Self-Healing Attack-Resilient PMU Network for Power System Operation

Hui Lin, Chen Chen, *Member, IEEE*, Jianhui Wang, *Senior Member, IEEE*, Junjian Qi, *Member, IEEE*, Dong Jin, *Member, IEEE*, Zbigniew T. Kalbarczyk, *Member, IEEE*, and Ravishankar K. Iyer, *Fellow, IEEE*

***Abstract*—In this paper, we propose a self-healing phasor measurement unit (PMU) network that exploits the features of dynamic and programmable configuration in a software-defined networking infrastructure to achieve resiliency against cyber-attacks. After a cyber-attack, the configuration of network switches is changed to isolate the compromised PMUs/phasor data concentrators to prevent further propagation of the attack; meanwhile, the disconnected yet uncompromised PMUs will be reconnected to the network to "self-heal" and thus restore the observability of the power system. Specifically, we formulate an integer linear programming model to minimize the overhead of the self-healing process (e.g., the recovery latency), while considering the constraints of power system observability, hardware resources, and network topology. We also propose a heuristic algorithm to decrease the computational complexity. Case studies of a PMU network based on the IEEE 30-bus and 118-bus systems are used to validate the effectiveness of the self-healing mechanism.**

***Index Terms*—Cybersecurity, phasor data concentrator (PDC), phasor measurement unit (PMU), resilience, self-healing, software-defined networking (SDN), system observability.**

I. Introduction

IN TODAY'S power grid, phasor measurement units (PMUs) are being deployed in the wide-area monitoring systems (WAMSs) to monitor the state of a power system in real time (e.g., static and dynamic state estimation, oscillation detection and control, power line outage detection) [1]–[4]. Based on the NASPInet architecture for a PMU network [5], the measurements collected by multiple PMUs are delivered and combined at a phasor data concentrator (PDC), which further sends the measurements to the next-level PDC or the control center.

Off-the-shelf computing and communication technologies are integrated with the intelligent electronic devices (IEDs), including PMUs and PDCs, to boost monitoring and control efficiency. However, this integration opens up new attack vectors: a PMU or a PDC can become the target of cyber-attacks. Recent studies reveal that PMUs or PDCs can suffer different types of cyber-attacks, including denial-of-service or man-in-the-middle attacks [6], [7]. To make things worse, the network connections make the further propagation of attacks possible [8]. Consequently, upon detection of attacks, compromised PMUs or PDCs can be disconnected from the communication network. Although quarantine of the compromised devices can prevent further propagation of the attacks, it can significantly reduce the system's observability (i.e., the capability to estimate the state of each bus in a power system), and thus affect state estimation and other power system applications.

Recent work has focused on the impact of compromised PMUs on the observability of power systems; very little work has studied the impact of compromised PDCs. When a PDC is compromised and quarantined from communication networks, it can cause more severe consequences than a single compromised PMU can, as all measurements that the PDC originally collected are lost. However, PMUs that originally report the measurements to the PDC may not be compromised and can still collect trusted measurements. It is possible to reroute these measurements to other PDCs immediately, instead of waiting for the compromised PDC to be fixed.

To restore the services of PMUs that were disconnected because of compromised PDCs, we propose a self-healing mechanism that exploits the feature of dynamic and programmable configuration enabled by software-defined networking (SDN) technology. When a group of PMUs or PDCs is disconnected, either by accidents or because of a cyber-attack, logical connections between uncompromised PMUs and PDCs are rearranged in order to restore the observability of the power system. The logical reconnection is mapped into the configuration of network switches, which establish new communication paths to deliver PMU measurements. After the reconnection, state estimation and other power system applications can resume working.

We construct the proposed self-healing mechanism as a two-stage procedure in which each stage is modeled as an

Manuscript received January 26, 2016; revised May 12, 2016; accepted June 20, 2016. Date of publication July 27, 2016; date of current version April 19, 2018. This work was supported in part by the U.S. Department of Energy's Office of Electricity Delivery and Energy Reliability, in part by the CREDC under Grant DE-OE0000780, and in part by the National Security Agency under Award H98230-14-C-0141. The work of H. Lin was supported by UChicago Argonne, LLC, Operator of Argonne National Laboratory (Argonne) under Contract DE-AC02-06CH11357. Paper no. TSG-00119-2016.

H. Lin was with Argonne National Laboratory, Argonne, IL 60439 USA. He is now with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: hlin33@illinois.edu).

C. Chen, J. Wang, and J. Qi are with the Energy Systems Division, Argonne National Laboratory, Argonne, IL 60439 USA (e-mail: morningchen@anl.gov; jianhui.wang@anl.gov; jqi@anl.gov).

D. Jin is with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: dong.jin@iit.edu).

Z. T. Kalbarczyk and R. K. Iyer are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Urbana, IL 61801 USA.

Digital Object Identifier 10.1109/TSG.2016.2593021

integer linear programming (ILP) model. In the first stage, we minimize the latency of configuring communication networks to restore the observability of a power grid. In the second stage, we further maximize the redundant observability in the transmission network of power grids. In both ILP models, we consider the constraints of hardware resources in both communication and transmission network infrastructures simultaneously, e.g., the size of the forwarding table in network switches and the connection space of PDCs. Based on the ILP models, we also propose a heuristic algorithm that considers the constraints of transmission and communication networks separately. The proposed heuristic algorithm can reduce the computational complexity in both stages of the self-healing mechanism, while maintaining near-optimal solutions which include the latency to configure communication networks in the first stage and the redundant observability of power grids in the second stage.

We evaluate the self-healing scheme, including both the ILP models and the heuristic algorithm, on PMU networks over both IEEE 30-bus and 118-bus systems. To demonstrate the optimality and the performance of the proposed methods, we compared them to a baseline method, which randomly reconnects PMUs to uncompromised PDCs. The experimental results show that the ILP model can reduce the latency of reconfiguring communication networks by up to 75% compared to the baseline method when hardware resources are limited. Compared to the ILP model, the proposed heuristic algorithm takes less than 25% more overhead on average to reconfigure communication networks, but it takes much less time (by at least one order of magnitude less) to obtain the solution on how to reconfigure the networks.

The remainder of this paper is organized as follows. Section II describes our research plan and the main idea of the proposed method. Section III discusses related work and our contributions as compared to existing work. Section IV describes the design of the self-healing mechanism for a PMU network, including the ILP formulation and heuristic algorithm. Section V provides the experimental results, and we conclude with a discussion of our results in Section VI.

II. Research Plan

We consider a power system that relies on a PMU network to perform state estimation. For each substation, we assume that a single, logical PMU is installed. This PMU can collect data on the state of the local substation (i.e., voltage magnitude and phasor angle). When more PMUs are deployed, a PDC is used to collect measurements from several substations and forward them to the next-level PDC or the control center.

Regarding the deployment of communication networks, we consider the case in which PMUs and PDCs are connected via an IP-based network. Even though in many of today's utility substations, PMUs and PDCs may still be connected through proprietary communications (e.g., serial links), the current trend suggests that the deployment of IP-based networks in power systems is growing; research experiments are already being performed under this assumption [7], [9]. As shown in Fig. 1, since PMUs are deployed on substations that

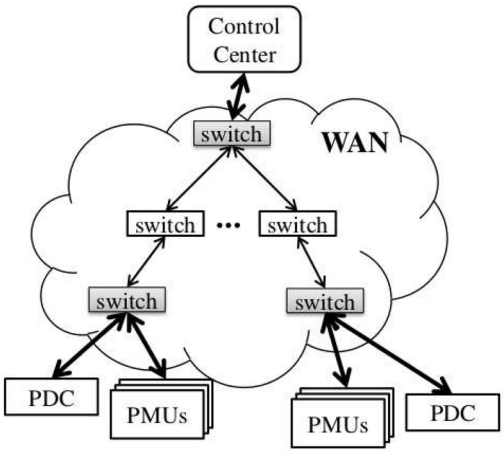


Fig. 1. The integration of a communications network and PMU network.

are distributed over the whole power system, PMUs and PDCs can be connected by a wide area network (WAN). In a WAN, network traffic is manipulated by routing and forwarding rules configured in each network switch. At the perimeter of a WAN, PMUs and PDCs are first connected to edge switches (as highlighted in Fig. 1). In this work, we assume that each PDC is connected with a single edge switch. The edge switches can further connect to the core switches, which are positioned within the backbone of the WAN.

Use of advanced communication network technology, e.g., SDN, can bring both benefits and risks for a power system environment [10]. On the one hand, the communication infrastructure allows attacks to easily propagate to other PMUs. As a result, attackers can gain access to more measurements simultaneously when performing cyber-attacks (e.g., false data injection attacks [11], [12]). On the other hand, programmability enabled by SDN can quickly isolate the compromised PMUs or PDCs and reroute the remaining devices to self-heal the PMU network and recover the system observability.

III. Related Work

Although the deployment and use of PMUs in the power system is still at an early stage, the cybersecurity issues have already been studied in the literature. In the National Electric Sector Cybersecurity Organization Resource (NESCOR) report [13], attack scenarios, impact, and potential mitigation actions are discussed for wide area monitoring, protection, and control (WAMPAC), including those for PMUs [6], [7]. In [9], the authors directly apply existing security mechanisms used in the general computing environment (e.g., firewalls, VPNs, access control mechanism) to a PMU network, to reduce the risk of cyber-attacks. The authors in [8] propose an attack mitigation scheme for a PMU network to prevent propagation of attacks. Those projects were pilot efforts to analyze and enhance the cybersecurity of PMU networks; however, they did not consider how to design mitigation mechanisms that self-heal the PMU network (e.g., reconnecting PMUs/PDCs) by considering the constraints exclusive to PMU networks. In [14], the SDN-enabled network is exploited to design the PMU network

to save network bandwidth. Our scheme further utilizes the potential reconfiguration features enabled by SDN to achieve the self-healing PMU network.

The concept of self-healing has previously been proposed for virtual circuit switching networks, such as the asynchronous transfer mode (ATM) network [15], [16]. When a link or node failure happens, the self-healing algorithms try to recover as many lost services as possible under the resource constraint of network switches. In this network environment, the self-healing is performed on predetermined backup or protection paths [17]. The self-healing mechanism proposed for PMU networks in this paper is quite different from the conventional algorithm. First, the optimization objective in reconnecting PMUs and PDCs is to achieve quick restoration of power systems' observability. This objective is different from the one that is used to restore failed links or nodes in conventional ATM networks, e.g., minimizing the cost of assigning spare links [15], maximizing the amount of traffic restored [18], [19], and maximizing the volume of remaining capacity in routing paths [17], [20]. Because PMUs and PDCs are expensive, deploying spare PMUs or PDCs can be costly, and thus we need to reuse existing uncompromised PMUs and PDCs. Second, the self-healing mechanisms for conventional networks consider the failure of a small range of components, e.g., single link or node failures caused by accidental events, while we assume that attackers can compromise multiple PDCs and PMUs. To the best of our knowledge, this paper is the first to design a self-healing mechanism that makes use of SDN technology to achieve a resilient PMU network.

IV. Proposed Self-Healing Mechanism for PMU Networks

In this paper, we assume that the cyber-attacks on the PMUs and PDCs have already been detected. In practice, system administrators can perform those detections by using security mechanisms, such as intrusion detection systems, designed for power grids. The detection of cyber-attacks in Supervisory Control and Data Acquisition (SCADA) systems, which include PMU networks, has been extensively studied in [21]–[25]. These methods utilize the information in communication networks or the power system's physical models to detect anomalies and intrusions.

In an IP-based network, malware at compromised PMUs or PDCs can infect other devices through network connections [26], [27]. As suggested by a report from the National Institute of Standards and Technology (NIST) [28], after system administrators detect compromised devices, they can place temporary restrictions on network connectivity of those devices to prevent further propagation of the attacks. In practice, system administrators can disconnect the compromised devices by removing routing rules in network switches connected to them; thus, network traffic initiated from the compromised devices can no longer reach any other devices. The work in [29] addressed optimal response strategies for disconnecting nodes in the network, to keep the network in a secure condition. Reference [8] further extended the response strategies in PMU networks.

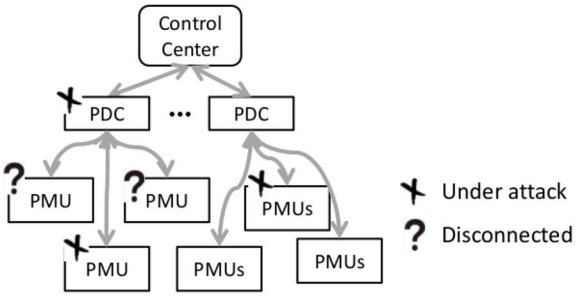


Fig. 2. Device condition after attacks.

When PMUs or PDCs are compromised and disconnected, the consequences can vary. As shown in Fig. 2, when system administrators detect that a PMU has been compromised (denoted by "X"), they disconnect it. The measurements collected by the PMU are lost, which can impact the observability of the power system. Likewise, when a PDC is detected as compromised and disconnected, all measurements that it originally collected are lost. However, the PMUs that originally reported the measurements to the PDC may not be compromised. In that case, we regard the measurements from the disconnected yet uncompromised PMUs, denoted by "?" in Fig. 2, as trusted, unless they are directly compromised. These trusted measurements can be used in power systems' applications, e.g., state estimation, if we can reroute the PMUs into other remaining uncompromised PDCs.

To achieve self-healing in PMU networks, we use the programmability enabled by SDN to reroute the remaining uncompromised PMUs and PDCs, and thus recover the observability of a power system after a cyber-attack. The top priority of a PMU network is to collect measurements from substations under the real-time communications requirements. For that purpose, we find that the existing self-healing algorithms previously proposed for general-purpose networks (e.g., the Internet) are not suitable for a WAN deployed in a PMU network [15], as explained below:

- *First, the optimization objectives of the existing self-healing algorithms are different from those of PMU networks.*
  The existing self-healing algorithms focus on maximizing the connections of end hosts. In the PMU network, however, the top priority is to restore measurements of the voltage phasor at each substation. Because the voltage phasor at a substation can be measured by the PMU deployed at the substation and also the PMU deployed at its neighbor substations, restoring all lost measurements is not equivalent to restoring all disconnected PMUs. Instead, the proposed self-healing algorithm needs to selectively reconnect PMUs, which can restore the observability of power systems more quickly than reconnecting all PMUs.
- *Second, the performance requirements of general communication networks are different from those of PMU networks.*
  The existing self-healing algorithms put more priority on maintaining the network performance, e.g., throughputs

or communication latency, than on the availability of transmitted data. Consequently, those algorithms always select the shortest path to reconnect nodes. In contrast, PMU networks put the availability of phasor measurements at higher priority than the network performance. In our experiments, we implemented in a baseline method an existing self-healing mechanism that reconnects PMUs with the shortest paths (see Section V-A for details); the experiments demonstrated that algorithm often ends up spending a long period of time to reconnect PMUs. Consequently, we propose the self-healing algorithm, which minimizes the time to restore the observability of power systems and maximizes the redundancy of measurements to provide more accurate estimation of system state.

- *Third, the existing algorithm does not consider the constraints in the physical infrastructure of power systems.* In PMU networks, the number of PMUs that can be connected to a PDC is limited by the computation capability and storage space of the PDC. These constraints can impact the paths selected to reconnect PMUs. The self-healing algorithm that we propose takes into consideration the constraints of both the cyber and physical infrastructure in power systems.

Based on that understanding, we propose a new self-healing algorithm that jointly reduces the performance overhead of reconfiguring the communication network and increases the observability of power systems, taking into consideration resource constraints on PDCs and network switches. Specifically, we focus on how to reroute the disconnected but uncompromised PMUs into uncompromised PDCs and do not change the connections of the remaining PMUs.

A. System Model

We use a graph $\mathbb{G}_t(\mathcal{V}, \mathcal{L})$ to denote the topology of a power transmission network, where $\mathcal{V}$ denotes the set of buses and $\mathcal{L}$ denotes the set of transmission lines. We assume that the system observability is achieved by the measurements from PMUs, and let $\mathcal{U}$ denote the set of buses that have PMUs installed, so $\mathcal{U} \subseteq \mathcal{V}$. The IP-based PMU communication network consists of PMUs (also denoted by the set $\mathcal{U}$), PDCs (denoted by the set $\mathcal{D}$), and network switches (denoted by the set $\mathcal{S}$). Thus, the topology of communication networks for delivering PMU measurements is represented by a graph $\mathbb{G}_p(\mathcal{S} \cup \mathcal{U} \cup \mathcal{D}, \mathcal{E})$, where $\mathcal{E}$ denotes the set of network links connecting PMUs, PDCs, and network switches.

A PMU network is a cyber-physical system. From cyber systems' perspective, a communication network should make sure that measurements from PMUs can be delivered to PDCs and the control center; from physical systems' perspective, the PMUs should make sure that the whole power system is observable, so that the state estimation and other advanced power system applications can be performed. We integrate these cyber-physical features into the design of our self-healing mechanism for PMU networks. These features differentiate the proposed algorithm from the existing self-healing schemes, which emphasize the maintenance of network performance and reconnection of end hosts in general communication networks [15]–[17]. In the remainder of this subsection, we will briefly describe models of power system observability by PMUs and rules in network switches; these models are integrated into the self-healing scheme design for PMU networks.

1) Power System Observability: When a PMU is installed at bus $i \in \mathcal{U}$, the voltage phasor at bus i and current phasor of all branches connected to it can be measured. The observability function of bus i is defined as a function of a PMU location:

$$O_i = \sum_{j \in \mathcal{U}} a_{i,j} x_j, \tag{1}$$

where x_j is a binary variable that is equal to 1 if a PMU is installed at bus j and 0 otherwise. $a_{i,j}$ is the connectivity parameter, defined as:

$$a_{i,j} = \begin{cases} 1 & i = j \text{ or } (i,j) \in \mathcal{L} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

$O_i \geq 1$ implies that bus i is observable, as the voltage phasor at bus i can either be measured by the PMU at bus i, or be calculated by PMUs at neighbors of bus i (e.g., the buses connected through transmission lines). The power system is observable if the observability function O_i for each bus is greater than or equal to 1, i.e.,

$$O_i \geq 1, \ \forall i \in \mathcal{V}. \tag{3}$$

With the disconnection of some PMUs due to cyber-attacks, the observability function O_i at some buses may become 0; thus, the system is no longer observable. However, by utilizing the reconfiguration features enabled by SDN, it is possible to reconnect some disconnected yet uncompromised PMUs to the communication network to restore the system observability.

2) Rules in Network Switches: The reconnection of disconnected yet uncompromised PMUs can be achieved by adding rules in network switches. In a communication network, the switch can include rules that specify both routing policies and endpoint policies [30]. Given a packet entering the network, the routing policy specifies the path that the network packet should take to reach its destination. The path is often expressed as a chain of ordered network switches. To implement a routing policy, we add a forwarding rule in each switch on the path, to direct the network packet to the appropriate following stop. A forwarding rule in a switch usually is uniquely decided by destination addresses. In other words, a path always corresponds to a unique destination.

An endpoint policy often defines the access control between two hosts. In other words, the policy specifies whether or not host A can communicate with host B, regardless of what path the communication should follow (which is decided by the routing policies). As explained in [30], the endpoint policy often "views the network as one big switch that hides internal topology details" and "specifies which packets to drop, or to forward to specific egress ports, as well as any modifications of header fields." In the PMU network, an endpoint policy specifies to which PDC a PMU measurement should be delivered.

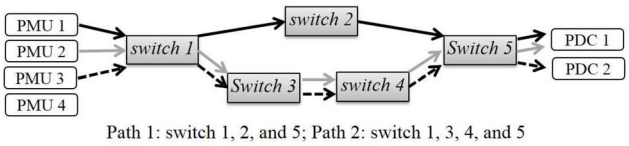


Fig. 3. An example network connection.

Unlike the routing policy, the endpoint policy is implemented once along the path that the packet travels. For example, in the network topology shown in Fig. 3, we want to deliver the measurements from PMU 1 to PDC 1 through path 1. In this case, the routing policy destined for PDC 1 is implemented by adding a forwarding rule in each switch of path 1 (i.e., switches 1, 2, and 5). However, not all packets destined for PDC 1 and traveling through this path are from PMU 1. To ensure that the measurements from PMU 1 are delivered to PDC 1 via path 1, we need to add a rule for this endpoint policy along the path. This rule can be implemented only once in switch 1, 2, or 5 before packets reach PDC 1.

B. Optimization Formulation

In this subsection, we describe how we model the self-healing mechanism for PMU networks as an integer linear programming problem. To better illustrate the optimization formulation, in addition to the parameters defined in Section IV-A1, we list key notations in Table I.

We define four groups of integer variables in the optimization model:

- Variable x_i indicates whether the PMU at bus $i \in \mathcal{U}$ is connected, regardless of which path in the communication network $\mathbb{G}_p$ it has taken and to which PDC it is connected.
- Variable y_p specifies whether path $p \in \mathcal{P}$ is used to achieve the reconnection, regardless of which PMUs and PDCs use this path for reconnection. Instead of considering the shortest path, we consider all paths that can be used to reconnect PMUs, as long as the delivery of PMU measurements over that path satisfies the timing requirement. Finding all paths between two nodes is regarded as an "all simple paths" problem; we use a "depth-first search" to search all paths that can be used for reconnections. Although there is no efficient algorithm to solve this problem, we can find all paths before running the optimization model at runtime.
- Variable z_s^p, as in [30], specifies the number of endpoint policies allocated for path p and assigned in switch s. Since a switch can be shared by different paths, a subscript p is used for each switch to distinguish the endpoint policies used for different "PMU-PDC" connections.
- Variable w_s^d indicates whether switch s contains a forwarding rule destined to PDC d. Because a forwarding rule is indexed by the destination address, a switch can include forwarding rules destined for different PDCs. Thus a subscript d is used to distinguish the forwarding rules assigned for different destination PDCs.

To better understand these decision variables, we provide an example in Fig. 3. If we assume that three PMUs at three buses (i.e., buses 1, 2, and 3) are reconnected, while PMU at bus 4 is not, we have $x_1 = x_2 = x_3 = 1$ while $x_4 = 0$. In addition, if we use two paths (i.e., paths 1 and 2) for reconnection, we have $y_1 = y_2 = 1$. In addition, we assume that the PMU at bus 1 is reconnected to PDC 1, while the PMUs at bus 2 and 3 are reconnected to PDC 2, as illustrated by lines with different patterns in Fig. 3. To appropriately deliver measurements to the right destination, the forwarding rules destined for PDC 1 are added to the switches of path 1, which consists of network switches 1, 2, and 5. Similarly, a forwarding rule destined for PDC 2 is added to the switches of path 2, namely switches 1, 3, 4, and 5. Consequently, we have $w_1^1 = w_2^1 = w_5^1 = 1$ and $w_1^2 = w_3^2 = w_4^2 = w_5^2 = 1$.

An endpoint policy is decided by the connection between PMUs and PDCs. Consequently, there are three rules for the endpoint policy in this example, which are three specific "PMU-PDC" connections: (PMU 1, PDC 1), (PMU 2, PDC 2), and (PMU 3, PDC 2). For each connection, we need to add a rule once along the path that established the connection. For example, if we add an endpoint policy that specifies a connection between PMU 3 and PDC 2, we can add the

TABLE I
SUMMARY OF NOTATIONS

Sets and Indices:	
$\mathcal{U}$	Set of buses with PMUs installed.
$\mathcal{U}_b$	Set of buses with compromised PMUs.
$\mathcal{U}_g$	Set of buses with uncompromised PMUs, $\mathcal{U} = \mathcal{U}_b \cup \mathcal{U}_g$.
$\mathcal{U}_d$	Set of buses with uncompromised but disconnected PMUs, $\mathcal{U}_d \subset \mathcal{U}_g$.
$\mathcal{D}$	Set of PDCs used in the PMU network.
$\mathcal{D}_g$	Set of uncompromised PDCs.
$\mathcal{S}$	Set of network switches in a communication network.
$\mathcal{P}$	Set of paths in the communication network, denoted by graph $\mathbb{G}_p$, that can be used to deliver measurements from a PMU to a PDC with the real-time requirements observed.
$\mathcal{U}^p$	Set of buses with disconnected yet uncompromised PMUs that can be reconnected by path p.
$\mathcal{D}^p$	Set of uncompromised PDCs that are connected by path p.
$\mathcal{P}^i$	Set of paths that can be used to connect a disconnected yet uncompromised PMU at bus $i \in \mathcal{U}_d$.
$\mathcal{P}^d$	Set of paths that are connected to a remaining PDC $d \in \mathcal{D}_g$.
$\mathcal{S}^p$	Set of network switches in path p.
$\mathcal{V}_z$	Set of zero-injection buses in the power system.
i, j, k	Indices of buses.
d	Index of PDCs.
s	Index of network switches in set $\mathcal{S}$.
p	Index of paths in set $\mathcal{P}$.
Parameters:	
C_d	Number of additional PMUs to which a PDC $d \in \mathcal{D}_g$ can connect.
R_s	Number of additional rules that switch $s \in \mathcal{S}$ can accommodate.
Decision Variables:	
x_i	Binary variable that equals 1 when the PMU at bus i is connected, and 0 otherwise.
y_p	Binary variable that equals 1 when path $p \in \mathcal{P}$ is selected to reconnect PMUs, and 0 otherwise.
z_s^p	Integer variable to specify the number of endpoint rules that are allocated for path p and are implemented in switch s.
w_s^d	Binary variable that equals 1 when a forwarding rule destined for the PDC d is added in network switch s, and 0 otherwise.

endpoint policy at switch 5 of path 2, which is established by routing policies. In that case, we should increase the value of variable z_5^2 by one.

With those notations, we model the constraints and objectives of the ILP formulation for our self-healing scheme as follows.

1) PMU Connection Status Constraints (PCSC): After the disconnection of compromised PMUs and PDCs in response to cyber-attacks, the PMU connection status needs updates. Let $\mathcal{U}_b$ denote the set of buses with compromised PMUs. Because these PMUs will remain disconnected, decision variable x_i for $i \in \mathcal{U}_b$ should be set to 0, i.e.,

$$x_i = 0, \ \forall i \in \mathcal{U}_b. \tag{4}$$

In addition, let $\mathcal{U}_g$ denote the set of buses with uncompromised PMUs, and $\mathcal{U}_d$ denote the set of buses with uncompromised and disconnected PMUs. As the remaining connected and uncompromised PMUs (denoted by set $\mathcal{U}_g \backslash \mathcal{U}_d$) will remain unchanged, we have the following constraints:

$$x_i = 1, \ \forall i \in \mathcal{U}_g \setminus \mathcal{U}_d. \tag{5}$$

The PMUs in set $\mathcal{U}_d$ can be reconnected by the self-healing scheme as the actual decision variables; thus, the following constraints apply:

$$x_i = \{0, 1\}, \ \forall i \in \mathcal{U}_d. \tag{6}$$

2) Power System Observability Constraints (PSOC): The constraints in (1)–(3) specify the observability function O_i at bus i as a function of the topology of the power system's transmission network and PMU locations. The considered power system is observable when $O_i \geq 1$ for all buses.

If there are zero injection buses, i.e., no generation or load units are connected to them, the voltage phasor at certain buses (e.g., those zero injection buses or their neighbors in transmission networks) can be calculated indirectly by applying Kirchoff's Current Law (KCL). Consequently, the number of PMUs needed to make sure that the system is observable can be reduced. We apply the formulation in [31] that considers zero injection buses. For each zero injection bus k, a linear relation between its voltage phasor and the phasor of its neighbor can be obtained based on KCL as:

$$\sum_{j \in \mathcal{V}} Y_{k,j} \bar{V}_j = 0, \tag{7}$$

where $\bar{V}_j$ is the voltage phasor at bus j, and $Y_{k,j}$ is the $k-j$th entry of the admittance matrix of the power system. A group of equations handling zero injection buses forms a system of linear equations, from which some voltage phasors can be calculated to make the corresponding buses observable. As shown in [31], the observability function O_i with zero injection buses considered can be modeled as:

$$O_i = \sum_{j \in \mathcal{U}} a_{i,j} x_j + \sum_{k \in \mathcal{V}_z} a_{i,k} v_{i,k}, \ \forall i \in \mathcal{V} \tag{8}$$

$$\sum_{i \in \mathcal{V}} a_{i,k} v_{i,k} = 1, \ \forall k \in \mathcal{V}_z \tag{9}$$

$$\sum_{k \in \mathcal{V}_z} a_{i,k} v_{i,k} \leq 1, \ \forall i \in \mathcal{V}, \tag{10}$$

where $\mathcal{V}_z$ denotes the set of zero injection buses, and the auxiliary binary variable $v_{i,k}$ is defined so that $v_{i,k} = 1$ implies that calculation of the voltage phasor at bus i is assigned to the equation associated with zero injection bus k. The constraints (9) and (10) guarantee the solvability of the group of equations corresponding to zero injection buses. The detailed derivation of (8)–(10) can be found in [31]. The power system observability constraints can be modeled as (2)–(3), and (8)–(10).

3) PDC Connection Space Constraints (PSC): The disconnected PMUs can be reconnected only to PDCs with sufficient connection spaces. The connection space of a PDC is defined as the maximum number of PMUs it can concentrate. This parameter can be found in the specification of the PDC, e.g., SEL-3373 PDC can concentrate up to 40 PMUs [32]. This constraint can be modeled as follows:

$$\sum_{p \in \mathcal{P}^d} \left(y_p \sum_{i \in \mathcal{U}^p} x_i \right) \leq C_d, \ \forall d \in \mathcal{D}_g, \tag{11}$$

where C_d denotes the additional PMUs that PDC d can connect. In the constraint, we use set $\mathcal{U}^p$ to include all disconnected yet uncompromised PMUs that can be reconnected by path p; this set is constructed by including all PMUs that are connected by the edge switch at the ending node of path p. Furthermore, we use set $\mathcal{P}^d$ to specify all paths that are connected to the remaining PDC $d \in \mathcal{D}_g$.

If path p is selected for the reconnection, y_p is set to 1. The innermost summation on the left side of constraint (11) is used to calculate the number of PMUs that path p can reconnect. Note that in the IP-based communication network, the routing path is uniquely decided by its destination address. A PDC can be used by different paths to connect different PMUs. When multiple paths connected to PDC d are selected, the outermost summation calculates all PMUs that can be reconnected by those paths. We use constraint (11) to ensure that the remaining connection spaces of PDC d can satisfy all those disconnected PMUs.

Obviously, constraint (11) is a nonlinear constraint. To lower the computational complexity, we reformulate the constraint (11) to linear forms in (12)–(14) using the big-M method with an auxiliary variable t_p. We use t_p to represent the number of PMUs that path p can reconnect if the path is selected. When $y_p = 1$, constraints (12)–(13) become an equality constraint, i.e., $t_p = \sum_{i \in \mathcal{U}^p} x_i$. On the other hand, if $y_p = 0$, the big number M ensures that the left side of constraint (12) is a nonpositive number. Consequently, constraints (12)–(13) are equivalent to another inequality, i.e., $0 \leq t_p \leq \sum_{i \in \mathcal{U}^p} x_i$. The optimization makes t_p equal to 0, to let the PDC's connection spaces C_d accommodate more PMUs via other paths. Consequently, we set the value of M as $|\mathcal{U}_d|$, where $|\cdot|$ denotes the number of elements in a set. In other words, we set the value of M to the total number of disconnected PMUs. Because $\sum_{i \in \mathcal{U}^p} x_i < |\mathcal{U}_d|$, setting M with that value ensures that the left side of constraint (12) is a nonpositive number.

$$-\left(1 - y_p\right) \cdot M + \sum_{i \in \mathcal{U}^p} x_i \leq t_p \leq \sum_{i \in \mathcal{U}^p} x_i \tag{12}$$

$$t_p \geq 0, \ \forall p \in \mathcal{P}, \tag{13}$$

$$\sum_{p \in \mathcal{P}^d} t_p \leq C_d, \ \forall d \in \mathcal{D}_g. \tag{14}$$

4) PMU Reconnection Constraints (PRC): When a PMU is selected for reconnection, at least one path must be selected to reconnect it to a PDC, i.e.,

$$x_i \leq \sum_{p \in \mathcal{P}^i} y_p, \ \forall i \in \mathcal{U}_d. \tag{15}$$

In constraint (15), we use set $\mathcal{P}^i$ to include any path that can be used to connect a disconnected PMU at bus i. When the PMU at bus i is reconnected, the constraint guarantees that at least one path can connect this PMU to a PDC.

5) Switch Rule Space Capacity Constraints (SRCC): There should be sufficient space for each network switch to add the rules of both endpoint and routing policies; this requirement is modeled as the following constraints:

$$\sum_{d \in \mathcal{D}_g} w_s^d + \sum_{p \in \mathcal{P}} z_s^p \leq R_s, \ \forall s \in \mathcal{S}, \tag{16}$$

$$w_s^d \in \{0, 1\}, \ \forall s \in \mathcal{S}, \ \forall d \in \mathcal{D}_g, \tag{17}$$

$$z_s^p \geq 0, \ \forall s \in \mathcal{S}, \ \forall p \in \mathcal{P}, \tag{18}$$

where $\sum_{d \in \mathcal{D}_g} w_s^d + \sum_{p \in \mathcal{P}} z_s^p$ is the total number of rules added to switch s.

6) Routing Policy Constraints (RPC): When a PDC is used to reconnect PMUs, a routing rule should be added to each network switch in the path that connects the PMUs to the PDC, i.e.,

$$w_s^d \geq y_p, \ \forall p \in \mathcal{P}, \forall d \in \mathcal{D}^p, \forall s \in \mathcal{S}^p, \tag{19}$$

$$y_p \in \{0, 1\}, \ \forall p \in \mathcal{P}. \tag{20}$$

We specify the constraint (19) for each network switch in the path p, i.e., $s \in \mathcal{S}^p$, where $\mathcal{S}^p$ denotes the set of network switches in path p. The set $\mathcal{D}^p$ includes PDCs that can be connected through path p. If path p is selected to connect PMUs, we add a forwarding rule in each switch in p that is used to transmit network packets destined for the corresponding PDCs. If the path is not selected, constraint (19) becomes equivalent to constraint (17).

7) Endpoint Policy Constraints (EPC): The total number of rules for the endpoint policy should be equal to the total number of PMUs that are reconnected, i.e.,

$$\sum_{i \in \mathcal{U}_d} x_i = \sum_{s \in \mathcal{S}} \sum_{p \in \mathcal{P}} z_s^p. \tag{21}$$

Constraint (21) specifies that if we choose to reconnect a PMU, we need to add one rule in the endpoint policy to specify this connection. In practice, however, it is possible to combine several rules into a single one by using techniques such as wildcards to specify multiple "PMU-PDC" connections [30], [33]. Designing algorithms to combine rules of endpoint policy is an active research area, but is outside the scope of this paper. Therefore, we do not apply any algorithm to combine the rules of the endpoint policy. As a result, constraint (21) is observed in the ILP model.

8) Optimization for Self-Healing Mechanism: After certain PMUs and PDCs are disconnected in response to cyber-attacks, the self-healing mechanism first checks the system observability by calculating the observability function O_i at each bus, according to (1)–(2) for a system without zero injection buses, or according to (8)–(10) and (2) for a system with zero injection buses.

Depending on the system observability status, we divide the self-healing mechanism into two stages. In the first stage, we try to recover the observability of the power system by a PMU network as quickly as possible. In the second stage, we recover the remaining disconnected PMUs to increase redundancies of PMU measurements and thus more accurately estimate system states.

Stage 1: Recover system observability

If the remaining connected PMUs (denoted by set $\mathcal{U}_g \setminus \mathcal{U}_d$) cannot make the system observable, i.e., $O_i \geq 1, \forall i \in \mathcal{V}$ is not satisfied, but the disconnected yet uncompromised PMUs (denoted by set $\mathcal{U}_d$) would make the system observable, then the self-healing mechanism will enter Stage 1.

At this stage, we want to minimize the time it will take to recover system observability. We use the total number of rules that all network switches need to modify to estimate the time to reconfigure network switches in order to reconnect PMUs. The number of rules in network switches plays an important role in the performance of both traditional and SDN-enabled networks. Ways to reduce or compress the size of network rules have been an active research area for almost a decade. The authors of [33] and [34] propose algorithms to reduce and compress the number of rules stored in a single switch. In later work, [30], [35], and [36] focus on how to distribute rules further in optimal locations in order to save storage space in an SDN-enabled network.

In the scenario that recovers the observability of power systems, the number of rules to add can impact the performance of PMU networks as well. In a traditional communication network or a more advanced network environment that uses SDN technology [10], modifying a rule in a network switch requires the system operator to interact with the switch and make the corresponding configuration. The round-trip time of this communication in a wide area network can take up to hundreds of milliseconds [37]. In PMU networks that perform real-time monitoring on system states, optimizing the configuration time by reducing the number of switch rules can help improve performance.

Consequently, the optimization at stage 1 can be formulated as:

$$\begin{aligned} \min_{x_i, y_p, z_s^p, w_s^d} \ & \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} z_s^p + \sum_{d \in \mathcal{D}_g} \sum_{s \mathcal{S}} w_s^d \\ \text{s.t.} \ & \text{PCSC: (4)} - \text{(6)}, \\ & \text{PSOC: (2)} - \text{(3)}, \ \text{(8)} - \text{(10)}, \\ & \text{PSC: (12)} - \text{(14)}, \\ & \text{PRC: (15)}, \\ & \text{SRCC: (16)} - \text{(18)}, \\ & \text{RPC: (19)} - \text{(20)}, \\ & \text{EPC: (21)}, \end{aligned} \tag{22}$$

which is an ILP problem. The objective of the self-healing mechanism at this stage is to minimize the number of rules for both the routing and the endpoint policies, in order to

restore the disconnected yet uncompromised PMUs to the uncompromised PDCs. The details of all constraints of the ILP formulation are discussed in previous sections, i.e., from Sections IV-B1 and IV-B2.

Stage 2: Maximize system observability

If the power system becomes observable after Stage 1, the self-healing mechanism will enter Stage 2 to continue recovering the remaining disconnected PMUs. Note that at the beginning of the self-healing process, if the power system is already observable with the remaining connected PMUs (denoted by set $\mathcal{U}_g \setminus \mathcal{U}_d$), or if reconnection of all disconnected yet uncompromised PMUs fails to make the system observable, the self-healing mechanism will omit the optimization at Stage 1 and directly enter Stage 2.

At this stage, the aim is to improve the observability of the power system. We use the minimum observability function of all buses, i.e., $\min_{i\in\mathcal{V}} O_i$, to quantify the observability function of the entire system. The objective at this stage can be formulated as:

$$\max_{x_i, y_p, z_s^p, w_s^d} \min_{i\in\mathcal{V}} O_i, \tag{23}$$

where all constraints except (3) in optimization (22) are still applied; constraint (3) is removed because the system observability either cannot be satisfied or has already been satisfied. The set defined in Table I needs to be updated according to the decisions made at Stage 1. Through introduction of an auxiliary variable O, the max-min optimization problem in (23) can be reformulated as:

$$\begin{aligned} \max_{x_i, y_p, z_s^p, w_s^d} \quad & O \\ \text{s.t.} \quad & O \le O_i, \forall i \in \mathcal{V}, \\ & \text{all constraints in (22) except (3)}, \end{aligned} \tag{24}$$

which is also an ILP problem.

C. Greedy Heuristic Algorithm

In the proposed ILP model, the number of variables is on the scale of $O(|\mathcal{U}_d| + |\mathcal{D}_g| + |\mathcal{P}| \times |\mathcal{S}|)$. Consequently, ILP solvers suffer from its dimensionality, especially when a power system and underlying communication network increase in size. In that case, solving the problem can introduce a long delay; the slow response to the attacks can result in the damage to PMU networks.

In Algorithm 1, we propose a greedy heuristic algorithm to find reroutes for disconnected PMUs one by one instead of finding a global optimal reroute for all PMUs. As shown in Step 3 in Algorithm 1, the heuristic algorithm always selects the PMU on the bus with the largest degree in power systems' transmission networks (i.e., the bus that has the largest number of neighbors in the transmission networks), to make the system observable by reconnecting a small number of PMUs. In Steps 4–28, among paths with sufficient hardware resources (e.g., rule spaces in network switches and connection spaces in PDCs), the heuristic selects the path with the smallest latency, as specified by *latency(p)* in Algorithm 1, to deliver measurements. In practice, the latency of delivering measurements can be measured by different metrics, such as the number of switches in a path or round-trip times.

Algorithm 1 Greedy heuristic algorithm

1: **while** $\mathcal{U}_d$ is not empty **do**
2: *CandidatePDCs* = {}, *CandidatePaths*={}, *Routes*={}
3: Select $i \in \mathcal{U}_d$ with the largest degree in transmission networks
4: **for all** $d \in \mathcal{D}_g$ **do**
5: **if** d has connection spaces **then**
6: Put d in *CandidatePDCs*
7: **end if**
8: **end for**
9: **for all** $d \in$ *CandidatePDCs* **do**
10: Find the communication path p between d and i with the smallest latency, calculated by *latency(p)*
11: **if** *latency(p)* is shorter than the latency requirement of delivering PMU measurements **then**
12: Put p in *CandidatePaths*
13: **else**
14: Remove d from *CandidatePDCs*
15: **end if**
16: **end for**
17: **while** *CandidatePaths* is not empty **do**
18: Select p of smallest *latency(p)* from *CandidatePaths*
19: **if** p has no rule space **then**
20: Remove p from *CandidatePaths*
21: **else**
22: Put p in *Routes*;
break;
23: **end if**
24: **end while**
25: Remove i from $\mathcal{U}_d$
26: Connect i through p
27: Set routing policy in each switch in p
28: Set endpoint policy in the first switch in p that has available space
29: **if** power grid becomes observable **then**
30: Continue; // mark the end of Stage 1
31: **end if**
32: **end while**

Unlike the ILP model, Algorithm 1 considers the different optimization objectives at Stages 1 and 2 together. In the heuristic, Steps 29–31 mark the end of Stage 1 (i.e., observability recovery), and the heuristic continues to select disconnected PMUs until all of them have been reconnected.

V. Experiments

In this section, we first present a simple case study on the small scale IEEE 30-bus system, to demonstrate how the proposed ILP model and the greedy heuristic algorithm reconnects PMUs. Then, in Section V-B, we evaluate how the proposed methods perform in both IEEE 30-bus and 118-bus systems.

A. Case Study

We use the IEEE 30-bus system to demonstrate how the self-healing mechanism reconnects compromised and disconnected PDCs and PMUs.

Based on the topology of transmission networks, we construct the topology of the communication network of a power system through the following procedure. First, we employ the minimum set cover problem to find a set of substations that

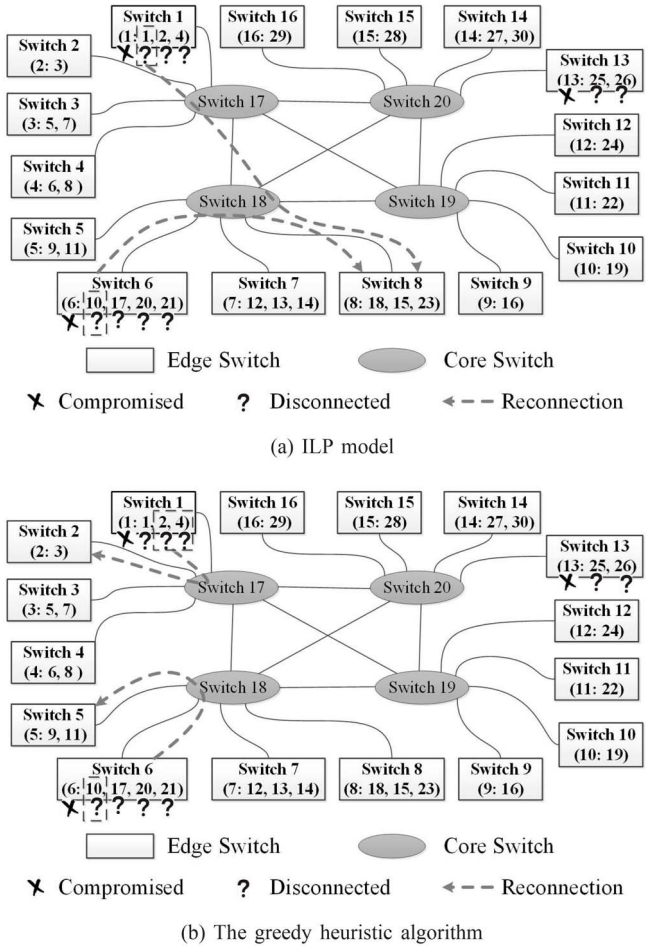


Fig. 4. Reconnection of PMUs in a communication network for the IEEE 30-bus system with (a) the ILP model and (b) the greedy heuristic algorithms.

cover all transmission lines [38], [39]. The consequence is that we classify substations into different sets. We assign a network switch for each set as an edge switch, which connects PMUs deployed among all substations in each set. To each edge switch, we add enough PDCs to combine all measurements of the PMUs that are connected to the same switch.

The communication network of the IEEE 30-bus system is shown in Fig. 4. We use white rectangles to represent edge switches. In each edge switch, we specify the index of the PMU and PDC that the switch connects in a format of (PDC#: PMU#, ...). In this network, there are 16 edge switches that connect PMUs in substations and 4 core switches that form a mesh backbone network; the edge switches are evenly distributed among the core switches.

In Fig. 4(a), we also demonstrate how the proposed self-healing mechanism (implemented in the ILP model) and the greedy heuristic algorithm are working. We consider a case in which PDCs 1, 6, and 13 are compromised by attackers. Consequently, measurements from PMUs 1, 2, 4, 10, 17, 20, 21, 25, and 26 cannot be delivered to the control center, even though none of them is compromised. We formulate the ILP problem in the OPTI toolbox, which can give us a solution that reconnects PMUs 1 and 10 to PDC 8 (shown by dotted arrows in Fig. 4(a)). Such reconnections need two rules to specify the endpoint policy that grants their access to the PDC 8. Meanwhile, a forwarding rule destined for the PDC 8 are added to switches 1, 6, 7, 18, and 8. As a result, there are a total of seven rules to configure.

In Fig. 4(b), we show that the greedy heuristic algorithm can introduce a different and nonoptimal result for the same case. Also, we use this case to demonstrate the procedure of the greedy heuristic algorithm shown in Algorithm 1.

- *Step 3*: We rank all disconnected PMUs based on their degrees in transmission networks (i.e., the number of neighbor substations that each PMU has); the proposed heuristic tries to reconnect PMUs one by one based on this order. Because a PMU can measure a voltage phasor not only at the substation where it is deployed, but also at its neighbor substations, we select PMUs in that order will restore the observability of the power system as quickly as possible. Specifically in the attack case considered in Fig. 4, we select PMUs in the following order: PMUs 4, 10, 2, 5, 1, 17, 20, 21, and 26. When we select a PMU, we find it a PDC (Steps 4–8) and a path (Steps 9–16). Without loss of generality, we describe the following steps to reconnect PMU 4, which is the first PMU to reconnect in this attack case.
- *Steps 4–8*: When we select PMU 4 for reconnection, we find all remaining PDCs (i.e., all PDCs except for PDCs 1, 6, and 13) that have sufficient connection space for this PMU. We regard all those PDCs as candidate PDCs that can reconnect the PMU (stored in a set *CandidatePDC* in Algorithm (1)).
- *Steps 9–16*: For each PDC in *CandidatePDC*, we find the shortest path (i.e., with the shortest round trip time) between the PDC and PMU 4. If the round trip time of this path meets the requirement of being able to deliver PMU measurements, we regard this path as a candidate path. We put all candidate paths in a set, i.e., *CandidatePaths* in Algorithm (1). To simplify discussions, we assume that the round trip times on all network links are the same; the round trip time of a path can be quantified by the number of links or the number of switches in a path.
- *Steps 17–24*: If set *CandidatePaths* is not empty, we select a path from the set that has the shortest round trip time to reconnect PMU 4. For example, we can select PDC 2 to reconnect PMU 4; the path contains only three switches, i.e., Switch 1, Switch 17, and Switch 2.
- *Steps 29–31*: After reconnecting a PMU, we check whether the power system has become observable or not. If the system has become observable, we continue to Stage 2 of the optimization to achieve more redundancy of PMU measurements.

To show the effectiveness and optimality of our proposed methods, including both the ILP model and the greedy heuristic algorithm, we compare them against a baseline method, which is based on self-healing mechanisms proposed for traditional communication networks. Note that the traditional self-healing mechanism intends to recover as many disconnected hosts or links as possible, which is different from the recovery of observability of a power system. In this baseline

method, we randomly reconnect PMUs with the shortest paths to any PDCs that have sufficient connection space.

In addition, we implement the self-healing mechanism in Mininet, a software platform to simulate SDN-enabled communication networks, to demonstrate the procedures for self-healing PMU networks against the attack case considered in Fig. 4.

B. Performance Evaluation of Stage 1

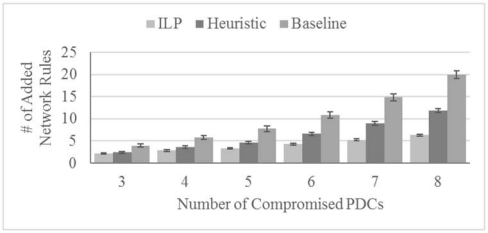
We present the performance evaluation of the proposed ILP model and the greedy heuristic algorithm, and compare them against the baseline method, on both IEEE 30-bus and IEEE 118-bus test systems. Specifically, we compare the optimization results and execution times of these two methods when the hardware resources, such as rule spaces of network switches and connection spaces of PDCs, become limited. We use the solver in the OPTI Toolbox to solve the ILP problem formulated in Section IV-B [40]. The heuristic algorithm presented in Section IV-C is implemented in MATLAB. All of our experiments were performed on a 64-bit desktop with two Intel Core I7 3.6 G processors and 16 GB of RAM.

1) Impact of Scale of Attacks: We assume that each PDC can combine measurements from up to 40 PMUs, which is the connection space of the SEL 3373 [32]. Each network switch can contain up to 1000 forwarding or routing rules, which is in line with the experiment settings in [30].

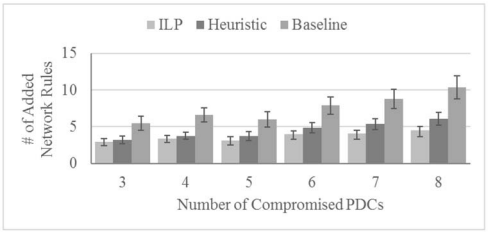
Fig. 5 shows the impact of the scale of attacks on the number of rules added to recover the observability of power systems. The horizontal axis specifies the number of compromised PDCs. In each case, we randomly selected the compromised PDCs and performed the experiments 500 times. The vertical axis specifies the average number of network rules that are added to reconnect PMUs. We use bars of different shades of gray to represent the results obtained by the ILP problem, the greedy heuristic algorithm, and the baseline method with a 95% confidence interval. Obviously, if attackers compromise more PDCs, we need to add more rules in network switches to reconnect PMUs.

Based on the result, we can see that the greedy heuristic algorithm performs better when a small number of PDCs are compromised in both 30-bus and 118-bus systems. The worst case happens when 8 PDCs are compromised in the 30-bus system; the greedy heuristic algorithm needs to add 5 to 6 more rules, on average, to recover the observability of the power system. The baseline method performs much worse than the greedy heuristic, as it can take a long time to reconnect enough PMUs to recover the observability.

Fig. 6 compares the execution times of the ILP model, the greedy heuristic algorithm, and the baseline method. The execution time of the ILP model is specified by the major vertical axis, while the execution times of the other two algorithms are specified by the secondary vertical axis. Because the execution times vary significantly for the compromise of different PDCs, we selected the 50 largest execution times for those three methods and include their average in Fig. 6. Because the number of compromised PDCs impacts the size of the search space in the ILP model, the execution time to solve the ILP

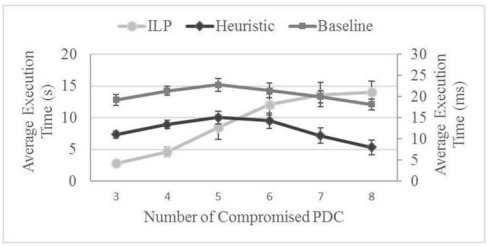


(a) IEEE 30-bus system

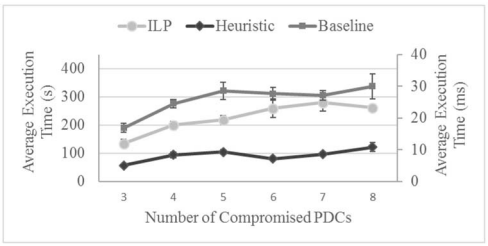


(b) IEEE 118-bus system

Fig. 5. Comparison of the number of network rules to add for (a) the IEEE 30-bus system and (b) the IEEE 118-bus system.



(a) IEEE 30-bus system

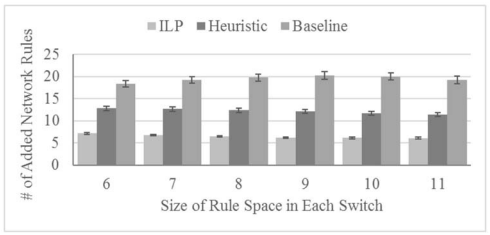


(b) IEEE 118-bus system

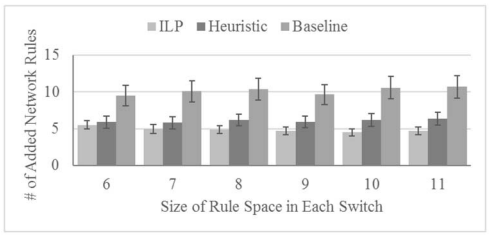
Fig. 6. Comparison of the execution times for (a) the IEEE 30-bus system and (b) the IEEE 118-bus system.

model (represented by the light grey line in both sub-figures) increases slightly with the number of compromised PDCs. On average, the greedy heuristic algorithm can reduce the execution time by approximately two orders of magnitude. In the greedy heuristic algorithm, however, we reconnect PMUs one by one, starting from the one on the bus with most neighbors in transmission networks. The number of reconnected PMUs is not directly related to the number of compromised PDCs, and the execution time does not change significantly. The baseline method randomly selects PMUs for reconnection; the method can take a long time to restore the observability of power systems.

The execution times of these methods scale differently with the size of the power system. Because there are more paths to reconnect PMUs in the 118-bus system, the search space in

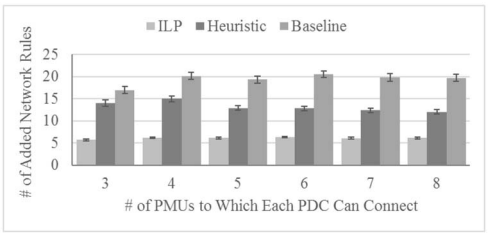


(a) IEEE 30-bus system

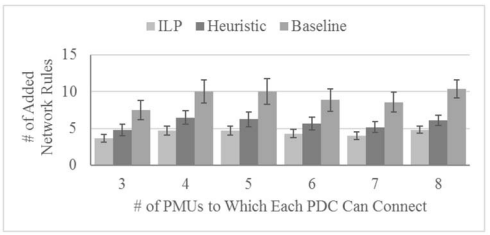


(b) IEEE 118-bus system

Fig. 7. Comparison of the numbers of added network rules with impact of limited network resources.

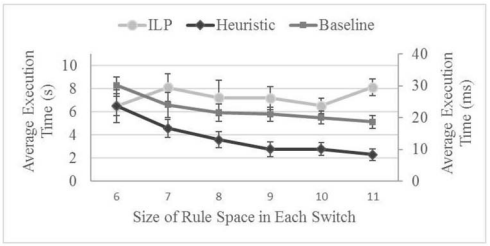


(a) IEEE 30-bus system

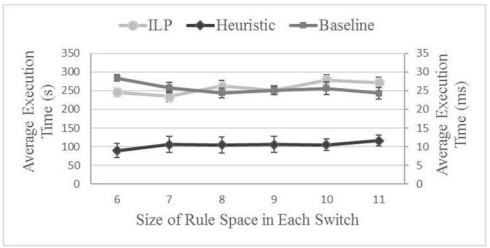


(b) IEEE 118-bus system

Fig. 8. Comparison of the numbers of added network rules with impact of limited PDC connection space.



(a) IEEE 30-bus system



(b) IEEE 118-bus system

Fig. 9. Comparison of the execution times with impact of limited network resources.

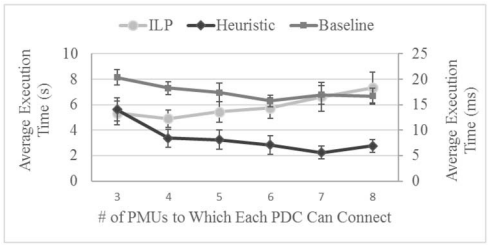
the ILP models increases dramatically, which can increase the execution time of the ILP model by more than ten times. The execution times of the greedy heuristic algorithm and the baseline method do not change significantly as these algorithms select PDCs that can connect PMUs with the shortest distance. For communication networks of the 30-bus and 118-bus systems, disconnected PMUs are often reconnected to PDCs in their neighbors based on the greedy heuristic algorithm and the baseline method.

2) Impact of Hardware Resources: In this section, we evaluate the impact of hardware resources, i.e., rule spaces in communication networks and connection spaces (the number of PMUs to which a PDC can connect), on the performance of the self-healing mechanisms. In practice, these resources can be shared with other devices in addition to being assigned to PMUs and PDCs. For the simulated communication network, we set the size of the rule space of each switch to be between 5 and 10 and the number of PMUs to which each PDC can connect to be between 3 and 8. For each value set for those two parameters, we randomly selected 8 compromised PDCs and performed the experiments 500 times.

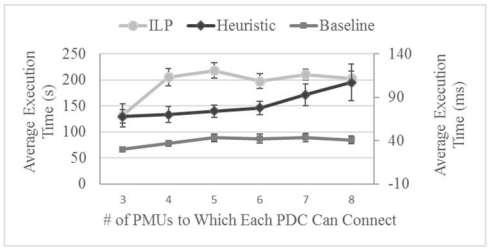
In Fig. 7 and Fig. 8, we show how solutions obtained by the ILP model and the greedy heuristic are affected. In the experiment, we ignored the solution that did converge in the ILP model. In both cases, because there were insufficient resources, some disconnected PMUs needed to reroute to PDCs distributed at different substations; that increased the number of forwarding rules added to switches. Compared to the results shown in Fig. 5 (when 8 PDCs were compromised), the solutions obtained from the ILP model at Stage 1 can increase by more than three times due to the increased number of forwarding rules in switches. The heuristic algorithm and the baseline method perform worse and give solutions which increase by more than six times.

As shown in Fig. 7 and Fig. 8, when hardware resources, i.e., rule spaces in switches or connection spaces in PDCs, become limited, the greedy heuristic algorithm performs better in the 118-bus system than in the 30-bus system. In the experiments, we limited the rule spaces in each switch or the connection spaces in each PDC. Because the 118-bus system include more switches and PDCs, it can provide more hardware resources for the greedy heuristic algorithm to use.

As shown in Fig. 9 and Fig. 10, the limited resources in network switches and PDCs have two impacts on the execution time. On the one hand, they reduce the search space. In the ILP problem, the search space specified by constraints (12)–(14), and (16)–(18) is reduced. The greedy heuristic algorithm stops adding rules into the switches at *Steps 17–24*, if there is insufficient rule space in a path. On the other hand, because of limited resources, it can take



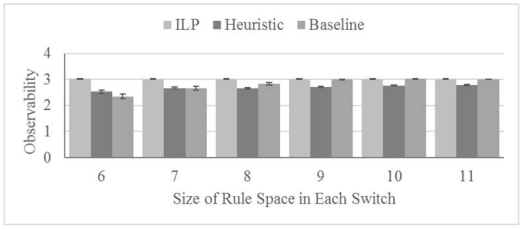
(a) IEEE 30-bus system



(b) IEEE 118-bus system

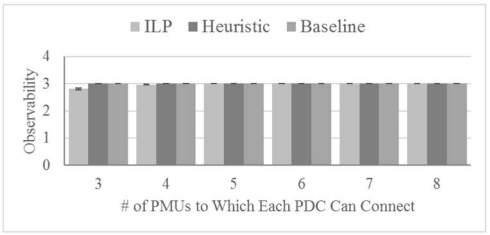
Fig. 10. Comparison of the execution times with impact of limited PDC connection space.

(a) IEEE 30-bus system

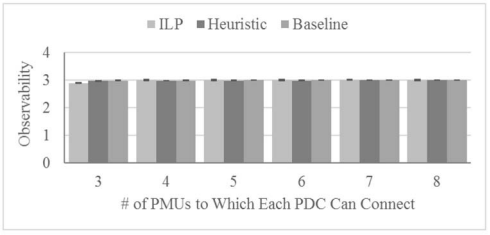


(b) IEEE 118-bus system

Fig. 11. Comparison of power system observability with the impact of limited network resources.



(a) IEEE 30-bus system



(b) IEEE 118-bus system

Fig. 12. Comparison of power system observability with the impact of limited PDC connection space.

more time to find a solution in both methods. When they are impacted by these two factors simultaneously, the execution times of the ILP model, the greedy heuristic algorithm, and the baseline method fluctuates without increasing or decreasing dramatically.

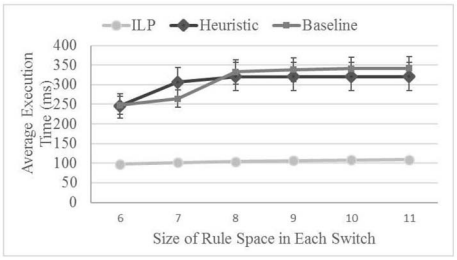
An interesting phenomenon happens in the 118-bus system when PDCs are short of connection spaces (shown in Fig. 10(b)); the greedy heuristic algorithm spends more time than the baseline method to obtain the result. The reason is that even though the heuristic algorithm reconnects PMUs in the order of their degrees in the power system's transmission networks, the limited connection space in PDCs mean that the algorithm must spend a lot of times searching possible paths for reconnection before it tries the next disconnected PMU.

C. Performance Evaluation of Stage 2

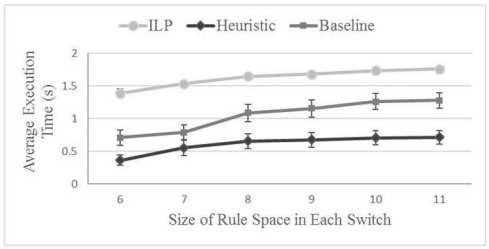
We further evaluate the second stage of the optimization, which reconnects PMUs to maximize the redundancy of measurements. In this evaluation, we consider the case when the power grid is still observable with the disconnected PMUs. When there are sufficient hardware resources, the ILP model, the greedy heuristic algorithm, and the baseline algorithm always reach the same result, i.e., all remaining uncompromised PMUs are reconnected.

Fig. 11 and Fig. 12 show how the limited rule space of network switches and the connection spaces of PDCs can impact the result of the self-healing mechanism. As shown in Fig. 11, the greedy heuristic algorithm and the baseline method are affected if there are a limited number of rule spaces in switches. Both the greedy heuristic and the baseline algorithm reconnect PMUs one by one. The algorithm can use up the rule space of a small number of switches; those switches can become bottlenecks that prevent other PMUs from reaching unused PDCs. The ILP model tries to optimize global reconnection of PMUs among all paths; it can avoid selecting paths that tend to use up the rule space of a few switches. As seen in Fig. 12, the ILP model, the greedy heuristic, and the baseline method usually give the same results, i.e., reconnect PMUs based on the existing connection spaces from all PDCs.

Fig. 13 and Fig. 14 show the execution times of the ILP model, the greedy heuristic algorithm, and the baseline method. Like Fig. 6, Fig. 9, and Fig. 10, Fig. 13 and Fig. 14 include the average of the 50 largest execution times spent by these three methods. By comparing Fig. 9 and Fig. 10 with Fig. 13 and Fig. 14, we can see that the execution times of the ILP model spent at Stage 2 is one order of magnitude smaller than the time spent at Stage 1. In other words, the ILP model uses a large amount of computation to satisfy constraints (3). However, the execution times spent by the greedy and baseline algorithms at Stage 2 are one magnitude larger than their times

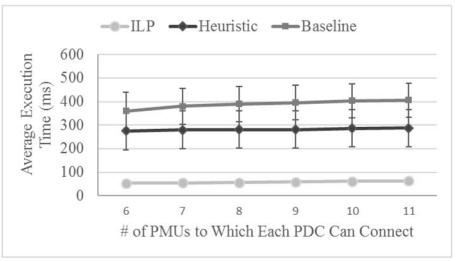


(a) IEEE 30-bus system

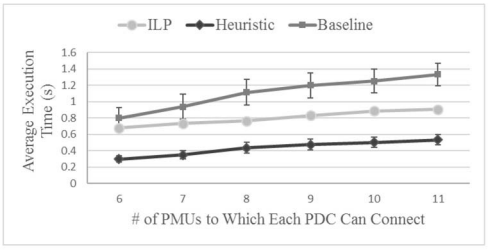


(b) IEEE 118-bus system

Fig. 13. Comparison of the execution times with the impact of limited network resources.



(a) IEEE 30-bus system



(b) IEEE 118-bus system

Fig. 14. Comparison of the execution times with the impact of limited PDC connection space.

spent at stage 1, as they often end up attempting to reconnect many PMUs.

The execution times of the ILP model does not change dramatically when rule space of network switches or the connection space of PDCs become limited. However, the greedy algorithm and the baseline method select PMUs in sequence. They can reconnect more PMUs if more hardware resources are available, and thus need more execution times to finish. Also, the greedy algorithm and the baseline method expect more variation in execution times. Because in both approaches, the search space of a PMU can be varied with the location of the disconnected PMUs when hardware resources become limited.

VI. Conclusion

In this paper, we have presented an innovative, self-healing mechanism for mitigation of cyber-attacks and recovery of power system observability on a PMU network. After a cyber-attack, the mechanism works by changing the configuration of the network switches enabled by SDN technology, so that the compromised PMUs/PDCs are isolated to prevent further propagation of the attack while the uncompromised PMUs are reconnected to the network to self-heal and therefore restore the observability of the power system. Specifically, ILP models are formulated to minimize the overhead of the self-healing process while considering the constraints of power system observability, hardware resources, and network topology. The proposed greedy heuristic algorithm reduces computational complexity. Experiments conducted on a PMU network over IEEE 30-bus and IEEE 118-bus systems validate the effectiveness of the proposed methods.

References

[1] P. Zhang, "Phasor measurement unit (PMU) implementation and applications," Elect. Power Res. Inst., Palo Alto, CA, USA, Tech. Rep. 1015511, Oct. 2007.

[2] J. Qi, K. Sun, J. Wang, and H. Liu, "Dynamic state estimation for multi-machine power system by unscented Kalman filter with enhanced numerical stability," *IEEE Trans. Smart Grid*, to be published.

[3] J. Qi, K. Sun, and W. Kang, "Optimal PMU placement for power system dynamic state estimation by using empirical observability Gramian," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 2041–2054, Jul. 2015.

[4] K. Sun, J. Qi, and W. Kang, "Power system observability and dynamic state estimation for stability monitoring using synchrophasor measurements," *Control Eng. Pract.*, vol. 53, pp. 160–172, Aug. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066116300132

[5] NASPI, "Data bus technical specifications for North American Synchro-Phasor initiative network (NASPInet)," U.S. Dept. Energy, Quanta Technol., Raleigh, NC, USA, 2009. [Online]. Available: https://www.naspi.org/File.aspx?fileID=542

[6] T. Morris *et al.*, "Cybersecurity risk testing of substation phasor measurement units and phasor data concentrators," in *Proc. 7th Annu. Workshop Cyber Security Inf. Intell. Res. (CSIIRW)*, Oak Ridge, TN, USA, 2011, p. 24.

[7] C. Beasley, G. K. Venayagamoorthy, and R. Brooks, "Cyber security evaluation of synchrophasors in a power system," in *Proc. Power Syst. Conf. (PSC)*, Clemson, SC, USA, 2014, pp. 1–5.

[8] S. Mousavian, J. Valenzuela, and J. Wang, "A probabilistic risk mitigation model for cyber-attacks to PMU networks," *IEEE Trans. Power Syst.*, vol. 30, no. 1, pp. 156–165, Jan. 2015.

[9] J. Stewart, T. Maufer, R. Smith, C. Anderson, and E. Ersonmez, "Synchrophasor security practices," Presented at the 14th Annu. Georgia Tech Fault Disturb. Anal. Conf., Atlanta, GA, USA, May 2011.

[10] X. Dong, H. Lin, R. Tan, R. K. Iyer, and Z. Kalbarczyk, "Software-defined networking for smart grid resilience: Opportunities and challenges," in *Proc. 1st ACM Workshop Cyber Phys. Syst. Security (CPSS)*, Singapore, 2015, pp. 61–68.

[11] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proc. ACM Conf. Comput. Commun. Security (CCS)*, Chicago, IL, USA, 2009, pp. 21–32.

[12] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 645–658, Dec. 2011.

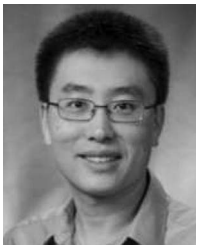
[13] A. Lee, "Electric sector failure scenarios and impact analyses," Elect. Power Res. Inst., Palo Alto, CA, USA. [Online]. Available: http://smartgrid.epri.com/doc/NESCOR%20failure%20scenarios09-13%20finalc.pdf

[14] A. Goodney, S. Kumar, A. Ravi, and Y. H. Cho, "Efficient PMU networking with software defined networks," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Vancouver, BC, Canada, 2013, pp. 378–383.

[15] K. Murakami and H. S. Kim, "Comparative study on restoration schemes of survivable ATM networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Kobe, Japan, 1997, pp. 345–352.
[16] R. Kawamura, K.-I. Sato, and I. Tokizawa, "Self-healing ATM networks based on virtual path concept," *IEEE J. Sel. Areas Commun.*, vol. 12, no. 1, pp. 120–127, Jan. 1994.
[17] T. Frisanco, "Optimal spare capacity design for various protection switching methods in ATM networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Montreal, QC, Canada, 1997, pp. 293–298.
[18] R. Doverspike and B. Wilson, "Comparison of capacity efficiency of DCS network restoration routing techniques," *J. Netw. Syst. Manag.*, vol. 2, no. 2, pp. 95–123, 1994.
[19] K. Murakami and H. S. Kim, "Near-optimal virtual path routing for survivable ATM networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, 1994, pp. 208–215.
[20] R. R. Iraschko, M. H. MacGregor, and W. D. Grover, "Optimal capacity placement for path restoration in STM or ATM mesh-survivable networks," *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, pp. 325–336, Jun. 1998.
[21] S. Ponomarev and T. Atkison, "Industrial control system network intrusion detection by telemetry analysis," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 2, pp. 252–260, Mar./Apr. 2016.
[22] C.-W. Ten, G. Manimaran, and C.-C. Liu, "Cybersecurity for critical infrastructures: Attack and defense modeling," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 4, pp. 853–865, Jul. 2010.
[23] Y. Yang *et al.*, "Multiattribute SCADA-specific intrusion detection system for power networks," *IEEE Trans. Power Del.*, vol. 29, no. 3, pp. 1092–1102, Jun. 2014.
[24] S. Cui *et al.*, "Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions," *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 106–115, Sep. 2012.
[25] A. F. Taha, J. Qi, J. Wang, and J. H. Panchal, "Risk mitigation for dynamic state estimation against cyber attacks and unknown inputs," *IEEE Trans. Smart Grid*, to be published.
[26] B. Stephenson and B. Sikdar, "A quasi-species model for the propagation and containment of polymorphic worms," *IEEE Trans. Comput.*, vol. 58, no. 9, pp. 1289–1296, Sep. 2009.
[27] B. Sun, G. Yan, Y. Xiao, and T. A. Yang, "Self-propagating mal-packets in wireless sensor networks: Dynamics and defense implications," *Ad Hoc Netw.*, vol. 7, no. 8, pp. 1489–1500, 2009.
[28] P. Mell, K. Kent, and J. Nusbaum, "Guide to malware incident prevention and handling," U.S. Dept. Commer., Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NIST SP800-83, 2005. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-83/SP800-83.pdf
[29] M. Altunay, S. Leyffer, J. T. Linderoth, and Z. Xie, "Optimal security response to attacks on open science grids," Argonne Nat. Lab., Lemont, IL, USA, Tech. Rep. ANL/MCS-P1593-0309, 2009. [Online]. Available: http://www.mcs.anl.gov/papers/P1593.pdf
[30] N. Kang, Z. Liu, J. Rexford, and D. Walker, "Optimizing the 'one big switch' abstraction in software-defined networks," in *Proc. 9th ACM Conf. Emerg. Netw. Exp. Technol. (Conext)*, Santa Barbara, CA, USA, 2013, pp. 13–24.
[31] S. Azizi, A. S. Dobakhshari, S. A. N. Sarmadi, and A. M. Ranjbar, "Optimal PMU placement by an equivalent linear formulation for exhaustive search," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 174–182, Mar. 2012.
[32] "SEL-3373 station phasor data concentrator, instruction manual," Schweitzer Eng. Lab., Pullman, WA, USA, 2014. [Online]. Available: https://selinc.com/products/3373/
[33] D. L. Applegate *et al.*, "Compressing rectilinear pictures and minimizing access control lists," in *Proc. ACM SIAM Symp. Discrete Algorithms (SODA)*, New Orleans, LA, USA, 2007, pp. 1066–1075.
[34] C. R. Meiners, A. X. Liu, and E. Torng, "TCAM Razor: A systematic approach towards minimizing packet classifiers in TCAMs," in *Proc. IEEE Int. Conf. Netw. Protocols*, Beijing, China, Oct. 2007, pp. 266–275.
[35] M. Moshref, M. Yu, A. Sharma, and R. Govindan, "vCRIB: Virtualized rule management in the cloud," in *Proc. 4th USENIX Workshop Hot Topics Cloud Comput. (HotCloud)*, Boston, MA, USA, 2012, p. 23. [Online]. Available: https://www.usenix.org/conference/hotcloud12/vcrib-virtualized-rule-management-cloud
[36] Y. Kanizo, D. Hay, and I. Keslassy, "Palette: Distributing tables in software-defined networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Turin, Italy, Apr. 2013, pp. 545–549.
[37] C.-Y. Hong *et al.*, "Achieving high utilization with software-driven wan," in *Proc. ACM SIGCOMM*, Hong Kong, 2013, pp. 15–26. [Online]. Available: http://doi.acm.org/10.1145/2486001.2486012
[38] F. Gori, G. Folino, M. S. Jetten, and E. Marchiori, "MTR: Taxonomic annotation of short metagenomic reads using clustering at multiple taxonomic ranks," *Bioinformatics*, vol. 27, no. 2, pp. 196–203, 2011.
[39] V. Chvatal, "A greedy heuristic for the set-covering problem," *Math. Oper. Res.*, vol. 4, no. 3, pp. 233–235, 1979.
[40] J. Currie, D. I. Wilson, and B. T. Young. (Mar. 2015). *OPTI Toolbox: A Free MATLAB Toolbox for Optimization*. [Online]. Available: http://www.i2c2.aut.ac.nz/Wiki/OPTI/index.php/Main/HomePage

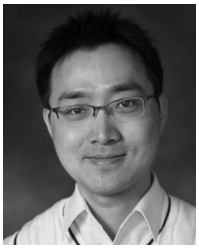
Hui Lin received the B.S. degree in electrical and computer engineering from the Huazhong University of Science and Technology in 2006, and the M.S. degree in electrical and computer engineering from the University of Illinois at Chicago in 2010. He is currently pursuing the Ph.D. degree with the University of Illinois at Urbana–Champaign. His research interests include cyber security, intrusion detection systems, and software-defined networking (SDN). His Ph.D. research explores applying intrusion detection systems and SDN in critical cyber-physical systems, such as power grids, to increase their resilience against cyber attacks, and accidental failures. His current work focuses on applying SDN in cyber-physical systems; he intends to use SDNs network programmability to design flexible cyber-physical systems which can quickly respond to cyber-attacks and accidents.

Chen Chen (M'13) received the B.S. and M.S. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2006 and 2009, respectively, and the Ph.D. degree in electrical engineering from Lehigh University, Bethlehem, PA, USA, in 2013. From 2013 to 2015, he was a Post-Doctoral Researcher with the Energy Systems Division, Argonne National Laboratory, Argonne, IL, USA, where he is currently a Computational Engineer. His primary research is in optimization, communications and signal processing for smart electric power systems, cyber-physical system modeling for smart grids, and power system resilience.



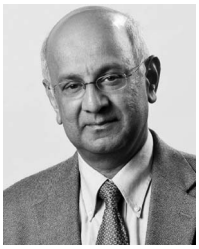
Jianhui Wang (M'07–SM'12) received the Ph.D. degree in electrical engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2007. He is currently the Section Lead for Advanced Power Grid Modeling, Energy Systems Division, Argonne National Laboratory, Argonne, IL, USA. He is also an Affiliated Professor with Auburn University and an Adjunct Professor with the University of Notre Dame. He has held visiting positions in Europe, Australia, and Hong Kong, including a VELUX Visiting Professorship with the Technical University of Denmark. He was a recipient of the IEEE PES Power System Operation Committee Prize Paper Award in 2015. He is the Secretary of the IEEE Power and Energy Society (PES) Power System Operations Committee. He is an Associate Editor of the *Journal of Energy Engineering* and an Editorial Board Member of *Applied Energy*. He is an Editor-in-Chief of the IEEE TRANSACTIONS ON SMART GRID and the IEEE PES Distinguished Lecturer.

Junjian Qi (S'12–M'13) received the B.E. degree in electrical engineering from Shandong University, Jinan, China, in 2008, and the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 2013. In 2012, he was a Visiting Scholar with Iowa State University, Ames, IA, USA. From 2013 to 2015, he was a Research Associate with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA. He is currently a Post-Doctoral Appointee with the Energy Systems Division, Argonne National Laboratory, Argonne, IL, USA. His research interests include cascading blackouts, power system dynamics, state estimation, synchrophasors, and cybersecurity.



Dong (Kevin) Jin received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign in 2013. He is an Assistant Professor with the Computer Science Department, Illinois Institute of Technology. His research interests include trustworthy cyber-physical critical infrastructures, simulation modeling and analysis, software-defined networking, and cyber-security. He was a recipient of the best paper awards at the ACM SIGSIM Conference on Principles of Advanced and Distributed Simulation in 2012 and 2015.

Zbigniew T. Kalbarczyk is a Research Professor with the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign. He has published over 130 technical papers and is regularly invited to give tutorials and lectures on issues related to design and assessment of complex computing systems. His research interests are in the area of design and validation of reliable and secure computing systems. His current work explores emerging technologies, such as resource virtualization to provide redundancy and assure system resiliency to accidental errors and malicious attacks. His research also involves analysis of data on failures and security attacks in large computing systems, and development of techniques for automated validation and benchmarking of dependable and secure computing systems using formal (e.g., model checking) and experimental methods (e.g., fault/attack injection). He served as the Program Chair for the International Conference on Dependable Systems and Networks in 2002 and 2007. He is an Associate Editor of the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. He is a member of the IEEE Computer Society, and the IFIP Working Group 10.4 on Dependable Computing and Fault Tolerance.



Ravishankar K. Iyer is the George and Ann Fisher Distinguished Professor of Engineering, University of Illinois at Urbana–Champaign. He holds appointments with the Department of Electrical and Computer Engineering, Coordinated Science Laboratory, and the Department of Computer Science, serves as a Chief Scientist of the Information Trust Institute, and is an Affiliated Faculty of the National Center for Supercomputing Applications. He currently co-leads the CompGen Center at IL, USA. He was a recipient of several awards, including the American Institute for Aeronautics and Astronautics Information Systems Award, the IEEE Emanuel R. Piore Award, the 2011 Outstanding Contributions award by the Association of Computing Machinerys Special Interest Group on Security, and the Degree of Doctor Honaris Causa from Toulouse Sabatier University in France. He is a fellow of the American Association for the Advancement of Science and ACM.