NIST Special Publication 1900-801

NIST Smart Grid Interoperability Test Tools

Dhananjay Anand
Kevin G. Brady Jr.
Eugene Song
Cuong Nguyen
Kang Lee
Gerald FitzPatrick
Allen Goldstein
Ya-Shian Li-Baboud

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1900-801

C Y B E R - P H Y S I C A L S Y S T E M S

National Institute of Standards and Technology
U.S. Department of Commerce

NIST Special Publication 1900-801

NIST Smart Grid Interoperability Test Tools

Dhananjay Anand
Cuong Nguyen
Eugene Song
Smart Grid and Cyber-Physical Systems Program Office
Engineering Laboratory

Kevin G. Brady Jr.
Ya-Shian Li-Baboud
Software Systems Division
Information Technology Laboratory

Kang Lee
Gerald FitzPatrick
Allen Goldstein
Quantum Measurement Division
Physical Measurement Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1900-801

February 2019

U.S. Department of Commerce
Wilbur L. Ross Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Undersecretary of Commerce for Standards and Technology

Certain commercial entities, equipment, or materials may be identified in this document in order to describe a concept. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that these entities, materials, or equipment are necessarily the best available for the purpose. All registered trademarks or trademarks belong to their respective organizations.

National Institute of Standards and Technology Special Publication 1900-801
Natl. Inst. Stand. Technol. Spec. Publ. 1900-801, 53 pages (February 2019)
CODEN: NSPUE2

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1900-801

Table of Contents

Table of Figures

Table of Tables

Acknowledgement

The National Institute of Standards and Technology (NIST) would like to acknowledge the valuable leadership of the following individuals who helped guide the participants through the framework activity:

Avi Gopstein, David Wollman, Christopher Greer, NIST Engineering Laboratory

John Pratt, NIST Physical Measurements Laboratory

John Messina, Ram Sriram, NIST Information Technology Laboratory

The UCA team, including Herbert Falk, Kay Clinard, and Margaret Goodrich, for organizing and hosting the event as well as use of figures and excerpts to help illustrate the use of the measurement and test tools described in this report.

Bernhard Baumgartner, Omicron

Chan Wong, Entergy

Executive Summary

The National Institute of Standards and Technology (NIST), participated in the Universal Communication Architecture International User Group (UCAIug) Interoperability (IOP) Test Event on October 14-19, 2017 in New Orleans, Louisiana. The objective of the event was to determine if current and upcoming equipment implementations can interoperate and perform cooperatively in a myriad of substation topology configurations to meet IEC 61850 substation communications standards requirements.

One of the key NIST roles is to advance metrology and support the development of test methodologies to provide objective assessment of device performance. The project focused on test simulation development, precision timing performance and conformance of devices in substation communications networks. NIST developed prototype hardware and software to provide support for the IOP from a neutral, third party perspective to gauge both the ability to evaluate the devices under test (DUTs) and the test tools.

The lessons learned will be applied to continue supporting industry needs and efforts for evolving precision timing and communications interoperability requirements in wide area measurement, protection and control (WAMPAC). The IEEE Conformity Assessment Program (ICAP) is one such program that has launched a Phasor Measurement Unit (PMU) test and certification program for IEEE C37.118 Standard for Synchrophasor Measurement [1] and is developing other test suite specifications of the use of IEEE 1588 Precision Time Protocol[2] profiles in power systems applications[3][4]. A test plan was developed for IEEE C37.238:2011[5][6] and serves as the basis for the Test Suite Specification (TSS) currently being drafted for the IEC 61850-9-3/ IEEE C37.238 Precision Time Protocol Profile (PTP) for Power Utility Automation [7].

The NIST Smart Grid team has developed three key test capabilities, (1) real-time digital simulation of a test power system and respective signal generation capabilities, (2) online monitoring of network clock synchronization performance among the DUTs designed to maximize compatibility with a diverse variety of hardware and software interfaces, and (3) an IEC 61850 Sampled Value (SV) message interoperability analysis tool to experiment with test methodologies in ensuring robust and reproducible test methodologies for striving towards IEC 61850 interoperability.

The tools are installed in a portable test harness to support power systems interoperability testing in testbeds, interoperability events, and potentially for field testing to acquire more realistic data for analysis and improvement of future standards. The portable test harness will also be instrumental in the development of robust and reproducible test methodologies well-aligned with realistic operating conditions under which utility end customers will be using the intelligent electronic devices (IEDs) to ensure the test methodologies are relevant and transferable to industry.

Through the experiences and results from the IOP event, the tools have demonstrated the capability to effectively interface with, acquire data in real-time and analyze, both offline and in real-time, the interoperability of the IED communication infrastructure and the precision timing infrastructure including the PTP Grandmaster, the Transparent (TCs) and Boundary Clocks (BCs), and the IEDs with a cumulative uncertainty within ±1 μs.

1 Introduction

This section provides an introduction for the document, comprised of the following sections:

- Section 1.1 provides a brief introduction to the UCAIug IoP event.
- Section 1.2 describes the motivation for NIST participation.
- Section 1.3 outlines the structure of the report.

1.1 UCAIUG Interoperability Test Event

The IEC 61850 users group held an interoperability test (IOP) event in New Orleans, Louisiana from 14 through 19 October 2017. Over 208 key stakeholders with an international representation of utilities, including several from North America, system integrators, device manufacturers, and application developers participated in the event serving to test and demonstrate substation communication and relevant aspects of configuration, data acquisition, timing, and security interoperability capabilities under both normal and stressed conditions.

The purpose of the IOP is to provide a collaborative environment where vendors can integrate their products, designed and manufactured to be compliant with the IEC 61850 series of standards, for interoperability testing with other makes and models of IEC 61850 products and prototypes. The benefits to the participants are technology improvement in terms of available products on the market, increased collaboration among end-users and product engineers, standards improvements through better understanding of customer performance needs and whether standards provide sufficient clarity to enable multiple vendor products to interoperate.

Through the IOP participation, the NIST team has developed the following:

- Real-time digital simulation and electrical signal generation as reference signals to the Intelligent Electronic Devices (IEDs).

- Online clock synchronization performance and conformance tools comprised of:

  (1) An eight-channel portable test harness to measure offsets between Pulse Per Second (PPS) clock pulses. The harness includes compensation for differences in interface electronics and in cable lengths.

  (2) A prototype software suite to enable self-discovery of PTP devices and distributed monitoring of PTP attributes and eventually clock synchronization performance.

- An offline interoperability analysis tool to verify the IEC 61850 Sampled Value (SV) messages.

This report will provide a discussion of the design considerations and implementation of the hardware and software suite which can be used at future interoperability test events as well as a portable tool set.

1.2 NIST Motivation

The objective of the IOP is to demonstrate and perform functional tests of products from multiple vendors working together in an electrical substation-like network configuration.

NIST's role in the IOP was to provide reference test signals, system timing error estimation and measurement, verification of pertinent PTP attributes, record system timing data, and record data under various test scenarios for development of offline tools and test operating condition simulations.

In building the portable test harness to support the event, NIST is better positioned to support industry by:

- Building technology competency and verifying tools and methodologies to ensure results are repeatable and reproducible under more realistic operating conditions compared to a testbed environment.
- Prototyping and establishing a web-based software architecture for federated online testbed device discovery, monitoring, data logging and analysis, as well as remote control of devices.
- Improving understanding of the product technology landscape and industry adoption of current, revised, and future standards. This will improve NIST's value to the standards setting organization as they update requirements and support the development of test methodologies for conformity assessment programs.
- Providing neutral guidance through direct interactions with device vendors' engineers who are implementing products.

1.3 Report Structure

Beyond the introduction to timing in this section, this report on NIST test and measurement tool suite for interoperability testing is organized as follows:

- Section 2: NIST Test and Measurement Tools
- Section 3: Test Event Summary
- Section 4: Key Challenges
- Section 5: Analysis and Conclusion

2 NIST Test and Measurement Tools

Among the key aspects of interoperability testing is the capability to reliably and accurately assess the performance of given devices, from heterogeneous vendors, working in concert to satisfy requirements. Clearly, these assessments are to be performed under operating conditions that reflect practical usage scenarios and in expected but stressed operating modes. The NIST Smart Grid team has developed a portable tool suite comprised of reference simulation capabilities with requisite signal emulation and distribution hardware to reproduce reference signals (electrical signal waveforms, time signals), monitor applications for timing network packet conformance and delay performance, as well as time reference offsets using the PPS

device output.

The tool suite includes a set of device configuration files to ensure repeatability and reproducibility of the tests. This section provides the technical design and implementation details for each of the capabilities. Figure 1 presents an overview of an interoperability test showing the time and electrical waveform signals produced and measurands, packets, digitized data, and timing pulse outputs monitored by the NIST tool suite. The signal types and monitoring systems are designed to work with as wide a variety of substation automation devices (IEDs) as possible.

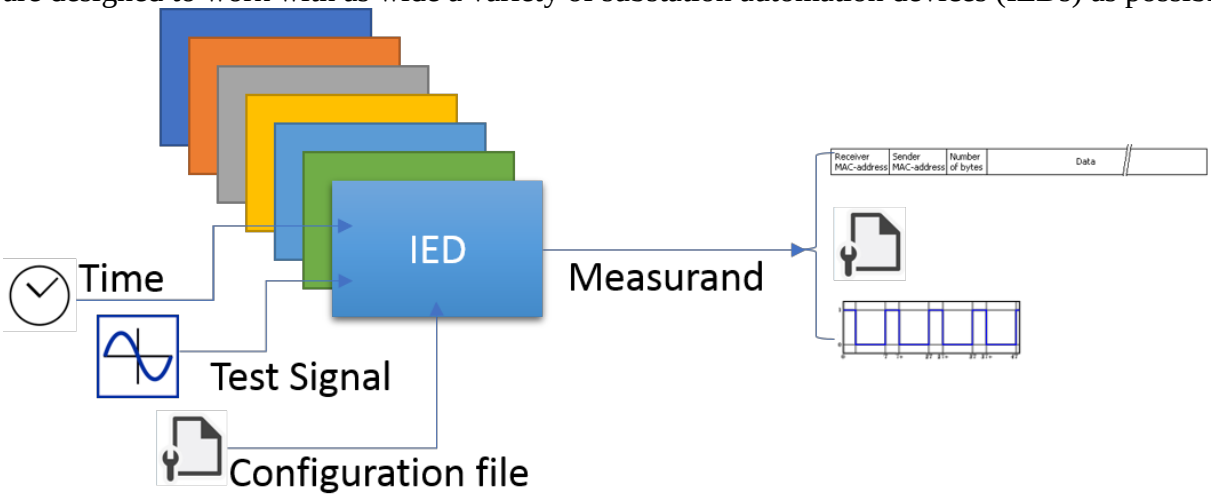

Figure 1. Overview of interoperability testing approach.

The tool suite developed by NIST is illustrated with callout boxes in Figure 2 showing where in the interoperability test they were used. Devices indicated with orange callout boxes are commercially available test and measurement tools including the real-time digital simulator, a Global Positioning System (GPS) emulator, and a network test and measurement tool. The tools developed by NIST to provide the timing reference signals and to measure the clock pulses are identified by the green callout boxes. The conformance and interoperability software to perform packet monitoring and analysis are identified by the blue callout boxes, including the IEC 61850 SV and NIST PTP dashboard applications. The following subsections discuss each of these tools in detail.

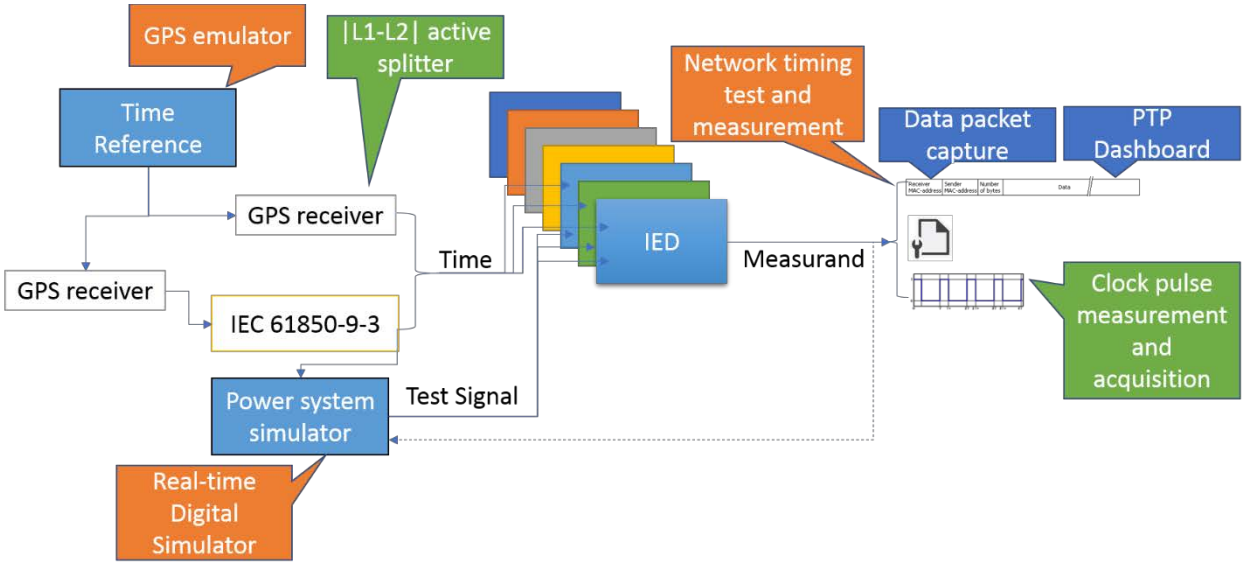

Figure 2. Smart Grid test and measurement suite.

2.1 Real-time digital simulations and signal generation capabilities

Reference electrical signal generation

NIST participated in the simulation group to provide simulated test signals for relays and merging units (MUs). Since the primary purpose of the plugfest was to evaluate the interoperability between substation equipment, a simulation was developed to provide reference signals to all the interacting devices to ensure neutrality and reproducibility. To this effect, a model of substation circuit was executed on a real time digital simulator and emulated signals from the model were distributed to all participants during the event. Nominally, this simulator produced pseudo-sinusoidal waveforms that recreated dynamic events that might be expected in a substation. Events like voltage and current transients that would trigger logic within relays and merging units (MUs) were of interest to the plugfest organizers. Many of these events do occur in response to stimuli, for example in the case of switching transients following a tap change of a circuit breaker relay action. Recreating these closed-loop events required the use of a hardware-in-the-loop (HIL) simulator to produce the simulated signals. The real time simulation was executed on an Opal-RT OP4600 Real-time Digital Simulator (RTDS).

Reference substation simulation

Several test scenarios were developed in conjunction with participants and engineers at Opal-RT. The base case used for producing the simulation was a shared reference substation design provided by Entergy (shown in Figure 3 below).

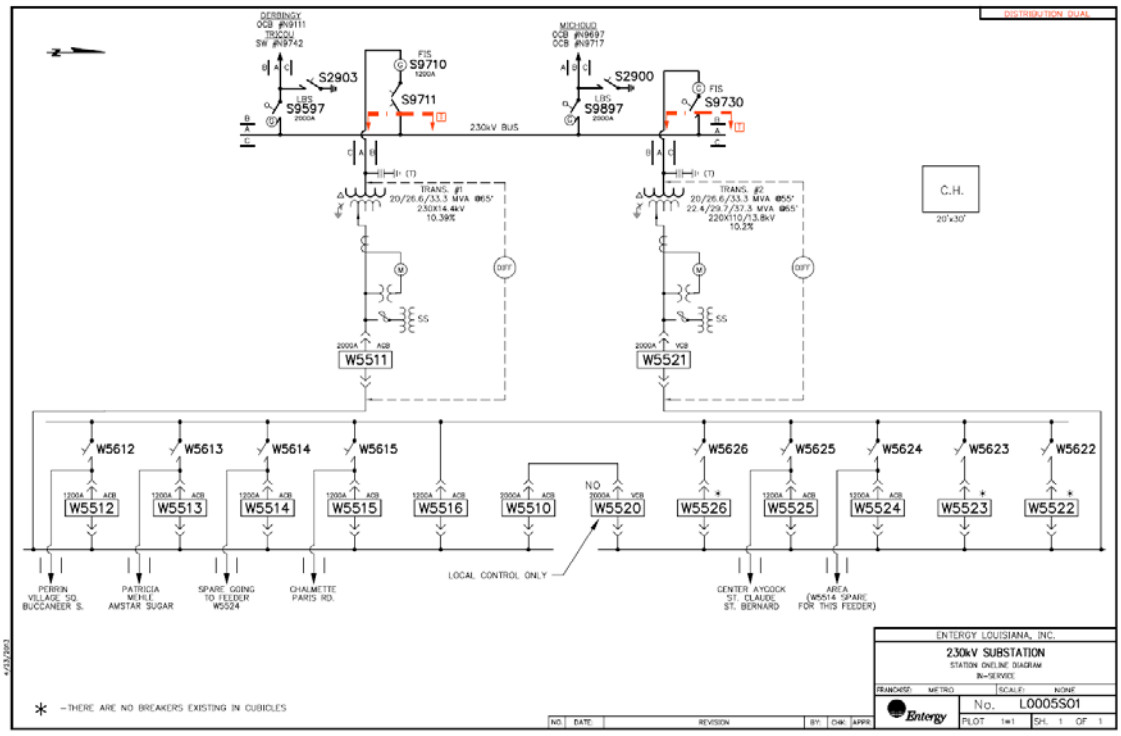

Figure 3. Reference substation design. Figure source: UCA [9].

This 230 kV substation was meant to capture the essential components of a distribution substation. Note the focus on differential protection, load tap changers, banked circuit breakers (load control) and voltage support components on each load transformer. This one-line diagram was instrumented with measurement devices, such as merging units (MUs), actuators (relays), controllers (Load Tap Changers (LTCs), Supervisory Control and Data Acquisition (SCADA) load control) and differential protection (process bus relays) from a variety of manufacturers to produce the integrated test setup shown the in Figure 4 below. This setup captures the base case of operation allowing multiple repeated tests of substation operations and communication with devices from different vendors.

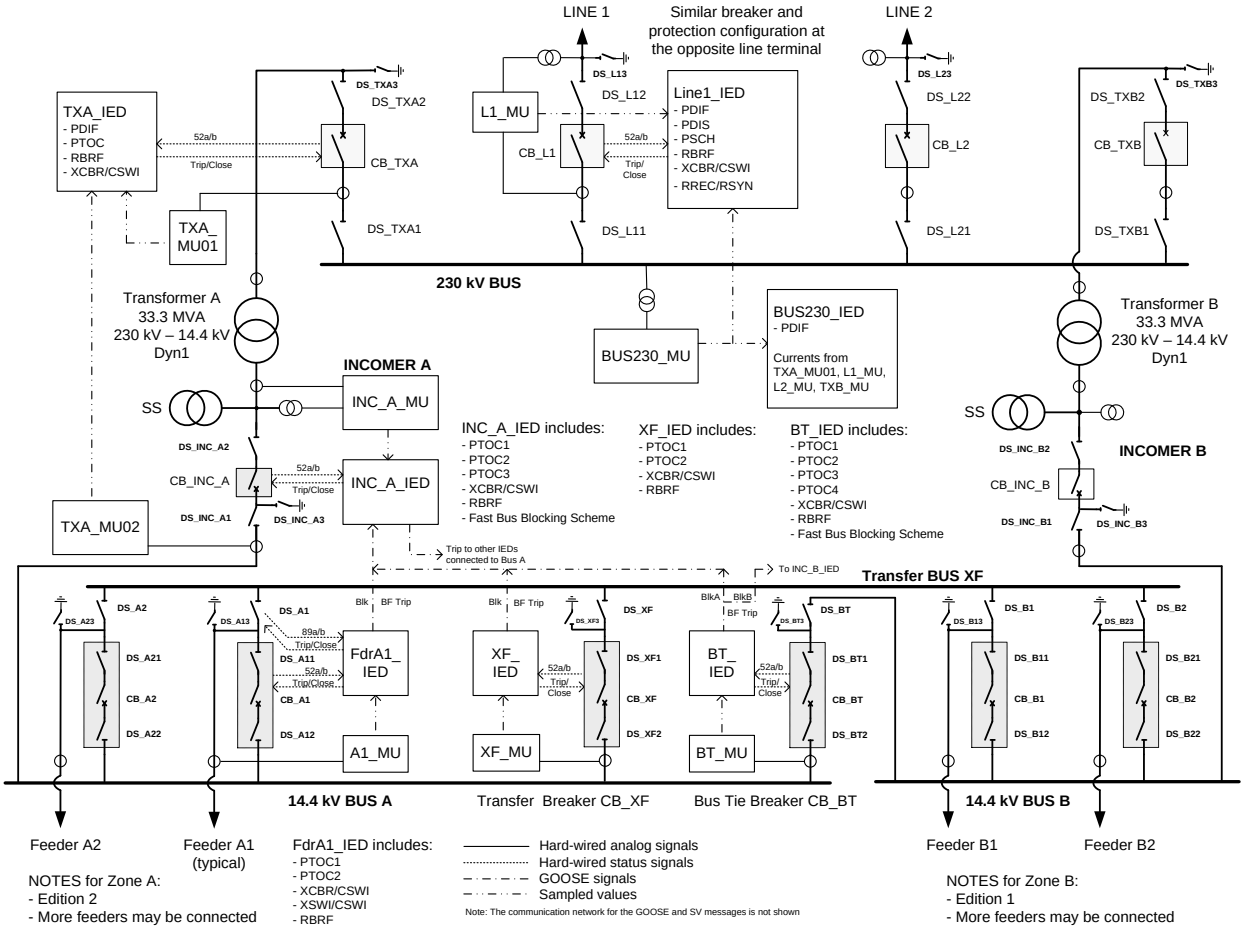

Figure 4. Integrated test layout. Figure source: UCA [9].

Given the reference design above, an integrated test application for bus protection scheme between two distribution substations connected by a 230 kV bus instrumented with measurement devices (merging units), actuators (process bus relays), and controllers (load tap changers) was developed. Refer to the locations labeled “RTDS” in Figure 5, these points in the substation were simulated using several different real-time simulators to provide the reference signals as well as to serve as virtual IEDs.

NIST built the simulation for the RTDS in Transformer Bay 1 and for the bus tie breaker in the process bus. Each shaded rectangle in the figure represents a section of the circuit emulated using an RTDS system. Note that the transformer bay is one such simulation island as is the bus tie breaker bay. The bus tie breaker simulation is an electromagnetic transient simulation intended to capture the transient recovery voltage (TRV) of a full load transfer between two substation buses. TRV is the voltage transient that occurs across the terminals of a pole of a switching device upon interruption of the current. The transient recovery voltage is a function of the Thevenin equivalent inductance in series with the breaker, where in our simulation the transformer inductance was the major source of the Thevenin equivalent inductance. When the fault current is interrupted by the circuit breaker, the source voltage supply will also supply

current to stray capacitance in the breaker assembly to bring the capacitor voltage to the system voltage. TRV response timescale is on the order of a few tens of microseconds.

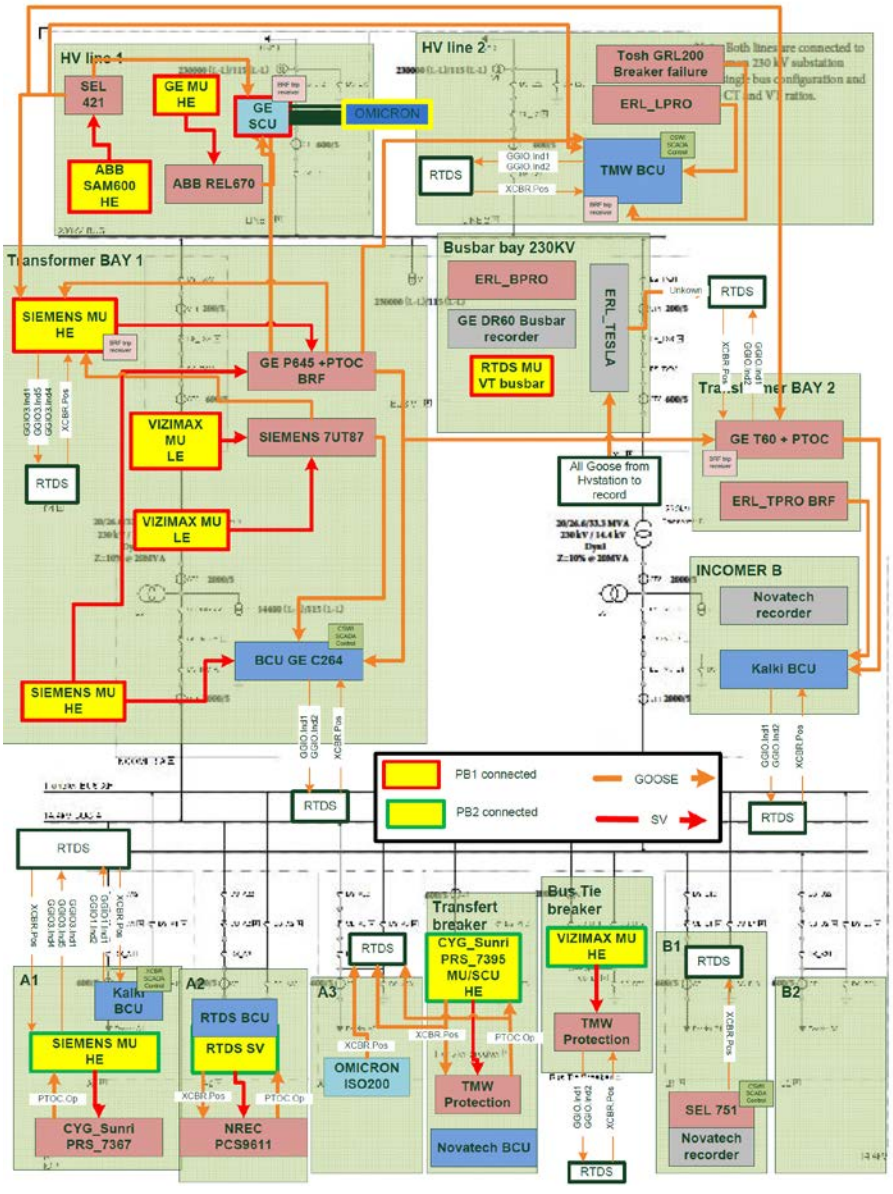

Figure 5. Integrated Test Layout. The goal of the test is to create a substation automation network. The RTDS provides the reference signals for each test. Figure source: UCA [9].

The current in the capacitor will have a natural frequency determined by the values of inductance and capacitance. The simulation generated voltage and current signals that were directly connected to a merging unit. The transients were triggered when a Generic Object-Oriented Substation Event (GOOSE) message ‘XCBR’ was transmitted by a relay or a human operator. The alignment of switching command to the transient signals was achieved by locking the simulator to the timing infrastructure used by the MU and the circuit breaker controller/relay.

This was achieved by deploying client drivers for PTP messages (Figure 6) and for GOOSE messages (Figure 7) in the RTDS and synchronizing the digital to analog converter to a precise Global Positioning System (GPS) locked clock.

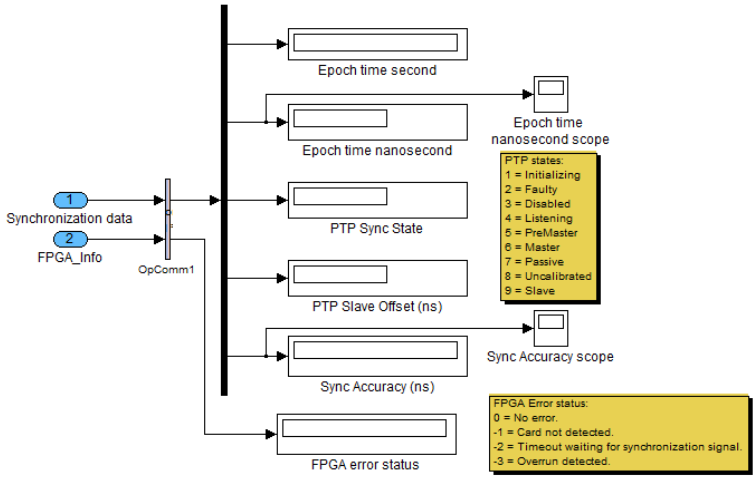

Figure 6. PTP Synchronization deployment in the real-time digital simulation.

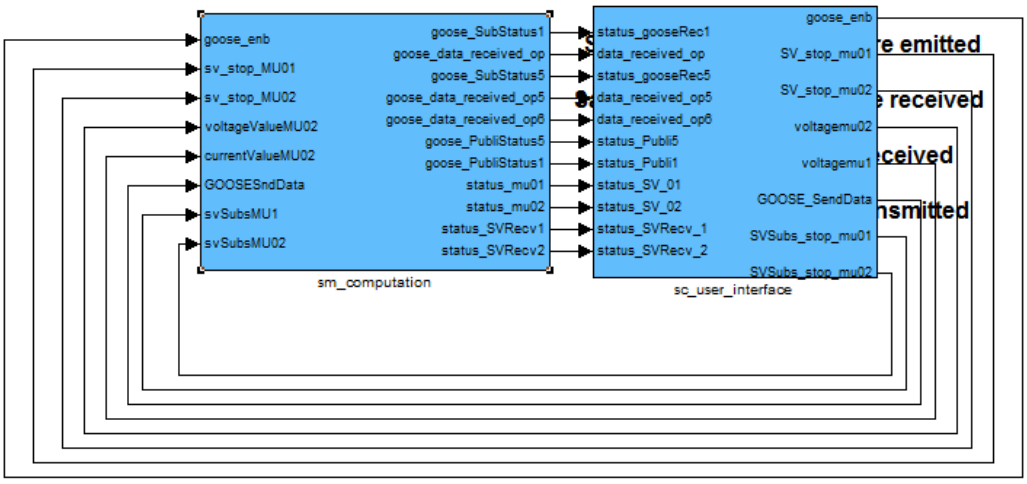

Figure 7. GOOSE message implementation on RTDS.

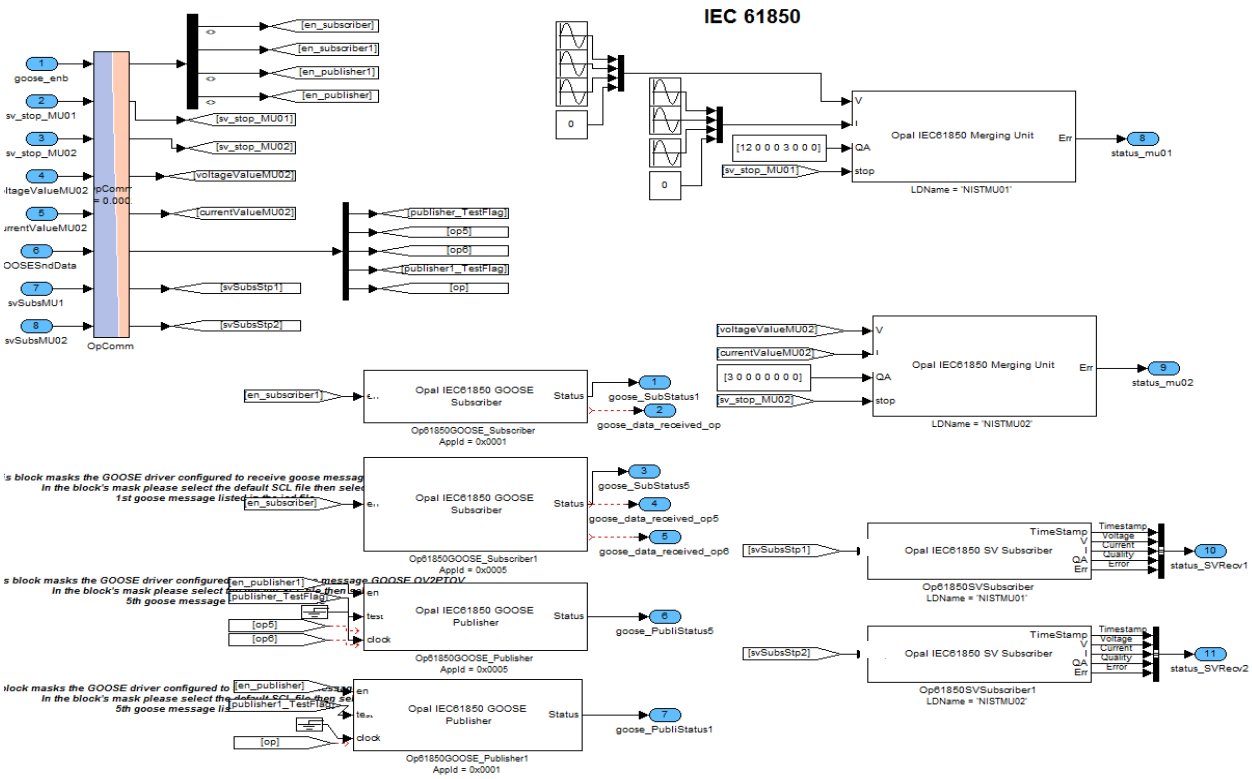

Figure 8. Real-time digital simulation of IED voltage and current waveform measurements using SV and GOOSE communications.

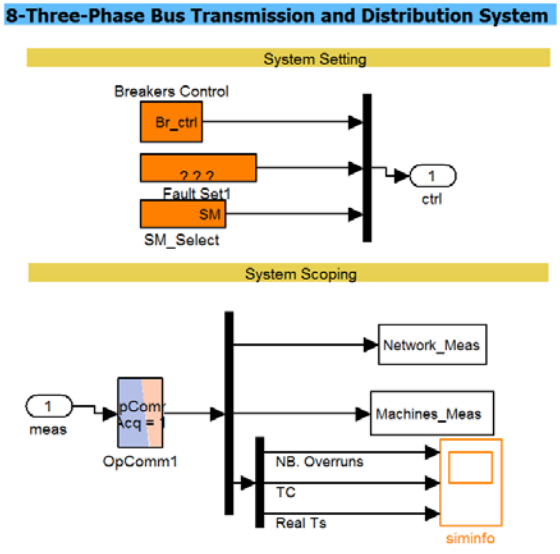

Figure 9. MU HIL integration.

The transformer bay simulation was intended to test differential protection that utilized MUs for measurements. As shown in Figure 8 and Figure 9, the RTDS was programmed to provide three-

phase voltage and current waveforms to MUs and to generate SV messages for simulated MUs to provide HIL simulation for the IEC 61850 message exchanges. Differential protection requires measurements from two locations that are processed by relays which in turn trigger protection circuit breakers when tripping conditions are met, depicted in Figure 10.

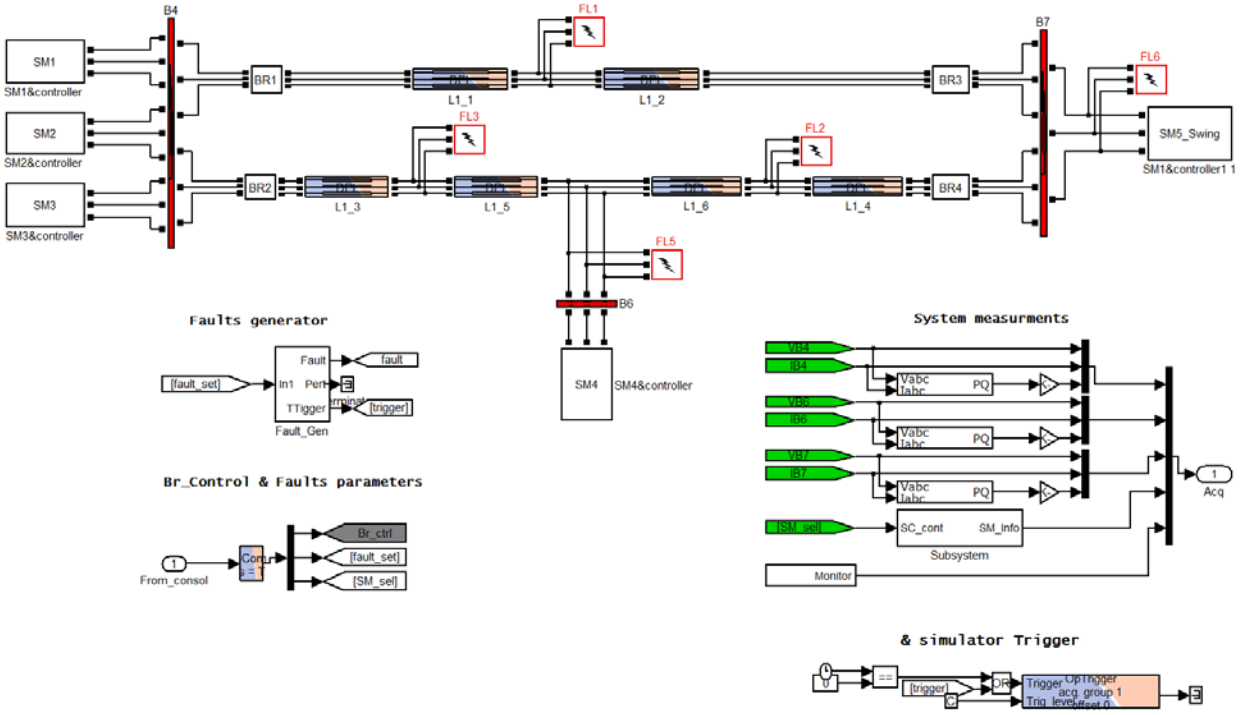

Figure 10. Transformer bay simulation where transient faults are triggered providing faulty voltage and current waveforms.

The IOP test procedure required the evaluation of tripping action when one of the DUT MUs and the other one is simulated. This is an interesting challenge since the differential phase measurement made by the relay is based purely on the SV messages which means that the phase of the signals must be calibrated so as to be indistinguishable when a real MU is swapped for a simulated virtual MU (VMU). This phase calibration had to hold under potential transient test cases as well. In order to achieve this, a dual simulator setup was constructed as shown in Figure 11. The signal generator RTDS system simulated a tap changing transformer subjected to varying load states including a fault state. Measurements from the transformer primary and secondary windings were output as analog waveforms on Signal Channels B and A respectively. In the base case, differential measurements between MU_1 and MU*_1 were used by the relay to trip a circuit breaker when appropriate. A second RTDS used a VMU. When MU*_1 was present, the VMU operated in calibration mode and measured the latency of the transduction process in MU*_1. Then VMU set its internal delay to match so that in the absence of MU*_1 the VMU could serve as a seamless replacement by simply disabling its “simulated” bit in the

SV message. This simulation requires prior knowledge of the SigGen sequence and precise synchronization between the two RTDS systems, which was achieved by transferring the clock from SigGen to VMU over a dedicated optical synchronization link.

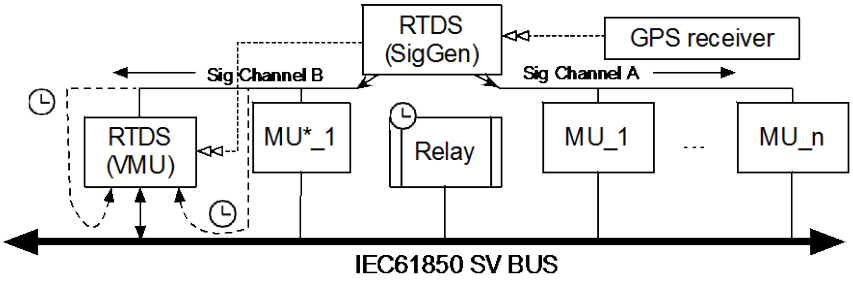

Figure 11. HIL integration of a MU and VMU.

2.2 GPS Signal Emulation and Distribution

Most substation automation devices require time synchronization for operation. Given the diversity of devices available, multiple potential sources of time were provided to vendors to synchronize their device clocks. For baseline operation, a GPS receiver was placed in the room connected to an antenna placed outside a second-floor window. This receiver had a precise oven-controlled crystal oscillator (OCXO) and provided time to devices in the plugfest room that did not have direct line of sight to the sky. Time was distributed among the devices via multiple modes including a 10 MHz monotonal frequency source, a 1 PPS source, Inter-rage Instrumentation Group's, (IRIG)-B time code [9] transmissions and IEEE 1588 based network time [2]. Some of the tests proposed by the timing subgroup also proposed manipulating the GPS signal to replicate time discontinuity events such as the leap second and anomalous events related to sporadic fading of the GPS signal strength due to cable or antenna malfunction. These test cases required the use of a GPS simulator to produce a GPS signal for those specific test events. NIST provided a 16-channel Spectracom GSG-5 GPS signal simulator capability of simulating up to 16 satellites, distributing GPS L1 signals. The GPS simulator includes a 1 PPS phase output calibrated to within 10 ns of the simulator's Radio Frequency (RF) output to serve as the reference in verifying the DUTs' time uncertainty. The simulator included scripted test scenarios for both positive and negative leaps seconds and GPS week rollover.

Since the simulated signal would diverge from the baseline GPS signal being used by many devices on the floor, it was critical to not allow the simulated signal to leak into the plugfest room. Once the simulator was installed and connected to a test device, a site survey was conducted using a micro-strip antenna with low directivity connected to a GPS receiver with a carrier-to-noise density (C/N_0) display to ensure there was no signal interference in the room between the true GPS signal and the simulated one. In cases where we observed spurious leakage from connectors and amplifiers, we applied adhesive shielding and attenuators to minimize radio leakage into free space.

The plugfest test plan required some of the GPS test scenarios to be conducted on up to 8 devices simultaneously, NIST provided a GPS signal splitter to distribute the simulated GPS signal to a maximum of 8 GPS receivers. The splitter had to meet the tight requirements of the test plan.

The most critical parameter was the phase offset between ports in the splitter. The phase offset between each signal in the splitter is ideally minimal and must be calibrated prior to the test event to ascertain the differences between splitter ports are either negligible or accounted for. The test cases at the plugfest used timing pulses from the DUT GPS receivers as a way of confirming their uncertainty, lock state and other time signal behavior during time discontinuity or anomalous events. Phase offset errors introduced by the splitter would propagate as time errors in the output signal.

The splitter provided used a low-noise amplifier with a Channel Gain ≥ 14 dB [10]. The amplifier was coupled to the 50 Ω impedance bridge ganged to express eight ports calibrated at L1(1575MHz) and L2(1227MHz) each with a specified standing wave ratio (SWR) ≤ 1.8, and phase matched to ≤ 1°. The GPS receivers in the plugfest typically use a specific antenna or low-noise amplifier (LNA) setup, namely all receivers used direct current (DC) power injectors of different specifications and use safety or line short logic to test the antenna wiring prior to start up. To ensure compatibility to the variety of voltages supported by the receivers, all the ports on the splitter were DC blocked with a blocking capacitor and a 200 Ω@0 Hz dummy load.

Figure 12 shows the combined frequency response properties of the amplifier and splitter circuitry. Note that at the GPS L1 frequency of 1575 MHz at Marker 1, each splitter provided 14.46 dB of net gain with a measured Standing Wave Ratio (SWR) of 1.710, which is a measure of the impedance matching between amplifier and splitter circuitry.

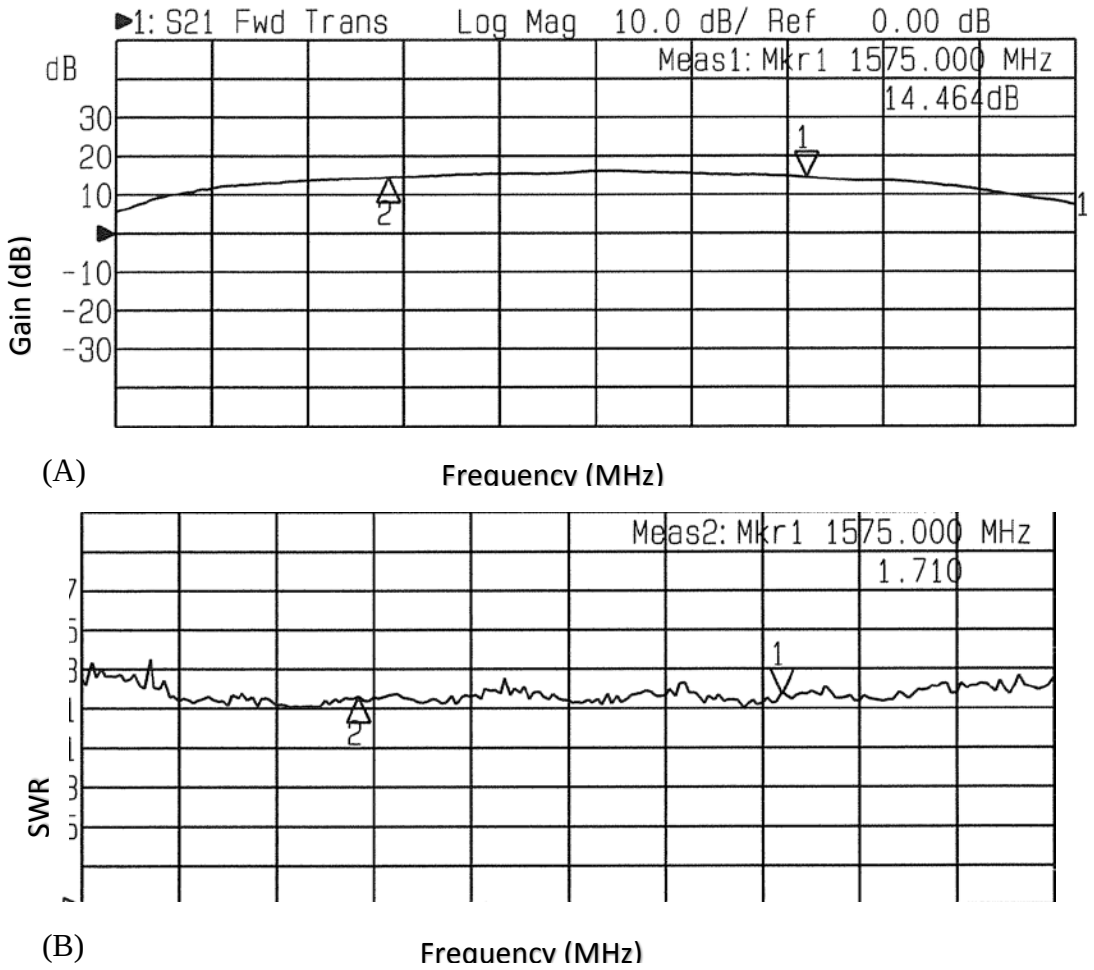

Figure 12 Frequency response as characterized by Gain (A) and SWR (B) of the GPS signal splitter

2.3 Phase measurement for 1 PPS signal

Pulse Per Second (PPS) is an electrical signal that has a width of less than one second and a sharp rising edge that nominally repeats at the top of each second according to the oscillator driving the PPS signal. The time between the oscillator's top of the second and the PPS output is also subject to uncertainty on the order of a few nanoseconds. Most of the GPS receivers and some of the IEDs had a PPS output. In all cases, the PPS generator in the equipment was expected to align the PPS signal to the Coordinated Universal Time (UTC) second when the device was locked to reference time. The time offsets between the rising edges from test devices, therefore, was used as a reliable measure for synchronization error and stability of time under fringe operation states. In order to simultaneously assess the interoperability of several devices, NIST built a PPS signal acquisition tool, Figure 13, to simultaneously measure the relative time error the PPS rising edges with up to 8 participating DUTs.

The requirement can be stated as follows:

1) Measurement interface supports diverse logic levels and impedances.
2) Support efficient calibration of different cable lengths.
3) Time error between edges must be measured at 2 ns resolution in reference to UTC.
4) All measured offsets to be logged to a file.
5) Edge detection must have sub-nanosecond jitter.
6) Support concurrent measurements of up to 8 DUTs.

The test harness consisted of a Tektronix MDO4104 (USB control) logic analyzer with triggered capture as shown in Figure 14.

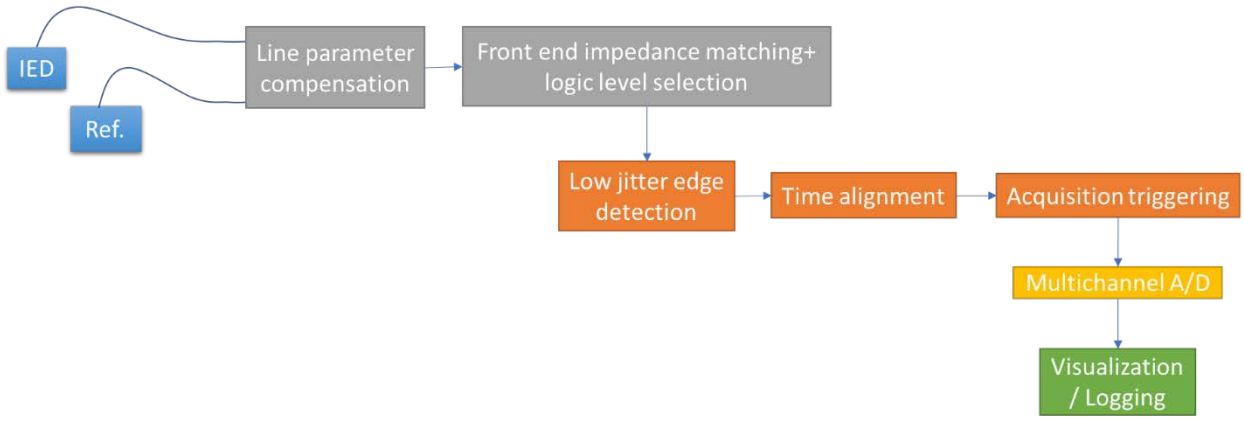

Figure 13. PPS Measurement test harness design. The test harness integrated a logic analyzer with TDR, multi-channel timestamping, low-jitter data acquisition, data logging, and visualization capabilities for interfacing with diverse DUTs.

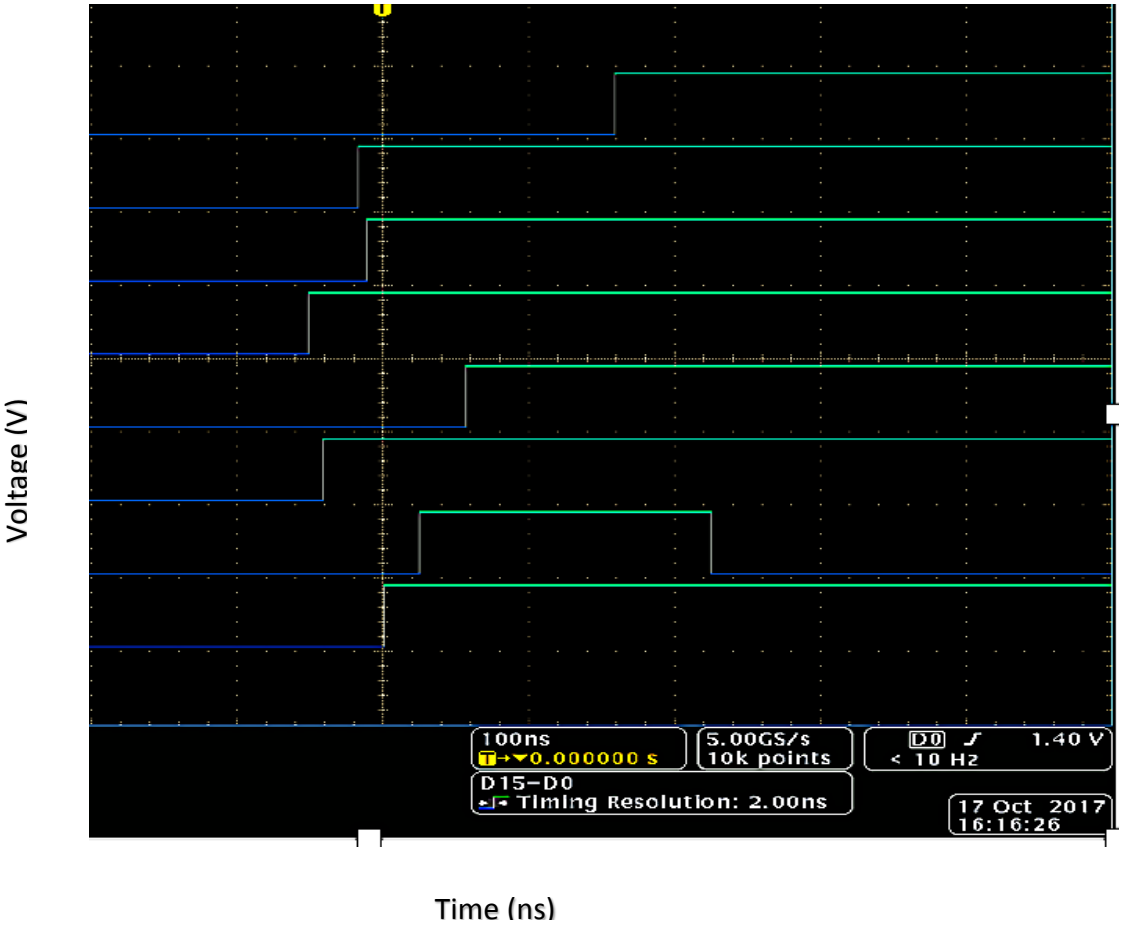

Figure 14. Oscilloscope capture of PPS offsets triggered from the reference GPS PPS output.

The logic satisfied conditions 5 and 6, but conditions 1-4 required the design of the test harness with additional capabilities. A National Instruments PXI-8196 embedded controller was integrated to trigger a data capture event in the logic analyzer over a universal serial bus (USB) port. The digital waveform data from the logic analyzer was then collected by the embedded controller using the Tektronix provided IVI driver. Each data point was time stamped and recorded to a comma separated value (CSV) file.

The process of reading multiple digital waveforms from the logic analyzer did occasionally take more than one second to execute and this posed a challenge since it constrained the requirement for at least 8 simultaneous measurements. This challenge required some modification of the driver to issue synchronization and configuration commands only once prior to the start of the capture process. We anticipate this modification will have to be improved over time as future iterations of this harness are developed.

Requirement 3 placed a 2 ns resolution budget for the acquisition and in addition required the error measurements to be traceable to UTC (or a provided timing reference). This was achieved by adding a National Instruments PXI-6682 timing module to the embedded controller. This module has a GPS receiver offering us the capability to lock its internal oscillator to a GPS

reference. This GPS time was used to time stamp the data received from the logic analyzer after every capture event prior to logging. We then used the PXI timing bus to transfer the GPS disciplined oscillator reference to a PXI-6653 clock and trigger signal synchronization module which in turn was used to provide several timing signals to drive the capture trigger as well as to serve as a time base for the logic analyzer. In this way, the skew in the edge detection and triggering was minimized. Also, by providing a time base reference to the logic analyzer, we were able to mitigate relative drift between the logic analyzers clock and obtain measurements traceable to UTC.

Requirement 2 was particularly challenging since the electrical distance of the cables used to collect the PPS signals introduced delays exceeding 100 ns since devices to be measured were located several meters away from the test harness. Further, each device had different electrical interconnects requiring cables and connectors to be provided by the vendor in some cases. The diversity in cables and connection hardware further introduced variance between the measurements that threatened the 2 ns time error resolution budget. To compensate for line delay, a time domain reflectometer (TDR) was designed on the PXI-8196 modular controller. Prior to the start of a data capture event, the logic analyzer reconfigured as a signal generator and triggered by the timing and synchronization module to produce a fast rise time pulse injected into the cable system at the measurement end. The far end of the cable was left disconnected when the impedance of the PPS output from the measured device was unknown. The open end of the cable presented a high impedance to the incident pulse. The change in the characteristic impedance (impedance discontinuity) would cause some of the incident signal to be reflected towards the source. The time delay of the reflected pulse was measured using the PXI-6682 and applied as correction to all future measurements.

Clearly the reflected pulse would also experience attenuation and dispersion. Based on known data specifications for the RG-58 cable, a 100 ns TDR pulse has an attenuation of approximately 0.9 dB per 304.8 m, which means the pulse must travel up to approximately 2 km to lose 50% of its initial amplitude. TDR pulse is composed of different frequencies that travel along the cable system at different speeds. This change in speed causes distortion known as dispersion. The distortion is a phase shift of each of the individual frequency components of the TDR pulse and is a function of the distance traveled by the pulse and its frequency component profile. Dispersion distorts TDR pulses regardless of whether the system is lossy or not – this is a concern since dispersion in the reflected pulse directly impacts the effective resolution of the TDR system. There is a significant body of work in improving resolution in TDR systems, we used a simple approach based on prior studies of using a bi-phasic truncated exponential pulse. The exponential frequency envelope has been shown to be preserved with minimal distortion to the exponential shape in TDR systems. Over 10E6 trials, we observed less than 500 ps variation in the reflected time delay for a 10 m RG-58 cable.

Lastly, the variety of voltage levels and driver electronics used by various manufacturers meant that the Requirement 1 had to be added to the feature set. This was achieved by providing independent edge detection levels for each of the 8 channels. This was straightforward to

implement using the Interchangeable Virtual Instrument (IVI) driver for the logic analyzer providing detection capability for both 3.3 V Complementary Metal Oxide Semiconductor (CMOS) or 5 V Transistor-Transistor Logic (TTL) levels.

Figure 15 shows the PPS capture utility providing a single graphical user interface (GUI) for all the hardware components used to measure the PPS offset to UTC to a 2 ns precision relative to the reference.

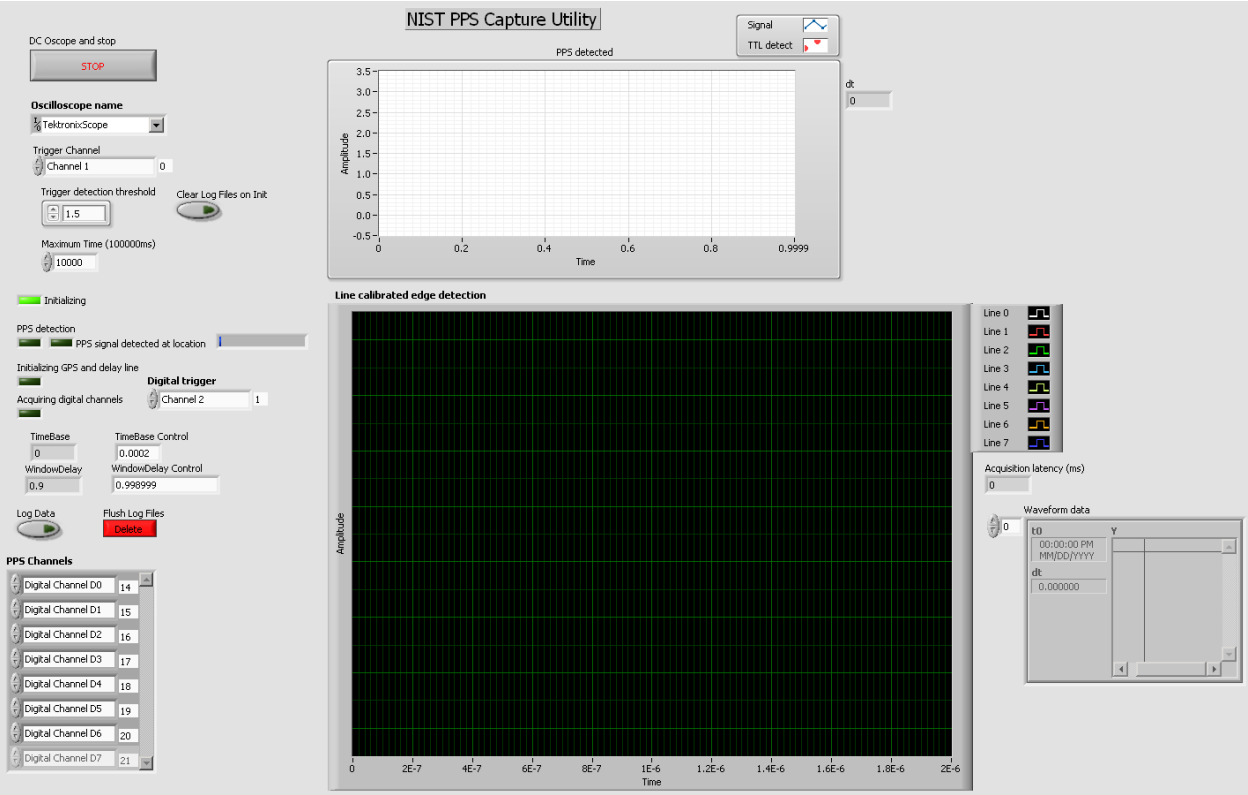

Figure 15. PPS capture utility interface.

Figure 16 shows the software user interface to the timing subsystem used to provide reference clocking functions to the logic analyzer, the data logger and the TDR system.

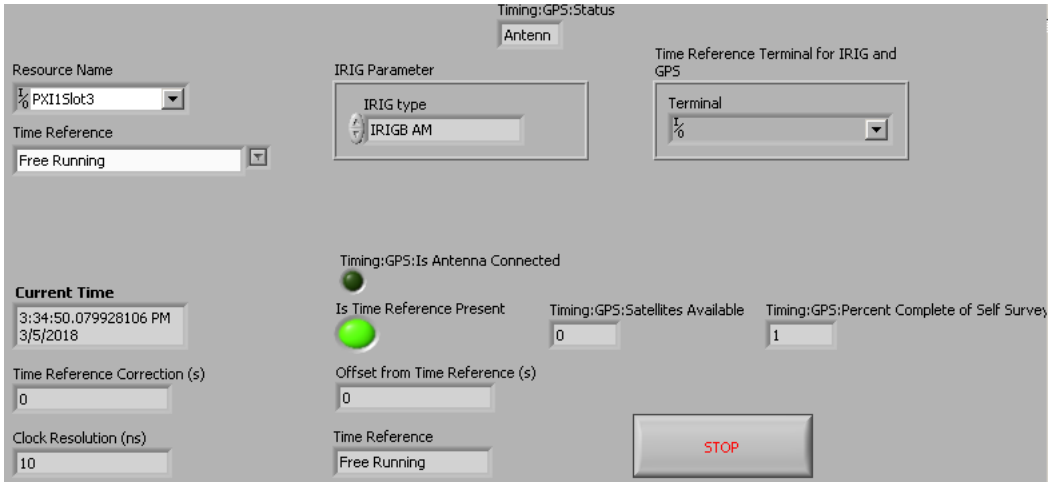

Figure 16. Reference clock status GUI.

2.4 Packet delay variation measurement

Jitter in a network synchronized clock may be due to a combination of 1) jitter in the reference, 2) jitter in the network communication channel and 3) jitter in the filtering and clock control circuitry. Factors 1) and 3) were addressed by using the GPS emulator and the PPS capture utility, respectively. Factor 2) requires accurate measurements of the variation in PTP synchronization packet transit times over the network. We used a Calnex Paragon-X to measure the Packet Delay Variation (PDV) of PTP Sync messages, as shown in Figure 17. The figure below shows the general setup for measuring the PDV of timing packets. The setup places the Paragon-X device in-line with the primary network link between the PTP master clock and the Slave clock. As shown in the diagram, the PDV measurements also need to be traceable to a single timing reference. We used the PXI-6682 to produce a 10 MHz reference clock to ensure that the Paragon-X remained locked to the same reference as the PPS capture tool and the GPS simulator.

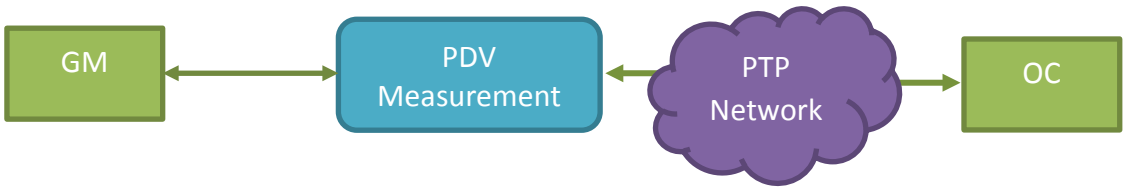

Figure 17. PDV Measurement Network topology.

2.5 PTP Dashboard

The PTP Dashboard is intended to be a software application capable of self-discovery and conformance monitoring of PTP devices on a network. The architecture of the software, where the key modules are depicted in Figure 18, enables distributed monitoring and conformance testing within a local area network (LAN) and devices at remote sites. The PTP Dashboard is a

real-time web application using a combination of Python, Django and PostGresSQL to process PTP messages and visualize changes in the PTP parameters to ensure devices update the attributes within the required time intervals. The application can also display how these clocks are responding to timing events and configurations changes based on the test cases[16] detailed by the Time Synchronization sub-group.

By monitoring the attributes during the execution of the tests, the tool supports conformance testing of each PTP DUTs for how well it complies with the IEC 61850-9-3 (2016) [3] and IEEE C37.238 (2017)[4] PTP Power Profile standards. The IOP participants can observe how the Grandmaster clock behaves given a change in operating conditions such as a loss of GPS signal (time server holdover) or an insertion of a leap second. The application also observes the parameter changes of how PTP clocks select a master clock based upon the best master clock algorithm (BMCA).

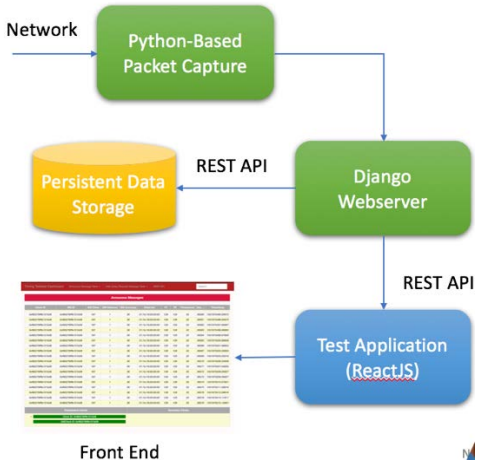

Figure 18. PTP Dashboard Software Architecture.

PTP Dashboard Design and Implementation

The PTP Dashboard is comprised of the Django web application development framework [11], a network packet capture library, Pyshark [12], a PostGresSQL persistent data store, along with ReactJS[13] to drive the front-end. The integration of Python-based technologies enabled rapid prototyping of the conformance and interoperability test software.

The design is based on the key application requirements for use in interoperability plugfests and field testing. To alleviate the need for manual configuration, a key requirement is device discovery. The second requirement is real-time packet capture, processing and analysis. Lastly, scalability to fully support all participant devices with a dashboard refresh rate fast enough to update once per second, as Announce messages and other relevant packets arrive.

PTP Device Discovery

The application can discover PTP Grandmaster Clocks, Ordinary Clocks (OCs), Boundary Clocks (BCs) and Transparent Clocks (TCs) on the network. P*yshark*, a network packet parser, enabled live captures of Announce, Sync, Follow_Up, Peer Delay Request and Peer Delay Response messages from the PTP network interface. Figure 19 demonstrates the use of the *Pyshark* library for verifying specific identifiers in each PTP message to determine whether it was an Announce Message, Peer Delay Request Message, Peer Delay Response Message, Sync Message, or a Follow_Up Message.

```
def sniffAnnounceMessages(self):

    interface_list = netifaces.interfaces()
    capture = pyshark.LiveCapture(interface_list)
    #capture = pyshark.FileCapture('/Users/kgb/Desktop/BMCA_P2P_MultiDomain2.pcapng', keep_packets=False)

    for packet in capture:

        try:

            if PyShark_Capture.get_message_type(packet) == 'Announce Message':

                Announce_Message.objects.create(

                    clockidentity=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.clockidentity'),
                    messageid=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.messageid'),
                    layer_name=PyShark_Capture.get_field(packet, 'ptp', 'ptp._layer_name'),
                    length=packet.length,
                    version_ptp=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.versionptp'),
                    messagelength=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.messagelength'),
                    flags=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.flags'),
                    security=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.flags.security'),
                    unicast=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.flags.unicast'),
                    twostep=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.flags.twostep'),
                    frequencytraceable=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.flags.frequencytraceable'),
                    timetraceable=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.flags.timetraceable'),
                    timescale=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.flags.timescale'),
                    utcreasonable=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.flags.utcreasonable'),
                    li59=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.flags.li59'),
                    li61=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.flags.li61'),
                    correction_ns=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.correction.ns'),
                    correction_subns=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.correction.subns'),
                    control=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.control'),
                    logmessageperiod=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.logmessageperiod'),
                    origintimestampseconds=PyShark_Capture.get_field(packet, 'ptp','ptp.v2.an.origintimestamp.seconds'),
                    origintimestampnanoseconds=PyShark_Capture.get_field(packet, 'ptp','ptp.v2.an.origintimestamp.nanoseconds'),
                    utc_offset=PyShark_Capture.get_field(packet, 'ptp','ptp.v2.an.origincurrentutcoffset'),
                    IP_SRC=PyShark_Capture.get_ip(packet, 'src'),
                    IP_DST=PyShark_Capture.get_ip(packet, 'dst'),
                    ETH_SRC=PyShark_Capture.get_field(packet, 'eth', 'eth.src'),
                    ETH_DST=PyShark_Capture.get_field(packet, 'eth', 'eth.dst'),
                    SRC_PORT=PyShark_Capture.get_field(packet, 'udp', 'udp.srcport'),
                    DST_PORT=PyShark_Capture.get_field(packet, 'udp', 'udp.dstport'),
                    sniff_timestamp=float(packet.sniff_timestamp),
                    sequence_id=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.sequenceid'),
                    GMClockIdentity=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.an.grandmasterclockidentity'),
                    localstepsremoved=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.an.localstepsremoved'),
                    alternateMasterFlag=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.flags.alternatemaster'),
                    GMClockAccuracy=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.an.grandmasterclockaccuracy'),
                    GMClockClass=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.an.grandmasterclockclass'),
                    GMClockVariance=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.an.grandmasterclockvariance'),
                    priority_1=PyShark_Capture.get_field(packet, 'ptp', 'ptp.v2.an.priority1'),
```

Figure 19. PTP Packet object using Pyshark.

Real-time processing

The application continuously captures Announce and Path Delay Request messages and inserts the packet data into a PostGresSQL database with Django's default functions for working with models, enabling automatic generation of APIs for interfacing with PTP packet data. Enabling real-time monitoring required multi-threaded processing to execute the packet capture code,

while concurrently running the database process, and integrating the webserver application process. The information in the database can subsequently be used through Django to connect to the front end as the back end retrieves the data from the packet capture process, providing real-time visualization of the full set of PTP devices accessible on the network.

Application data model

The web application is based on Django, Python's web development library based on the Model-View-Template architecture. In Django, models serve as information blueprint for the PTP messages to be observed, namely the Announce Messages and Path Delay Request Messages, coming from the Grandmaster clocks. Django relies on the data models to interact with a back-end database used to store data for analysis and display. The web application is driven by the database and a real-time analysis application software.

Django uses its models to drive the rest of the framework because it helps the developer to create a data driven application that can use the model to effectively communicate between the frontend and a backend database. Django contains an Application Programming Interface (API) for working with models, enabling the separation of data by the type of PTP message. The data models are based on the PTP message types. Each PTP message type, namely Announce and Path Delay Request messages, are organized as separate tables in the database for simplified data storage from the backend capture application as well as efficient data retrieval from the frontend application.

Each PTP message type has a single model for both the database and the software application allowing direction data insertion into and retrieval from the database model or retrieve information for the software application model, helping to reduce the amount of code.

The API handled the database operations such as retrieval and entry, since Django's default model functions generate the Structured Query Language (SQL) code to interact with the database. Along with being able to create database tables with models, the API for the PTP messages also provides a direct interface to the database allowing changes in the software data structure to propagate to the database without having to manually write SQL code. The API simplifies the development process and mitigates the potential for communication and consistency errors between the software and database applications. The database also enables the developer to quickly sort based on timestamps and sequence numbers from packets. Such features are useful in ensuring packets are displayed in the order it is received, notification of potentially missed packets, and avoiding errors such as displaying redundant packets.

The PTP Message APIs also allowed customization of views to allow the front-end to retrieve information from the database. The View component of the Model-View-Template architecture involves taking the data models of the PTP messages and constructing methods to access the database in a simple and efficient way. The database interface methods were organized in a separate view class for each model, which handles where the information from the database tables can be accessed and the respective data format.

Django's list generic view returns the contents of the database table when a GET request is made to the webserver. The software utilizes generic model serializers in order to transmit data from the model request function, such as a HyperText Transport Protocol (HTTP) GET request, with the format required by front end consuming the data. A View class using the QuerySet method from the API, which specifies the data set to retrieve from the database, and a serializer for the data set is executed. The application is flexible to the type of messages the frontend uses including JavaScript Object Notation (JSON) and extensible Markup Language (XML) request and responses.

PTP Dashboard Front End

The final component of Django's framework is the template which includes the website and the code to be rendered when a user makes a request to the server. Most of Django's built in utilities for templates involve a static website that a user interacts with. To create the real-time dashboard, JavaScript code execution is enabled using Webpack.

Webpack bundles all of the JavaScript code for the frontend into a single script, which can be included in our base template provided by Django. When a user makes a request the dashboard application, the base template with the script included will be returned to the user's web browser, enabling a higher-level library, such as ReactJS, to build the Dashboard within Django's framework. The Representational State Transfer (REST) APIs are simple URL endpoint functions for querying and retrieving database information for the individual test applications as shown in Figure 20. A JavaScript library can make calls to the URLs to get the data being captured from the PTP packets, giving us the ability to view our real-time network data on our webpage. Since separate view classes for each type of PTP message are created, there are also separate URLs to retrieve data for each PTP Message type.

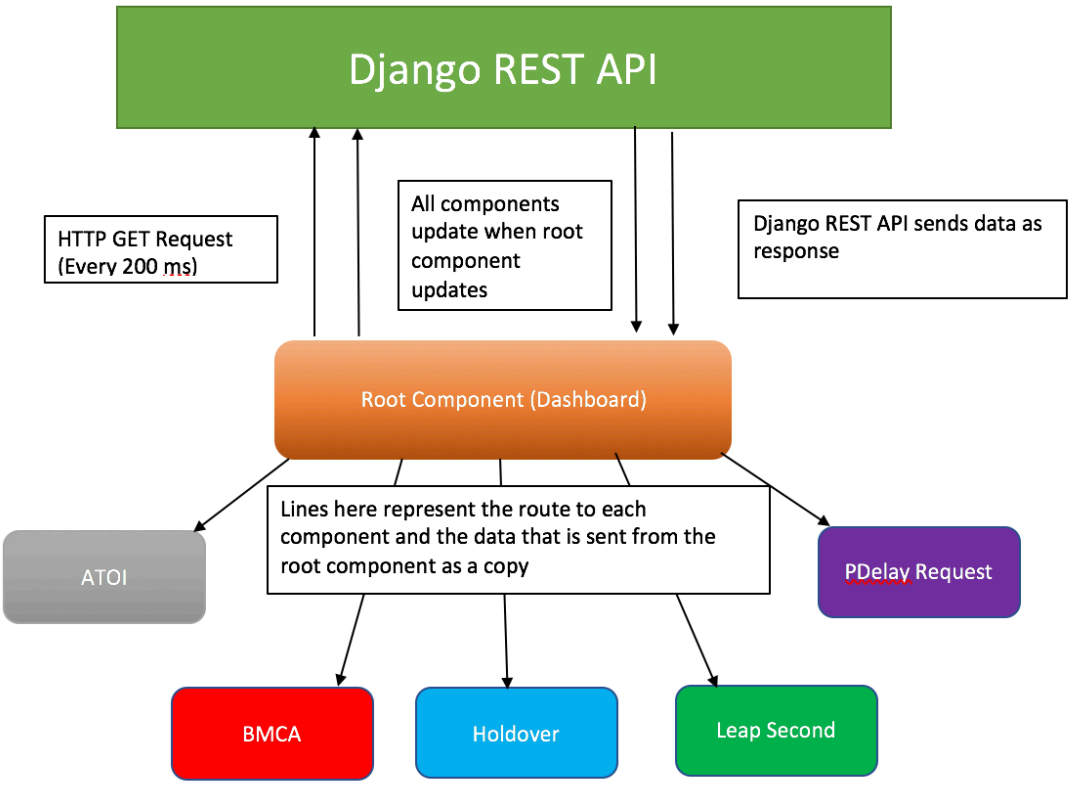

Figure 20. Test data communication to packet analysis modules using Django.

Webpack also has the ability to create a separate webserver for “hot-reloading”, which refers to the ability for changes in the JavaScript code to immediately load into the web page without the need for the user to refresh the browser. When paired with ReactJS, automatic reloading enables changes in JavaScript code to immediately propagate to the web application, another feature to streamline development. ReactJS also contains a plugin for most web browsers that allows the developer to debug problems with their site directly in the web browser. Using these tools allowed us to build a working dashboard with Django, Webpack, and ReactJS within a 3-month time frame.

ReactJS is a library that was developed for building dynamic websites to enable real-time data processing. ReactJS provided the ability to create a single-page website that encompasses our whole dashboard. A single-page website is a website that only needs to be loaded by the web browser once and allows you to have the full functionality of a website without needing to create multiple URLs. Since ReactJS is a high-level library it needs to be converted to JavaScript before it can be executed in a script through a process known as transpiling.

The PTP Dashboard application is a single page website where specific routes and the template of the website within ReactJS, when it is bundled with Webpack, is served to the user. The webpage content is continuously refreshed after the initial web server request. The web application is a single page website organized into individual subcomponents that act as child nodes of a larger component. When the parent component updates, all its child components update as well. When a user requests data in the parent component, all the child components of the PTP Dashboard will update simultaneously. The PTP Dashboard is designed to be modular, where all user requests receive data from a single parent component, and all child components update simultaneously with the parent component. Therefore, application code includes a single real-time dashboard with ReactJS and Django and specifies REST API URLs to obtain the data refresh for each test capability module. The dashboard is constructed by specifying the routes through the website and the data that should be rendered for these routes in ReactJS subcomponents.

Dashboard capabilities

The dashboard is comprised of six test modules to provide parameter verification support for interoperability test cases[16] provided by the Time Synchronization Sub-Group. The test modules enabled verification of PTP attribute compliance for Announce messages, holdover, leap second, ATOI, and multicast. The test modules chosen are based on our availability of test devices, which claim compliance to IEC 61850-9-3 (2016), which we had a single device, or IEEE C37.238 (2017), which were not yet available in our lab.

Live PTP attribute monitoring

The PTP attribute monitoring module is rendered when the Dashboard starts and consists of a scrollable table that will display the incoming Announce Messages from Grandmaster clocks on the network in real-time. This table will show attributes from the clock such as Media Access Control (MAC) addresses and Internet Protocol (IP) addresses, clock identities, as well as encoded information to indicate priority, clock accuracy, clock time source, and timestamps of when the packet was captured. The timestamps provide indication as to whether the most recent Announce Messages are being captured sequentially.

The module can be used to test and validate the backend of the dashboard as well as the network and device configurations based on the Announce messages. This first subcomponent also contains a list of TCs and BCs on the network, which can be determined from the clock identity fields in the Announce Message. The dashboard maintains a list of current clocks that based on the PTP packets captured in real-time. The module also keeps track of when it last saw a message from one of these clocks and will remove the clock from the list of active clocks on the network if it does not see a message based on a timeout value. Between the information encoded from the Announce Messages and the clock identities, the user should be able to determine that their clock is sending Announce Messages and that the basic configurations for the clock are set up properly in preparation for subsequent tests.

Best Master Clock Algorithm (BMCA) Conformance Testing

The second test module is the BMCA, used to determine the network Grandmaster, after a clock multicasts Announce Messages upon entering the network. Since BMCA is a critical part of testing Grandmaster clocks, the front end consisted of three tables to monitor and detect issues regarding the clock attributes on multiple domains. The design consideration is based on the need to validate the BMCA selection process of a multi-profile network. The test module compares messages from two clocks in each domain based on the default BMCA priority order fields, which indicates whether a clock has a higher-ranking priority on the network. If the priority attribute is identical, the default BMCA iteratively goes through the Clock Class, Clock Accuracy, Clock Variance, Priority 2 and Source Port ID until a decision for the master clock selection is made.

The first table consists of a map of the network, where the user could see which port and domain the clocks were running on. The test module verifies a single Grandmaster clock is selected per domain within a specified amount of settling time. Configuration issues could be identified through this network map. The network map remains online, keeping track of the domains, ports, and clocks on the network. The module would detect and subsequently add the new clocks to the map, when a new clock identity emerges. The module also removes clocks after a PTP message timeout threshold of 20 s. The application provides live updates on the information about the domains, ports, and clocks on their network, while information pertaining to old clocks are removed from the network map.

The second table in the test module displays the best clocks for each domain on the network based on the default BMCA. The table, shown in Figure 21, lets the user see that their network configuration for each port is correct, and helps the user to diagnose specific problems when they have multiple clocks running on each domain.

The third and final table of the test module displays the attributes of the Grandmaster clock on each domain and highlights the field determined by the best master clock decision. The interface enables the user to see the selected best master clock on the network, and the reason why the clock is selected. All the tables in this component share a common object where the data is stored, and therefore upon a topology change of the networked clocks, the change in information is updated in all relevant tables simultaneously. The application refreshes the information about the best master clock based upon the most recent message from the clock, allowing the application and the user to immediately identify errors with the configurations.

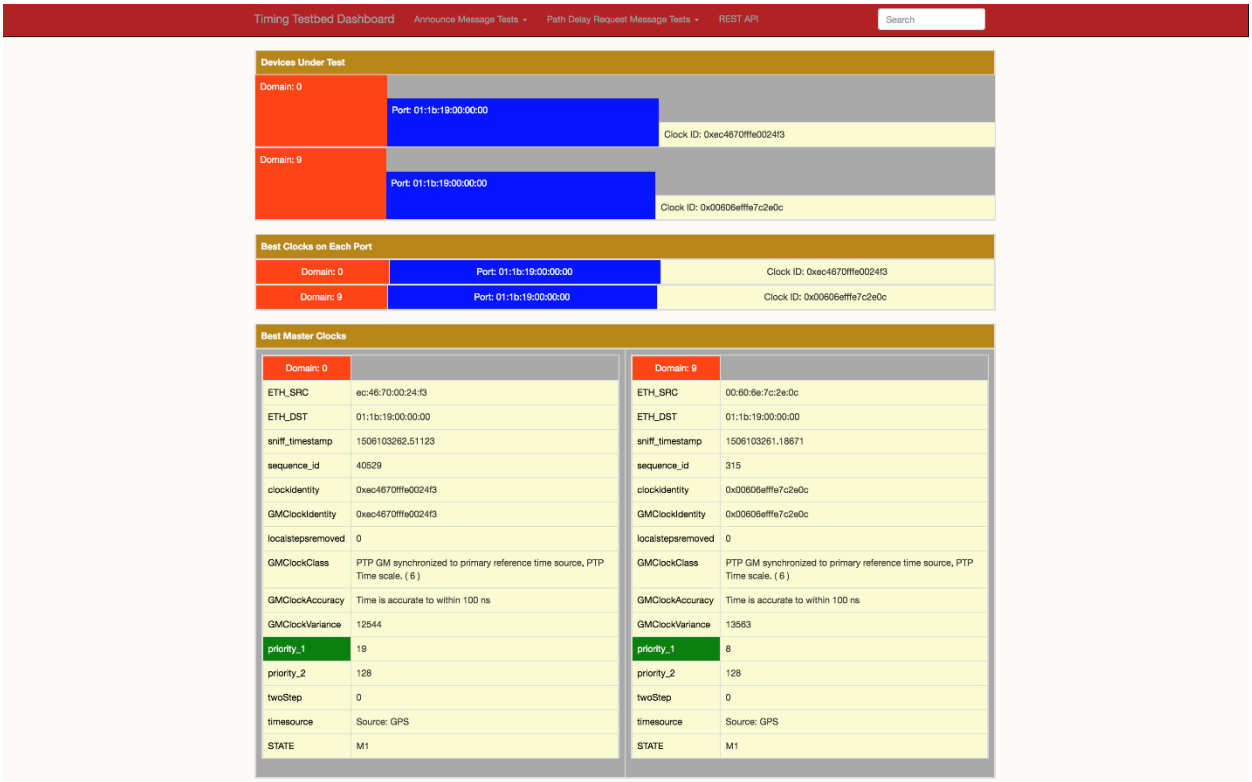

Figure 21. BMCA table supporting multiple domains.

Holdover

The test for a Grandmaster clock in holdover displays the current Grandmaster clock identity, class, accuracy, variance, time traceable, frequency traceable, and time source fields of the Announce Message to ensure compliance to the standards and test plan when it is disconnected from the GPS signal. The current clock class and accuracy are displayed in the table to allow users to estimate when a clock enters holdover mode and when it recovers, to ensure the DUTs provide the most current information on the status of their clocks. For the holdover duration estimation, the subcomponent logs when the Grandmaster clock class changes and determines the amount of time the clock stays within 250 ns based on the GM clock accuracy field. Finally, the holdover subcomponent estimates the amount of time a clock does not have a GPS signal and how much time it takes a clock to recover from holdover by keeping track of the changes in the clock class and accuracy after it has entered holdover. Users can look at this information to determine if their clock is behaving adequately when it enters holdover mode and if its behavior is compliant with the standard requirements for Grandmaster clocks.

Leap second

The leap second test module consists of a table that displays the clock identity, UTC offset, UTC reasonable, positive leap second, negative leap second, and time reference fields of the Announce Message in real-time to show whether a clock behaves correctly during a leap

second event. The positive and negative leap second flags will be set to true for about a minute before the leap second happens, and when the leap second is inserted the “action” field of the table will indicate whether a positive or negative leap second was inserted. The users can see that the Grandmaster clock reports a correct value for the UTC offset when a leap second occurs and determine whether the DUT is compliant with the standard.

ATOI

The alternative time offset indicator (ATOI) table will display the ATOI fields of the Announce Message if they are present along with the clock identity. This test users can verify the values in real-time.

Multicast MAC Address

The final test module of the dashboard includes a table to show whether the multicast MAC address in Path Delay Request Messages coming from the GM are correct, since the standard specifies a single address that needs to be set for this field. Since we are collecting Announce and Path Delay Request messages separately in the backend, we simply make the request to the REST API to access the messages in the parent component of the dashboard. The subcomponent parses the messages and verify the multicast MAC address field contains the correct value and displays the verification result along with the clock identity and destination Ethernet address. This final component is very simple but demonstrates the ability to parse and categorize multiple message types and use them independently of each other in the front-end web application.

3 IOP Test Event Summary

3.1 Test Network

Timing

The timing network topology comprised of PTP GM capable clocks, PTP-aware switches that can be configured as Transparent Clocks (TCs) or Boundary Clocks (BCs), and PTP ordinary clocks (OCs). The timing test network comprised of up to 16 TCs from 8 different vendors for the TC daisy chain test and up to 5 BCs for the BC daisy chain test.

The test and measurement suite was integrated into the network topology at multiple points. The PTP GMs used the GPS L1 signal from the GPS splitter, using the GPS emulator output as the reference source. The PPS test harness is integrated into the clock measurement and can compare 8 up to GM capable clocks simultaneously. The PTP dashboard is connected to one of the time aware switches via a secondary switch to capture the PTP packets on the network. An example of the timing interoperability test and measurement topology is shown in Figure 22.

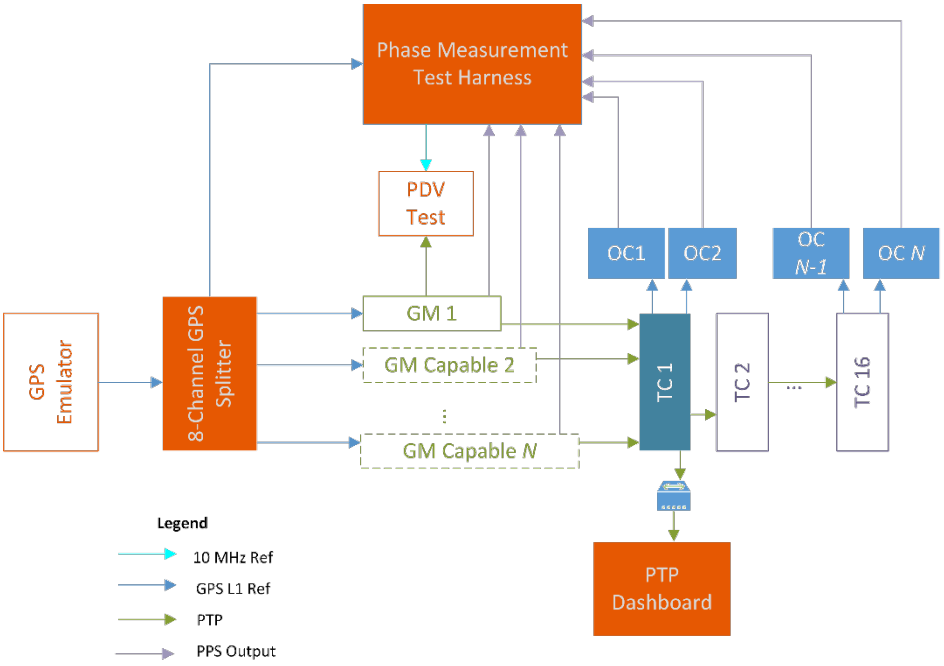

Figure 22. Timing sub-group test topology, adapted from [16].

3.2 Integrated Application Test Network

An integrated test network was put together to test the interoperability of various devices connected to substation networks. These devices included both physical and simulated MUs. The integrated test network topology is shown in Figure 23. It comprised a Control Center, Substation 1, and Substation 2 networks. The Control Center network consisted of two firewalls, two Ethernet switches, and two routers. The two Ethernet switches supported the IEC 62439-3 standard - Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR) [14]. These two protocols aimed to ensure no frame loss in the presence of an error. They also provide a hot-plugging capability that allows devices to be added to or removed from the network without interrupting communications in substation operations. The two routers supported Virtual Router Redundancy Protocol (VRRP) [15], which was a computer networking protocol that provided for automatic assignment of available Internet Protocol (IP) routers to participating hosts. This increased the availability and reliability of routing paths via automatic default gateway selections on an IP subnetwork. The Substation 2 network includes a network router and several Ethernet switches. The Substation 1 network consisted of two network buses: a Station Bus, where messages were sent to and from protection relays, and a Process Bus, where data were transmitted from the MUs to the relays. The Control Center, Substation 1, and Substation 2 networks are connected through network routers. There are five virtual local area networks (VLANs) built into the IEC 61850 integrated test network.

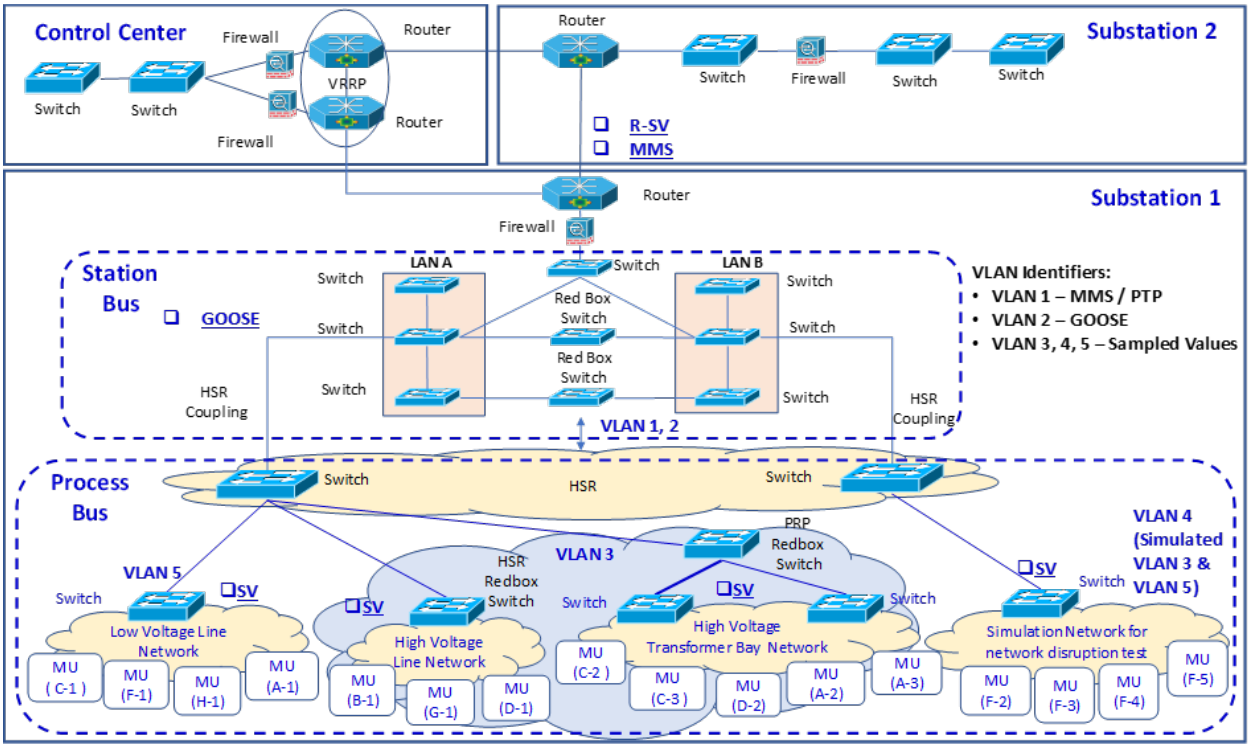

Figure 23. Integrated application test network topology. Figure Source: UCA [8].

Some members of the NIST 61850 IOP team participated in the IEC 61850 integrated application tests, which were conducted on the Substation 1 and Substation 2 networks. These team members focused on IEC 61850 interoperability including SV, GOOSE, and Manufacturing Message Specification (MMS) protocols. The Substation 1 network topology shown in Figure 23 consisted of three VLANs and two buses. A simulation network, comprising VLAN 3 and VLAN 5 and referred to as VLAN 4, was used to for a destructive network test. VLAN 3 consisted of the High Voltage Line Network and the High Voltage Transformer Bay Network. Whereas VLAN 5 contained the Low Voltage Line Network. Many physical and simulated MUs, devices to be tested for SV streams shown in Figure 1, were connected to VLAN 3, VLAN 5, and the simulation network. IEC 61850 GOOSE data for subsequent analysis were captured by NIST staff on VLAN 2 of the Station Bus. IEC 61850 MMS data were captured on VLAN 1 between two substations. IEC 61850 SV data were captured on VLAN 3 and VLAN 5. SV data from simulated MUs were also captured on the simulation network VLAN 4 (combination of VLAN 3 and VLAN 5) during the destructive network test.

3.3 Timing Tests

The test plan [16] for the IOP was drafted and reviewed based upon consensus of the Timing Sub-Group members. The tests deployed at the IOP included comprehensive testing of steady state performance, various topologies including a few hops to a maximum of 16 hops for TCs

and 3 hops for BCs upon timing disturbances such as timing discontinuities and anomalies (loss of a reference source). Agreed upon timing disturbances such as timing discontinuities and anomalies (loss of a reference source) was also executed to observe the GM response to the events whether it is holdover or the selection of a new master via the Best Master Clock Algorithm (BMCA). The timing tests were primarily done under the required or default settings. The Grandmaster and Grandmaster capable clock servers were configured to transmit PTP packets on Layer 2 Ethernet using the peer-to-peer delay mechanism.

3.3.1 Conformance Tests

The Dashboard provided the means to facilitate the verification of configuration and PTP Grandmaster parameters. During the IOP, the Dashboard application was able to discover, identify and support all the DUTs seamlessly. It was instrumental as a tool to support witnesses in determining whether a device passed the conformance tests. The tool was also used to trace the origin of the Announce messages and ensure the entire test network was able to see the correct Announce message based on the BMCA.

The application also provided support by computing the holdover time based on the reported clockAccuracy parameter, which can then be compared with the hardware signal PPS output to verify the clockAccuracy parameters are degraded in both a timely and accurate manner during loss of reference source. This information can be logged into the database to ensure test result repeatability.

The current limitations of the dashboard included the number of messages that can be stored in the local implementation and displayed to maintain the 1 Hz rate for Sync, Announce, PDelayRequest and PDelayResponse messages for about twelve devices simultaneously.

3.3.2 Timing packet delay variation

The Paragon-X was used to measure variation in the arrival rate of Sync. messages from the PTP Grandmaster over the network. In particular, it was important to note the implications of network topology changes to the jitter perceived by the slave. This is an important test since the filter designs used by slave clock manufacturers are not imposed by the 61850 standards. If the stochastic assumptions made in filter design are violated by changes in the network topology, some extensions to the standard may be required. The figures below show sample data for two instances of the experiment. In Figure 24, Sync messages were sent over a 9-hop network comprised of transparent clocks arranged in series over a radial branch. About 2 Mbps of ancillary management traffic was added to the network. Figure 25 shows the experiment with a 16-hop network, no other changes were made besides the increase in the number of hops. The data shows ~4 ns mean difference between the two cases but a significant increase in the jitter, note the 34% increase in |max- min| deviation in arrival rate.

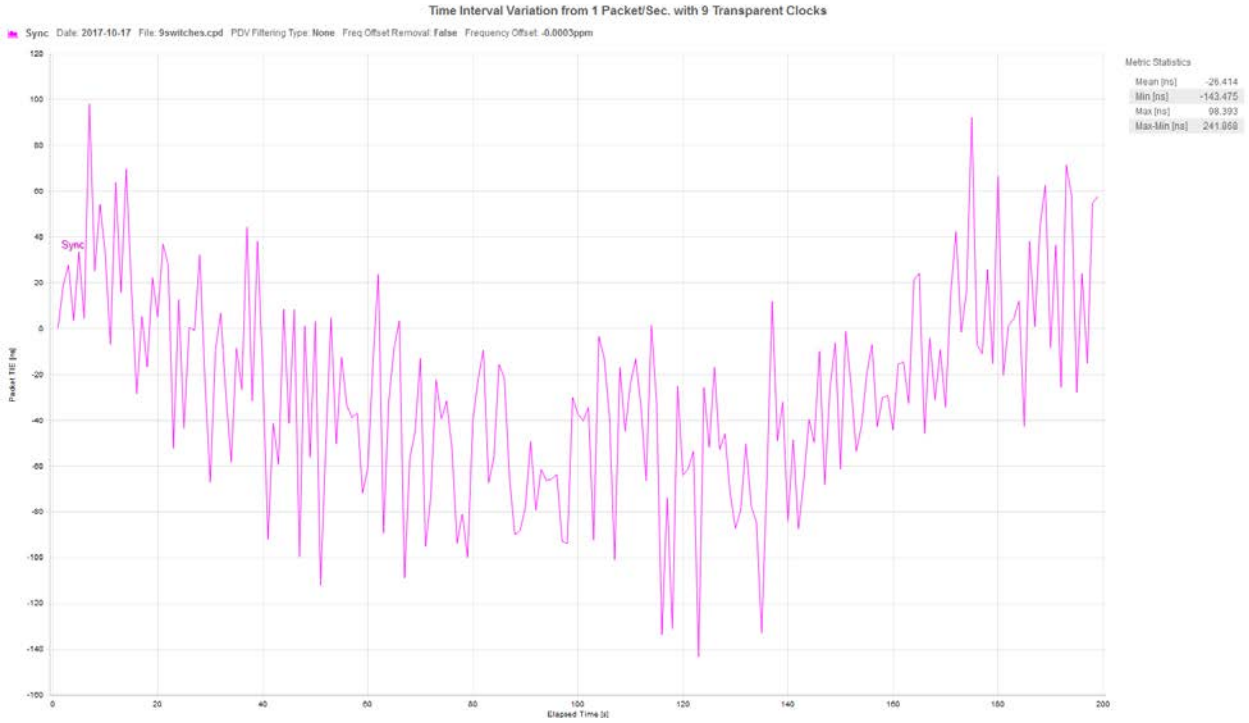

Figure 24. Time interval Variation of sync messages with 9 TCs.

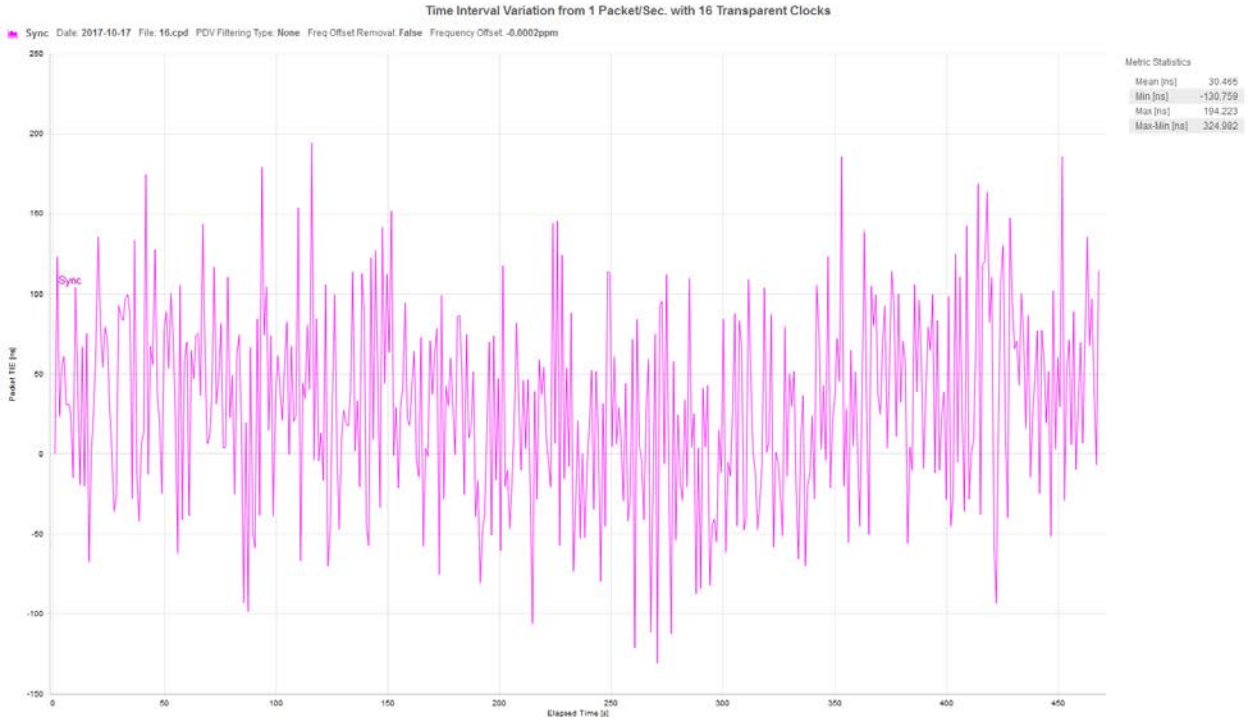

Figure 25. Sync message Time Interval variation with 16 TCs.

3.3.3 PPS variation

The PPS capture utility provided 10 ns resolution measurements of the behavior of each of the slave clocks in the network. The figures below highlight sample cases of the type of analysis this tool may be used for. In both cases, 8 slave clocks were simultaneously measured by the Time Sync subgroup. The data obtained show the offset between the slave clock and a rising edge synchronized to UTC.

Figure 26 provides a snapshot view of all the data captured in 500 s. The data provides valuable information about the stochastic variation between different clocks (inter-channel) and within samples obtained from the same clock (intra-channel) over the capture interval. This figure provides a view of the dynamic state of a synchronization network. Presumably if the variance across clocks is comparable to the variance within each clock, then a comment may be made about the baseline steady state accuracy of the timing network. Major differences between the inter-channel variance and intra-channel variance may provide guidance towards detection of malfunctioning clocks or malfunctioning network equipment.

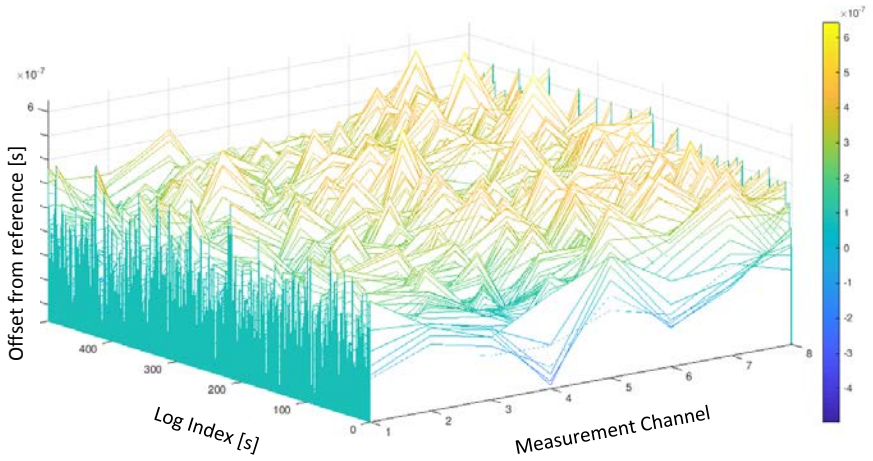

Figure 26. Slaves' PPS offset.

Figure 27 provides a visualization using the 2nd order spline fit on the data for each of the 8 channels based on a one-minute trace. The interpolated spline fit provides visual illustration of the dynamics of synchronization within each clock even when the system is nominally at steady state. We see 0.1 Hz modes in some clocks that persist even when the system appears to be in steady state. Clearly, there are significant differences in this behavior depending on the clock design and the phase lock loop (PLL) stabilization. For a system to be interoperable we expect information about the differences between devices shown in this relative manner will be vital to provide users insight about the dynamics of the PLL response for each of the devices, including relative uncertainties, in their network.

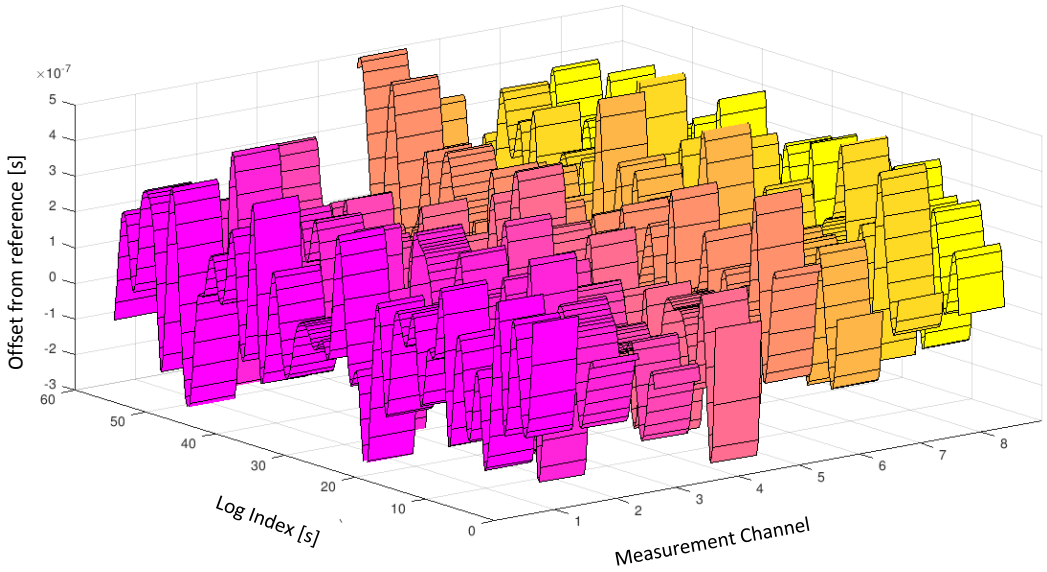

Figure 27. Variations in PLL response of PTP slaves in steady state.

3.4 Integrated application test cases

Integrated application test cases focus on testing applications based on IEC 61850-9-2 [17] and 61850-8-1[18] protocols. 23 GB of IEC 61850 SV, GOOSE, and MMS messages in the integrated application tests on the Substation 1 and Substation 2 networks were captured and analyzed for interoperability issues based on the five test cases as follows:

- Case 1: SV Test

SV streams from MUs provided by different vendors were identified and many SV packets were captured using commercial testing tools.

- Case 2: GOOSE Test

GOOSE messages from or to protection relays provided by different vendors were identified and captured using a commercial testing tool.

- Case 3: MMS Test

MMS message packets among clients and servers provided by different vendors were captured using a testing tool through a mirroring port at the station bus.

- Case 4: MMS Simulation Test

This case involved a simulated MMS client (NIST) and a vendor's server. The communications between client and server were tested and many MMS packets between the client and server were captured using a commercial testing tool.

- Case 5: R-SV test

The Routable-Sampled Values (R-SV) packets between Substation 1 and Substation 2 were captured using a testing tool.

We collected a total of 23 gigabytes (GB) of IEC 61850 packet data, which included SV, GOOSE, MMS, and R-SV packet data provided by many vendors. In this report, wefocused on Case 1: SV Test. Most commercially available MUs implement the IEC 61850-9-2LE to support a fast-track market introduction of IEC 61850-9-2-based equipment because it is a simpler implementation. Both the SV A-profile and T-profile in 61850-9-2 are mandatory. IEC 61850-9-2, a subset of IEC 61850-9-2 specification[17], supports only one service, SendMSVMessage [18]. Figure 2 shows the interoperability test system for the IEC 61850-9-2-based MUs. The system consists of a MU tester, two DUTs, a network switch, a network sniffer, and a MU interoperability analyzer. The MU tester is SVScout**[19], a commercially available SV testing tool, which supports the IEC 61850-9-2 protocol. The MU tester can identify SV streams and subscribe to SV streams from DUTs (MUs) through a network switch. The network sniffer, Wireshark**[20], is a free, open-source network packet analyzer, which can capture all network packets. Wireshark captures all SV packets from the DUTs and saves them in the packet capture (PCAP) format, while SVScout can identify, subscribe, and record all SV packets in the ".enc" format that is defined by the vendor and is also compatible with the PCAP format. The SV packets recorded by SVScout and captured by Wireshark can be converted into the Packet Detail Markup Language (PDML) format. The MU interoperability analyzer can be either a human operator or an application software, which performs SV message analysis of the captured IEC 61850-9-2 data packets to determine the degree of MU conformance to standards and interoperability.

Interoperability tests of many commercial and simulated MUs were conducted using the SVScout and Wireshark testing tools. These MUs are deployed in four network locations, such as the simulation network (VLAN 4), the Low Voltage (LV) network (VLAN 5), and two High Voltage (HV) networks (VLAN 3). The Interoperability test setup was shown in Figure 28. Figure 29 shows a screenshot of voltage and current waveforms created by the SVScout testing tool based on packet data from a commercial MU.

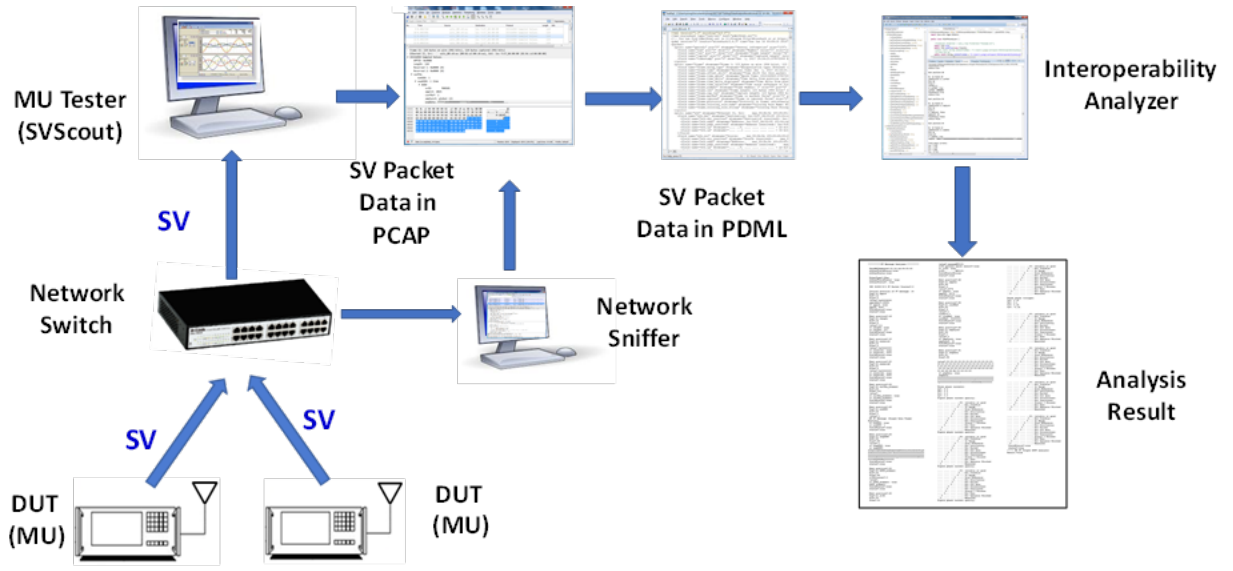

Figure 28. Interoperability Test Method of IEC 61850-9-2 based MUs.

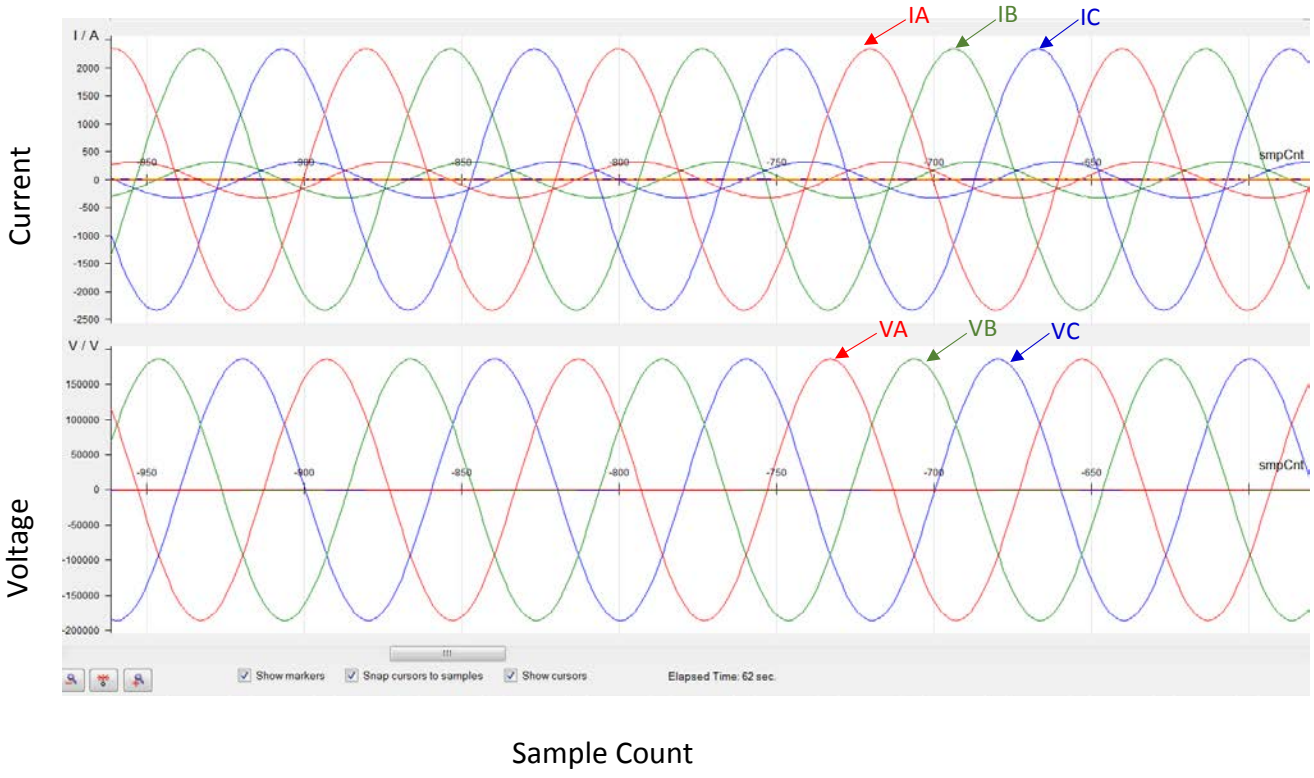

Figure 29. Test tool visualization of three-phase voltages and currents based on packet data from a MU implementation at the interoperability test event.

SV Analysis Tool: As shown in Figure 30, NIST has developed an interoperability analysis software tool for IEC 61850-9-2 based MUs. IEC 61850-9-2 provides the "tag length value (TLV)"

specifications of 61850-9-2 SV message. This software tool can automatically read and parse IEC 61850-9-2 packet data from a PDML file in XML format. It automatically analyzes the TLV of each field of the SV packet data prescribed by the IEC 61850-9-2 specification. The three-phase current and voltage data are scaled using the SV scale factor data, and they are output to a data file for future use, such as graphing waveforms as shown in Figure 31 to visually assess the data quality of SV message.

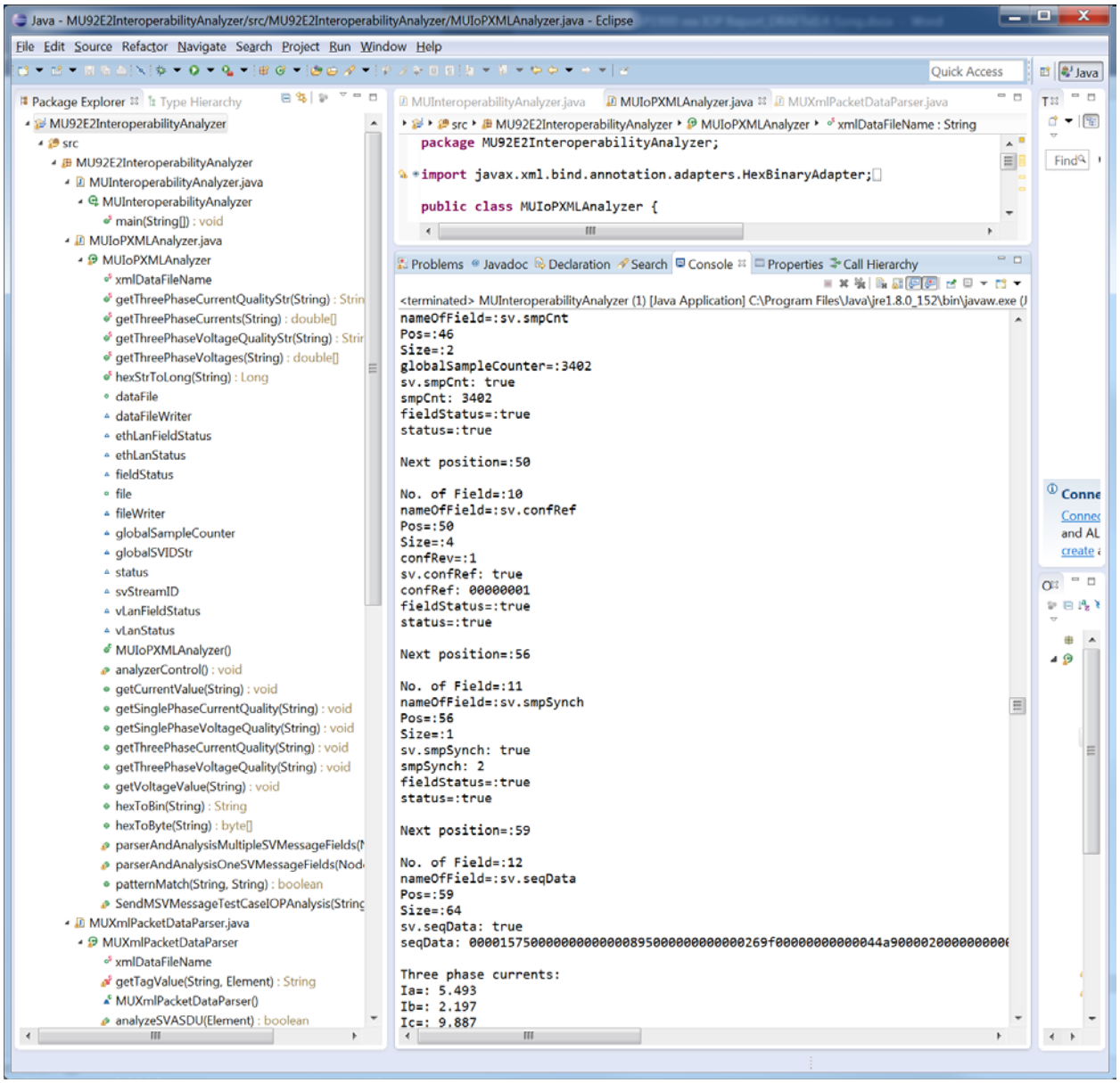

Figure 30. Screenshot showing some results of MU interoperability analyzer.

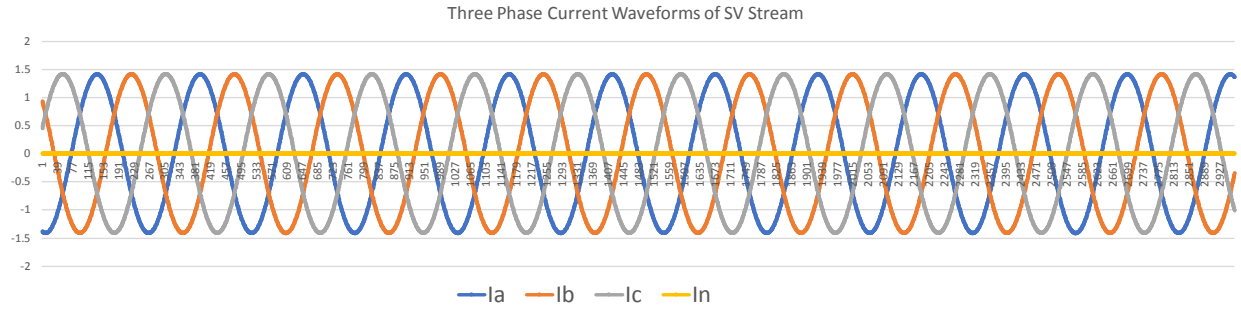

Figure 31. Three-phase current waveforms.

According to the agreement among IOP participants, all SV publishers should adhere to IEC 61850-9-2. Based on our analysis of some SV packet data adhering to that requirement, we would like to point out some potential interoperability issues. Firstly, the destination MAC addresses in some SV streams were outside the range specified in the IEC 61850-9-2 specification, which should range from 01-0C-CD-04-00-00 to 01-0C-CD-04-FF-FF. Secondly, as specified in the IEC 61850-9-2 specification, the SV identification (svID) should be a system-wide unique identification, but there is no method offered in the standard for this. Thus, there is a chance that two svIDs in a system from different vendors could be the same, which might cause an interoperability problem. In general, these problems can easily be fixed with a firmware upgrade by the vendors to support interoperability. There are two editions of IEC 61850-9-2 (61850-9-2:2004 and 61850-9-2:2011), but there is no edition information in the SV message. Therefore, we recommended that the version number (a single digit, e.g., 1 or 2) should be included in the svID field of the SV message to easily identify which version is implemented. In this way, one will be able to decode the SV message easily and correctly.

4 Discussion and Future

A key challenge in the interoperability testing event is the ability to perform real-time conformance verification without full prior knowledge of the scale, traffic saturation and types of equipment and equipment implementation variability. The Dashboard application was able to detect all the devices on the network automatically. However, the application can handle a limited number of packets in the database. The refresh rate of the real-time updates was therefore limited and the ability to retain information for post-event analysis was also limited.

For the Integrated Applications testing, traffic data rates for run-time monitoring and testing was a key challenge. The integrated applications testing included 360 devices generating SVs at 15,360 samples per second. The packets were captured for offline analysis.

Timing

Distributed Dashboard

The overall vision of the timing test harness is to create a distributed online time monitoring system integrating both hardware-based capabilities using physical timing signals and software-based packet analysis. The dashboard will be server-based to provide platform independence, a level of redundancy and scalability in terms of both the number of devices and clock behavior information from the packets over time. The server-based model would allow for monitoring of multiple locations to enable applications utilizing time-stamped measurements to have potential access to dynamic time error information as the network or system device would allow. With the trends towards industrial IoT devices as part of the power systems monitoring network, the feature would be beneficial to evaluate and weight the measurement and timestamp of the remote devices, based on application timing uncertainty requirements.

Scaling

One of the challenges of the event was to scale to the number of devices, while providing real-time updates equivalent to the rate of the DUTs' incoming packets. Incorporating Redux, as shown in Figure 32, into the Dashboard will help manage the state of the application to improve performance. Further improvements can be realized by modifying the REST API to send only the necessary information to the front end. The application has since been updated using Redux to manage the state of the application and to provide updates only to needed information. Improved software integration creates a communal "store" for the data where all the components can reference the same data in memory. Reduced resource usage through shared information allows the web application to be much lighter and alleviates communications, computing, and storage resources.

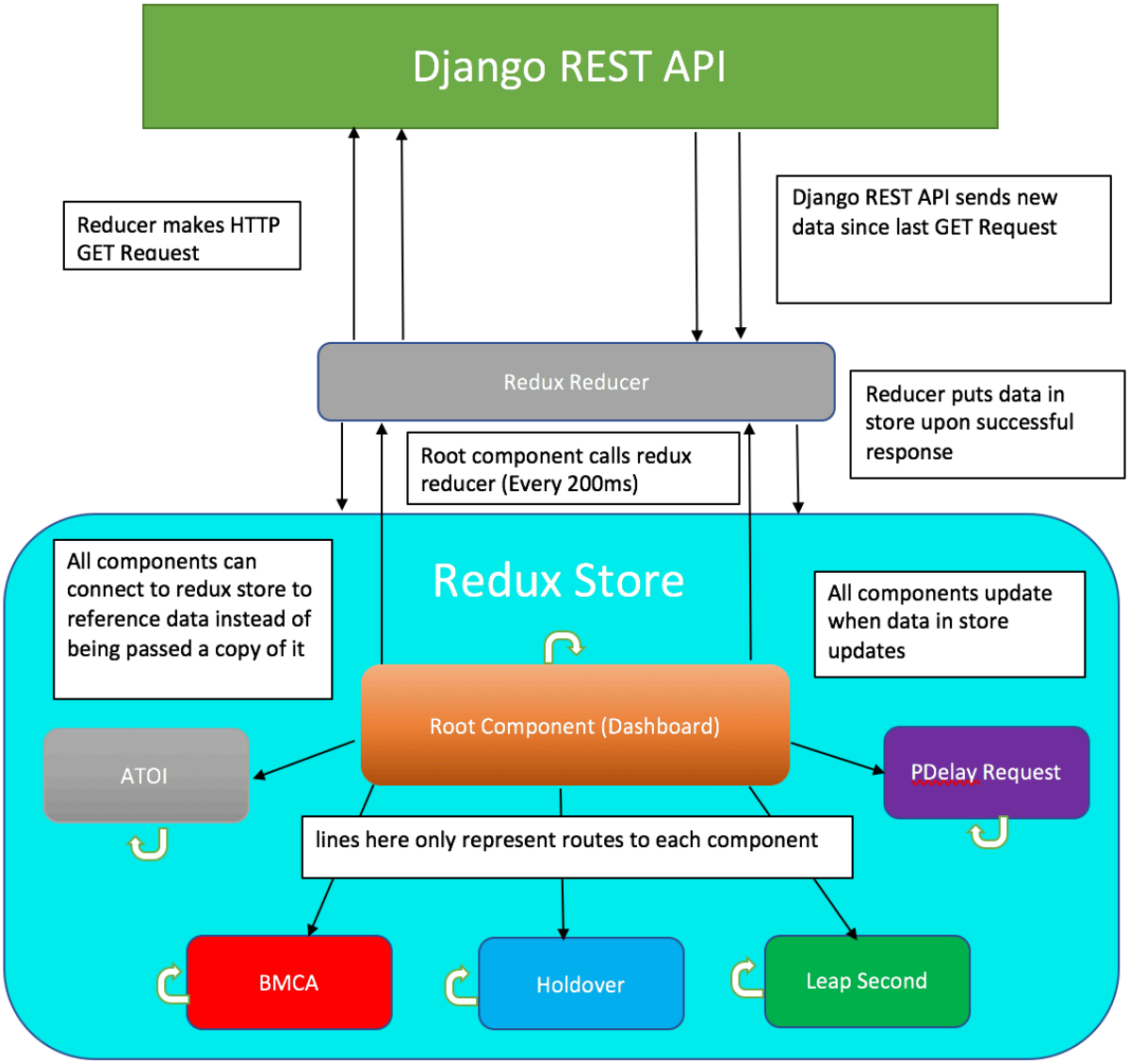

Figure 32. Integrating Redux.

Test and Measurement Device Integration

Additional features can also be added to monitor the behavior of the overall accuracy and precision of the synchronization system, by integrating the test and measurement devices such as the GPS Simulator, network test equipment, and oscilloscope into the test harness. With the trend toward Optical substations, the test harness would also need to support optical–based output signal testing including measurements of time error between clocks using 1 PPS, as well as PDV, PD of one-way PTP Messages. Specific device firmware functionalities such as watchdog implementation, aspirations for bug-free code such that the system never requires a power cycle

are all desired functionalities but require more sophisticated software assurance testing capabilities.

Future metrics to be integrated into the dashboard include:

- PDV as a function of network traffic saturation to improve testing of the network nodes.
- Software assurance metrics to determine system robustness to security or disruptive networks.
- Resiliency metrics, such as duration of link loss, time interval for link recovery, and continuous hardware-based time error monitoring during the link loss and link recovery events to capture results of HSR and PRP testing.

Timing Signal Impact Analysis

A key capability is to be able to assess the impact of timing signal perturbations on power systems voltage and frequency regulation with the increased use of highly variable generation from renewable sources. The prototype analysis tool would enable observation at both the DUT level as well as the system performance to assess the interoperability of devices and system level performance based upon the ability to extrapolate the desired decisions and behavior of the system.

Integrated

The prototype interoperability test tool developed for the IOP message analysis is limited to IEC 61850 SV messages. In the future, we plan to extend the prototype tool to a comprehensive IEC 61850 interoperability test tool encompassing GOOSE, MMS, R-SV, and R-GOOSE as an integrated test suite. The test tool can also be used to evaluate how well GOOSE messages can be integrated with horizontal (cross-domain) communication using publish-subscribe messaging framework, such as the Open Field Message Bus (OpenFMB) standard.

There are also opportunities to extend the HIL capabilities to support the simulation of a reference substation to be used for the interoperability testing. The reference substation would include simulated controllers, such as relays, where the PLLs can be tied to the reference or purposefully perturbed to observe DUT behavior during system timing and communications anomalies. The simulation capability would enable testing of novel substation configurations and capabilities, by entering theoretical models of controller components, such as solid-state functionality for medium voltage regulation.

Conclusion

Through the IOP participation, the Cyber-Physical System and Smart Grid testbeds at NIST can be designed and deployed to better meet industry needs. The testbeds are intended to be used to develop and validate prototype measurement and test tools to support industry interoperability and standards development. The key areas of the testbed include wide-area sensing (PMUs, MUs, wire-line monitoring sensors, smart meters, current and voltage transformers) and control (relays), which require a base information infrastructure for timing, communications, and cybersecurity to support the dynamic and critical nature of power systems.

Since the inception of the testbed, the NIST Smart Grid team has developed a portable hardware and software test and measurement harness capable of providing reference electrical and timing signals to multiple DUTs simultaneously. Real-time monitoring of time offsets enables the monitoring of how anomalous events, such as leap second changes and network link interruptions, impact the clock signals of the DUTs at both master and slave clocks. A web-based PTP monitoring system enables access to real-time monitoring of PTP packet parameters. This ensures changes in the best master clock, clock class and leap second flags are updated in a timely manner for continuous propagation of accurate and precise time information, along with accurate understanding of the time quality for time-sensitive end applications. Offline analysis of IEC 61850 SV messages ensures data are communicated seamlessly among sensors and controllers to ensure new devices are conformant to the IEC 61850 standards, can be readily integrated and applications have access to accurate and timely measurements and notifications.

The development of the tool suite also sets the foundation for the testbed and Smart Grid program to continue to support industry standards development as well as test and certification efforts for the IEEE/IEC 61850-9-3 (2016), IEEE C37.238-2017 PTP Power Profiles, and IEC 61850-9-2 communication standards.

Appendix A.

References

[1] IEEE Synchrophasor Measurement Test Suite Specification—Version 2, pp.1-43, 28 September 2015.

[2] IEEE Standard for a precision clock synchronization protocol for networked measurement and control systems," in *IEC 61588:2009(E)*, pp.C1-274, 2 Feb. 2009 doi: 10.1109/IEEESTD.2009.4839002.

[3] IEC/IEEE International Standard - Communication networks and systems for power utility automation Part 9-3: Precision time protocol profile for power utility automation," in

IEC/IEEE 61850-9-3 Edition 1.0 2016-05, pp.1-18, 31 May 2016. doi:10.1109/IEEESTD.2016.7479438

[4] IEEE Standard Profile for Use of IEEE 1588 Precision Time Protocol in Power System Applications," in *IEEE Std C37.238-2017 (Revision of IEEE Std C37.238-2011)*, pp.1-42, 19 June 2017. doi: 10.1109/IEEESTD.2017.7953616

[5] IEEE Standard Profile for Use of IEEE 1588 Precision Time Protocol in Power System Applications," in *IEEE Std C37.238-2011*, pp.1-66, 14 July 2011 doi: 10.1109/IEEESTD.2011.5963699

[6] Perkins, C., Laird, J., McEachern, R., Noseworthy, B., Amelot, J., Li-Baboud, Y.S. and Brady, K., 2014. *1588 Power Profile Test Plan*. US Department of Commerce, National Institute of Standards and Technology.

[7] Noseworthy, B., et. al. *Draft 1588 Power Profile Test Suite Specification.* 2018.

[8] UCAIUG 2017 IEC 61850 IOP Report, May 2018, http://www.ucaiug.org/IOP_Registration/IOP%20Reports/UCA%20IOP%202017%20%20Final%20Report%2020180529.pdf.

[9] IRIG Standard 200-04. IRIG Serial Time Formats, September 2004. http://irigb.com/pdf/wp-irig-200-04.pdf

[10] MAX2659 GPS/GNSS Low Noise Amplifier, May 2018. https://datasheets.maximintegrated.com/en/ds/MAX2659.pdf

[11] Django, Available online, https://www.djangoproject.com/, accessed June 20, 2018.

[12] PyShark. Available online, https://pypi.org/project/pyshark/, accessed June 20, 2018.

[13] ReactJS, Available online, https://reactjs.org/, accessed June 20, 2018.

[14] IEC 62439-3:2016-Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR).

[15] IETF RFC 3768: Virtual Router Redundancy Protocol (VRRP), https://tools.ietf.org/html/rfc3768.

[16] IEC 61850 IOP Use Cases and Test Cases for IEEE 1588 using IEC/IEEE 61850-9-3:2016 and IEEE C37.238:2017. http://iec61850.ucaiug.org/2017IOP-NOrleans/IOP%20Documents/TimeSync_Subgroup/2017_Test_Cases/IOP%20PTP%20Test%20Cases%20for%209-3%20and%2037.238_V1.0.docx, (Accessed April 2018).

[17] IEC 61850-9-2:2011, Communication networks and systems in substations – Part 9-2: Specific Communication Service Mapping (SCSM) – Sampled values over ISO/IEC 8802-3.

[18] IEC 61850-8-1:2011, Communication networks and systems in substations – Part 8-1: Specific Communication Service Mapping (SCSM) –Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3.

[19] SVScout. Avalable online, https://www.omicronenergy.com/en/products/svscout/.

[20] Wireshark. Available online, https://www.wireshark.org/.

A.1 Acronyms

TABLE 1 ACRONYM DEFINITIONS

Acronym	Expansion
GNSS	Global navigation satellite system
GPS	Global positioning system
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IRIG-B	Inter-Range Instrumentation Group Time Code B
ISO	International Organization for Standardization
MAC	Media Access Control
MACsec	Media Access Control Security
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
PDV	Packet delay variation
PTP	Precision Time Protocol
REST	Representational State Transfer
RF	Radio frequency
SCADA	Supervisory Control and Data Acquisition
SNMP	Simple Network Management Protocol
SW	Software
TAI	International Atomic Time (Temps Atomique International)
TCP	Transmission Control Protocol
UTC	Coordinated Universal Time