SANDIA REPORT
SAND2006-4083
Unlimited Release
Printed July 2006

National SCADA Test Bed: FY05 Progress on Virtual Control System Environment (VCSE)

Erik J. Lee, John M. Michalski, and Brian P. Van Leeuwen

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2006-4083
Unlimited Release
Printed July 2006

National SCADA Test Bed: FY05 Progress on Virtual Control System Environment (VCSE)

Erik J. Lee, John M. Michalski, and Brian P. Van Leeuwen
Networked Systems Survivability & Assurance
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0672

This page intentionally left blank.

Table of Contents

This page intentionally left blank.

Executive Summary

This document provides the status of the Virtual Control System Environment (VCSE) under development at Sandia National Laboratories. This development effort is funded by the Department of Energy's (DOE) National SCADA Test Bed (NSTB) Program. Specifically the document presents a Modeling and Simulation (M&S) and software interface capability that supports the analysis of Process Control Systems (PCS) used in critical infrastructures. This document describes the development activities performed through June 2006 and the current status of the VCSE development task.

Initial activities performed by the development team included researching the needs of critical infrastructure systems that depend on PCS. A primary source describing the security needs of a critical infrastructure is the *Roadmap to Secure Control Systems in the Energy Sector* [2]. A literature search of PCS analysis tools was performed and we identified a void in system-wide PCS M&S capability. No existing tools provide a capability to simulate control system devices and the underlying supporting communication network. The design team identified the requirements for an analysis tool to fill this void. Since PCS are comprised of multiple subsystems, an analysis framework that is modular was selected for the VCSE. The need for a framework to support the interoperability of multiple simulators with a PCS device model library was identified. The framework supports emulation of a system that is represented by models in a simulation interacting with actual hardware via a System-in-the-Loop (SITL) interface.

To identify specific features for the VCSE analysis tool the design team created a questionnaire that briefly described the range of potential capabilities the analysis tool could include and requested feedback from potential industry users. This initial industry outreach was also intended to identify several industry users that are willing to participate in a dialog through the development process so that we maximize usefulness of the VCSE to industry. Industry involvement will continue throughout the VCSE development process.

The team's activities have focused on creating a modeling and simulation capability that will support the analysis of PCS. An M&S methodology that is modular in structure was selected. The framework is able to support a range of model fidelities depending on the analysis being performed. In some cases high-fidelity network communication protocol and device models are necessary which can be accomplished by including a high-fidelity communication network simulator such as OPNET Modeler. In other cases lower fidelity models could be used in which case the high-fidelity communication network simulator is not needed. In addition, the framework supports a range of control system device behavior models. The models could range from simple function models to very detailed vendor-specific models.

Included in the FY05 funding milestones was a demonstration of the framework. The development team created two scenarios that demonstrated the VCSE modular

framework. The first demonstration provided a co-simulation using a high-fidelity communication network simulator interoperating with a custom developed control system simulator and device library. The second scenario provided a system-in-the-loop demonstration that emulated a system with a virtual network segment interoperating with a real-device network segment.

1 Introduction

One effective tool for the analysis of Process Control Systems (PCS) is modeling and simulation. This provides a means to examine the performance of the PCS and to examine the overall system operation. Simulation studies provide a means to collect data necessary to aid in the analysis, including tradeoff studies, of large complex PCS. Modeling and simulation provide the designer with a tool to represent a wide range of operating environments and application scenarios. For modeling and simulation tools to be effective in the analysis of specific types of systems, models must exist or be created of various devices and sub-components that make up the system under study.

This project's objective is to develop a modeling and simulation environment that can be used to support the analysis of PCS. A proof-of-concept framework to support the interoperability of multiple simulators with a PCS device model library and simulator has been developed and demonstrated. The framework supports emulation of a system that is represented by models in a simulation interacting with actual hardware via a System-in-the-Loop (SITL) interface. The capabilities described in this report make up the Virtual Control System Environment (VCSE).

The VCSE is an engineering tool to be used by PCS researchers, designers, integrators, and security analysts. The target user will have a technical understanding of PCS architectures and supporting protocols. More specifically, the target user will have a technical understanding of the system aspects that generate the questions the tool is being used to answer. However, we do not expect all users to have the necessary expertise in modeling formalism and software development needed to modify and extend the component library. It is expected that a more limited group of developers will develop models for the component library.

The ultimate recipient of value of the VCSE is much broader than the users listed above. The users are developers of PCS devices and systems. Tools that support the analysis of how specific devices will impact a PCS will benefit owner-operators in maintaining a reliable system. Additionally, system analysts performing assessments, such as security assessment, will benefit from the VCSE in performing what-if studies and understanding the impacts of proposed mitigation strategies.

The VCSE is a modeling and simulation environment and suffers from the limitations that are associated with this type of analysis. In most cases, models are not exact representations of devices, protocols, etc. and thus their behaviors will not exactly replicate the component being modeled. Analyst must understand the specific model range of operation and specific behaviors that are represented. In most cases, models include abstractions that will impact the fidelity of the results. In cases where device models do not exist or do not have the required fidelity the VCSE's System-in-the-Loop capability can be used to include real devices in an emulation experiment.

The primary driver for the development of the VCSE came from requirements identified in the *Roadmap to Secure Control Systems* [2]. Following is a list of specific goals, challenges, milestones, and selected priorities identified in the Roadmap. The VCSE can be used in the analysis or assessment of the PCS identified in the Roadmap.

Summary of Relevance to Roadmap to Secure Control Systems:

Roadmap Goal #1: Measure and Assess Security Posture
Challenges: • Insufficient tools and techniques exist to measure risk • Threats are hard to demonstrate and quantify
Milestones: • 2008: 50% of asset owners and operators performing self-assessments of their control systems using consistent criteria.
Selected Priority: Security Tools and Practices: • Set up and evaluate cyber attack and response simulators • Develop risk assessment tools
Description: The VCSE is a tool that can help an analyst assess a PCS by performing analysis on the modeled PCS. In most cases, an on-line operational system cannot be stressed by introducing attacks or failures to measure the systems resilience. Using a modeled system is a much more practical and cost effective solution for answering key security questions for large systems where it is not practical to build a full-scale test bed. Further integration of shared telecommunications technologies into normal business operations has spawned increased levels of interconnectivity among corporate networks, control systems, other asset owners, and the outside world. This expansion of connectivity provides increased potential of cyber attacks and new security measures are needed to mitigate attacks. The VCSE will support the analysis of security measures and their impacts on system operation.

Roadmap Goal #2: Develop and Integrate Protective Measures
Challenges: • Security upgrades are often hard to retrofit to legacy systems, may be costly, and may degrade system performance
Milestones: • 2014: Make available security test harness for evaluating next generation architectures and individual components
Selected Priority: Legacy Systems: • Improve performance of legacy communications to enable the application of security solutions **Selected Priority: Control System Architecture:** • Develop a security test harness with testing architecture and guidelines

Description:
Security solutions that are devised for legacy systems will be constrained by limitations of existing equipment and configurations. Analyzing the interactions and behavior between emerging security solutions and existing legacy control systems will be critical for identifying the introduction of any vulnerability into the proposed control system's security solution. The VCSE is a tool that will support the design, integration, and evaluation of security solutions used in legacy systems.

The VCSE will complement the use of a security test harness to evaluate next-generation control systems. Modeling and simulation is a cost effective and schedule effective method of determining operational characteristics and identifying interoperability issues without building costly prototype hardware and large scale test beds.

The VCSE also supports the following areas identified in the Roadmap's Appendix A – Key Challenges and Solutions.

Appendix A:
Key Challenges & Solutions: Security Tools and Practices

Challenges:
Adequate testing and validation tools for new security components and practices are needed.

Milestones:
Near Term (0-2years): Set up and evaluate cyber attack and response simulators

Description:
Tools, such as the VCSE, that can model and simulate energy sector threats and vulnerability information are needed to assist asset owners in prioritizing threats and implementing risk management strategies that will promote system survivability and recovery.

Appendix A:
Key Challenges & Solutions: Control Systems Architectures

Challenges:
Future threats are difficult to anticipate and define

Milestones:
- Mid Term (2-5years): Develop analytical tools to cost-effectively model system's architecture and determine efficacy.
- Mid Term (2-5years): Create robust modeling of envisioned architectures to identify vulnerabilities.

Description:
Modeling tools such as the VCSE are needed to combat the technology complexities associated with the challenges of securing both current and emerging control system components and control system architectures. As control system architectures grow in complexity (along with interconnectivity with other networks, the increased exposure to threat environments, and the trend to incorporate conventional Information Technologies (IT) solutions) modeling and simulation tools will be needed to assist asset owners in making better informed decisions when it comes to selecting security solutions for their

current and next-generation control systems.

Additional VCSE Analysis Capabilities:

In addition to supporting the Roadmap specific areas, the VCSE tool will aid in the analysis of PCS that employ IP networks and wireless communication for data exchange. The VCSE will enable an ability to analyze the following:

1. Performance of PCS extensions and modifications as future needs become apparent. The VCSE will benefit industry by providing an inexpensive way to analyze large proposed PCS for both performance and security.
2. System dependencies and estimating failure consequences.
3. Impacts of cryptographic protocols on system performance.
4. Expected performance of network architectures and technologies that use IP networks and wireless communications.
5. Impacts of wireless channel stressors, both environmental and adversarial, on PCS reliability.
6. Scalability issues with increasing number of sites and how best to deploy wireless for optimum spectrum use.
7. Interference mitigation via architecture layout.
8. Behavior of wireless communications over a range of environments to study sensitivities.

The remainder of this project report is organized as follows. In Section 2, we provide a description of the FY05 project objective. In Section 3, an overview of our M&S methodology is provided along with the tools we use. Descriptions of key models necessary for the VCSE are provided in Section 4. The scenario development status is described in Section 5. Concluding remarks are presented in Section 6.

2 Task Objective

The objective of this task was to develop a modeling and simulation tool to analyze and assess PCS. The modeling and simulation tool, or Virtual Control System Environment (VCSE) tool, consists of simulated control system devices, simulated network communications, and provides an interface to attach actual system devices to provide a means for real-time system-in-the-loop (SIL) emulation. The virtual environment tool will benefit industry by providing an inexpensive way to analyze and assess large virtual PCS. Results from the tool can help the analyst identify system dependencies and system vulnerabilities, estimate failure consequences, and assess the performance impacts of security approaches. Additionally, results from the tool will provide the PCS designer with a means to analyze a prototype system design, software upgrades, and security upgrades before they are actually purchased and deployed.

2.1 Requirements Development and Industry Outreach Subtask

Initial activities performed by the development team included researching the needs of critical infrastructure systems that depend on secure PCS [3, 4, 6, 7]. A primary source describing the security needs of a critical infrastructure is the *Roadmap to Secure Control Systems in the Energy Sector* [2]. We also performed a literature search of PCS analysis tools to determine the types of analysis tools that are currently available. There are a number of analysis tools that focus on the analysis of communication networks [12, 13, 14] but do not provide a control system focus. These tools employ application layers that create traffic based on user identified profiles and statistical data. To address the necessary analyses to support the objectives identified in the Roadmap, an analysis tool that performs control system analysis is necessary.

A draft document describing the requirements of the FY06 VCSE development has been initiated [9]. This document provides an initial cut at high-level requirements for the VCSE. It identifies the requirements that guided the FY05 modular framework selection. In addition, the document describes additional features than will be implemented under the FY06 funded development activity. This requirements document uses the terms "shall," "should," and "may," so that the differences in the planned FY06 feature set and the long-range feature set is clearly apparent in each subsequent section.

To help identify the value of specific features for the VCSE analysis tool the design team created a questionnaire to obtain industry input [10]. The questionnaire included a brief description of the VCSE and questions related to the range of potential capabilities the analysis tool could include. Potential industry users were requested to answer the questions and provide comments. The initial industry outreach was also intended to identify several industry users that are willing to participate in a dialog throughout the development process so that we maximize usefulness of the VCSE to industry.

The development team intends to increase the level of industry involvement during the FY06 funding period's development activities.

2.2 Control System Simulation Subtask

This subtask resulted in a modeling and simulation capability of control system modules. The capability developed in this subtask will lead to a useful set of control system models and the means for easy extension and modification of these models as future analysis needs become apparent. In addition, in this subtask we developed the capability for the virtual environment tool to interface with actual hardware to provide precise representations of the device. One potential use of interfacing to actual hardware is the capability to interface Human Machine Interface (HMI) software to the VCSE. HMI is complex enough that it may be difficult to develop high-fidelity models for that software during the FY06 effort.

2.3 Network Simulation Subtask

This subtask resulted in a co-simulation capability that integrated the VCSE's control system device models to interoperate with a network communication simulator. The tool provides capability to analyze control systems that depend on Internet Protocols (IP) such as IPSec, TCP, UDP, and peer-to-peer protocols for network communications. The development of this tool helps identify issues and concerns related to IP networks that may cause operational disruptions. Analysis of PCS that depend on IP networks is necessary to determine if their performance characteristics are sufficient to meet the real-time demands of substation automation. This tool provides a means to determine if specific IP network architectures can support time-critical communications and control needs in PCS. Experimentation with actual hardware experiments is important, however, it can pose issues because obtaining repeatable performance of dynamic events is difficult because the quality of communications can vary over time in test-bed networks. In contrast, simulations trade off model fidelity for improved repeatability of specific experiments.

Many PCS are migrating to wireless communications for data gathering and real-time control [1]. Since wireless communications have unique operating characteristics and security concerns, it is important to incorporate these aspects in the analysis of PCS operation. PCS incorporating wireless infrastructure must be analyzed in typical operational environments and during times of environmental and adversarial stressors. A capability to analyze the real-time response of PCS that depend on wireless communications is an important aspect of quantifying the PCS performance. Our FY05 task resulted in a capability to analyze PCS that depend on wireless communications. The wireless analysis capability results from the selected network communication simulator's (i.e., OPNET Modeler) ability to model wireless communication radios and links.

Analysis of systems that depend on wireless communications is a challenge because of the difficulties with modeling realistic wireless-channel conditions. The capability to analyze wireless systems in an M&S tool is important for the following reasons:

1. Thorough evaluation of the performance of a wireless communication system cannot be done unless the wireless channel is accurately represented in the experiment.
2. It is necessary to recreate identical wireless channel conditions for each experiment when comparing alternative system designs.
3. It is necessary in debugging wireless communication systems because it enables the re-creation of conditions that trigger rare but serious bugs.
4. It is necessary to understand the performance tradeoffs between different communication and security protocols when their performance is dependent on the underlying wireless environment.

M&S does have drawbacks for modeling wireless communications though. The propagation models used in simulation tools do not exactly model Radio Frequency (RF) propagation in complex environments. So, a system that works in simulation may need to

be “tweaked” (based on field experiments) in order to work in practice. However, a wireless system that does not work in simulation will seldom work in practice.

2.4 Physical System Simulator Interface Subtask

This subtask will develop an interface in the VCSE to provide a method of co-simulating with a physical system simulator. In FY05 we began the initial interface development to interoperate with a Sandia National Laboratories developed power system simulator that use Newton-Raphson techniques to model the steady-state power flow. During FY06, the VCSE team is exploring other power flow models in conjunction with the Pacific Northwest National Lab (PNNL). When completed, this capability will provides a means to analyze the effects the PCS has on the physical system and the effects the state changes of the physical system has on the PCS.

3 VCSE Modeling and Simulation Methodology

The FY05 VCSE software provides a proof-of-concept modeling and simulation (M&S) environment that supports the analysis of PCS via a modular framework methodology. Our proof-of-concept approach employs an extensible modular M&S capability. This capability uses a modular approach that supports interoperation with a network communication simulator to represent the necessary communications network components. The modular approach supports easy integration of future models that will represent specific communication protocols associated with PCS (e.g. DNP3 and ModBus). These models can be developed in either the selected network M&S tool or in the VCSE Framework.

An important feature of the VCSE’s modular framework methodology is a “co-simulation” capability that supports the interoperability of multiple simulation tools operating in unison. Our methodology addresses the co-simulation relationship among the multiple simulators. In a co-simulation framework the simulators must also support integration with other simulators and their associated model libraries. In this project an interoperability capability was developed during FY05 for the VCSE framework (described in the next subsection) and the OPNET Modeler network simulator.

3.1 VCSE Simulator Framework

The VCSE modular framework provides a capability to perform analysis with the necessary modules to address the questions that a specific analysis is attempting to answer. In some cases, the analysis may require high-fidelity modeling of the PCS’s supporting network communications. In these cases, the analyst will include a communication network simulator in the environment. High-fidelity models will represent network components that support the PCS under study. In the cases where network communications is not critical to the analysis no communication network simulator will be included. In this case, network communication may be assumed to be perfect; no communication delays, no dropped messages.

The architecture of the VCSE Framework is illustrated in Figure 1 and brief descriptions of each module are provided in Table 1. Note that the Table 1 describes the *target* capability of each module. Continued development is necessary during FY06 to implement all the capabilities.

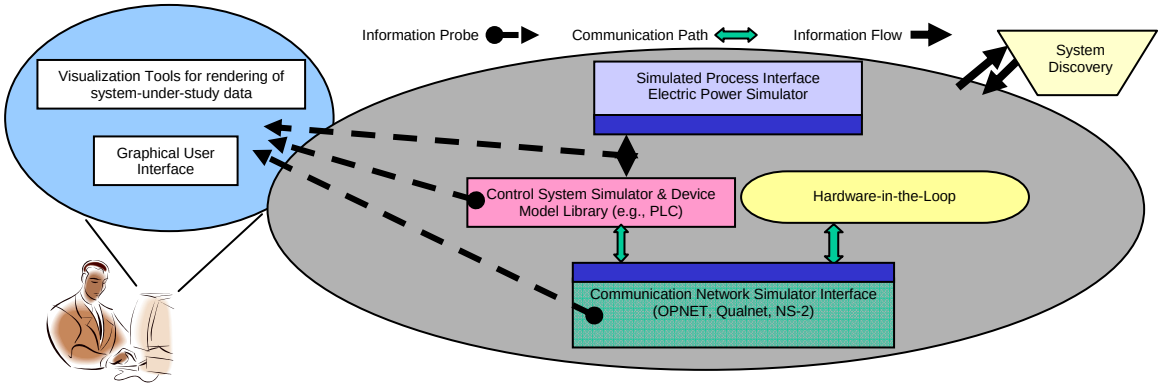

Figure 1: VCSE Architecture

The capability for the VCSE to incorporate high-fidelity models and simulation comes from its ability to merge several simulators into a *co-simulation*. The co-simulation approach allows complex communication protocols and network architectures to be simulated with a network simulator such as OPNET while being offered realistic traffic and operating environment conditions provided by the VCSE Framework's control system simulator. This co-simulation approach allows for the simulation of system operation when the impairments caused by realistic communication network operation are included in the model. This method results in an analysis capability that can provide more realistic system results than when a single simulation capability is used that typically abstracts some part of the overall system model to a much lower fidelity. In general, the co-simulation approach provides a means for specialized simulators to accurately portray the physical behavior, network behavior, and other aspects of the simulated world. (Note: during FY05, the VCSE Team demonstrated a proof-of-concept interface between the OPNET Modeler simulation tool and the VCSE Framework software, as described below.)

Table 1: VCSE Module Descriptions

VCSE Module	Description
Visualization Tools for rendering of system-under-study data	Visualization tools that render architecture under study, state of control system, data traffic characteristics, etc.
Graphical User Interface (GUI)	Interface for tool user to build and execute simulation/emulation experiments.
Control System Simulator & Device Model Library	Control system and device model library. The PCS device model library will be based on the SCADA

	Reference Model and can be extended to include vendor specific devices.
Communication Network Simulator (Note that OPNET Modeler was used during the FY05 work).	An API for communication network simulator and protocol model library. Many network simulators include vendor specific model libraries.
Hardware-in-the-Loop	Interface to incorporate actual hardware in simulation experiments. This module supports experiments with both a virtual part and actual hardware part.
Power System Simulator/Emulator	An API for tools that simulate the state of the physical system being monitored/controlled by the PCS under study.
System Discovery	An interface for tools used to discover the network and PCS under study. Creates a file that can be imported into PCS simulator and communication network simulator.

The primary development effort for the VCSE in the FY05 Funding Period was the modular framework. This innovative *plug-in* approach is the basis for the VCSE Framework. The framework provides the interoperability capability for co-simulations and provides the means to create simulations with models from supporting simulators and custom models created specifically for the VCSE. Additional developments in the FY05 Funding Period as associated with each module are described below.

Visualization tools for rendering of system-under-study data:
The VCSE Framework includes an interface to incorporate various visualization tools. The analyst can select how to represent the data to support answering system questions. With the FY05 funding the team developed the interface in the framework to support visualization tools and did a simple visualization for the FY05 demonstration. The FY06 visualization capability will represent simulation/emulation experiment in either real-time or post-experiment processing. Future developments will allow the analyst to place data collection probes and collect simulation data throughout the system modeled with the VCSE.

Graphical User Interface (GUI):
Initial GUI developments were done with the FY05 funding. The primary developments were the necessary GUI parts to support the framework initialization and configuration. The current GUI supports the launching of the VCSE and dynamic loading of modules or plug-ins. More specifically, the GUI interface to dynamically link the VCSE with OPNET Modeler was developed.

Control System Simulator & Device Model Library:
FY05 developments of the VCSE Framework include an event simulation/scheduler engine. This simulation engine manages the discrete event execution of the control system simulation and interleaves its events with the external simulator (i.e., OPNET Modeler) if used. In the case of a co-simulation with OPNET Modeler, the VCSE simulation engine manages the execution of events in OPNET Modeler.

FY05 developments also included a number of basic PCS device models. More specifically, generator functions, voltage sensor function, and limited RTU functions were modeled. These models were created in a developed model-template that interfaced with the VCSE Framework.

These models were used in the FY05 demonstrations. Critical to the value of M&S tools will be an extensive device model library. Follow-on developments during FY06 will focus on extending the device model library.

Communication Network Simulator Interface:
A major development in FY05 was the interface between the VCSE Framework and the OPNET Modeler communication network simulator. Developments included custom state machine development in OPNET to interface the communication protocol stack models to an application layer that interfaces to PCS devices in the VCSE. (Note: Details of this state machine are provided in a following section of this report.) In addition, the VCSE interface included developments in process initialization, launching, and interleaving. A process control capability was developed that supports the interoperation of OPNET Modeler with the VCSE Framework. This process control resulted in OPNET Modeler operating as a *slave* to the VCSE Framework software in a master/slave configuration.

Hardware-in-the-Loop Capability:
Our FY05 developments also included the employment of the OPNET Modeler System-in-the-Loop (SITL) feature to support the VCSE hardware-in-the-loop requirement. This capability merged actual hardware with the virtual environment. FY05 developments were limited to merging network communication devices through an IP interface on the computing platform. Further details are included in a following section of this report.

Power System Simulator/Emulator Interface:
Developments in FY05 included an interface that can merge a Sandia Labs developed steady-state power grid simulator with the VCSE. This interface module will manage the data exchange, both presenting new control system state to the power grid simulator and reporting back the resulting power grid steady-state condition the simulated PCS. Future developments during FY06 will address the possibility of interfacing to existing commercial power system simulators. This task will be done in conjunction with PNNL.

The Sandia Labs developed steady-state power grid analysis tool is a steady-state power flow program that uses an iterative technique to solve for the unknown values in a power system using the known values. With initial known values for a system, a steady-state

power flow simulation can provide the corresponding state the power system enters once it has stabilized. As known values change (e.g., load requirements, generator real power output) or as faults occur (e.g., a tree falling on a transmission line) the simulation can be run again to provide the new state of the power system.

System Discovery:
FY05 developments were limited to investigating methods of creating the system of interest in the virtual environment. Initial plans (for FY06) are to import the system-of-interest topologies with XML files. Our selected network communication simulators (e.g., OPNET) can create and configure large complex topologies with XML files.

3.2 Simulation Framework Development

In general, the VCSE Framework was designed to provide flexibility for future simulation development. Model development supports inheritance to facilitate model development, extension, and maintenance. The framework supports dynamic selection, loading, and configuration of simulation components. This allows the user to make use of existing simulation capability by interfacing it with the VCSE and loading it at runtime.

Software written for the VCSE Framework is broken into two levels of structural hierarchy. The first level is the *plug-in level* and below plug-ins is the *interface level*. Each plug-in may implement zero or more interfaces. Typically, a plug-in will implement a set of interfaces that share a common theme, such as a set of simulated power devices for use in power generation and distribution simulation.

VCSE Framework interfaces provide an object-oriented abstraction that allows the various components of the VCSE to work together seamlessly. Interfaces are further divided into a hierarchy so that items lower in the hierarchy (i.e., *subclasses*) are guaranteed to provide all of the functionality of items above them (i.e., *superclasses*). Each interface exports a set of functionality directly relevant to some simulation task. Other components within the system may then make use of that interface by either requesting it directly or requesting a superclass interface that it implements. This allows the interfaces to maintain a minimal level of connectivity to other interfaces, and thus reduces the burden of software maintenance and configuration. If more than one interface is loaded that implements a particular superclass interface, the user can either accept the system default for that interface or request a different default.

3.3 Communication Protocol Simulator

The communication network simulator used for the FY05 research is the OPNET Modeler and Radio simulation package Version 11.5.A. Optimum Network Performance (OPNET) [12] is a comprehensive engineering system capable of simulating large communication networks with detailed protocol modeling and performance analysis capability. OPNET features include: graphical specifications of models; a dynamic, event-scheduled simulation kernel; integrated tools for data analysis; and hierarchical, object-based modeling. OPNET analyzes system behavior and performance with a

discrete-event simulation engine. Discrete-event simulation is an approach that supports realistic modeling of complex systems that can be represented as a progression of related events. This approach models system behavior based on objects and distinct events such as the arrival of packets at various points in a network. Each object has associated attributes that control its behavior in the simulation.

In OPNET, a node is a collection of interconnected modules in which data is manipulated as defined by the modules. Modules represent the internal capabilities of a node such as data creation, transmission, processing, internal routing, reception, storage and queuing. The modules are used to model aspects of node behavior. A single node model is usually comprised of multiple modules. The modules are connected together by packet streams and statistic wires. Packet streams are used to transport data between the modules while statistic wires allow one module to monitor a varying quantity within another module. The ability to integrate the use of modules, packet streams, and statistic wires allows the developer to create highly realistic simulations of node behavior.

Each node module contains a set of inputs and outputs, some state memory, and a method for computing the module's outputs from its inputs and its state memory. OPNET provides a standard model library that represents standard functions and protocols and specialized libraries that represent vendor specific devices and specialized protocols. To support development activities, the models are open source and can be customized to represent user-defined behavior. In addition, completely new modules can be created to represent user developed functions and protocols. Example OPNET modules are processors, queues, generators, receivers, and transmitters. The processor, queue, and generator modules are strictly internal to a node. The transmitters, receivers, and antennas have external connections to data transmission links.

For wireless communication network simulation experiments, OPNET includes the effects of the wireless channel, such as path loss, channel interference, and bit-error-rates, within the pipeline models. The standard pipeline models do not, however, include a fading model. Additionally, OPNET has simple antenna models.

3.4 System-in-the-Loop Capability

A feature that can provide more fidelity to the modeling process is called *system-in-the-loop* (SITL). An SITL interface allows for real equipment to be introduced into a simulation. This allows the analyst to mix and match real components of a network scenario with those of the virtual world. This melding of virtual and real components provides an approach that can be used to provide a means of testing actual system components in virtual scenarios to better understand the impact of actual system components on operational scenarios. (Note: Section 5 describes the SITL capability that was prototyped and demonstrated during the FY05 work.)

A simple SITL illustration is shown in Figure 2.

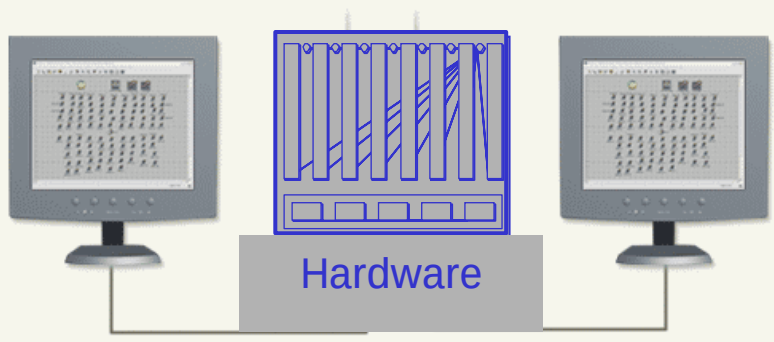

Figure 2: System-in-the-Loop

The OPNET Modeler network simulator has a SITL module that provides a real-time interface that allows networks and their associated hardware to be introduced into a simulation. With SITL there are three basic configurations; real-to-simulated network, real-to-simulated-to-real, and simulated-to-real-to simulated.

The simplest configuration, real-to-simulated, allows a real network component to interact with a simulated network component and pass communication messages. This can allow a real workstation to talk directly to a simulated workstation. Figure 3 displays this configuration. An example PCS experiment would be one that interfaces a real HMI to a virtual system represented in the VCSE.

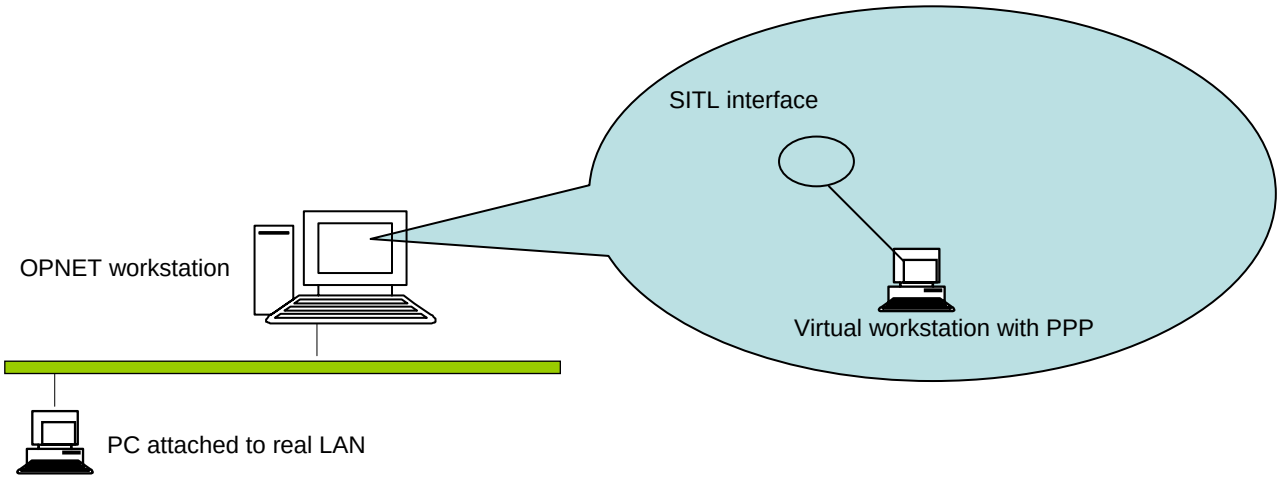

Figure 3: Real-to-Simulated Emulation

The second configuration connects two hardware devices together through a simulated environment. The simulation provides multiple gateway interfaces to connect the external hardware. This can allow real network message to be routed thru a simulated network to determine the impact of proposed networks. Figure 4 shows this configuration. An example experiment would be one that has an HMI issue a message to a real PLC with

the OPNET simulator modeling the effects of the PCS communication network. The communication network would model message latency and message loss.

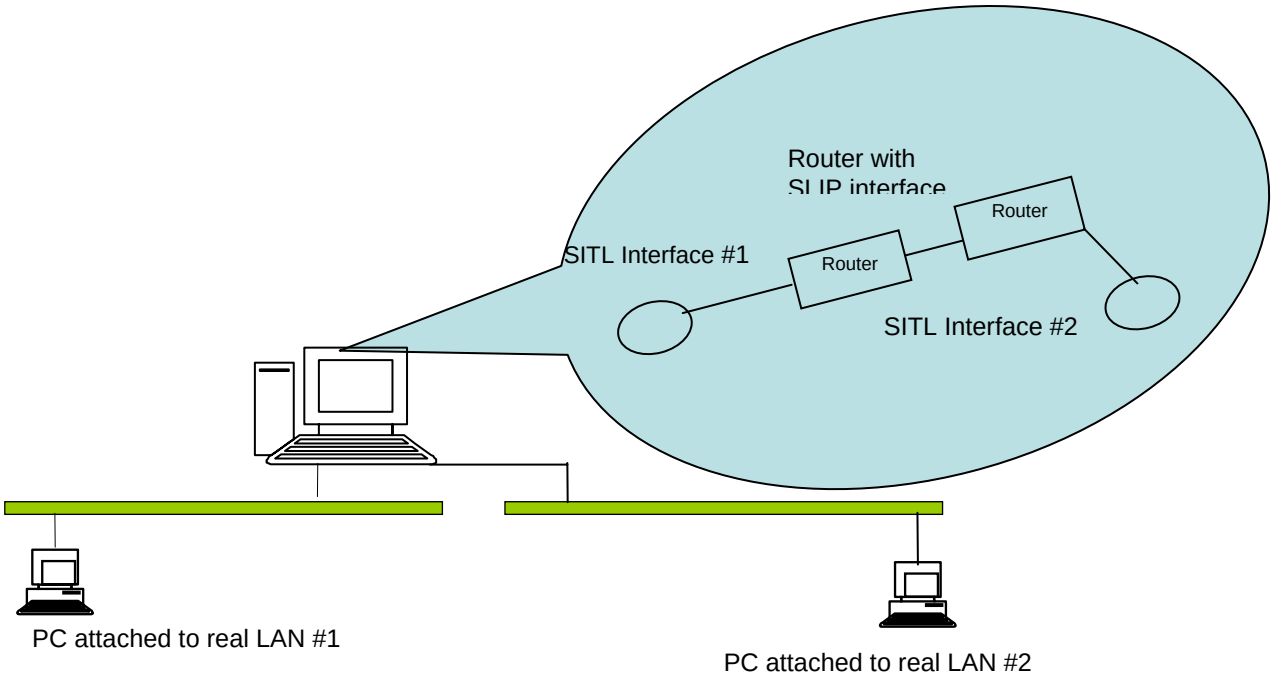

Figure 4: Real-to-Simulated-to-Real Emulation

The third configuration also provides multiple gateway interfaces in the simulation. This allows for two simulated networks to be interconnected by real hardware. The analyst can originate simulated traffic from one modeled network thru a real network segment back into a different simulated network. This configuration for example, can be used to test the interoperability of a real router that is running a real software instance of a routing protocol with a more expansive virtual network. Figure 5 displays this configuration. Another example experiment of this capability would be to allow device manufacturers to test new devices in larger networks. This capability would allow operators to test new devices before deploying them in their operational networks.

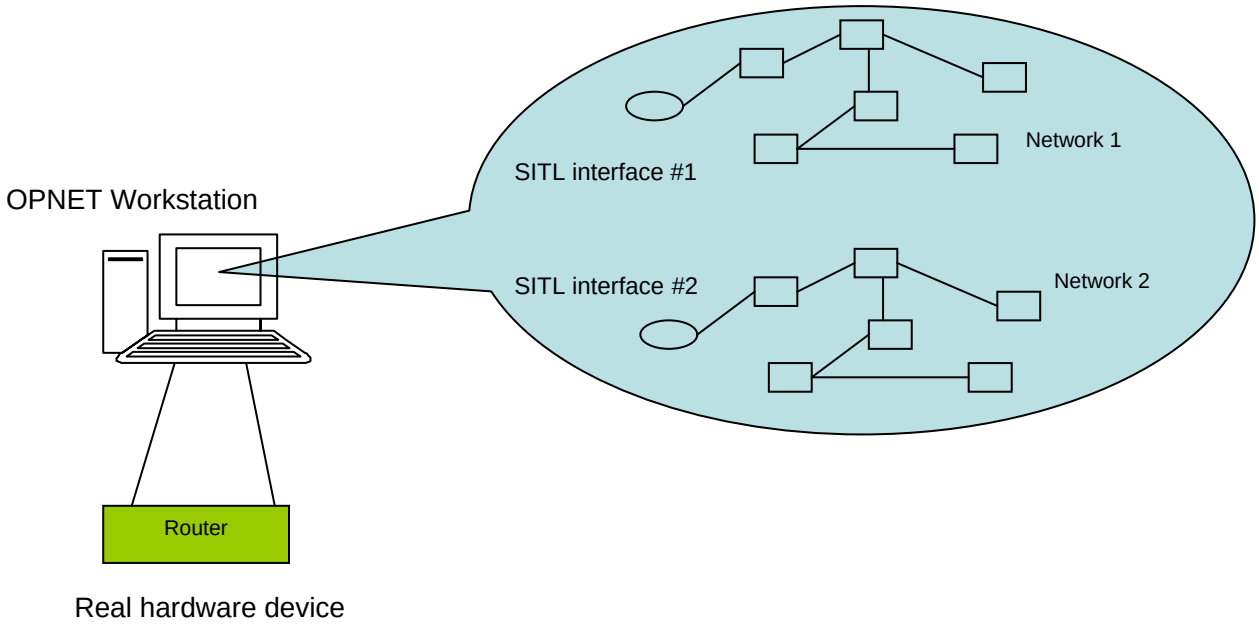

Figure 5: Multiple Simulated LANS Connected via Real Hardware

4 Model Development

Critical to any modeling and simulation environment is the model library that representations the components that make up the system under study. A PCS to be analyzed with the VCSE may include a large number of devices. An ideal situation is one where the user identifies the devices in the system under study and creates a virtual representation in the VCSE by selecting the representative models from the model library. This would limit the applicability of the VCSE to only those systems that had all its devices represented in the VCSE model library. Fortunately a model library of all possible devices is not necessary for an effective analysis tool. Our FY06 goal is to create models of the various PCS functions described in [15]. These function models can then be merged and modified to rapidly create reasonable models of vendor-specific devices if the time, expense and fidelity of a fully-custom model is not warranted. The VCSE framework software supports extending existing models or fully developing a model of a device or protocol as more specific models are needed.

The following sections describe a number of models developed during FY05 in the VCSE to represent a control system device and a model developed in OPNET Modeler to represent data exchange.

4.1 Control System Device Models

In the FY05 development activity a small set of limited control system device models were created in order to test the VCSE simulation framework. The devices included a Programmable Logic Controller (PLC), a generator, and a power line monitor. These models were used in a test configuration that contains a single PLC, two generators, and a power line monitor. The PLC was connected directly to the generators and was connected

to the power line monitor via an IP network link. The generators were intended to simulate the situation where the primary power generation system had a slow response speed and must be compensated for by a *slack generation* system. As the load varied, the slack generation system would rapidly adapt to the changing power conditions. The primary generator would then gradually catch up until it was generating the majority of the required power.

Ideas described in [15] will be used during FY06 to enhance the library of process control devices to achieve maximal utility with minimal configuration overhead. Figure 6 illustrates the potential set of system control functions that will be included in the VCSE model library during FY06.

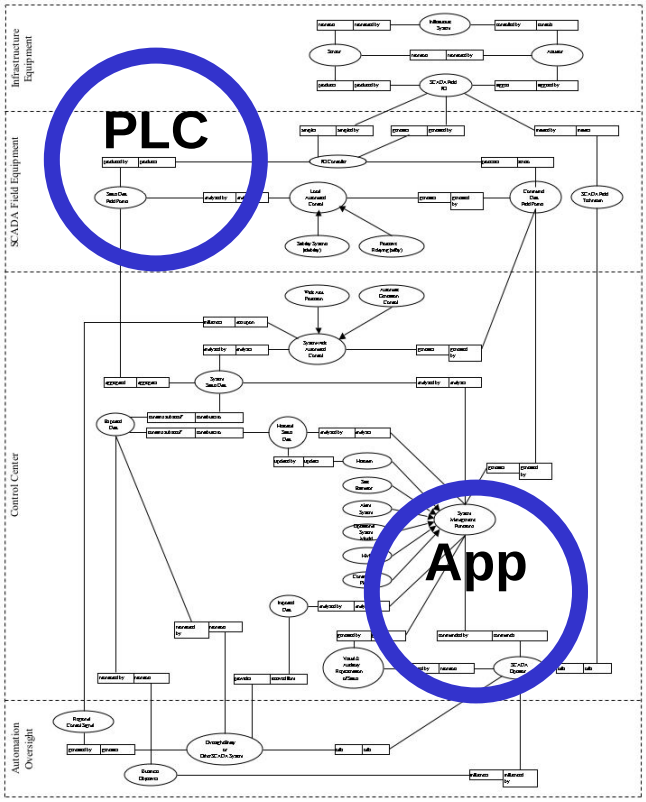

Figure 6: Process Control System Functions [15]

4.2 Network Communication Models

The primary reason for VCSE's co-simulation capability is to employ device models that are available in existing simulation tools. The network communication simulation is a perfect example of the co-simulation capability value. Rather than create models of communication network devices and protocols specifically for the VCSE, the tool can leverage an existing communication network simulator's model library. This benefits the VCSE by having a rich set of high-fidelity models that have been used by others and thus

has received some validation of its correctness. For the FY05 development activity OPNET Modeler was selected as the communication network simulator. The FY06 tasking includes an investigation of the open-source ns2 simulator which is widely used in academia. The goal is to provide a range of model fidelity, model complexity and tool costs options.

This section describes how models are represented in the OPNET Modeler network communication simulator. Other network communication simulators use similar approaches; however their specific representations of states may be different.

Network Communication Node:

Figure 7 illustrates an advanced work station node in OPNET Modeler. This node has a complete set of protocols associated with an IP network. The node can be configured to utilize specific protocols with specific configurations. Additionally, specific protocols can be modified or deactivated to more closely represent the actual device. An important aspect of the node shown in Figure 9 is the VSE_App process that was developed during the FY05 effort. This custom process manages the initialization and merging of devices that have their application exist in the VCSE control system simulator and their communication protocol stack exist in OPNET Modeler. This process manages the bidirectional data flows during simulation run time. Figure 8 illustrates the definition of the data exchange interface.

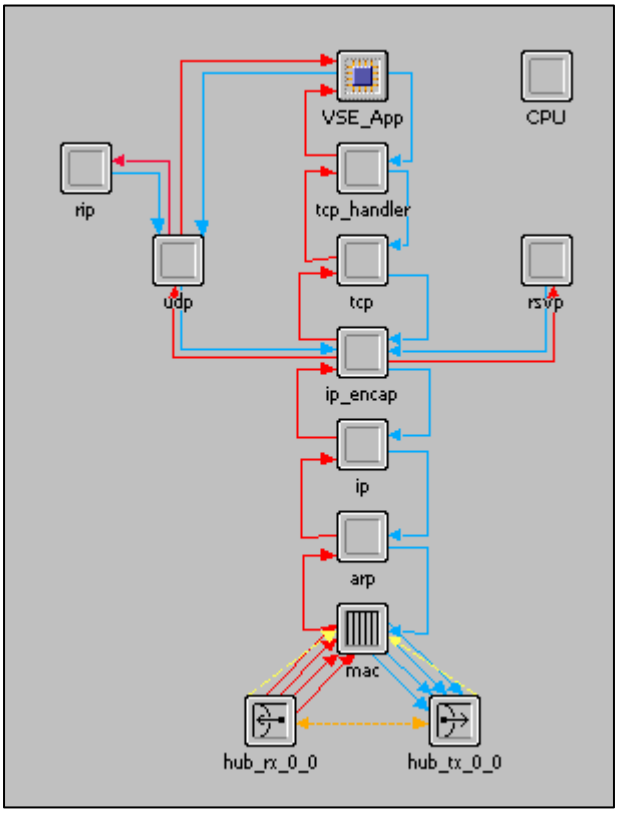

Figure 7: Network Communication Protocols

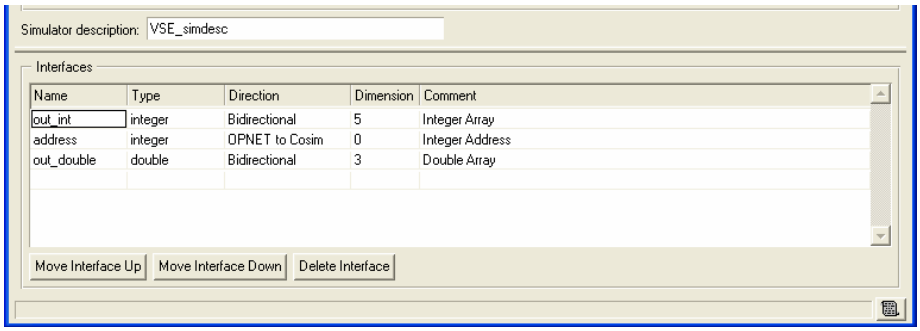

Figure 8: External System Configuration in OPNET Modeler

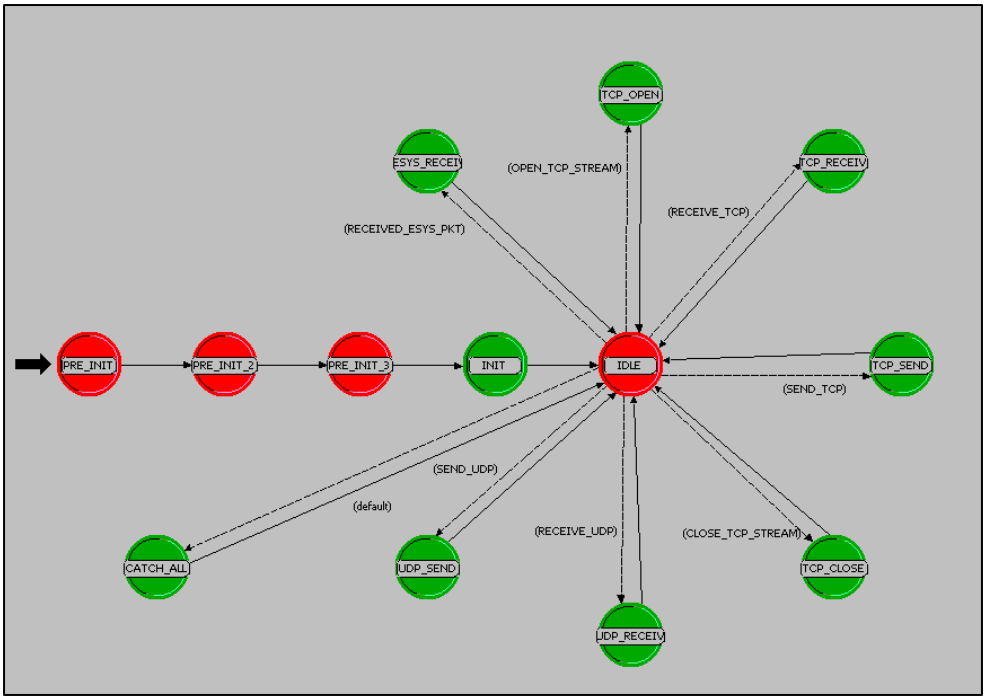

Figure 9: Process Module that Manages Bidirectional Data Exchange

Network Communication Process Module:
Figure 9 illustrate a custom process in the OPNET Modeler design environment. The process module is developed as a state machine with the various states represented in the circular objects in the figure. For the VCSE development a number of OPNET process modules were developed during FY05 to support the interface and data exchange between the VCSE Framework and OPNET Modeler.

In the VCSE an analyst can select network protocols from the OPNET Modeler protocol library. If specific protocols are not available, the analyst can create them in the OPNET Modeler environment by extending a similar protocol or by creating it from scratch. In cases the analyst may prefer to develop the protocol in the VCSE Framework and interface to OPNET via the external interface. This is an option the analyst has when

using the VCSE. In cases, this decision is based on the desired fidelity level. OPNET Modeler might be used if a very high fidelity model is needed. The VCSE framework might be more appropriate for the rapid development of lower-fidelity models.

System-in-the-Loop Module:
The SITL module consists of both transmit and receive processes along with an external system module (ESM). The external system module interfaces with a WinPcap software utility. WinPcap is the industry-standard tool for link-layer network access in the Windows Operating System (OS). WinPcap supports the capability for applications to capture and transmit network packets by bypassing the OS protocol stack. In addition, kernel-level packet filtering, a network statistics engine and support for remote packet capture are available in WinPcap. WinPcap consists of a driver that extends the OS to provide low-level network access, and a library that is used to easily access the low-level network layers.

The SITL interface uses Berkeley Packet Filter (BPF) expressions. WinPcap uses BPF filtering in SITL operation. BPF is a software utility that is comprised of two main components: the data link tap and the packet filter. The data link tap collects copies of packets from the network device drivers and delivers them to the packet filter. The packet filter decides if a packet should be accepted based on the configuration of the SITL node module. The SITL node module has an attribute, called *Filter String*, that allows the user to define a BPF expression that will be used to configure the BPF packet filter. Figure 10 shows the SITL node module attributes.

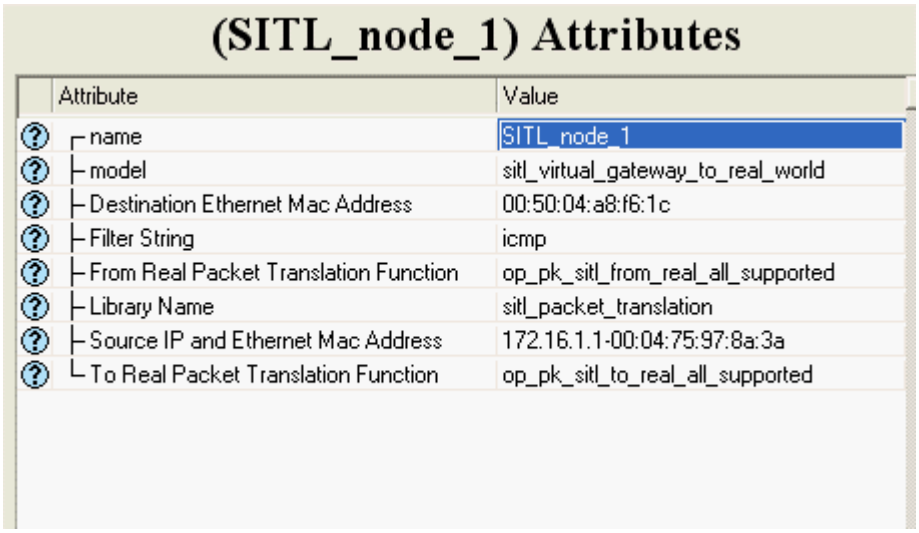

Figure 10: SITL Module Attributes

When a packet arrives at a network interface the link level device driver normally sends it up the system protocol stack. But when BPF is listening on this interface, the driver first calls BPF. BPF feeds the packet to each participating process filter as shown in Figure 11. This user-defined filter decides whether a packet is to be accepted and how many

bytes of each packet should be saved. For each filter that accepts the packet, BPF copies the requested amount of data to the buffer associated with that filter. The device driver then regains control. If the packet is not addressed to the local host, the driver returns from the interrupt. Otherwise, normal protocol processing proceeds. Figure 11 illustrates the BPF interface with the rest of the system

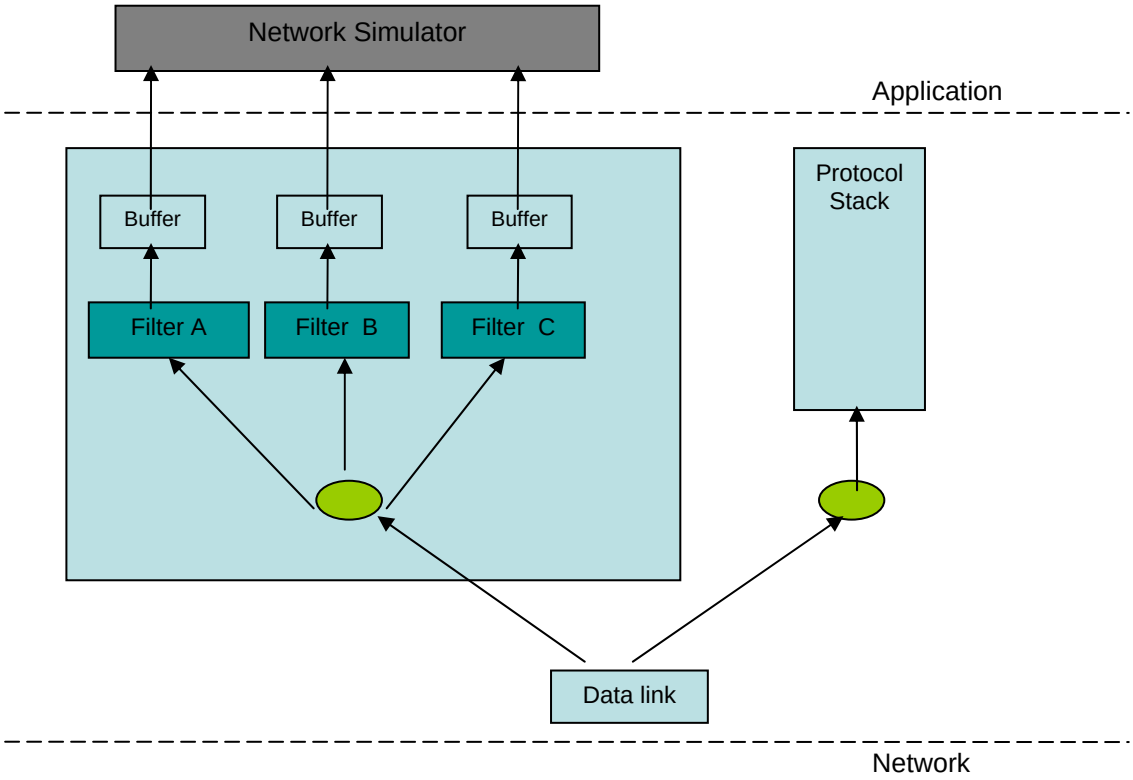

Figure 11: Berkeley Packet Filter (BPF) Interface

5 Demonstration Scenario Development

Our demonstration experiments are intended to demonstrate the operation of the VCSE framework. The FY05 demonstration experiments are based on scenarios that are comprised of a small number of nodes when compared to a real PCS. We begin with scenarios that have a small number of nodes to perform targeted experiments to evaluate the tool performance and its ability to generate useful data. These small-scale scenarios also help us to debug the software.

The two demonstration scenarios described below were developed during the FY05 VCSE Project.

5.1 VCSE Framework with OPNET Modeler Demonstration

A scenario was created to demonstrate the operation of the VCSE Framework co-simulation with our selected network simulator. This demonstration exercises the

framework's overall event management and bidirectional data exchange capability. This demonstration incorporates multiple PCS devices typical for an electric generation and distribution system. The PLC and Load Monitoring Node each have an Internet Protocol (IP) communication stack that supports IP communications. Each of the IP communication stacks are represented in OPNET Modeler and the application portion of each node is represented in both the VCSE Framework and OPNET Modeler. In this demonstration, data generated by the Load Monitoring Node is transmitted via a simulated IP network that is modeled and simulated with OPNET. A block diagram of the scenario is shown in Figure 12. Figure 13 illustrates a screenshot of the VCSE GUI during the demonstration runtime. In Figure 13 the image of the system under study represents the system being analyzed. For the FY05 activity this image is statically generated (i.e., created by hand) but will be dynamically generated (i.e., automatically generated based on system under study) in the FY06 product.

The following is a list of details describing the demonstration scenario shown in Figure 12. The goal is to show the basic interoperation of the network simulator (i.e., OPNET Modeler) with the VCSE framework and its associated control system models.

1. The demonstration network consists of a PLC, an IP network simulated with OPNET Modeler, two generators, and a load (i.e., town).

2. The PLC controls the generators directly (i.e., no communication network).

3. The PLC monitors the load (i.e., town) using an IP network (i.e., simulated network via OPNET Modeler) connected to a simple power monitor device.

4. Network events are scheduled through OPNET Modeler. The VCSE framework and OPNET simulator work together to interleave the event timing correctly.

5. The user has control over the load characteristics of the "town" via the GUI, and the PLC controls the generators to compensate for changes automatically.

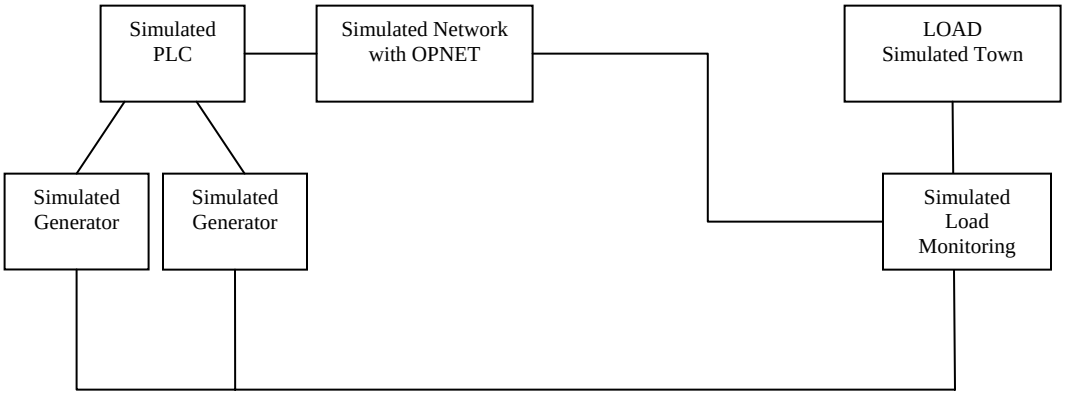

Figure 12: Demonstration Process Control System

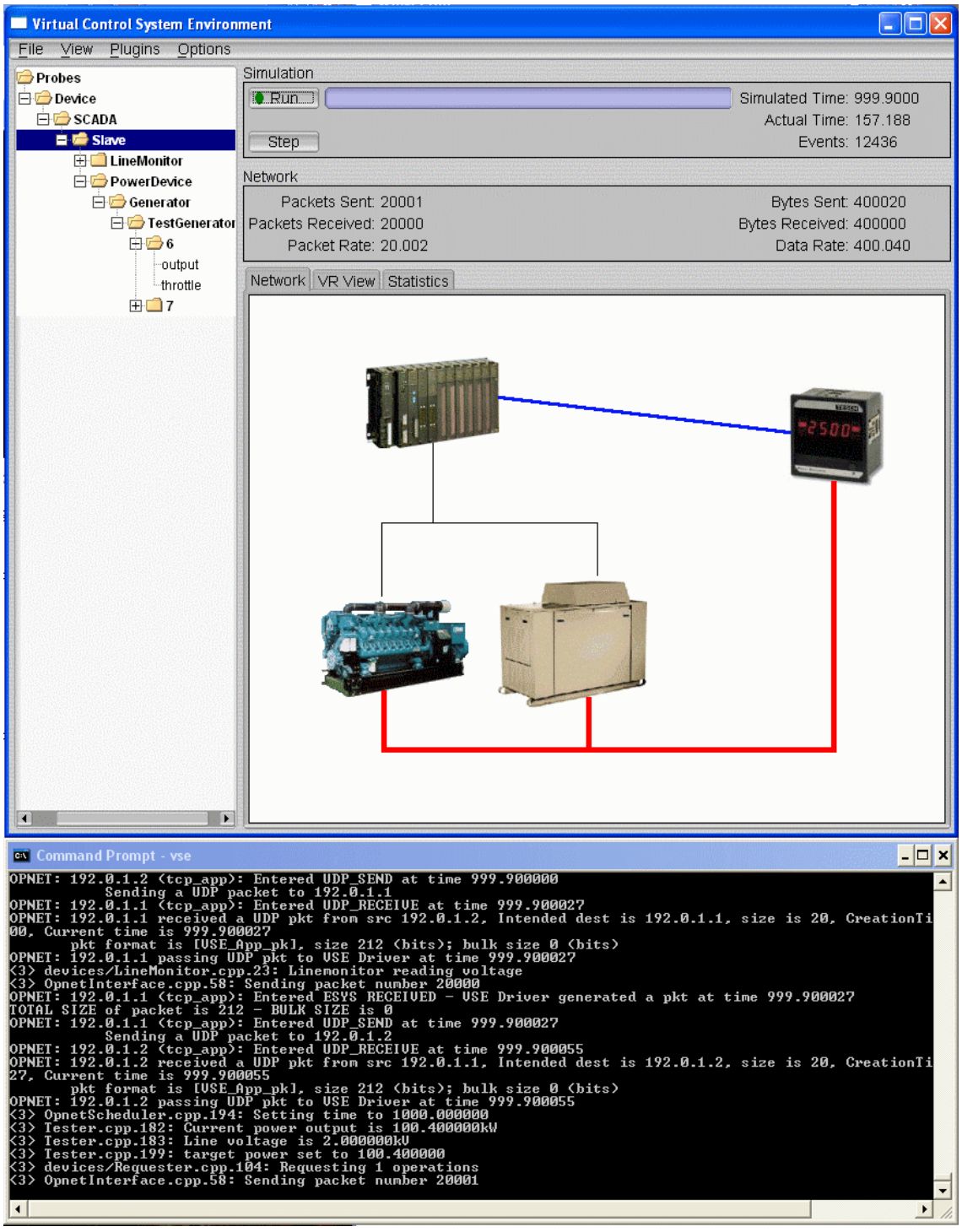

Figure 13: GUI Image of VCSE Demonstration

5.2 VCSE Framework with OPNET Modeler System-in-the-Loop Demonstration

The System-in-the-Loop capability was demonstrated with OPNET Modeler in two separate demos during the FY05 work. The demonstrations were configured to show a *real-simulated-real* scenario. A single computer running OPNET was interfaced to two real clients on separate subnets. The real clients exchanged data through the virtual network modeled in OPNET Modeler. The real clients also interacted with a virtual workstation modeled in the OPNET Modeler virtual network. The virtual network included multiple routers and a virtual client as shown in Figure 14.

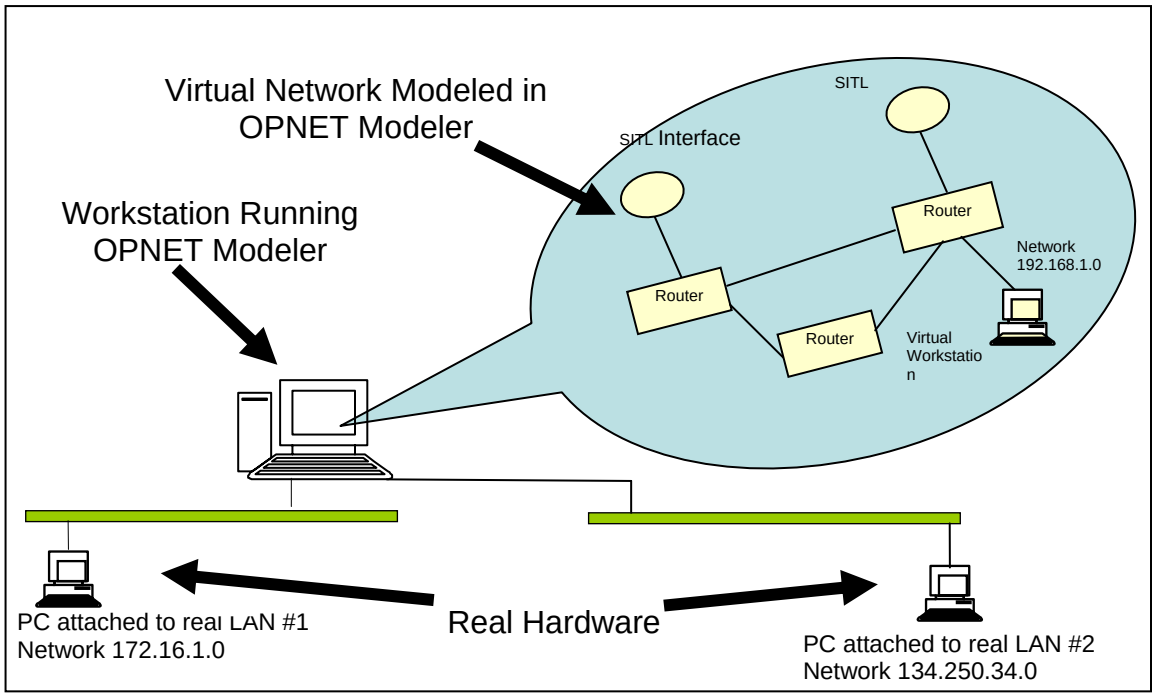

Figure 14: Diagram of System-in-the-Loop Demonstration

SITL Demonstration 1:

After the simulation is started each real workstation initiated communication with a virtual node. The communication was represented in the form of an Internet Control Message Protocol (ICMP) echo request (ping). The virtual node, after receiving the ping, responded with an ICMP echo response. The ping request originated on each real work station and after reaching the OPNET Modeler simulator interface was converted to a virtual packet and routed through the virtual network until it reached the intended virtual target. The virtual workstation then responded back to the originator of the ping request. The originator's local screen displayed a message that its ping request had been received with the evidence of the ping response.

In addition to the ping response, statistics from the simulator were gathered to keep a count of how many ping requests were received by the virtual node. This was shown in a graph that plotted the number of pings received vs. simulation time.

SITL Demonstration 2:

The second demonstration repeated the set-up of the first demo with the addition of running a *TCP Dump* utility that filtered router updates being sent from the virtual network within the simulator. These router updates showed the virtual networks being advertised into the real networks. Upon inspection the reviewer notes that the network addresses correspond to the virtual world. Table 2 shows a representative captured router advertisement originating form the virtual network and being advertised into the real network. The network interface that straddled both the real network and the virtual network was 172.16.1.1 (i.e., advertising node).

Table 2: RIP Routing Updates

Advertising node	IP broadcast address	RIP message ID	Destination address	Number of hops
172.16.1.1.route	172.16.1.255.route	Rip-resp 2	192.0.5.0	0
172.16.1.1.route	172.16.1.255.route	Rip-resp 4	192.168.1.0	2
172.16.1.1.route	172.16.1.255.route	Rip-resp 6	192.0.1.0	2

Another feature of the second demonstration showed the results of a failed link within the virtual network. The failed link was initiated after the establishment of connectivity between the external real network nodes and the participating virtual node. This showed the actual dependency that existed within the virtual communication network and the real network. Statistics gathered during this scenario, along with hardware displays, validated the communication failure along with the communication restoration.

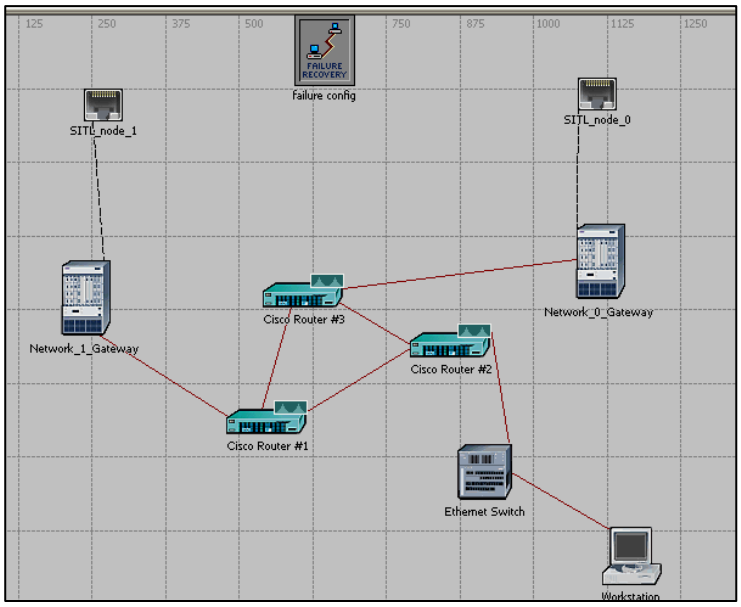

Figure 15: Screenshot of Demonstration Virtual Network

The FY05 program demonstrated a basic SITL interface capability. This capability will be enhanced during the FY06 effort to support industry-focused use cases.

6 Conclusions

In the FY05 task the design team identified an important analysis capability that can be used to examine the performance of PCS. The analysis capability is based on modeling and simulating the PCS along with its network communications and, in some cases, the physical system it supports as a system. The team has developed an environment that can provide a *control system centric* analysis capability. The developed environment, or Virtual Control System Environment (VCSE), provides a simulation framework that can integrate multiple simulators with a control system simulation and device library. The result is a tool that is extensible and able to provide analysis support for a wide range of PCS questions.

This report describes the primary advances made during the FY05 funded project. The main development is the VCSE modular framework which is based on a “plug-in” architecture. This framework provides the necessary flexibility in module selection to address a range of PCS analysis. The framework supports a co-simulation approach that integrates existing simulators (e.g., OPNET) into the VCSE to perform analysis of the control system. The VCSE obtains the benefits of existing analysis tools with extensive device model libraries that can be integrated into a single analysis environment. In addition, the co-simulation approach offers the benefit of existing simulators and models that have been used over time and thus have had their results vetted by other users.

The VCSE will be developed further with FY06 funding. The continuation funding will support developing the VCSE tool to meet the Technology Readiness Level 5 criteria. The development will focus on building PCS protocol and device model library to

support answering a broad range of questions in the analysis of PCS in critical infrastructures.

7 References

[1] T. Nolte, H. Hansson, “Aerospace, Gas Industry Need Wireless,” INTECH, May 2006.

[2] Prepared by Energetics of Columbia, MD, “Roadmap to Secure Control Systems in the Energy Sector,” January, 2006, http://www.controlsystemsroadmap.net/

[3] K.P. Birman, J. Chen, K. Hopkinson, R. Thomas, J. Thorp, R. van Renesse and W. Vogels, “Overcoming Communications Challenges in Software for Monitoring and Controlling Power Systems.” Submitted to Proc. of the IEEE, Special Issue on Energy Protection Systems. Sept. 2003.

[4] J. Farris and D. Nicol, "Evaluation of Secure Peer-to-Peer Overlay Routing for Survivable Scada Systems", Proceedings of the Winter Simulation Conference, December 5-8, 2004, in Washington DC.

[5] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, D. Coury; "EPOCHS: A Platform for Agent-Based Electric Power and Communication Simulation Built From Commercial Off-the-Shelf Components," Power Systems, IEEE Transactions on , vol.21, no.2, pp. 548- 558, May 2006.

[6] T. Skeie and S. Johannessen, “Ethernet in Substation Automation,” IEEE Control Systems Magazine 22(3):43-51, 2002.

[7] S. Lathrop, J. Hill, and J. Surdu, “Modeling Network Attacks,” Proc. 12th Conf. Behavior Representation in Modeling and Simulation, pp. 401-407, May 2003.

[8] B. Zeigler, D. Fulton, J. Nutaro, P. Hammonds; "M&S Enabled Testing of Distributed Systems: Beyond Interoperability to Combat Effectiveness Assessment": 9th Annual Modeling and Simulation Workshop, Dec. 8-11, 2003, ITEA White Sands Chapter.

[9] P. Sholander, “High-Level Requirements for Information Assurance Modeling and Simulation,” DRAFT, May 2006.

[10] R. Pollock, “Virtual Control System Environment (VCSE): Industry Questionnaire,” March 2006.

[11] Karlheinz Schwarz, “Standard IEC 61850 for substation automation and other power system applications,” International Conference: Power Systems and Communications Infrastructures for the future; Beijing, September 2002.

[12] OPNET Technologies. “OPNET Users Manual,” www.opnet.com.

[13] QualNet Home Page. www.qualnet.com

[14] NS2 Home Page. http://www.isi.edu/nsnam/ns/

[15] M. Berg, J. Stamp; “A Reference Model for Control and Automation Systems in Electric Power,” Sandia National Laboratories Report SAND 2005-1000C http://www.sandia.gov/scada/documents/sand_2005_1000C.pdf

8 Distribution

1 Department of Energy Headquarters
Attn: Hank Kenchington

1	MS0672	Erik Lee	5616
1	MS0672	John Michalski	5615
1	MS0672	Pete Sholander	5616
1	MS0672	Robert Pollock	5633
1	MS1368	Jennifer Depoy	5615
2	MS9018	Central Technical Files	8944
2	MS0899	Technical Library	4536