Demo: Transactive Energy Application with RIAPS

Scott Eisele*, Purboday Ghosh*, Keegan Campanelli*, Abhishek Dubey* and Gabor Karsai*
*School of Electrical and Computer Engineering
Vanderbilt University, Nashville, TN 37212
Email: {purboday.ghosh,keegan.m.campanelli,scott.r.eisele,abhishek.dubey,gabor.karsai}@vanderbilt.edu

***Abstract*—The modern electric grid is a complex, decentralized cyber-physical system requiring higher-level control techniques to balance the demand and supply of energy to optimize the overall energy usage. The concept of Transactive Energy utilizes distributed system principle to address this challenge. In this demonstration we show the usage of the distributed application management platform RIAPS in the implementation of one such Transactive Energy approach to control elements of a power system, which runs as a a simulation using the Gridlab-d simulation solver.**

I. Overview

Transactive energy markets are an example of a distributed application which requires real time coordination between the actors in the system. In a previous work we presented RIAPS (Resilient Information Architecture Platform for distributed Smart Systems) [1], which is a middleware designed to provide the software foundation for such applications. RIAPS employs a hierarchical architecture model that encapsulates the low level platform services such as communication, remote deployment of applications, concurrency control and fault tolerance. RIAPS provides a modeling language that facilitates the development of distributed applications through the creation of components which are contained within actors (i.e. Linux processes). The components have associated ports that can be used to send and receive RIAPS messages. Using RIAPS we developed a transactive energy application TRANSAX [2] which provides the infrastructure necessary to allow prosumers to safely and securely exchange energy resources for future time intervals. TRANSAX takes prosumer *offers* to consume or produce energy as inputs and outputs *matches* to the prosumers which inform them of how much energy they are able to safely produce or consume during the interval of interest. In that work the application was tested in open loop setting where the output was simply plotted to show the internal trading efficiency of the market, an example of which can be seen in figure 1.

TESP (The Transactive Energy Simulation Platform) [3] is an open-source platform developed by the Pacific Northwest National Laboratory (PNNL) that brings together multiple simulation agents like GridLAB-D, MATPOWER or PYPOWER and EnergyPlus and incorporates them to design interactive energy systems for evaluation and experimentation. Each of these systems is run as a simulation federate within TESP, which coordinates and synchronizes them using the Framework for Network Co-simulation (FNCS). FNCS is responsible for synchronizing the simulations and provides publish-subscribe

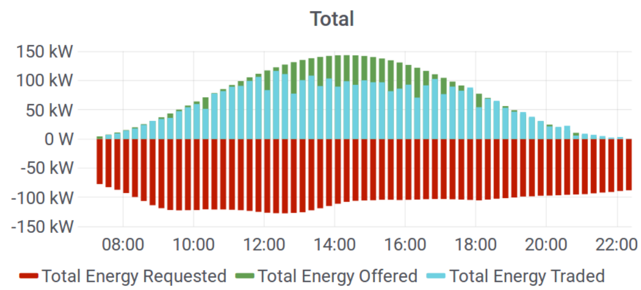

Fig. 1. Total energy production capacity (green), and energy demand (red) for each interval, as well as the total energy traded in each interval (blue) while subject to constraints. $C_{ext} = 2$MW, $C_{int} = 2.5$MW.

messaging allowing message exchange between agents. FNCS also allows customized agents to be built using Python. These agents act as FNCS federates in the TESP co-simulation framework. FNCS is also responsible for granting simulation time steps to the agents.

For this demonstration we integrate TRANSAX with an online power grid simulation by using those matches to control the charge and discharge rates of batteries connected to homes and on/off state of solar panels of the simulated power grid. The grid is implemented and simulated using GridLAB-D, a stepped steady-state simulator for power systems. TESP provides the integration between the two. The integrated package demonstrates the RIAPS capabilities in a complex real-time distributed system such as a power grid.

II. Test bed and Demonstration Scenario

In this section, we describe the testbed hardware, the simulated microgrid topology on which the experiment is performed as well as the different scenarios that will be tested.

A. Testbed Architecture

The overall architecture of the system consists of the following elements:

1) **Hardware**: We use a Hardware-in-the-Loop testbed, where embedded computing nodes are networked together with the simulation environment. The simulation provides the sensor data to the the embedded computers which have control processes running that send commands to the simulated system. The testbed can be seen in figure 2.
2) **Microgrid Topology**: In this demonstration we modify the TESP TE30 example to suit the model assumed by TRANSAX. TE30 consists of thirty houses modeled and

978-1-7281-0151-4/19/$31.00 ©2019 IEEE
DOI 10.1109/ISORC.2019.00024

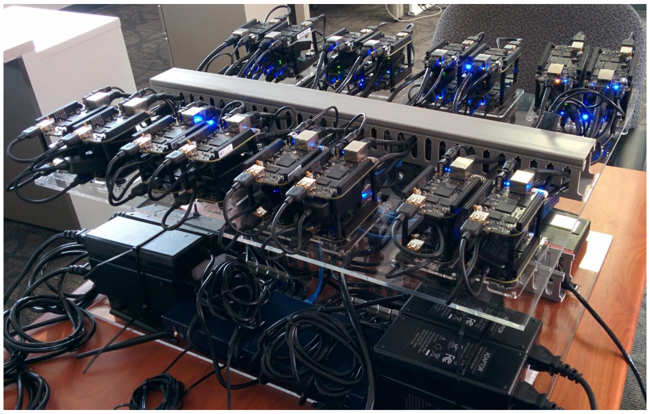

Fig. 2. 32 embedded compute node testbed.

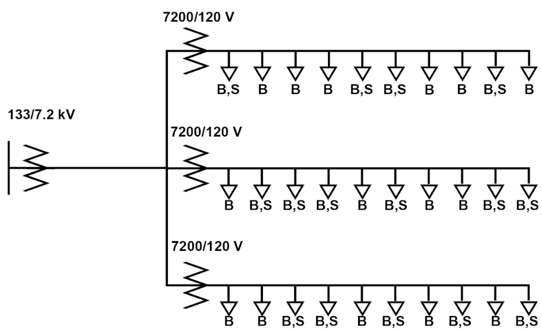

Fig. 3. 30 Home feeder topology. B stands for Battery and S stands for Solar panel

simulated using GridLAB-D, a commercial building simulated in EnergyPlus and the bulk system simulated in MATPOWER or PYPOWER. In order to highlight the transactive energy aspects we simplify the model by removing all elements, except for the homes whose loads are replaced by a battery model. Each house additionally retains the associated solar panel and corresponding inverter. The resulting model can be seen in figure 3.

3) **FNCS**: FNCS acts as the centralized coordinator for the various participating agents through the *fncs_broker*. It synchronizes the simulation, relays the sensor data collected from GridLAB-D to the TESP agent, and the set point commands in the opposite direction.
4) **TESP Agent**: The TESP agent is a RIAPS device component configured to run as a FNCS federate. It uses the FNCS library to subscribe to the battery usage metrics of the simulated network and passes them to the TRANSAX producer and consumer components. Once TRANSAX notifies the prosumers how much energy they can safely transfer the prosumers notify the TESP agent, which then converts them to a format appropriate for, and publishes them to GridLAB-D. The communication with TRANSAX components is handled via RIAPS communication ports.
5) **TRANSAX**: The transactive energy exchange application layer is TRANSAX implemented as a RIAPS application.

B. Scenarios

We modify the base model to create the two environment scenarios and corresponding simulations below:

1) **Stage I: Battery only**: The network topology is simplified by removing the solar panels and their corresponding inverters. Each load in the network contains only a battery and its inverter. In this model the TRANSAX prosumers receive sensor data for the charge level of their batteries via the TESP agent which is connected to the FNCS. Using this information the control logic of the prosumers is implemented using RIAPS to control the offers they post. As the simulation progresses, the application communication is handled by the RIAPS agents, and the energy transfers and grid load are visible in a dashboard similar to that of figure 1.
2) **Stage II: Battery with solar**: Solar panels and inverters are added to fifteen of the thirty homes in the network. In this scenario we again see the flow of RIAPS messages, posting offers in response to the detected battery state and incoming solar charge rate. This information is again used in the control logic of the prosumers implemented with RIAPS to post offers to provide or consume energy. When the energy offers are matched by TRANSAX and the prosumers notified, they send this data through the TESP agent which converts the energy to be transferred to a battery charge or discharge rate and actuates the corresponding control parameter. In the case that TRANSAX does not grant a transaction to a solar panel, then its inverter will be disconnected and the substation load will increase.

The transactive control, with the help of the RIAPS device component, maximizes the amount of energy traded within the microgrid without violating the safety constraints of the main grid.

Acknowledgment

This work was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000666 and a grant from Siemens, CT. The views and opinions of authors expressed herein do not necessarily state or reflect those of the US Government or any agency thereof or of Siemens, CT.

References

[1] S. Eisele, I. Mardari, A. Dubey, and G. Karsai, "Riaps: Resilient information architecture platform for decentralized smart systems," in *2017 IEEE 20th International Symposium on Real-Time Distributed Computing (ISORC)*, May 2017, pp. 125–132.

[2] A. Laszka, S. Eisele, A. Dubey, G. Karsai, and K. Kvaternik, "TRANSAX: A blockchain-based decentralized forward-trading energy exchange for transactive microgrids," in *Proceedings of the 24th IEEE International Conference on Parallel and Distributed Systems (ICPADS)(December 2018)*, 2018.

[3] PNNL. (2017) Pnnl, transactive energy simulation platform (tesp). [Online]. Available: https://tesp.readthedocs.io/en/latest/