

Software Defined Networking for Resilient Communications in Smart Grid Active Distribution Networks

Abdullah Aydeger, Kemal Akkaya, Mehmet H. Cintuglu, A. Selcuk Uluagac and Osama Mohammed
 Dept. of Electrical & Computer Engineering, Florida International University, Miami, FL, 33174 USA
 {aayde001, kakkaya, mcint015, auluagac, mohammed}@fiu.edu

Abstract—Emerging Software Defined Networking (SDN) technology provides excellent flexibility to large-scale networks in terms of control, management, security, and maintenance. In this paper, we propose an SDN-based communication infrastructure for Smart Grid distribution networks among substations. A Smart Grid communication infrastructure consists of a large number of heterogeneous devices that exchange real-time information for monitoring the status of the grid. We then investigate how SDN-enabled Smart Grid infrastructure can provide resilience to active distribution substations with self-recovery. Specifically, by introducing redundant and wireless communication links that can be used during the emergencies, we show that SDN controllers can be effective for restoring the communication while providing a lot of flexibility. Furthermore, to be able to effectively evaluate the performance of the proposed work in terms of various fine-grained network metrics, we developed a Mininet-based testing framework and integrated it with ns-3 network simulator. Finally, 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 during the times of link failures in reliable fashion.

Keywords: SDN; wireless networks; smart grid; substations; resilience

I. INTRODUCTION

The proliferation of smart devices with their ability to access the Internet created new avenues to build large-scale cyberphysical 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 [1]. 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 a similar fashion, the Power Grid can also benefit from the emerging SDN technology. Indeed, the existing Power Grid in the US is going through a massive transformation to make

it more reliable and connected with the ability to transfer data and power in two-ways, referred to as Smart Grid [2]. 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 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. 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 [1]. 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.

This work leverages SDN-enabled devices and SDN controllers to build a resilient communication infrastructure at the distribution level among the substations. Our contributions are threefold: First, we propose a novel SDN architecture that can be used for the Smart Grid at the distribution level. Second, we design and implement our proposed architecture by integrating and SDN emulator with a network simulator. Third, we test its resiliency by considering inter-substation communications that are exchanging IEC 61850 MMS protocol data among Intelligent Electronic Devices (IED) [3].

In our evaluations, our main goal was to evaluate the performance of proposed SDN-based substation network architecture under a variety of conditions and reveal its benefits with respect to other existing solutions. For this, we utilized the widely used Mininet emulator [4] for the SDN research and

ns-3 simulator [5] for networking research and built a Mininet-based framework that could function as an inter-substation network with both wired and wireless links. While wired links are assumed to be based on PLC or Ethernet, wireless links are WiFi or LTE. We developed new SDN applications with the ability to define multiple links among two Mininet substations. Furthermore, to realistically evaluate the wireless connections, ns-3 was integrated with Mininet for the flow of data among the emulated substations by defining some virtual bridge structures. To the best of our knowledge, this work is the first to provide such an integration for a more detailed network-oriented evaluation. Using this novel integrated environment, we evaluated the recovery characteristics of the proposed SDN system by using real IEC 61850 Manufacturing Message Specification (MMS) data obtained from the Smart Grid Testbed at Florida International University (FIU) [6]. Specifically, we analyzed the recovery delay and packet loss metrics when SDN is on the control of the network and discussed the advantages/disadvantages with respect to other solutions under different conditions. The results show that SDN can provide a seamless self-recovery with almost a negligible overhead on a few packets when UDP is used.

This paper is organized as follows. In Sections II and III, we discuss the relevant work and provide some background on SDN. In Section IV, we introduce the proposed SDN-enabled inter-substation network architecture. Detailed performance evaluation of the proposed work is given in Section V. Finally, we conclude the paper in Section VI.

II. RELATED WORK

The use of SDN in Smart Grid applications has just started to be considered recently [7]. 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 [8] within an SDN architecture [9]. 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, [10] studied SDN usage for QoS in wide area 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 [11]. 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, [12] 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 [13], 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.

III. PRELIMINARIES

A. SDN Advantages

SDN's main advantage and motivation is to move the control of the lookup tables inside the network devices to a separate location so that it can be controlled more easily centrally. 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 terminology: 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.

One main problem with the traditional network settings involves updating the network elements after configuration, topology changes, which requires too much effort and time [1]. By creating a programming interface to be able to update network elements centrally (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 is given in Fig. 1. As can be seen from this figure, the control plane is typically controlled by a single center through some applications.

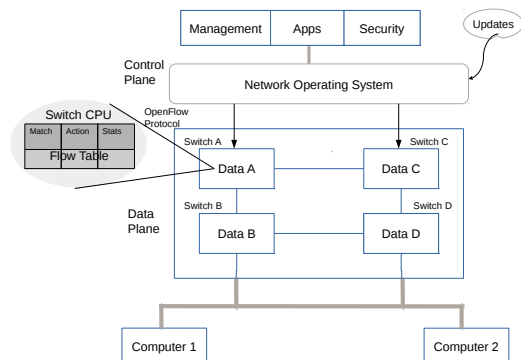


Fig. 1: SDN-based Network with data and control planes

In fact, today, SDN can be 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) [14]. Furthermore, due to devices and applications from different vendors, the equipments may not be interoperable. Therefore, the companies will need to deal with equipment maintenance and software upgrade, which is costly and laborous. SDN is a promising solution for the aforementioned problems.

B. Smart Grid Substation Networking Architectures

Smart Grid has three main components for generation, transmission and distribution of the power. For transmission and distribution, substations at different geographical locations are used. For each of these components, transmission and distribution data networks are also used for communicating the collected field data.

Utilities and energy companies own and operate network components that are part of the Smart Grid communication infrastructure. The management of these networks is a great challenge due to the scale. With each upgrade to the infrastructure, Smart Grid distribution networks are becoming more complicated to operate due to high penetration of renewable resources, energy storages, and micro-grids. Online information management is crucial to be able to maintain secure and reliable operation of this large-scale cyber-physical system. Similarly, distribution substations are the main control points for monitoring, control, and in particular transforming voltage level from medium voltage to low voltage level. A substation contains numerous IEDs, each generating various information about the status of some aspect of the substation. In new-generation substations, the IEC 61850 standard is used for substation automation and control [15]. SCADA systems are used to control centers to collect data from field devices such as PLCs, PMUs, and IEDs at substation in real-time and perform control decisions at the control center in terms of reliability and quality of service [15]. A sample active distribution network is shown in Fig. 2 along with the corresponding data network.

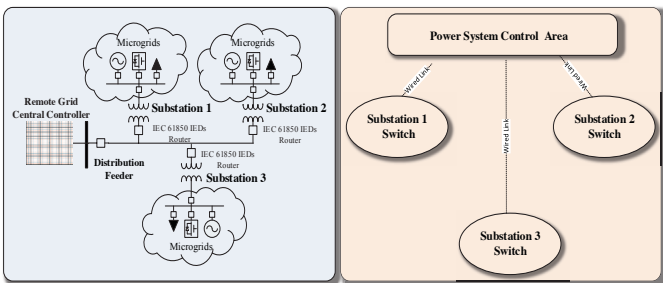


Fig. 2: Active distribution networks example connected to a control center.

In the next section, we propose how these components can be integrated with the SDN technology due to numerous advantages it brings.

IV. SDN-BASED INTER-SUBSTATION NETWORK

A. Proposed Model

We propose an SDN-based communication infrastructure that can be deployed both within and among the substations as shown in Fig. 3. 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.

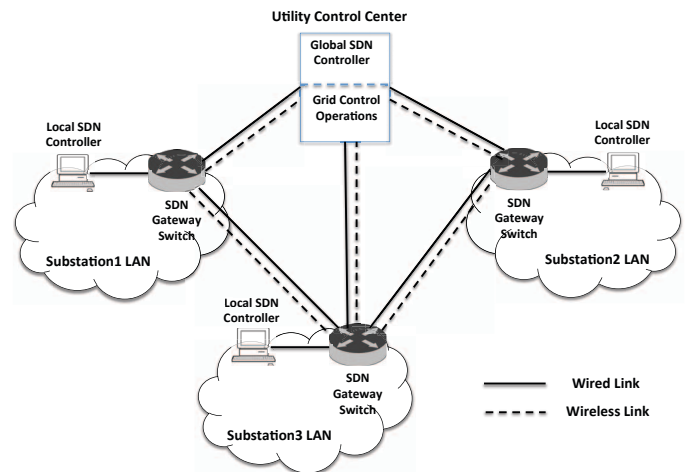


Fig. 3: Proposed SDN model for substation communications.

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.

B. 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 [4] 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 [4] 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. 4.

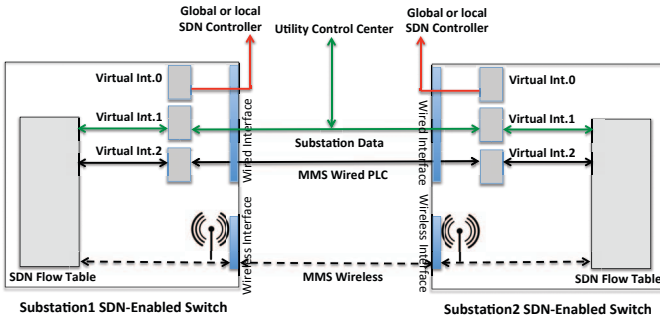


Fig. 4: Proposed connection of two substations over WAN.

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. 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.

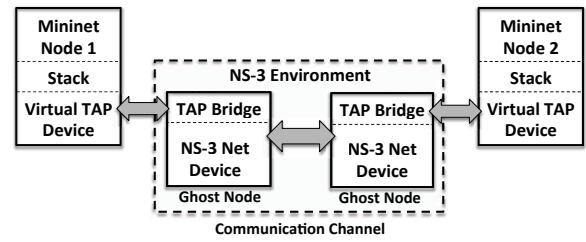


Fig. 5: Modeling channels in Mininet using ns-3 features.

V. EXPERIMENTAL EVALUATION

A. 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.

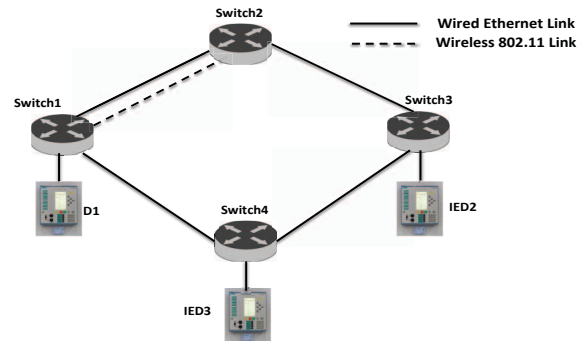


Fig. 6: Network topology used in Evaluations.

B. Benchmarks and Metrics for Testing

In order to test the proposed SDN architecture and resilience features, we created three different topologies:

1) *Baseline Topology*: This is the normal topology where we had a single link among the substation switches. This

is used for comparison purposes; 2) *Proposed Topology*: This is the topology we propose for resiliency as shown in Figure 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 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. 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.

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.

C. 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. 7 and Fig. 8 considering one particular experiment run.

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 [16] reported as 29.54ms (on UDP). The recovery delay is also acceptable for Smart Grid monitoring applications, which require a latency of 10-100ms [13].

However, some 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 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

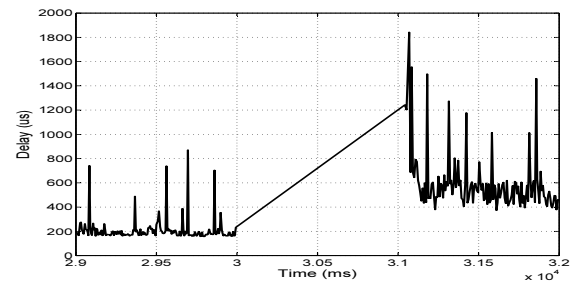


Fig. 7: End-to-end Delay packet distribution for the Proposed/Wireless Approach under TCP between the 29th and 32nd seconds of the simulation.

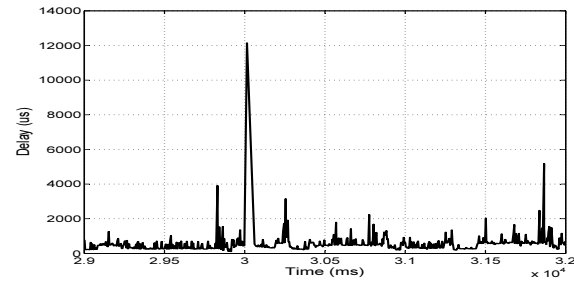


Fig. 8: End-to-end Delay packet distribution for Proposed/Wireless Approach under UDP between the 29th and 32nd seconds of the simulation.

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 I, 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 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 with wireless link failures [17] and also due to limited capacity in terms of channel capacity.

When we repeated the same experiments with UDP, we have seen a similar pattern but this time the gaps are not large (Table II). 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.

Packet Loss. One of the other concerns is the possibility of

TABLE I: 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

TABLE II: 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

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 approaches. The results are shown in Tables III and IV. 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 III: 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 IV: 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

VI. CONCLUSION

In this paper, we introduced 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. Specifically, 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 losing any packets. In the future, we plan to evaluate LTE and other protocols that can be used to send data.

REFERENCES

- [1] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: from concept to implementation," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 4, pp. 2181–2206, 2014.
- [2] N. Saputro, K. Akkaya, and S. Uludag, "A survey of routing protocols for smart grid communications," *Computer Networks*, vol. 56, no. 11, pp. 2742 – 2771, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128612001429>
- [3] R. Mackiewicz, "Overview of iec 61850 and benefits," in *Power Systems Conference and Exposition, 2006. PSCE '06. 2006 IEEE PES*, Oct 2006, pp. 623–630.
- [4] B. Lantz, B. Heller, and N. 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*, ser. Hotnets-IX. New York, NY, USA: ACM, 2010, pp. 19:1–19:6. [Online]. Available: <http://doi.acm.org/10.1145/1868447.1868466>
- [5] "Network simulator - ns - 3," <http://www.isi.edu/nsnam/ns/index.html>.
- [6] "Smart grid testbed at fu," <https://energy.fu.edu/2014/12/smart-grid-test-bed-lab/>.
- [7] J. Zhang, B.-C. Seet, T.-T. Lie, and C. H. Foh, "Opportunities for software-defined networking in smart grid," in *Information, Communications and Signal Processing (ICICIS) 2013 9th International Conference on*. IEEE, 2013, pp. 1–5.
- [8] B. Miller, "Concept for next generation phasor measurement: A low-cost, self-contained, and wireless design," Master's thesis, University of Tennessee, 2010.
- [9] A. Goodney, S. Kumar, A. Ravi, and Y. Cho, "Efficient pmu networking with software defined networks," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, Oct 2013, pp. 378–383.
- [10] A. Sydney, J. Nutaro, C. Scoglio, D. Gruenbacher, and N. Schulz, "Simulative comparison of multiprotocol label switching and openflow network technologies for transmission operations," *Smart Grid, IEEE Transactions on*, vol. 4, no. 2, pp. 763–770, 2013.
- [11] A. Cahn, J. Hoyos, M. Hulse, and E. Keller, "Software-defined energy communication networks: From substitution automation to future smart grids," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*. IEEE, 2013, pp. 558–563.
- [12] E. Germano da Silva, L. A. Dias Knob, J. Araujo Wickboldt, L. Paschoal Gaspary, L. Zambenedetti Granville, and A. Schaeffer-Filho, "Capitalizing on sdn-based scada systems: An anti-eavesdropping case-study," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE, 2015, pp. 165–173.
- [13] N. Dorsch, F. Kurtz, H. Georg, C. Hagerling, and C. Wietfeld, "Software-defined networking for smart grid communications: Applications, challenges and advantages," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*. IEEE, 2014, pp. 422–427.
- [14] V. Mann, A. Vishnoi, K. Kannan, and S. Kalyanaraman, "Crossroads: Seamless vm mobility across data centers through software defined networking," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*. IEEE, 2012, pp. 88–96.
- [15] K. C. Budka, J. G. Deshpande, T. L. Doumi, M. Madden, and T. Mew, "Communication network architecture and design principles for smart grids," *Bell Labs Technical Journal*, vol. 15, no. 2, pp. 205–227, 2010.
- [16] Y. Lei, C.-H. 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*, 2004, pp. 350–354.
- [17] G. Xylomenos, G. C. Polyzos, P. Mähönen, and M. Saaranen, "Tcp performance issues over wireless links," *Communications Magazine, IEEE*, vol. 39, no. 4, pp. 52–58, 2001.