Florida International University
FIU Digital Commons

7-7-2016

Software Defined Networking for Smart Grid Communications

Abdullah Aydeger
Florida International University, aayde001@fiu.edu

Follow this and additional works at: http://digitalcommons.fiu.edu/etd

Part of the Computer Engineering Commons

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

SOFTWARE DEFINED NETWORKING FOR SMART GRID

COMMUNICATIONS

A thesis submitted in partial fulfillment of the

requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

by

Abdullah Aydeger

2016

To: Interim Dean Ranu Jung
    College of Engineering and Computing

This thesis, written by Abdullah Aydeger, and entitled Software Defined Networking For Smart Grid Communications, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

_____
A. Selcuk Uluagac

_____
Ismail Guvenc

_____
Kemal Akkaya, Major Professor

Date of Defense: July 7, 2016

The thesis of Abdullah Aydeger is approved.

_____
Interim Dean Ranu Jung
College of Engineering and Computing

_____
Andres G. Gil
Vice President for Research and Economic Development and Dean of the
University Graduate School

Florida International University, 2016

ACKNOWLEDGMENTS

ABSTRACT OF THE THESIS

SOFTWARE DEFINED NETWORKING FOR SMART GRID

COMMUNICATIONS

by

Abdullah Aydeger

Florida International University, 2016

Miami, Florida

Professor Kemal Akkaya, Major Professor

Emerging Software Defined Networking (SDN) technology has provided excellent flexibility to large-scale networks in terms of control, management, security, and maintenance. On the other hand, recent years witnessed a tremendous growth of the critical infrastructure networks, namely the Smart-Grid, in terms of its underlying communication infrastructure. Such large local networks requires significant effort in terms of network management and security. We explore the potential utilization of the SDN technology over the Smart Grid communication architecture. Specifically, we introduce three novel SDN deployment scenarios in local networks of Smart Grid. Moreover, we also investigate the pertinent security aspects with each deployment scenario along with possible solutions. On the other hand, we conducted experiments by using actual Smart Grid communication data to assess the recovery performance of the proposed SDN-based system. The results show that SDN is a viable technology for the Smart Grid communications with almost negligible delays in switching to backup wireless links.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The proliferation of smart devices with their ability to access the Internet created new avenues to build large-scale cyber-phsycial systems. This, however, poses new challenges for not only keeping up with the dynamicity of the hardware and software, but also with the resilient and reliable data collection in such systems. The emerging software defined networking (SDN) paradigm can perfectly address these challenges by splitting controls of networks and data flow operations [HHB14]. One of the major goals in SDN is to be able to interact with the networking equipment (e.g., routers, switches) to create an open networking architecture for everyone. In this way, one can get a global view of the entire network and will be able to make global changes without having to access to each device's unique hardware. Eventually, various large-scale network architectures can be deployed and maintained with ease while still featuring resiliency and robustness.

In parallel with the advances in SDN technology, the existing power grid in the US is also going through a massive transformation to make it smarter (i.e., Smart Grid), which will be more reliable and connected with the ability to transfer data and power in two-ways [SAU12]. The need for data communication in the power grid necessitated upgrading the existing grid network infrastructure with different components such as home area networks (HANs), neighborhood area networks (NANs) and wide-area networks (WANs). In this way, a number of different applications such as Advanced Metering Infrastructure (AMI), Demand Response, Wide Area Situational Awareness, and green energy-based micro grids can be realized [SAU12]. Each of these Smart Grid applications deploys thousands of network devices that need to be managed continuously for reliable operations. Unfortunately, the management of this massive infrastructure requires additional labor and cost for the

utility companies who own these networks. Although minimizing the management cost is one of the goals of the utilities, this cost will always be relevant as long as the customers are served.

One sustainable solution to this network management problem is the use of the emerging SDN, which can provide excellent opportunities for reducing the network management cost by integrating a software-based control that can be flexible with respect to software upgrades, flow-control, security patching, and quality of service (QoS) [CHHK13]. Nonetheless, while a significant amount of work has been done in the SDN space, most of these efforts targeted the applications in the area of cloud computing, data centers, and virtualization [HHB14] and there is a need to adapt SDN for the existing needs of Smart Grid applications.

In this thesis, we both summarize and promote the use of SDN for various applications in the Smart Grid. Specifically, we target three different Smart Grid applications that heavily depend on the underlying infrastructure. 1)*AMI applications*, where meter data are collected via a mesh network that consists of smart meters and relays. Each of these equipment will have the ability to route the meter data through their routing tables. 2) *Supervisory Control and Data Acquisition (SCADA) Systems*, which connects field devices such as relays, IEDs, PLCs, and PMUs with the utility control center. The control center is typically equipped with routers and switches just like a data center. 3) *Microgrid Systems*, which integrates distributed power resources with the Smart Grid. The control and monitoring of these networks require the deployment of network devices for collecting data about them.

We explain how SDN can be utilized in these applications. Then, we articulate potential security threats that can arise as a result of deploying SDN in these applications and suggest solutions to alleviate the threats. Applying the maturing

SDN technology into the smart grid infrastructure presents ample unique research challenges in security and networking to engineers and scientists. This book chapter will be exploring those challenges within a SDN-enabled smart grid infrastructure.

In addition, the data communication motivation necessitates upgrading the existing network infrastructure with different components. With these new transformations, Smart Grid systems will need to maintain a large-scale heterogeneous network that brings a number of challenges. One of the challenge is the ability of this networking infrastructure to self-heal itself during man-made or natural (e.g., hurricane, earthquake) disasters so that potential blackouts and temporary outages can be minimized. In particular, since the underlying communication medium for most of the current substation-to-substation links is based on power-line communications (PLC) technology, they can be easily damaged during catastrophes which results in load imbalance and eventually blackouts. Through continuous interactions between different components of the smart grid, the new energy infrastructure should reconfigure the control of the physical assets and network topology in an efficient manner and achieve resilient operations.

Each of these networks deploy thousands of network devices that need to be managed continuously. Unfortunately, this massive infrastructure requires additional labor and cost for the utility companies who own these networks. Although minimizing the management cost is one of the goals of the utilities, this cost will always be relevant as long as the customers are served. In fact, the emerging SDN paradigm can provide excellent opportunity for reducing the network management cost by integrating a software-based control that can be flexible with respect to software upgrades, flow-control, security patching, and quality of service (QoS).

Self-healing in such unfortunate situations requires a comprehensive coordination among the components of the Smart Grid and needs to rely on the availability of

backup or redundant links/paths and devices. In this regard, we propose using SDN to enable interaction with the valuable networking equipment and assets by leveraging SDN's unique ability to maintain a global view of the Smart Grid. Note that while a significant amount of work has been done in the SDN space, most of these efforts targeted the applications in the area of cloud computing, data centers and virtualization [HHB14]. Thus, there is a need to adapt SDN for various Smart Grid resiliency applications and to the best of our knowledge SDN is considered for the first time in the Smart Grid settings.

CHAPTER 2
PRELIMINARIES

2.1 SDN

Switches in traditional networks enable the data flow without causing packets to wait for another even though the packets might arrive at similar times. Local Area Network (LAN) switches are based on the idea of packet switching. In this method, incoming packets are saved to temporary memory, MAC address contained in frame's header is read and compared to list of addresses in the switch lookup table. There are 3 different configuration ways:

- *Shared memory*: It stores all packets arriving in a common memory buffer shared by all the switch ports, then sends them out via the correct port for the destination node.

- *Matrix*: This has an internal grid with the input ports and the output ports crossing each other. When a packet is detected on an input port, the MAC address is compared to the lookup table to find the appropriate output port. The switch makes a connection on the grid where these two ports intersect.

- *Bus architecture*: Instead of a grid, an internal transmission path (common bus) is shared by all of the ports using Time division multiple access(TDMA).

When a new networking device joins the network, it will broadcast its information to let others know about its existence. Switches will pass that information to all other segments in their broadcast domain. For routers, this process is similar except that it occurs at the IP layer and thus uses IP addresses instead of MAC addresses used by switches. The routing tables are used for checking the incoming IP addresses.

In both cases, there is a lookup table for the incoming packets inside the devices. These tables can only be changed by the devices based on the changes in the network. Each device's lookup table is solely controlled by that device not through any other centralized mechanism. SDN's main motivation is to move the control of the lookup tables that are inside the network devices to a separate location so that they can be controlled more easily.

Specifically, this can be described as separation of the packet forwarding and the way how the forwarding tables are created and changed. These two processes are assumed to be on separate layers, which are referred to as *data plane* and *control plane* in SDN technology: *Control Plane* is also referred to as Network Operating System (NOS). This plane is supposed to supply network decisions for packet forwarding or dropping. On the other hand, data plane is responsible for data forwarding. SDN focuses on flow forwarding idea instead of dealing with IP or MAC-address based forwarding.

Main problem in the traditional networks was updating the network elements, which requires too much effort and time [HHB14]. By creating a programming interface to be able to update network elements from a center (i.e., at the control plane) via SDN, such complexity in network management can be eliminated. An illustration of how SDN reorganizes the network architecture with respect to the current ones are is illustrated in Fig. 2.1. As can be seen from this figure, the control plane is typically controlled by a single center through some applications.

However, this is not the only way to implement the control. The control can also be distributed to many other centers which will bring some overheads in terms of performance and reliability. Further information about the comparison of these approaches can be found in [ZWHJ14]. SDN enables innovation on network and each TCP/IP layer might have an independent innovation. The SDN-enabled net-
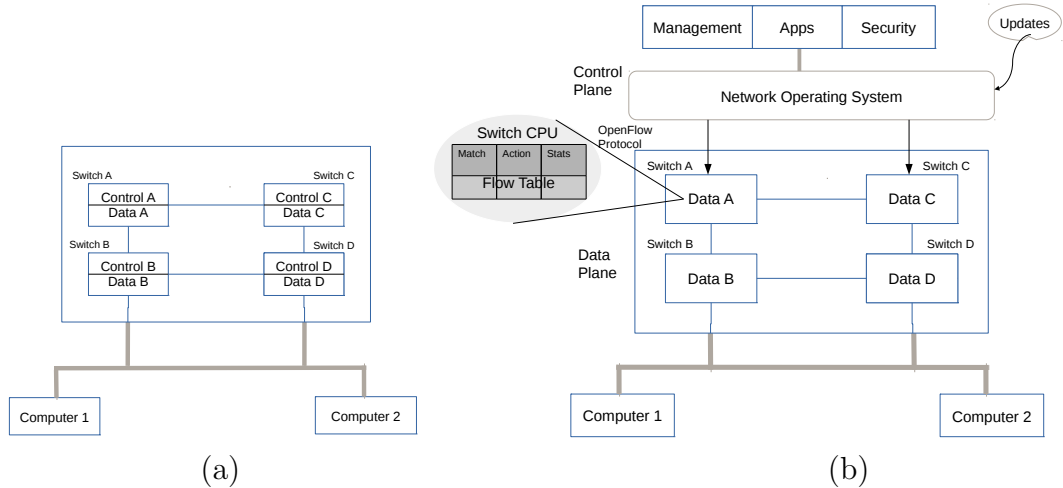
Figure 2.1: Current (a) vs SDN-based (b) Networks

works become more flexible and accessible networks with software interfaces making it very convenient for network management. SDN can provide more fine-grained control on traffic compared to traditional networks. It gives the network administrators the ability to arbitrarily change routing tables on routers without having to deal with each of the routers individually. It also provides the ability to setup new services through virtualization. Finally, it provides better and more granular security through the availability of the control plane. For instance, each user can have different abilities provided by the network to control their firewalls.

Today, SDN is used for a variety of networking settings. For instance, it is used in cloud computing and data centers for ease of network management and control of virtual machines (VMs) [MVKK12]. It can connect multiple data center networks by eliminating the problems of proprietary architectures. In this way, the workload can move from one network to another to save time and energy. This idea of workload offloading can also be used for mobile devices which needs more powerful machines with certain security requirements [BAH+12]. Finally, SDN can be used for Internet research for testing certain ideas without changing the current network [dOSdSPRK12].

2.1.1 SDN Controller

SDN Controller is the main part of SDN-based systems. It has the functionality part and can be considered as brain of a system. SDN Controller receives the data from switches in the network, and decides and responds back by considering the packet information. SDN Controller may drop the packet, create a rule and assign it to the switch etc.

SDN Controller should have two interfaces; one of them to communicate with user applications, and the other one to communicate with SDN switches. First of them called Northbound API. This API provides a developer to update the network elements from distant point. It could be done with a few ways including the REST calls, which is basically sending queries to a specific web URL (web link), and get responds from them. That respond can be used for such purposes. Second interface is the one sending rules to SDN switches from SDN Controller. Most common protocol for Southbound API is called OpenFlow.

There are lots of SDN Controller in the market. We have decided to use Open-DayLight SDN Controller since it is open-source and easy to install [ope]. It also provides REST interface to communicate with SDN Controller from our own applications.

2.1.2 OpenFlow

SDN requires some protocols for the control plane to communicate with the data plane. One of such protocols is OpenFlow [ope]. OpenFlow is the first standard communications interface defined between the control and forwarding layers of an SDN architecture. It allows direct access to and manipulation of the forwarding plane of network devices such as switches and routers, both physical and virtual. In
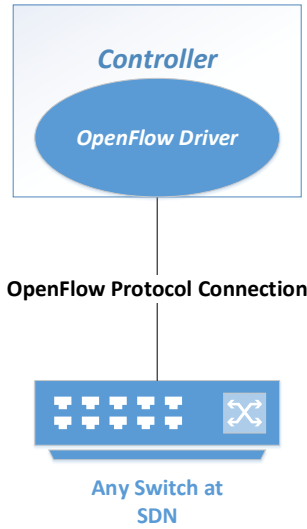
Figure 2.2: OpenFlow Protocol

other words, it is a communication protocol proposed for supplying communication between routers and network switches (data plane) to routing decision center (control plane). It simply gives programmer control over routing protocol of routers. Figure 2.2 shows OpenFlow protocol stands.

2.2 Mininet

Mininet is a network emulator. It is open-source and implemented with Python programming language, and can be expanded. It is the most popular emulator for OpenFlow based network environments. It supports OpenFlow protocol for every nodes, either switch or host, by default. It implements real behavior of network devices and gives an easy way to get and use correct system behavior. It is so simple to experiment with different topologies as well.

Mininet networks run real standard Unix/Linux code network stack at its own nodes. Because of this, the code can be moved to a real system with a few changes,

for real-world testing, performance evaluation, and deployment [min]. In addition, Mininet can work with any SDN Controller.

2.3   Smart Grid

Smart Grid is uses analog or digital information and communication technology to collect data from the Power Grid and behave according to that data in an automated way to increase the efficiency, reliability, economics, and sustainability of the production and distribution of electricity. Smart Grid has a large-scale communication network that consists of Home Area Networks (HANs), Neighborhood Area Networks (NANs) and Wide Are Networks (WANs). Each of these network components deploy thousands of devices such as Intelligent Electronic Devices, Phasor Measurement Units (PMUs), routers, switches and computers. In addition, these network employ different underlying technologies whether it be wired or wireless along with various communication protocols. This brings up the problems of interoperability as well as network management.

CHAPTER 3

RELATED WORK

The use of SDN in Smart Grid applications has just started to be considered recently [ZSLF13a]. In general, SDN has been employed for various purposes: resilience, network management, security and quality of service (QoS). For instance, one of the recent works proposed using SDN to support multi-rate multicasting for PMU data collection assuming the availability of wireless PMUs [Mil10] within an SDN architecture [GKRC13]. Since SDN controller has a global view of the network, it can provide efficient routes that can match the performance of the existing approaches with commodity switches. In addition to this work, [SNS+13] studied SDN usage for QoS in wide are networks for Smart Grid and compared the performance with MPLS technology. Different from this work, we study the performance of an SDN-enabled system in a IEC 61850 application as opposed to PMUs. In addition, our main goal is not to provide QoS, rather to test resilience capability during link failures.

Another recent work is on substation automation and virtualization using SDN [CHHK13]. The authors provided advantages of deploying SDN in a substation network in term of management and then tested its feasibility in Mininet with 3 switches. Compared to traditional Ethernet switches, they observed that the latency of data coming from IEDs is much less. However, in that work, there is no quantitative experimentation and the work is mostly dedicated to potential ideas with SDN. Our work, on the other hand, provides extensive experimentation in case of link failures and involves ns-3 for fine-grained result collection using real IEC 61850 MMS data. Moreover, [GdSDKAW+15] tested multi-path routing for eavesdropping services in a SCADA network of substations using Modbus data for routing. While we also follow the idea of redundancy, we achieve it at the link

level with wireless complementary links. Finally, in [DKG$^+$14], the work considers both resilience and QoS for Smart Grid applications when SDN is employed. Specifically, they considered the performance at the transmission and distribution networks focusing on MMS and GOOSE protocols. The tests were carried out on an actual testbed of OpenFlow switches. They studies the recovery delay when one of the links were removed manually. We follow a similar setup for link failure, but as opposed to this work, our goal is to assess the effectiveness of redundant wireless links.

CHAPTER 4

SECURE SDN ARCHITECTURES FOR THE SMART GRID

4.1 Integration of SDN with The Smart Grid

In this section, we first explain the communication infrastructure for Smart Grid and then detail the applications that can benefit from the use of SDN.

4.1.1 Smart Grid Networking Architectures

Before moving into the description of how SDN can be used for Smart Grid applications, we first briefly explain the existing Smart Grid network infrastructure. Basically, there are three major components in a Smart Grid infrastructure [SAU12]: 1) HANs that mainly connect home devices with the smart meters; 2) NANs which collect smart meter data from houses; and 3) WANs which provide long-haul communication with the utility control centers using various technologies including cellular ones. A typical Smart Grid networking infrastructure showing possible infrastructure for generation, transmission, and distribution components of power grid is depicted in Fig. 4.1 redrawn from [SAU12]. Under this infrastructure, many applications could run simultaneously. In this work, we focus on three different applications that utilize this networking infrastructure at different levels as will be discussed shortly.

4.1.2 Motivation for SDN-Enabled Smart Grid

Utilities and energy companies own and operate network components that are part of the Smart Grid communication infrastructure. In some cases, they also lease services from telecommunication companies or third party cloud services. In any
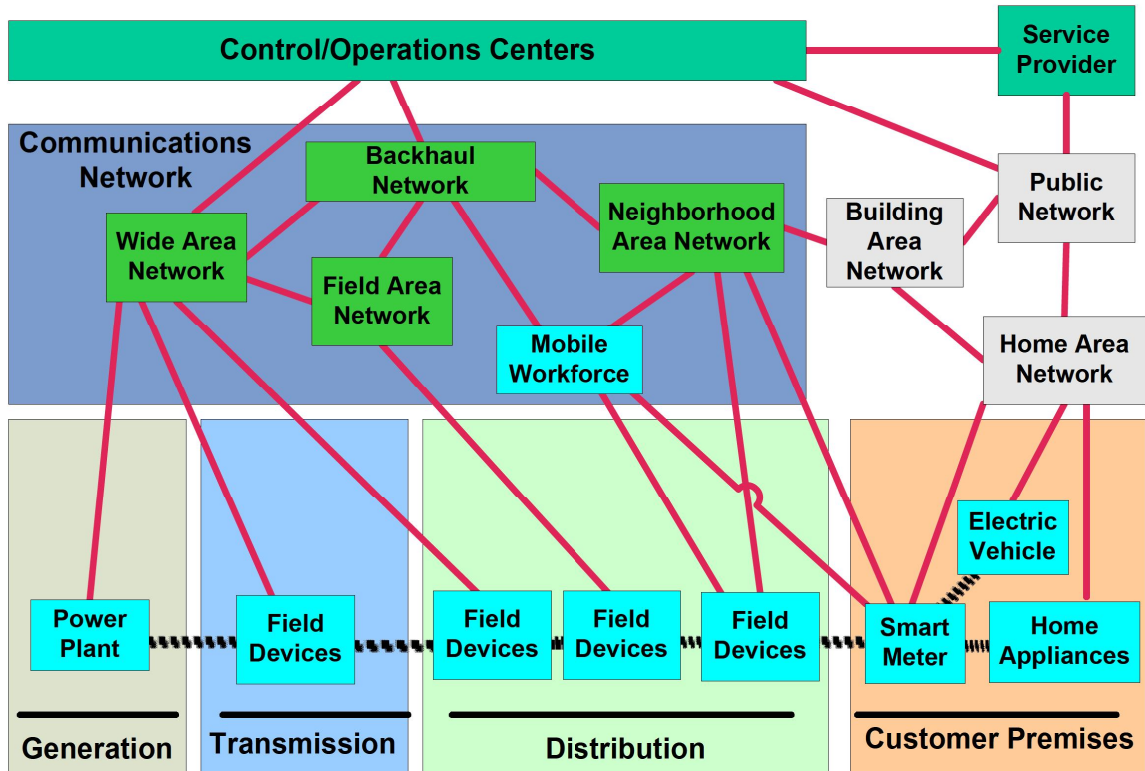
Figure 4.1: Smart Grid multi-tier communication network.

case, the management of the networks is a great challenge due to the scale. Furthermore, due to using different vendors and applications, the equipments may not be interoperable. Therefore, the utilities will need to deal with equipment maintenance and software upgrades that brings a lot of burden in terms of cost and labor. The following reasons necessitate a more flexible network management technology based on SDN:

- Software upgrades are also challenging in a large-scale network.

- Regulatory compliance in the security area also requires a form of software update in a fast fashion.

- Mergers and acquisitions of businesses which causes the task of merging multiple networks or seamlessly integrating one into the other.

- New equipment and improved technology necessitates replacing the software without any disruption to the existing older equipment.

- Peering arrangements with the regulatory and government agencies

SDN seems to be a promising solution for the aforementioned problems due to following advantages it brings:

- SDN provides a global end-to-end view of the network which make the large-scale management more effective.

- SDN adopts open standards and introduces technology abstraction, which provides a vendor-agnostic approach to configuring and maintaining various types of network elements that are common in Smart Grid.

- Hardware virtualization through SDN eases the burden of managing different networks while using resources efficiently.

- Due to its holistic view of network, the SDN-based network will provide superior control of delay and jitter in the network which is crucial for SCADA systems in terms of state estimation and control.

- SDN's bandwidth-on-demand capabilities will also create opportunities to increase revenue through accelerated service velocity in cases where the utility also serves as a communication service provider in the coverage area. More and more utility companies are functioning as service providers in rural areas.

4.1.3 SDN-enabled NANs

Smart Grid's NAN is mainly used for AMI applications. While there has been some wired options for building these communications, recent implementations solely targeted wireless solutions that depended on different standards such as IEEE
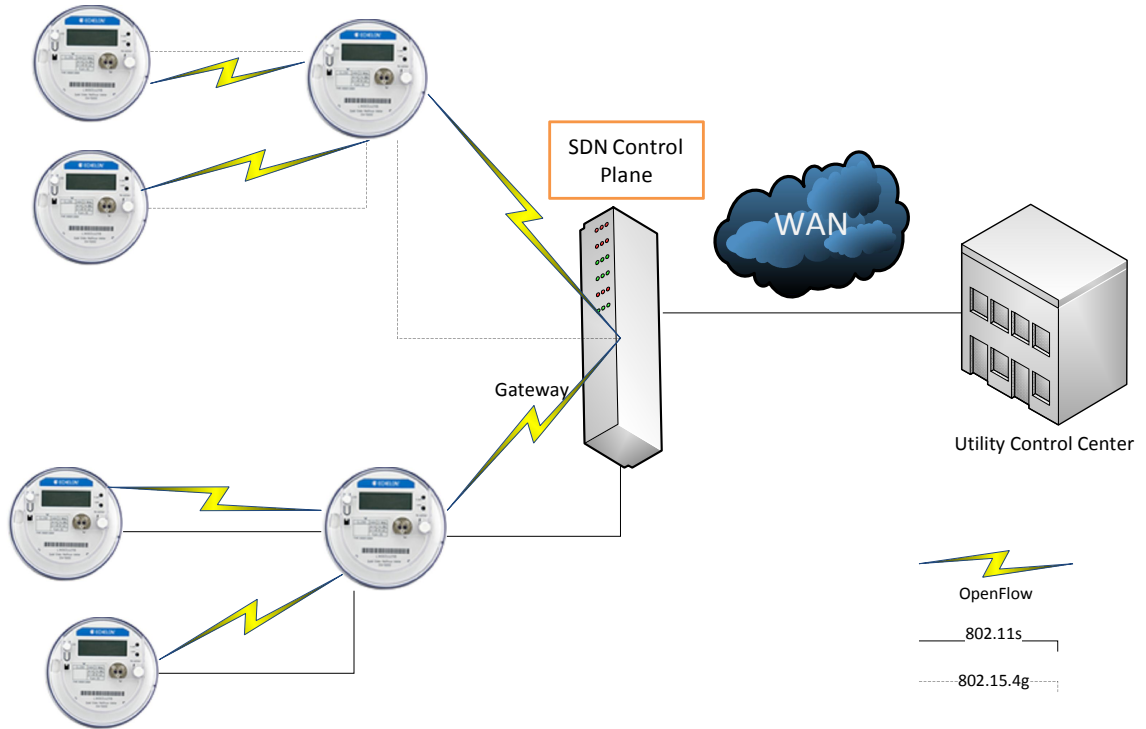
Figure 4.2: SDN for Smart Grid NANs [SAU12]

802.15.4g, IEEE 802.11s, RF-Mesh, and other proprietary mesh networks [SAU12]. As long as different vendors' products support OpenFlow, a NAN using a mixture of these standards can be easily controlled and re-tasked through a standard network control script programming.

In most cases, Smart Grid operators prefer exchanging the information among different NANs in order to get a better load state estimation. Therefore, being able to control such a network in a centralized manner for load balancing, security and QoS services is very valuable. While SDN can provide this novelty, there are still challenges that would require some research to enable the use of SDN in wireless environments.

One of these challenges is the performance of the centralized control. As opposed to wired networking interface among the controller and SDN switches, this will not be the case in wireless-mesh based NANs due to the scalability of the AMI. There-

fore, the control will be through wireless communication and most probably using multi-hopping (see Fig. 4.2). If the same channel is used for data communications, then this may create a lot of interference. While some of the very recent works investigated this issue for wireless mesh networks using [NMG+14], these works do not directly apply to AMI NANs where the scale is larger and the variety of nodes is significant in terms of used hardware/software. This suggests investigating the feasibility of distributed control in SDN-based NANs [DKB11].

4.1.4 SDN-enabled SCADA

SCADA systems were designed to collect data from field devices such as PLCs, PMUs, IEDs at substations in real-time and do control decision at the control center in terms of reliability and quality of the power [BDD+10]. A substation contains hundreds of different IEDs, each generating and/or consuming information about the status of some aspect of the substation. A standard called IEC 61850 is also used for substation automation [BDD+10]. Currently, these systems are not only scalable, but also their sensing coverage is very limited. Proper configuration and maintenance of IED communication requires significant effort. The network complexity further increases with the uses of other protocols.

However, with the modernization of the power grid, there will be opportunities to upgrade these systems based on a mix of wireless and wired infrastructure through the deployment of a large number of modern PMUs. For instance, the design and use of wireless PMUs have already begun [Mil10]. One of the recent works proposed using these wireless PMUs within an SDN architecture so that the network administrator would have a global view of the power grid computer network, which makes it is easier to manage PMU telemetry traffic compared to a traditional IP network
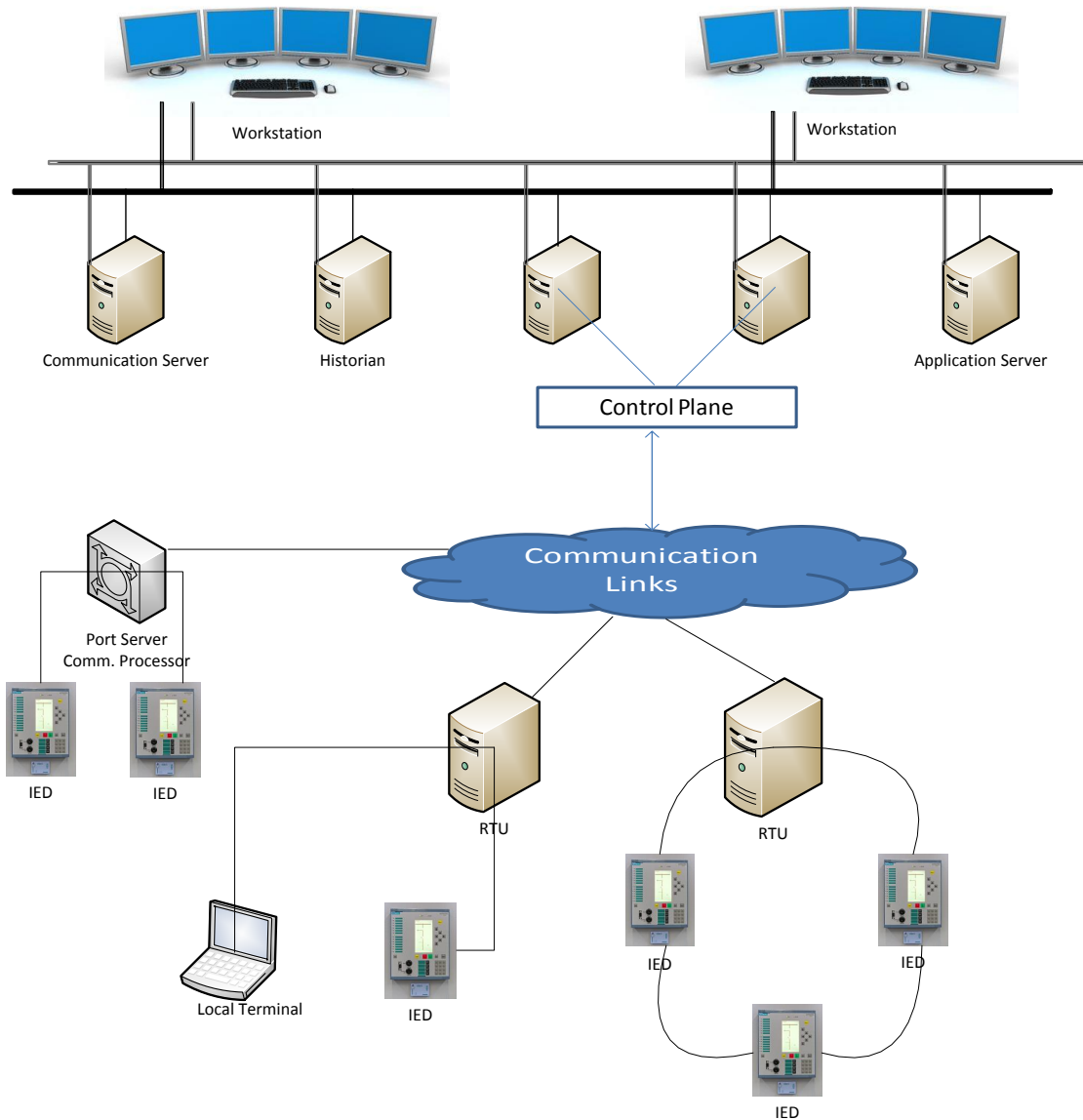
Figure 4.3: SDN for SCADA networks

[GKRC13].

There are also opportunities to reorganize the elements of these SCADA systems especially in terms of exploiting efficient ways to eliminate the complexities of multicasting and broadcasting. Massive amount of data is transmitted through these broadcasts or multicasts. Current architecture is a centralized one with hub-and-spoke model which is inadequate to address too many broadcasts or multicasts.

Researchers strived to address this issue by using middleware approaches in the past [GBHB09]. The idea in these approaches are to implement publish and subscribe mechanisms where the data sources publish data and the brokers in the middleware (at the application layer) are responsible for delivering this data to subscribers within their QoS requirements. This is in a way a sort of group communications among publishers and subscribers. However, since this was implemented at the application layer, it is not only slower, but also not flexible in terms of hardware requirements. SDN can be used to redesign this middleware by including the control plane in the middleware, but at the network layer [ZSLF13b].

Another advantage of such an SDN-based control is the ability to perform traffic engineering, which cannot be done with layer-2 switches using spanning tree-based routing [ALW+14]. SDN can replace MPLS-based systems for meeting QoS requirements of data coming from PMUs and IEDs as shown in Fig. 4.3. For instance, the work in [SNS+13] has explored a simulation-based study of using SDN in inter-substation networks from the perspective of substation failures.

4.1.5   SDN-enabled Microgrid

A microgrid is a miniaturized version of power grid which can supply electrical load of small communities such as university campuses, malls, camps etc. It includes numerous Distributed Energy Resources (DERs) (e.g., photovoltaic (PV) systems, micro-combined heat and power systems (μCHP), and electric vehicles (EVs)), load, storage, and protection devices that are controlled by a central controller. Microgrids are becoming viable options for saving energy and generating clean power along with their reliable electrical services. For instance, University of California San Diego had setup a micro grid for its own campus which saved them $850,000 a month [UCS].

The major issue with these microgrids is the risk of rapid changes that may cause instability and eventually collapse in the system. Therefore, it is crucial to perform fine-grained real-time monitoring and control. This can only be achieved via reliable communications that can provide QoS in support of low data latency, packet prioritization and traffic engineering. These features can be supported via the SDN technology to stabilize and optimize system operation. Another opportunity for the use of SDN in microgrids is on the problem of DER management and aggregation. Basically, the DERs are grouped together for different purposes and any mobility related group changes will affect the system. Currently, this management is done at the application layer. The complexity can be reduced by exploiting SDN capabilities that will be implemented at the network layer [ZSLF13b].

But network devices are ideally suited for being powered by micro grids that use renewable energy. Most network devices have various power modes, battery backup and can re-route traffic based on availability of power. They have a lot more intelligence than most devices connected to the power grid to enable smart power management.

4.2   Introduction to SDN Security

SDNs have potential to offer greater security by providing consistent access control, ability to apply security policies efficiently and effectively, and the ability to centrally manage and control network topology. However, SDN networks introduce some new security vulnerabilities on their own. These vulnerabilities are related to single point of failure in the SDN controller, potential to cause congestion in communications between data and control places. In the following, we summarize the existing works that studied SDN security.

Klinger et al., performed security analysis of three major SDN architectures, namely the Path Computation Element (PCE), 4D, and the Secure Architecture for the Network Enterprise (SANE) [KKMB14]. They use Microsoft threat modeling technique based on the STRIDE model which categorizes threats into six categories, namely Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege. STRIDE analysis of PCE revealed several vulnerabilities like tampering, information disclosure, and denial of service. On the other hand, the 4D architecture does not provide any mechanism to prevent against tampering, information disclosure, or denial or service threats. In the security analysis it was found that processes in SANE architecture are the weakest link in the security chain. They are susceptible to spoofing, tampering, repudiation, information disclosure, DoS, and elevation of privilege.

Researchers have also investigated potential for attacking the OpenFlow framework using DoS attacks [SG13]. For instance, one of the attacks known as the *control plane saturation attack* attempts to saturate the communications between the control and data planes in the OpenFlow network. If an adversary sends a large number of packets with no matching rules, it can saturate the communication between the switch and the controller resulting in a DoS attack. Shin et al. proposed a solution to this attack by introducing some intelligence in the data plane to distinguish between sources that will complete a TCP session with those that will not [SG13]. The second challenge that is studied by them is the issue of monitoring and controlling flow dynamics. Inability to monitor and control flow dynamics is a big weakness that results in the inability to detect and respond to DoS attacks. The authors propose a method called actuating triggers where control layer applications send alert signals when there are changes in the flow dynamics.

Shin et al., also proposed a variant of the saturating attack to launch a DoS attack on SDN networks by fingerprinting the network [SYPG13]. In this attack, an adversary sends packets for new and existing flows and measures their response times. The response times will tell the attacker whether the target network is using SDN or not. The attacker can then launch the resource consumption attacks.

4.3 Security of SDN-Enabled Smart Grid

In this section, we discuss the security of the SDN-enabled smart grid. First, we articulate the threat model, then we list the desired security services for the SDN-enabled smart grid.

4.3.1 Threat Model

Conceptually, the threats to the SDN-enabled smart grid could be listed from four different complementary perspectives: (1) *Method-specific*; (2) *target-specific*; (3) *software-specific*; and (4) *identity-specific*.

Method-specific threats define how the threats are executed. The method-specific threats can be either passive or active. In the passive method, the attacker only monitors (or eavesdrops), records the communication data occurring in the SDN-enabled smart grid, and analyzes the collected data to gain meaningful information. In the active one, the attacker tries to send fake authentication messages, malformed packets, or replay a past communication to the components of the the SDN-enabled smart grid. As passive threats are surreptitious, it is harder to catch their existence. However, it is easier to catch the existence of an active attacker, but its damage to the smart grid can be relatively higher than the passive threats.
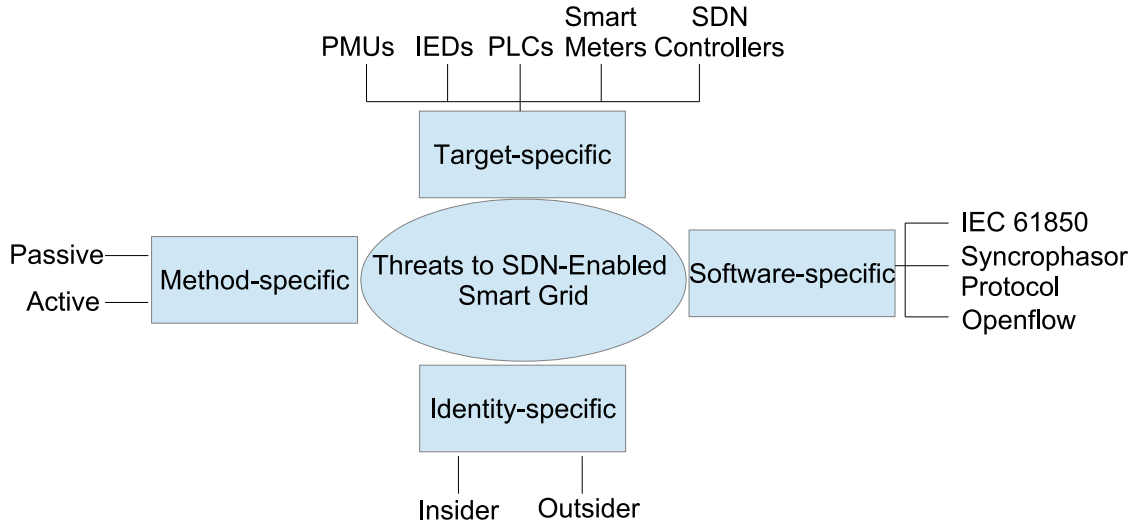
Figure 4.4: A threat model for the SDN-enabled smart grid

Target-specific threats classify the attacks according to which device the threats target. In an SDN-enabled smart grid, any device such as IEDs, PMUs, PLCs, Smart Meters could be valuable targets for potential malicious activities. In software-specific threats, the attackers aim to exploit the vulnerabilities associated with the networking protocols, software suits (IEC 61850, IEEE C37.118 Syncrophasor Protocol, Openflow of SDN) that run in the smart grid. Finally, depending on the identity of the attacker, i.e., whether an attacker is a legitimate member of the network during an attack or not, she can be defined as insider or outsider attacker. Insiders are more dangerous than the outsiders as they have more knowledge about the internal architecture of the SDN-enabled smart grid.

In reality, there is no hard line between these attacking models and they complement each other because an insider could be a passive attacker trying to exploit IEC 61850 on an IED in the SDN-enabled smart grid. The threat model for the SDN-enabled smart grid is presented in Fig. 4.4.

4.3.2 Desired Security Mechanisms

Desired security mechanisms are usually defined by the national and international standardization bodies (e.g., National Institute of Standards and Technology, International Telecommunication Union (ITU)) and are used by many researchers and practitioners who aim to develop secure systems. In this sub-section, we use the security architecture suggested by the ITU's Recommendation X.800 [X.891] documentation, which is referred to as the Security Architecture for Open Systems Interconnect (OSI) as our guideline in addressing the threats discussed in the previous sub-section.

Confidentiality: Confidentiality refers to the protection of the exchanged content (e.g., gathered data, reports, commands) among the components of the smart grid such as IEDs, PMUs, PLCs, Smart Meters. A malicious entity which has the privilege to access the content, should not be able to decode the exchanged messages in the network. Confidentially also entails the protection against any unintended *information leakage* from the applications, controllers, and devices within the SDN-enabled smart grid. This is particularly important because the data generated and collected by the smart grid equipment, e.g., PMUs, IEDs are very periodic in its nature. Data forwarding policies or flow rules associated with the collected data may be discovered with simple timing or side-channel analysis. Similarly, an increased delay for the establishment of a new flow rule in response to an incoming packet can inform a potential attacker about the behavior of the OpenFlow controller within the SDN-enabled smart grid. This unintended information disclosure from data plane devices, applications, flows, controllers should also be considered as part of any confidentiality service.

Traditionally, confidentiality can be provided by adopting either symmetric and asymmetric key-based encryption schemes [SB15]. In symmetric encryption, one

key is utilized among the PMUs, PLCs, smart meters, IEDs, applications, flows, network controllers. Examples of symmetric encryption that can be utilized for the smart grid include AES, RC4. On the other hand, in asymmetric encryption, a pair of two keys (aka public and private) are utilized among the communicating components of the smart grid. RSA and ECC are the two most important examples of asymmetric encryption that could be deployed. Moreover, the maturing state-of-the art encryption mechanisms based on fully-homomorphic-encryption could be utilized for specifically preserving the privacy of the flows.

Authentication: Authentication involves guaranteeing the genuineness of the communication among the devices in the data plane, controllers, and the applications. An authentication mechanism verifies if the exchanged information stems from the legitimate participants of the SDN-enabled smart grid because a malicious entity (e.g., a compromised IED) may be able to inject counterfeit content or resend the same content into the SDN-enabled smart grid. More specifically, an adversarial smart grid application may attempt to insert new flow rules that may circumvent flow rules imposed by other applications [SHOS13]. Adversaries may also insert new rules to damage the system by influencing the state estimation, which is crucial to evaluate the demand.

Authentication can fundamentally be provided based on three factors [SB15]: (1) *Knowledge factor*: the proof of the knowledge of some secret (e.g., passwords) is provided to the authenticator. Symmetric, asymmetric key-based encryption schemes and hashing algorithms can all be utilized as part of the authentication mechanism with the knowledge factor. (2) *Possession factor*: authenticator verifies the claimant using the credentials provided by a specialized hardware. Electronic cards, smart cards, smart tokens physically owned by the claimant can be utilized and integrated with the SDN-enabled smart grid devices and applications. (3) *Identity factor*: the

authenticator utilizes features uniquely identifying in the verification of the claimant. Both static or dynamic patterns that can identify the devices and applications can be utilized. For instance, behavioral information from the SDN-enabled smart grid devices and applications such as communication patterns, timing patterns, delays can all be utilized [LUB14] as part of this authentication method. Within the SDN-enabled smart grid, all of these authentication techniques can be individually or a combination of one or more of the techniques could be adopted. If more than one factor is utilized, the authentication is called multi-factor authentication.

Integrity: Integrity refers to the capability to detect if the exchanged content between the communicating devices of the smart grid have been altered or not. Furthermore, the integrity service involves ensuring that the exchanged content is not deleted, replication of old data, counterfeit, or stale because the nature of the messages in the smart grid is very time-sensitive. Within the SDN-enabled smart grid, modification of the flow rules or insertion of new Openflow rules [KKS13] by adversaries can cause severe damage to the healthy operations of the smart grid.

Integrity is usually provided by appending the cryptographic digest of the message content to the message itself [SB15]. When the PMUs, PLCs, smart meters, IEDs, applications, network controllers receive the message, they can check to see if the digest of the content matches the digest they computes on their end. If the digests match each other, then the message is deemed legitimate and not to have changed from its original content. Content digests in integrity are usually created with the usage of hashing algorithms. There are several hashing algorithms such (e.g., MD5, SHA-2) in use today, which do not require the presence of keys unless they are specifically designed to work with keys like keyed-hashing (e.g., HMAC, CMAC). Alternatively, integrity can be provided as part of a digital authentication mechanism utilizing symmetric and asymmetric encryption techniques. For

instance, the last block of the encrypted data in AES can be appended to the message that would be sent as the integrity code. In a similar fashion, a private key in the asymmetric encryption techniques (e.g., RSA, ECC) can be used to provide the integrity code appended to the message.

Access Control: With access control, unauthorized use of a resource in the SDN-enabled smart grid is prevented. Access control addresses which participant of the smart grid reaches which content or service. For instance, IEDs should not be allowed to have the privileges of PMUs. Proper security measures must prevent any unauthorized SDN controller access. An unauthenticated application might try to access to resources for which it does not have exclusive privileges. Or, an authenticated application, IEDs, PMUs, PLCs, and Smart Meters may abuse its privileges.

Access control is usually achieved through four different methods [SB15]: (1) *discretionary access control (DAC)*; (2) *mandatory access control (MAC)*; (3) *role-based access control (RBAC)*; and (4) *attribute-based access control (ABAC)*. In DAC, access control decisions are made based on the exclusive rights that are set for the flows, applications, IEDs, PMUs, PLCs, and Smart Meters. An entity in DAC can enable another entity for accessing resources. In MAC, access control function considers the criticality of the resources and the rights of the flows, applications, IEDs, PMUs, PLCs, and Smart Meters on the resources. In MAC, an entity can not enable another entity for accessing the resources. In RBAC, access control decisions are based on the roles created within the the SDN-enabled smart grid. A role can include more than one entity e.g., the flows, IEDs. Moreover, a role defines the capabilities what the entities can do or not do within a certain role. Finally, in ABAC, the access control decisions are based on the features of the flows,

applications, IEDs, PMUs, PLCs, and Smart Meters, resources to be accessed, and environmental conditions.

Availability: Due to the threats to SDN-enabled smart grid, some portion of the grid or some of the functionalities or services provided by the grid could be damaged and unavailable to the participants of the grid. For instance, some PLCs could be compromised and they could cease functioning. A Denial-of-Service (DoS) type attack [SAB14] can overflow the communication link of the SDN controller-switch [SHOS13]. SDN flow switch tables can be flooded by fake entries. In a similar fashion, a centralized SDN controller can be a single point of failure. Moreover, recent technological advances enabled the integration of the wireless technologies (e.g.,) into the smart grid infrastructure. In such cases, adversaries may jam the wireless medium, effectively hampering all the communications. Thus, availability service ensures that the necessary functionalities or the services provided by the SDN-enabled smart grid are always carried out, even in the case of attacks.

Usually, the smart grid includes redundant components in their infrastructure. This is to ensure the continuous operation during failures. In a similar fashion, the SDN-enabled smart grid can be designed with such redundancy to achieve the availability service.

Accountability: With accountability (aka non-repudiation [Sta03]), the SDN-enabled smart grid ensures that a device or a software component (e.g., applications, IEDs, PMUs, PLCs, and Smart Meters) can not refute the reception of a message from the other device or application or the sent of a message to the other device or application in the communication.

Accountability can be provided as a service bundled inside authentication and integrity. For instance, a digital signature scheme (DSS) [Sta03], which is based on utilizing encryption methods would address accountability. Additionally, proper

auditing mechanisms and logs should be utilized to provide accountability in the SDN-enabled smart grid.

4.3.3 Specific Security Problems for SDN-Enabled Smart Grid

With respect to the three examples we have provided where SDN can be applied, specific security concerns may arise. In the following, we discsuss these issues and possible problems to address these.

- *Security of OpenFlow Protocol*: In all cases, the security of the communication among the switches and control plane will be through OpenFlow protocol. This can be either via a wireless or wired connection. Obviously, the main security issues will be related to authentication and integrity of these communications. For instance, an adversarial Smart Grid application may attempt to insert new flow rules that may circumvent flow rules imposed by other applications. They may also insert new rules to damage the system by influencing the state estimation.

- *Information Disclosure*: This arises from timing analysis and can reveal certain aspects of a network's state as well as a controller's strategy to an attacker. Mitigation in this context means ensuring that the observable system parameters do not expose the internal system state. For example, the increased delay for the establishment of a new flow rule in response to an incoming packet can inform the attacker about the behavior of the OpenFlow controller.

- Single central controller can be single point of failure. Also a problem of fault-tolerance and scalability.

- Upper level protocols communication with the controller. Possible compromise.

- DoS on the controller-switch communication or on the network.

4.4   Security of SDN over Smart-Grid

Security: Link isolation is a critical requirement of the substation network not only for superfluous traffic congestion on IED network interfaces, but also for security and access issues within the operating environment. IED configuration is commonly carried out live when other devices on the substation network are performing monitoring and control of the substation. The risk of a malicious attack, masked as a live-reconfiguration event, is an attack vector that could be mitigated with a higher degree of network-level security. The SDN-enabled substation network controller has the capacity to support more security at the controller level. The nature of the software-defined controller also allows for greater flexibility in security policies and access control between connected IEDs. A group of devices that are linked through a message group can be enabled for one-way communication and only allow the authorized publisher to send traffic into the network. This addresses a common hole in substation network security.

CHAPTER 5

SDN FOR RESILIENT COMMUNICATIONS IN SMART GRID ACTIVE DISTRIBUTION NETWORKS

5.1 SDN-Based Inter-Substation Network

5.1.1 Proposed Model

We propose an SDN-based communication infrastructure that can be deployed both within and among the substations as shown in Fig. 5.1. Specifically, each substation can maintain an SDN gateway switch, which can be controlled by a global SDN controller that is located at the Utility Controls Center. The global SDN controller can maintain the traffic among the substations (e.g., IEC 61850 MMS traffic) by inserting the flow rules in the table. The regular communication with the gateway can be achieved through a control line or the existing data network, typically wired. The gateways at the substations can also be part of the local area network (LAN) within the substation and thus we also introduce a local SDN controller for controlling the traffic within this substation LAN.

In this way, the flow table in the gateway can be accessible by multiple controllers, but the scope of these controllers will be different. While the global SDN controller will adjust the inter-substation communications, the local SDN controller focuses on the interior traffic in a substation. The connection between two substations are based on redundant links as will be detailed next.
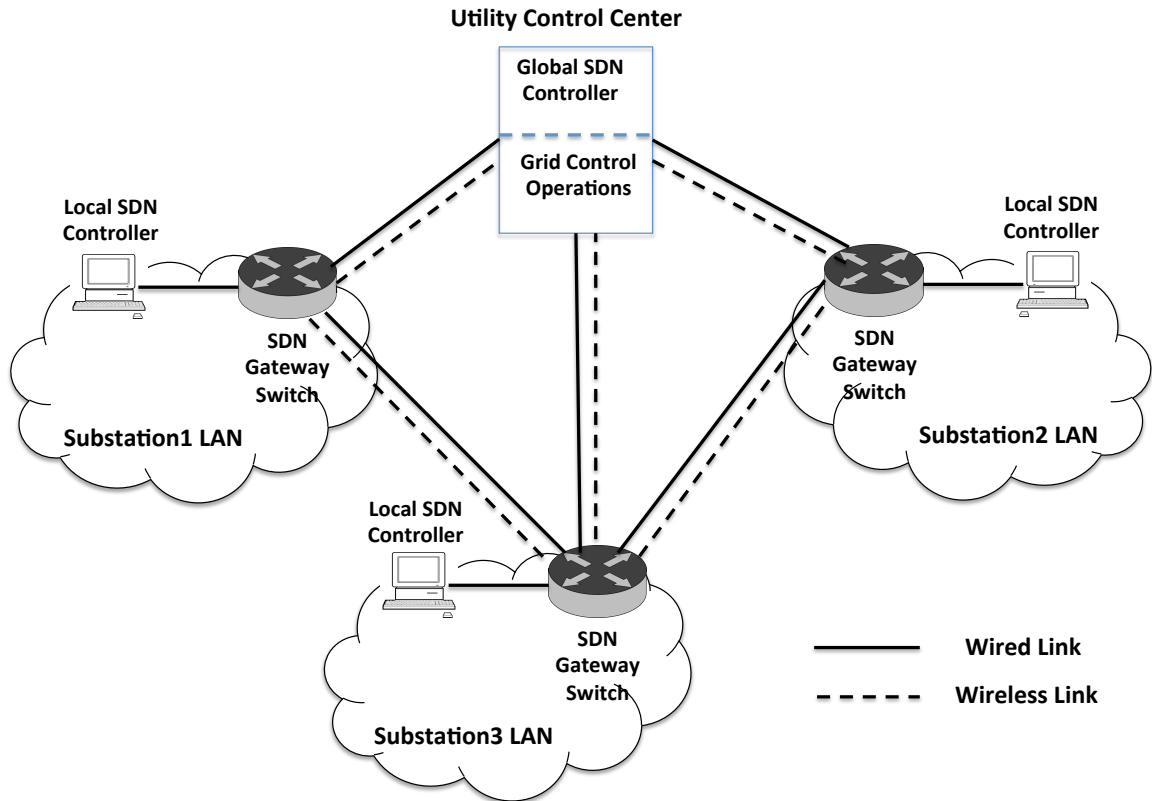
Figure 5.1: Proposed SDN model for substation communications

5.1.2 Mininet Setup and Integration of ns-3 with Mininet

Note that extensive performance evaluation of the proposed SDN-enabled architecture is not possible in a real substation network since access to these facilities is not always possible. In addition, since these systems are typically large-scale and spans different geographical regions, setting up small prototypes in a lab environment for testing purposes will not be adequate to do various evaluations even though the cost can be covered. Therefore, research mostly rely on emulators that are freely available. For SDN, Mininet [LHM10] has been evolved as one of the most widely used tool based on the Linux protocol stack. It is open source and has been implemented on Python. Mininet nodes come with Openflow protocol support. It has an ability to work with internal SDN controller as default, as well as external controllers

such as OpenDayLight, NOX. While it can produce various virtual nodes in a single computer environment, Mininet can only mimic the behavior of protocols that are available at the Linux protocol stack.

For this, we propose extending Mininet [LHM10] topology capabilities in terms of supporting redundant and heterogeneous links using network function virtualization (NFV). Specifically, we propose hosting virtual interfaces on top of the wired physical interface for making connections to: 1) SDN controller using a separate wired connection; 2) Utility control center using wired PLC connection; and 3) Other substations for exchanging IEC 61850 MMS data through wired PLC. We also add a wireless connection by introducing a new wireless interface for the switch that acts as the gateway for the substation. This wireless connection will be using MMS protocol for communication with the other substations. The underlying protocols can be either IEEE 802.11n-based for shorter distances or IEEE 802.11ah or LTE for longer distances. Within the substation, there can be various other link layer protocols. All these interfaces are shown in Fig. 5.2.
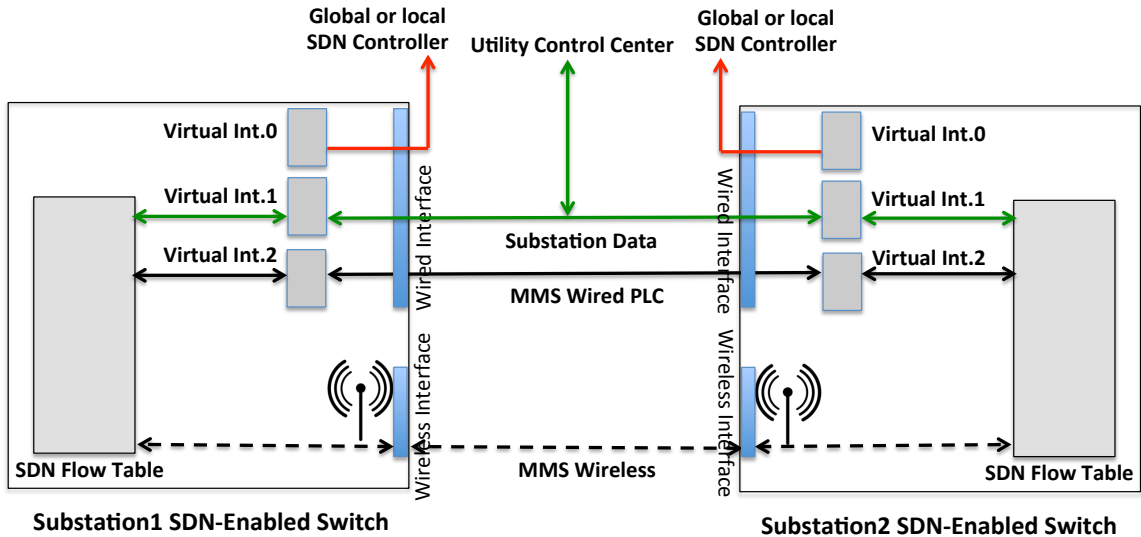

Figure 5.2: Proposed connection of two substations over WAN.

Nonetheless, there are a many wired/wireless protocols that are employed in SCADA systems, AMI and substation automation in Smart Grid, which cannot be implemented and tested under Mininet. This not only limits the research capabilities regarding SDN deployments but also is not flexible in terms of testing network resilience, fault-tolerance, real-time behavior and security at the nodes when SDN is deployed. Therefore, there is a great need to integrate Mininet with one of the existing network simulators for comprehensively evaluating the effectiveness of SDN-based control on the Smart Grid communication networks.

The proposed integration model is shown in Fig. 5.3. As can be seen, we propose using *Tapbridge* object in ns-3, which effectively allows host systems and virtual machines running native applications and protocol stacks to integrate with an ns-3 simulation. In our case, ns-3 connects to a *Virtual TAP Device* interface created on Mininet. Packets sent by Mininet host to the *Virtual TAP Device* are transmitted through a file descriptor to the ns-3 process. Next they are forwarded down by *Tapbridge* to the *ns-3 Net Device* and transmitted over the ns-3 emulated channel. This allows us to analyze the behavior of native protocol suites (such as 802.11, LTE etc) in large-scale networks that may not be supported by Mininet.
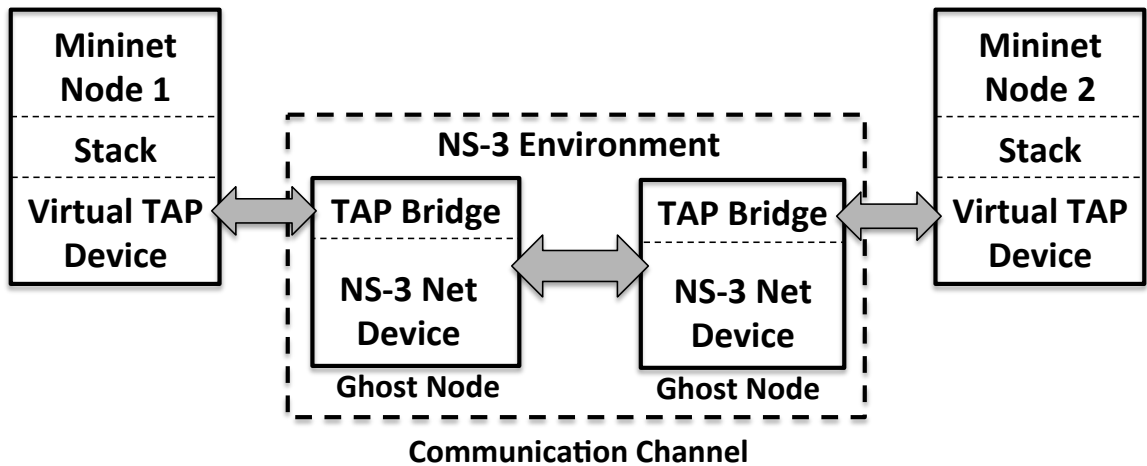

Figure 5.3: Modeling channels in Mininet using ns-3 features.

5.2   Experimental Evaluation

5.2.1   Experiment Setup

In order to evaluate the performance of our proposed work, we utilized Mininet as mentioned before for the Openflow protocol from the SDN controller to switches and used OpenDayLight SDN controller. Network topologies were created with Mininet and were integrated with ns-3 links as explained in the previous section. To generate the IEC 61850 MMS traffic, we used IEC 61850 based industrial feeder relays from the FIU Smart Grid Testbed. MMS is a data generation model based on publisher-subscriber paradigm within IEC 61850 framework. This data was generated at publisher IEDs to subscribers at 4ms intervals in compliance with the IEC 61850 framework. The data was read via a custom Python program at IED1 and transmitted via the network using TCP or UDP sockets. Data arrival times were also computed by the same Python program that checked the time labels of packets at the destination IED2. In all the experiments, we run the simulation for 60 seconds. The link failure was performed at the 30th second of the simulation from Mininet. The SDN controller then realizes the failure and updates the flow tables. In these experiments, we assessed the performance of this recovery process under a variety of networking conditions with different metrics as explained next. Note that we repeated each experiment for 30 times for statistical significance and report the average of the results.

5.2.2   Benchmarks and Metrics for Testing

In order to test the proposed SDN architecture and resilience features, we created three different topologies:
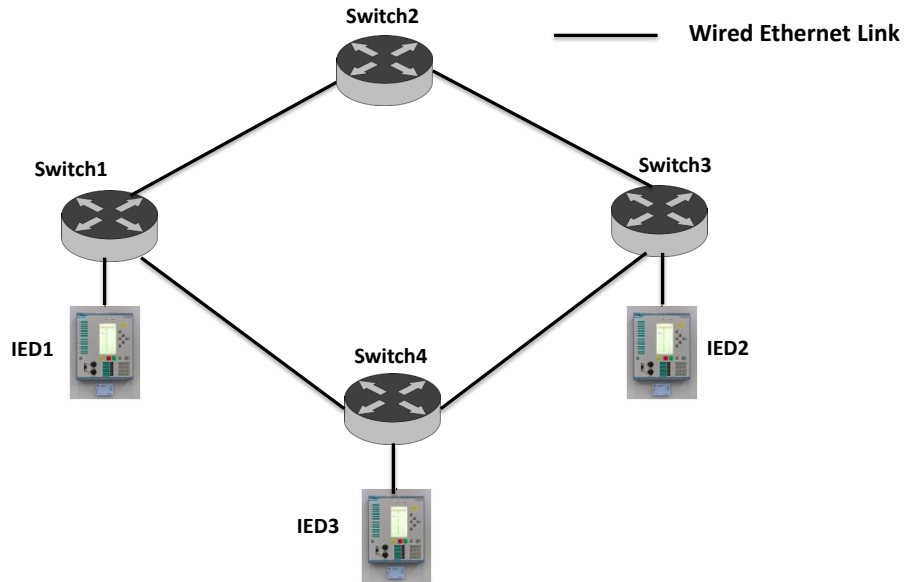
Figure 5.4: Baseline network topology used in evaluations.

1) *Baseline Topology*: This is the normal topology where we had a single link among the substation switches. This is used for comparison purposes and displayed at Figure 5.4; 2) *Proposed Topology*: This is the topology we propose for resiliency as shown in Figure 5.6. We were able to define a wireless link among two switches Switch1 and Switch2. The wireless protocol for testing purposes was IEEE 802.11a using using the ns-3 channel described earlier; and 3) *Test Topology*: We also aimed to compare the performance of our proposed topologies with the case when we have double wired links between Switch1 and Switch2 as shown in Figure 5.5 . The links were based on 100Mbps Ethernet simulated via ns-3. These topologies are referred to as Baseline, Proposed/Wireless and Proposed/Test respectively in the rest of the section. In addition, we consider two cases for TCP and UDP protocols respectively. This is because MMS works above the IP layer and thus we considered the impact on the performance of TCP connection establishment and how recovery was performed.
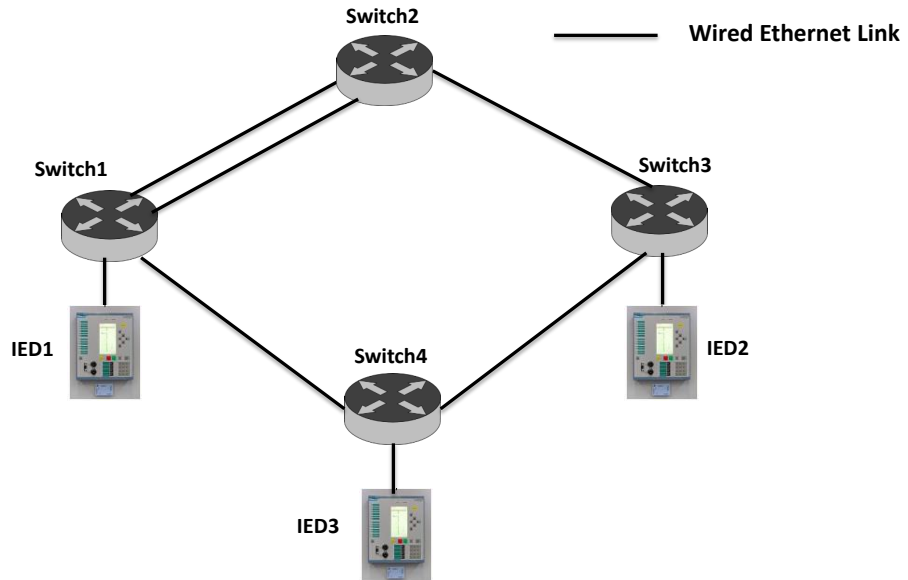
Figure 5.5: Network topology used to compare with proposed one in evaluations.

As performance metrics, we considered the end-to-end delay and packet loss metrics from source to a particular sub-station destination using the three different topologies with both TCP and UDP. Since myriads of packets were sent, we computed the average delay and packet loss ratio for all the packets.

5.2.3 Performance Results

End-to-end and Recovery Delay. One of the crucial things in the recovery is the total time to restore the connection. Therefore, we conducted TCP and UDP experiments for each of the topologies. Due to space constraints, we only reported the end-to-end delay of the Proposed/Wireless approach for each packet under TCP and UDP as seen in Fig. 5.9 and Fig. 5.12 considering one particular experiment run.
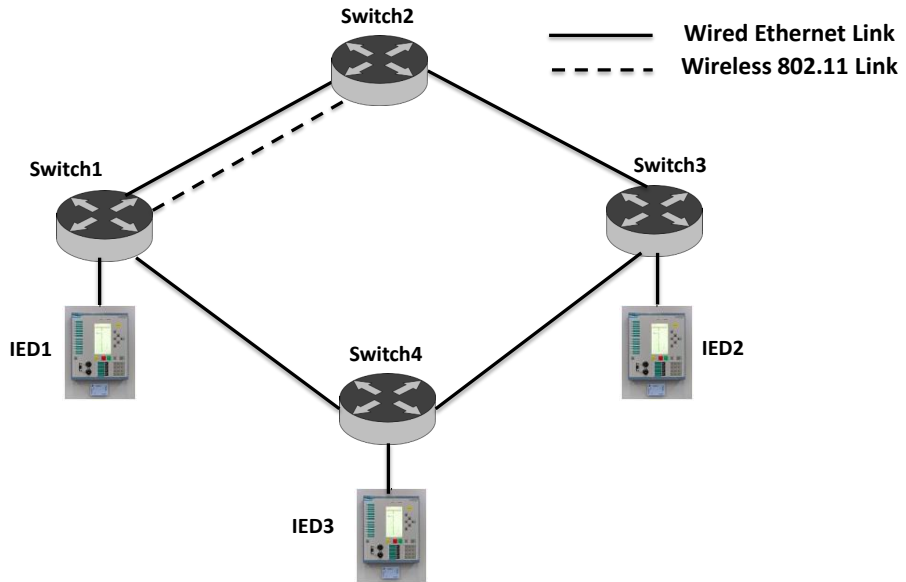
37

Figure 5.6: Proposed network topology used in evaluations.

The delay distribution for UDP demonstrates that there is only a peak of delay around 30th second for a few number of packets. Once the flow is changed to wireless, the delay becomes stable though it increases slightly due to wireless channel bandwidth. We also note that the recovery time in the Proposed/Wireless approach is totaling around 12.1ms on average, which is much better than that of MPLS-based Ethernet auto-negotiation [LLS04] reported as 29.54ms (on UDP). The recovery delay is also acceptable for Smart Grid monitoring applications, which require a latency of 10-100ms [DKG$^+$14].

However, same observations are not valid for TCP due to its reliability mechanisms such as re-transmissions and slow start with wireless. There is still a peak during the recovery time, but similar peaks may occur later due to wireless characteristics. When we closely analyzed the starting times of the packet transmissions, we realized that there is a transient period between 30th and 31st seconds where there
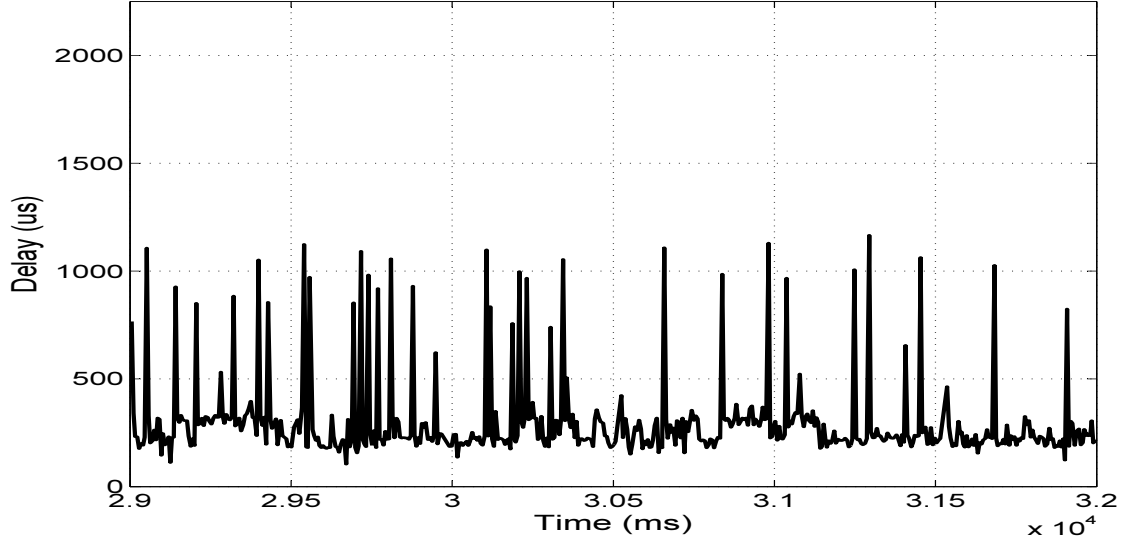
Figure 5.7: End-to-end Delay packet distribution for the Baseline Topology under TCP between the 29th and 32nd seconds of the simulation.

was no transmissions occurring. We observed that this was due to re-establishing the TCP connection phase (i.e., SYN-SYN/ACK-ACK) between the source and destination even though the link failure was handled. This is an interesting outcome due to changing ports of the ACK message coming back to Switch1 which might have triggered a TCP connection re-establishment. In total, we observed that there was about 1050ms in average impasse time for Proposed/Wireless approach. Obviously, this is much higher than UDP due to connection re-establishment. However, such increase brings the benefit of reliability as will be discussed in the next subsection.

Since the above results are only showing a particular shot, we wanted to test the significance of the proposed approach by computing the average of all 30 runs. To do this, we separately reported the behavior of packet delay before, during and after the failure by dividing the total time into three slots as Time1 (0-29sec), Time2 (30-31sec) and Time3 (31-60) respectively. As can be seen in Table 5.1, the delay increases during the period of failure of the main wired link between Switch1 and 2 for all cases. This is expected since there is the time for the controller to detect the

39

Figure 5.8: End-to-end Delay packet distribution for the Topology with 2 Wired Links under TCP between the 29th and 32nd seconds of the simulation.
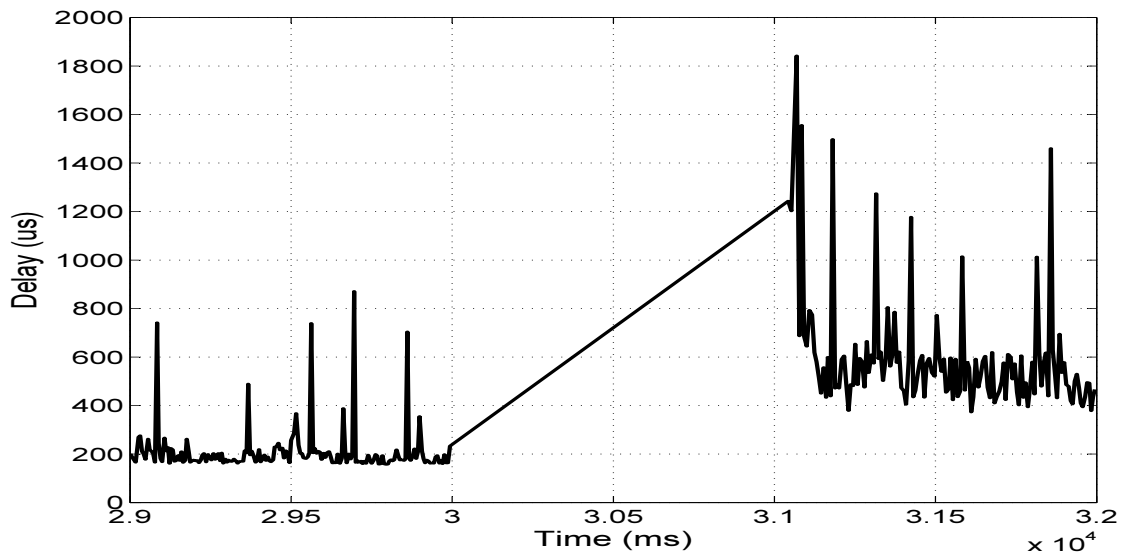
Figure 5.9: End-to-end Delay packet distribution for the Proposed/Wireless Approach under TCP between the 29th and 32nd seconds of the simulation.

failure and update the flow table of Switch1. However, the increase is much more in the wireless case which can be attributed to the nature and bandwidth of the wireless communication under TCP. For the packets that are lost due to link failure, there are retransmissions and these are taking more time due to poor performance of TCP

Figure 5.10: End-to-end Delay packet distribution for the Baseline Topology under UDP between the 29th and 32nd seconds of the simulation.

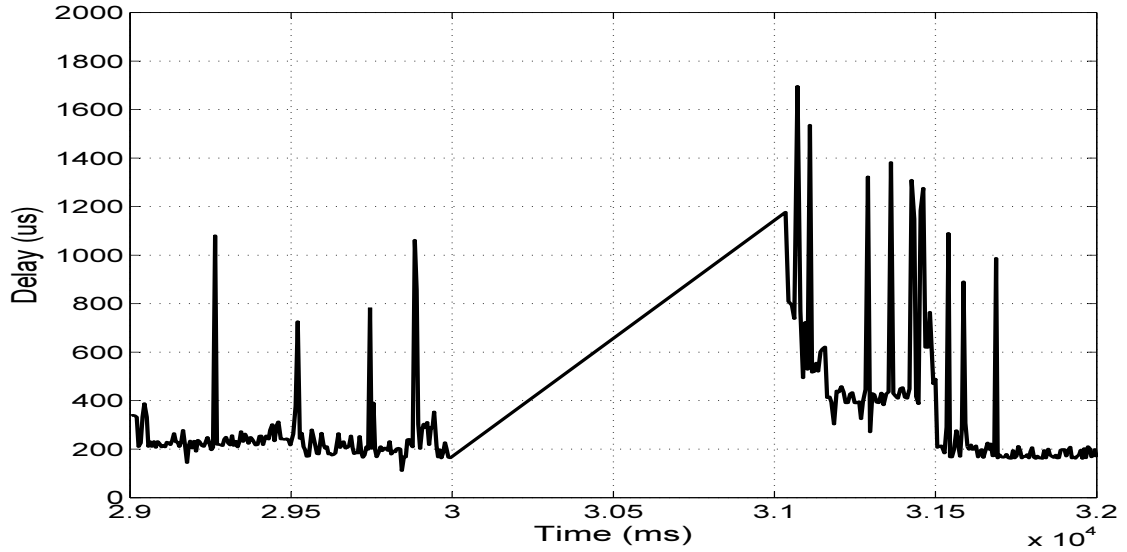
Figure 5.11: End-to-end Delay packet distribution for the Topology with 2 Wired Links under UDP between the 29th and 32nd seconds of the simulation.
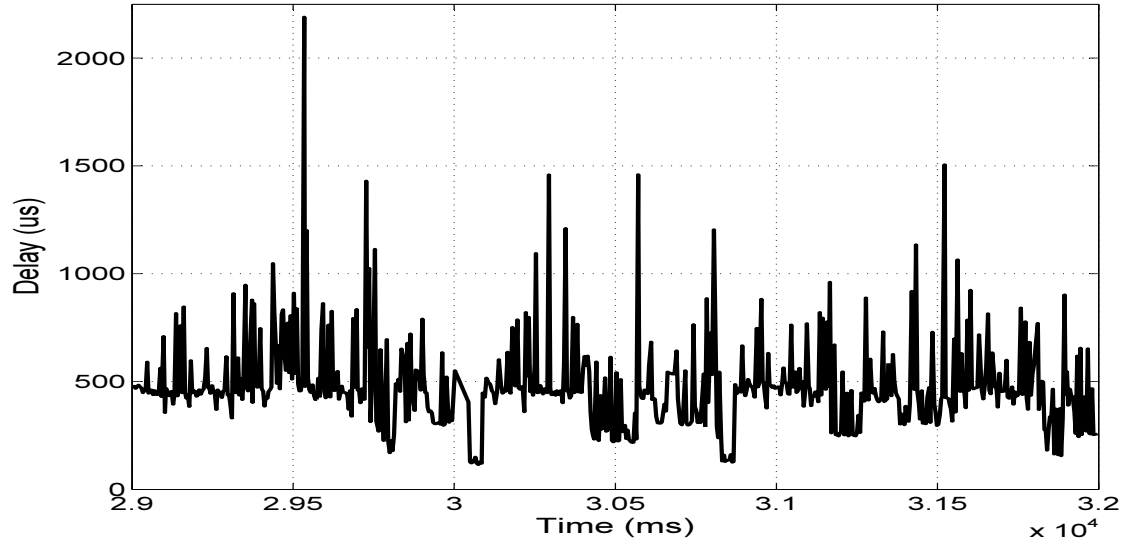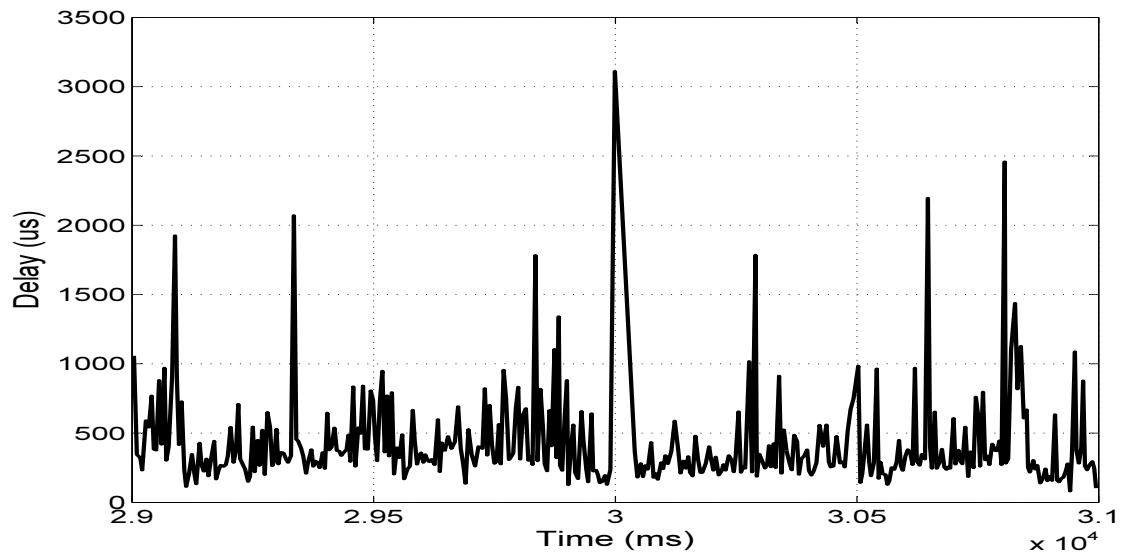
with wireless link failures [XPMS01] and also due to limited capacity in terms of channel capacity.
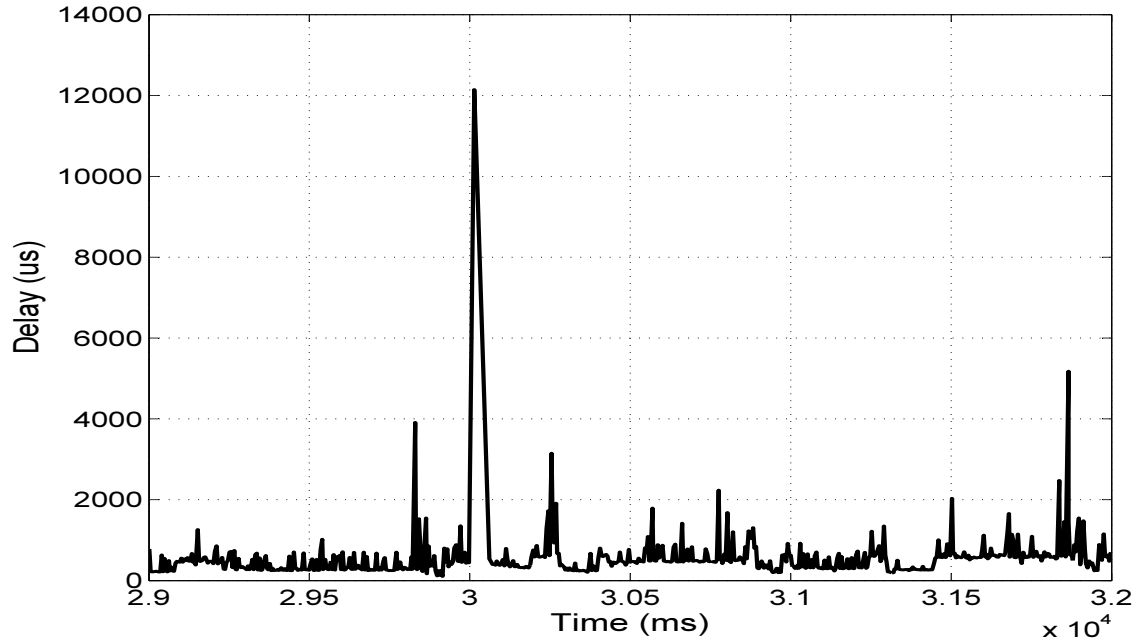
Figure 5.12: End-to-end Delay packet distribution for Proposed/Wireless Approach under UDP between the 29th and 32nd seconds of the simulation.

When we repeated the same experiments with UDP, we have seen a similar pattern but this time the gaps are not large (Table 5.2). This can be explained as follows: Since there is no retransmission mechanism in UDP, the lost packets are not retransmitted and thus their delays are not increased as in the case of TCP. Although there is still a delay due to recovery time, this does not dominate the overall packet delay.

Table 5.1: TCP Packet Transmission Times

Network Topology	Avg. Transmission Delay (microsecond)		
	Time-1	Time-2	Time-3
Proposed/Wireless	346	825	547
Proposed/Test	264	371	271
Baseline	295	301	290

Packet Loss. One of the other concerns is the possibility of packet loss when they are in transit to Switch2 and the link fails. During the same experiments conducted above, we also reported the number of lost packets for both TCP and UDP under all

Table 5.2: UDP Packet Transmission Times

Network Topology	Avg. Transmission Delay (microsecond)		
	Time-1	Time-2	Time-3
Proposed/Wireless	481	582	591
Proposed/Test	486	482	475
Baseline	494	452	460

approaches. The results are shown in Tables 5.3 and 5.4. Looking at these results, we observe that TCP can perfectly recover all the packets that might get lost due to switching to the redundant links for both wired and wireless cases. However, this is not the case for UDP where some of the packets can get loss compared to the baseline case where we do not have any link failures. The reason is due to the end-to-end reliability mechanism of TCP which retransmits the lost packets at the destination. Note that we determined in the previous subsection that TCP re-establishes the connection among the source and destination when a link in the path is broken. In this way, lost packets are re-transmitted through this new connection.

Table 5.3: Total Packet and Packet Loss at TCP

Network Topology	Total Packet Number	Packet Loss
Proposed/Wireless	8552	0
Proposed/Test	9499	0
Baseline	10197	0

Table 5.4: Total Packet and Packet Loss at UDP

Network Topology	Total Packet Number	Packet Loss
Proposed/Wireless	13266	3
Proposed/Test	13158	5
Baseline	13182	0

CHAPTER 6

CONCLUSION

We explained how the emerging SDN paradigm could be considered as a viable technology for the Smart Grid communication architecture, which is currently under massive modernization effort by the utility providers. We discussed how flexibility and ease of control, management, security, and maintenance provided by the SDN technology could make a compelling case for applying SDN in three unique smart grid deployments: Specifically, we focused on SCADA systems, AMI, and Microgrid Systems and discussed how an increasing number of smart grid devices that are being deployed to connect all the components of the Smart Grid together could benefit from the SDN. Finally, we summarized the existing research challenges regarding the security of SDNs in general before discussing potential security threats with SDN-enabled Smart Grid. We provided some general solutions to alleviate the threats. Applying the maturing SDN technology into the smart grid infrastructure presents ample unique research challenges in security and networking to engineers and scientists.

In addition, we focused on MMS-based substation communications and developed an SDN-enabled network for testing in Mininet. The communication channels were simulated using ns-3 and real Smart Grid traffic. Our goal was to test the ability of SDN to recover failed links in real-time without losing any packets and significantly increasing the packet delay.

Evaluation results indicated that SDN can provide seamless resiliency in case of the availability of redundant wireless links for real-time protocols such as MMS. Despite a minor delay increase in UDP, the maximum packet delay is still within the bounds of monitoring applications and the recovery time is much less than that of a route update or MPLS-based approach. For TCP, we observed that there is a

higher delay to restart transmissions due to re-establishment of the TCP connection without loosing any packets. In the future, we plan to evaluate LTE and other protocols that can be used to send data.

BIBLIOGRAPHY

[ALW+14]   Ian F. Akyildiz, Ahyoung Lee, Pu Wang, Min Luo, and Wu Chou.
             A roadmap for traffic engineering in sdn-openflow networks. *Comput. Netw.*, 71:1–30, October 2014.

[BAH+12]   Charlie Baker, Ashiq Anjum, Richard Hill, Nik Bessis, and
             Saad Liaquat Kiani. Improving cloud datacentre scalability, agility
             and performance using openflow. In *Intelligent Networking and
             Collaborative Systems (INCoS), 2012 4th International Conference on*, pages 20–27. IEEE, 2012.

[BDD+10]   Kenneth C Budka, Jayant G Deshpande, Tewfik L Doumi, Mark
             Madden, and Tim Mew. Communication network architecture
             and design principles for smart grids. *Bell Labs Technical Journal*,
             15(2):205–227, 2010.

[CHHK13]   Adam Cahn, Jose Hoyos, Matthew Hulse, and Eric Keller.
             Software-defined energy communication networks: From substa-
             tion automation to future smart grids. In *Smart Grid Communi-
             cations (SmartGridComm), 2013 IEEE International Conference
             on*, pages 558–563. IEEE, 2013.

[DKB11]   P. Dely, A. Kassler, and N. Bayer. Openflow for wireless mesh
             networks. In *Computer Communications and Networks (ICCCN),
             2011 Proceedings of 20th International Conference on*, pages 1–6,
             July 2011.

[DKG+14]   Nils Dorsch, Fabian Kurtz, Hanno Georg, Christian Hagerling,
             and Christian Wietfeld. Software-defined networking for smart
             grid communications: Applications, challenges and advantages.
             In *Smart Grid Communications (SmartGridComm), 2014 IEEE
             International Conference on*, pages 422–427. IEEE, 2014.

[dOSdSPRK12] Flavio de Oliveira Silva, Joao Henrique de Souza Pereira, Pe-
             dro Frosi Rosa, and Sergio Takeo Kofuji. Enabling future inter-
             net architecture research and experimentation by using software
             defined networking. In *Software Defined Networking (EWSDN),
             2012 European Workshop on*, pages 73–78. IEEE, 2012.

[GBHB09]   Harald Gjermundrod, David E Bakken, Carl H Hauser, and Anjan
             Bose. Gridstat: A flexible qos-managed data dissemination frame-

work for the power grid. *Power Delivery, IEEE Transactions on*, 24(1):136–143, 2009.

[GdSDKAW+15] Eduardo Germano da Silva, Luis Augusto Dias Knob, Juliano Araujo Wickboldt, Luciano Paschoal Gaspary, Lisandro Zambenedetti Granville, and Alberto Schaeffer-Filho. Capitalizing on sdn-based scada systems: An anti-eavesdropping case-study. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 165–173. IEEE, 2015.

[GKRC13] A. Goodney, S. Kumar, A. Ravi, and Y.H. Cho. Efficient pmu networking with software defined networks. In *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, pages 378–383, Oct 2013.

[HHB14] Fei Hu, Qi Hao, and Ke Bao. A survey on software-defined network and openflow: from concept to implementation. *Communications Surveys & Tutorials, IEEE*, 16(4):2181–2206, 2014.

[KKMB14] David Klingel, Rahamatullah Khondoker, Ronald Marx, and Kpatcha Bayarou. Security analysis of software defined networking architectures: Pce, 4d and sane. In *Proceedings of the AINTEC 2014 on Asian Internet Engineering Conference*, page 15. ACM, 2014.

[KKS13] R. Kloti, V. Kotronis, and P. Smith. Openflow: A security analysis. In *Network Protocols (ICNP), 2013 21st IEEE International Conference on*, pages 1–6, Oct 2013.

[LHM10] Bob Lantz, Brandon Heller, and Nick McKeown. A network in a laptop: Rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, pages 19:1–19:6, New York, NY, USA, 2010. ACM.

[LLS04] Yi Lei, Chung-Horng Lung, and A. Srinivasan. A cost-effective protection and restoration mechanism for ethernet based networks: an experiment report. In *High Performance Switching and Routing, 2004. HPSR. 2004 Workshop on*, pages 350–354, 2004.

[LUB14] Wenyi Liu, A.S. Uluagac, and R. Beyah. Maca: A privacy-preserving multi-factor cloud authentication system utilizing big

data. In *Computer Communications Workshops (INFOCOM WK-SHPS), 2014 IEEE Conference on*, pages 518–523, April 2014.

[Mil10]   Brian Miller. Concept for next generation phasor measurement: A low-cost, self-contained, and wireless design. Master's thesis, University of Tennessee, 2010.

[min]   Mininet emulator.

[MVKK12]   Vijay Mann, Anilkumar Vishnoi, Kalapriya Kannan, and Shivkumar Kalyanaraman. Crossroads: Seamless vm mobility across data centers through software defined networking. In *Network Operations and Management Symposium (NOMS), 2012 IEEE*, pages 88–96. IEEE, 2012.

[NMG⁺14]   V. Nascimento, M. Moraes, R. Gomes, B. Pinheiro, A. Abelem, V.C.M. Borges, K.V. Cardoso, and E. Cerqueira. Filling the gap between software defined networking and wireless mesh networks. In *Network and Service Management (CNSM), 2014 10th International Conference on*, pages 451–454, Nov 2014.

[ope]   Open networking foundation.

[SAB14]   L. Schehlmann, S. Abt, and H. Baier. Blessing or curse? revisiting security aspects of software-defined networking. In *Network and Service Management (CNSM), 2014 10th International Conference on*, pages 382–387, Nov 2014.

[SAU12]   Nico Saputro, Kemal Akkaya, and Suleyman Uludag. A survey of routing protocols for smart grid communications. *Computer Networks*, 56(11):2742 – 2771, 2012.

[SB15]   William Stallings and Lawrie Brown. *Computer Security: Principles and Practice (3rd edition)*. Prentice Hall, 2015.

[SG13]   Seungwon Shin and Guofei Gu. Attacking software-defined networks: A first feasibility study. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 165–166. ACM, 2013.

[SHOS13]   S. Scott-Hayward, G. O'Callaghan, and S. Sezer. Sdn security: A survey. In *Future Networks and Services, 2013 IEEE SDN for*, pages 1–7, Nov 2013.

[SNS+13]   Ali Sydney, James Nutaro, Caterina Scoglio, Don Gruenbacher, and Noel Schulz. Simulative comparison of multiprotocol label switching and openflow network technologies for transmission operations. *Smart Grid, IEEE Transactions on*, 4(2):763–770, 2013.

[Sta03]   William Stallings. *Cryptography and Network Security: Principles and Practices (3rd edition)*. Prentice Hall, 2003.

[SYPG13]   Seungwon Shin, Vinod Yegneswaran, Phillip Porras, and Guofei Gu. Avant-guard: Scalable and vigilant switch flow management in software-defined networks. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 413–424. ACM, 2013.

[UCS]   Ucsd microgrid.

[X.891]   ITU-T Recommendation X.800. Security architecture for open systems interconnection for ccitt applications. 1991.

[XPMS01]   George Xylomenos, George C Polyzos, Petri Mähönen, and Mika Saaranen. Tcp performance issues over wireless links. *Communications Magazine, IEEE*, 39(4):52–58, 2001.

[ZSLF13a]   Jianchao Zhang, Boon-Chong Seet, Tek-Tjing Lie, and Chuan Heng Foh. Opportunities for software-defined networking in smart grid. In *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*, pages 1–5. IEEE, 2013.

[ZSLF13b]   Jianchao Zhang, Boon-Chong Seet, Tek-Tjing Lie, and Chuan Heng Foh. Opportunities for software-defined networking in smart grid. In *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*, pages 1–5, Dec 2013.

[ZWHJ14]   Boyang Zhou, Chunming Wu, Xiaoyan Hong, and Ming Jiang. Programming network via distributed control in software-defined

networks. In *Communications (ICC), 2014 IEEE International Conference on*, pages 3051–3057, June 2014.