Software defined response and network reconfiguration for industrial control systems

Hunor Sandor, Bela Genge, Piroska Haller, Flavius Graur

Chapter 9

SOFTWARE DEFINED RESPONSE AND NETWORK RECONFIGURATION FOR INDUSTRIAL CONTROL SYSTEMS

Hunor Sandor, Bela Genge, Piroska Haller and Flavius Graur

Abstract The technological shift from isolated industrial control systems to system-of-systems architectures has introduced myriad security challenges. Following popular trends, modern industrial control systems are incorporating technologies such as Industry 4.0, Internet of Things and cloud computing. In these architectures, traditional information and communications hardware and software are glued together with physical components and modern technologies based on IP networks such as software defined networking. The ability of these systems to respond and reconfigure themselves to mitigate faults and attacks is immensely attractive. This chapter proposes a three-tier architecture that implements response and reconfiguration capabilities in an industrial control system. It adopts a software defined network tier for dynamic communications flow (re)configuration and whitelisting, an application tier for the optimal placement of anomaly detection systems and a supervision tier for gluing the three tiers together. The effectiveness and performance of the protection mechanism are demonstrated via use case based qualitative and quantitative assessments.

Keywords: Industrial control systems, security, software defined networking

1. Introduction

Modern industrial control systems are complex and heterogeneous system of systems that offer the advantages of traditional information and communications hardware and software. The pervasive integration of off-the-shelf information and communications technology in the core of industrial control systems has broadened the palette of features and applications, enabled flexible and efficient infrastructures and decreased provisioning and maintenance costs.

Advancements in industrial control systems have also triggered a new technological revolution that has introduced novel applications and services. Industry 4.0 is a most popular initiative that showcases the progress made in the field of manufacturing [13, 18]. These systems integrate Internet of Things technologies and the Internet of Services (IoS) to facilitate remote communications between cyber and physical components, and the interactions of human operators with the underlying infrastructure.

While the shift from isolated industrial control system environments to open and complex technological ecosystems provides numerous benefits, it has had a dramatic impact on security [3]. The integration of traditional information and communications technologies in modern control system architectures has significantly increased the exposure to cyber attacks. In the field of traditional computer systems the effects of cyber attacks are usually limited to the cyber dimension, but in the case of industrial control systems the effects propagate to the physical dimension. As a result, malware infections can impact critical infrastructure assets, causing economic losses and physical damage [7, 12].

The Industry 4.0 initiative has identified three key challenges related to security and resilience: (i) stability; (ii) data privacy; and (iii) cyber security [10]. Therefore, a key requirement in modern industrial control systems is a response and reconfiguration property that can mitigate cyber attacks [8]. The core elements of a response and reconfiguration technique are the detection and isolation of security threats. Implementing these elements requires the deployment of security devices across an industrial control system to support the enforcement of multi-level protection strategies.

This chapter proposes a three-tier security solution that provides semi-automated response and reconfiguration functionality in an industrial control system. The proposed solution comprises: (i) a software defined network (SDN) tier that supports dynamic communications flow (re)configuration and whitelisting; (ii) an application tier that enforces the optimal placement of anomaly detection systems (ADSs); and (iii) a software-assisted human supervision and intervention tier that glues the three tiers together. Several traditional protection mechanisms are integrated and harmonized with the aid of software defined networking, including firewalls, anomaly detection systems and communications flow reconfiguration in response to cyber attacks. To achieve its goal, the proposed solution leverages an optimization strategy that: (i) configures software defined network switches to whitelist permitted communications flows; (ii) minimizes inter-network communications flows to reduce the number of firewall deployments; and (iii) minimizes the number of detection systems that monitor industrial communications. Human operators are an integral part of the solution because they can tune the optimization engine to adjust the reconfiguration results. Indeed, the proposed solution provides comprehensive multi-layer protection via a defense-in-depth strategy.

2. Related Work

Several researchers have investigated the response and reconfiguration problem for industrial control systems. Combita et al. [8] have provided a detailed categorization of the most significant detection and reconfiguration techniques. They discuss possible attacks on sensors and actuators and list reconfiguration strategies based on game theory. The surveyed approaches perform reconfigurations at the physical process level by adapting control loop configurations. In contrast, the novelty of the proposed methodology is that, instead of tuning control loops, it reconfigures communications paths to mitigate the adverse effects of attacks and failures. Furthermore, the proposed methodology delivers an optimal reconfiguration solution that enforces the basic requirements of industrial communications (e.g., shortest routing path to minimize communications delays) and ensures that every data flow is monitored by an anomaly detection system.

The applicability of software defined networking has been proven in diverse areas ranging from disaster preparedness [21] to industrial control (e.g., smart grid [9]) and the Internet of Things [23]. Applications in these areas leverage the technology to enhance communications resilience. Researchers have studied link and node failures in software defined networks [15] and approaches for improving recovery mechanisms [24]. Jin et al. [14] have developed the Dionysus system that performs consistent network updates in a software defined network environment. Dionysus constructs a graph of network update dependencies and schedules updates by taking into account the performance of network switches.

Several researchers have analyzed the adoption of software defined networks in the industrial sector. The benefits have been demonstrated in a test infrastructure comprising an IEC 61850 based power system [20]. Dorsch et al. [9] have highlighted the advantages and disadvantages of using software defined networks in industrial environments. They acknowledge the benefits related to network management, quality of service optimization and system resilience, but raise serious concerns about the increased risks of cyber attacks against the centralized controllers of software defined networks.

Genge et al. [11] have proposed a hierarchical network infrastructure for industrial control systems that leverages software defined networking technology. They engage a network reconfiguration engine that performs traffic optimization to shorten communications paths while maintaining key industrial control communications requirements such as quality of service. The approach has been implemented in a software defined network controller named OptimalFlow. In contrast, this research defines a network optimization problem that focuses on the shortest route objective while considering the industrial control network topology and minimizing the interconnections and communications between segments, thereby minimizing the number of firewall deployments.

Finally, it should be noted that the majority of studies assume complete software defined network deployment scenarios in which all the switches have built-in support for software defined networking. However, such scenarios are rarely, if ever, encountered in real-world industrial control environments. In

fact, deploying such infrastructures would require major investments and architectural changes. Therefore, this research focuses on a more realistic hybrid infrastructure in which software defined network components function alongside traditional network equipment [26].

3. Proposed Security Solution

This section describes the proposed three-tier security solution that provides semi-automated response and reconfiguration functionality in an industrial control system to combat cyber attacks.

3.1 Overview

Given the vast number of recent cyber attacks that have targeted the industrial sector, it is imperative that modern industrial control systems be endowed with the ability to semi-automatically or automatically respond to cyber attacks. A number of technologies have contributed to the security and resilience of communications infrastructures. Significant improvements in protection have been achieved by migrating firewall filtering features into networking equipment such as switches. This enables a network switch to provide valuable defensive functionalities such as enforcing the whitelisting of network traffic flows and dynamically reconfiguring whitelists based on commands received from a central controller.

Software defined networking [22] is a promising advancement in the field of IP networking that has been recently categorized as the "Next Big Technology" [1]. It provides the means to create virtual networking devices and services and implement global networking decisions by decoupling the control plane where routing decisions are made from the forwarding plane that sends network packets to their destinations. Software defined networks often rely on OpenFlow to enable communications with remote devices [24]. OpenFlow ensures remote access to the forwarding plane of a network switch via an open protocol.

Industrial control system response and reconfiguration can be achieved by combining sensing, decision and intervention components. Figure 1 presents the proposed architecture for implementing industrial control system response and reconfiguration. The architecture incorporates three tiers: (i) a communications tier that leverages software defined networking to whitelist traffic based on firewall filtering principles; (ii) an application tier that optimally distributes anomaly detection systems to minimize costs while ensuring effective distributed monitoring of communications flows; and (iii) a human supervisory tier that glues all the layers by receiving anomaly alerts, computing optimal network configurations and deploying the optimal configurations.

Figure 2 presents the conceptual sequence of actions in the proposed protection scheme. The first step involves a human operator who collects the required parameters for initializing the system and computing an optimal network configuration that includes the network topology, flow demands, links, switches and

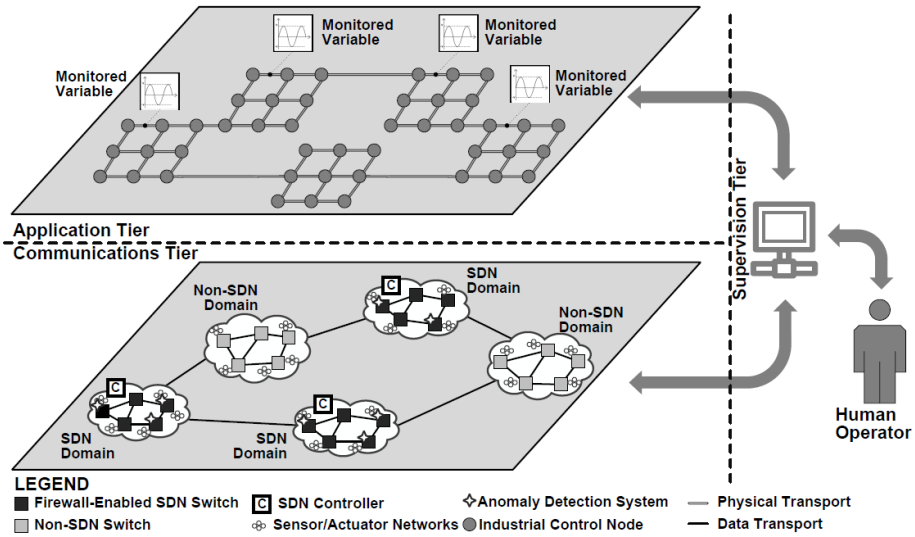


Figure 1. Architecture of the proposed three-tier protection scheme.

anomaly detection systems. In the next step, the operator, with the help of a reconfiguration engine, computes the optimal network configuration and optimal locations for the anomaly detection systems. The optimal whitelisted flow configurations (i.e., forwarding rules) are then applied to the network topology using a software defined network controller; the anomaly detection systems are positioned manually at the prescribed locations. When the network topology has to be changed – in response to a system enhancement or an anomaly alert – the human operator can intervene and change the parameters in order to recompute the optimal network architecture, which is subsequently deployed.

3.2 Communications Tier

The communications tier embodies the basic NIST principles for constructing industrial control system networks [25]. Accordingly, it defines security zones isolated by firewalls, thereby permitting only legitimate (i.e., whitelisted) flows to enter specific zones. The proposed protection scheme assumes a hybrid network infrastructure comprising software defined network and traditional network sub-domains.

Unlike communications flows in traditional information technology networks, communications flows in industrial control networks frequently follow established patterns [4]. In the proposed protection scheme, software defined network switches are used to create security zones and apply flow whitelisting. Thus, the network switches are configured to assume two key roles in the network infrastructure: (i) they function as boundary firewalls that isolate security zones and, thus, implement zone entry conduits (ZECs); and (ii) they function as firewalls by implementing traffic whitelisting.

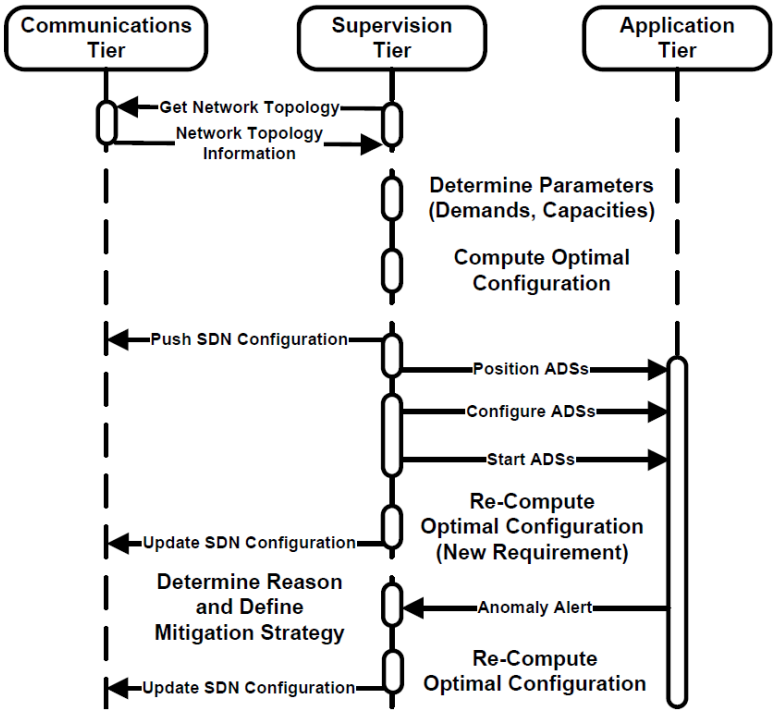


Figure 2. Conceptual diagram of the action sequences in the proposed architecture.

Figure 3 presents an example software defined network enabled topology with security features. In the topology, Switches 8 and 9 incorporate zone entry conduits for Zones 2 and 3, respectively; Zone 1 does not have a zone entry conduit because it has only exiting flows. Switches 1, 2, 3, 5, 7, 8, 9 and 12 serve as whitelisting firewalls; the remaining switches are present for redundancy purposes and to enable the insertion of additional communications flows.

The proposed scheme permits dynamic network reconfiguration in response to system maintenance, system extensions and upgrades, as well as emergency situations such as failures and cyber attacks. The principal advantage of the software defined network enabled zone entry conduits compared with traditional firewalls is that they are more flexible and are easily reconfigured using standard, open communications protocols.

From a practical deployment point of view, industrial control network switches are placed in two categories: (i) traditional dedicated network switches; and (ii) software based (software defined) network switches. The first category of switches include commercial ready-to-use single-box switches while the second category of switches are realized by running software defined networking software on computers equipped with multiple network cards. Generally, a software based switch is realized using a server running a Linux operating system

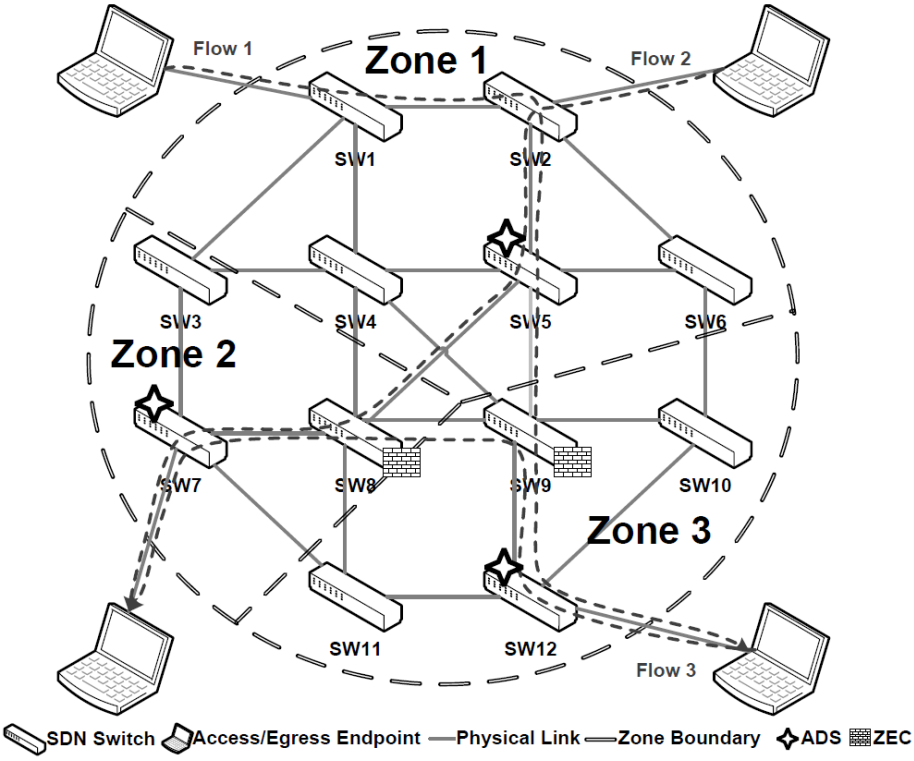


Figure 3. Example software defined network enabled topology with security features.

and dedicated software (e.g., Open vSwitch) that implements the switch features. In addition to its lower cost, a software based switch can run additional software such as an anomaly detection system. Thus, these switches can be transformed into complex protection units with the ability to whitelist flows at the network layer on one hand and the ability to monitor and report abnormal application/traffic behavior on the other hand.

3.3 Application Tier

Protection in the application tier is realized by positioning anomaly detection systems in the network topology, monitoring the behavior of the controlled processes and detecting abnormal events. The ability to implement anomaly detection functionality in software defined network switches greatly increases the flexibility of the security infrastructure by enabling dynamic network reconfiguration – specifically, changing the locations of anomaly detection systems in the network infrastructure. Incorporating an anomaly detection system in a switch enables anomaly-based packet filtering in real time. However, each anomaly detection system can only monitor a selection of flows to avoid congestion and communications delays.

An industrial control system typically has continuous sensor data flows that originate at the network edges. The protection mechanisms in the application tier can readily leverage these sensor data flows to enforce security properties.

Several anomaly detection techniques have been proposed for industrial control systems [6, 16]. A discussion of the properties and applicability of anomaly detection techniques is outside the scope of this research. Instead, an objective of this research is to determine the optimal locations of anomaly detection systems within the network topology in order to enhance communications performance and decrease installation costs.

3.4 Supervision Tier

A human operator resides at the center of the supervision tier. The human operator interacts bidirectionally with the communications and application tiers and can (re)configure the network and anomaly detection infrastructures.

Optimal Network Configuration. The human operator employs an integer linear programming (ILP) tool to obtain the optimal configuration of the industrial control network infrastructure. Note that conventional (i.e., non software defined) segments of the network correspond to static edges between software defined network switches [23].

The optimization problem has three objectives: (i) select the shortest routing paths for whitelisted flows; (ii) minimize the number of anomaly detection systems while satisfying the predefined monitoring requirements; and (iii) minimize the number of boundary switches that realize zone entry conduits in each security zone.

In addition to the three optimization objectives, three requirements are defined in the form of constraints: (i) maximum capacity limits of switches, links and anomaly detection systems cannot be exceeded; (ii) each flow must be monitored in every security zone that it traverses; and (iii) anomaly detection systems and boundary switches can only be positioned at allowed locations.

The remainder of this section formally defines the network optimization problem. Let I be the set of flows, J be the set of software defined switches and Z be the set of security zones. Furthermore, let d_i denote the demand of flow i ($i \in I$) and u_{jl} denote the capacity of link (j,l) ($j,l \in J$).

Assume that, if switches j and l are not connected, then $u_{jl} = 0$. Let u^I be the maximum processing capacity of an anomaly detection system and u^S be the maximum flow processing capacity of a software defined switch. Let g_j^z be a binary parameter with value 1 if switch j belongs to security zone z ($z \in Z$). Let r_z be a binary parameter with value 1 if an anomaly detection system can be positioned in security zone z. Furthermore, let the binary parameters x_{ij}^A and x_{ji}^E have values of 1 if the access and egress end-points of flow i are connected to switch j, respectively.

The optimization problem variables are defined as follows. Let t_{jl}^i be a binary variable with value 1 if flow i is routed on link (j,l). Let p_j be a binary variable with value 1 if switch j can support an anomaly detection system. Let q_j^i be

a binary variable with value 1 if flow i is monitored by an anomaly detection system in switch j. Finally, let f_j be a binary variable with value 1 if switch j acts as a boundary firewall that realizes a zone entry conduit.

The objective function to be optimized is defined as follows:

$$Minimize: \ \alpha \sum_{i \in I, j, l \in J} t_{jl}^{i} + \beta \sum_{j \in J} p_j + \gamma \sum_{j \in J} f_j \tag{1}$$

where α, β and γ are integers that represent the priorities of the three objectives. These parameters can be used by a human operator to tune the optimization problem according to the priorities of the three optimization objectives embodied in Equation (1). Specifically, α expresses the importance of selecting the shortest communications path, β expresses the importance of minimizing the number of anomaly detection systems and γ expresses the importance of minimizing the number of zone entry conduits.

The network design problem specified in Equation (1) is subject to a series of constraints. Due to space constraints, only the most significant constraints are presented.

The following switch capacity constraint ensures that the maximum number of forwarding rules installed in switch j does not exceed the switch capacity:

$$\sum_{i \in I, l \in J} t_{jl}^{i} + \sum_{i \in I} x_{ji}^{E} \leq u^{S} \ \ \forall j \in J \tag{2}$$

The following equation expresses the constraint on anomaly detection system capacity:

$$\sum_{i \in I} d_i q_j^i \leq u^{I} \ \ \forall j \in J \tag{3}$$

The following constraint ensures that the link capacity is not exceeded:

$$\sum_{i \in I} d_i t_{jl}^{i} \leq u_{jl} \ \ \forall j, l \in J \tag{4}$$

The following constraint ensures that an anomaly detection system is positioned in every security zone that requires monitoring:

$$\sum_{jl \in J} g_j^z p_j \geq r_z \ \ \forall z \in Z \tag{5}$$

The following constraint ensures that each flow is monitored as it traverses a security zone if an anomaly detection system in positioned in the zone:

$$\epsilon \sum_{j \in J} g_j^z q_j^i \geq \sum_{jl \in J} (g_j^z + g_l^z) t_{jl}^{i} r_z \ \ \forall i \in I, z \in Z \tag{6}$$

where ϵ is a large integer such that $\epsilon \geq \sum_{i \in I, j, l \in J} t_{jl}^{i}$.

Finally, the following constraint ensures that each flow enters a zone via a zone entry conduit:

$$g_j^y g_l^z t_{jl}^i \leq g_l^z f_l \quad \forall i \in I, j, l \in J, y, z \in Z, y \neq z \tag{7}$$

In order to ensure the proper functioning of the optimized infrastructure, the optimization problem incorporates additional constraints such as the selection of continuous flow paths and the selection of a switch for monitoring purposes only if the switch contains an anomaly detection system.

Solving the optimization problem yields: (i) optimal path – list of edges (software defined switch pairs) associated with each flow; (ii) optimal locations – software defined switches for positioning the anomaly detection systems; (iii) anomaly detection system monitoring rules that define the assignment of a flow to an anomaly detection system; and (iv) list of software defined switches with zone entry conduits.

Optimal Network Reconfiguration. In order to avoid complete network reconfigurations when computing a new optimal solution, the network design problem specified in Equation (1) is extended. The extension is designed to ensure: (i) the minimum number of path changes for each flow; (ii) the minimum number of anomaly detection system migrations; and (iii) the minimum number of changes with respect to boundary switches.

Three binary parameters, $\dot{t_{jl}^i}$, $\dot{p_j}$ and $\dot{f_j}$, are defined to express the optimal solution of an already-deployed network topology. Consequently, the objective function in Equation (1) is redefined as follows:

$$\begin{aligned} Minimize : \alpha \sum_{i \in I, j, l \in J} t_{jl}^i + \beta \sum_{j \in J} p_j + \gamma \sum_{j \in J} f_j + \\ \delta(\sum_{i \in I, j, l \in J} (\dot{t_{jl}^i} - \dot{t_{jl}^i} t_{jl}^i) + \sum_{j \in J} (\dot{p_j} - \dot{p_j} p_j) + \sum_{j \in J} (\dot{f_j} - \dot{f_j} f_j)) \end{aligned} \tag{8}$$

where the δ parameter expresses the priority of the objective that seeks to maintain the existing configuration.

The selection of the value for δ compared with the values for α, β and γ distinguishes between two cases: (i) when δ is greater than the other priority parameters, the network configuration yields a solution in which the changes are minimized; and (ii) otherwise, the existing configuration is changed to benefit from a more optimal configuration of paths, anomaly detection systems and/or zone entry conduits. Note that the constraints are the same as in the specification of the original optimization problem.

3.5 Implementation and Scalability

The OptimalFlow hierarchical software defined network controller [11] was employed to implement the proposed protection scheme. OptimalFlow provides a FlowControl unit that: (i) monitors the underlying domain for changes in the

network parameter values; (ii) changes the set of installed flows according to the solutions to the optimization problem; and (iii) transparently exposes the edge ports of the software defined network to the upper tiers via a software defined switch that is accessible via the OpenFlow protocol.

The proposed solution was integrated with OptimalFlow by changing the initial optimization problem to the new network optimization problem while retaining the original modules responsible for network parameter monitoring and flow installation. An advantage of OptimalFlow is that it enables the provisioning of hierarchical n-tier systems. Consequently, the solution proposed in this research can be scaled to an n-tier architecture as well. Thus, each security zone in the network topology can be reduced to a software defined network sub-domain that is exposed to an upper level as a software defined network switch. Security zones in the next level are represented by optimally-connected virtual software defined network switches.

4. Experimental Results

This section presents the results of the experiments that were conducted to qualitatively and quantitatively assess the performance of the proposed protection scheme. The protection scheme was evaluated using a realistic scenario on an emulated software defined networking infrastructure and simulated industrial data.

The experiments were conducted on a typical industrial network topology. Software defined switches were distributed in security zones structured according to the control system security guide from Tofino Security [5] (see Figure 4). The topology comprised 35 software defined switches connected via 86 physical links that offered redundant communications between two segments. The network topology had thirteen distinct security zones: External Network (X); Corporate Boundary Management (B); Enterprise Network (E); Process Information Network (I); Process Information Management Network (M); Supervisory Networks (J1, J2); Control Networks (C1, C2); and Process Networks (P1, P2).

First, the applicability of the proposed protection scheme to the scenario described above was evaluated using the OptimalFlow software defined network controller. The experiment was conducted using the Mininet network emulator [17]. The $\delta > \beta > \gamma > \alpha$ priority parameter configuration was employed and IEC 61850 network traffic was generated using the MATLAB Power System Analysis Toolbox (PSAT) running the IEEE 14-bus process model [19].

The analysis focused on three use cases: (i) complete network configuration; (ii) partial network configuration in the case of a new flow; and (iii) partial network configuration in the case of link failure.

Four flows, Flows 1-4, were defined for the first use case. Figure 4 shows the optimal configuration. The optimizer minimized the number of anomaly detection systems and zone entry conduits by reducing the number of communications switches used to route flows. On the other hand, switches were selected to ensure the shortest routing path. Obviously, the solution involved

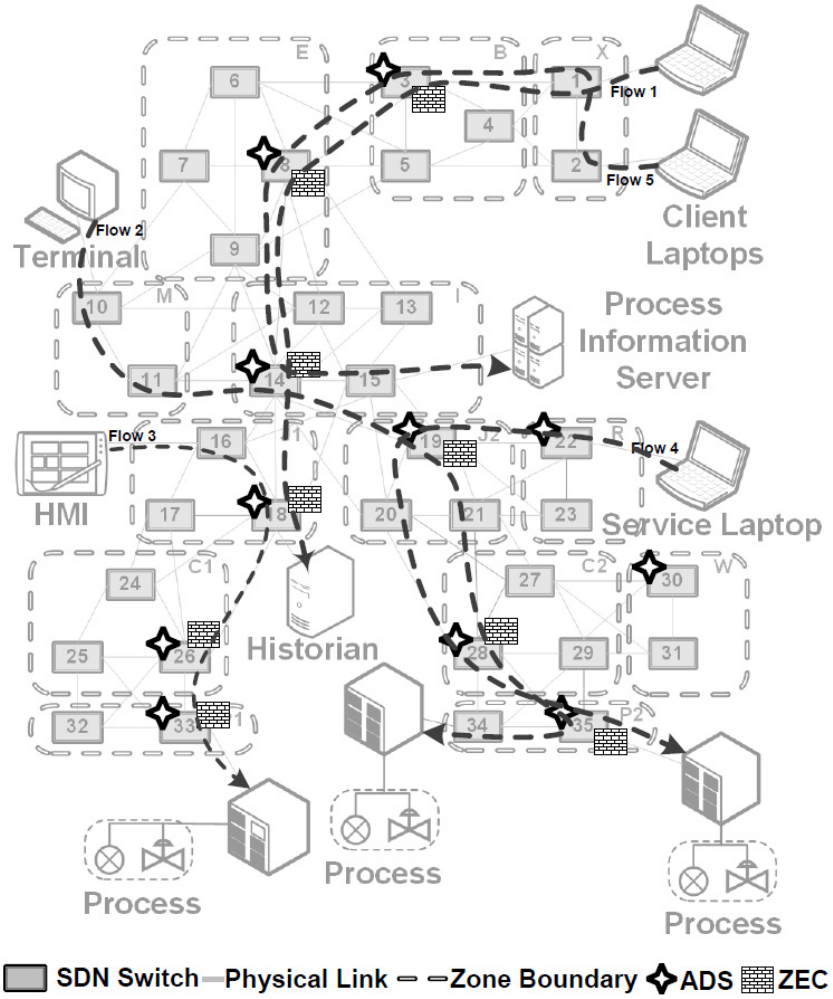


Figure 4. Experimental architecture with an SDN enabled topology.

a trade-off between the shortest routing path and the minimum numbers of anomaly detection systems and zone entry conduits. For example, Flow 2 was routed on a path comprising six links whereas the shortest path only required five links along Switches 10-11-14-20-28-35. This is because the optimizer selected Switch 19 for provisioning an anomaly detection system and as a zone entry conduit. Therefore, the optimizer reduced the number of provisioned anomaly detection systems and zone entry conduits by routing Flow 2 through Switch 19.

In the second use case, a new flow, Flow 5, was added to the network topology and the optimal network configuration was recomputed. Note that the initially-configured flows and the anomaly detection system and zone entry conduit

locations were not modified. This was achieved by setting the δ parameter to be larger than the β, γ, α parameters in Equation (8). Figure 4 shows the results of the optimization and the routing of Flow 5.

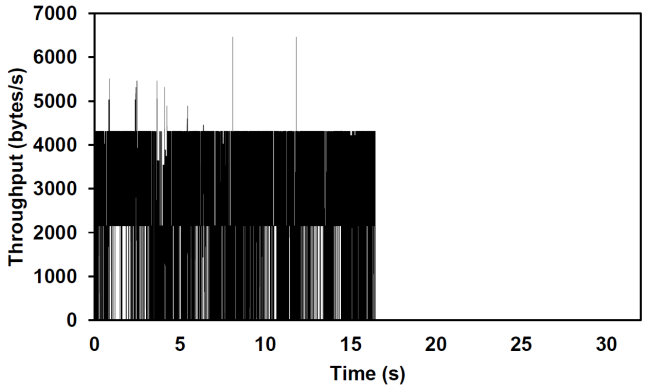
In the third use case, the physical link between Switches 8 and 14 was made to fail by assigning it a capacity of zero. This action automatically triggered the recomputation of the optimal configuration using OptimalFlow. The optimal solution re-routed Flows 1 and 5 on Switches 8, 9 and 14. Subsequently, routing rules were deleted on the link between Switches 8 and 14. The rest of the configuration was not affected because δ was selected to be greater than the other priority parameters.

The optimization and software defined network deployment times were 0.91 s and 1.17 s for the first use case, 1.11 s and 0.76 s for the second use case and 1.18 s and 0.46 s for the third use case, respectively. These results demonstrate that, even if the optimization times are increased slightly when performing network reconfiguration during the second and third use cases, the software defined network times are considerably low. Thus, the total number of reconfiguration operations are also reduced significantly.

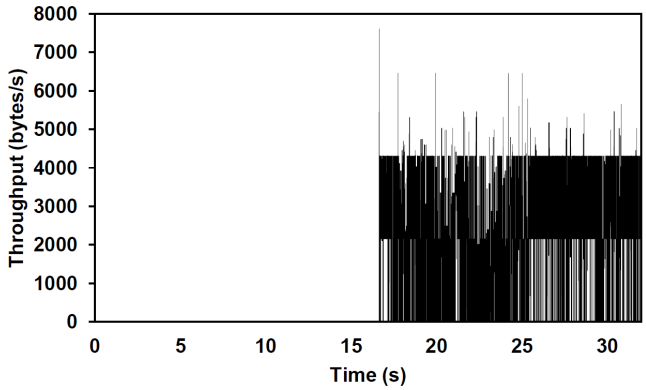
Next, the effects of network reconfiguration on industrial traffic were investigated. The experiment involved the execution of a MATLAB Power System Analysis Toolbox instance and an IEC 61850 server on the host attached to Switch 33. The traffic was monitored using an IEC 61850 client on the machine attached to Switch 16 and by positioning two traffic capturing units, one on the link between Switches 16 and 18 and the other on the link between Switches 16 and 1. In the next step, Flow 3 was re-routed from the link between Switches 16 and 18 to the path involving Switches 16, 17 and 18.

Figure 5 shows the effects of flow re-routing on the traffic monitored at the network segments of interest. Note that the throughput on the link between Switches 16 and 17 was initially around 4.5 KB/s while there was no traffic on the link between Switches 16 and 18. After the re-routing was triggered (after 16 seconds), the traffic throughput exhibited slight changes. During the re-routing, no data was transferred in Flow 3 for a short time period (around 200 ms). This corresponds to the time required to delete the forwarding rules for the failed link and to install new rules for the alternative links. This time period is always shorter than the total software defined network deployment time because it does not cover the communications and processing overhead between OptimalFlow and the software defined network controller. After the re-routing was completed, a temporary increase in the traffic was observed due to the TCP re-transmissions required to recover the lost packets.

Finally, the effects of parameter value changes on the optimization time were investigated. The experiments measured the optimization times for the two topologies presented in Figures 3 and 4 for different numbers of flows. The AIMMS optimization tool [2] was executed on a laptop with an Intel i7 quad core processor, 8 GB RAM and an SSD hard drive. Each configuration was repeated five times. Table 1 presents the optimization times for various parameter values and scenarios for the different configurations.



(a) Traffic between Switches 16 and 17.



(b) Traffic between Switches 16 and 18.

Figure 5. Effects of re-routing network traffic.

Table 1. Optimization times for various parameter values and scenarios.

Flows	**Optimization Time (s)**					
	Topology in Figure 3			**Topology in Figure 4**		
	Min	**Avg**	**Max**	**Min**	**Avg**	**Max**
10	0.09	0.12	0.17	1.59	1.71	1.75
30	0.52	0.66	1.17	5.67	6.51	7.63
50	1.47	1.71	2.05	9.70	10.28	18.89
100	4.34	4.67	6.44	30.47	48.28	56.06

The experimental results demonstrate that the number of flows and the complexity of the topology impact the optimization time. In both the topologies,

increasing the number of flows significantly increases the optimization time. Furthermore, increasing the complexity of the topology (i.e., numbers of software defined switches, links and security zones) also increases the optimization time. Nonetheless, the optimization time never exceeded one minute even in the worst-case scenario and was under ten seconds in the majority of cases.

Note that the optimization time can be reduced further by solving the reduced optimization problem specified in Equation (8) and by partitioning the network to leverage the ability of OptimalFlow to solve hierarchical optimization problems. The combination of these strategies enhances the applicability of the proposed network configuration strategy in real industrial control environments. However, current software defined networking technologies – and, thus, the proposed methodology itself – may not be applicable in scenarios involving time-critical communications where network packets need to be delivered within 10 to 20 ms. Further research is needed to speed up the methodology for network reconfiguration in industrial control environments with tight timing constraints.

5. Conclusions

This research has proposed a novel three-tier protection architecture that endows industrial control systems with response and reconfiguration capabilities to cope with failures and cyber attacks. The communications tier, which leverages software defined networking, enables dynamic flow configuration, reconfiguration and whitelisting; the application tier incorporates a distributed anomaly detection infrastructure; and the supervision tier, which incorporates a human in the loop, controls and synchronizes all three tiers. The protection solution is obtained by optimizing industrial control network flows and the numbers and locations of anomaly detection systems and software defined firewalls. The applicability of the proposed architecture is demonstrated by implementing it using the OptimalFlow hierarchical software defined network controller and conducting qualitative and quantitative assessments involving three use cases.

Future research will focus on developing a more efficient optimization algorithm. Additionally, heuristics will be identified and incorporated in the algorithms to reduce the computational time and enhance their application in industrial control environments with large infrastructures and tight timing constraints.

Acknowledgement

This research was supported by a Marie Curie FP7 Integration Grant within the 7th European Union Framework Programme (Grant No. PCIG14-GA-2013-631128).

References

[1] R. Ackerman, Software-defined networking looms as next big technology, *Signal*, May 12, 2014.

[2] Advanced Analytics, Prescriptive Analytics and Supply Chain Management, AIMMS, Bellevue, Washington (`aimms.com`), 2017.

[3] R. Anderson and R. Hundley, The Implications of COTS Vulnerabilities for the DoD and Critical U.S. Infrastructures: What Can/Should the DoD Do? P-8031, RAND Corporation, Santa Monica, California, 1998.

[4] R. Barbosa, R. Sadre and A. Pras, Flow whitelisting in SCADA networks, *International Journal of Critical Infrastructure Protection*, vol. 6(3-4), pp. 150–158, 2013.

[5] E. Byres, Using ANSI/ISA-99 Standards to Improve Control System Security, Version 1.1, White Paper, Tofino Security, Lantzville, Canada, 2012.

[6] M. Caselli, E. Zambon, J. Amann, R. Sommer and F. Kargl, Specification mining for intrusion detection in networked control systems, *Proceedings of the Twenty-Fifth USENIX Security Symposium*, pp. 791–806, 2016.

[7] A. Cherepanov, BlackEnergy by the SSHBearDoor: Attacks against Ukrainian news media and electric industry, *WeLiveSecurity*, January 3, 2016.

[8] L. Combita, J. Giraldo, A. Cardenas and N. Quijano, Response and reconfiguration of cyber-physical control systems: A survey, *Proceedings of the Second IEEE Colombian Conference on Automatic Control*, 2015.

[9] N. Dorsch, F. Kurtz, H. Georg, C. Hagerling and C. Wietfeld, Software-defined networking for smart grid communications: Applications, challenges and advantages, *Proceedings of the IEEE International Conference on Smart Grid Communications*, pp. 422–427, 2014.

[10] R. Drath and A. Horch, Industrie 4.0: Hit or hype? [Industry Forum], *IEEE Industrial Electronics*, vol. 8(2), pp. 56–58, 2014.

[11] B. Genge and P. Haller, A hierarchical control plane for software-defined-network-based industrial control systems, *Proceedings of the IFIP Networking Conference and Workshops*, pp. 73–81, 2016.

[12] M. Hagerott, Stuxnet and the vital role of critical infrastructure operators and engineers, *International Journal of Critical Infrastructure Protection*, vol. 7(4), pp. 244–246, 2014.

[13] M. Hermann, T. Pentek and B. Otto, Design principles for Industrie 4.0 scenarios, *Proceedings of the Forty-Ninth Hawaii International Conference on System Sciences*, pp. 3928–3937, 2016.

[14] X. Jin, H. Liu, R. Gandhi, S. Kandula, R. Mahajan, M. Zhang, J. Rexford and R. Wattenhofer, Dynamic scheduling of network updates, *ACM SIGCOMM Computer Communication Review*, vol. 44(4), pp. 539–550, 2014.

[15] J. Kempf, E. Bellagamba, A. Kern, D. Jocha, A. Takacs and P. Skoldstrom, Scalable fault management for OpenFlow, *Proceedings of the IEEE International Conference on Communications*, pp. 6606–6610, 2012.

[16] I. Kiss, B. Genge and P. Haller, A clustering-based approach to detect cyber attacks in process control systems, *Proceedings of the Thirteenth International Conference on Industrial Informatics*, pp. 142–148, 2015.

[17] B. Lantz, B. Heller and N. McKeown, A network in a laptop: Rapid prototyping for software-defined networks, *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks*, article 19, 2010.

[18] J. Lee, B. Bagheri and H. Kao, A cyber-physical systems architecture for Industry 4.0 based manufacturing systems, *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.

[19] F. Milano and M. Anghel, Impact of time delays on power system stability, *IEEE Transactions on Circuits and Systems – I: Regular Papers*, vol. 59(4), pp. 889–900, 2012.

[20] E. Molina, E. Jacob, J. Matias, N. Moreira and A. Astarloa, Using software defined networking to manage and control IEC 61850 based systems, *Computers and Electrical Engineering*, vol. 43, pp. 142–154, 2015.

[21] NEC Corporation, Software-Defined Networking (SDN) Solution, West Nippon Expressway Company Limited (NEXCO-West), Case Study, Tokyo, Japan (`www.nec.com/en/case/w-nexco/index.html`), 2016.

[22] Open Networking Foundation, Software-Defined Networking (SDN) Definition, Menlo Park, California (`www.opennetworking.org/sdn-resources/sdn-definition`), 2017.

[23] H. Sandor, B. Genge and G. Sebestyen-Pal, Resilience in the Internet of Things: The software defined networking approach, *Proceedings of the IEEE International Conference on Intelligent Computer Communication and Processing*, pp. 545–552, 2015.

[24] S. Sharma, D. Staessens, D. Colle, M. Pickavet and P. Demeester, OpenFlow: Meeting carrier-grade recovery requirements, *Computer Communications*, vol. 36(6), pp. 656–665, 2013.

[25] K. Stouffer, J. Falco and K. Scarfone, Guide to Industrial Control Systems (ICS) Security, NIST Special Publication 800-82, Revision 1, National Institute of Standards and Technology, Gaithersburg, Maryland, 2011.

[26] S. Vissicchio, L. Vanbever and O. Bonaventure, Opportunities and research challenges of hybrid software defined networks, *ACM SIGCOMM Computer Communication Review*, vol. 44(2), pp. 70–75, 2014.