Supporting vPLC Networking over TSN with Kubernetes in Industry 4.0

Lorenzo Rosa, Andrea Garbugli, Lorenzo Patera, Luca Foschini
University of Bologna
Bologna, Italy
{name.surname}@unibo.it

ABSTRACT

The shift in the industrial ecosystem from closed and specialized technologies to the open and general-purpose vision of Industry 4.0 faces numerous challenges. The absence of viable solutions to replace Programmable Logic Controllers (PLCs), vital components in control infrastructures, with their virtual equivalent (vPLCs) embodies those difficulties. In this paper, we introduce a framework that aims at truly materializing the integration between Operational (OT) and Information Technologies (IT) by defining an open, general ecosystem around vPLCs. Previous work either could not meet the performance and determinism requirements of the OT or did so by sacrificing the generality of IT. Building on these experiences, our framework provides both flexibility and efficiency by clearly separating the data path for OT and IT communications. To do that, we integrate tools from both domains: techniques to ensure low network performance and variability (TSN), to ease portability (OPC-UA), and to enhance management and deployment (Kubernetes). Experiments on a real testbed show that vPLCs within our framework can meet strict performance requirements and yet provide the same flexibility as cloud-based applications.

CCS CONCEPTS

• **Networks** → **Overlay and other logical network structures**; • **Computer systems organization** → **Embedded and cyber-physical systems**;

KEYWORDS

vPLC, Industry 4.0, TSN, Kubernetes, OPC-UA, DPDK

ACM Reference Format:
Lorenzo Rosa, Andrea Garbugli, Lorenzo Patera, Luca Foschini. 2023. Supporting vPLC Networking over TSN with Kubernetes in Industry 4.0. In *Enhanced network techniques and technologies for the Industrial IoT to Cloud continuum (IIoT-NETs '23), September 10, 2023, New York, NY, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3609389.3610566

1 INTRODUCTION

The fourth industrial revolution (Industry 4.0) is driving companies to expand their operations beyond local and limited environments to a broader, global, and interconnected industrial sector. Machines continuously generate and export data that are filtered, processed, and analyzed in near real-time to extract business insights and facilitate accurate and cost-effective decision-making. Thanks to the introduction of the Industrial Internet of Things (IIoT), sensors and software are embedded in smaller and smarter connected devices that allow *cloud-native* communications and *immediate* actions on the surrounding environment [6, 30]. The growing number of IIoT devices is driving companies to increase local computational power and actively pursue integration between their existing Operational (OT) and Information Technologies (IT).

In this context, the increasing amount of scattered data produced by machinery and the necessity of analyzing them is rapidly pushing companies to replace or adapt machine field technologies from proprietary ad hoc industrial protocols to open and more flexible standards (e.g. OPC-UA [9]). Industries have a real opportunity to enhance the automation level and the cohesion between OT and IT in a cost-effective and affordable manner by utilizing Commercial-off-the-Shelf (COTS) hardware and software. This has several benefits: increased community support, reduced maintenance effort, continuous updates, and improved cybersecurity.

A noticeable example of such integration is Virtual Programmable Logic Controllers (vPLCs), which enhance the functionalities of a Programmable Logic Controller (PLCs) with the flexibility only virtualized software can guarantee. Historically, the introduction of PLCs was an essential building block of the *automation revolution* in industrial control systems. It allowed centralizing the intelligence of systems, controlling all the machines' functional areas from a single

---

IIoT-NETs '23, September 10, 2023, New York, NY, USA

ACM ISBN 979-8-4007-0302-7/23/09.
https://doi.org/10.1145/3609389.3610566

logical point and interconnecting complex systems by exchanging signals and coordination breakpoints. Nowadays, vPLCs stand as the ideal choice to embody the integration of OT and IT. Easily scalable, interoperable, deployable everywhere, debuggable, replaceable on the fly, and enriched in features, vPLCs pave the way for the digital revolution and ease the development of digital twins [5, 14]. Coupled with containers that run on general-purpose computing hardware, the power of vPLCs encounters the flexibility of the microservice architecture, becoming even more portable and allowing migration of cloud services closer to the machine field.

However, despite the several advantages of such integration, the actual implementation of vPLCs and other IT-enabled components is still difficult to achieve: OT has demanding requirements in terms of latency, jitter, and Quality of Service (QoS), whereas IT is designed for best-effort behavior. As a consequence, current cloud-native virtualized controllers cannot offer the deterministic behavior and low network latency required by traditional specialized solutions or do so by sacrificing the generality of IT (§ 3).

In this paper, we propose an open framework that combines a set of vendor-agnostic technologies to fully support the adoption of containerized PLCs in industrial control infrastructures. At the same time, the framework guarantees compliance with typical OT requirements such as deterministic network behavior and low-latency communication with the controlled devices. Within our framework, containerized vPLCs are managed by the Kubernetes orchestrator [23] and use the OPC-UA middleware to communicate both with cloud-based nodes (*IT traffic*) and with the controlled devices (*OT traffic*). The key novelty of our solution is a clear separation between the infrastructural support for those communications. Whereas IT traffic follows the standard *best-effort* datapath of general-purpose operating systems, OT traffic uses the OPC-UA Time Sensitive Networking (TSN) profile to signal the need for network determinism. TSN packets are intercepted by KuberneTSN [12], a userspace TSN scheduler that puts them directly on the network fabric, removing typical virtualization overhead. The proposed solution, evaluated on a real testbed, showcases compliance with the most demanding industrial application performance requirements, yet retaining all the advantages of containerized applications.

The remainder of this paper is structured as follows. § 2 provides a concise background, laying the foundation upon which our solution is built and inspired. § 3 explores related works with similar objectives, emphasizing the distinctions and unique characteristics of our framework. A blueprint architecture of our solution is presented in § 4, detailing its design and structure. § 5 presents the results of our experiments conducted on a real industrial infrastructure. Finally, § 6 offers insights into potential future research directions.

2 BACKGROUND

Programmable Logic Controllers (PLCs) represent the first layer of the industrial control infrastructure, directly connected to sensors and actuators and responsible to command their low-level feedback control loops [11]. PLCs must cope with demanding performance constraints: typical industrial applications follow a cyclic behavior of periodic communication with the controller, and even minimal deviations from this pattern might lead to system failures. Thus, PLCs are usually designed as modular embedded systems, programmed with a specialized software and running on dedicated hardware. Controllers are co-located with the industrial machines and interact with them through industrial protocol stacks (e.g., PROFINET, EtherCAT, Modbus) that minimize packet processing overhead, reduce unpredictable delays, and guarantee timely feedback decisions. Although that high degree of specialization guarantees maximum operation efficiency, it also creates a closed ecosystem: PLC hardware is expensive and requires expert programmers; communication is confined to closed networks, thus preventing valuable production data to reach analytics platforms.

In contrast, the fourth industrial revolution breaks that isolation and proposes to replace specialized technologies with general-purpose tools. The novel concept of *virtualized PLC* embodies that transformation by moving the PLC control logic into software that runs in virtualized environments (containers, VMs) and on commodity hardware decoupled from the controlled machines. This approach makes PLC development and deployment easier and cheaper because the PLC control logic becomes a software programmable with general-purpose languages. The integration with standard IT tools, such as orchestrators (e.g., Kubernetes [23]), allows the dynamic (re)scaling and (re)configuration of the whole control infrastructure. Furthermore, while still managing the feedback loops of their controlled devices, vPLCs are also interoperable and integrated with typical IT platforms, thus empowering industrial automation with the advantages of modern big data analytics techniques (see Figure 1).

To cope with their mixed-criticality requirements, vPLCs use general-purpose network equipment and protocol stack. Although suitable for IT platforms, which have no timeliness or determinism constraints, the introduction of general-purpose networking between vPLCs and the controlled machines risks receiving inadequate support from machine vendors and introducing unacceptable delays and unpredictability. To address these concerns, new technologies have been recently introduced: the OPC Unified Architecture (OPC-UA), a platform-independent standard that promotes device interoperability, and the Time-Sensitive Protocol (TSN) which guarantees a deterministic behavior of Ethernet networks.

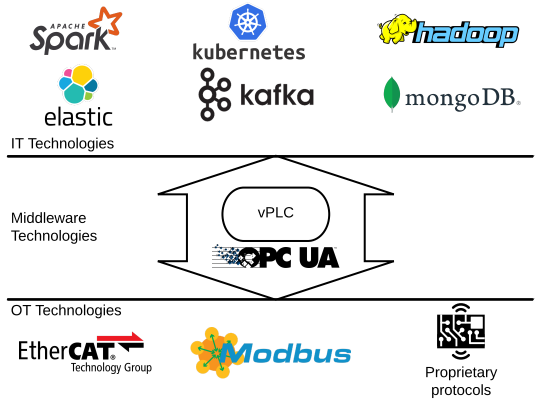


Figure 1: The vPLC embodies the integration between the Operation (OT) and Information Technologies (IT).

The OPC-UA [7] middleware plays an increasingly important role in Industry 4.0. On the one hand, it simplifies machine-to-machine connectivity by defining a unified *information model* that enables seamless interoperability between heterogeneous systems. This model defines a common set of data structures, services, and semantics, facilitating data exchange and integration between disparate industrial automation components. On the other hand, the scalable design makes OPC-UA capable of conveying data to IT platforms either in the cloud or at the edge [16]. To support those different usage patterns, OPC-UA may follow either a client-server or a publish-subscribe communication model [25].

When real-time communication and determinism are critical, OPC-UA can be used in combination with the TSN protocol, a set of standards that provide deterministic, low-latency, and time-synchronized communication over standard Ethernet networks [19, 26]. TSN is based on two fundamental ideas: time synchronization and priority-based frame scheduling. TSN requires all the communication participants to synchronize on a single time reference using a protocol called *generic Precision Time Protocol* (IEEE 802.1AS) [1]. Then, the common time reference can be divided into communication windows (*time-aware traffic windows*) that cyclically repeat; in turn, each window is split into *time slots*. Each participant can configure a scheduler, defined by the IEEE 802.1Qbv standard as Time-Aware Shaper (TAS), to associate time slots to different traffic classes. As a result, frames belonging to the same class are exchanged only during that slot. This way, TSN guarantees bounded latency and jitter for time-critical traffic and minimizes the interference from best-effort traffic.

Together, vPLCs, OPC-UA, and TSN promise to transition industrial control infrastructures toward full OT/IT integration. However, their combination in a framework that guarantees both flexibility and minimal performance overhead is still challenging, as we discuss in the remainder of the paper.

3 RELATED WORK

Despite their crucial role in the vision of Industry 4.0, vPLCs are still far from materializing in real-world industrial settings. Previous research was indeed successful in the definition of architectures for software-based PLCs [2, 4, 8, 16, 17, 24], but the introduction of flexible, cloud-native IT technologies often contrasts with the stringent requirements of the OT layer. In fact, these works agree that currently, their approaches can only support control scenarios without strong performance requirements. In particular, two major sources of variability and performance overhead obstacles to the evolution of traditional PLCs into their virtual equivalent, both related to the replacement of specialized control-to-machine communication with general-purpose techniques: the use of lightweight virtualization mechanisms, such as Virtual Machines (VMs) or containers, and the adoption of general-purpose communication protocols and equipment [13].

To improve the behavior of software-based controllers, other works trade the full generality and flexibility for a more predictable computing and network performance. For example, the use of Real-Time Operating Systems (RTOS) may reduce the variability associated to process scheduling [8, 27], and the adoption of hypervisors from specialized vendors may reduce the virtualization overhead for critical components, including vPLCs [3, 8, 15, 18]. However, RTOSs might harm network latency [27] and specialized hypervisors require specific expertise for software programming, configuration, and deployment, and thus might reproduce the same forms of *vendor lock-in* that Industry 4.0 promises to eliminate. Similarly, networking techniques such as *passthrough* reduce the number of software layers packets should cross, resulting in lower latency and more predictable network operations [15], but also in reduced flexibility (e.g., passthrough prevents live migration).

In recent work, we demonstrated that compromises between the flexibility of virtualized controllers and the OT performance requirements are not always necessary [12, 13]. By combining a novel implementation of the TSN scheduler (§ 2) with modern *kernel-bypassing* networking techniques (in particular, DPDK [10]), we created KuberneTSN, a plugin to create TSN overlay networks among containers. KuberneTSN runs on general-purpose OSes, does not require special hardware, and yet allows virtualized controllers to achieve network performance and determinism comparable to bare-metal deployments.

Building on that work, this paper proposes a novel framework that allows mixed-criticality virtualized control applications, such as vPLCs, to fulfill the typical OT requirements of low network latency and determinism, and yet to run on COTS hardware and software, decoupled from the controlled machines. The key insight of this contribution is the combination of a set of open-source, general-purpose technologies, including our KuberneTSN plugin, into a comprehensive solution that specifically targets Industry 4.0 scenarios. Thus, we enable the dynamic management of the control infrastructure and its full integration within IT platforms, but without sacrificing the primary goal of device control. The next Section describes our proposal in more detail.

4 ARCHITECTURE BLUEPRINT

This Section illustrates our proposed framework to support the requirements of vPLCs in Industry 4.0. On the one hand, we aim to provide an open solution based on COTS software and hardware for the integration of IT into industrial control infrastructures; on the other hand, we must preserve the low network latency and deterministic behavior required by the OT layer. Figure 2 shows a framework overview.

We begin the description of our solution from its core component, the vPLC, which we place within a container (§ 2). That choice minimizes the overhead of virtualization while retaining its several benefits in terms of enhanced portability and scalability, isolated and reproducible environments, simplified dependency management, and improved resource efficiency (§ 3). Containerization allows industrial control systems to benefit from orchestration tools: in our framework, we use Kubernetes as the orchestrator and insert the vPLC container in a Kubernetes *pod*, thereby ensuring its automated scaling, high availability, reliability, and efficient resource allocation in the control infrastructure.

As previously discussed, the hardest challenge for a framework supporting vPLCs is to fulfill their mixed-criticality communication requirements. On the one hand, the vPLC communicates with the IT infrastructure on a best-effort network, exchanging data with the cloud (or edge cloud). On the other hand, it must also interact with the controlled devices on the time-critical network fabric with no or minimal difference from traditional dedicated connections (§ 2). Given the substantial differences between those two classes of traffic, we decided to provide them a corresponding substantially different infrastructural support, at the same maintaining programming transparency for software PLC programmers as well as compatibility with existing PLC software.

We consider that vPLC adopts the OPC-UA middleware for both IT and OT communications. In fact, several vendors already adopt OPC-UA as their communication middleware for new and legacy PLC software [6, 22]. OPC-UA guarantees developers a single point of access to the network, transparent scalability for the interaction with the cloud, and also the rich and standard OPC information model to interact with the OT devices. Although several proprietary OPC-UA implementations exist, there are also open-source versions (e.g., open62541 [29]), which make that software available for vendor-independent deployments.

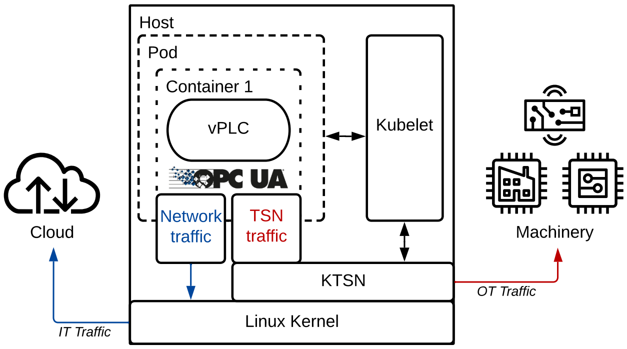


Figure 2: Overall Architecture.

We leverage a specific configuration of OPC-UA, the TSN profile, to let developers signal time-critical traffic directed to the OT fabric (*OT traffic*) and thus requiring determinism and bounded latency, whereas we assume that best-effort guarantees suffice for any other communication (*IT traffic*). We route IT traffic (blue line in Figure 2) through the standard datapath of containerized applications on general-purpose operating systems, which makes packets cross various software layers before reaching the physical network [12].

On the contrary, we leverage KuberneTSN(§ 3) to design a high-performance, TSN-enabled datapath for OT traffic. KuberneTSN (KTSN) creates a virtual overlay network between the vPLC and its controlled devices: any packet sent on that overlay is not forwarded to the standard path, but is intercepted by KTSN and forwarded to a userspace TSN scheduler. When the *transmission time* specified by the application (in this case, the vPLC) comes, the scheduler will send the packet directly to the physical network. This approach bypasses the performance overhead and the intrinsic variability of the standard datapath, thus ensuring deterministic and low-latency communication. KTSN is open-source [28], runs as a daemon on the host, is integrated with Kubernetes as a network plugin [12, 21], and can be controlled by the local Kubernetes component (Kubelet), thus making it easily available and configurable for integration in our framework.

However, by default, KTSN handles packets according to the standard UDP/IP protocol stack, whereas the OPC-UA Pub/Sub TSN profile uses the Unified Architecture Datagram Protocol (UADP) as the primary binary payload representation. UADP is a binary communication protocol within OPC-UA that optimizes data packet size and efficiency for secure and reliable exchange in industrial automation systems. To accommodate for this packet format, we extended KTSN to handle the forwarding of raw Ethernet packets and made also this modified version publicly available [28].

Overall, our framework combines a set of open-source tools, protocols, and technologies to support the effective deployment of vPLCs on COTS hardware, significantly reducing the development and operationalization cost of traditional PLCs, allowing much more flexibility, and guaranteeing the respect of the demanding performance requirements of OT. Furthermore, the use of open-source technologies protects our solution from new and hidden forms of vendor lock-in (e.g., the use of proprietary hypervisors). The next Section demonstrates these properties by running vPLCs within our framework over a real industrial testbed.

5 EXPERIMENTAL RESULTS

In this section, we evaluate the performance of a vPLC application running within our framework with a twofold purpose. First, we want to assess the *virtualization overhead* introduced on the network by the use of containers. Hence, we compare the behavior of the same vPLC application running in two configurations (within our framework and bare-metal) in a real industrial testbed, represented in Figure 3. Second, we evaluate the *compliance* of the results with the requirements of the strictest industrial communication scenarios.

For the purpose of this evaluation, we implement a simple software PLC that we consider a black-box, as we only investigate its networking performance, and that runs in a Docker container. Internally, the vPLC implements an OPC-UA publisher using the open-source OPC-UA implementation open62541 [29]. On the same host, we deploy the KuberneTSN daemon [28] through Kubernetes. On a remote host on the same local network, we run an OPC-UA listener to reproduce the behavior of an industrial device.

5.1 Experimental Settings

The evaluation analysis is conducted on a real testbed comprising two edge nodes and one industrial network switch, as shown in Figure 3. Each host node is equipped with an Intel I225 NIC that supports TSN, an Intel i9-10980XE 18/36 CPU, and 64 GB of RAM. The two hosts are interconnected through a physical TSN-compliant switch. Each host runs Ubuntu 22.04 with Linux kernel 5.16. The two nodes and the switch are synchronized using the PTP protocol, as required by TSN (§ 2). In particular, the two nodes run the linuxptp implementation and are configured as PTP slave clocks, where the switch works as the PTP master clock of the network.

5.2 Virtualization Network Overhead

In this first part, we evaluate the *virtualization overhead* associated with containerization by comparing the performance of the vPLC (1) containerized within our framework and (2) running bare-metal on the same hardware. In both cases, the vPLC is configured to publish OPC-UA messages with a cycle of 25 µs, a typical value in the most demanding industrial scenarios (§ 5.3). The test measures two representative indicators of time-sensitive communications: end-to-end latency and jitter. The end-to-end latency of a message is defined as the time interval between the transmission time set by the publisher and the actual reception time by the OPC-UA subscriber. The jitter measures how much the actual arrival time of each message differs from the expected arrival time: more precisely if t_i is the arrival time of the i-th message, its jitter is defined as $Jitter(i) = t_i - (t_{i-1} + T)$, where T is the transmission period (in this work, $T = 25\mu s$).

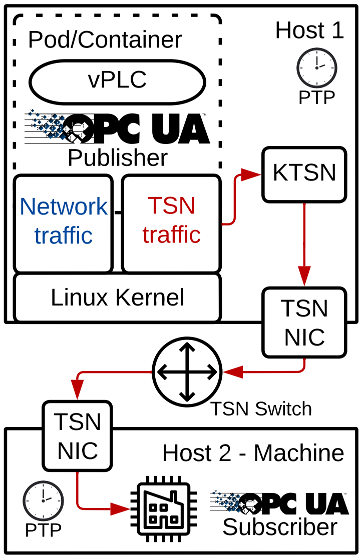
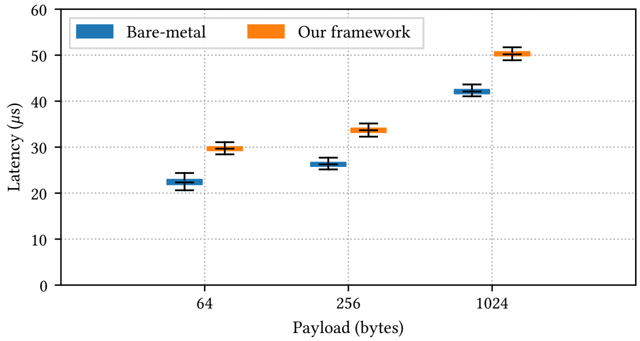


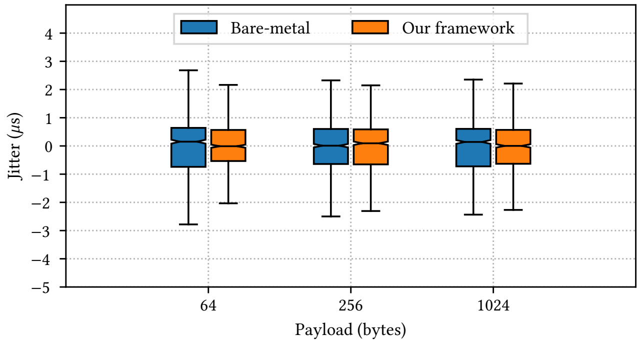
Figure 3: Schematic representation of the testbed.

Figure 4 reports the end-to-end latency and jitter measured for the two considered cases and for three typical payload sizes (64 B, 256 B, 1024 B). A first consideration is that the performance of the containerized version of the vPLC is always very good (orange boxes in Figure 4a), with median latency values ranging from 29.7 µs in the case of small packets (64 B) to 50.1 µs for 1024 B. These values are very close to those registered for the bare-metal deployment, showcasing a constant difference of about 7.6 µs, whereas latency variability is negligible in both cases. The constant performance difference originates in the additional network steps required for packets to reach the network in the containerization case: instead of being directly sent and received on the wire, in our framework they have to cross the KTSN scheduler and a virtual switch, as well as a VXLAN encapsulation step and vice-versa on the receiver side [12]. Nevertheless, the overhead of these steps is minimal.

The performance strength of our approach is even clearer by considering jitter, reported in Figure 4b. The median value



(a) Latency.



(b) Jitter.

Figure 4: Performance of the test vPLC running bare-metal (green) and within our framework (orange). The experiment is repeated for increasing payload sizes.

is around 0 in all cases, as expected on a deterministic network, but the variability, although minimal, is lower in the containerization case. This is the effect of KTSN: a userspace TSN scheduler introduces less variability than the standard kernel-based version, even in this small-scale experiment with no background traffic to introduce noise.

From these results, we conclude that containerization in our framework introduces minimal overhead in terms of network latency, and even improves determinism by supporting the OT traffic with a more efficient packet scheduler. In the next paragraph, we comment on how these results are suitable for the most demanding industrial control applications.

5.3 Industrial Communication Compliance

We now briefly comment on whether our framework effectively meets its design goals of flexibility and high-performance support for virtualized control applications. On the flexibility side, we execute vPLCs in Docker containers managed by Kubernetes, on a general-purpose operating system and COTS hardware, adopting standard communication protocol stacks. These are all open-source resources easy to integrate with IT platforms: hence, we consider meeting the goal of an open and vendor-independent framework for vPLCs. On the performance side, previous work [17] considers that the most demanding industrial applications, such as closed-loop motion control, require cycles under 1 ms with a jitter of at most 1 µs. Our evaluation proves that vPLCs within our framework can support even significantly shorter cycles (25 µs), with a jitter below the 1 µs for more than 90% of the times (Figure 4b), despite not being co-located with the controlled machines as in traditional PLC deployments. Therefore, our framework successfully enables vPLCs to also meet the strictest performance requirements of the OT traffic, thus paving the way for full integration of OT and IT in the next-generation industrial control infrastructures.

6 CONCLUSIONS AND FUTURE WORK

Toward the Industry 4.0 goal of open industrial technologies, this paper proposed a framework based on open-source tools (containers, Kubernetes, OPC-UA, TSN, KuberneTSN) to support the demanding needs of virtualized PLCs (vPLCs) without limiting their distinguishing flexibility. The evaluation of a vPLC on a real industrial testbed showed that our framework introduces minimal overhead, improves determinism, and still retains all the advantages of virtualization.

The work reported here should be understood as a first snapshot, as we are actively enriching our framework with additional capabilities. Short-term goals include at the OT layer the integration of monitoring capabilities within the TSN fabric, and at the IT layer the adoption of smarter forms of container orchestration for the dynamic reconfiguration of the entire control infrastructure. We also plan to assess the scalability of our solutions in more complex settings involving multiple vPLCs and switches. In the longer term, we are interested in the integration of our solution within existing open initiatives for PLC software development (e.g. OpenPLC [2]) and with existing standards for safety regulations and requirements (e.g., IEC 62443 [20]), so as to offer a single support platform from PLC development to operationalization. Finally, the recent availability of lightweight forms of Artificial Intelligence and Machine Learning motivates our research for further performance improvements in order to possibly accommodate AI logic directly within virtualized control components.

ACKNOWLEDGMENTS

The authors wish to express gratitude to the anonymous reviewers for their constructive comments and suggestions. This work does not raise any ethical issues. This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”) CUP: J33C22002880001.

REFERENCES

[1] A. Nasrallah *et al.* 2019. Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research. *IEEE Communications Surveys Tutorials* 21, 1 (2019), 88–145. https://doi.org/10.1109/COMST.2018.2869350

[2] Thiago Rodrigues Alves, Mario Buratto, Flavio Mauricio de Souza, and Thelma Virginia Rodrigues. 2014. OpenPLC: An open source alternative to automation. In *IEEE Global Humanitarian Technology Conference (GHTC 2014)*. 585–589. https://doi.org/10.1109/GHTC.2014.6970342

[3] Software Defined Automation. 2023. SDA vPLC Framework. (2023). https://www.softwaredefinedautomation.io/virtual-plc-management

[4] Mahyar Azarmipour, Haitham Elfaham, Caspar Gries, and Ulrich Epple. 2019. PLC 4.0: A Control System for Industry 4.0. In *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, Vol. 1. 5513–5518. https://doi.org/10.1109/IECON.2019.8927026

[5] Amjad Badar, David Zhe Lou, Ulrich Graf, Christian Barth, and Christian Stich. 2019. Intelligent Edge Control with Deterministic-IP based Industrial Communication in Process Automation. In *2019 15th International Conference on Network and Service Management (CNSM)*. 1–7. https://doi.org/10.23919/CNSM46954.2019.9012680

[6] Hugh Boyes, Bil Hallaq, Joe Cunningham, and Tim Watson. 2018. The industrial internet of things (IIoT): An analysis framework. *Computers in Industry* 101 (2018), 1–12. https://doi.org/10.1016/j.compind.2018.04.015

[7] Dietmar Bruckner, Richard Blair, M Stanica, A Ademaj, W Skeffington, D Kutscher, S Schriegel, R Wilmes, K Wachswender, L Leurs, et al. 2018. OPC UA TSN-A new solution for industrial communication. *Whitepaper. Shaper Group* 168 (2018), 1–10.

[8] Tiago Cruz, Paulo Simões, and Edmundo Monteiro. 2016. Virtualizing Programmable Logic Controllers: Toward a Convergent Approach. *IEEE Embedded Systems Letters* 8, 4 (2016), 69–72. https://doi.org/10.1109/LES.2016.2608418

[9] OPC Foundation. 2023. Unified Architecture. (2023). https://opcfoundation.org/about/opc-technologies/opc-ua/

[10] The Linux Foundation. 2023. Data Plane Development Kit (DPDK). (2023). https://www.dpdk.org

[11] Brendan Galloway and Gerhard P. Hancke. 2013. Introduction to Industrial Control Networks. *IEEE Communications Surveys & Tutorials* 15, 2 (2013), 860–880. https://doi.org/10.1109/SURV.2012.071812.00124

[12] Andrea Garbugli, Lorenzo Rosa, Armir Bujari, and Luca Foschini. 2023. KuberneTSN: a Deterministic Overlay Network for Time-Sensitive Containerized Environments. In *ICC 2023-IEEE International Conference on Communications*. IEEE, Rome, Italy, 1506–1511.

[13] Andrea Garbugli, Lorenzo Rosa, Luca Foschini, Antonio Corradi, and Paolo Bellavista. 2022. A Framework for TSN-enabled Virtual Environments for Ultra-Low Latency 5G Scenarios. In *ICC 2022-IEEE International Conference on Communications*. IEEE, Seoul, South Korea, 5023–5028. https://doi.org/10.1109/ICC45855.2022.9839193

[14] Christian Gehrmann and Martin Gunnarsson. 2020. A Digital Twin Based Industrial Automation and Control System Security Architecture. *IEEE Transactions on Industrial Informatics* 16, 1 (2020), 669–680. https://doi.org/10.1109/TII.2019.2938885

[15] Omid Givehchi, Jahanzaib Imtiaz, Henning Trsek, and Juergen Jasperneite. 2014. Control-as-a-service from the cloud: A case study for using virtualized PLCs. In *2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014)*. 1–4. https://doi.org/10.1109/WFCS.2014.6837587

[16] Thomas Goldschmidt, Mahesh Kumar Murugaiah, Christian Sonntag, Bastian Schlich, Sebastian Biallas, and Peter Weber. 2015. Cloud-Based Control: A Multi-tenant, Horizontally Scalable Soft-PLC. In *2015 IEEE 8th International Conference on Cloud Computing*. 909–916. https://doi.org/10.1109/CLOUD.2015.124

[17] Michael Gundall, Calvin Glas, and Hans D. Schotten. 2021. Feasibility Study on Virtual Process Controllers as Basis for Future Industrial Automation Systems. In *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, Vol. 1. 1080–1087. https://doi.org/10.1109/ICIT46573.2021.9453651

[18] IReflex. 2023. IReflex Suite for Control Systems. (2023). https://ireflex.io

[19] J. Farkas, L. L. Bello and C. Gunther. 2018. Time-Sensitive Networking Standards. *IEEE Communications Standards Magazine* 2, 2 (2018), 20–21. https://doi.org/10.1109/MCOMSTD.2018.8412457

[20] Erich Kronfuss. 2018. *Industrial cyber security standard-IEC 62443*. Technical Report.

[21] Kubernetes Network Plumbing Working Group. [n. d.]. Multus CNI. ([n. d.]). https://github.com/k8snetworkplumbingwg/multus-cni

[22] Seung-Yong Lee and Minyoung Sung. 2022. OPC-UA Agent for Legacy Programmable Logic Controllers. *Applied Sciences* 12, 17 (2022), 1–20. https://doi.org/10.3390/app12178859

[23] Marko Luksa. 2017. *Kubernetes in action*. Simon and Schuster.

[24] Jacob Mellado and Felipe Núñez. 2022. Design of an IoT-PLC: A containerized programmable logical controller for the industry 4.0. *Journal of Industrial Information Integration* 25 (2022), 100250. https://doi.org/10.1016/j.jii.2021.100250

[25] Lorenzo Patera, Andrea Garbugli, Armir Bujari, Domenico Scotece, and Antonio Corradi. 2022. A Layered Middleware for OT/IT Convergence to Empower Industry 5.0 Applications. *Sensors* 22, 1 (2022), 1–14. https://doi.org/10.3390/s22010190

[26] Julius Pfrommer, Andreas Ebner, Siddharth Ravikumar, and Bhagath Karunakaran. 2018. Open Source OPC UA PubSub Over TSN for Realtime Industrial Communication. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vol. 1. 1087–1090. https://doi.org/10.1109/ETFA.2018.8502479

[27] Federico Reghenzani, Giuseppe Massari, and William Fornaciari. 2019. The Real-Time Linux Kernel: A Survey on PREEMPT_RT. *ACM Comput. Surv.* 52, 1, Article 18 (feb 2019), 36 pages. https://doi.org/10.1145/3297714

[28] GitHub Repository. 2023. KuberneTSN. (2023). https://github.com/MMw-Unibo/KuberneTSN

[29] GitHub Repository. 2023. open62541. (2023). https://github.com/open62541/open62541

[30] Weijia Song, Yuting Yang, Thompson Liu, Andrea Merlina, Thiago Garrett, Roman Vitenberg, Lorenzo Rosa, Aahil Awatramani, Zheng Wang, and Ken Birman. 2022. Cascade: An Edge Computing Platform for Real-Time Machine Intelligence. In *Proceedings of the 2022 Workshop on Advanced Tools, Programming Languages, and PLatforms for Implementing and Evaluating Algorithms for Distributed Systems (ApPLIED '22)*. Association for Computing Machinery, New York, NY, USA, 2–6. https://doi.org/10.1145/3524053.3542741