This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2014.2364928, IEEE Transactions on Emerging Topics in Computing

Distributed Software Emulator for Cyber-Physical Analysis in Smart Grid

Song Tan WenZhan Song Dan Huang Qifen Dong Lang Tong

Abstract—A Smart Grid is a highly complex cyber-physical electrical power system that uses two-way digital communication and intelligent embedded devices to achieve sensing, control, computation and communication within power network. To validate the functionality, security and reliability of such an system requires the modeling and emulation of both communication network and power network, as well as the interactions between them. In this paper, we present Smart-Grid Common Open Research Emulator (SCORE), a distributed software emulator for cyber-physical analysis in Smart Grid. SCORE integrates the emulations of both power network and communication network, and it is highlighted by the following features: firstly, SCORE is the **first software emulation** platform for Smart Grid, which means that the same application program running in SCORE can be directly ported to embedded devices with little or no migration issues. Secondly, for one Smart Grid instance, SCORE supports **distributed** emulation when the instance is in very large scale. Thirdly, for multiple Smart Grid emulation instances running on different networked computers, SCORE allows them to **dynamically** connect or disconnect with each other in run time, such that each instance can capture its own interior system dynamics even without a prior knowledge of the entire Smart Grid topology.

Index Terms—Cyber-Physical System, Distributed Emulation, Smart Grid

1 Introduction

Smart Grid is a highly complex cyber-physical system envisioned to modernize the power network by leveraging the two-way communications of the data and information from the deployed embedded devices. Integrating the power system closely with communication system and intelligent control system, Smart Grid promotes customers' choice by enabling them to manage their energy usage and choose the most economically efficient offering. It also improves the safety of the power system by using automation and alternative resources to maintain delivery system reliability and stability. Moreover, through integrating renewable, storage, and generation alternatives, it brings sustainable and environment-friendly energy to the society.

The smart grid innovation brings many new challenges and research issues to cyber-physical analysis, such as demand response algorithms, routing protocols in Advanced Metering Infrastructure (AMI), market policy programs, and countermeasures against cyber attacks, etc. However, due to the system complexity, reliability and integrity requirements of smart grid, especially the high cohesion of communication and power system, validating, verifying or demonstrating those critical Smart Grid applications are not easy tasks. It is highly desirable in the Smart Grid research community to have a reliable and efficient platform to tackle this issue, especially in a lab environment.

In this paper, we present Smart-Grid Common Open Research Emulator (SCORE). It is built upon CORE [1], an open source communication network emulator from Naval Research Laboratory. SCORE emulates both communication networks and power networks and their interactions. The same Smart Grid application program running in SCORE can be directly ported to embedded devices in a real Smart Grid with little or no modification. Specifically, SCORE highlights itself by the following features:

- SCORE is a **software emulation** platform within which the Smart Grid applications execute inside each emulated virtual node. Each virtual node is a light weighted Linux virtual machine such that the tested Smart Grid applications, such as demand response algorithms, routing protocols, and market policy programs, and countermeasures against cyber attacks etc, can be directly ported from SCORE platform to real Smart Grid embedded devices with no or little migration issue.
- For a given Smart Grid instance, SCORE supports **distributed** emulation using general PCs, providing the capability to deal with large-scale Smart Grid instances.
- For multiple Smart Grid emulation instances running on different networked computers, SCORE allows them to **dynamically** connect or disconnect

- *Song Tan, WenZhan Song, Dan Huang are with Sensorweb Research Laboratory, Department of Computer Science, Georgia State University Email: stan,songwz,dhuang@cs.gsu.edu*
- *Qifen Dong is with College of Information Engineering, Zhejiang University of Technology, Zhejiang, China Email: qdong@cs.gsu.edu*
- *Lang Tong is with School of Electrical and Computer Engineering, Cornell University Email: ltong@ece.cornell.edu*

. Our research is supported by NSF-CPS-1135814, and partial results have been published in SmartGridComm2012.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2014.2364928, IEEE Transactions on Emerging Topics in Computing

with each other in run time when the interfaces between each other are specified. Each instance can capture its own interior system dynamics even without a prior knowledge of the entire Smart Grid topology.

2 Related work

Creating test platform for cyber-physical analysis in Smart Grid is challenging and it has been studied for years. From our literature reviews, the approaches to solve this issue generally fall into two categories: real hardware testbed and software simulation.

2.1 Real hardware testbed approach

Real hardware testbeds are further classified into two categories: flat-out hardware platforms and hardware-in-the-loop platforms.

2.1.1 Flat-out hardware platform

Flat-out hardware platforms are the ones consisting of pure hardware devices. The Korean government selected the whole Jeju Island to build the Smart Grid testbed to allow the testing of Smart Grid technologies and business models [2]. Renewable Energy Laboratory in Greece set up a central-controlled microgrid testbed consisting of PV-panels, battery banks and inverters to investigate the proposed Smart Grid topologies [3]. Sensorweb Reserach Laboratory from Georgia State University designed SmartGridLab testbed to test the distributed demand response algorithm. It consists of intelligent power switch, power generator, renewable energy sources, smart appliances, and power meter [4].

2.1.2 Hardware-in-the-loop platform

Hardware-in-the-loop platform mixed hardware devices with software simulators to achieve the cyber physical analysis of Smart Grid. Stanovich *et al.* in [5] integrates hardware from energy field, such as Remote Terminal Unit (RTU), fiber optical cables within the testbed. Hahn *et al.* in [6] employs devices like Programmable Logic Units (PLUs) and Intelligent Electronic Devices (IEDs) for communication networks and Real-Time Digtial Simulators for power network simulation.

2.2 Software simulation approach

The software simulation tools for Smart Grid analysis can be further classified into two categories: individual simulation platforms and co-simulation platforms.

2.2.1 Individual simulation platforms

Individual simulation platforms are those which encapsulate the simulation features for Smart Grid into one entity. In other words, it is one single simulator to complete the job. These platforms usually focus on one particular aspect of interests for Smart Grid. In 2008, Guo *et al.* designed and developed an energy demand management simulator (EDMS) to predict the response from different deployment strategies of distributed domestic energy management [7]. A self-adaptive demand management strategy is simulated and analyzed within this platform. In 2009, Molderink *et al.* built a simulation environment from scratch to analyze control algorithms for energy efficiency [8]. Micro-generators, energy buffers and appliances are modeled and different energy streams like heat and gas are considered. In 2010, Faria *et al.* presents Demsi, a simulator for demand response in the context of competitive electricity markets and intensive use of distributed generation [9]. Demis is extended from power system analysis tool PSCAD [10]. Energy service provider and demand side player are modeled and strategic decisions are supported. In 2011, Narayan *et al.* presents GridSpice [11], a cloud-based simulation package for Smart Grid. Leveraging the powerful component of Gridlab-D [12] and Matpower [13], GridSpice is being developed iteratively with an ultimate goal of modeling the interactions between all parts of the electrical network, including generation, transmission, distribution, storage and loads. Currently, GridSpice includes the ability to perform distribution simulations along with one-shot optimal power flow simulations and demand response features. All the individual software platforms can complete their own tasks in the specific application domain, but they all just concentrate on the power network simulation. The Communication network, as another critical component of Smart Grid is not considered in these platforms. This is why the co-simulation platform comes to the picture.

2.2.2 Co-simulation platforms

Co-simulation (co-operative simulation) is a simulation methodology that allows individual components to be simulated by different simulation tools running simultaneously and exchanging information in a collaborative manner [14]. In [15], Hopkinson *et al.* introduce a federated simulation combing NS2 [16], a discrete event network simulator with PSCAD [10], a continuous time power network simulator. In [17], Godfrey *et al.* simulate the Smart Grid using NS2 and OpenDSS [18], a power network simulator. In [19], Mallouhi *et al.* set up a co-simulation testbed specifically for security analysis of SCADA system by employing PowerWorld simulator and OPNET. In [20] and [21], Lin *et al.* introduces a global event queue to synchronize NS2 and PSLF [22]. The co-simulation approach typically needs iteratively running separate electrical and communication network simulation. The performance is affected by putting extra overhead of an intermediary of synchronization. Meanwhile, the interaction between communications and power system models is usually restricted to fixed synchronization interval. Mismatches can occur between the real dynamics and the simulated one, which exposes reliability issues of such systems. An improvement about this issue is to integrate one simulation component into the other. In [23], electric network is made into a component within OMNET++ [24], a network simulator. In [25], the adevs simulation tools are integrated into NS2 to provide a hybrid modeling of the continuous time power system

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2014.2364928, IEEE Transactions on Emerging Topics in Computing

and discrete event communication system through the discretization of the continuous power dynamics.

2.3 Remarks about related work

From the above discussion, we can see the characteristics of the real hardware testbed approach and the software simulation approach for cyber-physical analysis in Smart Grid.

The real hardware testbed approach achieves high fidelity by employing dedicated devices as part of the testbeds. The critical control programs, such as demand response algorithms, routing protocols, and security strategies, can be tested in real hardware testbeds and they could be directly migrated to the actual Smart Grid embedded devices. However, the problems with the real hardware testbed approach is that since the dedicated and specialized hardware are the integral parts of the testbeds, they cannot be easily accessed and used by the public research community and difficult to be scaled when the test case becomes quite large.

The software simulation approach, on the other hand, achieves better scalability and can be easily accessed and distributed. The software simulation tools typically run on a single PC and abstract the operating system, communication protocols and power dynamics into mathematical simulation models to produce overall statistical analysis. In other words, they just duplicate the behaviours of the Smart Grid system but not the execution environment. Therefore, the critical control programs of Smart Grid applications either cannot be tested or can be tested but cannot be migrated to physical Smart Grid devices directly.

2.4 Our approach: software emulation

SCORE bridges the gaps between real hardware approach and software simulation approach. The key advantages of SCORE are:

- First, software emulation achieves high fidelity by duplicating the execution environment, such that the programs running in the emulation platform can be directly ported to the embedded devices as firmware.
- Second, SCORE enables distributed emulation feature such that very large scale test case can also be supported.
- Finally, SCORE supports dynamic connection and disconnection between multiple Smart Grid emulation instances in real time. The significance of this feature is two folds. First, when users from multiple parties in different locations want to conduct the integration testing together, but want to preserve the privacy of power and communication networks configurations, this feature would make it happen without requiring explicit synchronization from all parties. Second, our in-progress research is integrating SCORE with real hardware testbeds. In this case, the dynamic feature would be necessary since we want to give the freedom of SCORE to dynamically connect and disconnect with the testbed at any time.

SCORE also has its own limits: the power network model is static DC power flow model such that we cannot use SCORE to capture transient power dynamics or renewable grid integrations, etc. The strengths and limitations of our approach compared with related works are listed in Table 1.

TABLE 1
Summary of features for Real hardware testbed, Software Simulation and SCORE

	Real hardware testbed	Software Simulation	SCORE
Accessibility	Difficult	Easy	Easy
Scalability	Low	High	High
Code migration	Yes	No	Yes
Time step	Real time	Real time/discrete time	Real time
Transient analysis of power system	Yes	Yes	No
Renewable integration	Yes	No	No

Our previous conference paper in [26] has draw great attention in research community and our open source release now has more than one hundred downloads from http://sourceforge.net/projects/score-sensorweb/.In this paper, we make substantial extensions by:

- Presenting implementation details of GUI, services layer, lighted weighted virtual nodes, communication module, power module.
- Adding domain decomposition based algorithms and implementations for dynamic connect and disconnect feature.
- Developing energy model programming API library.
- Conducting extensive survey of related work.

3 System Design and Implementation

3.1 Overview

Our design of SCORE takes advantage of CORE's structure. Figure 1 provides an abstract overview of SCORE's architecture and our integration approach. As shown, SCORE consists of GUI, Service Layer, Communication Module and Power Module.

3.2 GUI

The SCORE GUI is built using Tcl/Tk. The Tk toolkit provides almost sufficient widgets for all the X window system interface needs. The Tcl/Tk GUI provides an easily drag-and-draw canvas with various Smart Grid devices (Appliances, Solar Panel, and Wind Turbine,etc), which can be placed and linked together with communication links or power lines. Also, the communication interfaces, power interfaces and energy model parameters of each node can be configured. During emulation execution, a bash shell can be popped out when double clicking the selected node. Users can navigate the local file system or execute bash script through the interactive shell window. Distributed emulation can be conducted by assigning a selection of nodes to another emulation server in GUI. The message broker in Service Layer is used to forward messages from the GUI to the appropriate emulation server.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2014.2364928, IEEE Transactions on Emerging Topics in Computing

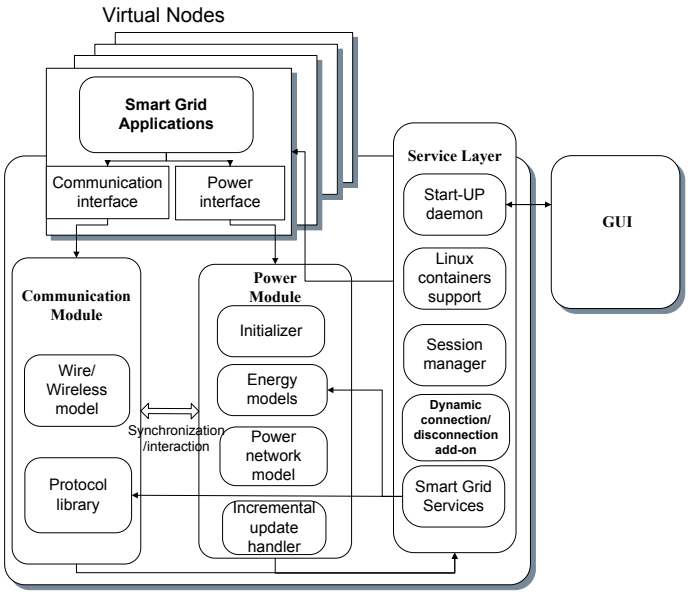


Fig. 1. SCORE Architecture

3.3 Service Layer

The Service Layer are python frameworks that is responsible for creating sessions, instantiating the virtual nodes, communication and power interfaces, communication links and power lines, based on the GUI input or a batch-mode configuration file. Note that start-up daemon in service layer interacts with GUI using a TCP socket-based API such that the emulation can run on a different machine with the GUI or even without a GUI. Pre-defined energy models and communication protocols, which are usually daemonized in the Linux operating system of emulation server, are all wrapped as Smart Grid services in this layer. These communication and energy services can all be employed to develop various Smart Grid Applications. Users are also allowed to add their own customized services to SCORE by providing their own implementations. Last but not the least, as our newly developed feature, the Dynamic connection/disconnection add-on in service layer can process the dynamic emulation request by employing the dynamic features in communication modules and power modules.

3.4 Light weighted virtualization

The emulation feature of SCORE are implemented using Linux namespace techniques, which is a recent light weighted paravirtualization technique supported by mainstream Linux kernel. Different from the typical virtual machines techniques like the ones in VMware or Virtual Box, each emulated virtual node in SCORE only has its separated copy of network interface, protocol stack and process control group. Other resources like operating system and local file system, are shared by all the virtual nodes. This light-weighted virtualization approach enables the scalability of SCORE. Moreover, from the perspective of codes running inside the virtual node, each emulated node is just another piece of hardware platform controlled by Linux OS, which equips SCORE with the property of portability that the emulated node can execute unmodified Smart Grid application codes running inside a real physical Linux-running hardware devices, and vice versa.

3.5 Communication Module

The communication module in SCORE leverages the comprehensive support of various wired and wireless communication network models and protocols from CORE. Each emulated node has its own instance of OS implemented TCP/IP stack supported by Linux namespaces, from the perspective of Open Systems Interconnection (OSI) model, thus SCORE allows high fidelity emulation of network layer and above. By default, a simplified simulation of link and physical layers is enabled, which is implemented using netem with Ethernet bridging in Linux. Statistical network effects such as bandwidth, loss rate, bit error rate and noise level can be configured and applied. For higher-fidelity link and physical layer emulation, other network simulation tools, such as EMANE [1], can be easily integrated. In addition, the virtualized Ethernet interface can be directly mapped to a physical Ethernet interface on the emulation host, such that all the traffic passing through that physical port can be transferred to the emulation environment, allowing real time communication between the virtual nodes inside a running emulation and external physical networks. By using the virtualized interfaces on each emulation host, the communication network emulated on different hosts can directly connect with each other in run time, which enables the dynamic emulation of the communication networks. Meanwhile, we employ this feature to enable the interactions and synchronization between the communication module and the power module discussed in the next section. The idea is that the power module is running on a host physically in the same network with the communication emulation host so that the power module can receive and react to the queued-up messages sent by all the emulated virtual node in real time.

3.6 Power Module

The power module in SCORE emulates the power flows analysis within Smart Grid and also provides implementations of pre-defined energy models. We use current model in circuit theory to emulate the real power flow within power network. Figure 2 shows the data flow diagram of power module. The power module receives initial power network topology, energy model configuration information and the dynamic connection/disconnection request from service layer to formulate the power network model, which will be introduced later. The model is updated when the corresponding model updates are received from the interactions with communication module.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2014.2364928, IEEE Transactions on Emerging Topics in Computing

The power network module of SCORE is highlighted by the following features:

- First, SCORE adopt incremental model updating in computation to react more efficiently to the system status changes.
- Second, as the size and order of the power network increase, distributed computation for power network becomes a necessity for efficient Smart Grid emulation. SCORE highlights itself in scalability by enabling the user to conduct the emulation in a distributed way when a single PC cannot provide enough computation capabilities. We choose to split the power network model into several subdomains and let each subdomains be computed and updated separately in parallel. With appropriate coordinating among those separate computing and updating processes, the merged result of power flows in Smart Grid is solid without any loss of precision compared with centralized computation.
- Finally, by only specifying the interfaces between each power network, SCORE allow dynamic connections and disconnections of multiple Smart Grid instances running on different hosts. The significance of this feature are two-folded: first, in the situation when each user is reluctant to expose their own Smart Grid topology details to others, they can still conduct the combined emulation with each other to see the impact of external networks on their own network. Privacy is protected while Smart Grid analysis is conducted. Second, even though we have not implemented in our platform, by adding an appropriate hardware interface serving as the gateway, this feature enables SCORE to communicate and exchange energy directly with testbeds or even the real power grid. This cyber-physical interaction is very valuable to the scalable and reliable testings of Smart Grid technologies [27].

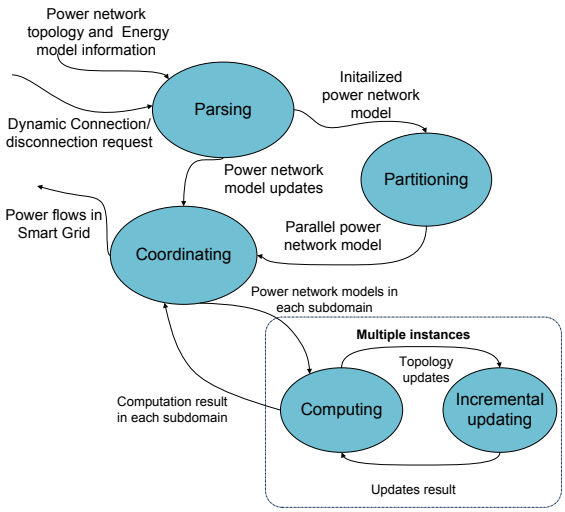


Fig. 2. Data flow diagram of Power Module

3.6.1 Power network model

Now let us introduce the power network model. Assume a power grid is composed of n nodes and b branches. Since the power network dynamics is subject to Kirchhoff's current and voltage laws (KCL and KVL), in order to calculate the voltages of all nodes, we apply nodal analysis to the grid and get the linear equations for the whole system:

$$AV = I \tag{1}$$

where coefficient matrix A is the $(n-1) \times (n-1)$ reduced nodal admittance matrix since we have chosen a reference node. Let $Nbr(i)$ represents the neighbor set of node i in the power network, we get:

$$a_{ij} = \begin{cases} \sum\limits_{s \in Nbr(i)} g_{is} & i = j. \\ -g_{ij} & j \in Nbr(i) \\ 0 & otherwise \end{cases} \tag{2}$$

g_{ij} is the admittance between node i and node j, V and I are the unknown node voltage vector and the known nodal current injection vector, respectively.

3.6.2 Incremental updating

Based on previous model, let's consider the situation when the power network topology changes. Suppose the power grid status changes, such as the admittance between node i and node j is changed by Δg_{ij}. So the new coefficient matrix $\widetilde{A}$ can be written as:

$$\widetilde{A} = A + U \Delta g_{ij} U^T \tag{3}$$

where

$$U = \begin{bmatrix} 0 & \cdots & \underset{i}{1} & \cdots & \underset{j}{-1} & \cdots & 0 \end{bmatrix}^T$$

Particularly, the changed admittance Δg_{ij} equals to $-g_{ij}$ when the branch is removed and $\Delta g_{ij} = g_{ij}$ when a new branch is added. Notice that [28]

$$\widetilde{A}^{-1} = A^{-1} - A^{-1} U \left({\Delta g_{ij}}^{-1} + U^T A^{-1} U \right)^{-1} U^T A^{-1} \tag{4}$$

then we can get the $\widetilde{A}^{-1}$ with a much lower computation cost when we store previously computed A^{-1}.

3.6.3 Power network model: domain decomposition

Power network is naturally suitable for emulation in distributed paradigms, since due to the economy and geographical reasons, power network is generally a network of loosely coupled sub power networks. Each sub network is a relatively independent partition of the whole energy system and only few in-between connection lines join them together. Inside each sub network, we divide the nodes into two sets:

- Internal nodes: nodes that only have connections with the nodes inside the same sub network.
- Boundary nodes: nodes that have connections with the nodes in other sub networks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2014.2364928, IEEE Transactions on Emerging Topics in Computing

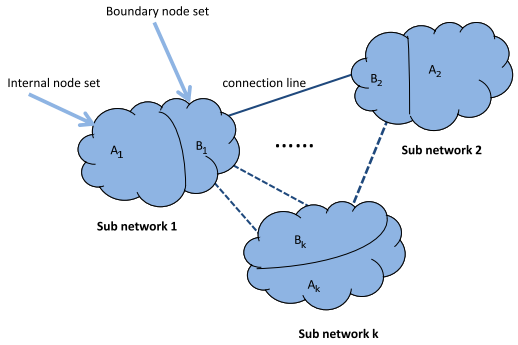


Fig. 3. The general architecture of power network

The architecture of the power network is illustrated in Figure 3. Based on the previous analysis, we apply the Schur complement domain decomposition method [29] to our power network model. Specifically, suppose there are k sub networks, by grouping the internal nodes of each sub network and putting all the boundary nodes of the network in the back, we formulate the nodal analysis model for the whole power network as the following:

$$\begin{bmatrix} Y_{A_1A_1} & 0 & \cdots & 0 & Y_{A_1B} \\ 0 & Y_{A_2A_2} & \cdots & 0 & Y_{A_2B} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & Y_{A_kA_k} & Y_{A_kB} \\ Y_{BA_1} & Y_{BA_2} & \cdots & Y_{BA_k} & Y_{BB} \end{bmatrix} \begin{bmatrix} V_{A_1} \\ V_{A_2} \\ \vdots \\ V_{A_k} \\ V_B \end{bmatrix} = \begin{bmatrix} I_{A_1} \\ I_{A_2} \\ \vdots \\ I_{A_k} \\ I_B \end{bmatrix} \quad (5)$$

Notice that B is the set of all boundary nodes in the whole network, consisting of $B_1, B_2, ..., B_k$. Therefore, Y_{A_iB} only has non zero entries in its submatrix $Y_{A_iB_i}$, for all $i = 1, 2, ...k$.

From (5), if the voltages for boundary nodes set V_B is known, then the voltages for the nodes in each sub network can be calculated as the following:

$$Y_{A_iA_i}V_{A_i} = I_{A_i} - Y_{A_iB}V_B, \forall i \in \{1, 2, ..., k\}. \quad (6)$$

Meanwhile, if we keep the corresponding part for the boundary node set B in equation (5), we can get:

$$\widetilde{Y_{BB}}V_B = \widetilde{I_B} \quad (7)$$

where

$$\widetilde{Y_{BB}} = Y_{BB} - \sum_{i=1}^{k} Y_{BA_i}{Y_{A_iA_i}}^{-1}Y_{A_iB} \quad (8)$$

$$\widetilde{I_B} = I_B - \sum_{i=1}^{k} Y_{BA_i}{Y_{A_iA_i}}^{-1}I_{A_i} \quad (9)$$

Define

$$x_i = Y_{BA_i}{Y_{A_iA_i}}^{-1}Y_{A_iB} \quad (10)$$

$$y_i = Y_{BA_i}{Y_{A_iA_i}}^{-1}I_{A_i} \quad (11)$$

for all $i = 1, 2, ...k$. Notice that x_i and y_i only requires local information for sub system i, we employ this feature to provide the following two power network emulation paradigms in our platform.

3.6.4 Distributed computation

Power network is usually involved with a huge amount of entities and complex interactions between them. Emulating such a system efficiently requires a large amount of computation resources. By decomposing the emulation instance of one power network into several sub power networks, distributed computing provides the approach to balance the computation resources among several computation hosts and increase the overall emulation performance of the system. Figure 4 demonstrates an example of decomposing one whole network into two subnetworks, which are interconnected by the connection line c with impedance z. Specifically, suppose

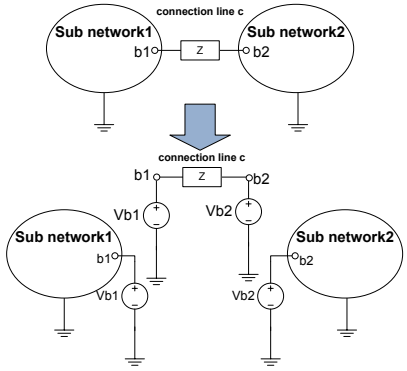


Fig. 4. The decomposition of one power network emulation instance into two.

the power network is decomposed into k sub power networks and each sub power network is assigned to one computation host. We choose one of them as the **Coordinator** and it also caches Y_{BB} and I_B for the boundary node set. Based on equations (6) $\sim$ (11), the decomposition process for each computation host i, $i = 1, 2, ..., k$, is executed as the following:

- Compute x_i and y_i based on equations (10) and (11) respectively. Send the results to the Coordinator host.
- The coordinator collects x_i and y_i from each host, and calculate V_B based on equations (7) (8) (9). After that, it sends V_B back to each host.
- Each host i receives V_B from the coordinator and calculate V_{A_i} based on equation (6).

3.6.5 Dynamic connection/disconnection between multiple instances

Our platform also highlights itself by enabling the dynamic connections and disconnections of several independent emulation instances. The emulation instances of different power networks running on different hosts can interact with each other in run time to evaluate the impact on system dynamics when other energy systems join in. This kind of analysis is critical for the transient stability analysis of power network when the network topology is reconfigured [30]. Figure 5 illustrates the composition example of two independent power net-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2014.2364928, IEEE Transactions on Emerging Topics in Computing

work dynamically connected with each other by a connection line c with impedance z. Specifically, suppose

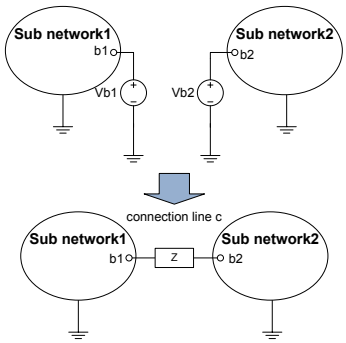


Fig. 5. The composition of two power network emulation instances into one.

there are already k connected emulation instances and another emulation instance $k+1$ joins in run time. Also assume that the $k+1$ instance connects with the instance set E directly, $E \subseteq \{1, 2, ..., k\}$. Then the composition process for each computation host i, $i = 1, 2, ..., k, k+1$, is executed as the following:

- If $i \in E$, then adjust the boundary node set B_i by adding the new boundary nodes connected with instance $k+1$ and also adjust the internal node set A_i by removing the corresponding boundary nodes connected with instance $k+1$. Compute x_i and y_i based on equations (10) and (11) respectively. Send the results to the coordinator host.
- The coordinator first reforms the boundary node set B by adding the new boundary nodes in $B_i, i \in E$ and B_{k+1}, then rebuilt Y_{BB} and I_B for the whole system. Secondly, it collects x_i and y_i from each host, and calculate V_B based on equations (7) (8) (9). Finally, it sends $Y_{BB}V_B$ back to each host.
- Each host i receives $Y_{BB}V_B$ from the coordinator and calculate V_{A_i} based on equation (6).

The dynamic disconnection steps are similar with the above steps except that instead of adding boundary nodes to the boundary set, the coordinator will remove the boundary nodes related with the exiting instances.

3.7 Energy model programming API

In order to support complicated power flow models for various objects in power network, we provide the programming APIs for the user to create various load dynamics, switch configurations and storage charging/discharging configurations. The users invoke the APIs in their own Smart Grid application program to interact with the power module, such that the system power dynamics could be updated. Since the power module is essentially a socket server, all the programming APIs are implemented as socket client and their requests are messages queued up in the server side in a FIFO basis. These APIs are wrapped as a static library and distributed with the SCORE software. The following is a code snippet from examples using the APIs.

```
/* include header file. */
 #include "EnergyDaemon.h"
 int main(int argc, char** argv) {
       ...
/**  This sets the desired energy
          rate of the energy model.
*/
    double desiredEnergy=rand()%50+1;
    ed.setDesiredEnergy(desiredEnergy);
   }
```

4 Evaluation

In this section, we first employ a simple case to evaluate the basic features of the building-in functionalities of SCORE, by presenting the results of power flow calculations, distributed emulation and dynamic connection/disconnection. Then, a comprehensive large scale cyber-physical test case is employed to demonstrate the complete capabilities of SCORE to support cyber-physical analysis.

The first case is created as a Smart Grid power distribution network with one power plant and five houses. Each house is connected with the power network through an intelligent power switch, which serves as the energy control center for the house. Within each house, there are four kinds of nodes: loads (represented by washer), renewable resources (represented by solar panel and wind turbine), and power storages (represented by battery). Wireless networks (the cloud icon) wlan32 equips each node with communication capabilities. We configure the wireless channel to be 54Mbps bandwidth, 200ms delay, 0.1% packet loss rate and the resistances of all the power lines to be the same constant value. Figure 6 shows the setting up of the study case and the initial power flow without any demand response involved. The thick lines means the power flow is relatively heavy at that moment. For each virtual time period h in a virtual

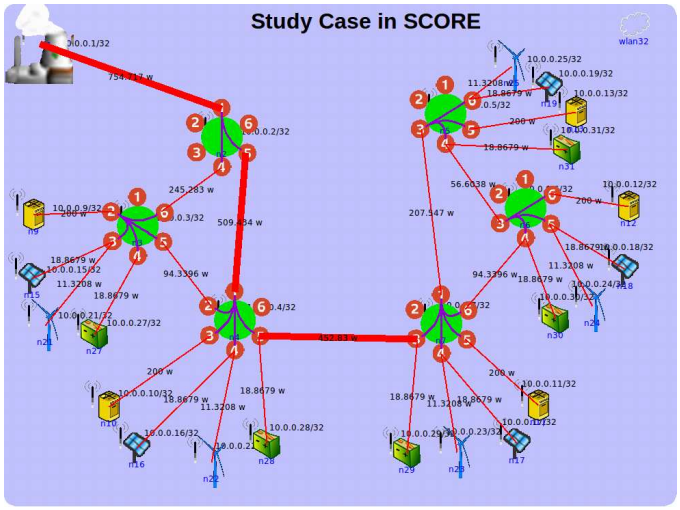


Fig. 6. The basic case in SCORE

day, $h = 0, 1, 2, 3, ...23$, each kind of nodes in the study case behaves as the following:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2014.2364928, IEEE Transactions on Emerging Topics in Computing

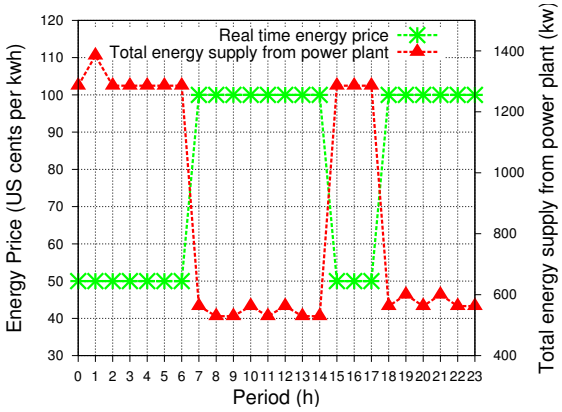


Fig. 7. Real time price and Total energy supply from power plant

- **Power Plant:** Broadcasting its real time energy prices to all the intelligent power switches. The energy price has only two values: HIGH (100 cents per kwh) and LOW (50 cents per kwh).
- **Intelligent Power Switch:** Receiving the energy price information from the power plant and then relay the messages to the other nodes within the same house immediately.
- **Load:** Receiving energy price information from intelligent power switch. When the price is HIGH, it lowers its load from 200w to 150w. When the price is LOW, it increases its load from 150w to 200w.
- **Solar Panel:** When $h \in H1 = \{h|h = 0, 1, 2, 3, 4, 5, 19, 20, 21, 22, 23\}$, setting its maximum output to 0. When $h \in H2 = \{h|h = 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18\}$, setting its maximum output to 30w.
- **Wind Turbine:** Randomly setting its maximum output to 30w or 60w.
- **Power Storage:** Its capacity is set to 50kwh and initial energy is 10kwh. After receiving energy price information from intelligent power switch. When the price is HIGH, it start discharging if there is still enough residual energy in the storage. When the price is LOW, it stops discharging. The battery keeps discharging until all the energy is used up.

Note that here the virtual hour is a test case iteration clock, not an hour in real time.

4.1 Power flow calculation

SCORE captures the power flow analysis using current model in circuit theory for Smart Grid applications in real time. Figure 7 shows the real time energy price and the total energy supply from power plant in a 24 hours virtual day. The impact of the demand response strategy on total energy supply is clearly illustrated by the two trend curves in the test case: when the price goes up, the total energy supply goes down. And when the price goes down, the total energy supply goes up correspondingly. Renewable resources result in the fluctuations of total energy supply curve. More specifically, table 2 presents the power values passing through the nodes and the power lines between intelligent power switches during execution periods. For example, when the virtual clock $h = 6$, the energy price is low, so that the washer is in 200w mode, the storage stops discharing. The power flows of all power lines conform to Kirchhoff's circuit and voltage laws in power network nodal analysis.

4.2 Distributed emulation

SCORE extends its scalability by distributed emulation. We evaluate SCORE's scalability using the previously introduced test case and extend it to a much larger scale. Our testing machines are 64 bits HP destop with Pentium(R) Dual-Core CPU E5700 @ 3.00GHz and 4GiB memory. Figure 8 and Figure 9 shows the peak CPU usage and memory usage of SCORE running on one, two and four machines when the scale of the Smart Grid increases. We can see from the figures that as the number of emulation servers increases, the CPU and memory usage of each machine is decreased since each of them can take care of the instantiation, computation and communication in parts of the Smart Grid. A single instance of our PC can support about 100 virtual nodes effectively. SCORE's distributed emulation capability greatly release the burden of each emulation server and enable large scale emulation.

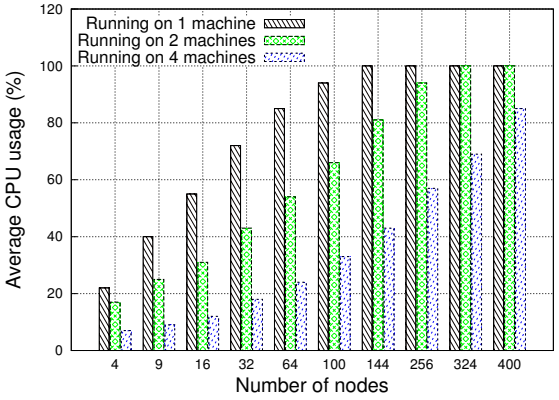


Fig. 8. Peak CPU usage running the study case in SCORE

Figure 10 shows the execution time of the study case for 24 clocks. Note that when the scale of power grid is small, which means the computation cost is relatively low, less machines can finish the emulation relatively faster because there is less communication overhead between emulation servers. However, when computation time dominates the execution time, the advantages of parallel computation begin to emerge.

4.3 Dynamic connections/disconnection between instances

SCORE supports dynamic connections between multiple Smart Grid emulation instances. We evaluate this feature like shown in Figure 11. Here we use two independent emulation instances running separately on different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2014.2364928, IEEE Transactions on Emerging Topics in Computing

TABLE 2
Test Case result

Period	h=0	h=2	h=6	h=8	h=12	h=14	h=16	h=18	h=20	h=22
Washer(n9)	200w	200w	200w	150w	150w	150w	200w	150w	150w	150w
Solar Panel(n15)	0w	0w	21.4286w	8.82353w	9.375w	8.82353w	21.4286w	9.375w	0w	0w
Wind Turbine(n21)	42.8571w	42.8571w	21.4286w	17.6471w	9.375w	17.6471w	21.4286w	9.375w	18.75w	18.75w
Storage(n27)	0w	0w	0w	17.6471w	18.75w	17.6471w	0w	18.75w	18.75w	18.75w
Power line (n2,n3)	334.286w	334.286w	334.286w	137.647w	146.25w	137.647w	334.286w	146.25w	146.25w	146.25w
Power line (n2,n4)	411.429w	411.429w	411.429w	169.412w	180w	169.412w	411.429w	180w	180w	180w
Power line (n4,n7)	540w	540w	540w	222.353w	236.25w	222.353w	540w	236.25w	236.25w	236.25w
Power line (n5,n6)	77.1429w	77.1429w	77.1429w	31.7647w	33.75w	31.7647w	77.1429w	33.75w	33.75w	33.75w
Power line (n5,n7)	180w	180w	180w	74.1176w	78.75w	74.1176w	180w	78.75w	78.75w	78.75w
Power line (n4,n6)	102.857w	102.857w	102.857w	42.3529w	45w	42.3529w	102.857w	45w	45w	45w
Power line (n5,n6)	77.1429w	77.1429w	77.1429w	31.7647w	33.75w	31.7647w	77.1429w	33.75w	33.75	33.75w
Power line (n3,n6)	231.429w	231.429w	231.429w	95.2941w	101.25w	95.2941w	231.429w	101.25w	101.25w	101.25w

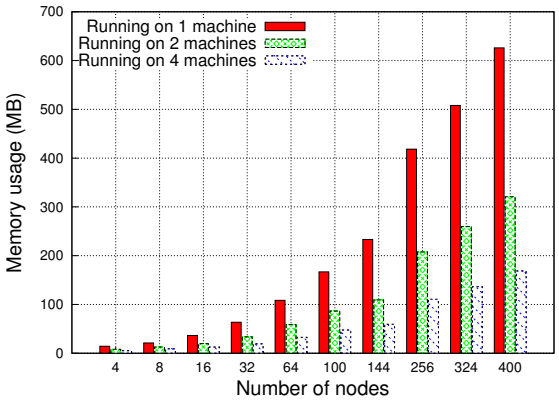


Fig. 9. Memory usage running the study case in SCORE

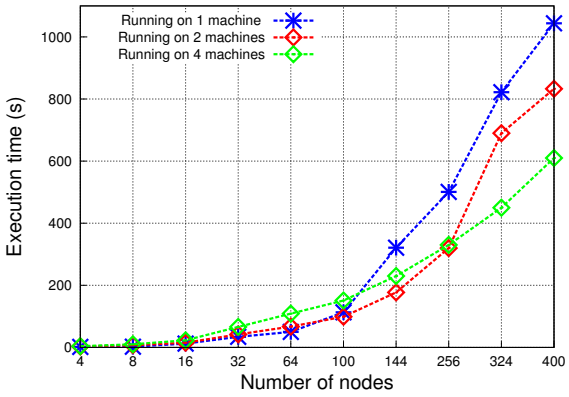


Fig. 10. Execution time of the study case for 24 clocks in SCORE

servers and dynamically connect with each other. Notice that the combined instance of the two is the same as the single instance in subsection 4.1, which is running in a single emulation server. Figure 12 shows the dynamics of the load of washer (n12). The two instances are connected at virtual clock $h = 9$. Before $h = 9$, the pricing messages broadcast by the power plant cannot be received by the nodes in Smart Grid instance Two, so the desired maximum load of washer(n12) is always 200w. However, Smart Grid instance Two forms a microgrid by itself and there is no power flows from the power plant. So the load of n12 totally depends on the solar energy , random generated wind energy and the storage energy. The desired load cannot be met. After $h = 9$, the behavior of n12 is almost the same as it is in subsection 4.1.until $h = 17$, when the two emulation instances are disconnected. After $h = 17$, Smart Grid instance Two forms a microgrid again, and the energy consumption behavior is similar as the one before $h = 9$. The fluctuations are resulted from the random generated wind energy.

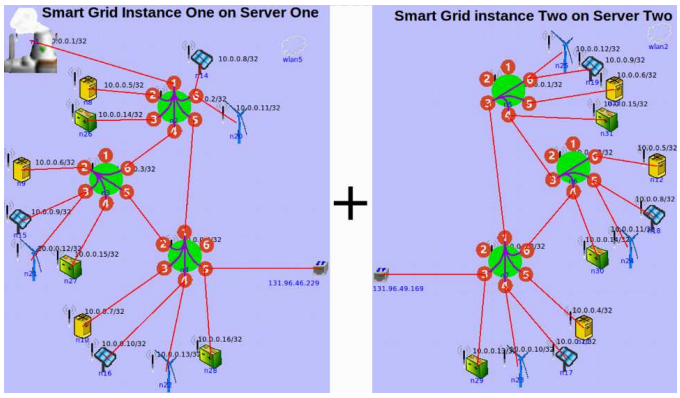


Fig. 11. Dynamic connections of two Smart Grid emulation instances in SCORE

4.4 Comprehensive cyber-physical analysis: under cyber attacks

This testing case we created is based on the AMI network test case from American Electric Power Company [31] and the IEEE PES 37 bus distribution system test feeders [32]. Through this case, we further illustrate the advantages of our platform over software simulators: 1) the actual control program written in C language (either correct or malicious modified) can be directly run on each virtual node; 2) the real time cyber-physical impacts (altered system routing table entries and power flow perturbations) of the control programs can be demonstrated. Figure 13 shows the building blocks of our experimental scenario:

- Operation layer: The control program in control center broadcasts real-time energy prices every 5 minutes and also collects meter reading data through

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2014.2364928, IEEE Transactions on Emerging Topics in Computing

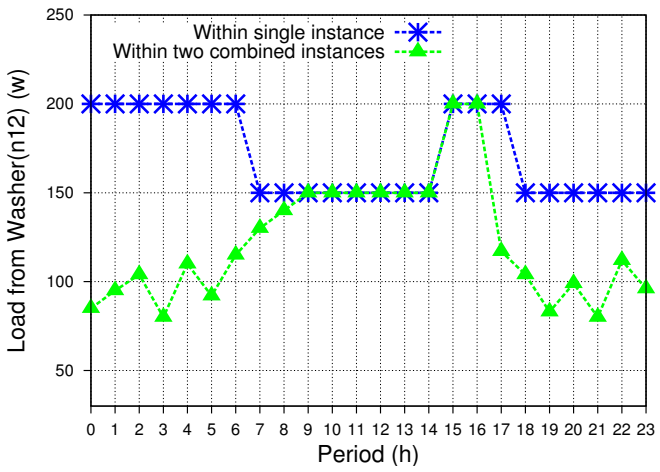


Fig. 12. Load dynamics of washer(n12)

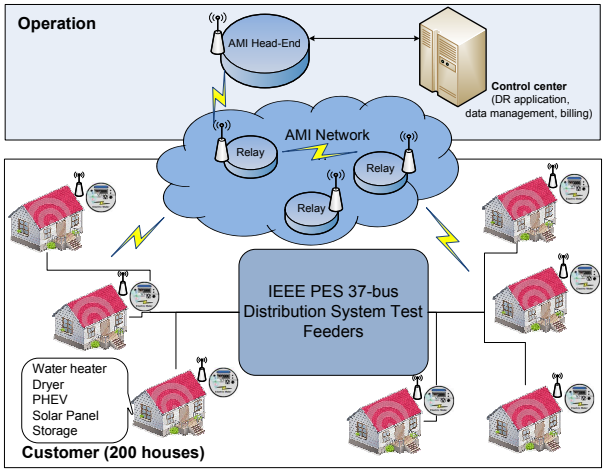


Fig. 13. The comprehensive case in SCORE

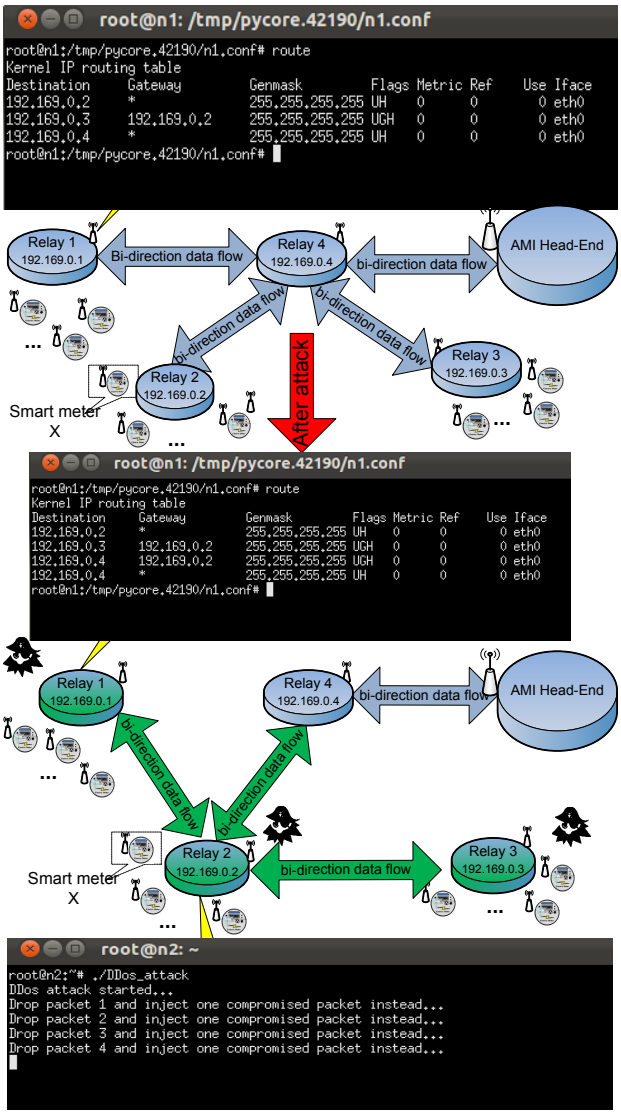


Fig. 14. Potential attack within AMI

AMI Head-End. Meanwhile, the control center also calculates the bills for each house, based on the real time price and the collected energy consumption data.

- Customer layer: The IEEE 37 bus distribution test feeders is set up to provide power for 200 residential houses. Each house is equipped with loads including a water heater, a dryer, a PHEV, a solar panel, and a storage. Moreover, a smart meter is employed to serve as the interface between the power network and AMI for each house. The program running in smart meter responses to the real time prices to adjust the setpoint of appliances within each house correspondingly based on the price-responsive control model in [33].
- AMI network: AMI enables communications and interactions between/within the operation layer and the customer layer. The control center and AMI Head-End is connected through Internet. AMI Head-End, the relay nodes and the smart meters are formed as a IEEE 802.11 Radio Frequency Mesh network.

Suppose the customer under smart meter X wants to manipulate his energy bill without being caught. In order to achieve this, he launches a Distributed Denial-of-service attack to the bi-direction data flow within AMI, which consists of the energy consumption data from the smart meters to the AMI Head-End, and the energy price data from the AMI Head-End to the smart meters. For the energy consumption data, the attacker modifies the ones from smart meter X and his targeted neighbors, such that each targeted neighbor has an increase in the reported energy consumption compared to the actual consumption, and the smart meter X has a decrease equal to the total increase of its targeted neighbors in the reported energy consumption. In this way, from the perspective of utility company, the total energy provided still conforms to the total energy being billed. For the energy price data, the attacker modifies the price to a lower value, such that based on the demand response model, the actual energy consumption of each targeted

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2014.2364928, IEEE Transactions on Emerging Topics in Computing

neighbors will also increase. In this case, from the perspective of each targeted neighbor, the minor increase in the reported energy consumption data due to attack will become even less noticeable.

Specifically, as shown in Figure 14, the customer of smart meter X attacks three relay nodes at the same time: its own direct cluster head (Relay 2) and two neighbor cluster heads (Relay 1 and 3). Originally, Relay 1 and Relay 3 will directly interact with Relay 4 for the bi-direction data. We can see this from the result of $route$ command in the terminal of Relay 1. To reach 192.169.0.4, which is the IP address of Relay 4, no intermediate gateway is needed and packets can be simply forwarded through interface $eth0$. However, after attack, there is one extra high priority entry in the routing table of Relay 1 such that the packets designated to 192.169.0.4 will be forwarded to 192.169.0.2 first instead of the original one hop reach. As a result, for Relay 2, besides the data packets of the 10 customers within its own cluster, it will also intercept the data packets of the other 20 customers within the clusters of Relay 1 and 3. By making the three Relay nodes working in concert to compromise the data, customer X could dramatically reduces its own reported energy usage. As shown in Figure 15, for smart meter X, even though the actual energy usage across the day is 64kwh, the reported data is manipulated to 35kwh. The remaining $64-35=29$kwh are evenly added to the other 29 customers' reported data. In this way, from the perspective of utility company, the total energy consumed still conforms with the total energy being billed. Moreover, from the perspective of each of the other 29 customers', since only $29/29=1$kwh is added to their energy consumption, which usually results in about 0.1$ increase in their bills,it is very much likely that the customer will just let it go. Also note that since the

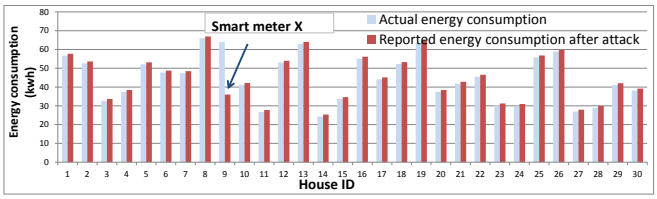


Fig. 15. Actual energy usage and reported energy usage after attack

energy price is modified to a lower value after the attack, the real power consumption paradigm of the attacked neighbors changes dramatically, compared to the normal situation when the correct real time energy price is given. As shown in Figure 16, the real power consumption of the attacked neighbors stays at a relatively higher level all the time after the attack and the demand response through real time pricing is not working any more. If more neighbors are involved in the attack, this will severely increase load of the system, which can result in a higher cost of power transmission or even an outage. An effective approach to detect this kind of attack is

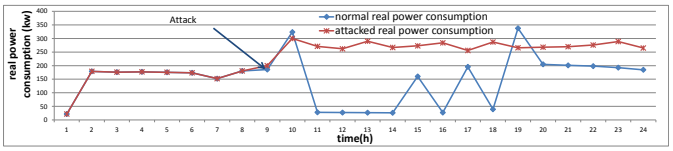


Fig. 16. The total real power consumption of the attacked neighbors

by monitoring the network traffic. As shown in Figure 17, since the routing path of the packets is changed and much more data packets are forwarded to Relay 2, the throughput of Relay 2 will be increased unusually from the moment of attack. Also, the network traffic congestion at Relay 2 will result in an increase in the communication delay from Relay 1 to the AMI meter head.

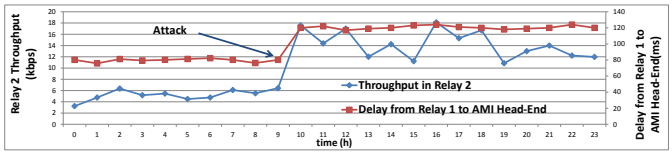


Fig. 17. Throughput and Communication delay

5 Conclusion and Future Work

In this paper, we present the design, implementation and operation of SCORE for Smart Grid emulation. One future direction would be integrating SCORE with real hardware testbed to create a uniform cyber-physical analysis platform. Also, a cloud based deployment for our platform could be built to provide universal access for the users.

References

[1] J. Ahrenholz, T. Goff, and B. Adamson, "Integration of the core and emane network emulators," in *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011*, nov. 2011, pp. 1870 –1875.

[2] "Jeju Smart-Grid Testbed." [Online]. Available: http://smartgrid.jeju.go.kr/eng/

[3] D. Stimoniaris, D. Tsiamitros, T. Kottas, N. Asimopoulos, and E. Dialynas, "Smart grid simulation using small-scale pilot installations. - experimental investigation of a centrally-controlled microgrid," in *PowerTech, 2011 IEEE Trondheim*, june 2011, pp. 1 –6.

[4] W.-Z. Song, D. De, S. Tan, S. Das, and L. Tong, "A wireless smart grid testbed in lab," *Special Issue on Recent Advances in Wireless Technologies for Smart Grid, IEEE Wireless Communications Magazine*, 2012.

[5] M. Stanovich, I. Leonard, K. Sanjeev, M. Steurer, T. Roth, S. Jackson, and M. Bruce, "Development of a smart-grid cyber-physical systems testbed," in *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*, 2013, pp. 1–6.

[6] A. Hahn, A. Ashok, S. Sridhar, and M. Govindarasu, "Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid," *Smart Grid, IEEE Transactions on*, vol. 4, no. 2, pp. 847–855, 2013.

[7] Y. Guo, R. Li, G. Poulton, and A. Zeman, "A simulator for self-adaptive energy demand management," in *Self-Adaptive and Self-Organizing Systems, 2008. SASO '08. Second IEEE International Conference on*, oct. 2008, pp. 64 –73.

[8] A. Molderink, M. Bosman, V. Bakker, J. Hurink, and G. Smit, "Simulating the effect on the energy efficiency of smart grid

technologies," in *Simulation Conference (WSC), Proceedings of the 2009 Winter*, dec. 2009, pp. 1530 –1541.

[9] P. Faria, Z. Vale, and J. Ferreira, "Demsi: A demand response simulator in the context of intensive use of distributed generation," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, oct. 2010, pp. 2025 –2032.

[10] PSCAD, "https://pscad.com/."

[11] A. Narayan, "GridSpice-A Virtual Test Bed for Smart Grid," Tech. Rep., 2008.

[12] D. Chassin, K. Schneider, and C. Gerkensmeyer, "Gridlab-d: An open-source power systems modeling and simulation environment," in *Transmission and Distribution Conference and Exposition, 2008. T.No.x00026;D. IEEE/PES*, april 2008, pp. 1 –5.

[13] D. D. G. Ray D. Zimmerman, Carlos E. Murillo-Snchez, "A MATLAB Power System Simulation Package," Tech. Rep., 1999.

[14] "Co-simulation overview." [Online]. Available: http://www.flowmaster.com/flowmaster_cosimulation.html

[15] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury, "Epochs: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components," *Power Systems, IEEE Transactions on*, vol. 21, no. 2, pp. 548 – 558, may 2006.

[16] NS2, "http://www.isi.edu/nsnam/ns/."

[17] T. Godfrey, S. Mullen, R. Dugan, C. Rodine, D. Griffith, and N. Golmie, "Modeling smart grid applications with co-simulation," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, oct. 2010, pp. 291 –296.

[18] OpenDSS, "http://sourceforge.net/projects/electricdss/."

[19] M. Mallouhi, Y. Al-Nashif, D. Cox, T. Chadaga, and S. Hariri, "A testbed for analyzing security of scada control systems (tasscs)," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, Jan 2011, pp. 1–7.

[20] H. Lin, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, "Power system and communication network co-simulation for smart grid applications," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, jan. 2011, pp. 1 –6.

[21] Y. Deng, H. Lin, S. Shukla, J. Thorp, and L. Mili, "Co-simulating power systems and communication network for accurate modeling and simulation of pmu based wide area measurement systems using a global event scheduling technique," in *Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), 2013 Workshop on*, May 2013, pp. 1–6.

[22] PSLF, "http://site.ge-energy.com/prodserv/gepslf/index.htm."

[23] K. Mets, T. Verschueren, C. Develder, T. Vandoorn, and L. Vandevelde, "Integrated simulation of power and communication networks for smart grid applications," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2011 IEEE 16th International Workshop on*, june 2011, pp. 61 –65.

[24] OMNET++, "http://www.omnetpp.org/."

[25] J. Nutaro, P. Kuruganti, L. Miller, S. Mullen, and M. Shankar, "Integrated hybrid-simulation of electric power and communications systems," in *Power Engineering Society General Meeting, 2007. IEEE*, june 2007, pp. 1 –8.

[26] S. Tan, W.-Z. Song, Q. Dong, and L. Tong, "Score: Smart-grid common open research emulator," in *The 3rd IEEE International Conference on Smart Grid Communications (IEEE SmartGridComm)*, 2012.

[27] S. Karnouskos, "Cyber-physical systems in the smartgrid," in *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, july 2011, pp. 20 –23.

[28] B. Zhang, S. Sun, and Z. Yan, *Advance Power analysis*, T. U. Press, Ed., 2007.

[29] Y. Saad, *Iterative Methods for Sparse Linear Systems, Second Edition*, 2nd ed. Society for Industrial and Applied Mathematics, Apr. 2003. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0898715342

[30] A. Hoballah and I. Erlich, "Transient stability assessment using ann considering power system topology changes," in *Intelligent System Applications to Power Systems, 2009. ISAP '09. 15th International Conference on*, nov. 2009, pp. 1 –6.

[31] R. Sarfi, B. D. Green, and J. Simmins, "AMI Network (AMI Head-End to/from Smart Meters)," August 2012.

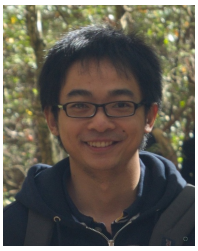
[32] IEEE PES Distribution System Analysis Subcommittee's Distribution Test Feeder Working Group, "IEEE 37 Node Test Feeder," Tech. Rep., September 2010.

[33] Hammerstrom, D. J., "Pacific Northwest GridWiseTM Testbed Demonstration Projects: Part I. Olympic Peninsula Project," Tech. Rep., October 2007.

Song Tan received his BSc degree in Computer Science from both Northeast Normal University (China) and Southern Polytechnic State University (USA) in 2010. He is currently a PhD Candidate in Sensorweb Laboratory, Georgia State University, USA. His research is about Smart Grid emulation, testbed design and power system modeling.

WenZhan Song is a tenured associate professor in Georgia State University. His research mainly focuses on sensor web, smart grid and smart environment where sensing, computing, communication and control play a critical role and need a transformative study. His research has received 6 million+ research funding from NSF, NASA, USGS, Boeing and etc since 2005, and resulted in 80+ journal articles, conference articles and book chapters in this area.

Qifen Dong received her BE degree from the department of Electronics Information Engineering, Zhejiang University of Technology, Hangzhou, China, in 2007. She has been working toward the PhD degree in the department of Control Theory and Control Engineering, Zhejiang University of Technology since 2007. Her research interests including Wireless Sensor Networks, Smart Grids and Embeded system.



Dan Huang received his MS in computer science from Southeast University, China. He is currently a graduate student in computer science department at Georgia State University and he works in Sensorweb Lab. His research interests include distributed computing, social networks, smart grid.

Lang Tong joined Cornell University in 1998 where he is now the Irwin and Joan Jacobs Professor in Engineering and the Cornell site director of the Power Systems Engineering Research Center (PSerc). Lang Tong is a Fellow of IEEE and received Best Paper Award (with Min Dong) from the IEEE Signal Processing Society, the Leonard G. Abraham Prize Paper Award from the IEEE Communications Society (with Parvathinathan Venkitasubramaniam and Srihari Adireddy).