Enabling a Resilient and Self-healing PMU Infrastructure Using Centralized Network Control

Yanfeng Qu, Xin Liu
Illinois Institute of Technology
{yqu9,xliu125}@hawk.iit.edu

Dong Jin, Yuan Hong
Illinois Institute of Technology
{dong.jin,yuan.hong}@iit.edu

Chen Chen
Argonne National Laboratory
morningchen@anl.gov

ABSTRACT

Many of the emerging wide-area monitoring protection and control (WAMPAC) applications in modern electrical grids rely heavily on the availability and integrity of widespread phasor measurement unit (PMU) data. Therefore, it is critical to protect PMU networks against growing cyber-attacks and system faults. In this paper, we present a self-healing PMU network design that considers both power system observability and communication network characteristics. Our design utilizes centralized network control, such as the emerging software-defined networking (SDN) technology, to design resilient network self-healing algorithms against cyber-attacks. Upon detection of a cyber-attack, the PMU network can reconfigure itself to isolate compromised devices and re-route measurement data with the goal of preserving the power system observability. We have developed a proof-of-concept system in a container-based network testbed using integer linear programming to solve a graph-based PMU system model. We also evaluate the system performance regarding the self-healing plan generation and installation using the IEEE 30-bus system.

CCS CONCEPTS

• **Networks → Network architectures**; **Network reliability**; • **Hardware → Smart grid**;

KEYWORDS

Software-Defined Networking (SDN); Phasor Measurement Unit (PMU); Cyber Resilience and Security; Power System Observability

ACM Reference Format:
Yanfeng Qu, Xin Liu, Dong Jin, Yuan Hong, and Chen Chen. 2018. Enabling a Resilient and Self-healing PMU Infrastructure Using Centralized Network Control. In *SDN-NFV Sec'18: 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, March 19–21, 2018, Tempe, AZ, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3180465.3180472

1 INTRODUCTION

Smart grid technologies are transforming today's power grid into a more efficient, autonomous and self-healing system with increasing computer intelligence. A key capability of the smart grid is the advanced measurement technology for power-system analysis and control. Phasor measurement unit (PMU) networks are being rapidly deployed in the wide-area monitoring protection and control (WAMPAC) systems to measure the state of the electrical system and manage power quality, from around 200 R&D grade PMU systems in 2007 to 2500+ networked PMU systems in North America in 2017 [1]. However, recent studies reveal that PMU networks can suffer different types of cyber-attacks [5, 6, 9]. Consequently, the attacks can significantly reduce the system observability and thus affect state estimation and other critical power system applications and operations.

Researchers have studied the cyber-security issues of PMU networks and proposed methods to protect them against malicious attacks and system faults [2, 5–7]. Our work aims to make the PMU network more resilient to growing cyber-attacks and system faults as the system gets more complex and interconnected. We argue that an efficient self-healing scheme should be a cross-layer solution by considering critical constraints in the power system application layer as well as the underlying communication network layer. For example, (1) we should focus on recovering power system observability rather than maximizing the PMU device connectivity described in the existing works; (2) we ought to update the communication paths while preserving the important requirements such as congestion-freedom and real-time operations rather than always taking the shortest paths; and (3) we ought to consider the performance overhead of network reconfiguration since PMU networks have more demanding requirements on availability. This motivates us to investigate a cross-layer self-healing scheme exclusive to PMU networks with the goal of efficiently recovering the power system observability while minimizing the self-healing time. Our scheme has three features. First, the scheme has a global view of the PMU network for computing a global-optimal solution. Second, the scheme incorporates essential constraints from the electrical grid applications and the communication network. Third, the scheme enables the direct network control to achieve flexible and fast network reconfiguration.

In this paper, we present a solution to self-heal PMU networks with centralized network control. The self-healing process has two stages. The first stage is to recover the power system observability by identifying a subset of disconnect PMUs and to which working phasor data concentrators (PDCs) they should reconnect. The second stage is to compute the full communication paths including all the intermediate network devices for each PMU-PDC reconnection generated by the first stage, while satisfying specific objectives, such as fastest recovery time, congestion-freedom, or least device reconfiguration. A vital component of the scheme is the centralized network controller. Software-defined networking (SDN) technology improves network manageability with direct and centralized control via a well-defined and open application programming interface, and thus well matches the design requirements. Our prior

SDN-NFV Sec'18, March 19–21, 2018, Tempe, AZ, USA

ACM ISBN 978-1-4503-5635-0/18/03…$15.00
https://doi.org/10.1145/3180465.3180472

work studied the feasibility of the SDN-based approach to self-heal PMU networks [4], which motivates us to further explore the SDN-based solutions with new system constraints and prototype system development. On the other hand, SDN-based solutions also face problems, such as heavy modifications to the current PMU network architecture (e.g., equipment upgrades and new management system installation) as well as intensive training for the current human operators (e.g., how to manage and debug an SDN network). Fibbing [8] was proposed to combine the advantages of SDN and traditional approaches. Fibbing essentially is a network architecture that enables central control over distributed routing by intelligently forcing routers to compute their own forwarding tables based on an augmented topology generated via virtual nodes and links. In this work, we also study the feasibility of Fibbing control within our self-healing scheme by developing the associated optimization model in the prototype system with intensive performance evaluation.

The main contributions of this paper are summarized as follows. First, we present a novel self-healing PMU network architecture based on centralized network control. Second, by considering the specific constraints across the power system applications as well as the underlying communication network, we formulate the self-healing process using an integer linear programming (ILP) model over graph-based networked system models. Third, we develop a prototype system using a container-based network testbed, Mininet [3] and conduct performance evaluation concerning self-healing plan generation and network reconfiguration with both SDN and Fibbing controllers.

The remainder of the paper is organized as follows. Section 2 overviews the self-healing PMU network architecture design. Section 3 describes the optimization model and formulation using centralized network control. Section 4 presents the performance evaluation results. Section 5 concludes the paper with future works.

2 SELF-HEALING PMU NETWORK ARCHITECTURE

We present a self-healing PMU system over a centralized network control infrastructure. Figure 1 depicts the four-layer architecture design. The control layer integrates the logically centralized network controllers (i.e., SDN and Fibbing) to the existing control center facility. The centralized network visibility enables us to explore optimization-based algorithms to optimally reconfigure the PMU network against compromised or faulty devices. Upon detection of compromised PDCs, our algorithm produces the recovery plan to restore the system observability in two stages. Stage one aims to recover the power system observability by pairing working PDCs and disconnected PMUs. Stage two aims to generate paths for the new PMU-PDC communication with the specific objectives and constraints, such as fastest recovery time, congestion-freedom, or least device reconfiguration. The communication network layer is composed of a set of conventional routers that run link-state routing protocols (e.g., OSPF) to calculate shortest paths and detect topology changes (e.g., link failures) as well as SDN-enabled switches that enable direct network programmability. Measurements of the underlying electrical grid layer are captured by PMUs in the device layer, and then transferred, quality-checked, and aggregated at PDCs. The synchrophasor data are eventually collected at the control center for power system state estimation and other critical power system applications.

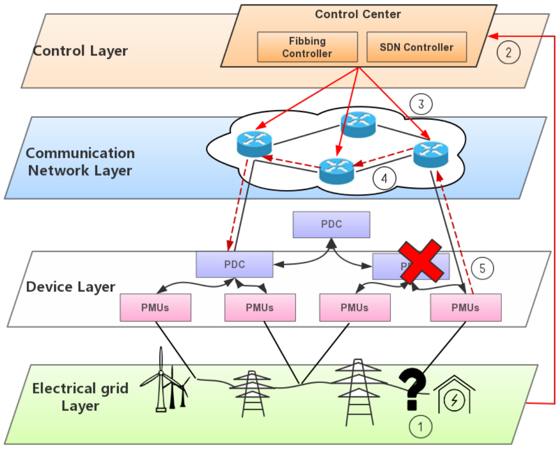

Figure 1: A Self-healing PMU Network Architecture

Figure 1 also illustrates the self-healing process step by step. (1) Assume a cyber-attack renders certain PDCs compromised or disconnected. As a result, we lose the measurement data from the associated PMUs that negatively affects the system observability. (2) Upon detection of the problematic devices, the control center computes the recovery plan using our two-stage optimization algorithm. (3) The network controller (SDN or Fibbing) generates a set of updates for the network devices base on the recovery plan (e.g., OpenFlow rules for SDN and virtual nodes for Fibbing). (4) SDN switches install new rules, and traditional routers update their routing tables to realize the recovery plan. (5) A subset of the disconnected PMUs are now reconnected to the network to report their measurements and thus restore the power system observability.

3 SELF-HEALING MECHANISM BASED ON CENTRALIZED NETWORK CONTROL

3.1 Power System Observability

A PMU measures the electrical waves of the bus i at which it is placed and all its adjacent buses. The observability function of bus i is defined as

$$O_i = \Sigma\, a_{i,j} p_j$$

where $a_{i,j}$ defines the bus connectivity.

$$a_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 1 & \text{if } i \neq j \text{ and bus } i \text{ and bus } j \text{ are connected} \\ 0 & \text{otherwise} \end{cases}$$

$p_j = 1$ if a PMU is placed at bus j or $p_j = 0$ otherwise. $O_i \geq 1$ implies that bus i is observable. The entire power system is observable if every bus is observable. In this paper, we use the observability definition above to formulate the self-healing optimization model based on centralized network control. An extended definition of the observability considering zero injection buses is available in our prior work [4].

3.2 Recovery Time Analysis

A key objective is to minimize the PMU network recovery time upon detection of cyber-attacks. The recovery time consists of two main components, the self-healing plan generation time and the self-healing plan installation time. While how to detect the cyber-incidents are essential, it is not the focus of this paper. In the self-healing plan generation stage, the goal is for the controller to quickly restore the power system observability defined in 3.1 while considering other system constraints, such as communication network congestion freedom, PDC-PMU connection limit, and network device capacity. The output is a set of new or updated communication paths to reconnect a set of chosen PMUs (origins) and PDCs (destinations). We call each (PMU, PDC) selection an origin-destination pair (OD-pair) in the remainder of the paper. In the self-healing plan installation stage, the goal is to quickly realize the plan in the network. For SDN-based networks, the network updates (e.g., OpenFlow rules) can be installed on the devices in parallel once the controller issues the updates. Given a set of end-to-end routes, the total installation time is bounded by the maximum number of updates to be installed on each network switch. For networks using the Fibbing controller, we concern about the configuration time needed for the non-shortest paths including the time to establish the associated virtual nodes. Fibbing relies on the OSPF routing protocol to handle the shortest paths and thus no extra configuration time is required. For each given communication path, the installation time is bounded by the number of virtual links injected from the Fibbing controller to the network. Therefore, a reasonable objective is to minimize the number of non-shortest paths in order to save the recovery time.

3.3 Optimization Model and Formulation

The objective is to minimize the recovery time while preserving the power system observability and satisfying various communication network constraints. We first model the power transmission network and the communication network, and then formulate the self-healing problem using Integer Linear Programming (ILP), and implement the solution using GNU Linear Programming Kit (GLPK) library. To better illustrate the system model and the optimization formulation, we list key notations in Table 1.

3.3.1 System Model and Constraints. The power transmission network is represented by a graph $G_p =< B, L_p >$ and the communication network is represented by a graph $G_c =< U \cup D \cup R, L_c >$. Assume that the detection system can produce a set of compromised PDCs, $D_c \subseteq D$, upon the occurrence of cyber-attacks. We then identify a set of disconnected PMUs, $U_d \subseteq U$. Given G_p, G_c, D_c, and U_d as the inputs, the self-healing solver computes a recovery plan in the form of a set of updated communication paths $p = \{x_1, x_2, ..., x_n\}$, where $x_1 \in U_d$ and $x_n \in D \setminus D_c$, and each tuple $(x_i, x_{i+1}) \in L_c$ is a communication link. These paths should satisfy the following constraints:

- Observability: the connected PMUs should make all buses observable.
- PDC connection capacity: each PDC should not exceed the maximum number of allowed PMU connection.

Notation	Description
B	Set of buses in the power transmission network
L_p	Set of transmission lines between buses
U	Set of PMUs in the network, each connecting to a bus
D	Set of PDCs in the network
R	Set of communication network forwarding devices
L_c	Set of communication links between end-hosts (i.e., PMU, PDC) and network forwarding devices
$N : U \mapsto \mathcal{P}(U)$	Function that maps a PMU to its neighboring PMUs
$C : D \mapsto \mathbb{Z}^+$	Function that maps a PDC to its PMU connection capacity
$W : L_c \mapsto \mathbb{R}^+$	Function that maps a communication link to its bandwidth
$T : U \mapsto \mathbb{R}^+$	Function that maps a PMU to its traffic demand
$F : R \mapsto \mathbb{Z}^+$	Function that maps a switch/router to its rule installation capacity
$D_c \in D$	Set of compromised/faulty PDCs
$U_d \in U$	Set of disconnected PMUs
$P = \{x_1, x_2, ..., x_n\}$	A full communication path connecting a PMU to a PDC
S_{OD}	Set $\{p = (u, d) \| u \in U_d, d \in (D \setminus D_c)\}$, the two end points of a communication path (i.e., original and destination nodes), the decision variables of stage 1
S_{EP}	Set $\{(e, p) \| e = (s, t) \in L_c, p \in \{(u, d) \| (u, d) = 1\}\}$, the communication path for each S_{OD}, the decision variables of stage 2
Y	Set $\{y_{(u,d)} \| (u, d) \in S_{OD}\}$, an auxiliary decision variable indicating whether the chosen path for (u, d) is a shortest path
$SL : (u, d) \mapsto \mathbb{Z}^+$	Function that maps an S_{OD} to the length of its shortest path
$M = \|L_c\|$	Total number of communication links
Z	Auxiliary variable indicating the maximum number of rules can be installed on an OpenFlow switch
$num_{in} : (p, v) \mapsto \mathbb{Z}^*$	Function that maps a tuple (p, v), where $p \in S_{OD}$ and $v \in (U \cup D \cup R)$, to the indegree of v with respect to p
$num_{out} : (p, v) \mapsto \mathbb{Z}^*$	Function that maps a tuple (p, v), where $p \in S_{OD}$ and $v \in (U \cup D \cup R)$, to the outdegree of v with respect to p

Table 1: Summary of Notations

- Congestion freedom: the traffic load on each link should not exceed the bandwidth.
- Network device capacity: each OpenFlow switch/OSPF router should not install more rules exceeding its space capability.

While satisfying all the constraints mentioned above, we construct the paths with the objective to minimize the recovery time. In SDN-based networks, we should minimize the maximum number of paths through a switch (assuming each new path requires the one-rule-per-switch installation); In OSPF-based networks with the Fibbing controller, we should minimize the number of non-shortest paths.

3.3.2 Optimization Formulation. Since the power transmission network and the communication network both affect the recovery time but in different ways, we decompose the problem into two stages. Stage 1 focuses on the power network layer by recovering the system observability, and stage 2 focuses on the communication network layer by constructing the paths for each O-D pair to minimize the recovery time.

Stage 1. The decision variables are the set of PMU-PDC pairs $S_{OD} = \{p = (u, d) | u \in U_d, d \in (D \setminus D_c)\}$ with binary values.

$$(u,d) = \begin{cases} 1, & \text{if } u \text{ sends synchrophasor data to } d \\ 0, & \text{otherwise} \end{cases}$$

The objective of stage 1 is to minimize the total number of O-D pairs chosen as shown in Equation 1 in order to save the network configuration time in the stage 2.

$$\text{minimize} \quad \textstyle\sum_{\forall (u,d) \in S_{OD}} (u,d) \tag{1}$$

To enforce the entire power system observability, each bus must be observable, which means either the bus or one of its neighbors are chosen to transmit the synchrophasor data as expressed in Equation 2.

$$\textstyle\sum_{\forall u' \in u \cup N(u)} \sum_{\forall d \in (D \setminus D_c)} (u', d) \geq 1, \quad \forall u \in U_d \tag{2}$$

For each PDC, the number of connected PMUs shall not exceed its capacity as shown in Equation 3.

$$\textstyle\sum_{\forall u \in U_d} (u,d) \leq C(d), \quad \forall d \in (D \setminus D_c) \tag{3}$$

One implicit assumption is that each PMU sends data to at most one PDC as shown in Equation 4.

$$\textstyle\sum_{\forall d \in (D \setminus D_c)} (u,d) \leq 1, \quad \forall u \in U_d \tag{4}$$

In summary, the problem stage 1 formulates an ILP problem with the objective to minimize the recovery paths needed and still preserve the observability of the system, while keeping the number of connected PMUs within each PDC's capacity:

$$\min : \sum_{\forall (u,d) \in S_{OD}} (u,d)$$

$$s.t. \begin{cases} \sum_{\forall u' \in u \cup N(u)} \sum_{\forall d \in (D \setminus D_c)} (u', d) \geq 1, \ \forall u \in U_d \\ \sum_{\forall u \in U_d} (u,d) \leq C(d), \ \forall d \in (D \setminus D_c) \\ \sum_{\forall d \in (D \setminus D_c)} (u,d) \leq 1, \ \forall u \in U_d \end{cases} \tag{5}$$

Stage 2. After identifying the set of PMUs and PDCs to establish connections, we then construct their communication paths. The decision variable set is $S_{EP} = \{(e, p) | e \in L_c, p \in \{(u, d) | (u, d) = 1\}\}$ with binary values.

$$(e,p) = \begin{cases} 1, & \text{if edge } e \text{ belongs to the path of } p \\ 0, & \text{otherwise} \end{cases}$$

The paths should contain no loop (i.e., no repeated intermediate nodes). For any given p, each network forwarding device has at most one incoming edge and at most one outgoing edge, and the amount of inbound and outbound traffic should be identical as shown in Equation 6.

$$0 \leq num_{in}(r,p) = num_{out}(r,p) \leq 1, \quad \forall r \in R, \forall p \in S_{OD} \tag{6}$$

The above constraint contains two auxiliary functions, $num_{in}()$ and $num_{out}()$. The function $num_{in}()$ takes an O-D pair and one intermediate forwarding device as inputs, and output the indegree of that device with respect to that O-D pair:

$$num_{in}(p,v) = \sum_{\forall e \in \{(s,t) | t = v\}} (p,e) \tag{7}$$

Similarly, $num_{out}()$ takes the same input and produces the outdegree of that device with respect to that O-D pair:

$$num_{out}(p,v) := \sum_{\forall e \in \{(s,t) | t = v\}} (p,e) \tag{8}$$

For each $p = (u,d) \in S_{OD}$, Equation 9 indicates that the origin (PMU) and the destination (PDC) must be included in the path. Equation 10 indicates that no edge in the path of $p = (u,d)$ should involve end devices (i.e., PMUs and PDCs) other than u and d.

$$\begin{aligned} &num_{out}((u,d),u) = 1, \quad \forall (u,d) \in S_{OD} \\ &num_{in}((u,d),d) = 1, \quad \ \forall (u,d) \in S_{OD} \end{aligned} \tag{9}$$

$$\begin{aligned} &\textstyle\sum_{\forall v \in U} num_{out}(p,v) = 1, \quad \forall p \in S_{OD} \\ &\textstyle\sum_{\forall v \in D} num_{in}(p,v) = 1, \quad \ \forall p \in S_{OD} \end{aligned} \tag{10}$$

We also consider other network-related constraints including congestion-freedom that ensures the total traffic load on each link never exceeds the bandwidth as shown in Equation 11.

$$\textstyle\sum_{\forall (u,d) \in S_{OD}} (e,(u,d)) \times T(u) \leq W(e), \quad \forall e \in L_c \tag{11}$$

At last, the number of rules installed on each OpenFlow switch or OSPF router should not exceed its rule capacity as shown in Equation 12. We assume that constructing a new path in an SDN network requires to install one rule per switch along the path.

$$\textstyle\sum_{\forall p \in S_{OD}} num_{in}(p,v) \leq F(v), \quad \forall v \in R \tag{12}$$

To minimize the recovery time, networks with Fibbing controllers and SDN controllers have different objectives. With Fibbing controllers, the objective is to minimize the number of non-shortest paths as shown in Equation 13. To formulate the ILP objective function, we apply a set of non-negative auxiliary variable $y_{(u,d)}$ for each (u,d) pair whose constraints are also listed as below.

$$\min : \sum_{\forall (u,d) \in S_{OD}} y_{(u,d)}$$

$$s.t. \begin{cases} y_{(u,d)} \geq (\sum_{e \in L_c} (e,(u,d)) - SL(u,d))/M, \ \forall (u,d) \in S_{OD} \\ y_{(u,d)} \in \mathbb{Z}^* \end{cases} \tag{13}$$

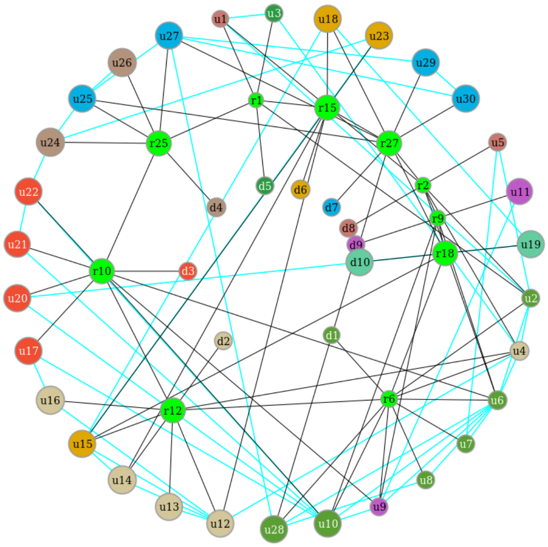

Figure 2: PMU Network Topology

where $SL(u, d)$ stands for the length of shortest path between u and d, and M is the number of communication links in the network (i.e., $M = |L_c|$).

For SDN-based networks, the objective is to minimize the maximum number of rules to be installed on the switches as shown in Equation 14. To formulate the ILP objective function, we apply an auxiliary variable Z whose constraint is also listed as below.

$$\min : Z$$
$$s.t.\ Z \geq \sum_{\forall p \in S_{OD}} num_{in}(p, v),\ \forall v \in R \tag{14}$$

4 EVALUATION

4.1 Optimization Model for Self-Healing Plan Generation

We first evaluate the proposed optimization model in terms of the computational time and the number of cases successfully preserving/restoring the power system observability. We modeled a PMU network installed on the IEEE 30-Bus system as shown in Figure 2. The outer ring is composed of the PMU nodes, and each PMU is attached to a bus. PMUs are connected by blue edges if their attached buses are adjacent. The middle ring is composed of the router nodes, and each router connects to multiple PMU nodes. The inner ring is composed of PDC nodes, and each PDC connects to a router. PMUs send synchrophasor data to a PDC in the same color. We assume that each PDC can connect up to 20 PMUs; each communication link has a bandwidth of 60 Mbps; each PMU offers a traffic load of 10 Mbps; each OpenFlow switch can install up to 100 rules.

There are 10 PDCs in the network, and the number of compromised PDCs ranges from 3 to 8. For each number, we ran 50 experiments for both SDN and Fibbing controller based networks. For each experiment, we randomly chose the locations of the compromised PDCs. The evaluation metrics include

Num_PDC	3	4	5	6	7	8
Num_T	12	6	1	0	0	0
Num_F	33	37	44	47	38	8
Avg_P	2.64	3.05	3.02	4.15	5.50	6.25
Avg_V	0.00	0.00	0.00	0.09	0.26	0.62
Avg_Time (ms)	37.66	40.88	42.77	49.95	56.30	57.76

Table 2: Results of the Fibbing-controller-based Model

Num_PDC	3	4	5	6	7	8
Num_T	14	2	1	0	0	0
Num_F	34	45	43	44	30	10
Avg_P	2.59	2.91	3.21	4.23	5.33	7.20
Avg_M	1.35	1.56	1.63	1.93	2.60	3.70
Avg_Time (ms)	38.95	42.37	42.77	50.03	55.22	58.92

Table 3: Results of SDN-controller-based Model

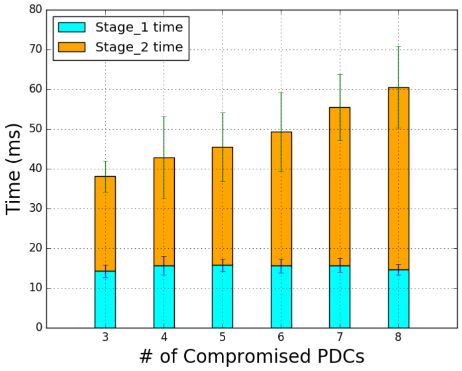

Figure 3: Computational Time: Fibbing Controller

- Num_T: the number of cases that the power system observability is preserved even with disconnected PMUs
- Num_F: the number of cases that the power system observability can be restored using our ILP solver
- Avg_P: the average number of newly generated paths
- Avg_V: the average number of non-shortest paths for the Fibbing controller base model
- Avg_M: the mean of the max number of rules installed on each SDN switch
- Avg_Time: the average computational time of the model in milliseconds

Table 2 and Table 3 show the results for the Fibbing controller and SDN controller based models respectively. The computational time is also decomposed into two stages and plotted in Figure 3 and Figure 4. In both tables, we observe that the number of cases successfully preserving/restoring the power system observability decreases as the number of compromised PDCs grows, because we have to generate more paths to reconnect more PMUs to recover the observability. In addition, the number of non-shortest paths increases for the model with Fibbing controllers, and the maximum number of installed rules per switch increases for the

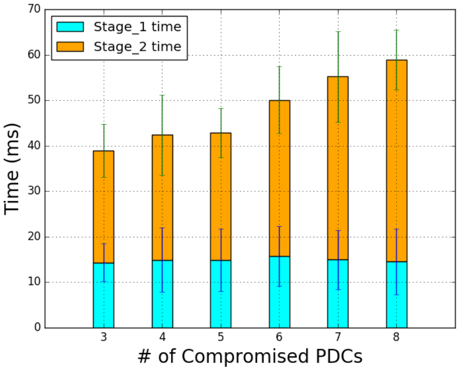

Figure 4: Computational Time: SDN Controller

model with SDN controllers. The small Avg_V values indicate good performance in the Fibbing controller cases. Even with 8 PDCs compromised, the average number of non-shortest paths generated by our model is 0.62 (see Table 2), i.e., Fibbing only needs to construct 0.62 new paths on average. For the SDN cases, with 8 PDCs compromised, the maximum number of rules to be installed per switch is 3.70 in average. In other words, the controller only has to install 3.70 rounds of network updates.

As shown in Figure 3 and Figure 4, we observe that stage 1 takes 12 to 15 milliseconds to complete the work of identifying the PMU-PDC pairs to reconnect, and stage 2 takes 25 to 45 milliseconds to complete the work of constructing the new communication paths. We also observe that the growing number of compromised PDCs does not affect our model to derive the list of end-devices for restoring the system observability. However, it makes the time to compute the paths to connect the end-devices linearly increase because of the increasing value of Avg_P and the enforcement of the congestion-freedom and network device capability constraints.

4.2 Self-Healing Plan Network Installation

We created the same PMU network topology (see Figure 2) in Mininet [3] to evaluate the communication path installation time. The controllers take the set of communication paths generated by the optimization model as inputs and install the updates in the network. For Fibbing controller based networks, we measured the time to complete updating each non-shortest path in the network. Let N be the number of non-shortest paths derived from the optimization model. We varied N from 1 to 5, and ran 50 experiments to collect the path installation time for each value, and plotted the cumulative distribution function (CDF) for each N in Figure 5. We observe that the installation time is less than 900 milliseconds for all the cases. For example, 90% of the paths are updated within 200 milliseconds for a single path update plan. The total self-healing time including the self-healing plan generation (Section 4.1) and the path installation is less than one second for all the experimental runs.

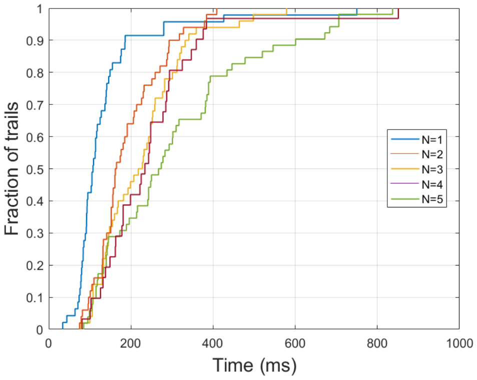

Figure 5: CDFs of Non-shortest Communication Path Installation Time

5 CONCLUSION AND FUTURE WORK

We present a self-healing PMU network design using centralized network control such as SDN and Fibbing controllers. In the future, we will compare the existing two-stage ILP formulation with a single-stage formulation with the goal of quickly achieving the global optimum. We will extend the attack scenarios in which compromised devices are not only the power measuring devices like PMUs and PDCs but also the network devices. In addition, we will study efficient self-healing mechanisms on a hybrid network architecture consisting of both SDN and traditional networking devices and protocols.

ACKNOWLEDGMENTS

The authors are grateful to the support by the Air Force Office of Scientific Research under Grant YIP FA9550-17-1-0240.

REFERENCES

[1] 2017. NARUC Summer Meeting, North American SynchroPhasor Initiative (NASPI). (2017). https://www.naspi.org/sites/default/files/reference_documents/naspi_naruc_silverstein_20170714.pdf

[2] Christopher Beasley, G. Kumar Venayagamoorthy, and Richard Brooks. 2014. Cyber security evaluation of synchrophasors in a power system. In *Proceedings of the 2014 Clemson University Power Systems Conference (PSC)*. 1–5.

[3] Bob Lantz, Brandon Heller, and N McKeown. 2010. A network in a laptop: rapid prototyping for software-defined networks. In *the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets-IX)*. ACM, 1–6.

[4] H. Lin, C. Chen, J. Wang, J. Qi, D. Jin, Z. Kalbarczyk, and R. K. Iyer. 2016. Self-Healing Attack-Resilient PMU Network for Power System Operation. *IEEE Transactions on Smart Grid* PP, 99 (2016), 1–1.

[5] Thomas Morris, Shengyi Pan, Jeremy Lewis, Jonathan Moorhead, Nicholas Younan, Roger King, Mark Freund, and Vahid Madani. 2011. Cybersecurity Risk Testing of Substation Phasor Measurement Units and Phasor Data Concentrators. In *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW '11)*. 24:1–24:4.

[6] S. Mousavian, J. Valenzuela, and J. Wang. 2015. A Probabilistic Risk Mitigation Model for Cyber-Attacks to PMU Networks. *IEEE Transactions on Power Systems* 30, 1 (2015), 156–165.

[7] John Stewart, Thomas Maufer, Rhett Smith, Chris Anderson, and Ersonmez Eren. 2011. Synchrophasor Security Practices. *Schweitzer Engineering Laboratories* (2011), 1–10.

[8] Stefano Vissicchio, Olivier Tilmans, Laurent Vanbever, and Jennifer Rexford. 2015. Central Control Over Distributed Routing. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15)*. 43–56.

[9] Xingsi Zhong, Paranietharan Arunagirinathan, Afshin Ahmadi, Richard Brooks, and Ganesh Kumar Venayagamoorthy. 2015. Side-Channels in Electric Power Synchrophasor Network Data Traffic. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference (CISR '15)*. 3:1–3:8.