

Technical Learning and Integration of Interns in Advanced Protection Lab Space: Enhancements to Testbed and Experiments to Improve Workflows for Producing Datasets



Aaron Werth
Raymond Borges Hink
Emilio C. Piesciorovsky
Gary Hahn
Timothy Alhorn
Brett Billingsley
Mike Soare

December 2024



DOCUMENT AVAILABILITY

Online Access: US Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via <https://www.osti.gov>.

The public may also search the National Technical Information Service's [National Technical Reports Library \(NTRL\)](#) for reports not available in digital format.

DOE and DOE contractors should contact DOE's Office of Scientific and Technical Information (OSTI) for reports not currently available in digital format:

US Department of Energy
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831-0062
Telephone: (865) 576-8401
Fax: (865) 576-5728
Email: reports@osti.gov
Website: www.osti.gov

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Cyber Resilience and Intelligence
Electrification and Energy Infrastructure

**TECHNICAL LEARNING AND INTEGRATION OF INTERNS IN ADVANCED
PROTECTION LAB SPACE: ENHANCEMENTS TO TESTBED AND EXPERIMENTS
TO IMPROVE WORKFLOWS PRODUCING DATASETS**

Aaron Werth
Raymond Borges Hink
Emilio C Piesciorovsky
Gary Hahn
Timothy Alhorn
Brett Billingsley
Mike Soare

December 2024

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-BATTELLE LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

| | |
|--|-----|
| LIST OF FIGURES | iv |
| LIST OF TABLES | iv |
| ABBREVIATIONS | v |
| ABSTRACT..... | 1 |
| 1. INTRODUCTION | 2 |
| 2. INTERN PROGRAMS AT ORNL..... | 2 |
| 2.1 INTEGRATION OF ADVANCED PROTECTION LAB WITH INTERNS | 2 |
| 2.2 AN OPPORTUNITY TO CREATE A HIGHLY SKILLED WORKFORCE FOR THE ELECTRICAL GRID | 3 |
| 3. ELECTRICAL SUBSTATION GRID TESTBED WITH CYBER GRID GUARD | 7 |
| 3.1 INTRODUCTION | 7 |
| 3.2 ONE-LINE DIAGRAM AND EQUIPMENT | 7 |
| 3.3 TESTBED AND ARCHITECTURE | 9 |
| 3.4 THREE-LINE DIAGRAM | 10 |
| 3.5 CYBER GRID GUARD | 11 |
| 4. ENHANCEMENTS TO NETWORKING IN TESTBED..... | 12 |
| 5. AUTOMATION FRAMEWORK FOR EXPERIMENTS | 14 |
| 6. PREPARING DATASETS | 18 |
| 7. CONCLUSIONS AND FUTURE WORKS | 21 |
| 8. ACKNOWLEDGMENTS | 22 |
| 9. REFERENCES | 22 |
| APPENDIX A. INSTRUCTIONS FOR NETWORKING | A-1 |
| APPENDIX B. AUTOMATION FRAMEWORK | B-1 |

LIST OF FIGURES

| | |
|---|-----|
| Figure 1. Dr. Aaron Werth (mentor, second from right) with summer 2024 interns Timothy Alhorn (left), Brett Billingsley, and Mike Soare (right) in the Advanced Protection Lab..... | 3 |
| Figure 2. Bloom’s taxonomy for interns in Advanced Protection Lab. | 5 |
| Figure 3. Mike Soare’s poster. | 5 |
| Figure 4. Brett Billingsley’s poster. | 6 |
| Figure 5. Timothy Alhorn’s poster. | 6 |
| Figure 6. (a) One-line diagram and (b) equipment rack..... | 8 |
| Figure 7. Electrical substation/grid testbed and workstations..... | 9 |
| Figure 8. Devices in Testbed. | 10 |
| Figure 9. Architecture of the electrical substation/grid testbed. | 11 |
| Figure 10. Illustration of multiple subnets: one for each electric utility, connected via a router..... | 13 |
| Figure 11. Multiple stages for launching experiments..... | 15 |
| Figure 12. Diagram showing activity of the agent..... | 16 |
| Figure 13. The developed workflow process. | 21 |
| Figure A-1. Router with various ports | A-1 |
| Figure A-2. From the Mac connected to the Utility C switch (192.168.102.0/24). | A-2 |
| Figure A-3. From the Ubuntu Laptop connected to the Utility B switch (192.168.101.0/24). | A-3 |
| Figure A-4. With the Mac acting as the iPerf server..... | A-3 |
| Figure A-5. Output on the server when receiving a connection from the other device. | A-3 |
| Figure A-6. Output with the Ubuntu Laptop acting as the server with multiple connections from the other device. | A-4 |
| Figure A-7. Output when the Mac is the client and using the default client command and the command with -t for different times. | A-5 |
| Figure A-8. Output with the Mac as the client and using -n for different amounts of data. | A-5 |
| Figure A-9. Output with the Ubuntu Laptop as the client and using -t for different times..... | A-6 |
| Figure A-10. Management Port on Router..... | A-7 |

LIST OF TABLES

| | |
|---|----|
| Table 1. Latency between utilities. | 13 |
| Table 2. Bandwidth between utilities..... | 13 |
| Table 3. Summary of launched cyberattacks. | 15 |

ABBREVIATIONS

| | |
|--------|---|
| DNP3 | distributed network protocol 3 |
| GOOSE | generic object-oriented substation event |
| HAT | Hoeffding adaptive tree |
| IEC | International Electrotechnical Commission |
| IED | intelligent electronic device |
| IRIG-B | inter-range instrumentation group time code B |
| ML | machine learning |
| NNGE | non-nested generalized exemplars classification |
| PTP | precision-time protocol |
| SCADA | supervisory control and data acquisition |
| SEL | Schweitzer Engineering Laboratories |
| SSH | secure shell |
| SV | sampled value |

ABSTRACT

This report presents a successful technical learning integration of student interns in the Advanced Protection Laboratory space, located in the Grid Research Integration and Deployment Center (GRID-C) at the Department of Energy's (DOE's) Oak Ridge National Laboratory (ORNL). The Advanced Protection Laboratory was created for the primary goal of supporting DOE's research projects and technical staff at ORNL. As a secondary goal, the space was used for collaborating with ORNL's intern programs, providing support to the lab's mentors and student interns. In 2024, three student interns spent a summer in the Advanced Protection lab space and were involved in the DarkNet Distributed Ledger Technology (DLT) project. The students had a great opportunity to gain hands-on experience with communication and protective relay equipment focused on information technology, data analytics, and cybersecurity. Experiences in the lab space with real equipment and software integration offer education and professional development for students, which is especially important because of a need in the energy industry to recruit highly skilled power and communication engineers.

This report also covers the interns' exploratory activities and some of their results. They focused on three principal areas of interest and developed prototype implementations for these activities: (1) Enhancements to Networking in Testbed, (2) Automation Framework for Experiments, and (3) Preparing Datasets. The unifying theme behind the activities is to create an automated workflow. Enhancements are made to the testbed so that more accurate experiments can be performed. The purposes of the automation framework are to perform experiments more efficiently and to produce raw data. Finally, automation through scripting curates and collates raw data to create labeled datasets for research and academic purposes that can include machine learning (ML) and Artificial Intelligence (AI).

1. INTRODUCTION

This report documents internship experiences in the Advanced Protection Laboratory at Oak Ridge National Laboratory's (ORNL's) Grid Research Integration and Deployment Center (GRID-C), a facility in ORNL's Hardin Valley Campus. The report describes the experiences of the interns involved in the Darknet Distributed Ledger Technology (DLT) Project and the ways that the experiences in a laboratory can help them in their education and development as researchers and technical professionals. Furthermore, the report discusses the importance of such educational experiences for workforce development as concepts like the smart grid have become more prominent and how the students' experiments fit into Bloom's taxonomy. As a second objective, the report also documents in detail the progress in the testbed and updated methods for experiments. The work presented in this report involved the summer interns. ORNL staff have worked to enhance the testbed and to develop workflows that go from experiments to the creation of datasets. The interns helped staff in various activities to support these enhancements, such as development of prototypes and updated device configurations.

Researchers previously created a basic testbed with various meters and relays starting in Phase II of Darknet and have since expanded the testbed to include multiple utilities. The testbed in use is the electric utility substation testbed at the Advanced Protection Laboratory. This report documents further enhancements to the testbed in terms of communication and networking and describes automated approaches to performing experiments, collecting resulting data, and curating the data to create datasets. These datasets can be used for training machine learning (ML) algorithms and AI. The project is divided into three main parts: (1) Enhancements to Networking in Testbed, (2) Automation Framework for Experiments, and (3) Preparing Datasets.

2. INTERN PROGRAMS AT ORNL

Every year at ORNL, the Technical and Professional Internship Programs provide opportunities for undergraduate and graduate students and recent graduates to engage with ORNL projects and connect with ORNL researchers and professional staff [1]. ORNL is the largest Department of Energy (DOE) science laboratory. Through its internship programs, ORNL offers opportunities to gain experience in information technology, facilities and operations, human resources, marketing, graphic design, data analytics, cyber security, legal counsel, communications, finance, health services, and many other professional fields. In the case of this report, the areas the interns focused on were information technology, data analytics, and cybersecurity.

2.1 INTEGRATION OF ADVANCED PROTECTION LAB WITH INTERNS

The Advanced Protection Laboratory welcomed multiple student interns in the past two summers. In 2024, three student interns mentored by Dr. Aaron Werth and Raymond Borges Hink came to the lab to assist in activities. These interns—Timothy Alhorn, Brett Billingsley, and Mike Soare—supported the Darknet DLT Project. It was an excellent experience for them because they could perform electrical grid use case simulations and tests in an electrical substation grid testbed using the Cyber Grid Guard (CGG) with DLT. Technical staff member Gary Hahn, who has worked in the testbed for more than 3 years, collaborated with the mentors and students in important communication and software aspects of the project. Dr. Emilio C. Piesciorovsky, who manages the Advanced Protection Lab, provided guidance and safety training to the interns and gave an overview of the testbed and utilities so that the interns could have the proper background information to help in their activities. This opportunity provided the interns experience with communication, protection, and metering devices that are usually installed in electrical grids and substations. The Advanced Protection Lab includes an electrical substation grid testbed with real-time simulators and more than 12 relays/meters in the loop used for research applications in multiple

projects. Figure 1 shows the electrical substation grid testbed, and Dr. Aaron Werth with ORNL interns in the Advanced Protection Lab.



Figure 1. Dr. Aaron Werth (mentor, second from right) with summer 2024 interns Timothy Alhorn (left), Brett Billingsley, and Mike Soare (right) in the Advanced Protection Lab.

2.2 AN OPPORTUNITY TO CREATE A HIGHLY SKILLED WORKFORCE FOR THE ELECTRICAL GRID

In the US electrical grid, several occurrences have all required a highly skilled power workforce: (1) the increasing number of distributed energy resources (DERs); (2) the rise in the number of protective relays and meters; and (3) new technologies in sensors and protocols of communication [2]. This workforce is necessary to succeed in grid implementation focusing on certain main topics (measurement, control, protection, and communication). The penetration of DERs have yielded new topics of interest (e.g., wind generator turbine farms, inverter-based photovoltaic array farms). A considerable number of power engineers currently employed in electrical utilities are eligible for retirement [3], and the pandemic has accelerated the retirement of employees in some electrical contractors, energy equipment manufacturers, and electrical industries. Electrical utilities need new power and communication engineers to maintain a utility workforce and approximately two to three times that number to satisfy the whole market demand [2]. The current energy industry is in a competitive market in which highly skilled engineers trained in protection and communications are difficult to recruit and retain. As a result, education and research communities have a valuable opportunity to collaborate on educational projects.

The Advanced Protection Laboratory was developed to perform research activities for projects related to protective relaying that cross different research areas such as adaptive protection, anomaly event detection, power line sensors, distributed ledger technology, and protective relay commissioning. Increasing the number of DERs to reach zero-carbon emission goals and using more protective relays and power meters on modern electrical grids both require more highly skilled engineers. In the Advanced Protection Lab, like in other electrical grid labs [4], protection, control, measurement, and communication testing applications require electrical substation equipment and professionals with multidisciplinary education and technical skills [5].

In power engineering education, training tomorrow's workforce requires introducing students to both current and new smart grid concepts. Training would provide them with knowledge in a diverse set of

technical areas (e.g., communications, controls, electronics, instrumentation, electromagnetics, computers) and help them understand how these areas integrate within the electric power grid [6]. The evolution of the traditional electricity grid into a state-of-the-art power grid will need innovation in several dimensions (e.g., seamless integration of renewable energy sources, management of intermittent power supplies, real-time demand response, energy pricing strategy) because the grid configuration will change from a central broadcasting network into a more distributed, dynamic network with two-way energy transmission [7].

Power system engineers deal with many new challenges such as solar power installations, wind power generation, superconductors used in substations, more use of High Voltage Direct Current (HVDC), use of phasor measurement units (PMUs), increased use of digital relays, digital smart meters at residences and businesses, computer agents connected to smart meters, and communications to form a modern power grid [8] (i.e., in areas like power system protection, renewable energy sources, power electronics, computer program, communication, and cybersecurity).

In 1956, Benjamin Bloom developed a schematic, or taxonomy, of the learning process represented by six levels [9]: knowledge, comprehension or understanding, application, analysis, synthesis, and evaluation. This process explains how students learn and to what depth of understanding learning can ultimately be applied. Based on Bloom's taxonomy, the Advanced Protection Lab included learning safety and technical activities for the interns. The first group of activities involved performing the corresponding lab site training that defines possible hazards in the lab space based on areas, equipment, and testbeds. The second group of activities presented a review of electrical equipment and distribution power grid concepts. Breaker operations and bus measurement concepts are then introduced to interns using the testbed and equipment for electrical substations based on protective relays; furthermore, power meters installed on the testbed are discussed, the implementation of communication devices and protocols are presented, and the use of real-time simulators to represent the electrical grid with protective relays, meters, and communication devices-in-the-loop is presented.

These two groups of activities—one based on safety, and another based on technical aspects—were crucial for interns to achieve the knowledge, understanding, application, evaluation, and synthesis taxonomy. The interest level was added to Bloom's taxonomy to promote engineering education. Figure 2 shows the Advanced Protection Lab's objectives based on Bloom's taxonomy.

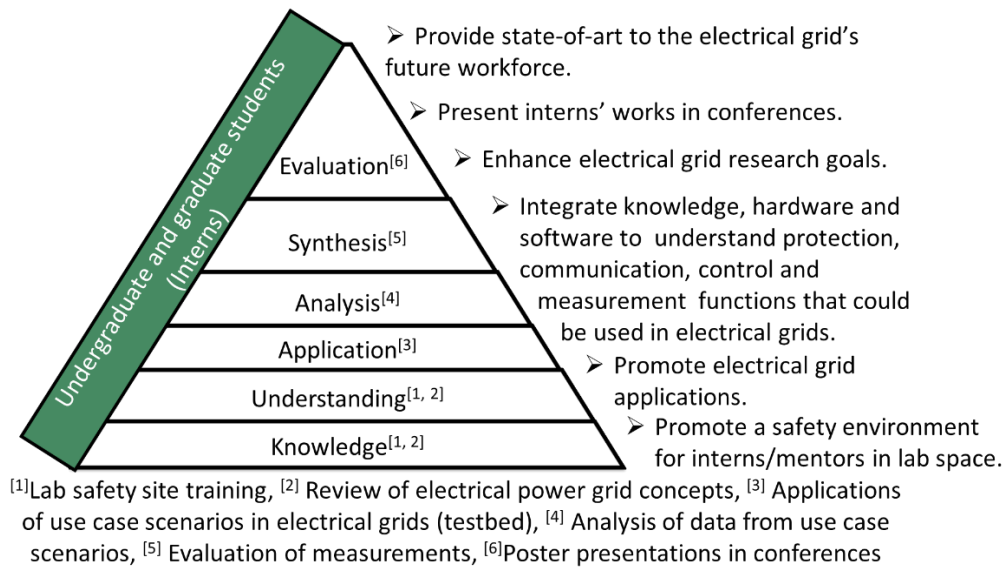


Figure 2. Bloom's taxonomy for interns in Advanced Protection Lab.

After performing their tasks in the Advanced Protection Lab, each intern individually presented a poster at the DOE Cybersecurity and Technology Innovation Conference, July 29–August 1, 2024, in Dallas, Texas (<https://www.doecybercon.com/>). Figure 3, Figure 4, and Figure 5 show the interns' posters presented as deliverables that met the Synthesis phase of the Advanced Protection Lab objectives based on Bloom's Taxonomy.

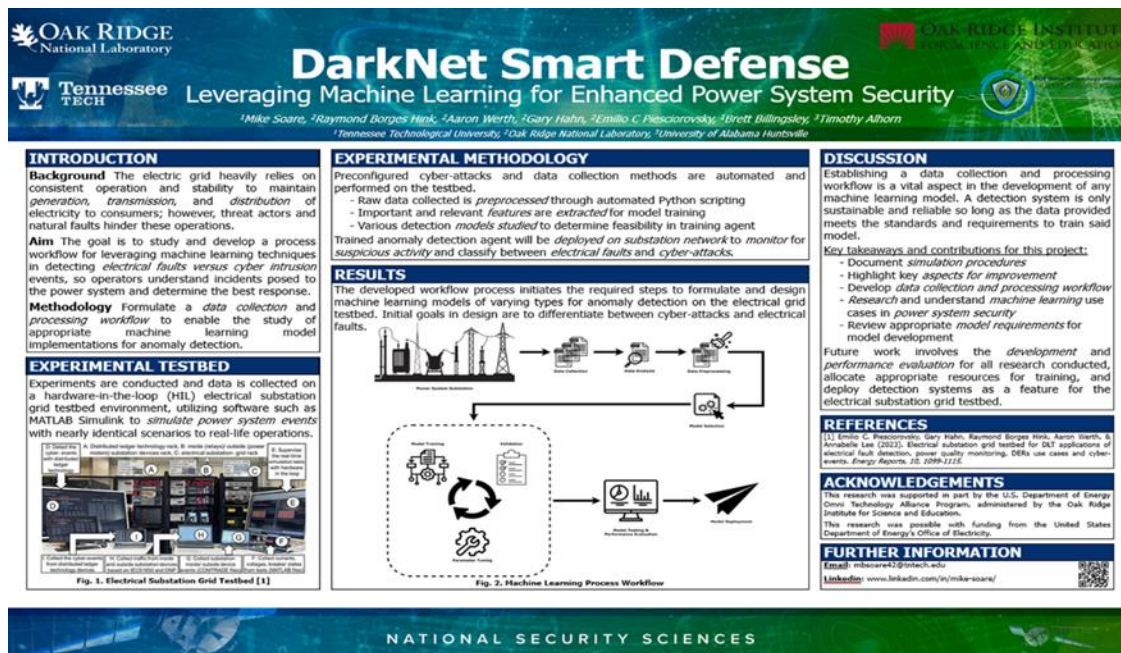


Figure 3. Mike Soare's poster.

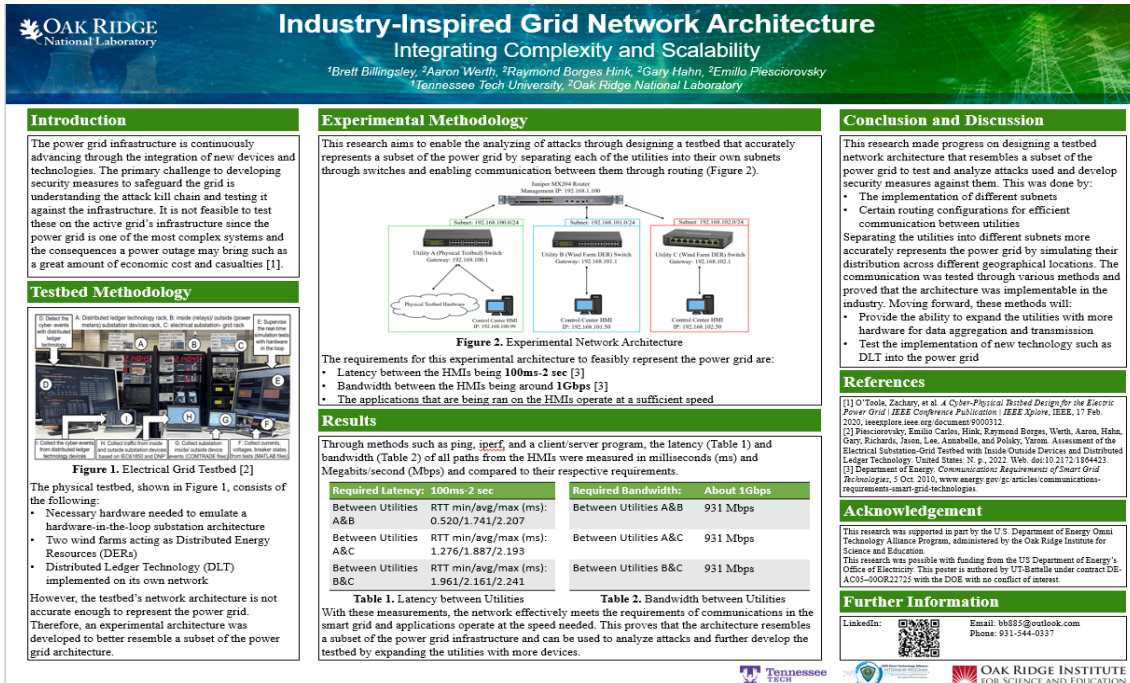


Figure 4. Brett Billingsley's poster.

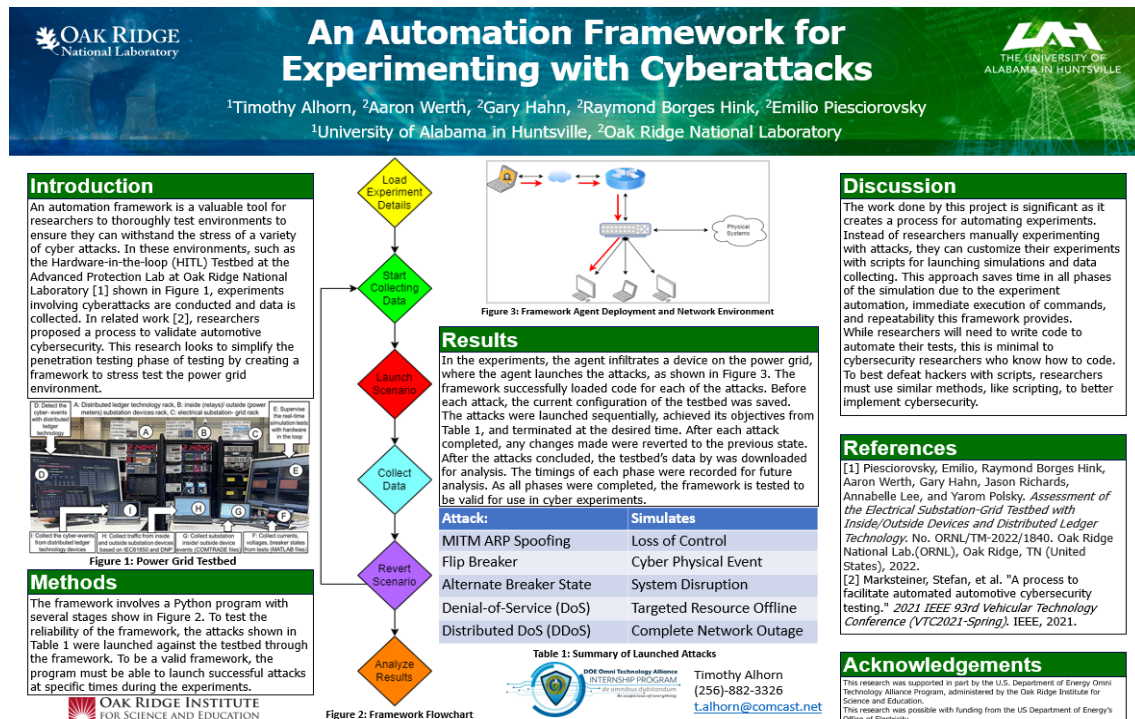


Figure 5. Timothy Alhorn's poster.

3. ELECTRICAL SUBSTATION GRID TESTBED WITH CYBER GRID GUARD

3.1 INTRODUCTION

Electrical utilities continue to deploy more intelligent electronic devices (IEDs) inside and outside electrical substations that are associated with DERs. The integrity and confidentiality of data from IEDs are crucial, and DLT could improve microgrid resilience by helping to make these data more secure. Although general monitoring for DLT applications could be evaluated in operational electric grids, other DLT research applications such as defense against undesired cyber-events and/or electrical fault detection are not likely to be performed in a real infrastructure because of possible risks to network security and damaged equipment. This study presented a testbed with integrated DLT; the testbed is based on a CGG system using DLT with a real-time simulator, power meters, and protective relays in-the-loop.

As market penetration of DERs increases, so have measurements that rely on communications between IEDs inside and outside the substation perimeter. Dynamic management capabilities are possible with customer-owned and managed DERs and with deployment of smart sensors with IEDs. Therefore, DLT applications could be developed to focus on control, measurement, and protection. The integrity and confidentiality of data and control commands between IEDs are important. The establishment of, and reliance on, communications across the utility–customer interface to enhance grid dispatch and control has created a significant threat vector for secure power system operations, such as cyber intrusion and/or communications failures.

3.2 ONE-LINE DIAGRAM AND EQUIPMENT

A typical substation testbed with a DER and IEDs was managed for controlling and monitoring applications using DLT-based applications. This testbed used a software model–simulated power system that performed electrical faults and cyber-events. The objective of the testbed was to determine whether the architecture of the CGG system using DLT was effective in controlling the utility grid and managing its assets/equipment. The one-line diagram below represents the design of a 34.5 kV (primary)/12.47 kV (secondary) electrical substation. The electrical substation was based on a sectionalized bus configuration [10], with two power transformers and two radial feeders, where one radial feeder was connected to a customer-owned DER (wind farm), as shown in Figure 6a. This one-line diagram was based on a previous published report [11], and the wind farm was added. Utility A included an electrical substation and distribution grid that had the control center. Utility A’s electrical substation had two power transformers of 10 MVA and primary and secondary voltages of 34.5 and 12.47 kV, respectively.

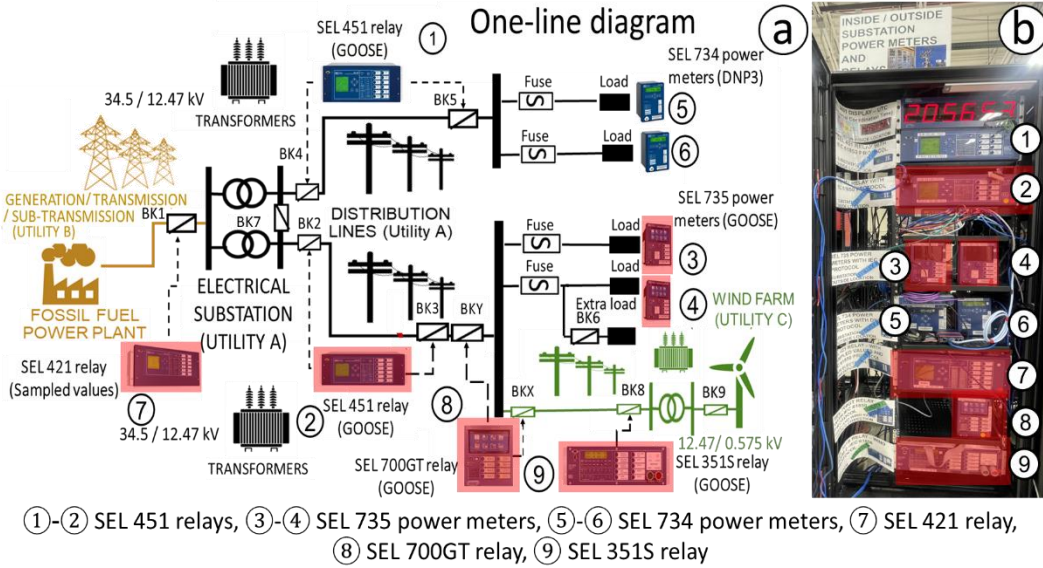


Figure 6. (a) One-line diagram and (b) equipment rack.

The electrical grid was a 12.47 kV power system with load feeders that are usually connected in a radial configuration; however, the load feeders could be connected to the wind farm (Utility C). Utility C was the customer-owned DER with six 1.5 MW wind turbines (two 9 MW wind farms), and Utility B was the main source based on a fossil fuel power plant. In the fuses, feeders were configured at the Schweitzer Engineering Laboratories (SEL) 735 [12] and SEL 734 [13] power meters with generic object-oriented substation event (GOOSE) International Electrotechnical Commission (IEC) 61850 and DNP3 protocols, respectively. The SEL 421 [14] relay at the 34.5 kV side of the electrical substation was configured with the sampled values (SV) IEC 61850 protocol; the SEL 451 [15], SEL 351S [16], and SEL 700GT [17] relays were configured with the GOOSE protocol. The SEL 700GT relay was installed in the point of common coupling between the main electrical grid side and the wind farm side. These protective relays measured phase voltages and currents; real, reactive, and apparent power; total power factor; frequency; and breaker states collected by the control center (Utility A). The CGG system was operated using the red highlighted relays and meters shown in Figure 6. In Figure 6b, the protective relays and power meters of the one-line diagram are shown in the equipment rack. These IEDs were wired to a real-time simulator and communication devices connected to a synchronized-time system. Table 1 shows the main research areas, electrical grid characteristics, references, and main deliverables of applications using the advanced electrical substation grid testbed, which was created and led by Dr. Piesciorovsky. This advanced testbed was used in different research areas like harmonics metering [18], distributed ledger technology [19], relays and meters commissioning in electrical grid testbeds [20], and electrical fault-type detections with CGG [21].

This advanced testbed was used with different DERs. In [18], the testbed was used as a substation/grid with a wind farm to assess measured total harmonic distortions for meters and relays with different sampling frequencies. In [19], the testbed was used as a substation/grid with a customer-owned wind farm using CGG to monitor power quality using DLT. In [20], the testbed was used as a substation/grid without DERs to assess a method for commissioning relays, meters, and real-time simulator testbeds for different power system applications. In [21], the testbed was used as an electrical substation grid with power lines and feeders to detect faulted phases in a power line using a feeder relay and CGG for different electrical faults. In this occasion the electrical substation grid testbed was used by interns participating in various communication and cyber-events projects, using communication equipment,

defining use case scenarios, collecting event results, and analyzing data to present their posters at the DOE Cybersecurity and Technology Innovation Conference.

3.3 TESTBED AND ARCHITECTURE

The testbed (Figure 7) collected and recorded relevant data during the simulations. The devices (e.g., protective relays, power meters) produced data that were synchronized with a time source. The electrical substation grid testbed had six computers at desks and on racks. The four-computer, desk-based workstations are shown in Figure 7: the host computer is located at (6), human machine interface (HMI) computer at (7), traffic network computer at (8), and SCADA computer at (9).

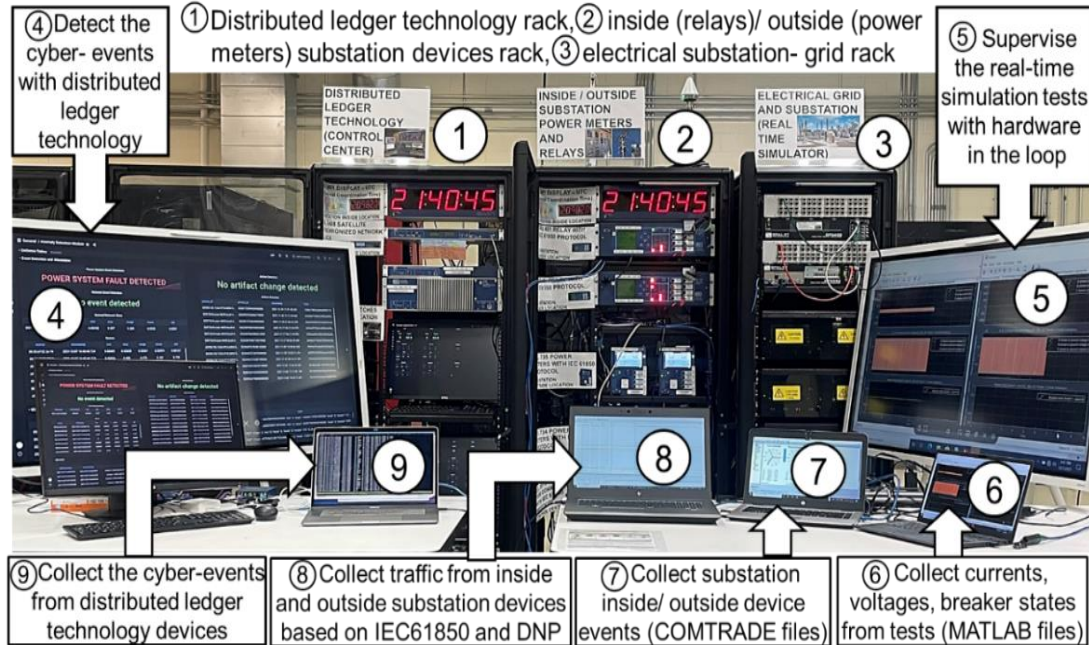


Figure 7. Electrical substation/grid testbed and workstations.

Figure 8 shows the control center HMI, local substation HMI, Blueframe virtual machine, and EmSense high-speed emulated sensor servers/computers in the rack for the CGG system. The architecture of the electrical substation/grid with customer-owned DERs (wind farms) and the CGG system for the testbed had four layers (Figure 8). The physical layer (level 1) included the power lines, breakers, transformers, feeder loads, and other power system elements simulated by the real-time simulator. The protection and metering layer (level 2) contained the hardware-in-the-loop, represented by the protective relays and power meters. The automation layer (level 3) contained the remote terminal units and the ethernet switches. The control layer (level 4) contained the supervisory control and data acquisition, HMI, and synchronized-time system for the CGG system.

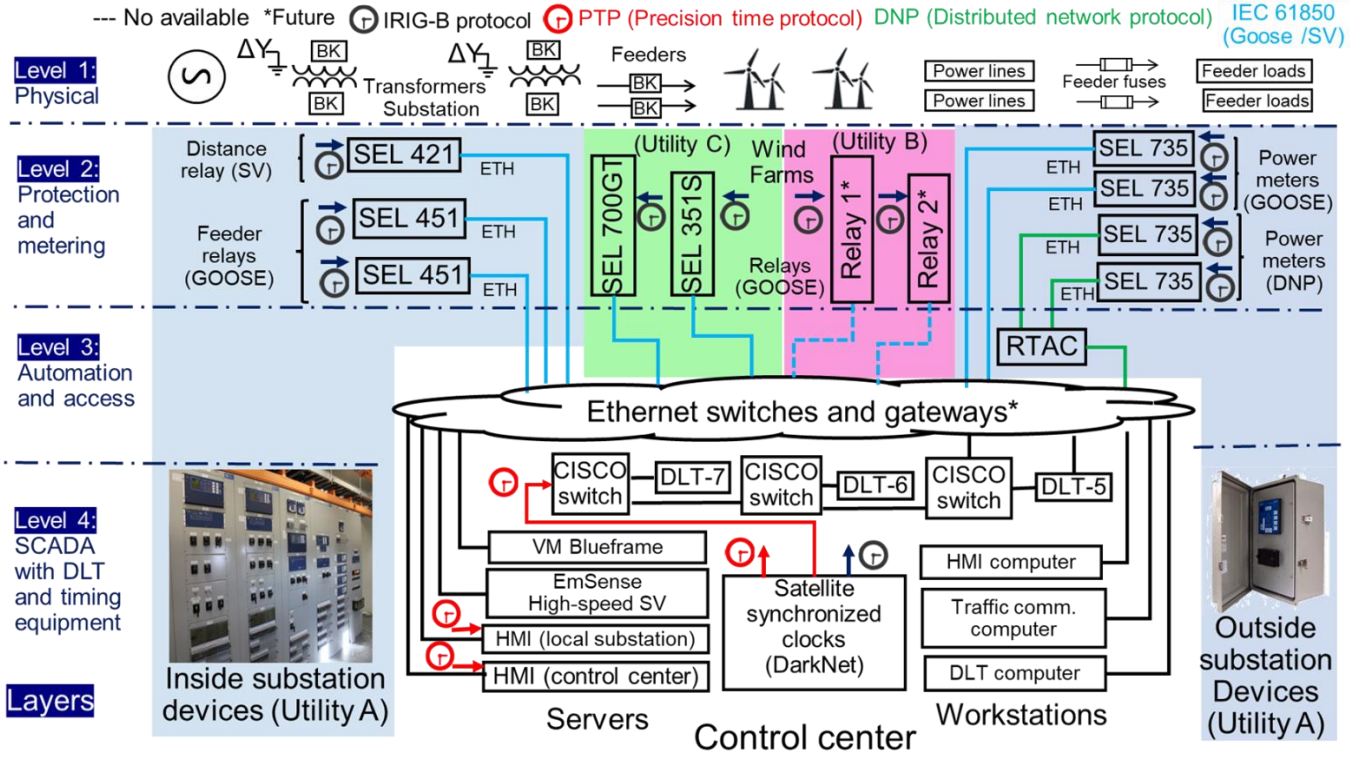


Figure 8. Devices in Testbed.

The synchronized-time protocols used in this architecture implemented the precision-time protocol (PTP) and inter-range instrumentation group time code B (IRIG-B) signals. The PTP communication was implemented in the CGG through the Ethernet network, and the IRIG-B communication was implemented at the power meters and feeder relays. The SEL 421 protective relay transmitted IEC 61850 SV messages. The SEL 451 protective relays and SEL 735 power meters transmitted IEC 61850 GOOSE messages, and the SEL 734 power meters transmitted distributed network protocol 3 (DNP3) messages.

3.4 THREE-LINE DIAGRAM

The three-line diagram was created in a project using RT-LAB, a software product of Opal-RT. This project involves MATLAB/Simulink models to run the tests with the real-time simulator and the IEDs in-the-loop. The electrical substation grid (Utility A) with the customer-owned wind farm (Utility C) is shown in Figure . Utility B is a fossil fuel power plant that feeds the main electrical substation of Utility A. This electrical substation (Utility A) was a 34.5/12.47 kV primary/voltage power system, respectively, and the wind farm (Utility C) was on a 0.575/12.47 kV primary/voltage power system, respectively. The wind farm had doubly-fed induction generator wind turbines. The wind farm of Utility C comprised six 1.5 MW doubly-fed induction generator wind turbines each (two wind farms totaling 9 MW each). Utility A had two 34.5/12.47 kV power transformers of 10 MVA connected in parallel, and two feeder breakers of 12.47 kV controlled by two SEL 451 protective relays in-the-loop. Thus, the phase currents and voltages were collected from the feeder breaker locations. Each feeder breaker was connected to a radial power grid, with two 12.47 kV power lines connected to the feeder loads. One power line had two power loads with 50 T fuses [22], and the other power line had two power loads with 100 T fuses [22]. The phase currents and voltages for the 50 and 100 T fuses were measured with the SEL 734 and SEL 735 power meters, respectively, based on the one-line diagram of Figure 6.

In Figure , the electrical substation grid (Utility A) could be connected to a DER (Utility C). Utility B was represented by a fossil fuel power plant generator, transmission, and subtransmission block (Figure a). Utility A consisted of the electrical substation (Figure b), power lines (Figure c), load feeders (Figure d), and wind farm (Figure e). The fault block (Figure i) is set before running the simulations. The substation feeder breakers (Figure h) are controlled by the SEL 451 relays, and the point of common coupling between the main utility grid side and wind farm side was controlled by a SEL 700GT relay. In the load feeders (Figure d), the phase voltages and currents are measured by the SEL 735 power meters.

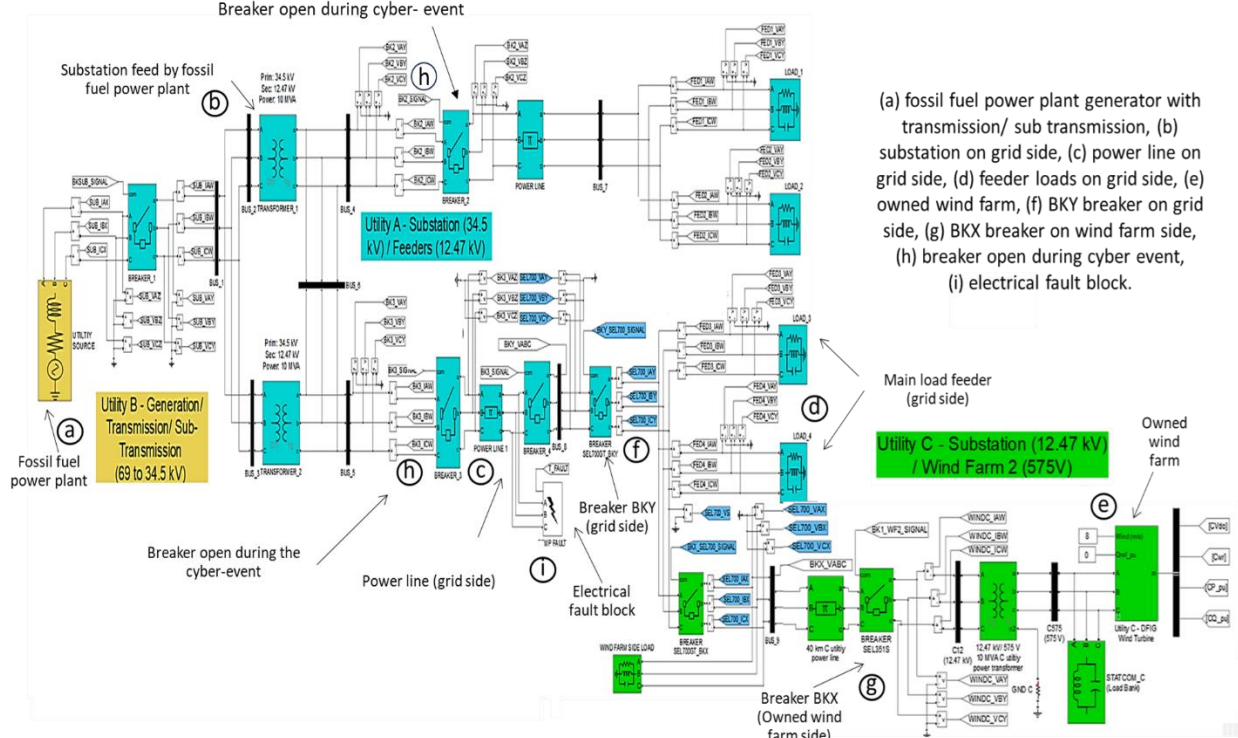


Figure 9. Architecture of the electrical substation/grid testbed.

3.5 CYBER GRID GUARD

CGG is a modular software system designed to protect data integrity in electrical utility substations. The primary architecture involves modules that capture and process diverse types of data, including communication protocols such as DNP3, IEC 61850 GOOSE and SV, Telnet, secure shell (SSH), and Modbus, as well as file artifacts like configuration settings, logs, event records (such as COMTRADE), and others downloaded from IEDs via FTP or secure FTP (SFTP). These data are then formatted into JSON before storage to facilitate easier querying and analysis.

DLT refers to platforms where multiple participants can record, store, and share their data across network nodes in a decentralized manner. This structure ensures that all records are secure, transparent, and resistant to alteration. It can be used to create an immutable history of all data transactions, which ensures that any attempt at modification is detected owing to its consensus-based validation process. However, DLTs are typically not optimized for querying and can induce data storage costs and processing latency when compared with traditional databases designed for managing structured data efficiently. DLT can complement traditional databases by providing an additional layer of protection against data manipulation and tampering. CGG implements this complimentary storage layer by combining an off-chain database using PostgreSQL with the TimescaleDB extension to efficiently handle timestamped measurements from

IEDs, and a Hyperledger Fabric DLT stores hashes of the raw data using the SHA256 cryptographic hash function. This design enhances data security by providing trust-anchoring through the ledger's tamper-resistant nature, while reducing storage requirements and processing associated with storing raw data in the ledger. Because storing data is the focus of the system, CGG relies on other technologies and methods such as Transport Layer Security (TLS), virtual private networks (VPNs), medium access control (MAC), and isolated networks and connections to ensure data integrity during transmission.

CGG uses various anomaly detection methods implemented as separate modules that trigger attestation checks. These checks are based on network statistics, power quality thresholds using GOOSE data measurements, and "diffs" (the results of using the diff command to find differences) of files within compressed file archive artifacts. The integrity of stored data is ensured by comparing recomputed hashes from the off-chain store with those in the ledger. Attestation check intervals can be configured to occur periodically, randomly, and/or upon specific events or anomalies. If an anomaly is detected, an alert is generated, and an attestation check is initiated. In case of a failed attestation check, another alert is issued. Further actions are currently user-initiated but could potentially be automated in future versions.

4. ENHANCEMENTS TO NETWORKING IN TESTBED

One of the main objectives within DarkNet is to continue to develop the testbed to make it more accurate in representing utilities for experiments. The testbed must be accurate in its electrical and communication architectures to facilitate developing new technology, such as CGG and its more advanced versions and features. CGG uses DLT to enable deploying technology on actual power systems after testing the technology on a testbed in the lab.

The world's power grid infrastructure is becoming more advanced through the integration of new devices and technologies. The primary challenge to developing security measures to safeguard the grid is understanding cyber events and testing them against the infrastructure. It is not feasible to test these on the active grid's infrastructure because the power grid is complex, and the consequences of a power outage can incur a great economic cost and casualties [23].

The physical testbed, shown in Figure 7, consists of the following: (1) necessary hardware needed to emulate a hardware-in-the-loop substation architecture, (2) two wind farms, and (3) DLT implemented on its own network. However, the testbed's network architecture is not accurate enough to represent the power grid. Therefore, an experimental architecture was developed to better resemble a subset of the power grid architecture. The requirements for this experimental network architecture to feasibly represent the power grid are latency between the HMIs of 100 ms to 2 s [24] and bandwidth between the HMIs of around 1 Gbps [24]. The applications running on the HMIs operate at a sufficient speed. Figure shows an illustration of how the various subnets correspond to different electric utilities.

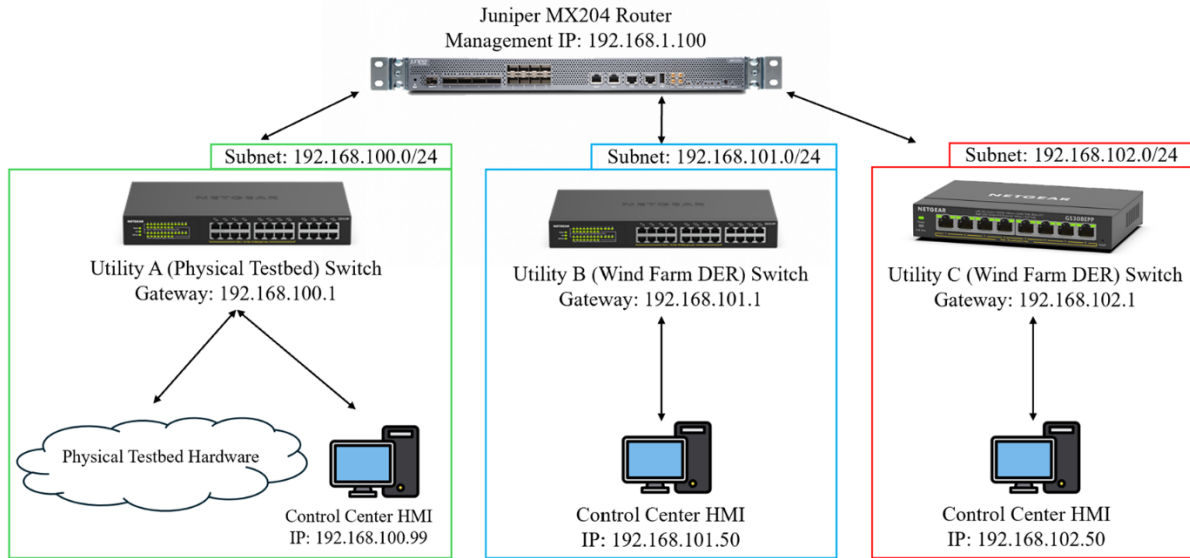


Figure 10. Illustration of multiple subnets: one for each electric utility, connected via a router.

Through methods such as ping, iperf, and a client/server program, the latency (Table 1) and bandwidth (Table 2) of all paths from the HMIs were measured in milliseconds (ms) and megabits/second (Mbps) and compared with their respective requirements.

Table 1. Latency between utilities.

| Required Latency: 100ms-2 sec | |
|-------------------------------|--|
| Between Utilities A&B | RTT min/avg/max (ms): 0.520/1.741/2.207 |
| Between Utilities A&C | RTT min/avg/max (ms): 1.276/1.887/2.193 |
| Between Utilities B&C | RTT min/avg/max (ms): 1.961/2.161/2.241 |

Table 2. Bandwidth between utilities.

| Required Bandwidth: About 1Gbps | |
|---------------------------------|----------|
| Between Utilities A&B | 931 Mbps |
| Between Utilities A&C | 931 Mbps |
| Between Utilities B&C | 931 Mbps |

With these measurements, the network effectively meets the communication requirements in the smart grid, and the applications operate at the speed needed. This fact proves that the architecture resembles a subset of the power grid infrastructure and can be used to analyze attacks and further develop the testbed by expanding utilities with more devices.

This research made progress on designing a testbed network architecture that resembles a subset of the power grid to test and analyze cyber events and develop security measures against them. The following accomplishments enabled this progress: (1) the implementation of different subnets; (2) routing configurations for efficient communication between utilities; (3) separating the utilities into different subnets that more accurately represent the power grid by simulating their distribution across different geographical locations.

The communication was tested through various methods and proved that the architecture was implementable in the industry. Moving forward, these methods will provide the ability to expand utilities with more hardware for data aggregation and transmission and to test the implementation of new technology such as DLT into the power grid.

5. AUTOMATION FRAMEWORK FOR EXPERIMENTS

To perform experiments more efficiently and in larger quantities, the ORNL research team for the DarkNet DLT Project developed an automation framework for the experiments. The framework exists as software on a computer node connected to the operational technology or supervisory control and data acquisition (SCADA) network of the testbed.

An automation framework is a useful tool for researchers to thoroughly test environments to ensure that they can withstand the stress of a variety of cyber-events. This tool is also useful for running many experiments with different variations to produce datasets for ML. In these environments, such as the hardware-in-the-loop testbed at the Advanced Protection Lab [11] shown in Figure 7, experiments involving cyber-events are conducted, and data are collected.

The framework involves a Python program with several stages, shown in Figure . To test the reliability of the framework, the cyber-events shown in Table 3 were launched against the testbed through the framework. To be a valid framework, the program must be able to launch successful scenarios at specific times during the experiments.

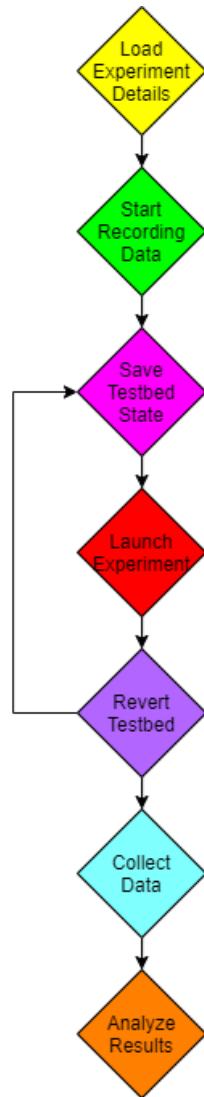


Figure 11. Multiple stages for launching experiments.

Table 3. Summary of launched cyberattacks.

| Cyber Event | Simulates |
|---|---------------------------|
| Man in the middle/address resolution protocol (MITM/ARP) spoofing | Loss of control |
| Flip breaker | Cyber-physical event |
| Alternate breaker state | System disruption |
| Denial of service (DoS) | Targeted resource offline |
| Distributed denial of service (DDoS) | Complete network outage |

In the experiments, the agent infiltrates a device on the power grid and launches the cyber-events, as shown in Figure .

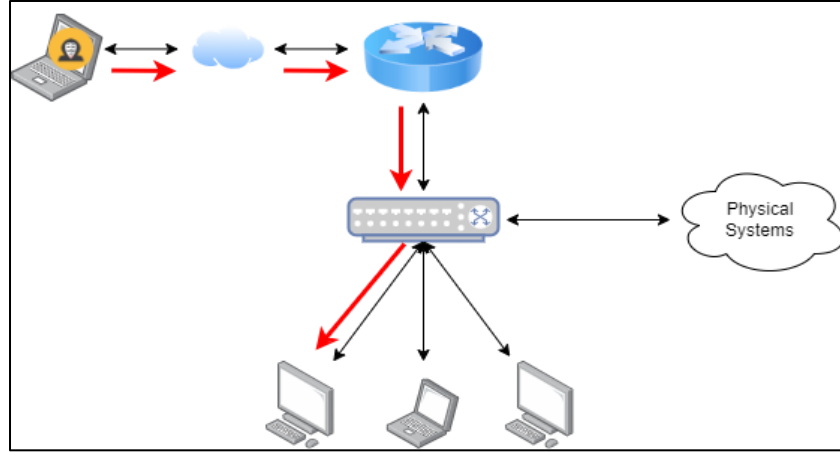


Figure 12. Diagram showing activity of the agent.

The framework successfully loaded code for each of the scenarios. Before each scenario, the current configuration of the testbed was saved. The scenarios were launched sequentially, each achieving its objectives and terminating at the desired time. After each scenario was completed, any changes made were reverted to the previous state. After the scenarios concluded, the testbed's data were downloaded for analysis. The timings of each phase were recorded for future analysis. As all phases were completed, the framework is tested to be valid for use in cyber experiments. Researchers formulated a data collection and processing workflow to enable the study of appropriate machine-learning model implementations for anomaly detection. Experiments are conducted and data are collected on a hardware-in-the-loop electrical substation grid testbed environment, using software such as MATLAB Simulink to simulate power system events with scenarios nearly identical to real-life operations. The following preconfigured cyber events and data collection methods are automated and performed on the testbed:

- Raw data collected are preprocessed through automated Python scripting.
- Important and relevant features are extracted for model training.
- Various detection models are studied to determine feasibility in the training agent.

The hypothesis behind the project was that creating a reliable and repeatable framework that could launch multiple scenarios is possible. The intended use of this framework is for researchers to have a method of implementing scenarios to cause cyber events. By causing multiple cyber events, systems can be tested for their reliability and hardness. After extensive efforts, this project validated the theory and resulted in a framework with working Python code to launch multiple scenarios at a testbed.

The goal of the framework was to be a versatile and reusable framework run on Linux to simulate cyber events on the testbed. To accomplish this, it had to be divided into several stages, so each stage could be modified independently and so the steps would not interfere with each other. The seven stages in the process are the following: (1) load experiment details, (2) start recording data, (3), save testbed state, (4) launch experiment, (5) revert testbed state, (6) collect data, and (7) analyze results. To make the framework implement several scenarios in a simulation, steps 3–5 can be repeated.

First, the overall framework code runs through the stages accomplishing the scenarios. In step 1, the details of the experiments are loaded from the configuration file, the arguments are parsed to determine the actions to perform, and the modules are loaded while ensuring they have the required functions. The second step is where the program connects to the data-generating devices and starts recording data before the scenarios begin. For the third step, the state of the testbed is saved so it can be reverted to after the scenario is completed. With the fourth step, the desired scenario is launched. Subsequently, for the current

scenario in the experiment, any changes made to the testbed are reverted in step 5. This allows for multiple scenarios to be performed in a simulation because steps 3–5 are repeated for each of the scenarios listed. With step 6, data are collected from the generating devices during the simulation, and in step 7, the collected data are analyzed in the final stage.

The data collection code, organized into the “ssh_into_device” Python module, is designed to securely access the DLT devices and download corresponding files. To securely access the target device, the DLT is accessed via the SSH protocol through the Python Paramiko library. Files are subsequently downloaded via the Python scp library. The integrity of the downloaded files is ensured by calculating the SHA256 hash of each file on the target machine and on the agent’s machine.

The scenario configuration file is where the specifics for each scenario are listed. When each scenario is run, the scenario is sent its corresponding data from the configuration file. By sending only the provided data, unnecessary data transmission is reduced, and only one parameter is required, thus reducing the complexity of the code and making it more modular. In the configuration files, each scenario is numbered in the format “attackx,” where *x* is the scenario number. To help the program analyze the configuration file, the maximum attack number needs to be included in the simulation file. For each scenario, the type of scenario is also required in the “scenarioType” key. Most importantly, the duration of the scenario needs to be stated with the key “simulationRunTime.” For user customization, two optional parameters can be provided. Other than these previously stated parameters, any other parameters in the configuration file will be the arguments required for the scenario to function. For simplicity, all the configuration data for the current scenario, formatted as json, will be passed to the scenario’s “main()” function.

Each scenario code file must have certain functions; otherwise, the program will not execute. The three required functions are “main(),” “reset(),” and “saveSettings()” Other functions can be added to each scenario file. The format of each is the same: return the value “true” if the corresponding function completed successfully, and return “false” otherwise. For example, if the simulation to launch takes less time than allotted in the configuration .json file, the main() function will return true.

One function that program must have is a saveSettings() function. As its name indicates, this function saves any settings that could be modified during the scenario. Specifically, any settings that need to be reverted after the scenario is completed should be saved in this phase. For example, in an address resolution protocol (ARP) confusion attack, where multiple devices on a network are sent fraudulent ARP messages to cause devices to send their messages to unintended devices, each device’s IP and MAC addresses would have to be saved by the agent to restore them after the attack concludes.

Logically, the function where the scenario occurs is the main() function. This function has only one argument sent to it, which is the data from the scenario’s portion of the .json configuration file. Any scenario-specific details are to be listed in the configuration file. This scenario will receive a SIGINT, also known as a keyboard interrupt, signal when the time allotted in the configuration file has expired. In all scenarios, this signal should be caught in a try-except statement block, so the program knows when to terminate and continue with the rest of the scenarios. If this exception is caught, the program should return “true.” However, if any other exception is caught when running the scenario, the function should return “false.”

Finally, the reset() function is responsible for reverting any settings that were changed during the scenario. For example, in the ARP confusion attack, devices have incorrect associations between IP and MAC addresses. If a simulation like this has the potential to disrupt communications from the original state, the original network state needs to be restored. In this scenario, legitimate ARP messages would be sent to all devices indicating all other IP addresses and their associated MAC addresses.

To research a subject and contribute to its field of study, examining research done previously is important. In a dissertation, Pan describes an approach that involves a hierarchy of scripts to run experiments of various scenarios on a power system testbed [25]. The main script is an AutoIT scripts that can run other scripts for the various scenarios. The scenarios include faults and cyber-attacks.

In [26], a system for the automated cybersecurity testing of automobiles was produced, and its merits were tested. Instead of applying cybersecurity principles to defend automobile systems, this framework's intent is to increase the cyber defense of systems by thoroughly applying the concepts of endurance testing to a sample power grid.

In another article [27], researchers discuss a framework for examining relationships between control and power systems in power systems automation. Their HARMer system is designed to intelligently pen test systems and determine their robustness. These researchers used an algorithm that examines the vulnerabilities of a system and exploits them. The project developed a general framework, where users decide the attacks to launch, and those attacks are iterated through. Instead of using an algorithm to automate examining a system for vulnerabilities to exploit them, this framework is designed for researchers to use select scenarios to generate certain cyber events and resulting data.

Other researchers designed an intrusion prevention system that examines command packets in network traffic and tries to predict if these packets can be harmful to the system [28]. Instead of examining packets and predicting their impacts, this framework seeks to test systems by launching repeated scenarios to cause cyber events.

In a different study [29] researchers investigated automated tools to augment cyber ranges automation. These researchers investigated CRATE for its ability to automate a cyber range for experimentation, training, and exercises. Whereas their analysis was focused on a system's automation and its capabilities, rather than on how a system can be tested, the framework designed by this project is interested in only automating the testing of systems with scenarios the researcher wants to implement.

In the research done previously, attempts have been made to increase the cyber defense of systems, either by designing new methods of intrusion protection or by implementing automated exploitation of a system. Although these projects are useful, a framework is needed for researchers to test the defensive capabilities of a system by going on the offensive and launching multiple scenarios against the target system, causing cyber events. This research provides this framework as a novel proof-of-concept proving the framework launches multiple scenarios and generates the intended cyber events.

An experiment is composed of several computers, the overall framework's code, the data collection code, the scenario configuration file (which contains data for each scenario), and the scenario code file(s). Each experiment is designed with a specific goal, such as to test the endurance of the power grid testbed or to test the CGG system. A product of all scenarios is the creation of data that can be used to train ML models to process.

6. PREPARING DATASETS

Once the automation framework has produced and collected raw data from the experiments, further steps must be taken to curate the data for other purposes such as ML.

The electric grid heavily relies on consistent operation and stability to maintain generation, transmission, and distribution of electricity to consumers; however, threat actors and natural faults hinder these operations. The goal is to study and develop a process workflow for leveraging ML techniques in detecting electrical faults versus cyber intrusion events, so operators understand incidents posed to the

power system and determine the best response. They can also formulate a data collection and processing workflow to enable the study of appropriate ML model implementations for anomaly detection.

Cyber-physical system security research of industrial control systems, specifically power systems, has been an ongoing and lucrative field of study for researchers around the world. The primary cause for concern among government officials and regulators is the possibility for a highly coordinated attack on critical components within the power system infrastructure, causing a citywide blackout. This type of large-scale attack could have everlasting effects on communities within the blackout. Prominent emergency services on which citizens rely (e.g., police, fire, medical, and rescue) would be unreachable and have difficulty keeping up with the influx of calls. City roadways would be at a standstill with traffic signals (e.g., traffic lights) out of service. Critical services (e.g., hospitals and wastewater treatment) would halt operations, significantly impacting human life. In short, the effects would be devastating.

Modern structuring of power systems from the grid follows the baseline process established long ago: generation, transmission, and distribution. *Generation* (e.g., wind, solar, coal, nuclear) is the low-level process of converting stored energy within a fuel resource to electrical energy. *Transmission* involves the transport of this newly formed energy across the wire to substations within a municipality for the eventual *distribution* of energy to facilities, homes, and stations around the city. Future implementations of power grid systems involve the smart grid in which the grid can adapt its behavior and energy requirements based on load necessity in certain sectors of a city. A valuable component for smart grids is microgrids, which can meet electricity demands without the traditional connection to the macrogrid. Microgrids can function and maintain operations autonomously with the integration of DERs (e.g., solar panels, wind farms, and energy storage systems).

As history has shown, security and encryption mechanisms for industrial control systems are not always top priorities for engineers. Although this notion has changed in recent times, demonstrations of commonly used attack vectors and methods are seen in mostly modern transmission protocols (e.g., IEC 61850). Differing attack types on the IEC 61850 protocol [30] range from security attacks on the IEC 61850 network to the IEC 61850 messaging system (GOOSE and SV). For network attacks, Ashraf et al. reference a malformed packet attack, denial of service, ARP spoofing attack, man in the middle attack, and configuration tampering. Furthermore, on the exploitation of GOOSE/SV messaging, the authors reference GOOSE and SV modification, denial of service, and replay attacks. Many of these attack types involve disrupting communication between network devices, maliciously manipulating data, or causing failures/trips in the system's functionality. These can have severe consequences in operation and reliability of a substation network with the possibility of outages occurring for customers served by that substation.

This research primarily focuses on the security components within these substations. IEDs can unintentionally derive new attack vectors when introduced with additional network connectivity functions. In combination with electrical faults, the detection and classification of cyberattack versus electrical fault can be a challenging distinction. Researchers in [31] discuss the proposal of a data-driven scheme to define normal operations, three-phase faults, and stealth cyberattacks that target IEC 61850 in digital substations. The cyberattacks demonstrated in this paper focuses on bay and station level operations. The authors conclude their findings by reiterating the real-time distinction between GOOSE cyberattacks and three-phase electrical faults in digital substations with the use of a spatial autoregressive moving average (SARMA) model and phasor measurement units readings alongside the GOOSE traffic. This architecture was shown to be able to distinguish between cyberattacks and electrical faults within a 200 ms time frame. The model would in turn make an appropriate decision regarding the cause of a circuit breaker opening in a substation.

Authors in [32] propose a centralized and distributed cyberattack and fault detection and isolation (CAFDI) methodology approach where two filters are placed on the plant and on the command and control (C2) portion of the cyber-physical systems (CPS), respectively. In addition, an unknown input observer-based detection mechanism is placed on the plant side. The proposal characterizes several types of cyberattacks the methodology can detect, including covert, zero-dynamics, and replay attacks. The researchers conclude their findings by reiterating the proposed methodology can detect and isolate cyberattacks and faults, alongside anomalies in nearby subsystems. An individual can detect machine-induced actuator and sensor faults alongside undetectable attacks and isolate those attacks, faults, and anomalies while isolating such occurrences.

In complex topological environments, locations of electrical faults within the distribution network can be a difficult problem owing to the extending branches and varieties of energy sources. Fault localization, isolation, and power restoration may be primary concerns and goals for service providers when integrating systems to a larger infrastructure. Researchers in [33] propose an adaptive multiagent system for improving reliability, speed, and protection of the power system. This methodology can prevent and mitigate electrical faults and cyberattacks within a distribution system. The authors conclude their findings by reiterating the purpose of the proposed technique, which is to maintain the advantages of hierarchical diagnosis to detect cyberattacks within the power grid and take measures to mitigate effects of a circuit breaker fault following a cyberattack. Using this technique decreased the time required to clear a fault and prevented cascading breaker failures.

In [34], Adhikari provides preliminary background information and understanding of cyber-physical power system security alongside a research and analysis of real-time cyber-power event classification. Adhikari uses the non-nested generalized exemplars classification (NNGE) algorithm and Hoeffding adaptive tree (HAT) classifier algorithm for event classification of cyber-power systems. HAT performed well for binary and multiclass classification with overall low system resource utilization and high accuracy. NNGE, with State Tracking and Extraction Method (STEM) preprocessing, provides satisfactory classification accuracy in the mid to high 90th percentile. Like the HAT algorithm, NNGE has reduced memory utilization costs and classifies events in rapid succession.

Experiments are conducted and data are collected on a hardware-in-the-loop electrical substation grid testbed environment, using software such as MATLAB Simulink to simulate power system events with scenarios nearly identical to real-life operations.

Preconfigured cyberattacks and data collection methods are automated and performed on the testbed, as follows:

- Collected raw data are preprocessed through automated Python scripting.
- Important and relevant features are extracted for model training.
- Various detection models are studied to determine feasibility in the training agent.

The trained anomaly detection agent will be deployed on substation networks to monitor for suspicious activity and classify them as electrical faults or cyberattacks.

The developed workflow (Figure) process initiates the required steps to formulate and design ML models of varying types for anomaly detection on the electrical grid testbed. Initial goals in design are to differentiate between cyberattacks and electrical faults.

Establishing a data collection and processing workflow is a vital aspect in the development of any ML model. A detection system is sustainable and reliable only so long as the data provided meet the standards and requirements to train the model.

Following are key takeaways and contributions for this project:

- Develop document for simulation procedures.
- Highlight key areas for improvement.
- Develop a data collection and processing workflow.
- Research and understand ML use cases in power system security.
- Review appropriate model requirements for model development.

The expectation for this project involves two stages. First is the development of a sound, dependable workflow for data collection, analysis, and preprocessing. Ensuring the proper collection of raw data into a polished dataset for training, validation, and testing purposes enables the implementation of stage two. In this stage, researchers will begin model selection and development of a model for detection purposes on the substation network. The model development stage will involve model training, validation, and hyperparameter tuning. Once the model has been tuned, researchers can measure the performance on the testbed using common metrics evaluators and revise model components if necessary. The overall project aims to develop the process workflow for data collection and cleaning and then develop and test an ML detection model.

Future work involves the development and performance evaluation of all research conducted, allocation of appropriate resources for training, and deployment of detection systems as a feature for the electrical substation grid testbed.

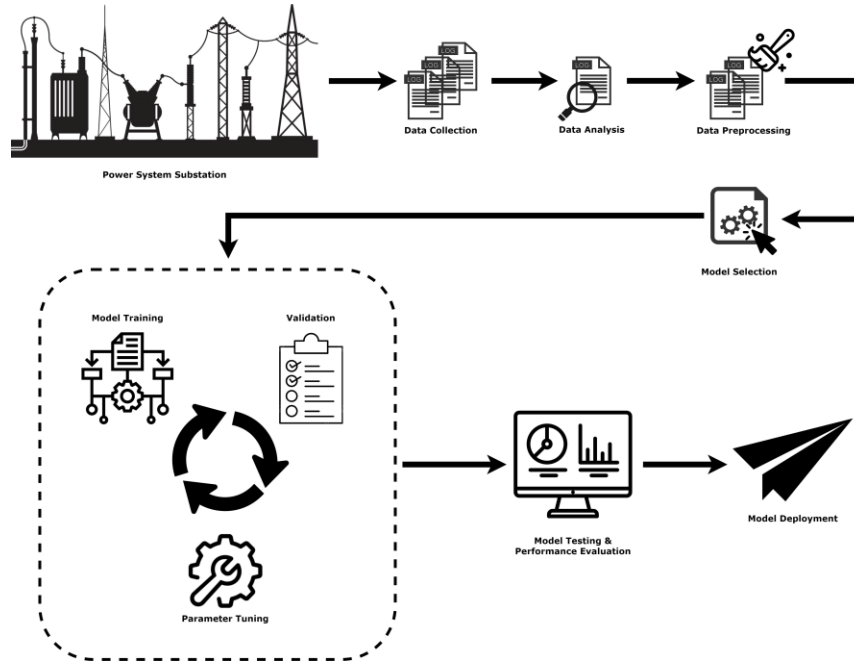


Figure 13. The developed workflow process.

7. CONCLUSIONS AND FUTURE WORKS

For future work, we will continue to develop our testbed, automation framework for experiments, and our methods for producing datasets. Specifically, for the testbed, we will add more control centers that can communicate in the Inter-Control Center Protocol (ICCP). This work can be expanded to include a wider

variety of scenarios. More work could also be performed to integrate common penetration testing tools, such as Metasploit, Nmap, and Nessus, with this framework so industry standard tools could be offered as a part of sample scenario set. While this tool was designed for a hardware-in-the-loop testbed at Oak Ridge National Laboratory, it could be expanded to test other offline systems for their robustness and reliability. The potential expansions for this tool are limited only by the imagination of those who implement it to automate research.

8. ACKNOWLEDGMENTS

This research is supported by the US Department of Energy (DOE), Office of Electricity. This manuscript has been authored by UT Battelle, LLC, under Contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<https://energy.gov/downloads/doe-public-access-plan>).

This research was supported in part by an appointment to the U.S. Department of Energy's Omni Technology Alliance Program, sponsored by DOE and administered by the Oak Ridge Institute for Science and Education.

9. REFERENCES

- [1] Oak Ridge Educational Program web link. <https://education.ornl.gov/tpi/> [accessed 16 August 2024].
- [2] United State Department of Energy. Workforce Training for the Electric Power Sector: Transforming the Nation's Electric Grid by Training Skilled Workers, November 2011, pp. 1-4. https://www.energy.gov/sites/prod/files/2016/12/f34/Workforce_Training_Case_Study_9-29-11_0.pdf. [accessed 16 August 2024].
- [3] IEEE PES. Preparing the U.S. Foundation for Future Electric Energy Systems: A Strong Power and Energy Engineering Workforce, April 2009. <https://www.cewd.org/documents/USPowerEnergy.pdf>. [accessed 16 August 2024].
- [4] Piesciorovsky EC and Schulz NN, Burns & McDonnell — K-State Smart Grid Laboratory: Protection, communication & power metering, 2014 IEEE PES General Meeting | Conference & Exposition, National Harbor, MD, USA, 2014, pp. 1-5.
- [5] Kezunovic M. Teaching the Smart Grid Fundamentals using Modeling, Simulation, and Hands-on Laboratory Experiments, Power and Energy Society General Meeting, IEEE-2010, pp. 1-6.
- [6] Schulz NN, Integrating smart grid technologies into an electrical and computer engineering curriculum, 2011 IEEE PES Innovative Smart Grid Technologies, Perth, WA, Australia, 2011, pp. 1-4.
- [7] Lu G, De D, and Song W-Z, SmartGridLab: A Laboratory-Based Smart Grid Testbed, 2010 First IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, USA, 2010, pp. 143-148
- [8] Wollenberg B and Mohan N. The Importance of Modern Teaching Labs, IEEE Power and Energy Magazine, vol. 8, no. 4, pp. 44-52, July-Aug. 2010.

- [9] Adams NE. Bloom's taxonomy of cognitive learning objectives. *Journal of the Medical Library Association: JMLA*. 2015 Jul;103(3):152-153. <https://doi.org/10.3163/1536-5050.103.3.010> [accessed 16 August 2024].
- [10] Edvard C. Electrical Engineering Portal. Six common bus configurations in substations up to 354 kV, <https://electrical-engineering-portal.com/bus-configurations-substations-345-kv>, March 18, 2019 [accessed 16 August 2024].
- [11] Piesciorovsky EC, Borges Hink R, Werth A, Hahn G, Lee A, Richards J, Polsky Y. Assessment of the Electrical Substation-Grid Test Bed with Inside/Outside Devices and Distributed Ledger. ORNL/TM-2022/1840. Oak Ridge, Tennessee: Oak Ridge National Laboratory; 2022, p. 1–87.
- [12] Schweitzer Engineering Laboratories Inc. SEL 735 Power Quality and Revenue Meter Instruction Manual, <https://selinc.com/products/735/docs/> [accessed 16 August 2024].
- [13] Schweitzer Engineering Laboratories Inc. SEL 734 Advanced Metering System Instruction Manual, <https://selinc.com/products/734/docs/> [accessed 16 August 2024].
- [14] Schweitzer Engineering Laboratories Inc. SEL 421-4, -5, Protection, Automation, and Control System Instruction Manual, <https://selinc.com/products/421/docs/> [accessed 16 August 2024].
- [15] Schweitzer Engineering Laboratories Inc. SEL 451-5 Protection, Automation, and Bay Control System and SEL 400 Series Relays Instruction Manual, <https://selinc.com/products/451/docs/> [accessed 16 August 2024].
- [16] Schweitzer Engineering Laboratories Inc. SEL 351S Protection System Instruction Manual, <https://selinc.com/products/351S/docs/> [accessed 16 August 2024].
- [17] Schweitzer Engineering Laboratories Inc. SEL-700G Generator and Intertie Protection Relays Instruction Manual, <https://selinc.com/products/700G/docs/> [accessed 16 August 2024].
- [18] Piesciorovsky, E.C.; Stenvig, N.; Gui, Y.; Olama, M.M.; Bhusal, N.; Yadav, A. Advanced testbed to assess disturbances in electrical grids with DERs using relays/meters with varying sampling frequencies. *Energy Reports* 2024, 11, 6032–6047.
- [19] Piesciorovsky, E.C.; Hahn, G.; Hink, R.B.; Werth, A.; Lee, A. Electrical substation grid testbed for DLT applications of electrical fault detection, power quality monitoring, DERs use cases and cyber-events. *Energy Reports* 2023, 10, 1099–1115.
- [20] Piesciorovsky, E.C.; Borges Hink, R.; Werth, A.; Hahn, G.; Lee, A.; Polsky, Y. Assessment and Commissioning of Electrical Substation Grid Testbed with a Real-Time Simulator and Protective Relays/Power Meters in the Loop. *Energies* 2023, 16, 4407.
- [21] Hahn G, Piesciorovsky EC, Borges Hink R, Werth A. Detection of Faulted Phases in a Medium-Voltage Main Feeder Using the Cyber Grid Guard System with Distributed Ledge Technology. *International Journal of Electrical Power and Energy Systems*, ELSEVIER, vol. 161 (110162), pp. 1-15, October 2024.
- [22] S&C Electric Company. Total Clearing Time-Current Characteristic Curves, Positrol® Fuse Links–S&C “T” Speed (TCC 170-6-2), <https://www.sandc.com/en/products--services/products/positrol-fuse-links/#Construction> [accessed 16 August 2024].
- [23] O’Toole, Zachary, et al. A Cyber-Physical Testbed Design for the Electric Power Grid | IEEE Conference Publication | IEEE Xplore, IEEE, 17 Feb. 2020, ieeexplore.ieee.org/document/9000312.
- [24] Department of Energy. Communications Requirements of Smart Grid Technologies, 5 Oct. 2010, www.energy.gov/gc/articles/communications-requirements-smart-grid-technologies.

- [25] Pan, Shengyi. Cybersecurity testing and intrusion detection for cyber-physical power systems. Mississippi State University, 2014.
- [26] Marksteiner, Stefan, et al. "A process to facilitate automated automotive cybersecurity testing." 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring). IEEE, 2021.
- [27] Enoch, Simon Yusuf, et al. "HARMer: Cyber-attacks automation and evaluation." IEEE Access 8 (2020): 129397-129414.
- [28] Werth, Aaron W., and Thomas H. Morris. "Intrusion prevention for payloads against cyber-physical systems by predicting potential impacts." Journal of Cyber Security Technology 6.3 (2022): 113-148.
- [29] Gustafsson, Tommy, and Jonas Almroth. "Cyber range automation overview with a case study of CRATE." Nordic Conference on Secure IT Systems. Cham: Springer International Publishing, 2020.
- [30] Ashraf S, Shawon MH, Khalid HM, Muyeen SM. Denial-of-Service Attack on IEC 61850-Based Substation Automation System: A Crucial Cyber Threat towards Smart Substation Pathways. Sensors. 2021; 21(19):6415. <https://doi.org/10.3390/s21196415>
- [31] A. Abedi, V. S. Rajkumar, A. Ştefanov and P. Palensky, "Towards Real-Time Distinction of Power System Faults and Cyber Attacks," 2023 IEEE Power & Energy Society General Meeting (PESGM), Orlando, FL, USA, 2023, pp. 1-5, doi: 10.1109/PESGM52003.2023.10253241.
- [32] M. Taheri, K. Khorasani, I. Shames and N. Meskin, "Cyberattack and Machine-Induced Fault Detection and Isolation Methodologies for Cyber-Physical Systems," in IEEE Transactions on Control Systems Technology, vol. 32, no. 2, pp. 502-517, March 2024, doi: 10.1109/TCST.2023.3324870.
- [33] Albarakati, A.J.; Azeroual, M.; Boujoudar, Y.; EL Iysaouy, L.; Aljarbough, A.; Tassaddiq, A.; EL Markhi, H. Multi-Agent-Based Fault Location and Cyber-Attack Detection in Distribution System. Energies 2023, 16, 224. <https://doi.org/10.3390/en16010224>
- [34] Adhikari, Uttam, "Event and Intrusion Detection Systems for Cyber-Physical Power Systems" (2015). Theses and Dissertations. 2089. <https://scholarsjunction.msstate.edu/td/2089>

APPENDIX A. INSTRUCTIONS FOR NETWORKING

To get latency/bandwidth measurements of the network between the different subnets and different devices, you will need to follow these instructions:

1. Make sure that all the switches are connected to the router on the right interfaces via Ethernet cables as seen in Figure A-1. The interfaces configured to be used right now are the main switch on interface 2, Utility B switch on interface 4, and Utility C switch on interface 6. **IF CHANGED, MAKE SURE TO CHANGE CONFIGURATION ON ROUTER TO REFLECT NEW LAYOUT.**

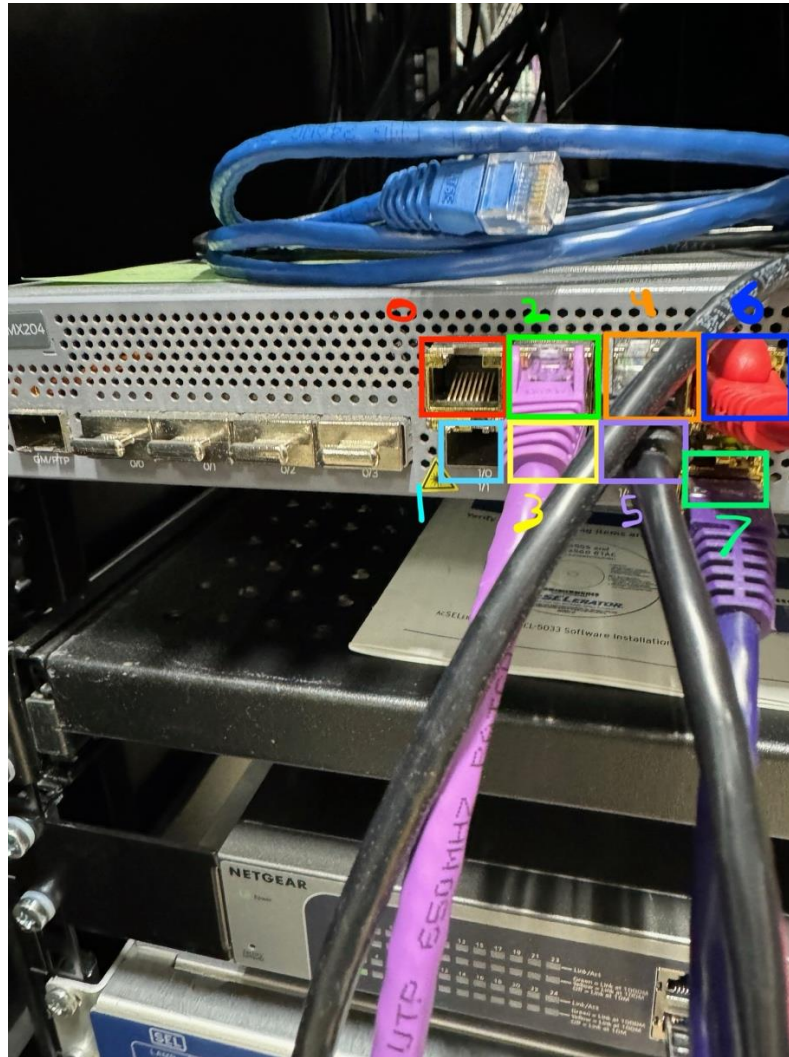
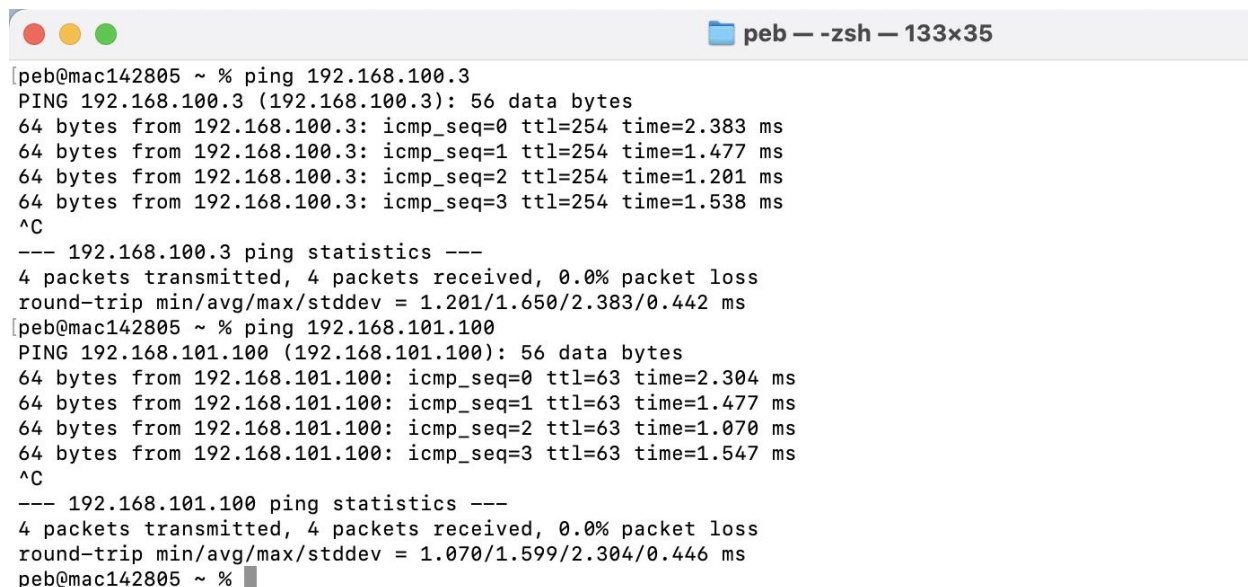


Figure A-1. Router with various ports

2. Note which switches correspond to which subnet:
 - a. Main switch – **192.168.100.X**
 - b. Utility B switch (Netgear 24 port switch) – **192.168.101.X**
 - c. Utility C switch (Netgear 8 port switch) – **192.168.102.X**

3. For the device to be connected to the switch and on the corresponding subnet, make sure it is connected to the switch via Ethernet cable and the port on the switch is blinking green. Also, make sure the laptop is assigned to the correct IP, subnet mask (**255.255.255.0**), and default gateway (**192.168.X.1**).
 - a. To make sure everything is connected successfully, you can try to access the corresponding switch's web interface:

Main switch IP – **192.168.0.239**
Utility B switch IP – **192.168.101.15**
Utility C switch IP – **192.168.102.11**
4. To first test if there is communication between devices or subnets, you can run a ping command from each device and see if you get responses.
5. If you get responses from the ping commands, you can test the network bandwidth and latency several ways:
 - a. Ping commands – checking the round-trip time and dividing it in half to see the one-way trip time (Figures A-2, A-3). The round-trip time is going to be the measurement that is all the way to the right and is usually measured in milliseconds (ms). This is only for ICMP packets which have a low priority, so it is best to get these measurements with TCP packets as well.



```
peb@mac142805 ~ % ping 192.168.100.3
PING 192.168.100.3 (192.168.100.3): 56 data bytes
64 bytes from 192.168.100.3: icmp_seq=0 ttl=254 time=2.383 ms
64 bytes from 192.168.100.3: icmp_seq=1 ttl=254 time=1.477 ms
64 bytes from 192.168.100.3: icmp_seq=2 ttl=254 time=1.201 ms
64 bytes from 192.168.100.3: icmp_seq=3 ttl=254 time=1.538 ms
^C
--- 192.168.100.3 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.201/1.650/2.383/0.442 ms
peb@mac142805 ~ % ping 192.168.101.100
PING 192.168.101.100 (192.168.101.100): 56 data bytes
64 bytes from 192.168.101.100: icmp_seq=0 ttl=63 time=2.304 ms
64 bytes from 192.168.101.100: icmp_seq=1 ttl=63 time=1.477 ms
64 bytes from 192.168.101.100: icmp_seq=2 ttl=63 time=1.070 ms
64 bytes from 192.168.101.100: icmp_seq=3 ttl=63 time=1.547 ms
^C
--- 192.168.101.100 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.070/1.599/2.304/0.446 ms
peb@mac142805 ~ %
```

Figure A-2. From the Mac connected to the Utility C switch (192.168.102.0/24).

```

localadmin@LAP0135666: ~
localadmin@LAP0135666:~$ ping 192.168.100.3
PING 192.168.100.3 (192.168.100.3) 56(84) bytes of data.
64 bytes from 192.168.100.3: icmp_seq=1 ttl=254 time=26.6 ms
64 bytes from 192.168.100.3: icmp_seq=2 ttl=254 time=0.849 ms
64 bytes from 192.168.100.3: icmp_seq=3 ttl=254 time=2.62 ms
64 bytes from 192.168.100.3: icmp_seq=4 ttl=254 time=1.35 ms
^C
--- 192.168.100.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 0.849/7.865/26.638/10.857 ms
localadmin@LAP0135666:~$ ping 192.168.102.100
PING 192.168.102.100 (192.168.102.100) 56(84) bytes of data.
64 bytes from 192.168.102.100: icmp_seq=1 ttl=63 time=0.986 ms
64 bytes from 192.168.102.100: icmp_seq=2 ttl=63 time=0.999 ms
64 bytes from 192.168.102.100: icmp_seq=3 ttl=63 time=1.21 ms
64 bytes from 192.168.102.100: icmp_seq=4 ttl=63 time=1.33 ms
^C
--- 192.168.102.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.986/1.131/1.327/0.144 ms
localadmin@LAP0135666:~$

```

Figure A-3. From the Ubuntu Laptop connected to the Utility B switch (192.168.101.0/24).

- b. iPerf commands – you can run multiple iPerf commands to measure bandwidth and latency with two laptops as long as you have the iPerf packages installed on both (Figures A-4, A-5, A-6).
 - i. On server side – **iperf -s**

```

peb — iperf -s — 133x35
peb@mac142805 ~ % iperf -s
-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----

```

Figure A-4. With the Mac acting as the iPerf server.

```

peb — iperf -s — 133x35
peb@mac142805 ~ % iperf -s
-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 1] local 192.168.102.100 port 5001 connected with 192.168.101.100 port 59088
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.00-10.01 sec  1.01 GBytes  864 Mbits/sec

```

Figure A-5. Output on the server when receiving a connection from the other device.

```
localadmin@LAP0135666: ~  
localadmin@LAP0135666:~$ iperf -s  
-----  
Server listening on TCP port 5001  
TCP window size: 128 KByte (default)  
-----  
[ 1] local 192.168.101.100 port 5001 connected with 192.168.102.100 port 61677  
[ ID] Interval      Transfer      Bandwidth  
[ 1] 0.0000-10.0341 sec  1.10 GBytes   941 Mbits/sec  
[ 2] local 192.168.101.100 port 5001 connected with 192.168.102.100 port 61704  
[ ID] Interval      Transfer      Bandwidth  
[ 2] 0.0000-30.0283 sec  3.29 GBytes   941 Mbits/sec  
[ 3] local 192.168.101.100 port 5001 connected with 192.168.102.100 port 61710  
[ ID] Interval      Transfer      Bandwidth  
[ 3] 0.0000-120.0296 sec 13.2 GBytes   941 Mbits/sec
```

Figure A-6. Output with the Ubuntu Laptop acting as the server with multiple connections from the other device.

- ii. Client side – **iperf -c *server_ip_address*** (replacing *server_ip_address* with the actual IP address of the device acting as the server)

This is the default command to run the network test. It measures how much data can be transferred in 10 seconds as well as the bandwidth (Figures A-7, A-8, A-9).

1. You can modify this command by using **-t** and **-n**.

Using **-t** allows you to specify the amount of time, in seconds, you want to test the network.

Using **-n** allows you to specify a fixed amount of data you would like to send between the client and server while the network is being tested. You can specify the units as well with adding K, M, or G at the end to represent kilobytes, megabytes, and gigabytes.

An example of using these additions would look something like this: **iperf -c *server_ip_address* -t 30** or **iperf -c *server_ip_address* -n 100M**

```
peb@mac142805 ~ % iperf -c 192.168.101.100
-----
Client connecting to 192.168.101.100, TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 1] local 192.168.102.100 port 61677 connected with 192.168.101.100 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.00-10.04 sec  1.10 GBytes  941 Mbits/sec
peb@mac142805 ~ % iperf -c 192.168.101.100 -t 30
-----
Client connecting to 192.168.101.100, TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 1] local 192.168.102.100 port 61704 connected with 192.168.101.100 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.00-30.04 sec  3.29 GBytes  941 Mbits/sec
peb@mac142805 ~ % iperf -c 192.168.101.100 -t 120
-----
Client connecting to 192.168.101.100, TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 1] local 192.168.102.100 port 61710 connected with 192.168.101.100 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.00-120.03 sec 13.2 GBytes  941 Mbits/sec
peb@mac142805 ~ % █
```

Figure A-7. Output when the Mac is the client and using the default client command and the command with -t for different times.

```
peb@mac142805 ~ % iperf -c 192.168.101.100 -n 1G
-----
Client connecting to 192.168.101.100, TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 1] local 192.168.102.100 port 61765 connected with 192.168.101.100 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.00-9.14 sec  1.00 GBytes  940 Mbits/sec
peb@mac142805 ~ % iperf -c 192.168.101.100 -n 2G
-----
Client connecting to 192.168.101.100, TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 1] local 192.168.102.100 port 61769 connected with 192.168.101.100 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.00-18.28 sec  2.00 GBytes  940 Mbits/sec
peb@mac142805 ~ % iperf -c 192.168.101.100 -n 100M
-----
Client connecting to 192.168.101.100, TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 1] local 192.168.102.100 port 61797 connected with 192.168.101.100 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.00-0.91 sec   100 MBytes  927 Mbits/sec
peb@mac142805 ~ % █
```

Figure A-8. Output with the Mac as the client and using -n for different amounts of data.

```
localadmin@LAP0135666: ~  
localadmin@LAP0135666:~$ iperf -c 192.168.102.100 -t 30  
-----  
Client connecting to 192.168.102.100, TCP port 5001  
TCP window size: 85.0 KByte (default)  
-----  
[ 1] local 192.168.101.100 port 38896 connected with 192.168.102.100 port 5001  
[ ID] Interval      Transfer    Bandwidth  
[ 1] 0.0000-30.1596 sec 2.88 GBytes 819 Mbits/sec  
localadmin@LAP0135666:~$ iperf -c 192.168.102.100 -t 120  
-----  
Client connecting to 192.168.102.100, TCP port 5001  
TCP window size: 85.0 KByte (default)  
-----  
[ 1] local 192.168.101.100 port 52682 connected with 192.168.102.100 port 5001  
[ ID] Interval      Transfer    Bandwidth  
[ 1] 0.0000-120.0404 sec 11.4 GBytes 813 Mbits/sec  
localadmin@LAP0135666:~$
```

Figure A-9. Output with the Ubuntu Laptop as the client and using -t for different times.

- c. A simple client/server program – you can test the latency and bandwidth and actual data being sent over the network between two devices by a simple client and server program. This can be achieved by specifying a file to be a certain amount of data or just having the program generate that much data and then sending it to the other device with timestamps and the total number of bytes received. Utilize the timestamps for when the data is sent and when the data is received to gauge an accurate measurement of the latency.

Steps for the Juniper MX204 router

1. Connect the laptop to the router via ethernet cable and the MGMT port on the front panel of the router. The port is inside the red box in the figure below and should have the label “MGMT” below the port itself (Figure A-10).
2. Make sure the laptop is on the same subnet as the router before ssh'ing
 - Router's IP address is **192.168.1.100**
 - Set the laptop's IP address to **192.168.1.99**
 - Set the default gateway to **192.168.1.1**
 - To do this and set the laptop's IP address on a Windows laptop, you will have to go into “view network connections” through the start menu search bar, click on Ethernet 2, and then click Properties, then go to the IPv4 bar and open it.
 - Enter in the addresses of the laptop and default gateway above while leaving the subnet mask default (**255.255.255.0**)
 - To do this on an Ubuntu laptop, you can go to settings, then network, and configure the Wired Connection and make a new connection profile. If there is already an existing connection profile that is not configured the way you want it, you will have to first delete that connection profile.
 - To do this on a Mac, you go to settings, network, and see which Ethernet port is active. Then you click on that port and go to details and then TCP/IP to configure everything you need.

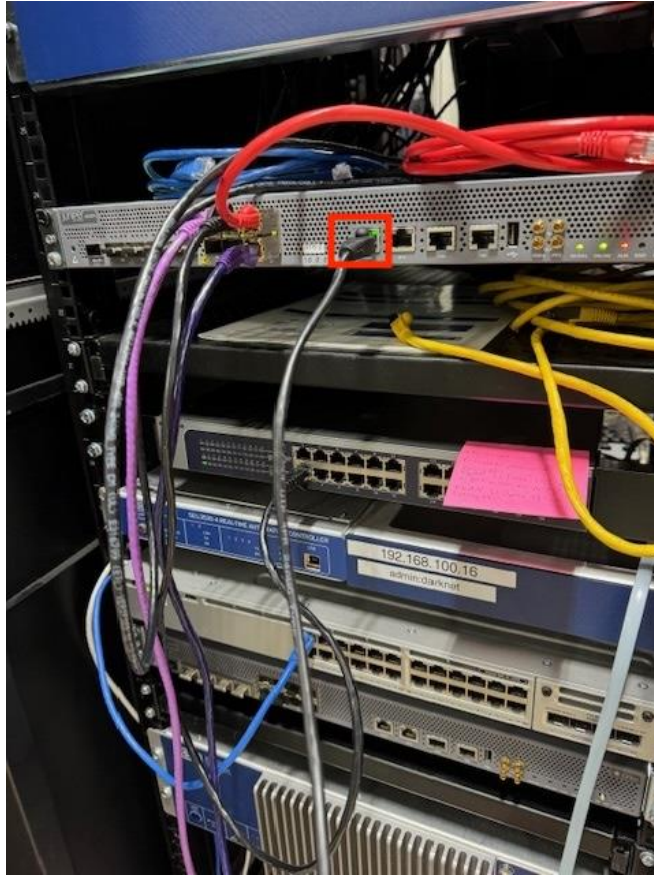


Figure A-10. Management Port on Router.

3. SSH into the router with the command **ssh root@192.168.1.100**
 - Enter yes if prompted about a fingerprint being permanently added
4. Once in the router, to access the current and past .gz config files, you need to go to the **/config** directory in the root director (root directory is /)
5. To enter the command line interface of the router, enter the command **cli**
6. To show the current configuration of the router, enter the command **show configuration** once in cli mode
7. To edit the configuration, you will need to enter the configuration mode by entering the command **configure** while in cli mode
8. To exit any mode, use the **exit** command
9. If you need more information about configuration of certain sections, resort to juniper's website or ChatGPT

Configuration Overview

Configuration Mode Overview:

- you can configure all Junos OS properties including interfaces, general routing information, routing protocols, and user access, and monitor device properties
- you can create the config interactively, or create an ASCII text file containing the config and load it on the device and then commit it
- the following link is for the list of configuration mode commands and their description
 - <https://www.juniper.net/documentation/us/en/software/junos/cli/topics/topic-map/cli-configuration.html>
- the 50 most recently committed config files on a device are saved so that you can return to a prev config. The names are:
 - juniper.conf.gz—The current active configuration
 - juniper.conf.1.gz to juniper.conf.49.gz—Rollback configurations
- must use config mode in the CLI to make changes to the files
- the candidate configuration enables you to make config changes without causing operational changes to the active configuration
 - when you commit the candidate config, the system updates the active config
- the current and three previous versions of the config are saved on the device CompactFlash card
 - currently operational device config is stored in juniper.conf.gz while the other 3 are stored as juniper.conf.1.gz, etc. These files are stored in the CompactFlash in the directory /config
 - the remaining 46 previous versions are stored in the directory /var/db/config on the hard disk
- you can compare the candidate configuration with the current committed configuration and display the differences between the two configs in XML with the command
 - show configuration | compare | display xml for operational mode
 - show | compare | display xml for config mode

APPENDIX B. AUTOMATION FRAMEWORK

To provide data to the program, certain flags are implemented when running the program through the command line. These flags tell the program how to operate and specific behaviors to use, such as using different files, or changing where files are loaded from. The flags and their purposes are shown below:

-a: Attack file location. This is the location where the simulation file is. If not specified, the default location of “./attacks/attack_data.json” will be used. If the .json file does not exist, the program will terminate.

-c: Collect data. Instead of running the full scenario, the data generated is collected from the generating devices. The code behind this was generated to collect “.pcaps” and artifacts from a DLT device. If this flag is specified, the only. **Currently, the other flags are still required when this flag is given, even if data collection is the only action.**

-f: File name prefix. The string passed after this will be the first part of the names of the files generated on the target system. Required to run a scenario.

-i: Load Modules from ‘-a’ flag file, or from the default simulation configuration file. In the configuration file, there is a value called “attackFile.” The subsequent string is the location where the scenario file, coded in Python, is to be launched. If this value is missing when this option is given, the program will not run.

-l: Enables live mode. This lets the user press the ‘Enter’ key when they want to launch each scenario in the simulation. Mutually exclusive and will not run with option ‘-f.’

-n: List of attacks to run. This can be a single integer, or a range of values. A range of values can be selected by giving a comma separated list of values, or a range of values, e.g., 1,2,3; 1-3. This option is used in conjunction with the scenario data file. The attacks selected must be labeled. Duplicate attacks can also be run.

-r: Option to record all timestamps of events in the follow output file location. File format used is a utf-8 formatted .txt file.

