

# Heterogeneous Distributed Embedded Linux System for Hardware-in-the-Loop Smart Grid Testbed

[Final Project Report]

Christopher Hannon  
Illinois Institute of Technology  
10 West 31st Street  
Chicago, Illinois, 60616  
channon@hawk.iit.edu

Neil Getty  
Illinois Institute of Technology  
10 West 31st Street  
Chicago, Illinois, 60616  
ngetty@hawk.iit.edu

## ABSTRACT

The successful operations of modern electric power infrastructure are dependent on a reliable and efficient communication network. Researchers and utility companies are exploring how software-defined networking (SDN) technology can enhance efficiency and resilience of the Smart Grid. This trend calls for a simulation-based platform that provides sufficient features and controllability for evaluating network application designs, as well as the facilitation of transitions from in-house research ideas to real systems. In this work, we present a hybrid distributed testing platform that combines a power distribution system simulator with a heterogeneous distributed embedded Linux system. This system supports distributed communication network emulation using real networking hardware to support high fidelity analysis of communication network applications and their impacts on the power systems. Our main contributions are two-fold, the first is in the design of a virtual time system for distributed Linux devices with precision controllability of the execution of the emulation systems. In other words, pausing and resuming any specified container processes in the perception of their own virtual clocks with minimal overhead. The second contribution is the efficient synchronization of the two virtual time based sub-systems.

## 1. BACKGROUND

The power grid is composed of power generation, transmission, distribution and loads. Traditionally, power is generated in mass quantities from hydro, coal, nuclear, and gas sources. The power is then transmitted at high voltages to distribution systems where the power is distributed to residential and commercial consumers. As the power grid is moving towards a smarter grid, the efficient energy management is increasingly dependent on the underlying communication network supporting reliable information transfer among the various entities in the grid.

With distributed power generation—such as solar and wind energy—and more storage technology, there is a need for understanding the state of the power network in real time. A challenge with the integration of such generation, is the uncertainty and intermittency of the availability of power generation. In order to combat this challenge, there needs to be an infrastructure that allows for the monitoring and control of the system state. To do this effectively, requires a reliable and resilient communication network.

Researchers have developed systems to co-simulate the power and network components of the smart grid [1–3, 5, 6, 8, 10]. [9] surveys the existing technologies and motivations for co-simulation.

In [10], a system is proposed using OpenDSS to allow for sending real-time signals to hardware integrating with simulation. Real time simulators are used for hardware-in-the-loop simulations, allowing for simulation-emulation closer to the real system [3]. This gives high fidelity, but requires power equipment and often specific simulator hardware. While real-time simulators exist, they are costly and often require specific hardware to operate, our challenge is to design a hardware-in-the-loop testbed that can use standard electric power simulators that run slower than real-time while maintaining temporal fidelity.

In [8], the authors create a co-simulation between PSLF and ns-2. They use a global event driven mechanism for sending synchronization messages between the two simulators. In simulation, events are sorted by time stamps, typically in a priority queue and are executed as fast as possible without regard to the wall clock.

EPOCHS [6] uses commercial power simulators to co-simulate network and power systems through the use of agents. This platform uses agents to effectively co-simulate power and communication elements. The authors define agents as having the properties of autonomy and interaction. That they exhibit properties of mobility, intelligence, adaptivity and communication. In our design, our models run real processes on real hardware but can influence and extract values from the power simulator. This allows for us to make use of agents to as entities that exist in both systems.

FNCS [2] is a federated approach for co-simulation of power and electrical simulators by combining multiple power simulators, both distribution and transmission and use ns-3 as a communication simulator. In [1], the same authors improve the synchronization between systems. The difference is that our approach is focused on running real network pro-

cesses which has different synchronization challenges due to the inherent difference between the execution mechanisms in simulation and real-time processes.

There are two main features that set our design apart from the existing tools. The first is that we are using real hardware for networking components rather than a simulator. The second is that our testbed's design works without real-time simulators making our approach general and low-cost.

## 2. PROJECT GOALS

While simulation systems offer a convenient and low cost method of performing evaluation on models of systems it lacks the fidelity that real systems contain.

This project aims to achieve the following:

- create a distributed system composed of embedded linux devices including general purpose and router devices.
- provide synchronization solution for real-time processes to synchronize with a discrete time step solution electric power simulator.
- establish a distributed emulation system

The challenges in the creation of such testbed stem for the synchronization issues between nodes in the distributed system, the processes running on the nodes, and the simulator. Specifically we will implement the following in a testbed consisting of 5 nodes, a embedded router, 2 Banana Pi devices, and 2 Raspberry Pi linux devices:

- Creating a kernel module that provides a user-space utility for sending hardware interrupts between the devices in the system. The distributed synchronization between the devices is decentralized with the kernel module.
- Time synchronization strategy with the simulation server.
- Modify the scheduler to control process execution.

## 3. DESIGN

The design of the testbed includes 2 main components: the simulation and management server, and the distributed embedded communication network system.

### 3.1 Simulation and Management Server

### 3.2 Distributed Linux System

### 3.3 Synchronization Challenges

While it is possible to run electric power simulation in real-time [4], such simulators often require specialized hardware and proprietary closed software. Thus real-time simulation is infeasible for low-cost hybrid testbeds. Additionally, the real-time nature of power simulation can not account for the exchange of messages with the network system for passing messages.

Therefore due to the nature of the electric power simulator, it must run slower than real time. This is inherently different from the way that communication systems and communication emulation systems execute.

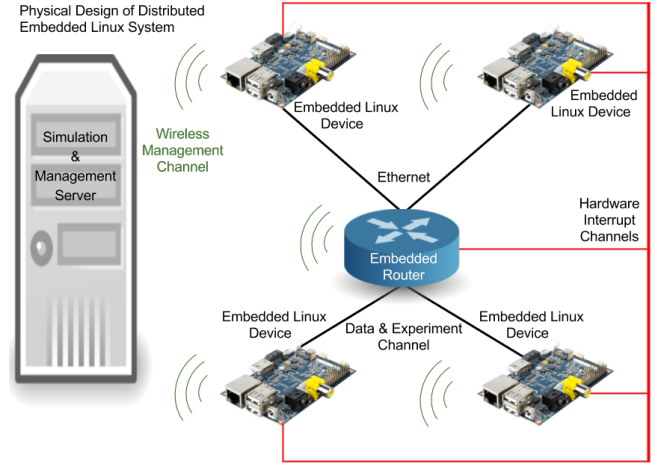


Figure 1: Architecture of distributed system composed of 3 communication channels: Ethernet, wireless management, and direct hardware connection with general purpose and router linux hardware.

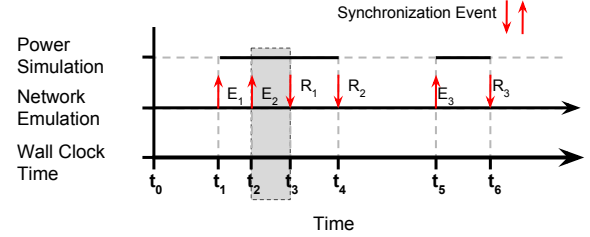


Figure 2: The execution of the system is shown with respect to the wall clock. The network (emulation) runs concurrently with the power simulation, and is not paused which allows for synchronization errors to occur, when requests arrive before the responses are sent, i.e.,  $R_1$  occurs after  $E_2$ . The shaded box highlights the location of the error.

A simulation system can be thought of a queue of events with associated time stamps. Each event may change the state of the system, and create new events. When a simulator progresses in time, it simply processes the next event and moves the simulation clock to the time of the event. In the case of time-step simulation, the maximum progression of time is the size of the time-step.

On the other hand, emulation and real systems' clocks progress with the real wall clock time. This difference in the systems can lead to temporal error. To remedy this problem we utilize the virtual time kernel [11], [12] based off [7] and expand it to work on Arm32 processor family. Additionally one of the major design differences is the distributed nature of our communication network.

In Figure ??, there are three cross-system events ( $E_i$ ), each with a response ( $R_i$ ).  $E_1$  occurs before  $E_2$ , however,  $E_2$  may require information from  $R_1$ . Since the response occurs after the second event, the global causality is violated, and thus reduces experiment fidelity. An example of 1 is a request to retrieve power flow values while  $E_2$  sets the value of a discharging battery based on the value returned previously. Since the reply  $R_1$  occurs after  $E_2$  this can introduce

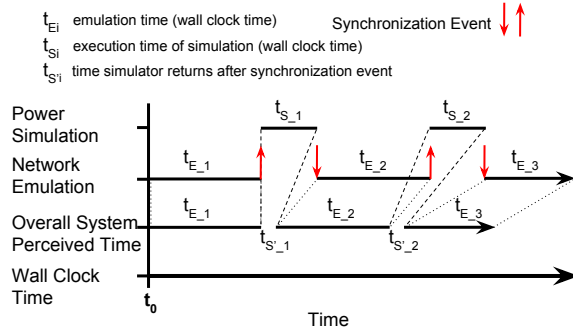


Figure 3: The execution of the system is shown with respect to its own perceived time, i.e., the sum of the emulation execution time (can be dilated or not dilated) and the virtual time elapsed in simulation. The network emulation is paused to allow for the simulation to catch up to the emulation—this also ensures synchronization errors in the early example do not occur.

an error. Furthermore, such errors can be accumulated if the simulation keeps out of synchronization with the emulation.

## 4. IMPLEMENTATION

## 5. EVALUATION

## 6. CONCLUSION AND FUTURE WORK

We present an distributed embedded linux device testbed, a testing platform that combines an electrical power system simulator and real communication network and distributed emulator platform. This testbed can be used to model and simulate power flows, communication networks, and smart grid control application, and to evaluate the effect of network applications on the smart grid. Our future work includes exploring means to extract emulation lookahead to improve the performance of this hybrid system, as well as evaluating the scalability of the testbed for large-scale experiments. We also plan to investigate several novel SDN applications for microgrid security and resilience, such as context-aware intrusion detection with PMU networks. Finally to improve the performance of the virtual time system we plan to explore real-time linux operating systems including direct control over the linux scheduler.

## 7. ACKNOWLEDGEMENTS

Thanks to Jiaqi Yan for his original Virtual-Time kernel source code and his assistance. Thanks to professor Lan for providing us with the knowledge of distributed algorithms and systems. Thanks to Professor Kevin Jin for his support and funding the testbed.

## 8. REFERENCES

- [1] S. Ciraci, J. Daily, K. Agarwal, J. Fuller, L. Marinovici, and A. Fisher. Synchronization algorithms for co-simulation of power grid and communication networks. In *Modelling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2014 IEEE 22nd International Symposium on*, pages 355–364, Sept 2014.
- [2] S. Ciraci, J. Daily, J. Fuller, A. Fisher, L. Marinovici, and K. Agarwal. Fncs: A framework for power system and communication networks co-simulation. In *Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative*, DEVS '14, pages 36:1–36:8, San Diego, CA, USA, 2014. Society for Computer Simulation International.
- [3] C. Dufour and J. Belanger. On the use of real-time simulation technology in smart grid research and development. *Industry Applications, IEEE Transactions on*, 50(6):3963–3970, Nov 2014.
- [4] M. D. O. Faruque, T. Strasser, G. Lauss, V. Jalili-Marandi, P. Forsyth, C. Dufour, V. Dinavahi, A. Monti, P. Kotsampopoulos, J. A. Martinez, K. Strunz, M. Saeedifard, X. Wang, D. Shearer, and M. Paolone. Real-time simulation technologies for power systems design, testing, and analysis. *IEEE Power and Energy Technology Systems Journal*, 2(2):63–73, June 2015.
- [5] 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*, pages 291–296, Oct 2010.
- [6] 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*, 21(2):548–558, May 2006.
- [7] J. Lamps, D. M. Nicol, and M. Caesar. Timekeeper: A lightweight virtual time system for linux. In *Proceedings of the 2Nd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM PADS '14, pages 179–186, New York, NY, USA, 2014. ACM.
- [8] H. Lin, S. Veda, S. Shukla, L. Mili, and J. Thorp. Geco: Global event-driven co-simulation framework for interconnected power system and communication network. *Smart Grid, IEEE Transactions on*, 3(3):1444–1456, Sept 2012.
- [9] K. Mets, J. Ojea, and C. Develder. Combining power and communication network simulation for cost-effective smart grid analysis. *Communications Surveys Tutorials, IEEE*, 16(3):1771–1796, Third 2014.
- [10] D. Montenegro, M. Hernandez, and G. Ramos. Real time opendss framework for distribution systems simulation and analysis. In *Transmission and Distribution: Latin America Conference and Exposition (T D-LA), 2012 Sixth IEEE/PES*, pages 1–5, Sept 2012.
- [11] J. Yan and D. Jin. A virtual time system for linux-container-based emulation of software-defined networks. In *Proceedings of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM PADS '15, pages 235–246, New York, NY, USA, 2015. ACM.
- [12] J. Yan and D. Jin. Vt-mininet: Virtual-time-enabled mininet for scalable and accurate software-defined network emulation. In *Proceedings of the 1st ACM*

*SIGCOMM Symposium on Software Defined  
Networking Research*, SOSR '15, pages 27:1–27:7, New  
York, NY, USA, 2015. ACM.