Dynamic Data-Driven Self-healing Application for Phasor Measurement Unit Networks

Yanfeng Qu(✉), Xin Liu, Jiaqi Yan, and Dong Jin

Illinois Institute of Technology, Chicago, IL 60616, USA
{yqu9,xliu125,jyan31}@hawk.iit.edu, dong.jin@iit.edu

Abstract. This paper describes an approach to apply the dynamic data-driven applications systems (DDDAS) paradigm to enhance cyber security and resilience of wide-area monitoring systems in electrical grids. In particular, we explore a DDDAS-aware application to self-heal phasor measurement unit (PMU) networks that monitor the states of power systems in real-time. The application is built on top of a novel software-defined networking (SDN) architecture. The main components include a dynamic data-driven model that efficiently abstracts the PMU network behavior at run time and an optimization-based solution to quickly reconfigure network connections to restore the power system observability. The application also compresses network updates of the recovery plan to further reduce the recovery time. We develop a prototype system in a container-based network testbed and evaluate the recovery time of the self-healing application using the IEEE 30-bus system.

Keywords: Dynamic data driven application systems · Software-defined networking · Phasor measurement unit · Smart grid resilience and security

1 Introduction

Phase measurement units (PMU) have been increasingly and rapidly deployed in the wide-area monitoring systems to capture the states of electric grids in real-time. PMUs are time-synchronized by GPS timestamps and measure power system states, such as magnitudes and phase angles of current and voltage at each bus, at rates between 30 and 240 Hz. The measurements are then aggregated at phasor data concentrators (PDC) and eventually transmitted to the control center to support state estimation and other critical control and analytic applications. Recent studies reveal that PMU networks are vulnerable to different types of cyber-attacks [1,2], which negatively impact the visualization and situational awareness of power systems.

To address this challenge, we develop a self-healing PMU network scheme with the objective of preventing the propagation of the attacks and maintaining

F. Darema et al. (Eds.): DDDAS 2020, LNCS 12312, pp. 85–92, 2020.
https://doi.org/10.1007/978-3-030-61725-7_12

the complete observability of the power system. We take a DDDAS-based approach to design the self-healing scheme. DDDAS stands for dynamic data-driven applications systems, which is a paradigm that involves dynamically incorporating real-time data into computations in order to steer the measurement and control process of an application system [3]. The DDDAS concept has been successfully applied to many emerging application areas over decades, such as smart cities, manufacturing, transportation, health care, critical infrastructures, and many others [4,5].

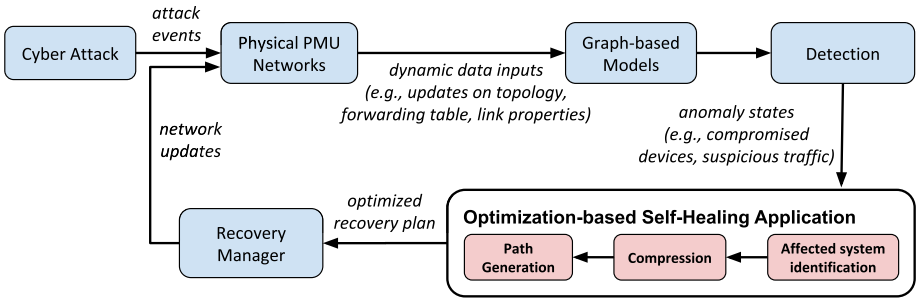

Fig. 1. DDDAS-aware PMU network self-healing application design

Figure 1 depicts the design of our DDDAS-aware self-healing PMU network. The communication network is represented as a dynamic data-driven model that efficiently abstracts the physical PMU network behavior (e.g., packet forwarding) under a dynamic system environment (e.g., network updates caused by cyber-attacks, recovery plans, and other operations). The graph-based model is capable of accepting real-time data at execution time as system states evolve. When the model enters into an abnormal state (e.g., dropped or suspicious traffic from compromised devices), the self-healing scheme is triggered to isolate the traffic from those PMUs and PDCs. The scheme consists of three steps. First, it identifies the portion of the network affected by the cyber incident, such as the list of PMUs to reconnect; Second, it solves an optimization problem to compute the destination PDC for each PMU in the list as well as the immediate switches by meeting the specified device and network operation constraints. Third, it generates an optimal recovery plan to restore power system observability and translate them into network updates for each affected switch. As a result, the scheme steers the control and measurement process by installing network updates on the physical network to self-heal the PMU systems. The updated measurement data and control events are then fed into the graph-based model for further processing. An effective feedback loop is thus enabled to steer the entire self-healing process.

One key component to support this DDDAS-based PMU network self-healing application is the underlying software-defined networking (SDN) based communication infrastructure. SDN is a programmable open-source approach to design-

ing, building, and managing communication networks [6]. SDN decouples the network control from the forwarding functions in network devices and offloads its decision functions to a logically centralized SDN controller. With the increasing size and complexity of the communication networks for wide-area control and monitoring systems, SDN has been increasingly investigated to improve their resilience and security [7–9]. The SDN controller provides the global network visibility that enables us to develop the optimization-based scheme to self-heal the PMU network connection against cyber-attacks. The communication network is composed of a set of SDN switches that enable a quick execution of the recovery plan through SDN's direct network programmability. Moreover, our scheme also applies a rule compression mechanism that compresses the SDN network updates of the recovery plan to further reduce the recovery time. Finally, we develop a proof-of-concept system in a container-based SDN emulation testbed and conduct performance evaluation using the IEEE 30-bus system. The PMU network connection is successfully recovered even when half of the PDCs are compromised, and the recovery time including the plan generation and network updates installation is all within 850 μs.

The remainder of the paper is organized as follows. Section 2 introduces an SDN-based architecture design that enables fast self-healing of PMU networks. Section 3 describes the DDDAS-aware self-healing application including the system model, SDN rule compression method, and optimization model formulation. Section 4 presents the experimental results for performance evaluation. Section 5 concludes the paper with future works.

2 SDN-Based PMU Network Architecture

We present an SDN-based network architecture to automatically self-heal PMU connections and preserve power system observability. This is useful to handle the growing cyber-attacks in wide-area monitoring and control systems that comprise PMU/PDC devices to drop and manipulate measurement data and control messages. Figure 2 depicts the architecture design that consists of five layers. The PMUs measure the states of the underlying power system and the measurements are aggregated at PDCs through the communication network layer, which is composed of a set of SDN-enabled switches to enable direct network programmability. The novelty of the design is mainly at the control layer, in which we integrate an SDN controller to the existing power grid controller. As a result, we now have global visibility and centralized control over the underlying communication network including the end-hosts (i.e., PMU and PDC) and the networking devices (e.g., switches, routers, gateways, and other middle boxes). Within the SDN controller, we develop an optimization-based self-healing scheme to reconfigure the PMU network against compromised or faulty devices. Upon detection of compromised devices, the scheme quickly generates a recovery plan that contains optimal communication path updates to reconnected lost PMUs to PDCs. The scheme also employs a compression module to reduce the number of SDN rules to be installed to further reduce the recovery time.

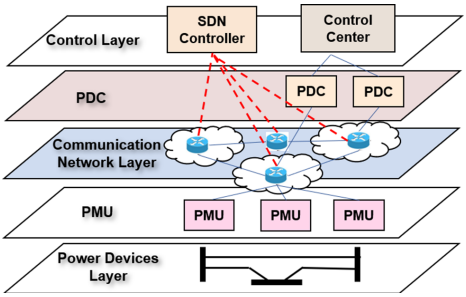

Fig. 2. An SDN-based self-healing PMU network architecture

3 System Modeling and Problem Formulation

3.1 System Model and Power System Observability

The power transmission network is represented by a graph $G_T = <B \cup U, T_U>$, where B is the set of buses, U is the set of PMUs, and T_U is a $|B| \times |U|$ connectivity matrix.

$$t_U[i,j] = \begin{cases} 1, & b_i \; u_j \text{ are connected} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The communication network is represented by another graph $G_C = <U \cup D \cup S, L>$, where each PMU connects to a bus; D is the set of PDCs; and S is the set of SDN switches. L is a connectivity matrix merged via common columns from a $|U| \times |S|$ matrix, a $|S| \times |S|$, and a $|D| \times |S|$ matrix.

$$l[i,j] = \begin{cases} 1, & (u_i \text{ and } s_j) \text{ or } (s_i \text{ and } s_j) \text{ or } (d_i \text{ and } s_j) \text{ are connected} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

We represent the recovery plan as the following binary variable matrix X

$$x_{ij} = \begin{cases} 1, & u_i \text{ connects to } d_j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

A bus is observable if it can be measured by a PMU or estimated by the PMU located on an adjacent bus. Also, measurement data by the PMU has to be reported to a PDC. The power system is observable if all buses are observable. For each bus i, let $A(i)$ denote a set of its adjacent buses and the bus i itself. We define the power system observability as follows.

$$O = \wedge_{\forall i \in B, \forall j \in A(i)}((\vee_{\forall k \in U} t_U[j,k]) \wedge (\vee_{\forall l \in D} x_{k,l})) \tag{4}$$

3.2 Optimization Model and Formulation

We assume the power system is observable before a cyber attack. The attack event compromises a set of PDCs, $D_c \subseteq D$, and triggers the detection system. We then further identify a set of disconnected PMUs, $U_d \subseteq U$, which reduces the power system observability. Observability redundancy exists in the power system because a bus may be monitored by multiple PMUs or estimated through measurements from other related PMUs. Therefore, reconnecting a subset of PMUs in U_d can restore the complete observability. The self-healing scheme computes a recovery plan in the form of a set of updated communication paths $p = \{p_1, p_2, ..., p_n\}$, where $p_1 \in U_d$ and $p_n \in D \setminus D_c$, and each tuple $(p_i, p_{i+1}) \in L$ is a communication link segment. The SDN controller can directly program the switches and install updated rules to realize these paths. Certain paths in the recovery plan may involve a common switch, and it is likely that those paths re-routes different PMUs to the same destination PDC. Hence, we consider using wildcards in the source field of the corresponding SDN rules to further reduce the number of network updates.

We expand $x_{i,j}$ to a new binary decision variable $y_{s,i,j,k}$ defined as follows:

$$y_{s,i,j,k} = \begin{cases} 1, & u_i \text{ reconnects to } d_j \text{ through port } k \text{ of switch } s \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where s is the switch, i is the source PMU, j is the destination PDC, and k is the switch out-port. For simplicity, we assume that every switch has the same number of out-ports. Based on switch s and port k, we define a function, $n(s, k)$, to map the next hop of $p(i)$ in communication path p is $p(i + 1)$.

We assume the SDN controller can install rules on switches in parallel. Let the auxiliary variable Z indicate the maximum number of rules to install on each switch. The objective is to minimize Z with the following constraints.

$$\begin{aligned} &\min : Z \\ s.t.\ \ &\forall s \in S : Z \geq \sum_{j \in D \setminus D_c} \sum_{k} \cup_i y_{s,i,j,k} \end{aligned} \tag{6}$$

Constraint of Power System Observability. Assume that each bus is attached to one PMU, we revise Eq. 4 and obtain the following constraint.

$$\forall i \in U_d : \sum_{i \in N(i)} \sum_{j \in D \setminus D_c} \vee_s \vee_k y_{s,i,j,k} \geq 1 \tag{7}$$

where $N(i)$ denotes a set of PMUs including PMU i and all its neighboring PMUs.

Constraint of Switch Forwarding. For each switch s, it takes at most one port to forward the measurement data from PMU i. Note that each PMU can connect up to one PDC.

$$\forall s \in S, \forall i \in U_d, : \sum_{j \in D \setminus D_c} \sum_{k} y_{s,i,j,k} \leq 1 \tag{8}$$

Constraints of Communication Path. Assume that a source PMU i connects to switch $\alpha(i)$ and its destination PDC j connects to switch $\alpha(j)$. For switch $\alpha(i)$, the difference in the number of output flows and input flows is

$$\forall i \in U_d: \quad \sum_{k} \sum_{j \in D \setminus D_c} y_{\alpha(i),i,j,k} - \sum_{u} \sum_{v} \sum_{n(s,k)=\alpha(i)} y_{s,u,v,k} = \sum_{j \in D \setminus D_c} \vee_s \vee_k y_{s,i,j,k} \tag{9}$$

For switch $\alpha(j)$, the difference in the number of output flows and input flows is

$$\forall j \in D \setminus D_c: \quad \sum_{u} \sum_{v} \sum_{n(s,k)=\alpha(j)} y_{s,u,v,k} - \sum_{i} \sum_{k} y_{\alpha(j),i,j,k} = \sum_{i} \vee_s \vee_k y_{s,i,j,k} \tag{10}$$

For all other switches in the communication path, the number of output flows is equal to the number of input flows.

$$\begin{aligned} &\forall p \notin \{\alpha(i)\} \wedge p \notin \{\alpha(j)\}, \forall i \in U_d, \forall j \in D \setminus D_c: \\ &\sum_{i} \sum_{j} \sum_{k} y_{p,i,j,k} - \sum_{i} \sum_{j} \sum_{n(s,k)=p} y_{s,i,j,k} = 0 \end{aligned} \tag{11}$$

4 Evaluation

4.1 Experimental Setup

We develop a prototype system in an SDN emulation testbed, and place our self-healing scheme as an application in the SDN controller. We use the GNU Linear Programming Kit (GLPK) solver for the ILP problem formulated in Sect. 3.2. To conduct evaluation experiments, we generate a PMU network based on the IEEE 30-bus system. We place one PMU on each bus, and then get the neighboring PMU list according to the adjacent matrix of each bus in the transmission system. We now apply the minimum set cover problem to obtain the least number of PMU sets. We also place one PDC in each set and connect the PDC to PMUs through a switch. All the switches are connected using a ring topology. The original power transmission system is shown in Fig. 3(a), and the constructed PMU network is shown in Fig. 3(b), which is composed of 30 PMUs, 10 PDCs, and 10 switches.

4.2 Performance Evaluation of PMU Network Self-healing Scheme

Model Computational Time is the time spent on the optimization model execution to produce the recovery plan of reconnecting the necessary PMUs to restore power system observability. We vary the number of compromised PDCs from 1 to 5, and run 30 experiments for each case. The means and standard deviations are plotted in Fig. 4(a). We can observe that the PMU network is successfully recovered for all the experiments, even when for the cases when half

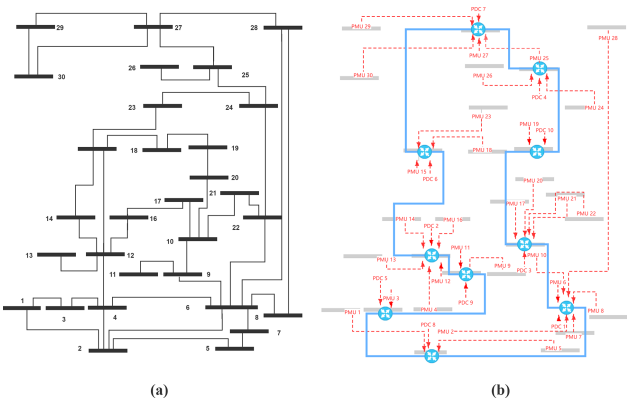

Fig. 3. PMU network construction from the original transmission network (a) IEEE 30-bus system, to (b) PMU network

of the PDC (i.e., 5 out of 10) are compromised. The average computational time is fast, from 265.6 ms to 643.4 ms, with the standard deviation around 20%. With the growing number of compromised PDCs, the computational time increases at first because the generated recovery paths become more complex. However, when the computational time does not keep increasing as more compromised PDCs do not always result in more PMUs to recover.

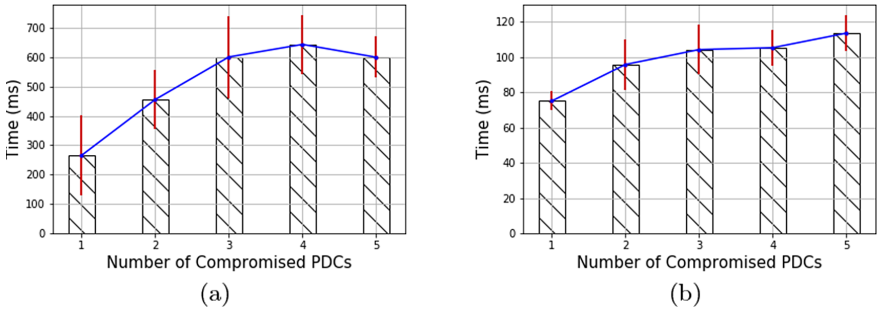

Fig. 4. Recovery plan: (a) computational time, and (b) installation time

Rule Installation Time is the time spent on realizing the recovery plan in the PMU network, including the rule generation at the SDN controller, the rule transmission from the controller to the switches, and the actual rule installation on the switches. We again vary the number of compromised PDCs from 1 to 5, and run 30 experiments for each case. The results are plotted in Fig. 4(b). We observe that it takes 75.1 ms to 113.4 ms on average to install the recovery plan. The standard deviation is within 10%. The installation time increases as

the number of compromised PDCs grows. Compared with the computational time, the installation time is much faster in general. The total time to generate and install the recovery plan is quick in general as all the experiments complete within 850 μs.

5 Conclusion and Future Works

We apply the DDDAS paradigm to protect PMU networks and restore the power system observability. Our DDDAS-aware network self-healing application considers both the power system and communication network characteristics with the help of an SDN-based cyber-infrastructure. The current version focuses on PMUs in the power transmission systems and we will extend it to micro PMUs on the distribution systems and microgrids.

Acknowledgment. This work is partly sponsored by the Air Force Office of Scientific Research (AFOSR) under Grant YIP FA9550-17-1-0240, the National Science Foundation (NSF) under Grant CNS-1618631, and the Maryland Procurement Office under Contract No. H98230-18-D-0007.

References

1. Khan, R., Maynard, P., Mclaughlin, K., Laverty, D., Sezer, S.: Threat analysis of blackenergy malware for synchrophasor based real-time control and monitoring in smart grid. In: Proceedings of the 4th International Symposium for ICS and SCADA Cyber Security Research (2016)
2. Liu, X., Li, Z.: False data attacks against AC state estimation with incomplete network information. IEEE Trans. Smart Grid **8**(5), 2239–2248 (2017)
3. Darema, F.: Dynamic data driven applications systems: a new paradigm for application simulations and measurements. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2004. LNCS, vol. 3038, pp. 662–669. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24688-6_86
4. Blasch, E., Ravela, S., Aved, A. (eds.): Handbook of Dynamic Data Driven Applications Systems. Springer, Cham (2018)
5. Fujimoto, R., et al.: Dynamic data driven application systems: research challenges and opportunities. In: Proceedings of the 2018 Winter Simulation Conference (WSC), pp. 664–678 (2018)
6. McKeown, N., et al.: Openflow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. **38**(2), 69–74 (2008)
7. Lin, H., et al.: Self-healing attack-resilient PMU network for power system operation. IEEE Trans. Smart Grid **9**(3), 1551–1565 (2018)
8. Sarailoo, M., Wu, N.E.: An algorithm for resilient sensor network upgrade with fewest PMUs. In: Proceedings of the 2017 Resilience Week (RWS), pp. 77–82 (2017)
9. Qu, Y., Liu, X., Jin, D., Hong, Y., Chen, C.: Enabling a resilient and self-healing PMU infrastructure using centralized network control. In: Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, pp. 13–18 (2018)