Low latency and resilient industrial network with next generation SDN and PLC virtualization

Taposh Kumer Sarker
Department of Computer Science
University of Texas at El Paso
El Paso, USA
tsarker@miners.utep.edu

Deepak K. Tosh
Department of Computer Science
University of Texas at El Paso
El Paso, USA
dktosh@utep.edu

MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM) | 979-8-3503-7423-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/MILCOM61039.2024.10773754

***Abstract*—The critical infrastructure and industrial control systems (ICS) typically function in isolation and make use of specialized hardware, software, protocols that are often sourced from heterogeneous vendors. Rise of Industry 4.0 and Industrial Internet of Things (IIoT) means that these isolated old networks are gradually being connected with Internet and other telecommunication networks. This leads to various problems and challenges, such as incompatibility with outside networks, cyber security challenges, limited automation, lack of features or mechanisms demanded by modern devices and sensors. On the other hand an ICS has strict operational requirements from the network, such as reliability, temporal consistency, ruggedness etc. Emerging technology like network function virtualization (NFV), and software defined network (SDN) have excellent opportunity to achieve network resiliency in ICS, however their application has been under-explored. Therefore, we present a novel architecture for industrial control networks (ICN) that take advantage of NFV and SDN to build robust, cost effective, secure, and scalable ICNs. We leverage the novel P4 paradigm to virtualize the network infrastructure and evaluate its effectiveness.**

***Index Terms*—Software-defined Network, Network Function Virtualization, Programmable Data Plane, P4, Industrial Control Network**

I. Introduction

Industrial control system (ICS) is a general term that is used to encompass several types of control systems, including supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC) often found in the industrial sectors and critical infrastructures [1]. Industrial control network (ICN) is a specialized network specifically designed to connect the various components of an ICS. ICSs have become the nervous system of modern critical infrastructure and manufacturing which has significantly attributed to rapid adoption of Industrial-Internet-of-Things (IIoT) technology.

On the other hand the past decade has witnessed a paradigm shift in IT infrastructure management. Traditional practices involved the deployment of individual physical servers for each application. This approach has been supplanted by a more efficient resource utilization strategy through infrastructure virtualization. Network Function Virtualization (NFV) consolidates network functions, traditionally delivered by proprietary hardware appliances, onto standard commercial off-the-shelf (COTS) devices, such as x86-based machines [2]. While NFV can definitely deliver its core benefits independently, when integrated with other popular concepts, such as Software-Defined Networking (SDN), it unlocks significant synergies.

Unlike traditional networks, which rely on dedicated hardware like routers and switches for traffic control, SDN leverages software-based controllers and Application Programming Interfaces (APIs) to communicate with and manage the underlying network infrastructure. Although, SDN has effectively separated the control plane from data plane of the network layer, another innovative idea has emerged that incorporates programmable data planes in SDN. Programmable data planes enables network operators to define custom data plane behaviors for network devices. Protocol-independent Packet Processors (P4) has emerged as the de facto standard for data plane programmability [3]. This open-source language, developed and maintained by a collaborative community, provides a high-level abstraction for programming data plane functionalities.

Cyber attacks on ICSs have risen dramatically because of the operational technology (OT) systems are often IP-based and connected to other networks, or even to the Internet. With these attacks, malicious external actors seek to compromise or steal information from the underlying technologies in industrial processes, such as critical controllers [4]. However, cyber attacks are not the only factors, hardware failures can also contribute to significant downtime. Due to the fact that ICSs need to have a high availability, the network infrastructure needs to be resilient against disruption. Besides resilience an industrial network must ensure that latency of the network is very low and a certain level of quality of service (QoS) is maintained. According to existing research [5], on average a failure can last for one hour, since it will take 30 minutes to summon a technician and another 30 minutes to repair the faulty network module (e.g. switch).

Very little research has been done on how NFV technologies can be incorporated in critical infrastructure. Given the criticality of industrial networks we are eager to investigate whether innovations such as server virtualization, network virtualization, NFV, SDN can address different existing challenges of industrial networks such as resilience, latency and QoS, congestion management, quick recovery from failure etc.

The challenge here is to take advantage of these cutting

979-8-3503-7423-0/24/$31.00 ©2024 IEEE

edge technologies and actually design a solution for ICNs that is practical, cost effective, secure, scalable and maintainable. Such a design will have to address the challenges of practicality, effectiveness, new knowledge, standardization, new workflow etc.

The rest of the paper is organized as the following: In Section II we present various problems and bottlenecks associated with legacy ICNs. In Section III we analyze different research opportunities that have emerged in different areas of ICNs. In Section IV we present a novel NFV-enabled architecture, and in Section V we discuss the design challenges for our proposed architecture. Finally, in Section VI we present preliminary results and then we conclude the paper.

II. Bottlenecks of legacy ICNs

Although recent advances in industrial networking have started to blur the line between industrial and commercial networks, at their cores they each have fundamentally different requirements. The primary function of industrial control system is to control sensors and physical equipment within critical infrastructures, while conventional networks focus on data processing and transfer in domestic and corporate settings. Industrial networks demand extremely high reliability and low failure severity, whereas typical networks can tolerate moderate levels. Moreover, industrial networks often have stringent round-trip time requirements due to real-time process and equipment operations, which are less critical in conventional networks [6].

Such demanding requirements for the industrial networks engender unique challenges that require specialized solutions. In search of workable solutions industrial control networks often deploy proprietary, specialized, often incompatible heterogeneous software and hardware components, which creates other sets of problems. Below we discuss the problems that existing industrial networks face:

- **Hardware-Centric Network Functions:** Reliance on dedicated hardware appliances for network functions (firewalls, intrusion detection) creates inflexibility and management overhead. Scaling network resources to meet changing demands becomes a cumbersome process.
- **Limited Automation:** Traditional ICNs often lack built-in automation capabilities, hindering the ability to automate repetitive tasks and respond dynamically to changing operational needs.
- **Static Configurations:** Traditional ICN configurations are often static and difficult to modify. Changes and reconfigurations in the legacy ICNs are rare and usually done from the top of the pyramid with human intervention [7], which is not ideal for modern IIoT.
- **Demanding Latency Requirements:** Many industrial processes require real-time communication with very strict and often deterministic latency constraints. Traditional networks might struggle to meet these stringent timing requirements due to factors like limited processing power and reliance on fixed configurations.
- **Network Failure Recovery:** Industrial environments can be harsh, with extreme temperatures, electromagnetic interference, and this could lead to failure of networks links/nodes. Even when there are redundancy in networks, traditional networks often need manual intervention to reroute data packets.
- **Difficulty in ensuring QoS:** Now a days ICSs are often part of the industrial internet of things (IIoT). In such environments different applications require access to ICSs and applications often have requirements for different QoS. It will be extremely to ensure such dynamic requirements for traditional networks.

III. Research Opportunities

Some of the recent innovations such as SDN, NFV, programmable data plane etc. discussed in previous sections open the door for new possibilities and opportunities. Below we discuss some of these opportunities in the context of ICNs.

A. Robustness and scalability

Modern SDN runs on virtualized environment. This allows SCADA, HDMI etc. components of ICS to be run on virtualized environment on commodity hardware or on commercial cloud. This also opens avenues for use of Network Function Virtualization (NFV), Virtualized Network Function (VNF) etc. There has also been efforts to virtualize PLCs of industrial control network. Virtualization can help improve robustness of industrial network by enabling faster disaster management, data redundancy, improved security management, scalability, availability, better utilization of resources and logically centralized network management using SDN.

Embedded virtualization can be used to simulate components that function in real-time. Existing results show that container based embedded virtualization can be a solution for virtualized PLC, especially when latency is not that big of a concern [8].

SDN combined with NFV can enhance the availability of ICS environments in case of a network hardware failure and thus it reduces the risk of disruptions and widespread damage. This allows dynamic decision making based on the situation and thus deployment of NFV dynamically. This also makes ICS more maintainable and agile by reducing the amount of required human intervention [9].

It is obvious that maintaining HMI and SCADA system on a virtualized environment can provide significant benefits not only in terms of resource utilization but also in terms of flexibility and robustness [10].

B. Network failure recovery using programmable data plane

SDN's centralized control plane does help fast failure recovery but this is a reactive approach and device flow tables are updated as a consequence of events. With programmable data plane forwarding behaviors can be dynamically activated or deactivated as a consequence of packet-level events and timers, and based on per-flow state maintained by the switch itself [11]. This approach can ensure failure detection based on switches' periodic link probing and fast reroute of traffic without availability of or intervention from the controller.

One of techniques for maintaining high availability of networks is fast failure recovery such as the multiple routing configurations (MRC) algorithm. MRC provides fast failure recovery for IP networks [12] and it can realize fast recovery within a few tens of milliseconds after failure detection. With introduction of programmable data plane some of the functions can be pushed to the programmable switches and thus making it more flexible and relieving the controller from additional load [13].

Even though above mentioned approaches can ensure fast link/node failures, there are still packet losses and these packets have to be retransmitted, which increases recovery time. Programmable data plane opens the possibility of in-network computing and thus ways to deal with packet losses on the switches themselves. In this approach to completely mask the effect of packet loss and the resulting long recovery latency, the network takes the responsibility for packet loss recovery (for example, by caching packets in switches), instead of relying on end-to-end recovery [14].

C. Low latency and QoS

Industrial networks require low latency and a specific Quality of Service (QoS) to ensure proper operations. Low latency ensures time bound responsiveness by minimizing delays in data transmission. QoS complements low latency by prioritizing critical traffic and allocating sufficient bandwidth. This ensures that time-sensitive data packets are delivered reliably and without congestion.

In industrial network it is possible to reduce response times using in-network computing through data plane programming. Architecture and test-bed implementations have been developed and it has been shown that response times can be reduced by 74% while processing, and delay introduced by the data plane processing is insignificant [15].

In recent times services with very strict QoS requirements have emerged. Moreover applications have varying requirements for QoS. In this context programmable data plane in SDN can bring not only innovative but also effective solutions. We can leverage capabilities of SDN and programmable data plane by adding additional necessary information in packet header and make decision on data plane based on that information. In this context can use additional QoS header containing to ensure packet deadlines [16].

D. Congestion monitoring and avoidance

It is possible to develop congestion avoidance methods by harnessing the power of programmable data planes. The programmable switches can be programmed to gather and react to important packet meta-data, such as queue load, while the data packets are being processed. In particular, it is possible to program switches to (1) track processing and queuing delays of latency-critical flows and (2) react immediately in the data-plane to congestion by rerouting the affected flows [17].

E. Enhanced Security

Older ICS networks did not have security as a main design goal; they were designed under the assumption that they would always be physically isolated. The integration of telecommunication networks, higher computational capabilities, and Internet of Things components to ICS networks have created new security challenges. Another source of security vulnerabilities stem from the use of legacy protocols that are not appropriately hardened for open environments. Popular industrial protocols such as Profinet [18] and Modbus [19] lack integrity or authenticity protection mechanisms in their default configurations.

To secure modern ICN several SDN/NFV based frameworks and techniques have been proposed. Andrés F. Murillo et al. proposed a proof of concept of an access control framework for virtual ICS systems in an SDN environment that uses the concepts of NFV [20]. The authors have also proposed methods for attack detection [21] and a denial of service (DoS) defense system for SDN enabled virtual ICN environment [22].

Capabilities of programmable switches can also be used to enhance security of the network. Dominik Scholz et al. analyzed SYN flood attacks and proposed different defense strategies against such attack. Their proposed solutions implement filtering mechanisms on the data plane taking advantage of programming capabilities of programmable data plane [23].

IV. NFV-enabled ICN Architecture

In this context we propose an architecture that takes advantage of recent advancements in the areas of virtualization and SDN. Our three-layer industrial control system architecture leverages SDN and virtualization for improved manageability and scalability. The sensor layer gathers sensor data, while the Control layer transports it using a mix of physical programmable switches and virtual switches. Finally, the supervisory layer runs a SCADA system for monitoring and a HMI for operator interaction; this layer also runs ONOS network operating system which controls the switches. Server and network virtualization provide necessary virtual components for each layer. Commodity servers are used for virtualization. RT hypervisors have been used in this architecture to virtualize PLC which has real time requirements. For regular applications COTS virtualization has been used and vSwitch and other VNFs can be created using this virtualization.

This architecture retains the broad architectural design of traditional ICN, but our architecture heavily use virtualization. In communication layer we deploy vPLCs and vSwitches alongside programmable P4 switches. Our supervisory layer is completely virtual. Figure 1 depicts our proposed architecture. In figure solid lines indicate physical connections whereas dotted lines indicate virtual connections.

The proposed ICN mainly depicts the OT network which can be organized in three main layers, which are described in details below.

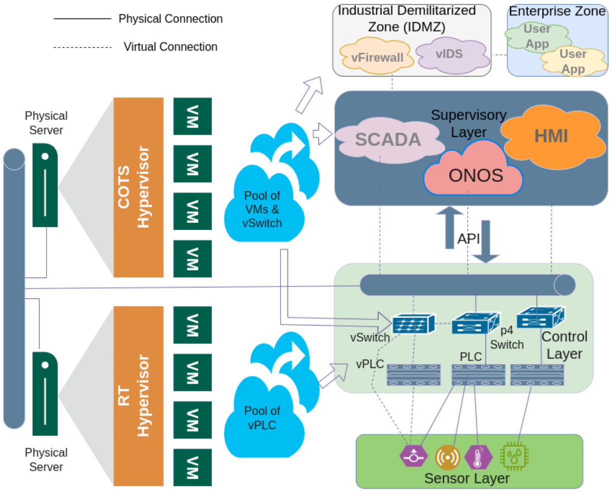

Fig. 1. NFV-enabled ICN Architecture

A. Sensor Layer

This layer contains sensors, actuators, and other field devices. Sensors are connected to either physical PLCs or virtual PLCs through P4 switches which are on the control layer. Through the physical switches these sensors will be virtually connected with the vPLCs and even vSwitches. Sensors would communicate with the control layer using standardized industrial protocols. Common examples include: Modbus, PROFIBUS, Ethernet/IP, MQTT etc.

B. Control Layer

This layer acts similar to a transport layer, facilitating communication between the sensor layer and upper levels. This layer includes:

- Programmable P4 Switches: Offer flexibility by enabling custom processing functions to be implemented directly on the network switches. This can include data filtering, aggregation, or preliminary processing tasks.
- Virtual Switches: Virtual switches enhance scalability by creating software-defined switches within the virtualized environment. Unlike physical switches virtual switches can be deployed very quickly and these switches can dynamically adapt to changing network demands.
- PLCs: PLCs are customized rugged computers. Their custom design is based on the requirements of operations and nature of the sensors they control and communicate with. These generally run real time operating systems and adept in handling real time or near real time data.
- vPLCs: Virtual PLCs perform the same tasks as the physical PLCs. The availability of low-latency deterministic network connectivity for converged Ethernet and real-time hypervisors have made it possible to virtualize components of the PLC architecture [24]. Virtual channels on I/O fabric replaces the dedicated PLC I/O bus to provide deterministic and high-speed networking infrastructure [25]. Even then, vPLCs may not perform as well for real-time operations with very strict requirements, in those cases physical PLCs will be used.

C. Supervisory Layer

Supervisory Layer controls and monitors all physical operations. This layer also provides necessary interfaces to control and monitor sensors and other components. Supervisory layer runs Open Network Operating System (ONOS), which functions as the controller of the SDN network. This layer also runs different supervisory applications such as Supervisory virtual Control and Data Acquisition (SCADA) server and virtual Historian server etc. This layer leverages virtual environments for enhanced manageability:

- ONOS is an open-source software platform that acts as the brain of a Software-Defined Network (SDN). It centralizes control by providing northbound APIs for applications to interact with the network and southbound protocols to communicate with network devices (switches).
- SCADA System (Supervisory Control and Data Acquisition): Runs within a virtual machine, allowing for centralized monitoring, data visualization, and historical data analysis.
- HMI System (Human-Machine Interface): Also virtualized, providing a user interface for operators to interact with the system, monitor real-time data, and issue control commands.

Beyond the supervisory layer, there are Industrial Demilitarized Zone (IDMZ) and Enterprise Zone. The IDMZ delimits the boundary of supervisory layer and consists of firewalls, intrusion detection systems etc. The Enterprise Zone consists of user applications.

V. Design Challenges

Though our proposed architecture offers a lot of benefits in dealing with challenging requirements of ICNs, there are also some challenges. These challenges have to be assessed and handled appropriately based on the requirements and situations. Here we discuss some of the challenges that should be kept in mind while adopting our proposed architecture.

- **Overhead in virtualizing real-time components:** Virtualizing real-time hardware presents unique challenges. Deterministic performance can be hampered by the additional processing overhead introduced by the virtualization layer. Additionally, ensuring hardware resources are consistently available for real-time tasks within a virtualized environment can be difficult. Further investigation on the topic might be needed to estimate the actual potential of PLC virtualization on hard real-time applications [8].
- **Standardization Issues** The programmable data plane landscape is still evolving, with various vendors offering different programming languages and tools. This lack of standardization can create compatibility challenges and hinder interoperability between devices from different vendors.

- **Increased Complexity** Some of the technologies used here are very new and require additional configuration and expertise compared to traditional fixed-function devices. Defining and implementing custom processing functions can be complex, and debugging any errors can be time-consuming. This complexity can also introduce potential security vulnerabilities if not managed carefully.

VI. Preliminary Results

Setup: For our experiment to set up SDN environment with programmable data plane we used open-source tools, such as - Mininet for network emulation, and *bmv2* (behavioral model version 2) as P4 reference switch. This setup facilitates rapid development and testing of SDN applications and P4 programs within a virtualized Mininet environment.

We have performed several experiments and here we describe the results that demonstrate some of the potential benefits of the proposed architecture. Here we describe a simple experiment that allows us to identify congestion and reroute traffic by taking necessary decisions on the data plane using programmable switches.

For our experiment, we used the topology shown in Fig. 2 and we considered three scenarios:

- **Scenario 1:** There is no congestion. Packets flow from one node to another without packet loss and additional delay.
- **Scenario 2:** We introduce congestion in the network and we observe effects of congestion.
- **Scenario 3:** We program P4 switches using P4 language to reroute traffic. We observe and measure effects of routing in data plane.

To simulate *Scenario 1*, we send UDP packets from H1 to H9 and from H4 to H9. Bmv2 switches are reference implementation to be used for research purposes and can handle bandwidth upto around 40 Mbits/s. So we set maximum capacity of each link at 10 Mbits/s to ensure we do not overwhelm the switches. Flow tables have been populated on the switches in a way that direct packets from H1 to take the route H1 → S1 → S3 → S5 → H9, and we will refer to this route as *route 1*. Similarly packets from H4 takes the route H4 → S2 → S4 → S5 → H9 and we will refer to this path as *route 2*. We send packets at the rate of 3 Mbits/s. We measure the jitter and packet loss for both *route 1* and *route 2*.

While packets are being transmitted at 3 Mbits/s through *route 1* and *route 2*, to simulate *Scenario 2* we introduce traffic from H2 to H7 which follows the route H2 → S1 → S3 → H7 and from H5 to H8 which follows the route H5 → S2 → S4 → H8. For these two routes we send packets at 10 Mbits/s and this creates congestion in links S1 → S3 and S2 → S4 since the maximum capacity of the links are 10 Mbits/s. This has been shown in Fig. 2 where red lines indicate congestion. We take measurements of jitter and packet loss for *route 1* and *route 2* the same way we did for *Scenario 1*.

At this point to simulate *Scenario 3* we program the switches using P4 language to detect congestion and reroute packets. When *standard_metadata.enq_qdepth* crosses a threshold in a switch, the program determines that there is a congestion and the program reroutes the packets without any changes in flow entries in switches. Once rerouted, the packets from H1 to H9 takes the route H1 → S1 → S4 → S5 → H9, and we will refer to this route as *reroute 1*. Rerouted packets from H4 to H9 takes the route H4 → S2 → S3 → S5 → H9, and we will refer to it as *reroute 2*. *Reroute 1* and *reroute 2* have been shown in green color in Fig. 2. We take measurements of jitter and packet loss for *reroute 1* and *reroute 2*.

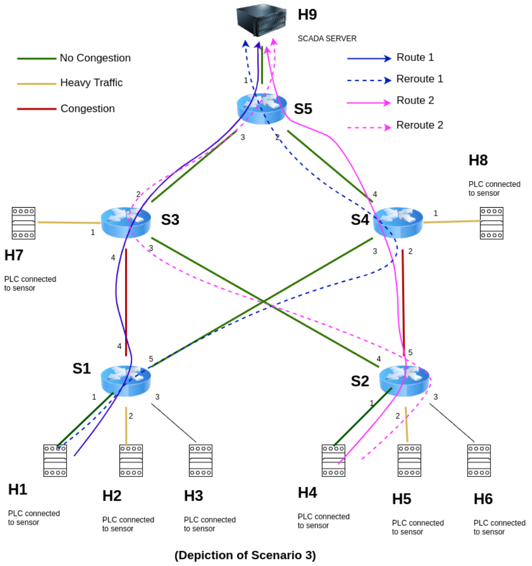

Fig. 2. Topology: network with congestion and rerouting

We have iterated the experiments many times and here we show results for ten consecutive runs of experiments for each scenario, where each experiment ran for 20 seconds. Results for average jitter from ten experiments have been plotted in Fig. 3. We can see when there is no congestion (*scenario 1*) the jitter in *route 1* and *route 2* are around 0.337 ms and 0.352 ms respectively on average. When congestion is introduced for *Scenario 2*, the average jitter increases to around 2.936 ms and 3.081 ms respectively for *route 1* and *route 2*. When we reroute packets for *Scenario 3*, average jitter comes down to around 0.977 ms and 1.058 ms respectively for *reroute 1* and *reroute 2*.

Average packet loss for ten consecutive experiments have been plotted in Fig. 4. For *scenario 1* there is no congestion and we can see the packet loss for both *route 1* and *route 2* is 0%. Once congestion is introduced for *Scenario 2*, packet loss for *route 1* and *route 2* increases to 24.655% and 24.66% respectively on average. Once rerouting kicks in on data plane to simulate *Scenario 3*, the packet loss comes down to 0% again for both *reroute 1* and *reroute 2*.

VII. Conclusion and future directions

Innovations in SDN, NFV, programmable data plane etc. offer huge opportunities for the ICNs to address many of the

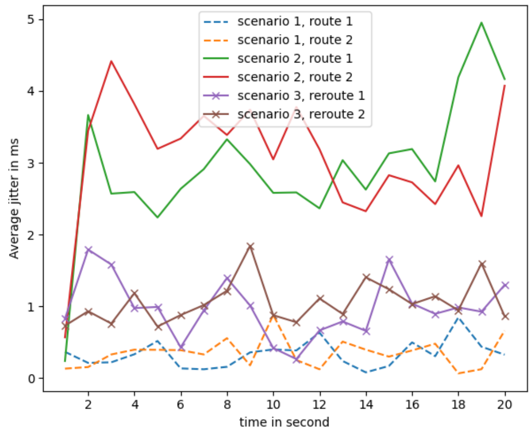

Fig. 3. Average jitter for 10 consecutive experiments

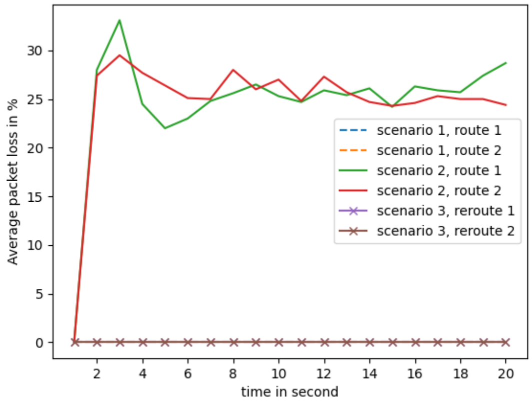

Fig. 4. Average packet loss for 10 consecutive experiments

challenges introduced by today's needs. But they also present a lot of challenges for researchers and engineers when it comes to taking advantage of those specifically for ICNs. Especially bringing all these technologies together and making them work under demanding and harsh environment is a challenge. In this paper we have presented an architecture for future ICNs that incorporate some of the recent innovations. Though the experiments we have done so far are simple, they do validate our design decisions. There is no doubt that more experiments are needed to validate all aspects of our architecture. Our future work will focus on full simulation of our proposed architecture.

References

[1] K. Stouffer, J. Falco, K. Scarfone *et al.*, "Guide to industrial control systems (ics) security," *NIST special publication*, vol. 800, no. 82, pp. 16–16, 2011.

[2] E. NFV, "Network functions virtualisation white paper 3," in *SDN and OpenFlow World Congress*, 2014.

[3] S. Ibanez, G. Brebner, N. McKeown, and N. Zilberman, "The p4-¿ netfpga workflow for line-rate packet processing," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2019, pp. 1–9.

[4] O. Alexander, M. Belisle, and J. Steele, "Mitre att&ck for industrial control systems: Design and philosophy," *The MITRE Corporation: Bedford, MA, USA*, vol. 29, 2020.

[5] D. S. Kim, F. Machida, and K. S. Trivedi, "Availability modeling and analysis of a virtualized system," in *2009 15th IEEE Pacific Rim International Symposium on Dependable Computing*. IEEE, 2009, pp. 365–371.

[6] B. Galloway and G. P. Hancke, "Introduction to industrial control networks," *IEEE Communications surveys & tutorials*, vol. 15, no. 2, pp. 860–880, 2012.

[7] I. Álvarez, D. Bujosa, B. Johansson, M. Ashjaei, and S. Mubeen, "Centralised architecture for the automatic self-configuration of industrial networks," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2023, pp. 1–7.

[8] M. Peltonen, "Plc virtualization and software defined architectures in industrial control systems," 2017.

[9] M. Andreou, J. J. Both, P. Lontorfos, D. Rusek, and C. de Laat, "Improving availability in industrial control systems using software-defined networking," 2021.

[10] J. Reeser, T. Jankowski, and G. M. Kemper, "Maintaining hmi and scada systems through computer virtualization," *IEEE Transactions on Industry Applications*, vol. 51, no. 3, pp. 2558–2564, 2014.

[11] C. Cascone, D. Sanvito, L. Pollini, A. Capone, and B. Sanso, "Fast failure detection and recovery in sdn with stateful data plane," *International Journal of Network Management*, vol. 27, no. 2, p. e1957, 2017.

[12] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, "Multiple routing configurations for fast ip network recovery," *IEEE/ACM Transactions on networking*, vol. 17, no. 2, pp. 473–486, 2008.

[13] H. Miura, K. Hirata, and T. Tachibana, "P4-based design of fast failure recovery for software-defined networks," *Computer Networks*, vol. 216, p. 109274, 2022.

[14] T. Qu, R. Joshi, M. C. Chan, B. Leong, D. Guo, and Z. Liu, "Sqr: In-network packet loss recovery from link failures for highly reliable datacenter networks," in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 2019, pp. 1–12.

[15] A. Atutxa, D. Franco, J. Sasiain, J. Astorga, and E. Jacob, "Achieving low latency communications in smart industrial networks with programmable data planes," *Sensors*, vol. 21, no. 15, p. 5199, 2021.

[16] B. Turkovic, S. Biswal, A. Vijay, A. Hüfner, and F. Kuipers, "P4qos: Qos-based packet processing with p4," in *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*. IEEE, 2021, pp. 216–220.

[17] B. Turkovic, F. Kuipers, N. van Adrichem, and K. Langendoen, "Fast network congestion detection and avoidance using p4," in *Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies*, 2018, pp. 45–51.

[18] W. Su, A. Antoniou, and C. Eagle, "Cyber security of industrial communication protocols," in *2017 22nd IEEE International conference on emerging technologies and factory automation (ETFA)*. IEEE, 2017, pp. 1–4.

[19] T. H. Kobayashi, A. B. Batista, A. M. Brito, and P. S. M. Pires, "Using a packet manipulation tool for security analysis of industrial network protocols," in *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*. IEEE, 2007, pp. 744–747.

[20] A. F. Murillo Piedrahita, "Securing virtual industrial control systems using sdn/nfv platforms," 2019.

[21] A. F. Murillo, L. F. Cómbita, A. C. Gonzalez, S. Rueda, A. A. Cardenas, and N. Quijano, "A virtual environment for industrial control systems: A nonlinear use-case in attack detection, identification, and response," in *Proceedings of the 4th Annual Industrial Control System Security Workshop*, 2018, pp. 25–32.

[22] A. F. M. Piedrahita, S. Rueda, D. M. Mattos, and O. C. M. Duarte, "Flowfence: a denial of service defense system for software defined networking," in *2015 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, 2015, pp. 1–6.

[23] D. Scholz, S. Gallenmüller, H. Stubbe, and G. Carle, "Syn flood defense in programmable data planes," in *Proceedings of the 3rd P4 Workshop in Europe*, 2020, pp. 13–20.

[24] T. Cruz, R. Queiroz, P. Simões, and E. Monteiro, "Security implications of scada ics virtualization: Survey and future trends," in *Proc. 15th Eur. Conf. Cyber Warfare Security (ECCWS)*, 2016, pp. 74–83.

[25] T. Cruz, P. Simões, and E. Monteiro, "Virtualizing programmable logic controllers: Toward a convergent approach," *IEEE Embedded Systems Letters*, vol. 8, no. 4, pp. 69–72, 2016.