future internet

Article

An SDN-Enabled Architecture for IT/OT Converged Networks: A Proposal and Qualitative Analysis under DDoS Attacks

Luca Foschini , Valentina Mignardi, Rebecca Montanari and Domenico Scotece *

Department of Computer Engineering, University of Bologna, via Risorgimento 2, 40136 Bologna, Italy; luca.foschini@unibo.it (L.F.); valentina.mignardi@studio.unibo.it (V.M.); rebecca.montanari@unibo.it (R.M.)
* Correspondence: domenico.scotece@unibo.it

Abstract: Real-time business practices require huge amounts of data directly from the production assets. This new thirst for accurate and timely data has forced the convergence of the traditionally business-focused information technology (IT) environment with the production-focused operational technology (OT). Recently, software-defined network (SDN) methodologies have benefitted OT networks with enhanced situational awareness, centralized configuration, deny-by-default forwarding rules, and increased performance. What makes SDNs so innovative is the separation between the control plane and the data plane, centralizing the command in the controllers. However, due to their young age, the use of SDNs in the industry context has not yet matured comprehensive SDN-based architectures for IT/OT networks, which are also resistant to security attacks such as denial-of-service ones, which may occur in SDN-based industrial IoT (IIoT) networks. One main motivation is that the lack of comprehensive SDN-based architectures for IT/OT networks making it difficult to effectively simulate, analyze, and identify proper detection and mitigation strategies for DoS attacks in IT/OT networks. No consolidated security solutions are available that provide DoS detection and mitigation strategies in IT/OT networks. Along this direction, this paper's contributions are twofold. On the one hand, this paper proposes a convergent IT/OT SDN-based architecture applied in a real implementation of an IT/OT support infrastructure called SIRDAM4.0 within the context of the SBDIOI40 project. On the other hand, this paper proposes a qualitative analysis on how this architecture works under DoS attacks, focusing on what the specific problems and vulnerabilities are. In particular, we simulated several distributed denial-of-service (DDoS) attack scenarios within the context of the proposed architecture to show the minimum effort needed by the attacker to hack the network, and our obtained experimental results show how it is possible to compromise the network, thus considerably worsening the performance and, in general, the functioning of the network. Finally, we conclude our analysis with a brief description on the importance of employing machine learning approaches for attack detection and for mitigation techniques.

Keywords: software-defined network; denial of service; IIoT; IT/OT convergence; SDN security issues

check for updates

Citation: Foschini, L.; Mignardi, V.; Montanari, R.; Scotece, D. An SDN-Enabled Architecture for IT/OT Converged Networks: A Proposal and Qualitative Analysis under DDoS Attacks. *Future Internet* **2021**, *13*, 258. https://doi.org/10.3390/fi13100258

Academic Editors: Maarit Tihinen and Athanasios Panagopoulos

Received: 5 August 2021
Accepted: 5 October 2021
Published: 8 October 2021

1. Introduction

Nowadays, almost everything is connected and accessible anytime and anywhere. Computer networks are in constant evolution, growing up day by day very rapidly, facing challenges in scalability, mobility, and security [1] to be ready for the innovations of next-generation networks [2]. Traditional IP networks are becoming unsuitable to address emerging challenges due to their management complexity. It is both difficult to configure a traditional IP network according to predefined policies and to reconfigure it to respond to faults, loads, and changes [3], and this has sharply restricted such evolutions.

The smart factory scenario is currently under great consideration by both big companies and governments, pushed by the expectation of a ground-breaking change of perspective in the manufacturing sector. Different from traditional IoT devices that have been used in consumer applications for years, industrial IoT (IIoT) devices exchange data over the network by exploiting a variety of network protocols and making large amounts of new types of data available, contributing to streamlining production [4]. As a result, it has become possible for components and machines to manage production autonomously in a flexible, efficient, and resource-saving manner. Therefore, especially for the manufacturing industries, the automatic collection and management of different data coming from the operation technology (OT) floor and information technology (IT) support became an essential enabler for new business models and opportunities in terms of servitization of the shop floors, enabling core services for I4.0 environments such as predictive maintenance.

To make matters even more difficult, current networks are also vertically integrated; the control and data planes are bundled together. Therefore, vertical integration becomes one of the biggest obstacles to fast evolutions and incessant innovations on both protocols and infrastructures in response to all these needs. The emerging paradigm of software-defined networking (SDN) breaks the vertical integration by separating the network control logic (i.e., the control plane) from the underlying routers and switches that forward the traffic (i.e., the data plane). The paradigm needs to be equipped with complex and proprietary networking devices, as it needs to separate the infrastructure layer (network device) from the control layer (network OS, which provides a central view and control over the network and network services) and the application layer (software or business application) [5]. These centrally managed data flow rules are easier to understand and manage for OT network users, since they view the data paths from a high-level, network-wide perspective.

In parallel, the diffusion of wireless sensor networks, embedded systems, and low-cost sensors has promoted the rise of the industrial IoT (IIoT) [6]. Nonetheless, it should be noted that there are some subtle differences between the concepts of the IIoT and the IoT, since the first one needs more communication bandwidth, less latency, and high resilience to effectively handle real-time big data transmissions. For these reasons, the IIoT platform for Industry 4.0 has been widely studied and employed in various scenarios [7]. Basically, the IIoT concept provides an integrated platform for gathering data from different kinds of sensors, transmitting it to the edge or to the cloud, and updating the system state in the form of a closed-loop system [8]. In this way, the IIoT is capable of effectively detecting failures and triggering maintenance processes. Nevertheless, there are some challenges which need to be addressed, and the most important challenge is securing industrial infrastructure and its components. Cyber-attacks on critical infrastructure and industries raise concerns, as the losses incurred due to such attacks are very huge. The IIoT is sometimes referred to as the integration of IT and the OT layers, where IT comprises the enterprise network and OT involves components interacting directly with the physical machines and extracting data from them. These two components have different security requirements which need to be considered to prevent compromising the IIoT infrastructure [9]. The OT layer involves components interacting directly with the physical machines and extracting data from them. Security here is a key requirement, because malicious intruders can cause extensive damage to people and machinery. On the contrary, the IT layer can have much more relaxed constraints than the OT layer because it contains only some services for data analysis and does not involve physical machines.

However, despite the benefits deriving from the integration of SDN with the IIoT [10], to the best of the authors' knowledge, there are not many works that propose an SDN-based architecture for IT/OT convergence. Therefore, this paper aims to fill that gap by proposing an architectural solution based on the SDN paradigm that aims to merge the IT and OT layers. The proposed architecture is composed of three layers: the machine layer that involves physical machines directly and the OT layer, the control layer that contains the SDN controller functionalities and the IT layer, and the application layer that contains the business applications. Moreover, the proposed architecture is built upon the SIRDAM4.0 support infrastructure for reliable data acquisition and management within IT/OT convergent networks [11], developed within the framework of the SBDIOI40 project. Then, focusing on security aspects that are central issues for industrial networks, especially in the OT layer, we qualitatively analyze the proposed architecture under cyber-attacks.

Indeed, in this field, a qualitative investigation into the causes and possible effects of cyber-attacks is still missing.

In fact, the SDN-enabled IIoT networks are vulnerable to different types of attacks on each level; in other words, the application plane, data plane, and above all the control plane. In general, control system vendors and OT networks have several delays in IT when it comes to advancing cyber-security controls, since such environments have typically been physically separated and rely on specialist hardware and software [12]. From a security standpoint, the ability to secure centralized controllers, espoused as a primary risk to SDNs, becomes a possibility in which the benefits of SDNs and convergence seem to outweigh the negatives. The most widespread attack on these networks and what they suffer the most is the distributed denial of service (DDoS) attack, for which a comprehensive security methodology and framework have not yet been assessed. To investigate this crucial security issue, this paper proposes an SDN-enabled IIoT architecture and evaluates its behavior under DDoS attacks. In particular, we extensively analyze the proposed architecture with three long-term simulations by using the Mininet network emulator tool [13], which shows how the performance of the networks easily degrades when under attack. The simulation is useful for understanding the threshold that allows the architecture to maintain its functioning despite the ongoing DoS attack.

The rest of the paper is organized as follows. Section 2 provides the necessary background material, including a discussion of how standardization efforts have been addressing IT/OT convergence and the security issues affecting the three SDN layers of the application plane, control plane, and data plane. Section 3 provides our proposed SDN-enabled architecture for the IIoT and converged IT/OT. Section 4 provides the descriptions of the simulations and contains the set-up of our experiment environment and performance evaluation. Finally, Section 5 provides the final remarks of this work and mentions our ongoing works.

2. Background

This section provides background definitions for the involved technologies and paradigms and then summarizes the mainstream directions of the literature. In particular, we provide, in a brief, the state of the art and the main efforts from community and standardization entities of IT/OT convergence. Then, we provide background information about the use of SDNs in industrial IoT networks. Finally, we summarize the major security challenges in SDNs, focusing specifically on the denial of service (DoS) issue.

2.1. IT/OT Convergence

In the I4.0 perspective, the fully connected factory concept is a key technology enabler for manufacturing operators to drive the production environments up to an expected development, impacting both the IT and OT layers. The technological improvement of the industrial IoT (IIoT) brings new highly reliable devices with advanced built-in communication capabilities, since new state-of-the-art devices, together with the work machines operating at production sites, generate a huge amount of data. Any division of the organization, from design departments to shop floors, provided with adequate access permission, can take advantage of this unprecedented data depot. The attainment of most goals of the I4.0 transition is bound to the gradual integration of algorithms used by manufacturing machines with information about the surrounding infrastructure. In the literature, this trend is referred to as IT/OT layer convergence, which denotes the decrease in the gap between the manufacturing processes in the shop floor on the one hand and the IT department resources (e.g., storage, networking, and computing facility) on the other.

IT/OT convergence is a topic covered by most authoritative standardization bodies, such as the International Society of Automation (ISA) [14] and the International Electrotechnical Commission (IEC) [15]. In particular, the multi-layer IEC 62264 standard based on the ISA-95 specifications [16] defines an information exchange framework model for enabling the integration of applications running in business and operational departments.

Enterprises complying with that standard can define interfaces between control and business functions, allowing them to make informed decisions on the data to exchange so that costs and risks can be kept low in case of implementation errors. The US Industrial Internet Consortium (IIC) [17] proposes the Industrial Internet Reference Architecture (IIRA) [18] as guidance in developing smart IIoT architectures. The specification strongly stresses the concept of convergence of the IT and OT layers at various levels, underlining the common requirements for tackling the transition from an industrial perspective with proper computer architectural patterns. Other authoritative standards, like the European Reference Architectural Model Industrie 4.0 (RAMI 4.0) [19] and the Chinese Intelligent Manufacturing System Architecture (IMSA) [20], push toward a tighter integration of IT and OT. RAMI 4.0 and IMSA are comprehensive models proposing high-level reference architectures, addressing all Industry 4.0 concepts. The concept of the I4.0 component is of particular interest in the vision depicted by the RAMI 4.0 specification [21]. This concept highlights how the smart factories embracing the I4.0 transition will have smart objects as main actors, with industrial equipment outfitted with the right level of intelligence to allow secure control coming from the highest tiers of architecture (i.e., the managerial ones). Furthermore, the communication layer of the RAMI 4.0 specification connects the I4.0 scenario to a key standard sketching many implementation directions, namely Open Platform Communications Unified Architecture (OPC UA) [22–24]. The OPC UA [25] produced by the OPC foundation [26] collocates in the panorama of I4.0 standard initiatives as a concrete effort to propose low-level and technical specifications. The OPC UA defines a standard set of objects and interfaces, facilitating the interoperability among control processes and manufacturing automation applications [25]. The many challenges of the I4.0 transition forced the OPC foundation to think about the heterogeneity of software and hardware components by proposing a unifying architecture tackling these diversities, namely the Unified Architecture (UA) specification [26] released in 2006 and what beceme the IEC-62541 standard [27]. The OPC UA defines a common infrastructure to exchange information and control industrial processes specifying interactions between applications, communication models, and the semantic structure of the information.

2.2. *SDN for IIoT Networks*

Nowadays, industrial communication is described by the well-known automation pyramid, which shows the hierarchical planes of automation systems [28]. The communication requirements inside various planes of the hierarchy are different from one plane to another, such as the timelines of the horizontal communication inside the same plane, which is one of the most important characteristics. Indeed, networks play a central role in the implementation of smart factories. Principally, the networks shall allow the platform to transmit data, commands, and other information between the shop floors and the cloud. In other words, the networks are used to integrate the internal and external resources in a smart factory. Therefore, it is crucial to achieve high network scalability and flexibility [29]. The best research efforts in this field have been focused on making networks programmable and, consequently, dynamically configurable. The SDNs guarantee low hardware usage by using a software version of a centralized controller, which allows for managing the networks and satisfies the strict network requirements of smart factories [30].

As the IIoT characteristic, the networks become populated by a large number of IoT devices with heterogeneous connections. An example where an SDN is used to facilitate IoT networks can be found in [31], where specific service requirements are translated by a central controller into network requirements (e.g., a minimum data rate or the maximum tolerable delay or packet loss for each separate flow). Wan et al. [10] have proposed an SDN-enabled architecture fitted to the IIoT context able to determine network resource allocation and accelerate information exchange mechanisms through an easily customizable networking protocol. Furthermore, with the support of SDN-based architectures, some innovative industrial applications can be realized through well-defined APIs.

In the last few years, both academia and industry have made a lot of efforts for the standardization of software-defined IIoT networks, but compared with the traditional networks, the complexity of IoT networks is greatly increased. Applying the SDN technology to IIoT networks involves several practical implementation issues [32]. First, there is the design of a southbound interface that is used for the interaction between the controller and the physical layer. Due to the higher device heterogeneity, there is a risk of excessively consuming the devices' energy for exchanging OpenFlow messages with the controller. Second, there is the forwarding plane delay. As the SDN works in a centralized manner, all new data are constantly forwarded toward the controller, this can result in forwarding delays and even packet loss. Lastly, there is the management complexity of the controllers to ensure the whole system's stability. For instance, in order to achieve the programmability of the network, system control applications must be allowed to have several access privileges for system control, which increases the probability of system crashes.

In conclusion, there is no one-size-fits-all solution, and consequently, the SDN paradigm is not suitable for all IIoT scenarios. An SDN applied to the IIoT environments solves several open issues, but its applicability depends on the network's scale and complexity.

2.3. SDN Security Issues

As the controller in the SDN paradigm has a global network view, SDN networks are considered more secure than traditional networks. In particular, in SDN networks, all network elements are capable of collecting and reporting traffic information and statistics. This is considered the big difference from traditional distributed networks, which require several steps (sometimes not possible) to partially infer the state of the network and where only a few network elements are logging traffic information and statistics.

However, the SDN paradigm is still a novelty in the world of industrial networks, as they have not yet maturated resilience to attacks [33]. In fact, each component and level of the SDN architecture is prone to attacks. It is possible to identify four attack vectors [34] (i.e., system layers subject to threats that can be exploited by attackers). The first vector of attack is the trust between controllers and applications. Controllers usually work on some form of Linux-based operating system, so they do not have predefined mechanisms to establish trust. The second attack vector is the presence of physical devices, which can lead to malware infection at runtime. The third vector is the lack of adequate authentication, authorization, and accounting mechanisms. The last attack vector is constituted by the vulnerabilities of the controllers. The number of weak points of the controllers is unfortunately high, and above all, they are all of a fair severity.

In general, the main security issue of the control plane is DoS and DDoS attacks that can turn the SDN controller into a single point of failure, thus taking the network down. Network resource visibility is an important characteristic in the SDN paradigm, but the visibility must be hidden to all applications that do not need controller functionalities. However, the number of security issues in the SDN paradigm is expected to grow in the future due to the adoption of the SDN paradigm in the industrial context. To take full advantage of the SDN paradigm in future industrial networks, it will be very crucial to face all security issues so that countermeasures can be implemented in an efficient way. Consequently, in the following subsections, we propose an overview of the most relevant security issues that involve all the SDN stacks: (1) the application plane; (2) the control plane; and (3) the data plane. A list of the most common security issues in the SDN paradigm is presented in Table 1 [35].

2.3.1. Application Plane Security

One of the key points of the SDN paradigm is the ability to control a network via software. Since most of the network functions can be implemented as SDN applications, malicious applications, if not stopped early enough, can spread havoc across a network.

The main point is that there is no standard or specifications allowing SDN applications to manage network services and functions (i.e., to control the control plane) [36]. Hence, applications can pose a threat to the whole network. The myriad of vendor and third-party applications developed in different independent development environments using different programming models and paradigms could create interoperability limitations and security policy collision. Typically, in OpenFlow-based applications, the applications that manage the control plane are developed by parties other than the controller vendors, and this can lead to a security privilege issue. In fact, these applications can inherit the privileges of access to network resources and can manage the behavior of the network if proper security mechanisms are not provided for malicious activities [37].

Table 1. Major SDN security issues for each SDN layer.

SDN Layer	Security Issues
Application Plane	- Lack of authentication - Fraudulent flow rule insertion - Lack of access control
Control Plane	- DoS attack - Unauthorized controller access - Scalability and availability
Data Plane	- Fraudulent flow rules - Flooding attacks - Controller compromising - TCP-level attacks - Man-in-the-middle attack

Kreutzet et al. [38] demonstrated the need to have secure mechanisms in the SDN paradigm to establish secure connections between the controller and applications. Otherwise, due to the abstractions created by the SDN controller, malicious applications can exploit these abstractions to inject malicious configuration commands and theoretically break the network.

Hartman et al. [39] identified three different classes of applications that can impact network security in the SDN paradigm. First, there are the network-sensitive applications that require particular network characteristics. Second, there are applications that provide services for the network, such as a firewall. Finally, there are applications that combine the characteristics from the first and second classes.

In general, malicious applications can bypass access control and will be a real challenge in SDNs. SDN-aware applications can locate and directly communicate with the SDN controllers, while SDN-agnostic applications can communicate indirectly with application datagrams in specific formats.

2.3.2. Control Plane Security

The controller has a pivotal role in the SDN paradigm, since decisions are made on the control plane in a centralized way (e.g., an OpenFlow controller). This makes it highly targeted for malicious activities. In particular, the controller security can be seen from the point of view of its capability to properly authenticate and isolate applications [36]. Moreover, it is important to separate different applications before providing access to the network information and resources according to their security implications. Indeed, SDN applications have different functional requirements from the underlying controller and data path. For instance, a load balancer application might need network statistics, while an intrusion detection application might need to inspect packet header fields. Consequently, all applications (including both vendor and third-party applications) must have different privileges to access network information and resources. For instance, in [40], the authors presented the concept of participatory networking, which enables users and their applications to take part in SDN network configuration. These kinds of user applications must be analyzed properly before access to the network is provided, having comparatively fewer privileges than vendor applications. Therefore, a customized security enforcement mechanism for various types of applications is required in the northbound API of the controller.

Another important open challenge for available SDN controller implementations is to avoid the controller being the single point of failure. To avoid this, the simplest way is to use a distributed SDN controller approach. Nonetheless, it has been shown that using a distributed SDN controller approach cannot effectively protect the network from a single point of failure [41]. The problem is that the load of a failed controller is processed by another controller until its capacity is not exceeded. This may cause cascading failures of controllers.

Since DoS and distributed DoS attacks are capable of completely breaking an SDN network, from the end user's perspective, they are considered one of the most threatening security issues for the SDN controller. The authors in [42] demonstrated that a DoS attack can leverage the logical separation of the control plane and the data plane. Moreover, another work [43] demonstrated that a DoS attack may cause a breakdown of the SDN controller if an attacker continuously sends IP packets with random headers to the controller. To improve the control plane security, the authors suggested using a multiple controller approach, but nevertheless, each controller may suffer from DoS and DDoS attacks, and thus attack detection mechanisms are required to completely avoid a breakdown of the network. Finally, as described above [41], the distributed SDN controller approach is prone to cascading failure of multiple controllers and is not a complete solution for facing DoS and DDoS attacks.

2.3.3. Data Plane Security

For data plane security issues, attention is paid to OpenFlow flows. The number of flow tables that can be installed in a switch are limited, depending on the switch's capacity. In addition, there are two different ways for installing a new flow rule in a switch: in a reactive way, where the rule is installed once the first packet comes from a new host, or in a proactive way, where rules are installed before a new host sends packets. Since the decision-making capacity is not a task of the switches, first, it is very important to differentiate the benign flows from the malicious flows. Second, a switch's capacity is limited by the number of flows that can be stored in the buffer flow. In particular, the buffer flow is a buffer that stores flow rules until the controller emits flow rules. The buffer flow limitation makes the data plane prone to saturation attacks due to storing malicious flows in the buffer.

Security issues in the SDN control plane can directly impact the security of the data plane [41]. This is because if the security of the SDN controller is compromised, the behavior of the underlying network is consequently compromised. Indeed, the data plane might be unavailable, and consequently, the whole network goes offline if the switches do not receive flow instructions from the control plane due to security issues at the control plane [44]. In particular, several attacks, such as man-in-the-middle and black hole attacks, can exploit the control and data plane separation to maliciously manipulate flows and compromise the OpenFlow rules.

Transport Layer Security (TLS) [45] and Datagram Transport Layer Security (DTLS) [46] are security protocols inserted into the OpenFlow protocol specification for ensuring secure controller–switch communication. However, the latest version of OpenFlow (version 1.4.0) defines the TLS protocol as optional due to its configuration complexity. These complex configuration steps include several aspects, such as the generation of controller and switch certificates, which makes switch vendors skip the TLS support in them.

Consequently, the optional use of TLS in OpenFlow networks opens the way for several types of attacks that can compromise the security of the control channel. For instance, Benton et al. [47] demonstrated that a man-in-the-middle attack can be more lethal in OpenFlow networks compared with traditional networks. This is because without TLS, there is no authentication mechanism in the plain-text OpenFlow TCP control channel, and therefore, a malicious user can take full control of any switches and execute fine-grained sniffing attacks.

2.3.4. Denial of Service in an SDN

Denial of service (DoS) and distributed DoS (DDoS) attacks work by sending many requests for a certain period, congesting the system, and forcing it to slow down and decrease performance until it goes offline. The most affected aspects are the bandwidth, memory, and processing power. To consume all available bandwidth in a network so that normal traffic flow cannot be served and the service goes down, an attacker floods the network with malicious traffic at very high rates. In particular, an attacker may exploit vulnerabilities at the protocol level, such as TCP timeout retransmission [48], congestion control mechanisms [49], and the HTTP keep-alive mechanism [50]. Note that from the last report of Kaspersky (Q4 2020 DDoS attacks report [51]), we registered an increase in the number of DDoS attacks of around 10% compared with the same period in 2019. Even with the number of DDoS attacks is falling compared with Q3 2020, it is still a critical security issue to face.

DoS and DDoS attacks in the context of SDN networks, in addition to conventional risks, can affect specific points of the SDN architecture, causing performance degradation of the whole network [52]. An attacker may create large amounts of malicious traffic with unknown or random protocol headers to trigger table-miss processes in the switch. If a switch does not find a match for such packets, it encapsulates and forwards them to the controller for further processing and decision making. This kind of attack can affect different components and characteristics of the SDN network. First, if an attacker creates a huge amount of malicious traffic, such as with fake headers, a switch has to forward each unknown packet to the controller. This may cause exhaustion of the bandwidth in the communication between switches and the SDN controller. Second, malicious traffic can consume all of a switch's capacity due to continuous table-miss processes in the switches. Finally, for the same reason, the controller may run out of resources and continue to process fake traffic for all packets sent from the switches to the controller.

However, recently in the literature, several solutions have been proposed to face DoS and DDoS attacks in SDN-based networks. For instance, the work presented by Ali et al. [53] proposed an interesting approach based on a mathematical model that identifies compromised hosts. Another interesting work in this field is that of Yang and Zhao [54], which introduced an entire SDN framework based on machine learning algorithms capable of identifying and defending DDoS attacks. Finally, as described in [55], many other works have been proposed to challenge DDoS attacks in SDN-based networks. Some of them use machine learning approaches, while others use information theory approaches, always with the same goal of detecting and mitigating DDoS attacks in an efficient way. Even if this work does not propose approaches to mitigate and detect DDoS attacks, it will be different from the other works in the literature because it investigates the consequences that cyber-attacks may cause in industrial networks. To ensure this, we propose an SDN-based network for IT/OT convergence that is based on a real scenario (see SIRDAM4.0 support infrastructure [11]), and we study its behavior under cyber-attacks.

3. Proposed Architecture

After stating a complete overview of the SDN in the IIoT environments, this section presents our reference architecture and its main features. First, we present our SDN-enabled IIoT reference architecture, continuing then with the description of the studied use case. The presented architecture leverages a multi-layer architecture for several reasons. First, a multi-layer architecture is one of the key enablers for IT/OT convergence, as demonstrated in [9]. Second, it is possible to have different security policies at each layer according to the specifications. Finally, the tasks at each layer are completely different, such as gathering data from the bottom layer to the upper layer for analysis and visualization.

3.1. SDN-Enabled Architecture for IIoT

Figure 1 shows our proposed SDN-enabled IIoT architecture. In order to handle IIoT challenges and enable IT/OT convergence, our proposed SDN architecture is composed of three main structures: the SDN controller, SDN switch and IIoT services, and end user applications. These structures are spread on four layers, those being the machine layer, the operation technology layer, the information technology layer, and the application layer:

1. *Machine Layer*: The bottom layer encompasses the manufacturing machines and is usually equipped with heterogeneous sensor networks. In particular, this layer does not contain any SDN equipment due to strict security requirements. Therefore, this layer generates a vast amount of data that shall be transmitted to the application layer (cloud) in real time, and the network bandwidth and real-time requirements may change depending on the different applications (the SDN controller can change the network behavior dynamically). Moreover, this layer should maintain an elevated level of security in fields where there may be dangerous situations for workers near the machines when automatic actuation is in progress.
2. *Operation Technology Layer*: This layer involves components interacting directly with the physical machines and extracting data from them. Security in these first two levels is a key requirement, because a malicious intruder can compromise the whole operativity of production in addition to the possibility of causing extensive damage to people and the machinery. Moreover, this layer can host submodules that provide computation resources to analyze distributed IIoT data and enable the integration of SDN equipment (i.e., switches) to improve management performances. The IIoT services can include data analysis, computing, and acting as a broker. The IIoT devices are generally limited computation resources. This is why the deployment of a local data analysis service is better in terms of latency. In addition, we propose using an OpenFlow switch in this layer to connect it to the controller (i.e., IT layer). Therefore, the OT layer will be controlled and managed by the SDN controller, which makes the architecture more flexible and scalable. Finally, due to security reasons and time constraints, this level was immediately above the machine layer, and furthermore, we decided to not place SDN controllers in this layer.
3. *Information Technology Layer*: This layer does not directly involve workers and machinery, so it is possible to have more relaxed requirements compared with the OT. Here, the SDN controllers are the glue that realizes the interaction between the machine layer and the application layer. In particular, the SDN controller represents the network control plane, and therefore, we propose the use of a distributed SDN controller that provides scalability for IIoT networks at scale. On the one hand, this layer directly manages the physical devices such as the IIoT devices via several interfaces and protocols (e.g., the southbound interface) and adapts their behavior according to the application requirements. On the other hand, this layer provides information to the application layer through the northbound interface and API. In the context of Industry 4.0, the SDN can customize the services provided according to the application requirements (e.g., data transmission rate and real time constraint).
4. *Application Layer*: This layer provides a set of APIs that can be used to design end user applications, such as shop floor monitoring, predictive maintenance, and also providing an abstraction of data generated by the shop floors.

Finally, we claim that organizing the architecture in layers is a key requirement to provide the right security policy in each layer. Places that directly involve workers and machinery must ensure maximum security. On the contrary, the upper layers should have lighter security policies to allow users to provide applications and services. Moreover, by combining our IIoT deployment with the SDN, which guarantees the separation of the control plane and the data plane, it is possible to manage the network behavior by changing traffic routes.

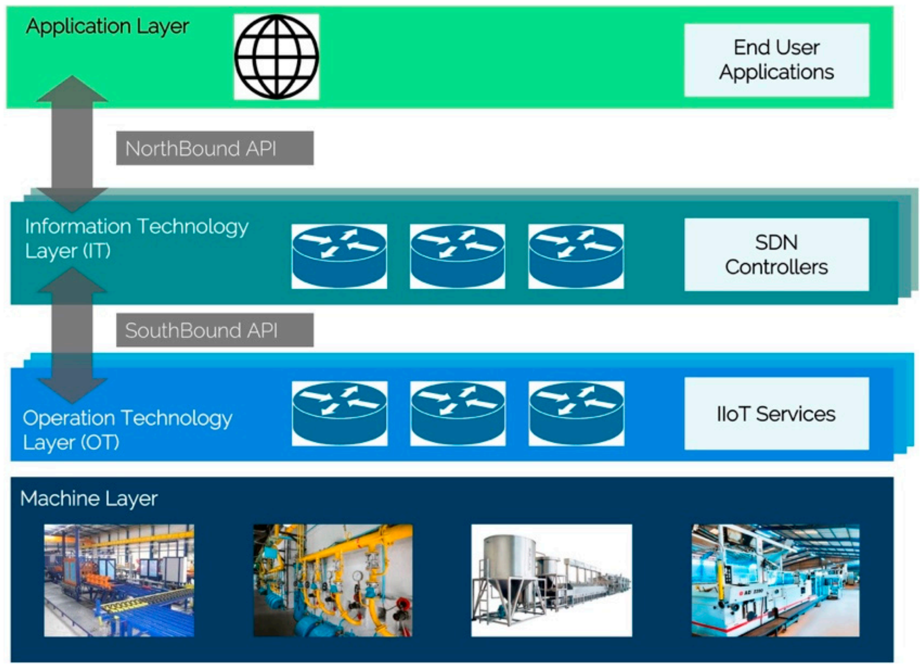

Figure 1. SDN-enabled architecture for the Industrial Internet of Things.

3.2. Use Case

This section introduces our use case in order to illustrate how to leverage SDN technology in IIoT environments, focusing on the security and system reliability aspects. On one hand, the data generated by the IIoT have great commercial value, so secure management of the IIoT devices will be a great challenge to preventing data theft. On the other hand, security is important because humans and production machines work in strict collaboration, and an intrusion may cause serious problems. According to Raptis et al. [56], data management in Industry 4.0 is and will be the key enabler for all those applications that want to take advantage of IIoT devices. As stated before, SDN technology can assist with having dynamic networks in the IIoT environment that face a variety of devices, thus facilitating the extraction of data. Our use case is focused on cases that necessitate a combination of multiple complex accomplishments in terms of data requirements. For example, in a chemical plant, the chemical leakage must be informed in predefined times [57], which means that a decision must be made under specific time latencies in order to guarantee human safety.

In general, data generated by IIoT devices about parameters and abnormal events are processed for decision making, thereby improving production and predicting maintenance and failures for the industrial equipment. Data usually come from sensor devices in small volumes, typically including sensor measurements of various types. Therefore, in IIoT environments, we cannot have high data latency or even data loss. Thus, potential attacks on the SDN network may cause data delivery time increments. Our use case's purpose is to offer an overview of what may happen if an SDN-enabled IIoT network is under attack, highlighting how the network performance degrades. In addition, this work would like to make clear the importance of having secure SDN networks in IIoT environments.

In particular, we devise an Industry 4.0 application aligned with one of the 5G and factories of the future use cases [58], such as video-driven interaction between collaborative robots. Specifically, we focus on a scenario related to managing Autonomous Driving Vehicles (AGVs) that are connected to the 5G infrastructure to perform decentralized decision making, such as object recognition tasks. AGVs are commonly used for the internal movement of goods through an interconnected external logistics system to dynamically manage incoming and outgoing goods flows from the factories. Precisely for this reason, they are considered systems that need a high reliability rate (mostly 99.999%) and a cycle time of less than 2 ms [59]. In particular, industrial AGVs are usually equipped with several sensors for transportation goods to detect obstacles and estimate their positions and orientations. All this will require stable wireless connectivity and resilient computing power at the cloud or edge with the use of wireless and 5G technologies.

4. Simulation and Experimental Results

The objective of the simulations was to analyze the behavior of the IT/OT convergent SDN-based architecture when it was under attack, particularly in the presence of denial of service (DDoS) attacks. The current section first presents the set-up of the simulation environment and then a significant selection of experimental results obtained in our lab deployment scenario. We analyzed the experimental results to measure the latency introduced by the attack that involved the IoT service and the degradation of the channel quality between the client and the server. In this way, the results obtained were a reasonable metric to how SDN security issues affect IIoT networks.

4.1. Simulation Implementation Details

We used the Ryu SDN controller framework [60] as the SDN controller implementation for our testbed due to its component-based implementation, which blends well with the microservices perspective. In particular, we foresee that the integration of detection and mitigation modules will be conducted in a microservices way as external functionalities (like plug-in functions) for the SDN controller core. This is because in resource constraint scenarios, such as in Industry 4.0, we do not want to overload the SDN controller with all functionalities if we do not need them. Furthermore, the management of those modules will be handled by a service orchestrator such as Open Source MANO [61] if the boundaries are 5G networks to dynamically (de)activate functionalities at the SDN controller.

In particular, a Ryu application consists of a Python script that extends the RyuApp base class and implements the observable and observer interface. These interfaces allow the application to interact with the event-based communications in the Ryu framework. First, the Ryu framework starts the application manager that loads both the Ryu-based applications and external applications and registers the associated events. The most important application in the Ryu framework is *ofp_handler*, which allows the framework to interact with the OpenFlow protocol (*OpenFlowController* class). To efficiently communicate with the switches, the Ryu framework creates a virtual representation of switches, called the *Datapath*. Figure 2 shows the logical architecture of the Ryu SDN framework in our proposed architecture. The *ofp_handler* operates as the SDN controller in the IT layer. In particular, it manages the data paths that are at the OT layer. The Ryu apps, once invoked by an event, can reply directly to the data paths.

4.2. DDoS Simulation Attacks

To extensively simulate the network, we carried out up to five relevant DDoS simulations, first in a single-attack scenario and then in a scenario where multiple DDoS attacks occurred at the same time. To run the simulations, we provided two Ubuntu virtual machines—one for the controller and the other for the simulating network—on a machine with an Intel Core i7-4600U processor with 8 GB of RAM. Mininet was used as a support tool for simulating the network and creating the network topology and the hosts. Figure 3 shows the architecture topology that we implemented in Mininet for the preliminary tests; no attacks occurred here. The figure provides an intuition of the network set-up that is required to operate such an edge scenario. For the video streaming task, we used the well-known *FFmpeg* software, which is a Linux-based standard tool for video and audio streaming. The following results are intended to be a basilar comparison between networks without attacks and networks facing attacks. In particular, we calculated the performance of the fully operative network without attacks. We recorded an average throughput of around 4 Mbit/s (the upper bound limit we chose in Mininet) and an average RTT of around 0.2 ms. Finally, in the following experiments, we observed the behavior of the network in an 80-s time window. In each observed experiment, DoS attacks started after around 30 s of the observed window.

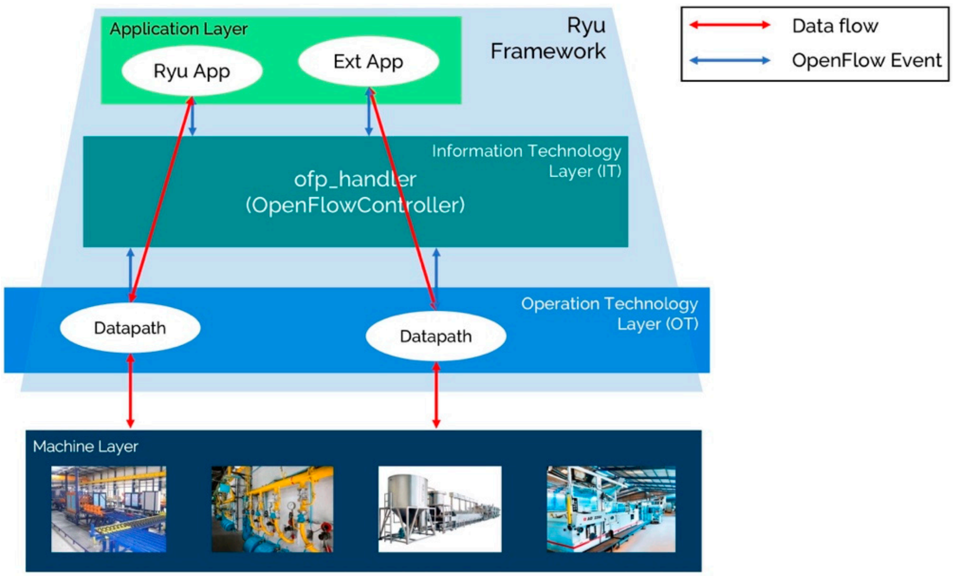
Application Layer
Ryu App
Ext App
Ryu Framework
Data flow
OpenFlow Event
ofp_handler (OpenFlowController)
Information Technology Layer (IT)
Datapath
Datapath
Operation Technology Layer (OT)
Machine Layer

Figure 2. Ryu internal implementation architecture.

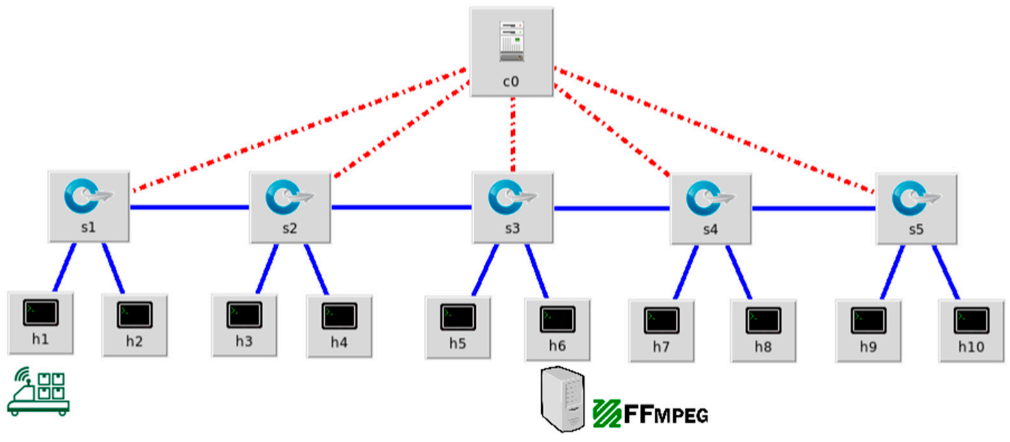

Figure 3. The SDN-enabled architecture used for simulations.

To see how DoS attacks affect possible services in the network, two scripts were used to simulate a streaming service via the *FFmpeg* command. In particular, as shown in Figure 3, host H6 was used as a server and host H1 as a client that was sending an mp4 video to the server continuously, just as in video streaming, to simulate object recognition tasks, for instance. In all simulations, the *hping3* tool was used to simulate DoS attacks to the SDN controller, and the options used are described in Table 2.

Table 2. HPING3 options.

Parameter	Description
–flood	Flooding mode sends packets as fast as possible without taking care to show incoming replies
–rand-source	This option enables hping to send packets with random source addresses
<IP ADDRESS>	IP destination target of the attack

Furthermore, enabling the random source address is a very important option for the attack. In the SDN paradigm, when a switch receives a packet, it checks if there is a flow rule for that packet. If that rule exists, the switch can forward the packet to the destination. Otherwise, the packet is stored in a buffer, and the packet is forwarded to the controller. The use of a random source IP for packets allowed us to generate a large number of packets missing the switches and, consequently, to forward packets to the controller. Indeed, the controller will be flooded with many packets to process and write flow tables back to the switches. Nonetheless, the capacity of a switch's buffer is limited, and some benign rules may be overwritten, thereby rendering the switches completely offline. As a result, many packets are being dropped, causing a bottleneck to the controller, which results in low throughput and a longer delay in the whole network.

A large number of simulations was completed. In particular, five different simulations are presented in this paper in increasing order of relevance. The global goal was to see the network's response to DDoS attacks and hence the idea to simulate several possible scenarios. In particular, we differentiated the experiments by changing the attackers and target hosts in order to figure out how different attack positions affected the service. Moreover, we divided the architecture topology shown in Figure 3 hypothetically into two networks: the first half went from host 1 up to host 6, and the second half went from host 7 up to host 10. This allowed us to have a subnetwork completely agnostic from our service. Finally, a further division was possible in the first half of the network; we could have either attackers or targets that were in the same switches as the client–server hosts. In this way, the TCP flow of the DoS attack affected the TCP flow of our service directly.

We started with a one-host attack, starting from host 7 and targeting host 10 to see the behavior of the network with an attack outside of the client–server path (i.e., attacking the second half of the network that did not affect our service). The case scenario was very simple and observed the infected host 7, which started producing malicious traffic within the network with the aim of slowing down its performance. H7 sent TCP requests every second, using H10 as a target and increasing the flow rate gradually.

The second attack scenario was based on the same topology as the previous case, but in this case, the targeting host was in the first half of the network, which contained our service. H9 was the infected host, and H3 was the target host. Again, in this case, the nodes directly connected to switches S1 and S3 were not involved in the attack. However, H3 was in the middle between H1 and H6.

Then, the third attack scenario referred to the same scenario proposed in the second attack but targeted two hosts from the same switch: H9 and H10 as attackers and H3 and H4 as targets.

The fourth attack involved both nodes that were directly connected to the same switches as the client and server hosts. In particular, attacks started from H8 and point to nodes H2 and H5, allowing us to watch what happened at the service when an attack involved switches that were directly connected to the service.

Finally, the last attack scenario involved up to four hosts, starting from H7 and attacking H10, H5, H3, and H2, involving a host for each switch except the one where the attack started from. All attacks had a rather "slow" approach (i.e., they started producing a fairly modest amount of malicious flow and then went on to increase it as they proceeded). All the results of these simulations are shown and discussed in the following section.

4.3. Simulation Results

From the DDoS attack simulations carried out in the different scenarios, significant slowdowns were recorded as the attacks increased, even going as far as the fall of a controller in the last scenario. All the results were obtained by capturing the network traffic via Wireshark. In particular, we captured the TCP traffic between the client and the server to observe how the service degraded between the two, focusing majorly on the QoS and QoE policies. For instance, we measured the quality of the channel, showing the throughput and the average Round Trip Time (RTT).

The first simulation was the attack outside the client–server path, with H7 attacking H10. In Figure 4, we see how the network degraded after the attacks started, even if it was not really significant. The average throughput was 2.02 Mbit/s.

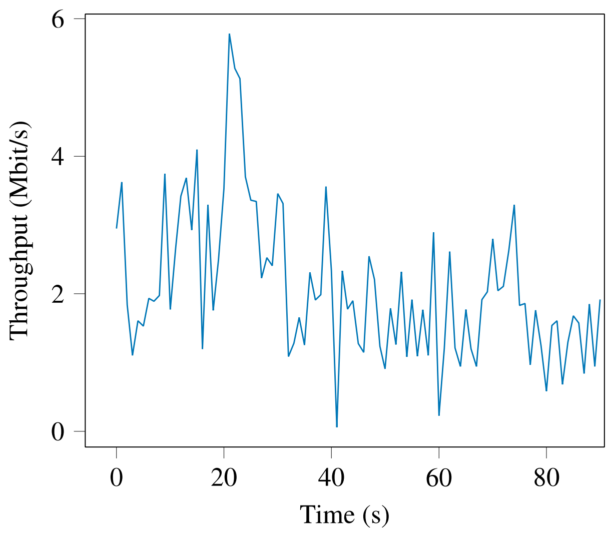

Figure 4. Service channel throughput during H7's attack of H10.

For the attack where H9 attacked H3, the result was very similar, as shown in Figure 5. Even if the attack crossed the client–server path, the decrease in the throughput was modest. The average throughput was 2.49 Mbit/s.

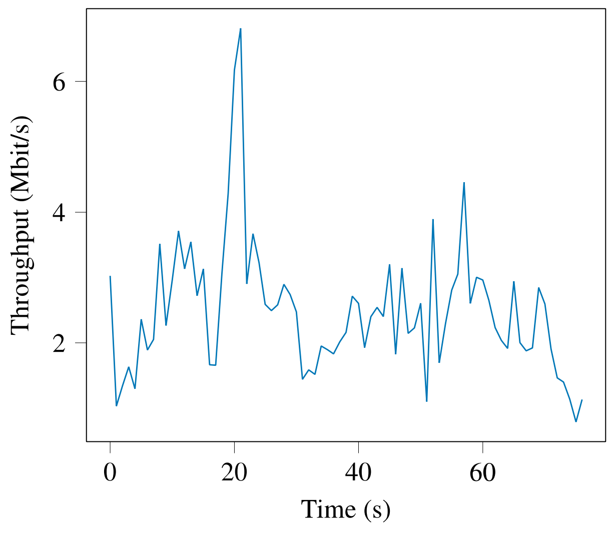

Figure 5. Service channel throughput during H9's attack of H3.

The next scenario involved two host attacks. Figure 6 shows that if we attacked two hosts of the same switch, the results improved in the sense that the networks degraded a little bit more but not significantly. In particular, the attack started from H9 and H10 and targeted H3 and H4. The average throughput was 1.92 Mbit/s.

We then conducted another simulation attacking two hosts, with H8 targeting H2 and H5, H2 being under the same switch as the client, and H5 as the server. In Figure 7, the results show a more degraded communication compared with the other two-host attacks. The average throughput was 1.73 Mbit/s.

Finally, we conducted a four-host attack, with H7 targeting H2, H3, H5, and H10. As shown in Figure 8, this attack was clearly the most powerful, and communication between the client and server was very compromised. The average throughput was 1.44 Mbit/s.

Lastly, Figure 9 shows the average Round Trip Time (RTT) observed in the service channel for all attack scenarios. Similar to the throughput analysis, the average RTT was lower when we increased the number of involved nodes during the attacks. A lower RTT is crucial for latency-sensitive applications, especially in the I4.0 context. The analysis of the packet traces that we collected unveiled that we also observed many more retransmission timeouts (RTOs), which made the delays significantly longer for retransmitting lost packets.

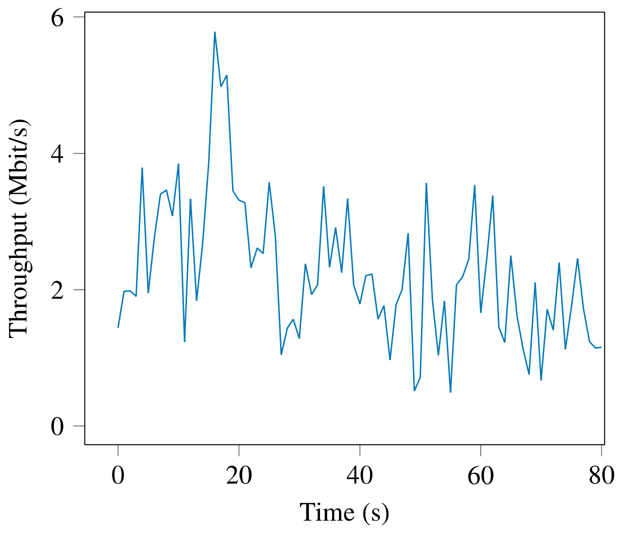

Figure 6. Service channel throughput during H9 and H10's attack of H3 and H4.

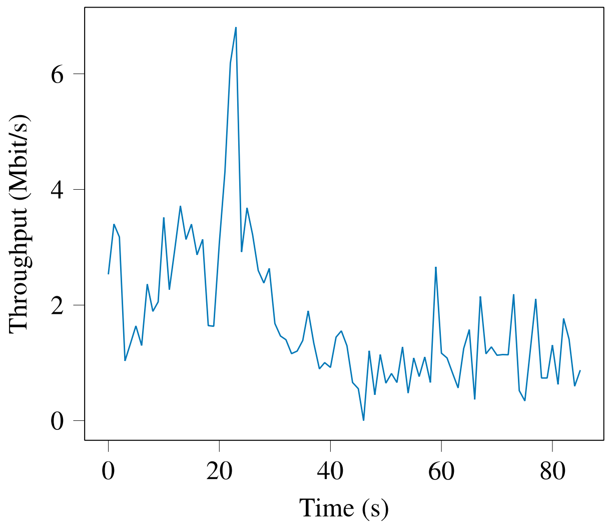

Figure 7. Service channel throughput during H8's attack of H2 and H5.

As shown in Figure 9, the average RTT recorded in the observed time window was around 30 ms for all scenarios. Moreover, as discussed above, most time-sensitive applications in the I4.0 context have a strict cycle time nearly in the range of 1 ms [59]. What we observed was an increase in the average RTT due to a large number of packets that passed through the switches. This behavior can seriously compromise delicate tasks such as remote-control motion, which can endanger workers' lives when working on a farm. Finally, we referred interested readers wanting further experimental results on, for instance, scalability and reliability to the SIRDAM4.0 paper [11], where they can find more details and experimental results about the ongoing implementation.

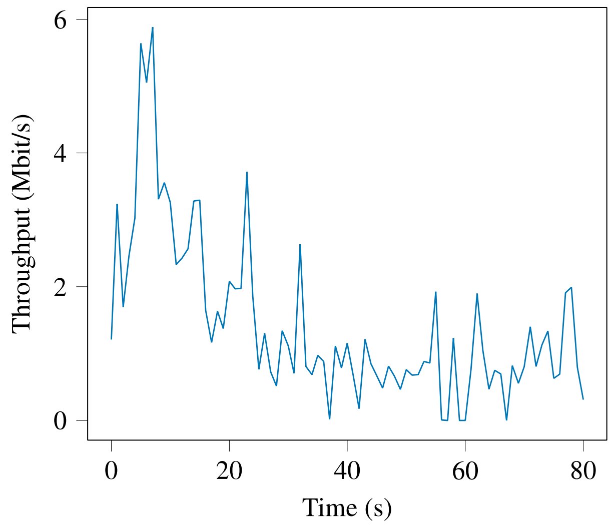

Figure 8. Service channel throughput during H7's attack of H2, H3, H5, and H10.

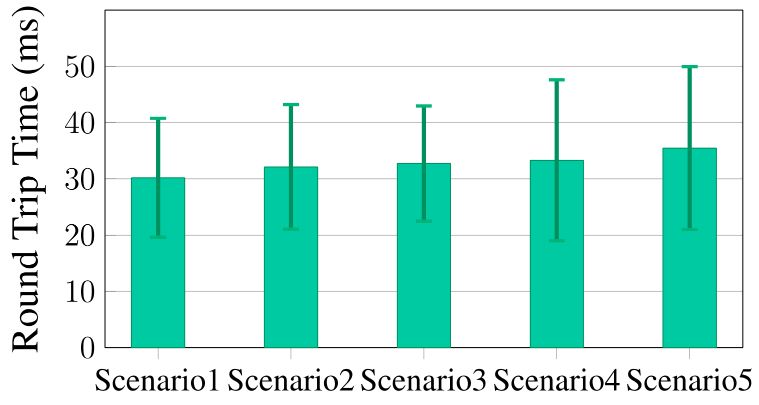

Figure 9. Average Round Time Trip (RTT) between H1 and H6 during attacks.

5. Conclusions and Future Work

This article presented the use of SDN technology for the industrial environments in Industry 4.0. In particular, we investigated SDN technology and how it could be implemented in an IT/OT converged network that enabled process optimization applications and investigated the level of security and performance. The applied simulation scenarios were discussed by focusing mainly on the security aspects. The results from different scenarios showed how serious the effects of a DDoS attack could be, especially in a multi-attack scenario compared with the fully operative network. In particular, the average RTT was substantially increased during the attacks. This is not allowed in the industrial context, especially for motion control applications and for applications that involve humans in their processes in general. Therefore, in such contexts, countermeasures are necessary to avoid security issues both at the IT and OT network levels. However, it is impossible for a single solution to be universally applicable at the same level. For example, strict detection and mitigation strategies should be involved at the OT layer, where several human workers are employed.

For these reasons, there might be some more directions to work upon in the future, such as the safety of the network, which includes detection and mitigation strategies. On the one hand, we are working extensively on detection algorithms based on machine learning (ML) techniques, because we foresee that ML can be a very useful tool to distinguish benign and malicious traffic. While ML is a promising tool to provide generic and automated detection, it is not void of bias. To be effective, learning-based approaches require constant training over time. Both the classification of attackers as well as the notion of normality established for anomaly detection need to be retrained and kept up to date. On the other hand, by decoupling the data plane and the control plane, an SDN is capable of mitigating DDoS attacks. Indeed, we are working on many techniques, such as connection migration, packet migration, limiting inflow bandwidth, adjusting time outs, and controller-to-controller communication protocols, to mitigate DDoS attacks in networks based on SDN architectures.

Author Contributions: Conceptualization, L.F., R.M. and D.S.; Data curation, V.M.; Investigation, D.S.; Writing—original draft, D.S.; Writing—review & editing, L.F. and R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not Applicable, the study does not report any data.

Acknowledgments: This work has been partially funded by Services Big Data In & Out for Industry 4.0: from shopfloor to post-sales (SBDIOI40) CUP: J41F18000040009.

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

References

1. Taheri, J. *Big Data and Software Defined Networks, ser. Computing*; Institution of Engineering and Technology: London, UK, 2018. Available online: https://digital-library.theiet.org/content/books/pc/pbpc015e (accessed on 30 June 2021).
2. Pal, C.; Veena, S.; Rustagi, R.P.; Murthy, K.N.B. Implementation of simplified custom topology framework in Mininet. In Proceedings of the 2014 Asia-Pacific Conference on Computer Aided System Engineering (APCASE), Bali, Indonesia, 10–12 February 2014; IEEE: New York, NY, USA, 2014; pp. 48–53.
3. Kreutz, D.; Ramos, F.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined networking: A comprehensive survey. *Proc. IEEE* **2015**, *103*, 14–76. [CrossRef]
4. Sisinni, E.; Saifullah, A.; Han, S.; Jennehag, U.; Gidlund, M. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4724–4734. [CrossRef]
5. Dhamecha, K.; Trivedi, B. Article: Sdn issues—A survey. *Int. J. Comput. Appl.* **2013**, *73*, 30–35.
6. Wang, S.; Wan, J.; Zhang, D.; Li, D.; Zhang, C. Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination. *Comput. Netw.* **2016**, *101*, 158–168. [CrossRef]
7. Da Xu, L.; He, W.; Li, S. Internet of things in industries: A survey. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2233–2243. [CrossRef]
8. Zhu, R.; Zhang, X.; Liu, X.; Shu, W.; Mao, T.; Jalaian, B. ERDT: Energy-efficient reliable decision transmission for intelligent cooperative spectrum sensing in industrial IoT. *IEEE Access* **2015**, *3*, 2366–2378. [CrossRef]
9. Bellavista, P.; Bosi, F.; Corradi, A.; Foschini, L.; Monti, S.; Patera, L.; Poli, L.; Scotece, D.; Solimando, M. Design guidelines for big data gathering in industry 4.0 environments. In Proceedings of the 2019 IEEE 20th International Symposium on A World of Wireless, Mobile and Multimedia Networks, Washington, DC, USA, 10–12 June 2019; pp. 1–6. [CrossRef]
10. Wan, J.; Tang, S.; Shu, Z.; Li, D.; Wang, S.; Imran, M.; Vasilakos, A.V. Software-defined industrial internet of things in the context of industry 4.0. *IEEE Sensors J.* **2016**, *16*, 7373–7380. [CrossRef]
11. Corradi, A.; Di Modica, G.; Foschini, L.; Patera, L.; Solimando, M. SIRDAM4.0: A support infrastructure for reliable data acquisition and management in industry 4.0. *IEEE Trans. Emerg. Top. Comput.* **2021**, 1. [CrossRef]
12. Barbosa, R.R.R.; Sadre, R.; Pras, A. Flow whitelisting in SCADA networks. *Int. J. Crit. Infrastruct. Prot.* **2013**, *6*, 150–158. [CrossRef]
13. Mininet Team. Mininet an Instant Virtual Network on Your Laptop (or Other Pc). Available online: http://mininet.org/ (accessed on 30 June 2021).
14. International Society of Automation. Available online: https://www.isa.org/ (accessed on 30 June 2021).
15. Iec—International Electrotechnical Commission. Available online: https://www.iec.ch/ (accessed on 30 June 2021).
16. Isa95, Enterprise-Control System Integration. Available online: https://www.isa.org/isa95/ (accessed on 30 June 2021).
17. Industrial Internet Consortium. Available online: https://www.iiconsortium.org/ (accessed on 30 June 2021).
18. Industrial Internet Reference Architecture. Available online: https://www.iiconsortium.org/IIRA.htm (accessed on 30 June 2021).
19. Reference Architecture Model Industrie 4.0 (rami4.0). Available online: https://www.beuth.de/en/technical-rule/din-spec-91345/250940128 (accessed on 30 June 2021).
20. Ministry of Industry and Information Technology. National Intelligent Manufacturing Standard System Construction Guide. 2015. Available online: https://www.cdti.es/recursos/doc/Programas/Cooperacioninternacional/Chineka/Documentacionrelacionada/17668273273201814238.pdf (accessed on 30 June 2021).
21. Ye, X.; Hong, S.H. Toward industry 4.0 components: Insights into and implementation of asset administration shells. *IEEE Ind. Electron. Mag.* **2019**, *13*, 13–25. [CrossRef]
22. Panoramix, M. Opc Ua in the Reference Architecture Model Rami 4.0. Available online: https://opcconnect.opcfoundation.org/2015/06/opc-ua-in-the-reference-architecture-model-rami-4-0/ (accessed on 30 June 2021).

23. There is no Industrie 4.0 Without Opc Ua. 2017. Available online: https://opcconnect.opcfoundation.org/2017/06/there-is-no-industrie-4-0-without-opc-ua/ (accessed on 30 June 2021).
24. Industrie 4.0 Management Group at Zvei Celebrates its Fifth Anniversary and Has a Lot More to Look Forward to. Available online: https://www.zvei.org/en/subjects/industrie-4-0/industrie-40-management-group-at-zvei-celebrates-its-fifth-anniversary-and-has-a-lot-more-to-look-forward-to/ (accessed on 30 June 2021).
25. Opc Foundation. Available online: https://opcfoundation.org/ (accessed on 30 June 2021).
26. Opc Unified Architecture. Available online: https://opcfoundation.org/about/opc-technologies/opc-ua/ (accessed on 30 June 2021).
27. González, I.; Calderón, A.J.; Figueiredo, J.; Sousa, J.M.C. A literature survey on open platform communications (OPC) applied to advanced industrial environments. *Electronics* **2019**, *8*, 510. [CrossRef]
28. Wollschlaeger, M.; Sauter, T.; Jasperneite, J. The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. *IEEE Ind. Electron. Mag.* **2017**, *11*, 17–27. [CrossRef]
29. Shu, Z.; Wan, J.; Lin, J.; Wang, S.; Li, D.; Rho, S.; Yang, C. Traffic engineering in software-defined networking: Measurement and management. *IEEE Access* **2016**, *4*, 3246–3256. [CrossRef]
30. Bizanis, N.; Kuipers, F.A. SDN and virtualization solutions for the internet of things: A survey. *IEEE Access* **2016**, *4*, 5591–5606. [CrossRef]
31. Qin, Z.; Denker, G.; Giannelli, C.; Bellavista, P.; Venkatasubramanian, N. A software defined networking architecture for the internet-of-things. In Proceedings of the 2014 IEEE Network Operations and Management Symposium (NOMS), Krakow, Poland, 5–9 May 2014; IEEE: New York, NY, USA, 2014; pp. 1–9.
32. Hu, L.; Qiu, M.; Song, J.; Hossain, M.S.; Ghoneim, A. Software defined healthcare networks. *IEEE Wirel. Commun.* **2015**, *22*, 67–75. [CrossRef]
33. Prasad, A.S.; Koll, D.; Fu, X. *On the Security of Software-Defined Networks*; IEEE: New York, NY, USA, 2015; pp. 105–106.
34. Kumar, H.; Gupta, P. Sdn security issue and resolution. *Indian J. Appl. Res.* **2017**, *7*, 2.
35. Ahmad, I.; Namal, S.; Ylianttila, M.; Gurtov, A. Security in software defined networks: A survey. *IEEE Commun. Surv. Tutorials* **2015**, *17*, 2317–2346. [CrossRef]
36. Nadeau, T.; Pan, P. Software Driven Networks Problem Statement. NetworkWorking Group Internet-Draft. Available online: https://tools.ietf.org/html/draft-nadeau-sdn-problem-statement-00 (accessed on 30 June 2021).
37. Wen, X.; Chen, Y.; Hu, C.; Shi, C.; Wang, Y. Towards a secure controller platform for openflow applications. In Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking—HotSDN ’13, Hong Kong, China, 16 August 2013; ACM: New York, NY, USA, 2013; pp. 171–172.
38. Kreutz, D.; Ramos, F.M.; Verissimo, P. Towards secure and dependable software-defined networks. In Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics In Software Defined Networking—HotSDN ’13, Hong Kong, China, 16 August 2013; ACM: New York, NY, USA, 2013; pp. 55–60.
39. Hartman, S.; Wasserman, M.; Zhang, D. Security Requirements in the Software Defined Networking Model. Network Working Group Internet-Draft. 2013. Available online: https://tools.ietf.org/html/draft-hartman-sdnsec-requirements-01 (accessed on 30 June 2021).
40. Ferguson, A.D.; Guha, A.; Liang, C.; Fonseca, R.; Krishnamurthi, S. Participatory networking: An API for application control of SDNs. In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM (SIGCOMM ’13), Hong Kong, China, 12–16 August 2013; Association for Computing Machinery: New York, NY, USA; pp. 327–338.
41. Yao, G.; Bi, J.; Guo, L. On the cascading failures of multi-controllers in software defined networks. In Proceedings of the 2013 21st IEEE International Conference on Network Protocols (ICNP), Goettingen, Germany, 7–10 October 2013; IEEE: New York, NY, USA, 2013; pp. 1–2.
42. Shin, S.; Gu, G. Attacking software-defined networks: A first feasibility study. In Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN ’13), Hong Kong, China, 16 August 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 165–166.
43. Fonseca, P.; Bennesby, R.; Mota, E.; Passito, A. A replication component for resilient OpenFlow-based networking. In Proceedings of the IEEE Network Operations and Management Symposium, Maui, HI, USA, 16–20 April 2012; pp. 933–939.
44. Zhang, Y.; Beheshti, N.; Tatipamula, M. On resilience of split-architecture networks. In Proceedings of the 2011 IEEE Global Telecommunications Conference—GLOBECOM, Houston, TX, USA, 5–9 December 2011; IEEE: New York, NY, USA, 2011; pp. 1–6.
45. The Transport Layer Security (TLS) Protocol Version 1.3. Available online: https://datatracker.ietf.org/doc/html/rfc8446 (accessed on 30 June 2021).
46. Datagram Transport Layer Security Version 1.2. Available online: https://datatracker.ietf.org/doc/html/rfc6347 (accessed on 30 June 2021).
47. Benton, K.; Camp, L.J.; Small, C. OpenFlow vulnerability assessment. In Proceedings of the second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking—HotSDN ’13, Hong Kong, China, 16 August 2013; ACM: New York, NY, USA, 2013; pp. 151–152.
48. Shevtekar, A.; Anantharam, K.; Ansari, N. Low rate TCP denial-of-service attack detection at edge routers. *IEEE Commun. Lett.* **2005**, *9*, 363–365. [CrossRef]

49. Luo, X.; Chang, R.K.C. On a new class of pulsing denial-of-service attacks and the defense. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 20 January 2005; pp. 61–79.
50. Adi, E.; Baig, Z.; Lam, C.P.; Hingston, P. Low-rate denial-of-service attacks against HTTP/2 services. In Proceedings of the 2015 5th International Conference on IT Convergence and Security (ICITCS), Kuala Lumpur, Malaysia, 24–28 August 2015; IEEE: New York, NY, USA, 2015; pp. 1–5.
51. Kaspersky. Ddos Attacks in q4 2020. Available online: https://www.kaspersky.com/about/press-releases/2021_a-matter-of-profit-ddos-attacks-in-q4-2020-dropped-by-a-third-compared-to-q3-as-cryptomining-is-on-the-rise (accessed on 30 June 2021).
52. Swami, R.; Dave, M.; Ranga, V. Software-defined networking-based DDoS defense mechanisms. *ACM Comput. Surv.* **2019**, *52*, 1–36. [CrossRef]
53. Ali, S.; Alvi, M.K.; Faizullah, S.; Khan, M.A.; Alshanqiti, A.; Khan, I. Detecting DDoS attack on SDN due to vulnerabilities in OpenFlow. In Proceedings of the 2019 International Conference on Advances in the Emerging Computing Technologies (AECT), Al Madinah Al Munawwarah, Saudi Arabia, 10 February 2020; IEEE: New York, NY, USA, 2020; pp. 1–6.
54. Yang, L.; Zhao, H. DDoS attack identification and defense using SDN based on machine learning method. In Proceedings of the 2018 15th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN), Yichang, China, 16–18 October 2018; Institute of Electrical and Electronics Engineers: New York, NY, USA, 2018; pp. 174–178.
55. Singh, J.; Behal, S. Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. *Comput. Sci. Rev.* **2020**, *37*, 100279. [CrossRef]
56. Raptis, T.P.; Passarella, A.; Conti, M. Data management in networked industrial environments: State of the art and open challenges. *arXiv preprint* arXiv:1902.06141.
57. Kumar, M.; Tripathi, R.; Tiwari, S. Critical data real-time routing in industrial wireless sensor networks. *IET Wirel. Sens. Syst.* **2016**, *6*, 144–150. [CrossRef]
58. 5G and the Factories of the Future. Available online: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Factories-of-the-Future-Vertical-Sector.pdf (accessed on 30 June 2021).
59. Qualcomm. Ultra-Reliable Low-Latency 5G for Industrial Automation. Available online: https://www.qualcomm.com/media/documents/files/read-the-white-paper-by-heavy-reading.pdf (accessed on 30 June 2021).
60. Ryu. Ryu SDN Framework. Available online: https://ryu-sdn.org/ (accessed on 13 September 2021).
61. Open Source MANO. Available online: https://osm.etsi.org/ (accessed on 13 September 2021).