann, E. Network Fault Tolerance by Means of Diverse Physical Layers. In Proceedings of the 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; pp. 1697–1704.
65. Media Redundancy Concepts—High Availability in Industrial Ethernet. Available online: https://www.belden.com/hubfs/emea/resources/Picture%20Park%20Assets\/Files%20for%20Redirection/WP_%20Media%20Redundancy%20\Concepts_Original_64020.pdf (accessed on 2 February 2021).
66. Araujo, J.A.; Lázaro, J.; Astarloa, A.; Zuloaga, A.; Gárate, J.I. PRP and HSR for High Availability Networks in Power Utility Automation: A Method for Redundant Frames Discarding. *IEEE Trans. Smart Grid* **2015**, *6*, 2325–2332. [CrossRef]
67. Khoshnevisan, M.; Joseph, V.; Gupta, P.; Meshkati, F.; Prakash, R.; Tinnakornsrisuphap, P. 5G Industrial Networks With CoMP for URLLC and Time Sensitive Network Architecture. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 947–959. [CrossRef]
68. Suljic, H.; Muminovic, M. Performance Study and Analysis of Time Sensitive Networking. Master's Thesis, Malardalen University School of Innovation Design and Engineering Vasteras, Västerås, Sweden, 2019.