GECO: Global Event-Driven Co-Simulation Framework for Interconnected Power System and Communication Network

Hua Lin, *Student Member, IEEE*, Santhosh Veda, Sandeep Shukla, *Member, IEEE*,
Lamine Mili, *Senior Member, IEEE*, and James Thorp, *Life Fellow, IEEE*

***Abstract*—The vision of a smart grid is predicated upon pervasive use of modern digital communication techniques to today's power system. As wide area measurements and control techniques are being developed and deployed for a more resilient power system, the role of communication network is becoming prominent. Power system dynamics gets influenced by the communication delays in the network. Therefore, extensive integration of power system and communication infrastructure mandates that the two systems be studied as a single distributed cyber-physical system. This paper proposes a power system and communication network co-simulation framework (GECO) using a global event-driven mechanism. The accuracy is tunable based on the time-scale requirements of the phenomena being studied. This co-simulation can improve the practical investigation of smart grid and evaluate wide area measurement and control schemes. As a case study, a communication-based backup distance relay protection scheme is co-simulated and validated on this co-simulation framework.**

***Index Terms*—Co-simulation, event-driven, wide area protection and control.**

I. Introduction

THE MODERN power system has advanced to the point where the system can no longer be operated without wide area control systems [1], [2]. The advent of system infrastructure restructuring and the vision of smart grid have further prompted the concomitant control systems to reach an unprecedented level of sophistication. It can be predicted that more such state-of-the-art computation and communication techniques will be integrated into the power system to carry the control system from local to wide area scope. As a result, the power system will be operated and controlled with the help of an underlying communication network where large amount of information will be exchanged. This new interdependent configuration of the power system and communication network brings challenges which have not been seen before. The structure of the communication network to be laid out in the national power grid, the communication protocols to be used, the physical media, the distributed algorithms to make decisions on power system state and required control actions, the hierarchy of communication and control network, and many other issues remain unsettled to date. This mandates that we need power system and communication network co-simulation as opposed to only a stand-alone power grid simulator. It is prudent that we take into account these considerations during the designing phase.

A report prepared by the US Department of Homeland Security [3] advocates the need for a national power grid simulator. It is recommended that such a simulator should allow for modeling various possible disruptive events, studying interdependencies between the power grid and other critical infrastructures, and allow for planning and design of smarter capabilities of the national grid to enhance resilience, robustness, integration of renewable energy sources. Even though this report does not specifically deal with the embedded computational and communication capabilities envisioned for the future smart grid, there are already many efforts worldwide to enable various power grids to communicate data in real-time over wide areas, and use networked and distributed control to avoid various disastrous scenarios including blackouts, unwarranted generation shutdowns, unwarranted frequency excursions, inter-area oscillations, voltage instability, and so on.

If we can implement an effective, scalable and efficient power grid and communication network co-simulator, we can design wide-area measurement and control schemes that have hitherto not been considered yet, and easily simulate its effectiveness and optimize the design and cost. For instance, PMU-based wide area measurement systems (WAMS) would have readily benefit from such a simulator [4]. However, it is not an easy task to get this done, as there is a mismatch in the models of computation in the two simulation worlds [5]. Continuous time simulation of power system and discrete event simulation of communication network have to be seamlessly integrated and this is the key issue that is being discussed in this paper.

The following paper is organized as follows: Section II summarizes the related work. Section III presents the co-simulation framework GECO and an implementation of it using PSLF and NS2. A communication-based backup relay protection scheme is discussed as a case study on GECO in Section IV. Several co-simulation scenarios and simulation results are shown in Section V to demonstrate the effectiveness of our method. Section VI summarizes the discussion and concludes the paper.

Manuscript received October 09, 2011; revised February 02, 2012; accepted March 14, 2012. This work was supported in part by the National Science Foundation under Grant NSF EFRI-0835879. Paper no. TSG-00572-2011.

The authors are with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: birchlin@vt.edu; ssan@vt.edu; shukla@vt.edu; lmili@vt.edu; jsthorp@vt.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2012.2191805

TABLE I
COMPARISON OF INTEGRATED POWER/NETWORK SIMULATORS

	Target	Components	Synchronization	Scalability	Real-time
EPOCHS[13]	Dynamic simulation for WAMS applications	PSCAD, PSLF, NS2	Time-stepped	Good for large system	No
ADEVS[14]	Dynamic simulation for WAMS applications	Adevs, NS2	DEVS	Limited, have to rewrite codes for different systems	No
[15]	Dynamic simulation for WAMS applications	Simulink, OPNET	Not addressed	Medium size	No
VPNET[16]	Remotely controlled power devices	Virtual Test Bed, OPNET	Time-stepped	Limited to single or small number of power devices	No (but have plans to integrate RTDS)
PowerNet[17]	Remotely controlled power devices	Modelica, NS2	Time-stepped	Limited to single or small number of power devices	No
[18]	General network controlled system	OPNET only, power system part is virtualized	Delay estimation	Limited size due to virtualized power system	No
SCADA CST[19]	SCADA cyber security, system virtualization	PowerWorld, RINSE	N/A (static)	Good for large system	Yes (communication network only)
TASSCS[20]	SCADA cyber security, system virtualization	PowerWorld, OPNET	N/A (static)	Good for large system	Yes (communication network only)
GECO	Dynamic simulation for WAMS applications	PSLF, NS2	Global event-driven	Good for large system	No

II. RELATED WORK

Co-simulation of heterogeneous systems which integrates different simulation models is not a rarity in other research domains [6]–[9]. However, it is relatively new for power system applications. There has been some research on power system control analysis with consideration of communication networks [10], [11]. However, owing to lack of proper modeling tools, the characteristics of the communication networks in their research have to be either largely simplified or some very optimistic assumptions have to be made [12]. Apparently, one-sided simulation is insufficient for the investigation of fully integrated power system and communication network. In the progress of power system restructuring, co-simulation of power system and communication network has gradually become researchers' favorite. Table I summarizes the related work in this new research area.

EPOCHS [13] pioneered the efforts to build a power system modeling tool with attention to the underlying communication network. The EPOCHS approach is based on federated dynamic simulation using multiple components. Three off-the-shelf simulators: PSCAD/EMTDC for transient time scales, PSLF for power system modeling, and NS2 for computer network modeling, are configured as an integral platform. A carefully designed software mediator, called "runtime infrastructure" (RTI), is responsible for interfacing and synchronization between the individual simulators by allowing them to exchange data periodically. The synchronization algorithm is a simple time-stepped method. In this method, the individual simulators run respectively but halt at fixed synchronization points where information is exchanged between simulators. However, if certain system event which requires interacting among those simulators happens between the synchronization points, the event has to be buffered in a cache and wait to be processed until the next synchronization point. Therefore system errors could be accumulated and hamper the simulation fidelity, in particular if the application is time-critical and requires numerous interfacing between the power system and communication network. The users of EPOCHS will face a dilemma between precision and efficiency when choosing the proper size of a synchronization step.

A work similar to EPOCHS is reported in [14] where the authors try to improve the synchronization algorithm. The power system is modeled using DEVS formalism and integrated with NS2. Theoretically, this hybrid simulation environment gives better synchronization than EPOCHS since DEVS is designed for discrete event system modeling. However, the DEVS package that has been used is designed for general discrete event system and not for power system simulations specifically. Therefore, the users have to implement their own code conforming to DEVS specification for power system dynamic simulation which may affect the reliability of power system models and scalability of the hybrid simulation. At the same time, since most commercial power system modeling and simulation tools do not adopt this approach, this implementation cannot be readily applied to federations of simulators.

In [15], an integration of MATLAB Simulink and OPNET is reported to study the Information & Communication Technology (ICT) architecture's impact on the reliability of WAMS applications. The authors use OPNET to model a detailed hierarchical ICT infrastructure which includes all the processes pertaining to phasor data collection. The communication delays are tuned to study the sensitivity of PMU-based applications. Although this paper presents an interesting way showing how to use integrated simulation of power system and communication network to study system interdependence, information on synchronization method and the actual integration of the two simulators have not been addressed.

In [16], an integration of Virtual Test Bed (VTB) software and OPNET called VPNET is introduced for simulating remotely controlled power electronic devices in the system. The synchronization method used in this paper is similar to the EPOCHS's method. The co-simulation coordinator samples value from both simulators based on a global simulation time step. Therefore it accumulates the same kind of system errors as EPOCHS. Moreover, VTB is a software tool for simulating power electronics and energy systems, and so may not scale well to fit large scale power networks. So far as we are concerned with a few power devices, the magnitude of inter-device communications may not be significant. In the case study reported in this paper, the network infrastructure consists of only two nodes. A similar work called PowerNet is reported in [17] which integrates Modelica and NS2. The synchronization method and scalability feature are about the same as described above.

An extension of OPNET to simulate wide area communication network in power system is built in [18]. In this framework, the power system dynamic simulation is simplified as a virtual demander. Whenever the demander requests to transmit data on the network, it suspends itself and creates a packet in OPNET. OPNET will simulate the total communication delay of this packet and report it back to the virtual demander. At this time, the virtual demander will reactivate itself and simulate for the same time as the communication delay before further processing. In this way, no synchronization errors are accumulated. But this method is only suitable for one agent, one request scenario. If there are multiple agents in the system willing to transmit data within the same time period, this framework would fail due to single-threaded implementation. To simulate a complex hybrid system, alternative solutions are necessary.

Another kind of integration of power system and communication network is reported in [19] where a SCADA cyber security test bed is designed. The research focus of this test bed is to assess the vulnerability of the communication infrastructure of the power system to cyber attacks, and therefore static power system simulation is sufficient and synchronization considerations can be neglected. Moreover, the test bed runs on several different computers. The power system is simulated in PowerWorld software on an individual server. There are also several computers called network clients which can read data from PowerWorld through a VPN network and a real-time network simulator RINSE. The network attacks can be generated and studied as part of network simulations and the power system dynamics is not a big concern here. A very similar SCADA cyber security test bed is proposed in [20] which integrates PowerWorld and OPNET.

Most of the works reported in Table I involve the reuse of existing off-the-shelf software. This is a natural choice since they are more reliable and scalable as long as they can be properly modified and customized. Rewriting new simulation engines from scratch is costly and time-consuming. Other possible options include software/hardware hybrid emulation environments or hardware testbeds. For example, since the scale of some SCADA systems is smaller than WAMS applications, it is possible to build emulation environments for SCADA testbeds. In [21], [22], a SCADA testbed PowerCyber using scale-down field devices to represent the real system is documented. However, our co-simulation framework Global Event-driven CO-simulation platform (GECO) aims at the modeling and simulation for the wide area power system monitoring, protection and control schemes. Building hardware emulation system at the national level is prohibitively expensive. Even if it is possible to make assumptions to scale down the system, the fidelity of the emulation cannot be guaranteed. For example, the communication infrastructure dedicated to the power system could be isolated from other overwhelming networks such as Internet. The communication topology, protocols, routing scheme and background traffic at different levels can be significantly different.

Another attractive solution is to use real-time simulators to represent the real world system. RTDS is a well-known real-time power system simulator which is capable of performing closed loop testing of devices [23]–[25]. RTDS simulation related to IEC 61850 communication has been reported in [26]. However, the scale of the hardware in the closed loop is limited to local scope. Deploying RTDS simulation results on a large scale distributed network is difficult. Therefore integrating another real-time communication network simulator with RTDS will be a better option. But real-time simulators allowing open access are always rare so that this kind of real-time co-simulation implementation has not been published. Synchronizing two real-time simulators is also a challenging problem as both simulators are synchronized to real world clock. This requires that a real-time simulation coordinator be designed to exchange information between the simulators. Nevertheless, Real-time co-simulation platform will draw more interest in the future.

III. CO-SIMULATION FRAMEWORK

The key issue of the power/communication co-simulation framework is to accurately synchronize the simulation time in two distinct simulation models. In this section, the simulation techniques for power system and communication network will be briefly reviewed. Then the co-simulation framework will be introduced based on careful analysis of those simulation techniques.

A. Power System Dynamic Simulation

Power system dynamic simulation is commonly modeled as a continuous time system simulation. In a continuous time system, the system state variables change in a continuous manner with respect to time. Typically the system dynamics is represented by a set of differential equations in which the transitions between continuous state variables are defined. For simple cases, the differential equations can be solved analytically to get closed form solutions. However in most cases such closed form solutions are not available. Instead, numerical algorithms are studied for general cases. Usually the differential equations are discretized and the time base is divided into small steps. The next system state is derived from current system state. Then the small variations of the state variables are integrated to approximate the system trajectory. The discretized time step is often very small so that the system variables do not have an abrupt transition within the time step.

An example of this numerical algorithm for power system dynamic simulation is illustrated in Fig. 1. The system is ini-

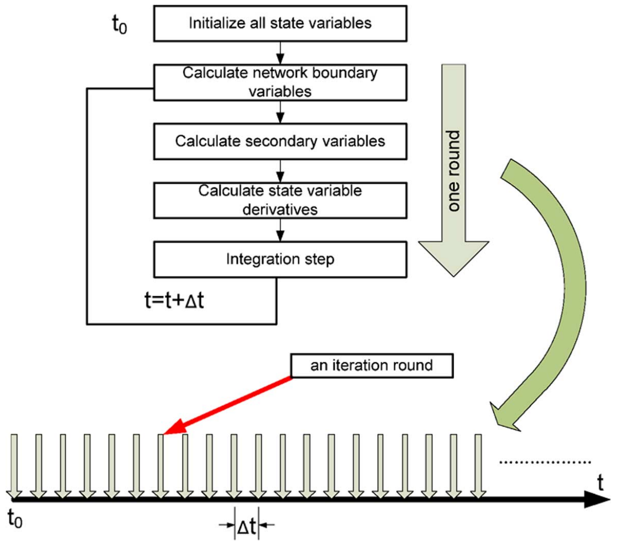


Fig. 1. Example of the power system dynamic simulation.

tialized by solving power flows which calculates initial system state values. Then the simulation enters a loop which represents the main part of the algorithm. Within this loop, the network boundary variables for dynamic models connected directly to the system network are calculated. Then the secondary variables of dynamic models are calculated from system state variables. A complete iteration in this loop is completed by calculating state variable derivatives and integrations. At this point, the system time is advanced by a preset time step or a time calculated from the current system state. The loop continues until the simulation reaches the stop time or an accept state. Alternatively, if the simulation loop is expanded on a time axis which is shown in Fig. 1, a sequence of discrete iteration rounds can be found with small time intervals in between. This sequence actually shows that a continuous time system in fact is numerically solved in a discrete manner.

B. Communication Network Simulation

Communication network simulation is usually performed using a discrete event-driven method. Discrete event-driven simulation is suitable for systems whose state is only subject to change due to discrete events. The occurrences of events are usually unevenly distributed with respect to time. Time discretization into small time intervals as done in continuous time systems cannot be appropriately applied to discrete event systems since the time step is difficult to select. If the time step is selected too small then it will waste simulation times since system state remains unchanged during many consecutive time steps. If the time step is selected too long, then many events could be missed during a single time step. Instead, in discrete event-driven simulation, the system time instead hops between events. An event scheduler is designed to record current system time and also to maintain an event list. Event list is a queue that stores system events with timestamps in a chronological sequence. The scheduler initializes the system state and the event list in the beginning of the simulation. When the simulation starts the scheduler proceeds with the event on top of the list and sorts out the relevant processes. Then the scheduler adjusts the system time directly to the timestamp of the next event in the list. The entire simulation stops when the system time reaches the stop time or the system reaches a certain state. Fig. 2 shows an example of a communication network simulation that uses event-driven method. When the simulation starts, node 1 sends a packet to node 4 via node 2. The first event in the list should be “node 1 sends a packet to node 2” with its timestamp. A receiving event by node 2 is predicted based on the communication link properties. Then the second event “node 2 receives a packet from node 1” will be created and placed in the event list. The simulation will continue this way until the ending criteria is satisfied.

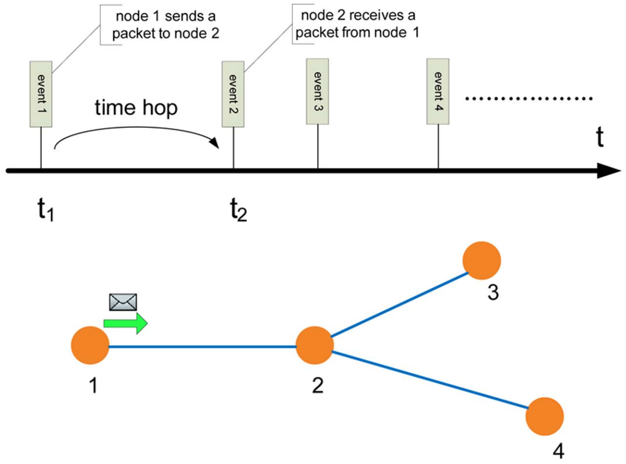


Fig. 2. Example of the communication network simulation.

C. Co-Simulation Framework

Since the simulation techniques for power system and communication network are different, synchronization mechanism between them is the most crucial issue leading to a successful co-simulation design. An intuitive method is to use explicit time-stepped synchronization [13] as shown in Fig. 3. In this method, several synchronization points are predefined. In Fig. 3, the top axis represents the power system dynamic simulation process and the bottom axis represents the communication network simulation process. When the co-simulation starts, two processes run independently until both of them reach a synchronization point, as denoted by dashed vertical lines. It is here that the two processes suspend themselves and exchange information. Typical interaction information includes power devices uploading data to control center or smart relay receiving a remote command to trip the circuit breaker. After that, two processes restart and repeat the synchronization as done before.

This synchronization method can easily bring in simulation errors. If an interaction request appears between the synchronization points, it has to wait until the next synchronization to be processed. This problem is indicated by “Error 1” and “Error 2” in Fig. 3. These errors create unwanted time delays which do not exist in a real system and might accumulate over time. Theoretically, each error can be the same as one synchronization time step.

A new co-simulation framework is accordingly proposed which avoids these synchronization errors. Our co-simulation runs globally in a discrete event driven manner as shown

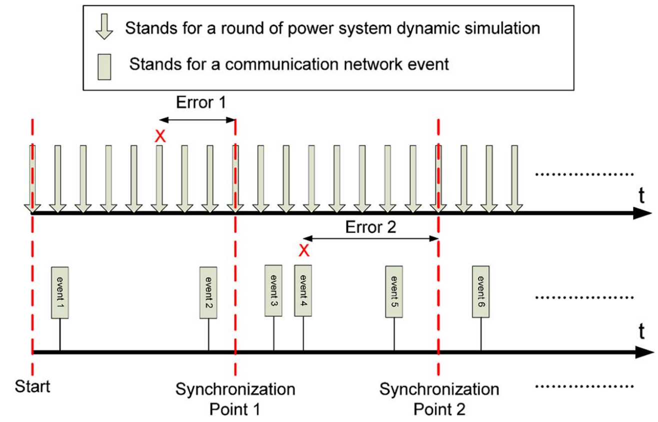


Fig. 3. Synchronization with errors.

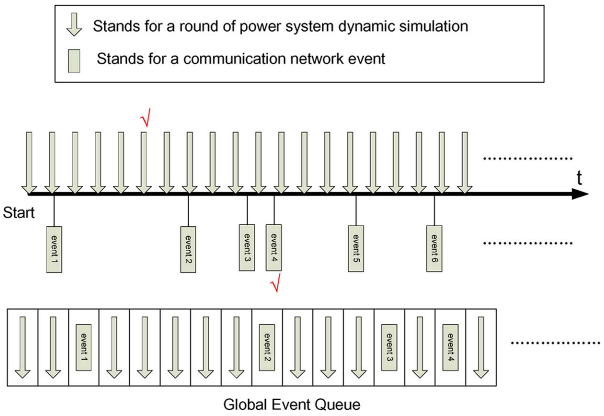


Fig. 4. Event-driven synchronization without errors.

in Fig. 4. Since the power system dynamic simulation is in fact solved in discrete manner as shown in Fig. 1, each of the iteration rounds is treated as a special discrete event in this framework. A global event scheduler is designed as the global time reference and coordinator. A global event list is also prepared by mixing up the power system iteration events with other communication network events according to their timestamps. Therefore, only one event process is allowed to run at the same time. This is illustrated by the only axis in Fig. 4. The global scheduler checks the global event list to identify if the next event is a power system simulation event or a communication network event and yields the control accordingly. More importantly, the simulation processes can suspend themselves after each event and yield the control back to the global scheduler. In this way whenever there is an interaction request, it can be processed immediately by the global scheduler without unnecessary time delay. Both errors in Fig. 3 are eliminated in this framework.

D. Formalism

It is necessary to show that our global even-driven co-simulation does not undermine the simulation integrity in each of the individual simulator since all the events are mixed up. In this subsection, we will verify it using a formal approach.

Discrete Event System Specifications (DEVS) is a popular formalism to model and analyze general discrete event systems. There are also many other equivalent formalisms but DEVS is more suitable for this co-simulation framework. It is defined as a 7-tuple [27]:

$$M = \langle X, Y, S, ta, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda \rangle \tag{1}$$

where:

X	is the set of input events;
Y	is the set of output events;
S	is the set of system partial states;
$ta{:}S \rightarrow T^{\infty}$	is the lifespan function of the partial state;
$\delta_{\text{int}}{:}S \rightarrow S$	is the internal transition function;
$\delta_{\text{ext}}{:}Q \times X \rightarrow S$	is the external transition function;
$\lambda{:}S \rightarrow Y$	is the output function.

($Q = \{(s,e) | s \in S, e \in (T \cap [0, ta(s)])$ is the set of total states including e which is the time elapsed since last transition; $T = [0, \infty); T^{\infty} = [0, \infty]$)

The interpretation of (1) for a communication network simulation is straightforward where X, Y are still the system input/output; S is the system state when a certain discrete event is being processed; ta represents the time delay between the current event and the next event in the event list. δ_{int} stands for the relevant processes associated with an event where δ_{ext} stands for the impact of the input to the system state. Even-driven simulation is commonly used for the system which can be modeled by DEVS.

It has also been shown that the power system dynamic simulation can be modeled by DEVS [14]. For this particular case, S is the set of system state variables (voltage, current, etc.) after each iteration round. ta represents the iteration time step and δ_{int} stands for the system change after the integration of each time step.

The co-simulation framework in fact couples the power system and communication network together. That is: the output event of the power system simulation is the input event of the communication network simulation and vice versa. From the DEVS formalism point of view, these two atomic DEVS systems actually form a coupled-DEVS which is another 7-tuple [27]:

$$N = \langle X, Y, D, \{M_i | i \in D\}, \text{ EIC}, \text{ ITC}, \text{ EOC}, \text{Select} \rangle \tag{2}$$

where:

X	is the set of input events;
Y	is the set of output events;
D	is the name set of sub-components;
$\{M_i\}$	is the set of DEVSs that form the coupled-DEVS;
$\text{EIC} \subseteq X \times \bigcup_{i \in D} X_i$	is the set of external input couplings;

$\mathrm{ITC} \subseteq \bigcup_{i \in D} Y_i \times \bigcup_{i \in D} X_i$	is the set of internal couplings;
$\mathrm{EOC} \subseteq \bigcup_{i \in D} Y_i \times Y$	is the set of external output couplings;
Select	is a tie-breaker function for time conflict of events.

The couplings define how the atomic DEVSs are connected to form a coupled-DEVS. For the co-simulation framework, D will be $\{P, C\}$ which represents power system and communication network respectively. M_p, M_c will be the DEVS models for them. EIC and EOC will be both empty and ITC will be $(Y_p, X_c) \cup Y_c, X_p)$.

It has been proved that DEVS is closed under coupling which means a coupled-DEVS N is equivalent to a DEVS M'. The proof can be done by construction [27] where:

$$\begin{aligned}
X_{M'} &= X_N \\
X_{M'} &= X_N \\
S_{M'} &= \times_{i \in D} Q_i \ (Q_i \text{ is the total} \\
&\quad \text{state set of each component}) \\
ta_{M'}(S_{M'}) &= \min\{ta_i(S_i) - e_i | i \in D\} \\
\delta_{\mathrm{ext}_{M'}} &= (\ldots, (s_i, e_i), \ldots) \text{ when EIC is empty} \\
\delta_{\mathrm{int}_{M'}} &= (\ldots, \delta_{\mathrm{ext}_i}, \ldots) \text{ if it} \\
&\quad \text{is an internal coupling or} \\
\delta_{\mathrm{int}_{M'}} &= (\ldots, (s_i, e_i), \ldots) \text{ for other cases} \\
\lambda_{M'} &= \mathrm{EOC}(\lambda_i(s_i))
\end{aligned}$$

The new $ta_{M'}$ is the lifespan of the new partial state considering that event from other DEVSs can potentially reduce its own original lifespan. This equivalent DEVS M' can also be simulated using event-driven method. Therefore the integrity of the individual simulators still holds under our co-simulation framework. The proof also indicates that global event-driven simulation is an effective approach for the interconnected power system and communication network.

E. Implementation

The co-simulation framework is realized by carefully integrating two individual simulators: GE's Positive Sequence Load Flow (PSLF) and Network Simulator 2 (NS2). The integration involves major modifications and extensions on both parts. The simulators we choose are the same as EPOCHS [13] but our internal design is different and the difference will be shown in later sections through comparison of simulation results.

PSLF is a power system simulator designed by GE which provides both steady-state and dynamic power system simulations. PSLF is able to simulate a system with up to 60 000 buses and is equipped with a rich library of power system dynamic models. The software is written in Java and provides plenty of APIs in the format of a script language called EPCL for further customized extensions. New models written in EPCL can also be integrated into the existing software package. EPCL can access the runtime simulation data and change the simulation settings as needed. Although PSLF is not an open-source software, its design feature enables users to build flexible extensions.

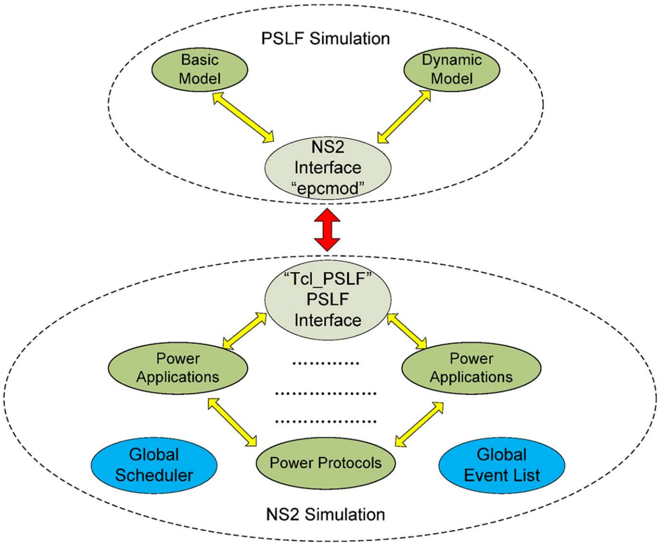


Fig. 5. The structure of the co-simulation framework.

NS2 is a well-known communication network simulator aiming at the evaluation of network performance. It is open-source and thus widely used in networking research domain. NS2 is basically a general discrete event simulator with a rich library of network models which covers four protocol layers in the network reference model excluding the physical layer. The core of NS2 is written in C++ which is complex. Therefore a script language called Object Tcl (OTcl) is provided to users for easier simulation configuration and reuse. A framework called "OTcl linkage" links the OTcl codes to the background implementation by C++. Following this way, users can write and compile new network protocols or models in C++ and manipulate them in OTcl.

Fig. 5 shows the structure of our co-simulation framework implementation. The global scheduler and global event list are derived directly from the counterpart in NS2 so that a subcomponent in NS2 drives the whole co-simulation overall. A bi-directional interface is designed between NS2 and PSLF to exchange information.

On the PSLF side, a new dynamic model "epcmod" is added as the main port to the NS2. Within each iteration round, this model updates all the power data for NS2 and receives feedbacks from NS2 to change the settings of the power system accordingly. After each round, it is also able to suspend the PSLF simulation, yield the control to the global scheduler and wait for the command to run the next round.

On the NS2 side, a new C++ class "tcl_PSLF" is written to drive the simulation of PSLF and coordinate the actions in between. This new class is independent from all the other networking classes but still compiled together with other components in NS2. When the simulation starts, this class pre-allocated a sequence of power system iteration rounds and put them in the global event list. When an iteration round needs processing, it sends the command to PSLF to restart the suspended simulation. Potential network-based power system control strategies are designed in the power application classes which are derived from class "Application" in NS2. These classes represent the functionalities of the software

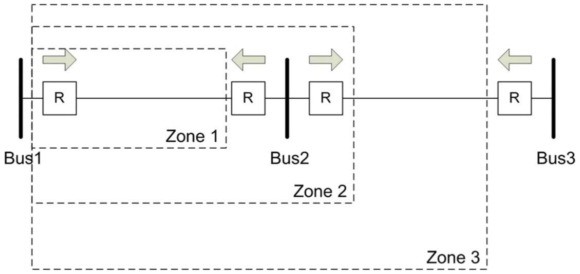


Fig. 6. Distance relay protection zones.

agents in current power devices such as digital relays, Phasor Measurement Units (PMUs) and Intelligent Electronic Devices (IEDs). The power data updated from PSLF will be distributed to them for further analysis. The software agents are able to communicate with each other through the network infrastructure modeled in NS2 and control decisions are also made by them. The communication protocols of the power applications are variants of existing famous network protocols like UDP and TCP. Minor changes are applied to UDP and TCP classes to enable them to carry power data.

In the following sections, a communication-based distance relay backup protection scheme will be introduced as a case study on this co-simulation platform. The purpose of the case study is to show what kinds of applications are suitable for co-simulation and how to use co-simulation to test a design and to study system interdependences.

IV. COMMUNICATION-BASED DISTANCE RELAY BACKUP PROTECTION

Distance (impedance) relays are usually utilized on the transmission system level. The operation of the distance protection relays is governed by apparent impedance which is the ratio of the magnitudes of voltage and current measured by the relay. When a short circuit fault occurs, the fault can be identified by a sharp drop in the measurement apparent impedance. Also this impedance tells the relay the distance from the fault location to itself.

As shown in Fig. 6, it is common practice to assign three protection zones for the distance relays. Zone 1 protection is the primary protection for each distance relay. It covers about 80%–90% of the length of the first transmission line as shown in Fig. 6. Zone 2 protection covers a little bit longer than zone 1 extending beyond the bus 2 which is about 120% of the length of the first line. Zone 3 protection provides the longest coverage which includes the entire first line and about 80% of the second line. By properly adjusting zone 1, zone 2 and zone 3 settings the distance relays can achieve both primary and backup protection of the transmission lines. Usually zone 1 protection operates instantaneously while zone 2 and zone 3 protections are associated with time delays as backups. It is common practice to use longer time delay for longer reach of the relays so that they can provide effective system protection without unnecessary power loss. The time delay can be as long as 1 second for zone 3 relays.

Although the transmission systems with protective relaying usually have the redundancy in the form of backup relays, it is reported that such a system still may suffer from different kinds of failures. For instance, zone 3 backup relays work in a time-delayed manner, the system may encounter instability issues during the delay. It is also known that zone 3 relays can actually erroneously trip due to hidden failures [28]. A hidden failure is usually rare but could happen due to software or hardware errors in the zone 3 relay. It may go unidentified for a long time. However, such problems may manifest as extra sensitivity of a Zone 3 relay to even remote line overloading. Even though such an overloading might be transient, or might not have reached a level where the zone 1 and zone 2 relays need to take action, an over active zone 3 relay may trip, starting a sequence of other trips which may lead to a cascading failure [29], [30]. New protection techniques are being sought out to solve this problem [31]–[37]

Accordingly a new communication-based distance relay backup protection scheme is introduced in this section that leverages the present distance relay protection framework with the addition of an underlying network infrastructure. Modern microprocessor-based digital relays are more reliable and efficient than traditional electromechanical ones, thus it is possible to enhance them with software agents in order to design more elaborate protection schemes. The distance relays can communicate with each other through their software agents from which a coordinated system protection scheme can be formed. By virtue of extensive communication new protection schemes could have faster backup relay protection and additional robustness to prevent false tripping. Based on the communication type, two related protection schemes are discussed: supervisory (master to slave) and ad-hoc (peer to peer).

A. *Supervisory Protection*

In the supervisory protection scheme, distance relays are inter-connected as a network using a communication infrastructure and their functioning is coordinated through extensive communication links. A central protection controller called “Master Agent” coordinates the operations of the digital relays in the system. Each distance relay has a software agent (Slave Agent) associated with it. This software agent works as an interface for information exchange between the Master & the corresponding distance relay. In this mode, the protection scheme system is able to provide more secure protection by avoiding hidden failure induced false tripping.

The primary protection for the distance relays remains the same as the traditional distance protection scheme, while the backup protection is different. In this scheme, when a backup relay sees zone 2 or zone 3 faults, instead of waiting for a pre-set time delay to trip, the relay proactively collects information from other relays to evaluate the status and make the decision. This procedure is done by communication between slave agents and the master agent. Firstly, the slave agent whose associated relay sees a remote fault submits a request to the master agent for decision. The master agent then asks other slave agents in the fault zone to see if others see the fault as well. Based on the feedback from other slave agents, the master agent sends the final decision to the original slave agents.

Detailed operations of this master-slave mode supervisory protection scheme are shown in Figs. 7 and 8 using a finite state machine (FSM) representation. In the FSM, a circle represents

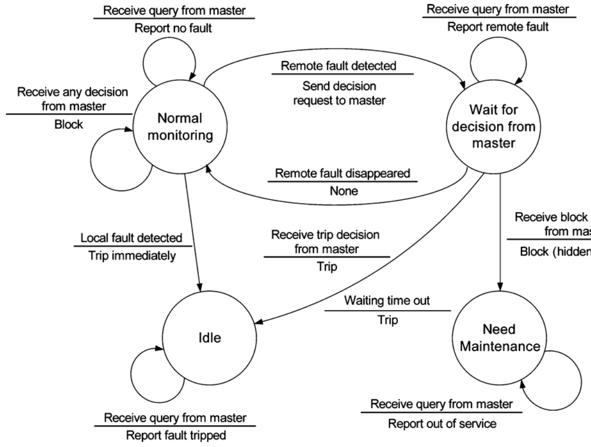


Fig. 7. FSM of supervisory protection: slave agent.

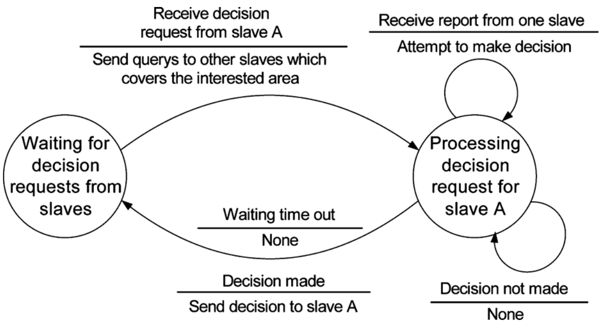


Fig. 8. FSM of supervisory protection: master agent.

a certain state. An arrow line represents a transition from one state to another. There is a fraction associated with each transition. The numerator position shows the event which causes the transition and the denominator position shows the action taken due to that event.

On the slave agent side, the relay starts from the "normal monitoring" state and keeps monitoring the transmission line. If a zone 1 (local) fault is observed the relay should trip the transmission line immediately. If a zone 2 or zone 3 fault (remote) is observed, the slave agent sends a decision request to the master agent and enters the "wait for decision" state. If the fault disappears during the wait state, the slave agent would go back to the normal state and block whatever decision received from the master agent since this condition indicates the fault may have been cleared by its own primary protection. If the fault persists and the slave agent receives a trip decision from the master agent, the relay should trip the line since the primary protection may have failed. If the slave agent receives a block decision from the master agent but still sees the fault, this indicates the relay may have a hidden failure or wrong setting. Then the slave agent should put the relay out of service and call for maintenance. In this manner the slave agents can both expedite the backup protection and prevent hidden failure induced false tripping.

On the master agent side, when it receives a decision request from a slave agent, it enters the "processing decision request" state. A group of relays which are entrusted with the fault area are selected and queried by the master agent. When the software agents of the selected relays receive the queries, no matter which state they are in, they should report if they see a fault back to the master agent. The master agent will try to make the final decision when it receives a feedback. As long as a final decision could be made, the master should send it to the original slave agent to take action. Fig. 8 only shows the master agent operation for one slave agent. Actually when a fault happens in the system, multiple slave agents could send request to the master agent. Therefore, the master agent should be a multi-threaded program which can handle all the requests simultaneously.

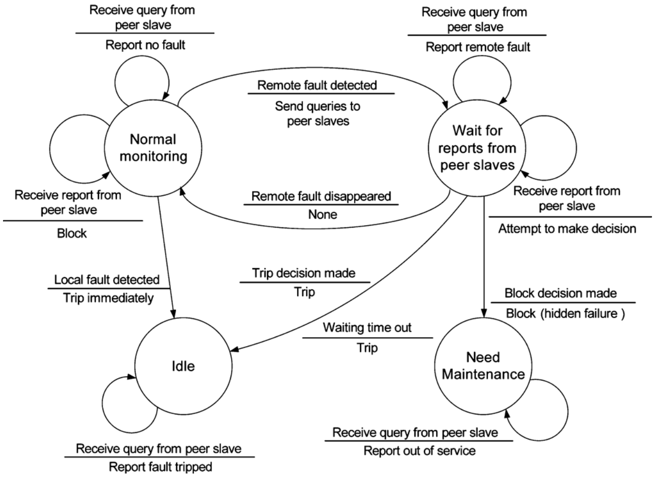


Fig. 9. FSM of ad-hoc protection.

Although there is extensive communication, the total communication time could still be shorter than the traditional time delay settings for the zone 2 or zone 3 protections. However the time delay associated with zone 2 or zone 3 should not be eliminated since the network itself may fail. Either link failure or traffic congestion may significantly increase the communication delay or even result in messages dropping. Hence if the communication-based protection cannot complete within a certain time, the relays would revert to the traditional distance protection mode.

B. Ad-Hoc Protection

In the supervisory protection scheme, the master agent is the most crucial component since it coordinates all the slave agents. If the master agent fails, the entire protection scheme fails. Another issue of the supervisory scheme is that the slave agents always communicate with the master agent. This could lead to long and unstable communication times, depending on how far the slave agent is from the master agent. In order to overcome these difficulties, an ad-hoc protection scheme is considered. In this scheme, the master agent is removed and its functions are duplicated in every slave agents. Now the slave agents can directly communicate with each other in a peer-to-peer manner.

Fig. 9 shows the FSM representation of the ad-hoc protection operations. The only type of agent in this scheme is the peer agent. Each peer agent actually combines the operations of the slave agent and the master agent. The main difference is that when a peer agent sees a remote fault, it queries other peer agents in its zone directly. On receiving a report from other peer agents, the peer agent makes the decision on its own. Hence this is a fully distributed and autonomous application based on ad-hoc communication.

```
Algorithm
Input:  A modified undirected system graph $G(V,E)$
        A relay represented by $((m,n),m)$
Output: A relay set $R$
Steps:
  1. Find the possible faulted lines set $L$: for each edge
     $(u,v) \in E$ except $(m,n)$, if $n=u$ or $n=v$, add
     $(u,v)$ to $L$
  2. Find the responsible relays for each line in $L$:
     a. For each $(u',v') \in L$, add $((u',v'),u')$ and
        $((u',v'),v')$ to $R$
     b. If $n=u'$, for each edge $(u'',v'') \in E$ except
        $(u',v')$, if $v'=u''$ add $((u'',v''),v'')$ to $R$, if
        $v'=v''$ add $((u'',v''),u'')$ to $R$
     c. If $n=v'$, for each edge $(u'',v'') \in E$ except
        $(u',v')$, if $u'=u''$ add $((u'',v''),v'')$ to $R$, if
        $u'=v''$ add $((u'',v''),u'')$ to $R$
```

Fig. 10. The steps of relay searching.

C. Relay Search and Decision Making

In both protection schemes, a relay search procedure is required for the agents to determine the responsible relay group when a fault is observed. A relay searching algorithm is implemented on a graph abstraction of the power system topology. The step by step instruction of this algorithm is shown in Fig. 10. The power system topology is represented by an undirected graph $G(V,E)$. All the transmission lines are represented by edges and the buses connecting to transmission lines are represented by vertices. The relay in this graph can be represented by an ordered pair $((m,n),m)$ which means the relay locates at the side of bus m of the transmission line (m,n). The algorithm basically consists of two major steps. First, based on the relay who submits the decision request, the algorithm find out the possible faulted lines. Then for each possible faulted line, the algorithm finds out two primary protection relays and all the backup relays for this line.

As long as the responsible relay group is determined, the protection decision will be made based on the feedbacks from this group. Since relay protection is a time-critical application, a simple but effective decision making method is applied to the agents:

a. Whenever a second relay sees the fault, a trip decision is made for the relay who submits the request.
b. If and only if none of the relays see a fault, a block decision is made for the relay who submits the request.

V. CO-SIMULATION RESULTS

In this section, the communication-based protection schemes are validated and studied on the co-simulation platform GECO.

A. Simulation Settings

The protection schemes are applied to the New England 39-bus system. In this benchmark, there are in total 34 transmission lines and consequently 68 distance relay agents are placed in the system—two for each line. The 10 generators in the system are cylindrical rotor machines represented by equal mutual inductances on the direct and quadrature axes. Each generator is equipped with an IEEE type 1 excitation system model with added speed multiplier and basic steam turbine and governor. The PSLF simulation time step is set as 0.001 s

The relay agents are connected with each other by a communication infrastructure which is modeled in NS2. This communication infrastructure contains two levels: substation level and wide area level. We assume that Ethernet is adopted for the local area network (LAN) for each substation. Then at the substation level, all the relay agents in the same substation share a 100 Mbps Ethernet. For example, there are 5 transmission lines connected to bus 16, then 5 distance relay agents should be placed in bus 16 and connected by an Ethernet. In NS2, these relay agents are represented by individual network nodes in the Ethernet model. The relay agents can communicate with other relay agents at different substations via a gateway router. On the wide area level, the substations are connected by high speed direct communication links. These links have the same topology as the transmission lines. Each communication link is of 1 Gbps bandwidth and 5 ms communication delay. These parameters stay constant through the simulation. Since the size of messages exchanged among relay agents are small [10], UDP is selected as the main transport protocol between them. The network is assumed to be dedicated to the protection scheme so that no background traffic is considered at this stage. However its effect can be easily evaluated in NS2 as long as the detailed traffic model is available.

B. Validation of Protection Schemes

In the supervisory protection scheme, the master agent is placed at bus 16 since this bus has the highest connection degree. Two different protection scenarios are co-simulated respectively:

1. There is a real fault but the primary relay fails
2. There is no fault but the backup relay has false reading

First, a real short circuit fault is created at 0.1 second on the transmission line between bus 4 and bus 14 as shown in Fig. 11. Then the primary relay covering this line at bus 4 is assumed to fail to isolate the fault so that its zone 3 backup protection relays can take action instead. In this case, the backup relays are bus 3 and bus 5 will submit requests to the master agent and wait for decision. The master agent will collect information from all the responsible relays to make a decision and send it back to the backup relays at bus 3 and bus 5. Second, there is no fault placed in the system. However, the same backup relay at bus 3 is assumed to see a fake zone 3 fault due to a false reading as shown in Fig. 12. According to the protection scheme, it will send a request to the master for decision. However, the decision is expected to be different from the first protection scenario. The same protection scenarios are also tested for the ad-hoc protection scheme. The results are compared with the supervisory case.

The simulation results of these four protection scenarios are shown in Fig. 13. From Fig. 13(a) we can see that the fault has been removed after the communication between the agents and the voltage level at bus 3 recovers back to a normal value. This

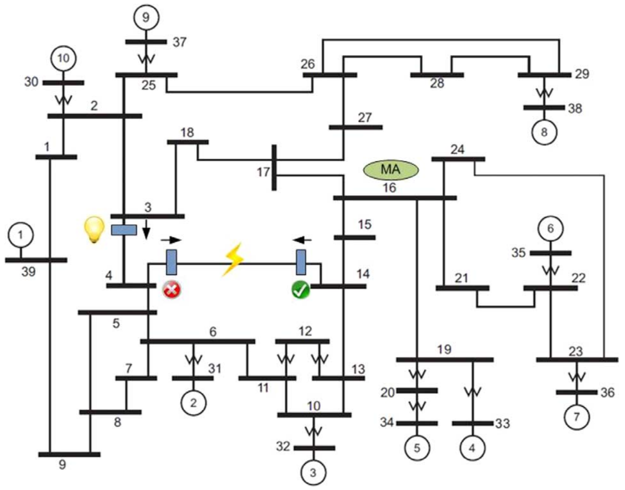


Fig. 11. Backup protection for a real fault [38].

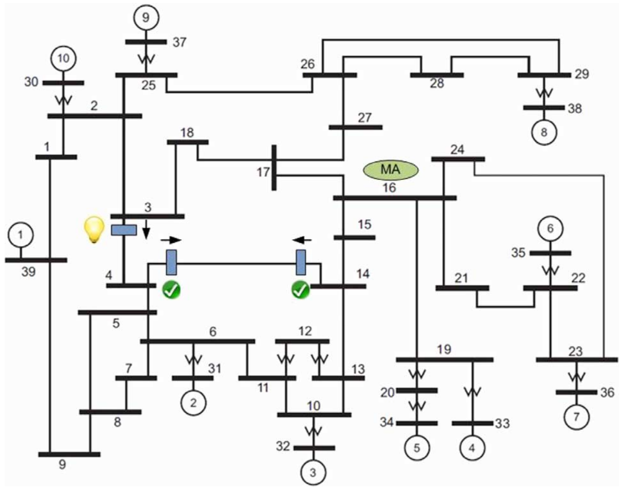
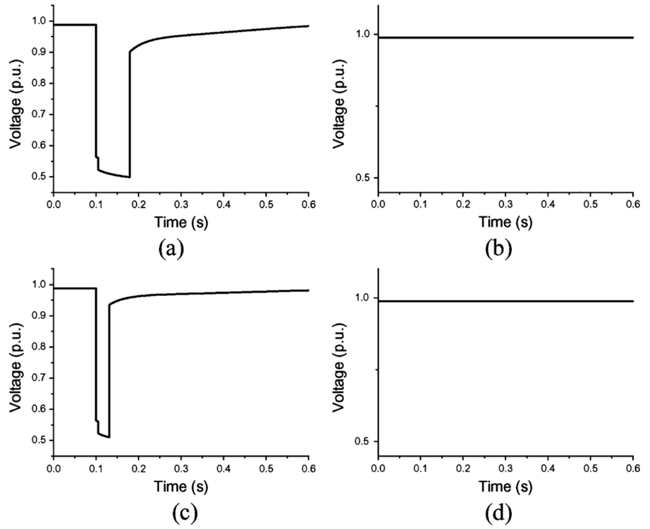


Fig. 12. Backup protection for a fake fault [38].

means that after the master agent collects information from the responsible relays, the situation is determined as a real fault and a trip decision is sent to the backup relays. Traditional backup distance relays can also protect the system from this fault but the supervisory protection is faster. As in Fig. 13(a), the fault is removed within 0.1 second. From Fig. 13(b) we can find the false-tripping by backup relay is avoided after the communication. The voltage level at bus 3 remains the same. This shows that after the master agent collects information from the responsible relays, the situation is determined as a fake fault and a block decision is sent to the backup relay with false reading. Traditional backup distance relays, however, will trip the line and may result in cascading failures due to very limited system visibility. Figure Fig. 13(c) and (d) shows the protection results for the ad-hoc protection scheme. It shows that the real fault is successfully isolated and the false-tripping is blocked as well. But the total time needed to finish the action is less than the supervisory protection scheme.

It is very important that the communication time between agents has to be limited within a certain threshold. The communication time for the four protection scenarios is shown in Table II. All of the protection actions can be completed within the general Zone 2 time delay of 100 ms and that the ad-hoc protection scheme takes significantly lesser time than the supervisory protection. Moreover, a block decision always requires longer time than a trip decision which is reasonable considering the decision making mechanism we have adopted.



(a) (b)

(c) (d)

Fig. 13. Voltage magnitude seen by the relay at bus 3: (a) supervisory protection for real fault; (b) supervisory protection for fake fault; (c) adhoc protection for real fault; (d) adhoc protection for fake fault.

TABLE II
COMMUNICATION TIME OF THE PROTECTION SCHEMES

	Trip Decision	Block Decision
Supervisory Protection	76.661 ms	96.635 ms
Ad-hoc Protection	28.290 ms	38.499 ms

C. Comparison of Different Synchronization Methods

In previous sections, the disadvantages of alternative synchronization method have been discussed. Further, the communication-based protection scheme is also experimented on a co-simulation platform using the time-stepped synchronization method [13]. The platform tested in this part is not the original platform as in [13], but instead a similarly reproduced one. The protection scenario in this experiment is the supervisory protection for a real fault. The initial fault time, fault location, master agent location and the relay agents involved are all the same. This scenario is repeated on the time-stepped synchronization platform using different synchronization steps and the results are compared with the one on GECO. As a simulation index, the voltage levels at bus 3 among all the simulation results are plotted all together in Fig. 14. From the figure, we can easily tell the difference among simulations. As the synchronization time step increases, simulation errors are accumulated and the protection action is delayed accordingly. With this delay in hand, the real system dynamics will be difficult to estimate. In general, for the time-stepped synchronization method, the larger the time step is chosen, the more inaccurate results are expected. However, in the extreme case, if the time step is as small as the power system iteration time step, this method can provide the

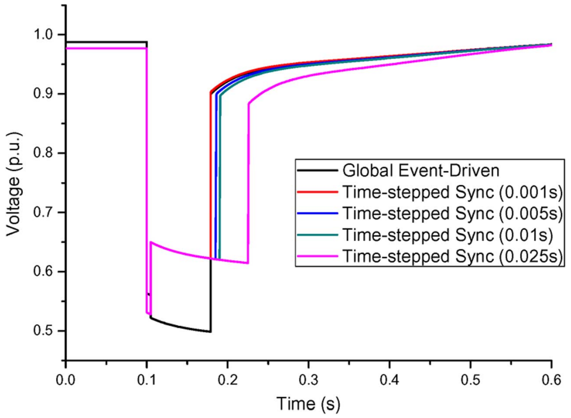


Fig. 14. Simulation results using different synchronization methods.

same simulation fidelity as the global event-driven co-simulation framework. In Fig. 14, the voltage level of time-stepped synchronization using 0.001 s time step is almost the same as GECO. This similarity further proves the advantage and necessity of our co-simulation framework for fully integrated power system and communication network.

D. Co-Simulation Scalability

The scalability of the co-simulation platform is another important factor considering the actual power system of interest can be much larger than this 39-bus benchmark. Since GECO integrates two individual simulators, the overall co-simulation scalability will be largely determined by the scalability of the individual simulators themselves and how the integration interface is handled. More specifically, in this case, PSLF is able to simulate a system as large as 60 000 buses and NS2 is able to simulate a network with at least 20 000 nodes and the simulation time is on the order of $N\log(N)$ [39]. Therefore, GECO has the capacity to model and simulate large national systems like WECC. On the other hand, the two simulators are integrated using a bi-directional interface where system information is exchanged. As the system scale grows, the amount of system information through this interface will increase accordingly. The time needed to complete a co-simulation case may also increase depending on the number of interactions between the two simulators.

As an example, the same communication-based protection scheme is implemented on a 127-bus WECC system and the co-simulation speed is compared to the 39-bus case. There are 112 transmission lines, 28 generators in this 127-bus system in comparison to 34 lines and 10 generators in the 39-bus system. The co-simulation speeds are shown in Table III. The stop time in both simulators is set as 0.5 second. Two PSLF simulation time steps: 0.001 second and 0.01 second, are selected for comparison. Smaller PSLF time step results in more discrete power system events and more interactions through the interface between the PSLF and NS2. In Table III, the total simulation time required for different settings are measured on a regular PC. It is clear that co-simulations for larger systems or with smaller power steps both require longer simulation time. However, the latter factor contributes significantly more than the former one.

TABLE III
COMPARISON OF THE SIMULATION SPEED

Run for 0.5sec	39-Bus System	127-Bus System
Power Step: 0.001sec	19min16sec	39min40sec
Power Step: 0.01sec	1min26sec	2min20sec

The co-simulation results in Table III indicate that the interface between PSLF and NS2 may be a bottleneck for GECO as far as larger scale systems are concerned. This is due to the nature that these two individual simulators are not designed for the purpose of integration with each other. The interface is mainly designed to make information exchange and global event scheduler feasible rather than to optimize the overall co-simulation speed. However, there are many potential ways to improve the co-simulation speed since PSLF and NS2 is not the only solution for GECO. Many other power system and communication network simulators can be readily integrated using GECO framework like PSS/E, InterPSS, OPNET, OMNET++ etc. Depending on the simulators, it is possible to parallel the co-simulation or use distributed resources to expedite the simulation speed [40], [41]. However it requires great support from the simulators and the coordination between simulators can be much more complicated. In the case of current implementation of GECO, PSLF is not an open source software. Therefore very limited change can be made to facilitate the speedup.

VI. CONCLUSION

In this paper a global event-driven co-simulation framework GECO is proposed which integrates the simulations of power system and communication network. Compared to other people's related work, our co-simulation framework provides better synchronization accuracy and the feasibility of this method is proved using formal method. The co-simulation framework is implemented using PSLF and NS2 software. A communication-based backup distance relay protection scheme is discussed as a case study on the co-simulation platform. In this scheme, the relay agents proactively communicate with each other to gain better system visibility and make coordinated protection decisions. The co-simulation results validate the protection scheme and the communication time needed is less than the threshold. In the end, a comparison between the global event-driven method and other synchronization methods is shown. The results present the advantages of our co-simulation framework.

REFERENCES

[1] A. G. Phadke and R. M. de Moraes, "The wide world of wide-area measurement," *IEEE Power Energy Mag.*, vol. 6, pp. 52–65, 2008.

[2] D. Karlsson, M. Hemmingsson, and S. Lindahl, "Wide area system monitoring and control—Terminology, phenomena, and solution implementation strategies," *IEEE Power Energy Mag.*, vol. 2, pp. 68–76, 2004.

[3] "National power grid simulator workshop report," U.S. Dept. Homeland Security, 2008.

[4] J. De La Ree, V. Centeno, J. S. Thorp, and A. G. Phadke, "Synchronized phasor measurement applications in power systems," *IEEE Trans. Smart Grid*, vol. 1, pp. 20–27, 2010.

[5] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE Control Syst.*, vol. 21, pp. 11–25, 2001.

[6] M. C. D'Abreu and G. A. Wainer, "Models for continuous and hybrid system simulation," in *Proc. 2003 Winter Simul. Conf.*, vol. 1, pp. 641–649.

[7] F. Bouchhima, M. Briere, G. Nicolescu, M. Abid, and E. M. Aboulhamid, "A SystemC/Simulink co-simulation framework for continuous/discrete-events simulation," in *Proc. 2006 IEEE Int. Behav. Modeling Simul. Workshop*, pp. 1–6.

[8] J. H. Taylor and Z. Jie, "Rigorous hybrid systems simulation with continuous-time discontinuities and discrete-time components," in *Proc. Mediterranean Conf. Control Autom. (MED'07)*, pp. 1–6.

[9] J. Eker, J. W. Janneck, E. A. Lee, L. Jie, L. Xiaojun, J. Ludvig, S. Neuendorffer, S. Sachs, and X. Yuhong, "Taming heterogeneity—The ptolemy approach," *Proc. IEEE*, vol. 91, pp. 127–144, 2003.

[10] A. G. Phadke and J. S. Thorp, "Communication needs for wide area measurement applications," in *Proc. 2010 5th Int. Conf. Critical Infrastruct. (CRIS)*, pp. 1–7.

[11] M. Chenine and L. Nordstrom, "Modeling and simulation of wide-area communication for centralized PMU-based applications," *IEEE Trans. Power Del.*, vol. 26, pp. 1372–1380, 2011.

[12] J. W. Stahlhut, T. J. Browne, G. T. Heydt, and V. Vittal, "Latency viewed as a stochastic process and its impact on wide area power system control signals," *IEEE Trans. Power Syst.*, vol. 23, pp. 84–91, 2008.

[13] K. Hopkinson, W. Xiaoru, 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," *IEEE Trans. Power Syst.*, vol. 21, pp. 548–558, 2006.

[14] J. Nutaro, P. T. Kuruganti, L. Miller, S. Mullen, and M. Shankar, "Integrated hybrid-simulation of electric power and communications systems," in *Proc. IEEE Power Eng. Soc. Gen. Meet. 2007*, pp. 1–8.

[15] K. Zhu, M. Chenine, and L. Nordstrom, "ICT architecture impact on wide area monitoring and control systems' reliability," *IEEE Trans. Power Del.*, vol. 26, no. 4, pp. 2801–2808, Oct. 2011.

[16] W. Li, A. Monti, M. Luo, and R. A. Dougal, "VPNET: A co-simulation framework for analyzing communication channel effects on power systems," in *Proc. 2011 IEEE Electr. Ship Technol. Symp. (ESTS)*, pp. 143–149.

[17] V. Liberatore and A. Al-Hammouri, "Smart grid communication and co-simulation," in *Proc. 2011 IEEE Energytech*, pp. 1–5.

[18] T. Xiaoyang, "The co-simulation extending for wide-area communication networks in power system," in *Proc. 2010 Asia-Pacific Power Energy Eng. Conf. (APPEEC)*, pp. 1–4.

[19] C. M. Davis, J. E. Tate, H. Okhravi, C. Grier, T. J. Overbye, and D. Nicol, "SCADA cyber security testbed development," in *Proc. 38th North Amer. Power Symp. (NAPS 2006)*, pp. 483–488.

[20] M. Mallouhi, Y. Al-Nashif, D. Cox, T. Chadaga, and S. Hariri, "A testbed for analyzing security of SCADA control systems (TASSCS)," in *Proc. 2011 IEEE PES Innov. Smart Grid Technol. (ISGT)*, pp. 1–7.

[21] A. Ashok, A. Hahn, and G. Manimaran, "Cyber-physical security testbed: System architecture and studies," presented at the Cyber Security Inf. Intell. Res. (CSIIR) Workshop, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN, 2011.

[22] A. Hahn, B. Kregel, M. Govindarasu, J. Fitzpatrick, R. Adnan, S. Sridhar, and M. Higdon, "Development of the PowerCyber SCADA security testbed," presented at the 6th Annu. Workshop Cyber Security Inf. Intell. Res., Oak Ridge, TN, 2010.

[23] T. Chiocchio, R. Leonard, Y. Work, R. Fang, M. Steurer, A. Monti, J. Khan, J. Ordonez, M. Sloderbeck, and S. L. Woodruff, "A co-simulation approach for real-time transient analysis of electro-thermal system interactions on board of future all-electric ships," presented at the 2007 Summer Comput. Simul. Conf., San Diego, CA.

[24] I. Leonard, T. Baldwin, and M. Sloderbeck, "Accelerating the customer-driven microgrid through real-time digital simulation," in *Proc. IEEE Power Energy Soc. Gen. Meet. (PES'09)*, 2009, pp. 1–3.

[25] C. Zhang, V. K. Vijapurapu, A. K. Srivastava, N. N. Schulz, J. Bastos, and R. Wierckx, "Hardware-in-the-loop simulation of distance relay using RTDS," presented at the 2007 Summer Comput. Simul. Conf., San Diego, CA.

[26] R. Kuffel, D. Ouellette, and P. Forsyth, "Real time simulation and testing using IEC 61850," in *Proc. 2010 Proc. Int. Symp. Modern Electr. Power Syst. (MEPS)*, pp. 1–8.

[27] B. P. Zeigler, *Theory of Modeling and Simulation*, 2nd ed. New York: Academic, 2000.

[28] A. G. Phadke and J. S. Thorp, "Expose hidden failures to prevent cascading outages [in power systems]," *IEEE Comput. Appl. Power*, vol. 9, pp. 20–23, 1996.

[29] H. Wang and J. S. Thorp, "Optimal locations for protection system enhancement: A simulation of cascading outages," *IEEE Trans. Power Del.*, vol. 16, pp. 528–533, 2001.

[30] J. De La Ree, Y. Liu, L. Mili, A. G. Phadke, and L. DaSilva, "Catastrophic failures in power systems: Causes, analyses, and countermeasures," *Proc. IEEE*, vol. 93, pp. 956–964, 2005.

[31] W. R. Lachs, "A new horizon for system protection schemes," *IEEE Trans. Power Syst.*, vol. 18, pp. 334–338, 2003.

[32] S. Song, Y. Zhu, M. Huang, and H. Yu, "Multiagent and WAN based adaptive coordinated protection system," in *Proc. 2005 IEEE/PES Transm. Distrib. Conf. Exhib. Asia Pacific*, 2005, pp. 1–6.

[33] T. Shono, K. Sekiguchi, T. Tanaka, and S. Katayarna, "A remote supervisory system for a power system protection and control unit applying mobile agent technology," in *Proc. 2002 IEEE/PES Transm. Distrib. Conf. Exhib. Asia Pacific*, vol. 1, pp. 148–153.

[34] W. Dong-Qing, M. Shi-Hong, L. Xiang-Ning, L. Pei, W. Ying-Xin, and Y. Dan, "Design of a novel wide-area backup protection system," in *Proc. 2005 IEEE/PES Transm. Distrib. Conf. Exhib. Asia Pacific*, pp. 1–6.

[35] X. Tianqi, Y. Xianggen, Y. Dahai, L. Yan, and W. Yangguang, "A novel communication network for three-level wide area protection system," in *Proc. 2008 IEEE Power Energy Soc. Gen. Meet.—Convers. Del. Electr. Energy 21st Century*, pp. 1–8.

[36] M. M. Eissa, M. E. Masoud, and M. M. M. Elanwar, "A novel back up wide area protection technique for power transmission grids using phasor measurement unit," *IEEE Trans. Power Del.*, vol. 25, pp. 270–278, 2010.

[37] R. Giovanini, K. Hopkinson, D. V. Coury, and J. S. Thorp, "A primary and backup cooperative protection system based on wide area agents," *IEEE Trans. Power Del.*, vol. 21, pp. 1222–1230, 2006.

[38] I. Hiskens, "Significance of load modeling in power system dynamics," presented at the Symp. Specialists Electr. Oper. Expansion Plan., 2006.

[39] S. Bak, "Large-scale network simulation scalability and an FPGA-based network simulator," Univ. Illinois Tech. Rep. [Online]. Available: http://hdl.handle.net/2142/30815, 2012

[40] S. Kristiansen and T. Plagemann, "Ns-2 distributed clients emulation: Accuracy and scalability," presented at the 2nd Int. Conf. Simul. Tools Tech., Rome, Italy, 2009.

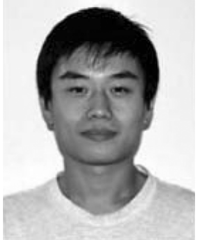
[41] G. Carl, "Towards large-scale testing of policy-based routing via path algebraic and scaled-down topological modeling," D.Phil. dissertation, Dept. Electr. Eng., Pennsylvania State University, State College, PA, 2008.

[42] L. Hua, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, "Power system and communication network co-simulation for smart grid applications," in *Proc. 2011 IEEE PES Innov. Smart Grid Technol. (ISGT)*, pp. 1–6.

[43] L. Hua, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, "Ad-hoc vs. supervisory wide area backup relay protection validated on power/network co-simulation platform," presented at the 17th Power Syst. Comput. Conf. (PSCC), Stockholm, Sweden, 2011.

[44] S. Garlapati, L. Hua, S. Sambamoorthy, S. K. Shukla, and J. Thorp, "Agent based supervision of zone 3 relays to prevent hidden failure based tripping," in *Proc. 2010 1st IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, pp. 256–261.

[45] G. N. Ericsson, "On requirements specifications for a power system communications system," *IEEE Trans. Power Del.*, vol. 20, pp. 1357–1362, 2005.



Hua Lin (M'08) received the B.S. degree from Nankai University, China, in 2004 and the M.S. degree from Shanghai Jiao Tong University, China, in 2007 respectively, both in electrical engineering. Currently he is working toward the Ph.D. degree in electrical and computer engineering department at Virginia Tech, Blacksburg.

His research interests focus on simulation techniques for WAMS and smart grid applications.

Santosh S. Veda (M'07) received the B.S. degree in electrical and electronics engineering from Anna University, Chennai, India, in 2008 and the M.S. degree in electrical engineering from Virginia Tech, Blacksburg in 2011. He is currently working toward the Ph.D. degree in the Electrical and Computer Engineering Department at Virginia Tech.

His research interests include power system stability analysis, modeling and simulation, and wide area monitoring protection and control (WAMPAC).

Mr. Veda was conferred Vocational Excellence Award by the Rotary Club of Coimbatore in 2008.

Sandeep K. Shukla (SM'03) received the Ph.D. degree from the State University of New York at Albany in 1997.

He is an Associate Professor in the Department of Electrical and Computer Engineering at Virginia Polytechnic and State University, Blacksburg. He is also the director of the Center for Embedded Systems for Critical Applications (CESCA).

Prof. Shukla is Associate Editor of IEEE TRANSACTIONS ON COMPUTERS and *IEEE Embedded Systems Letters*. He was associate editors for IEEE *Design & Test of Computers* (2003–2011), and IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS (2005–2006). He was awarded the PECASE in 2004, Virginia Tech Engineering Faculty Fellow award in 2005, SUNY Albany Distinguished Alumni Award in 2007, and Humboldt Foundation's Bessel award in 2008. He is an IEEE Computer Society Distinguished Visitor, and an ACM Distinguished Speaker.

Lamine Mili received an Electrical Engineering Diploma from the Swiss Federal Institute of Technology, Lausanne, in 1976, and the Ph.D. degree from the University of Liège, Belgium, in 1987.

He is a Professor of Electrical and Computer Engineering at Virginia Tech, Blacksburg. He has five years of industrial experience with the Tunisian electric utility, STEG. At STEG, he worked in the planning department from 1976 to 1979 and then at the Test and Meter Laboratory from 1979 till 1981. He was a Visiting Professor at the Swiss Federal Institute of Technology, Lausanne, the Grenoble Institute of Technology and the École Supérieure d'Électricité in in France, the École Polytechnique de Tunisie, in Tunisia, and did consulting work for the French Power Transmission company RTE. His research has focused on power system planning for enhanced resiliency and sustainability, risk management of complex systems to catastrophic failures, robust estimation and control, nonlinear dynamics, and bifurcation theory. He is the co-founder and co-editor of the *International Journal of Critical Infrastructures*, http://www.inderscience.com/browse/index.php?journalCODE=ijcis.

James Thorp (F'89–LF'03) received the Ph.D. degree from Cornell University, Ithaca, NY, in 1962.

He is an Emeritus Professor in the Department of Electrical and Computer Engineering at Virginia Polytechnic Institute and State University, Blacksburg.

Prof. Thorp was an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1985 to 1987. He is a member of the National Academy of Engineering, a member of the IEEE Power System Relaying Committee, CIGRE, Eta Kappa Nu, Tau Beta Pi, and Sigma Xi. He is also co-recipient of the Benjamin Franklin Medal in 2008.