Validation of SDN Emulator Based on Mininet and ONOS Controller for IEC 61850 Packet Delay Measurement

Saran Jitnukulsiri*† and Chaodit Aswakul†

* *Communication System Division, Electricity Generating Authority of Thailand (EGAT), Nonthaburi, Thailand*

† *Wireless Network and Future Internet Research Unit, Department of Electrical Engineering, Faculty of Engineering Chulalongkorn University, Bangkok, Thailand*

saran.ji@egat.co.th, chaodit.a@chula.ac.th

***Abstract*—To guarantee the electrical power stability for economic and industry growth, the power grid would need to adopt the technology of smart grid with the enablement of new communication technologies. IEC 61850 has been introduced for the upgrading of an electrical substation to digital substation composed of many digital devices that help automate the electrical power control for the purpose of power stability. *Sampled Values* is a basic message protocol for the application of Wide Area Measurement and Control (WAMPAC), which requires manageable communication in a wide area over Ethernet layer. In this regard, Software Defined Networking (SDN) has recently been introduced as one of the communication technologies with that capability. This paper has reported on our first-step findings towards the development of a tool to help checking the feasibility of SDN technology in smart grid applications. Since the smart grid applications must be delay-critical, timing performances become essential and all the modules to be used in the tool development must be verified on that aspect. Particularly, for this paper, *Mininet* and *ONOS controller* have been tested. Their ability to emulate transmission link delay values is quantified and reported for the practical range of smart grid application usages.**

***Index Terms*—Software defined networking, IEC 61850, smart grid, wide area networks**

I. Introduction

An electrical system is an important basis of a country for all the system in the entire energy chain. The stability of electrical power utility encourages the growth of economic and industry by ensuring the availability of energy sources for reliable productions. To increase the capacity of stability control in a power grid, the digital substation technology has been introduced by IEC 61850 [1] with the purpose to decrease the hard-wiring and increase the speed of protection and control in a substation. With the update of IEC 61850-90-5 and IEC 61850-90-12, the standard expands further into a wide area and establishes the required wide area communication for smart grid applications [2]. New protocols such as IEC 61850 Generic Object Oriented Substation Event (GOOSE) and Sampled Values (SV) are designed for rapid transfer of information between equipment of power system control to enhance the fault-tolerance and fast responses for a power system [3]. WAMPAC is one of the new smart grid applications that can be enabled by using the standard in order to gather wide-area data and react to the changing conditions of the transmission system in a coverage area of multiple substations.

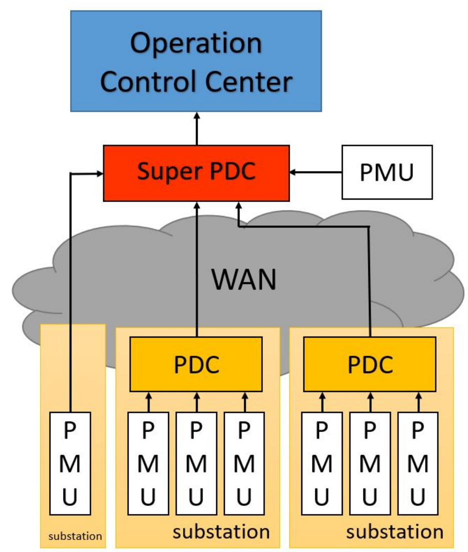

Fig. 1. Typical communication for PMUs and PDCs [4]

WAMPAC consist of Wide Area Monitoring (WAM) and Wide Area Protection and Control (WAPAC). WAM system uses Phase Measurement Unit system (PMUs) to gather data from various points of power system in a substation, and merge those pieces of information by Phase Data Concentrators (PDC) and send to the operation control center at another remote station [4]. The typical communication for PMUs and PDCs is shown in Fig. 1. WAPAC system includes the control and auto-protection for a power system in a wide area. A trip command is generated manually by an operator or automatically by Intelligence Electronic Device (IED) inside a substation to keep the stability of the power grid. In a digital substation, the command is transferred by GOOSE while important parameters of power system status are transferred by SV.

To achieve the fast protection and control in a power system, the communication system demands the requirement of message performance for the application in the order of milliseconds. Circuit networks running such as Time Division Multiplexing (TDM) has therefore been used widely in power utility because of the guaranteed time delay, circuit switching time, and asymmetrical delay for the delivery of messages, with the clear view for the operator by Network Management System (NMS). However, nowadays Packet switch networks (PSN) become more popular increasingly than legacy circuit networks, since the worldwide use of Internet Protocol (IP) and Multiprotocol Label Switching (MPLS). Future packet switch networks controlled by SDN evolves towards the managing and controlling ability by using the capability of an SDN controller, e.g. Open Network Operating System (ONOS) [5] and OpenFlow [6]. However, in utilizing this SDN controlled packet switch network, there is still a concern on the applicability to meet the delay requirement for transmission and distribution of smart power. Many open questions remain. For instance, how to achieve a good MPLS-TP (Multi-Protocol Label Switching Transport Profile) design for time critical application requirements in SDN [7]. An optical ground wire (OPGW) in the transmission line is used as the primary media for the connection of the wide area network. With the restriction from the optical pathway and long-range inter-substation connection, the network engineer has to prepare for the changing of the communication network well. Tools for network management and design need to be upgraded to prepare for the communication path quality of service and reliability in future smart grid applications.

II. SDN-EMULATED TESTBED FOR VALIDATION OF SV MESSAGE TRANSFER DELAY

In this paper, a network emulation tool has been developed for evaluating the communication time delay between virtual hosts, which function as PDCs in WAM system. The tool has been used to measure the delay and jitter of packets sent from one virtual host to another virtual host over a wide-area emulated Ethernet network. Recommended standard settings in [8] are used as our reference for SV-specific frame simulation.We use *Scapy* [9] to generate SV messages over Ethernet frames with the APDU frame size of 771 bytes, without any real power system information included. Also, only SV-header and APDU-header frame are simulated according to the SV standard frame, with the purpose for the detection in Wireshark program, while the rest of the packets are filled with dummy bits. The virtual hosts and communication infrastructure, i.e., OpenFlow switches and links, are emulated by Mininet [10] while network configuration and visualization are done by ONOS within each VM, as shown in Fig. 2. The emulation system is installed on Ubuntu 16.04 VM guest inside Oracle VM Virtual box with 4-core CPU Intel(R) Core(TM) i7-8750H CPU at 2.20GHz with 8192 MBytes of RAM and 50-GBytes space hard disk. The emulation use ONOS application named Virtual Private LAN Service (VPLS) to emulate Ethernet-based connection over L2VPNs for OpenFlow switches. All tests have been done under a CPU utilization lower than 80%, and RAM usage is generous.

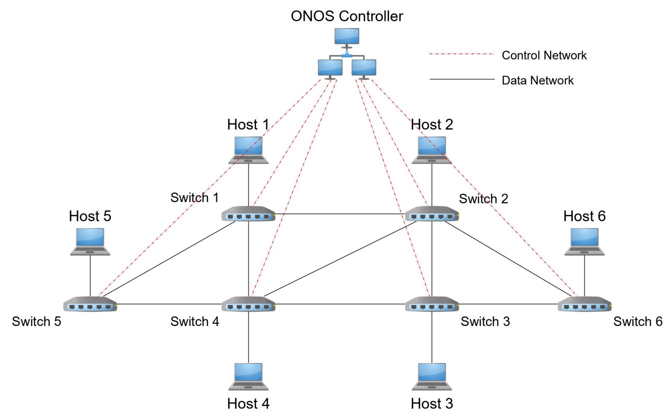

Fig. 2. Testbed SDN emulation by Mininet and ONOS

The network emulation must try to mimic the delay of packets in the precision of milliseconds in order to fulfill the requirements of communication infrastructure which serve time-critical smart grid applications. This requirement is referenced here according to latency classes for WANs in Table I [2].

TABLE I
LATENCY CLASSES FOR WANS [2]

Class	Latency	Use cases
TL1000	$\leq$ 1000 ms	All other messages
TL3000	$\leq$ 300 ms	Operator commands
TL100	$\leq$ 100 ms	Slow automatic interactions
TL30	$\leq$ 30 ms	Fast automatic interactions
TL10	$\leq$ 10 ms	Teleprotection
TL3	$\leq$ 3 ms	Differential protection

There are uncontrollable involvements which affect our emulation and results in packet delay *error* compared to initially intended settings. First is a virtual SDN characteristic, as a packet switch naturally creates processing delay, transmission delay, and queuing delay while transmitting data. We have considered the effect of this characteristic as part of our emulation since it can imitate the real switch devices behavior. The second is from TCLink function performance. TCLink in Mininet is based on the Linux traffic control which schedules to rearrange packets for the output port. TCLink cannot imitate the propagation delay of optical fiber or telecommunication link, which can be considered as our link input parameters in the emulation, correctly and can be a source of errors in the emulation. Besides, another concern is about the SV generation which should be able to mimic the SV publisher behavior. In this regard, we have developed a simple Scapy-based Python program to transmit SV packets with the periodic rate of 15, 30, 60, 120 and 240 packets per second to evaluate the error of time interval between adjacent SV packet generations. We also evaluate whether there is an effect from the change in the transmission rate on the timing of the emulated packet delay.

III. Experimental Results

A. Effect of TCLink adjustment

The delay between two hosts has been calculated from the difference in the timestamps, tagged by the virtual devices or hosts when receiving the SV packets. Define a_i for ith packet ($i = 1, \ldots, n$) as this measurable delay. In the testbed, a_i is measured from the difference between the packet's timestamps as captured by Wireshark at the source and destination host interfaces, respectively. The packet route is shown in Fig. 3.

Denote M as the target link delay, set by Mininet's TCLink. Here, M is varied from 2 ms to 500 ms, which covers the whole practical smart grid application range of concerns. The SV packet transmission rate is 120 packets per second with n = 3600 for each adjustment. Table II depicts the summary of obtainable delay time results.

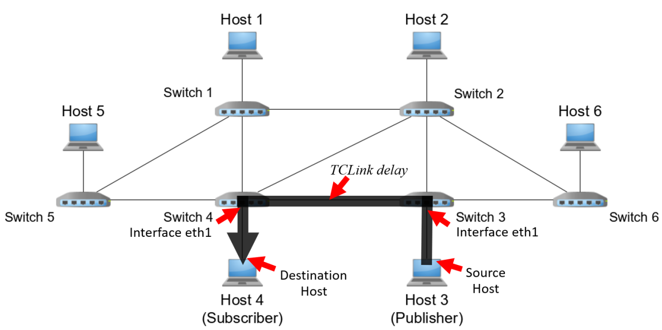

Fig. 3. Packet routing and interfaces in test of TCLink adjustment effect

TABLE II
AVERAGE DELAY OF ARRIVING PACKETS AT DIFFERENT VIRTUAL DEVICES ALONG HOST-TO-HOST PATH COMPARED WITH MININET LINK TARGET DELAY

Target Link Delay	Average Delay (ms) of Packets arriving at		
	Switch 3 -eth1	*Switch 4 -eth1*	*Destination Host*
2 ms	0.002	2.274	2.276
5 ms	0.002	5.509	5.511
10 ms	0.002	10.449	10.451
20 ms	0.002	20.476	20.478
50 ms	0.002	50.401	50.403
100 ms	0.002	100.439	100.441
200 ms	0.002	200.466	200.467
500 ms	0.002	500.604	500.606

The average error of emulated packet delay, compared with the target link delay M, is obtainable from:

$$\text{Average Error} = \left|\frac{1}{n}\sum_{i=1}^{n} a_i - M\right| \quad (1)$$

In addition, the percentage error of emulated packet delay, compared with the target link delay M, is obtainable from:

$$\text{Average Percentage Error} = \frac{\left|\frac{1}{n}\sum_{i=1}^{n} a_i - M\right|}{M} \times 100 \quad (2)$$

The results from (1) and (2) are shown in Fig. 4 and Fig. 5 the average error of emulated packet delay with 95% confidential interval and the average percentage error, respectively.

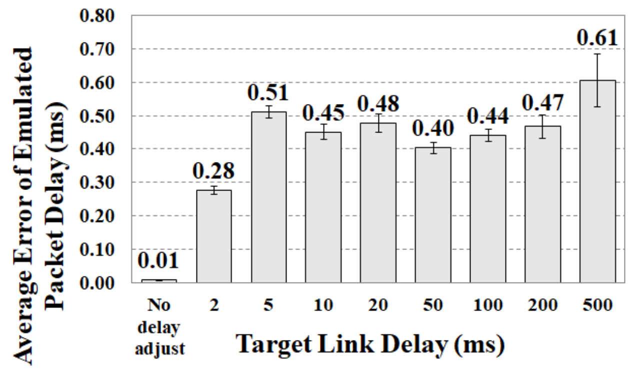

Fig. 4. Average error of emulated packet delay when target link delay is varied

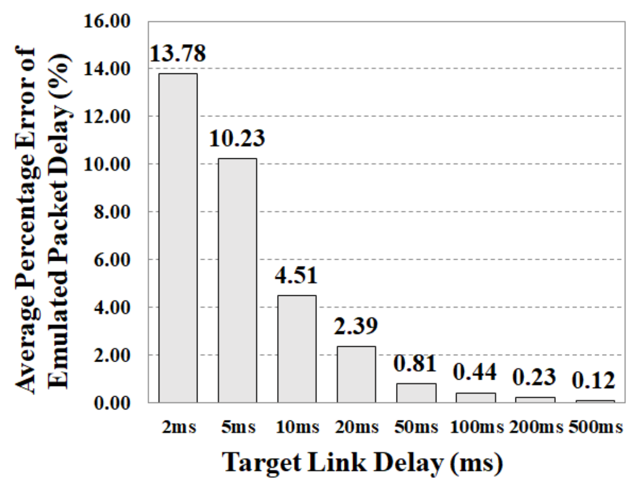

Fig. 5. Average percentage error of emulated packet delay when target link delay is varied

In overall, the obtained results from Fig. 4 suggest a good capability of measuring the time delay between two virtual host examples with the precision in milliseconds. Within this range of error, our emulation can be used in emulation of PMUs under the most time-stringent requirement of WAN latency class TL3 in Table I. Note also from Fig. 5 that when the target link delay is small, the average percentage error of emulated packet delay is misleadingly amplified. Therefore, in subsequent results, we would focus only on the average error without the renormalization in the percentage scale.

B. Effect of SV transmission rate adjustment

This test has been performed by varying SV transmission rate in the same network settings as displayed in Fig. 3. The rate has been varied from 15 packets per second to 240 packets per second with n = 3600 in every case. Without any usage of TCLink, the average error of emulated packet delay for every target SV transmission rate, with 95% confidential interval, is shown in Fig. 6

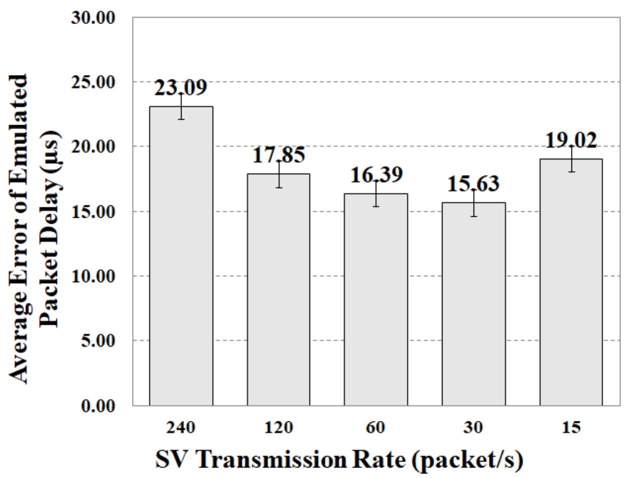

Fig. 6. Average error of emulated packet delay when SV transmission rate is varied

From the same experiment, the average error of SV transmission interval of each transmitted packet, with 95% confidence interval, is depicted in Fig. 7. The error is calculated from the timestamp of each packet as captured by Wireshark at the source host.

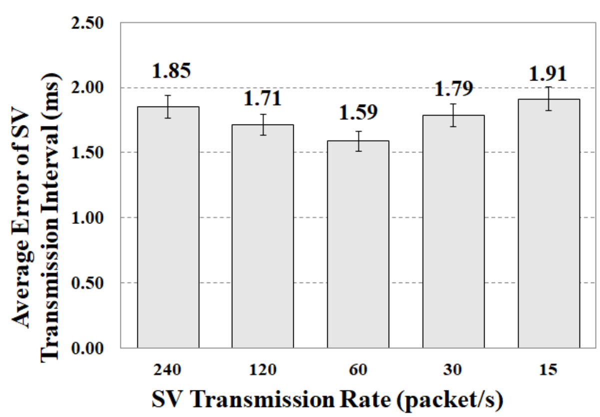

Fig. 7. Average error of SV transmission interval with TCLink set to 0 when SV trasmission rate is varied

Fig. 6 suggests that one can neglect the SV transmission rate on the emulated packet delay since the errors are all insignificant, i.e., within the range of 25 μs. Note also from Fig. 7 that there is a generally average error in the range of 2 ms for SV transmission interval. The problem comes from Python code with the sleep function used pausing the code in between each packet transmission together with CPU speed performance in processing the packet. This behavior would result in packet delay jitter in smart grid applications in the real situation if there is no synchronization via GPS timestamps. However, such an effect is negligible in our testbed since every element in our emulator runs on the same virtual machine where the clock synchronization is always guaranteed.

C. *Effect of route traversing multiple switches and links*

We have carried out our test to the scenario where routes traverse multiple switches and links with the same source and destination hosts (host 3 to host 4). In this test, the traffic flows are shown in Fig 8, with a transmission rate of 120 packets per second and n = 3600. We consider four routes respectively with 2-5 links configured by TCLinks. TCLink delay adjustment is here varied by 5 ms, 10 ms, 50 ms, and 100 ms for each link.

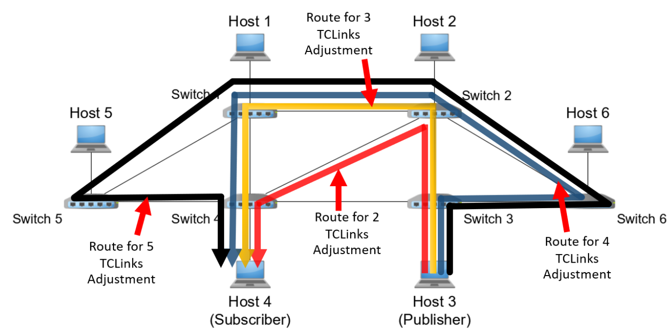

Fig. 8. Network topology settings in test case of routing with multiple switches and links

The result in Fig. 9 shows increasing errors of emulated packet delay proportional to the number oflinks and switches on the route. This error increasing trend is understandable as it is the combined error from each TCLink adjustment and packet switch characteristic in each switch that becomes cumulative along the route. This effect of TCLink adjustment in multi-switch and multi-link scenarios will limit the performance of our emulation in the bigger testbed with more virtual switches and more links. In such cases, care must be exercised before applying the emulator testbed to mimic a network of large size and future investigations along that direction would, therefore, become essential.

IV. Conclusion

In this paper, we provide the validation of our developed network emulation tool, based on Mininet and ONOS, to evaluate the usage potential as a one-way delay estimation tool for planning of wide-area-network infrastructure in the standard-based smart grid environment. The emulation simulates the traffic according to IEC 61850 SV standards, with the standard header and frame structure without the need to consider actual payload of power system parameters. From our validation testing, the average error of emulated packet delay from the source host to the destination host in our emulation, compared to expected settings, are lower than 2 ms for the cases of one TCLink per route. This precision level is considered acceptable for the packet delay evaluation measurement in smart grid applications. It is also found that the errors increase

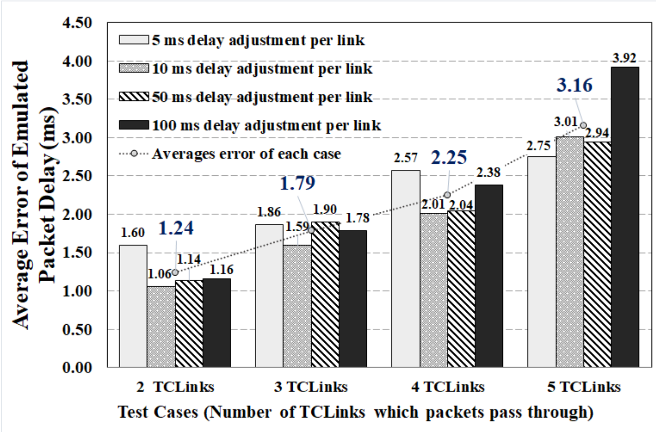

Fig. 9. Average error of emulated packet delay in routes traversing varied number of switches and links

with the number of TCLinks per route and care must be excercised to mimic network of larger sizes. The averages error of SV transmission interval shown in this paper also validate the sufficiently percise imitation of PMUs transmission rate in the network emulation. With these positively potential results, our ongoing work focuses on integrating *Mininet* and *ONOS controller* to emulate and verify the communication time delay of WAMPAC applications with a region-wide area coverage under the circumstance of network configuration and topologies.

Acknowledgment

The first authors would like to thank Electricity Generating Authority of Thailand (EGAT) for the support in the Master-degree scholarship. This research is supported by the Data-centric IoT-cloud service platform for smart communities (IoTcloudServe@TEIN) project under the WP4 Future Internet of Asi@Connect. The authors would like to thank Prof Jong-Won Kim, Gwangju Institute of Science and Technology, South Korea, for helpful suggestions throughout the course of IoTcloudServe@TEIN project.

References

[1] *IEC standard for Communication networks and systems for power utility automation - Part 1: Introduction and overview,* IEC/TR 61850-1, 2013.

[2] *IEC standard for Communication networks and systems for power utility automation – Part 90-12: Wide area network engineering guidelines,* IEC/TR 61850-90-12, July. 2015.

[3] C. M. Adrah, Ø. Kure, J. Yellajosula, S. Paudyal, and B. Mork, "A methodology to implement and investigate performance of sampled values for Wide-Area Protection," *2018 2nd International Conference on Smart Grid and Smart Cities (ICSGSC),* Kuala Lumpur, Malaysia, 2018, pp. 84-90.

[4] C. F. M. Danielson, L. Vanfretti, M. S. Almas and Y., J. O. Gjerde, "Analysis of Communication Network Challenges for Synchrophasor-Based Wide-Area Applications," *2013 IREP Symposium-Bulk Power System Dynamics and Control -IX (IREP),* Rethymnon, Greece, 2013, pp. 1-13.

[5] ONOS. Open Network Operating System. [Online]. Available: https://onosproject.org/. Accessed on: May 10, 2019.

[6] OpenFlow archives - Open Networking Foundation. (2019). [Online] Available: https://www.opennetworking.org/tag/openflow/. Accessed on: May 10, 2019.

[7] N. Dorsch, F. Kurtz, H. Georg, C. Hägerling, and C. Wietfeld, "Software-defined networking for smart grid communications: Applications, challenges and advantages," in *proc. IEEE Int. Contf. Smart Grid Commun. (SmartGridComm),* Venice, Italy, Nov. 2014, pp. 422-427.

[8] J. Konka, C. Arthur, F. Garciad, and R. Atkinson, "Traffic generation of IEC 61850 sampled values," in *Smart Grid Modeling and Simulation(SGMS), 2011 IEEE First International Workshop on,* Oct. 2011, pp. 43-48.

[9] Scapy. Packet crafting for Python2 and Python3 [Online]. Available: https://scapy.net/. Accessed on: May 10, 2019.

[10] Mininet. An instant virtual network on your Laptop (or other PC) [Online]. Available: http://mininet.org/. Accessed on: May 10, 2019.