Practical Implementation of a Security-Dependability Adaptive Voting Scheme Using Decision Trees

Ryan D. Quint

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Virgilio A. Centeno, Chair

Jaime de la Ree Lopez

Robert P. Broadwater

November 11, 2011

Blacksburg, Virginia

Keywords: adaptive protection, decision trees, phasor measurement unit,
phasor data concentrator, wide area measurements

Practical Implementation of a Security-Dependability Adaptive Voting Scheme Using Decision Trees

Ryan D. Quint

Abstract

Today's electric power system is operated under increasingly stressed conditions. As electrical demand increases, the existing grid is operated closer to its stable operating limits while maintaining high reliability of electric power delivery to its customers. Protective schemes are designed to account for pressures towards unstable operation, but there is always a tradeoff between security and dependability of this protection.

Adaptive relaying schemes that can change or modify their operation based on prevailing system conditions are an example of a protective scheme increasing reliability of the power system. The purpose of this thesis is to validate and analyze implementation of the Security-Dependability Adaptive Voting Scheme. It is demonstrated that this scheme can be implemented with a select few Phasor Measurement Units (PMUs) reporting positive sequence currents to a Phasor Data Concentrator (PDC). At the PDC, the state of the power system is defined as Stressed or Safe and a set of relays either vote or perform normal operation, respectively.

The Adaptive Voting Scheme was implemented using two configurations: hardware- and software-based PDC solutions. Each was shown to be functional, effective, and practical for implementation. Practicality was based on the latency of Wide Area Measurement (WAM) devices and the added latency of relay voting operation during Stressed conditions. Phasor Measurement Units (PMUs), Phasor Data Concentrators (PDCs), and relay operation delays were quantified to determine the benefits and limitations of WAMS protection and implementation of the voting scheme. It is proposed that the delays injected into the existing protection schemes would have minimal effect on the voting scheme but must be accounted for when implementing power system controls due to the real-time requirements of the data.

To My Grandfather, Dr. Sidney C. Jackson

Acknowledgments

I would like to thank my academic advisor, Dr. Virgilio Centeno, for his perpetual involvement in my research. Both inside and outside of the research realm, Dr. Centeno has been an invaluable mentor and source of information.

Dr. Jaime de la Ree also provided countless hours of useful discussion and advice towards my work, as well as incomparable motivation and inspiration towards extending my studies. The teachings and interest of Dr. Robert Broadwater towards my research is also appreciated.

Mr. Anthony Faris and Mr. Greg Stults from the Bonneville Power Administration provided laboratory equipment, experience, and expertise on the majority of my research; this work would not have been possible without their help and devotion.

My friends and family, from Washington State all the way to Virginia, have been an inspiration during my graduate studies thus far. Thank you all.

And lastly, I wish to thank my Scotti Rose and Tesla Monster. I only dream of being as supportive, loving, and distracting as you two have been throughout this journey; I can't thank you enough.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Today's electric power grid has evolved into a highly reliable set of interconnected network, system, and devices. Due to simple brilliance, sources such as coal, gas, wind, and even solar drive a system of electric machines to power a variety of loads. Whether these loads are light bulbs in the remotest of towns or a silicon wafer factory deep in a metropolis, utilities seek the highest reliability possible for their customers. It takes an enormous investment to ensure that power will be delivered when required, and it is the focus of this paper as to means of reducing this investment while maintaining that reliability.

A wide variety of systems and schemes exist in the electric power industry today to detect problematic conditions and to eradicate any deviation from normal operation. From localized distance relaying schemes to under voltage load shedding schemes, the size and complexity of these systems are expanding every day [1,2]. The use of newer technology combined with existing practices, is an effective means of evolving the means in which we protect the power grid and thus the customers it serves.

The advent of the Phasor Measurement Unit (PMU) allowed the power grid to be viewed in an entirely new light due to the access of phasors measurements synchronized to a common time source it provided. Since its inception in the late 1980's and early 1990's, the PMU has proved to be an effective application for obtaining a Wide-Area perspective on the power system. Following the growth of synchronized phasor measurements, often called synchrophasors, a need for effectively collecting these measurements was quickly noticed. There began the advancement of the Phasor Data Concentrator (PDC), to aggregate and time align synchrophasor data.

In 1995, the electric utility industry was deregulated nationwide by Federal law. Deregulation laws enacted required a separation of generation and transmission services, and also required an open access to power marketers. It also sought to shift from vertically integrated utilities to ones in which generation, transmission, and distribution of power was separated [3]. The evolution of deregulation engulfed the industry, requiring a paradigm shift in the way the grid was operated. Since each entity interacting with the grid required less of a wide-area perspective, the need for technology such as synchrophasors quickly died down for a number of years.

Regardless of the industry, research efforts persisted to develop a means of monitoring the power system in a time synchronized manner. Companies such as the Bonneville Power Administration, working together with universities and businesses such as Virginia Tech and Macrodyne, installed a number of PMUs that still exist today. More importantly, a push for Wide-Area Measurement Systems (WAMS) nationwide has driven initiatives such as the North American Synchrophasor Initiative (NASPI) and Western Interconnection Synchrophasor Project (WISP). Utilities under the direction of these initiatives and projects are actively installing hundreds of PMUs to meet WAMS demands [4-6].

The shift from PMUs as a research oriented technology to a utility-integrated application is an important aspect of the work presented in this thesis. The purpose of the work performed was to demonstrated the use of a research topic, and explore valid questions that the utility industry may have when it comes to phasor measurement technology. Deregulation drove PMUs into an over-researched, under-implemented technology for a number of years, but that perspective is rapidly changing. It was intended that past research performed would be demonstrated as a viable option for field implementation, rather than simply a theory. It was also intended that those possible questions posed by the utilities be explored and some light shed on a somewhat gray area of the technology.

This chapter will explore the different components of phasor measurement technology in more detail and introduce the work performed. Chapter 2 elaborates on the theory behind the adaptive protection scheme implemented. Chapter 3 describes data latency throughout a wide-area measurement and control system (WAMCS), and its effects on making effective control actions. Chapter 4 details the laboratory implementation of the Security/Dependability Adaptive Protection Scheme using phasor measurement technology in conjunction with decision trees. Chapter 5 describes the testing procedure and results for determining latency of WAMS components in a laboratory setting. Finally, Chapter 6 presents conclusions drawn from the research performed and possible future work.

1.1 Synchronized Phasor Measurement Technology

Since the majority of the work for this thesis dives into the inner-workings of the various WAMS devices, it is important to present the functionality and importance of each device in the network. As mentioned, synchronized phasor measurement technology has been around for a number of years, yet lacks physical implementation of various applications and even testing of key aspects of the devices.

Although well developed, WAMS technology is primarily driven by the applications requiring time-aligned, high-speed measurement of the power system. A diverse array of applications currently exist, from power system control requiring real-time data to offline analysis such as model validation requiring highly accurate measurements. This thesis analyzes most concepts of a Wide-Area Measurement and Control System (WAMCS), from measurement at the PMU to control action at the relay controlling a breaker.

1.1.1 Phasor Measurement Unit Functionality

Synchronized phasor measurement provides a means of real-time sampling of power system quantities time-synchronized to a universal clock system. The analog power system phase quantities from the secondary side of the instrument transformers, such as current and voltage transformers, are sampled through analog-digital (A/D) converters to provide positive sequence (as well as negative and zero if required) voltage and current samples. The PMU collects a window of samples to calculate a phasor representation of the input signal. Algorithms such as the recursive phasor algorithm are used for making an accurate phasor representation of the system values [7].

Time synchronization is typically attained through a satellite broadcast system. Possible systems include the Geostationary Operational Environmental Satellite (GOES), Global Positioning System (GPS), and a Russian-based radionavigation system, GLONASS. At least for North American utilizes, GPS has been the primary venue for acquiring a universal time. GPS is a free service provided by the United States Department of Defense for civilian use, and provides an accurate location and time. Timing of the GPS 1 pulse-per-second (1 PPS) signal is accurate to less than 1 microsecond [8]. Following A/D conversion and time synchronization, the phasor data is converted to a standard protocol C37.118-2005, Standard for Synchrophasors for Power Systems. The output from the PMU in this standard format commonly lands on a substation local-area network (SS LAN). C37.118-2005 calls for PMUs to report phasors at rates up to 60 samples per second, once reported phasor per 60Hz cycle.

The basic operation of a PMU is standardized using the C37.118-2005 standard, while the actual design and implementation is open to new advancement. The new standard C37.118-2011 has not been release, but is expected to set stricter requirements on the accuracy of PMU measurements, and include dynamic conditions when considering errors. Figure 1.1 shows a block diagram of the basic operational components of a PMU [9,10].

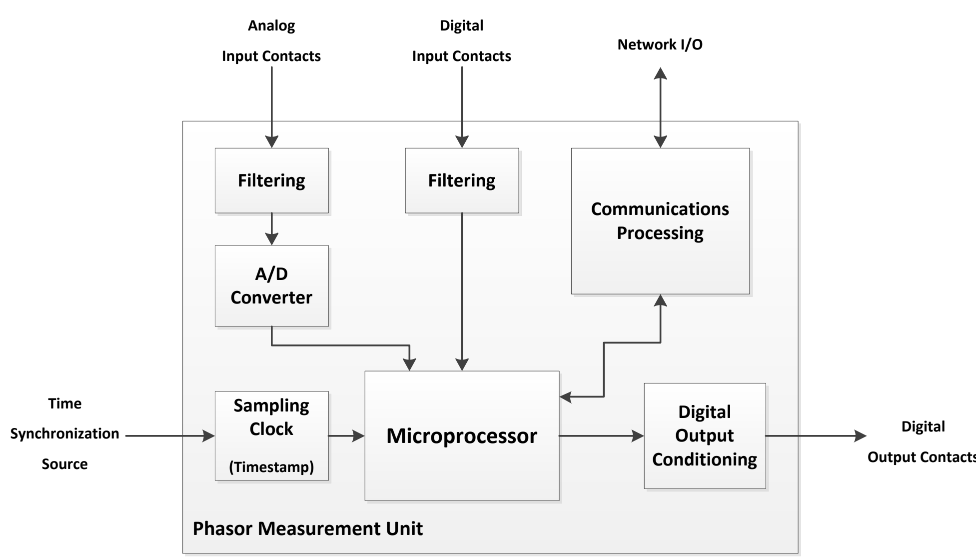


Figure 1.1: Phasor Measurement Unit Architecture

1.1.2 Wide-Area Network Functionality

The transmission of phasor measurement data packets from remote substation or control center applications to a central location for aggregation and time-alignment is the purpose of the wide-area network (WAN). The WAN is often defined as the link connecting remote substation networks (SS LAN) to each other and to the central data network often situated at the control center (CC LAN). The communications channels of this subsystem are connected via routers, switches, multiplexers, and a variety of other interfacing devices between the substation equipment and control center equipment. Each component adds to the complexity of this subsystem of the WAMS due to the different technologies used and the designs implemented. Figure 1.2 provides a visualization of the WAN, and its connection to the rest of the WAMS.

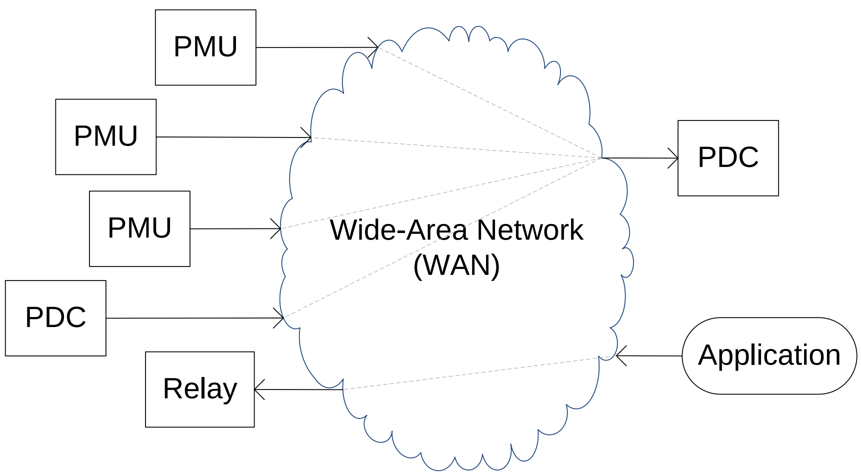


Figure 1.2: Wide Area Network Phasor Data Flow

Primary concern for the WAN is reliability and latency of the information from source to destination. These variables are directly affected by the communications medium, bandwidth, protocols, and data protection employed. Phasor data streaming from the PMU onto the SS LAN is usually converted to microwave or fiber optic communication for wide-area transmission. Ethernet protocols such as UDP or TCP are most commonly used, and each communications protocol has benefits and limitations associated with it. The factors contributing to the WAN complexity are discussed in more detail later in this paper, but actual implementation and testing of a subsystem that could be hundreds or thousands of miles apart was obviously a challenge to test in a lab.

1.1.3 Phasor Data Concentrator Functionality

Phasor measurements taken at the PMUs are time stamped by the PMU and sent to a central location. Measurements from across a system can be time stamped using the same time reference and these measurements can be aligned such that a snapshot of the state of the system can be made. A device is required to aggregate the measurements from these remote sites and align their time stamps; this occurs at the phasor data concentrator. (PDC). Ideally, the C37.118-2005 packets with the same time stamp sent by the different PMUs would arrive at the PDC at the same instant; however, this is rarely, if ever, the case. PMU data sent over the WAN usually will not arrive at the PDC at the same instant, and some measurements from a particular location or device may arrive many milliseconds after measurements from other devices with more recent time stamps. This is primarily due to delays in the communications network and physical distance the measurements must travel. The aggregation of PMU data simply involves collecting the measurements from each PMU in the system, while time alignment is also necessary for applications requiring the data because the PMU measurements must have the same time stamp as the others. PDCs require a buffer to store incoming PMU data packets while waiting for other measurements with the same time stamp. Output capability of the PDC uses the same C37.118-2005 protocol, so that PDC outputs can be sent to other PDCs for further aggregation at a higher level in the network. This may include aggregating multiple utilities' PMU measurements at a Regional Entity such as the Western Electricity Coordinating Council (WECC).

The output function of the PDC typically attempts to collect all PMU or PDC measurements as quickly as possible, validate those measurements, and output a C37.118-2005 data packet. Each PDC operates differently based on the design of the manufacturer, but common examples are explained below. A PDC may have a Wait Time threshold setting, where the PDC will output the PMU data when all measurements have arrived, or wait for a preset Wait Time before outputting the available phasors. It is up to the system designer to ensure that this Wait Time is longer than the longest PMU delay, but yet shorter than any specifications set by the applications requiring the data [11]. Another option may be that the PDC waits for a given time period before resuming normal operation, omitting the missing PMU measurements from the stream. This design seeks to maintain the "real-time" functionality of the PDC while relaxing the requirement of all PMU measurements arriving.

1.2 Power System Reliability: Security and Dependability

Of significant interest in the work presented was the improvement of power system reliability using an adaptive protection scheme which employs the use of wide-area measurement with data mining techniques. Therefore it is important to define reliability as it pertains to the power system.

Reliability of power system protection is often broken into two definitions: security and reliability. Security is defined as *the measure of a device's certainty not to operate falsely; that is, the relay or protection system should not operate for a fault in which it is not designed to operate for*. Dependability is the converse component of reliability and is defined as *the measure of a device's certainty to operate when required; if a condition occurs such that operation of a relay or protection system is required, the appropriate devices will operate* [1]. The setting of dependability and its adverse effect on security of the protection system is inherent in its design. Increasing the dependability of the system (certainty of operation) decreases the security of the system (certainty of no misoperation).

Protection systems are typically biased towards dependability due to the ability of the power system to accommodate a potential misoperation. During normal operation, a fault or disturbance on the transmission system will cause a temporary, or possibly permanent, line outage but the power can usually be rerouted through parallel or looped paths. Stressed system conditions may not exhibit this behavior. As it becomes increasingly harder to construct generating units near load centers and permit for expanding the transmission system, demand continues to grow. This is pushing the bulk electric system towards its loading limits, and a need for system security becomes eminent. During these stressed system conditions, a relay misoperation could exacerbate the problem and lead to a cascading blackout event. The use of protection schemes such as the adaptive protection scheme described in this thesis is an effective and cost-efficient means of improving the reliability of the power system.

1.3 Adaptive Protection using Wide Area Measurement

Devices such as relays used for protecting the power system are traditionally set based on system studies, simulations, and engineering judgment. For example, distance protection settings are set based on the apparent impedance seen by the relay for different system conditions. Out-of-step protection settings use different operating cases to determine the best allowable settings within the R-X impedance plane for setting protection boundaries [1]. The definition of adaptive, as defined by the Encarta World English dictionary, is "*useable in different conditions: able to be adjusted for use in different conditions*." Older protective devices such as mechanical relays cannot alter their settings, but the advent of microcomputer-based relays has allowed power system protection to embody this concept. These devices can change their settings, operation, or logic to adapt to prevailing system conditions. *The ability to modify, update, or change the settings of the protection scheme is defined as adaptive relaying* [12].

Prior to the concept of wide-area measurement, protective devices used local measurements such as voltage and current to determine the state of the nearby system based on the current settings applied to the device. This idea is being challenged by engineers now that synchronized measurements across the system are realizable. Phasor measurements collected by PMUs and aggregated by PDCs can provide a system-wide analysis of the state of the system. Research proposed by Virginia Tech has shown that data mining techniques such as decision trees are a viable option for determining the state of the system in real-time. This concept will be discussed in greater detail later in this thesis.

The backbone of this concept lies with the ability to take measurements of the power system with a wide-area perspective. Without synchronization of key measurements across the system, or a portion of the system, an analysis and action based on system conditions is not available. The real-time capability (accommodating for

WAMS latency) and synchronized phasor measurement create an entirely new means of taking protective action on the system based on state determination.

The proposed methodology mentioned above to determine the state of the power system is one concept of adaptive protection. The end use of this state determination is potentially endless and requires further exploration. One application of this signal is for arming a protection system voting scheme. Relay voting is not a new concept, and has been used to increase the reliability of protection systems. Voting can be applied at the device level or protection scheme level. For example, a line may be protected by multiple or redundant relays controlling a breaker. Normally, the functionality of this system would allow any one relay to trip the breaker; essentially, an ‘or’ function is used to determine which relay to permit tripping. This concept is even more useful when multiple protection schemes are protecting a common device or line. An example of this is bus protection, where a differential protection, phasor-based differential protection, and directional comparison scheme each protect the bus. When stressed, an ‘and’ function or majority voting scheme may be deployed to increase security. [13]. One of the main focuses of the work performed for this thesis is to apply the concept of adaptive protection to connect the decision tree state determination with the relay voting scheme.

1.4 Effects of Data Latency on a WAMCS

Synchronized phasor measurement of the power system allows engineers the ability to capture snapshots for an electrically expansive area of the system. The system quantities are measured, time-aligned, and sent to applications in real-time for monitoring and control applications. With the data synchronized to a common clock, and time-aligned for comparison across the system, the question that arises is, “How long does it take for the data to arrive?” Many utilities are installing these measurement systems with rough estimates of the overall expected latency of the system. Documentation of the delays associated with each device in the network is limited, or absent; often the latency of certain components cannot be measured until installed.

The necessity of knowing, or at least understanding, these latencies is due to the requirement of this data in real-time applications. For those applications such as post-mortem analysis or model validation, the latency is generally not a concern; rather, reliable and accurate data would be considered more important. Even for applications such as situational awareness or wide-area monitoring, the latency requirements set by the applications can be relaxed due to the speed in which the visualization and user can react to the changing data. It is those applications of wide-area control or protection, such as angular or voltage stability and reactive power monitoring, that the data must be processed and control action performed within a determined amount of time to maintain system stability. Latencies may cause the Wide-Area Measurement and Control System (WAMCS) to initiate action outside of this limit, creating a useless synchrophasor application. The quantification of these latencies is one of the primary focuses of the work performed for this thesis.

Chapter 2

Adaptive Protection using Decision Trees

Data Mining can be defined as the nontrivial extraction of implicit, previously unknown, and potentially useful information from data [14]. This process is used to transform a data base of information into a human-comprehendible format. Techniques for data mining include but are not limited to: artificial neural networks, fuzzy logic models, and decision trees. The most common application of data mining in power systems is decision trees [15]. Examples of decision tree (DT) applications to power system problems range from security assessment and fault detection to load forecasting and economic dispatch. This is primarily due to their efficiency in processing extensive amounts of data and the real-time processing time once the decision trees are implemented for determining power system operating conditions.

One of the latest uses of data mining techniques such as decision trees in adaptive protection of the power system is presented in [12]. The work described in [12] demonstrated that DTs provide a highly accurate, near real-time assessment of system conditions. Based on offline simulations of the power system, a decision tree (DT) was grown using CART® to classify the condition of the power system. Classification and Regression Trees (CART) are a recursive partitioning method used for building prediction models from data. CART® is a software program by Salford Systems that applies the mathematic theory of CART to generate decision trees based on large amounts of data.

Decision trees applied to power system security assessment provide a classification of “stressed” or “safe” operating conditions based on a predetermined set of measurements across the network. This classification can be used for a number of different applications including arming a protection scheme or adapting relay settings. Real-time arming or alteration of a protection scheme provides a number of possibilities for improving the reliability of the power system. One such application is the use of a voting scheme; voting schemes in power system protection have been applied for many years but adaptation of the scheme based on the current state of the system is a newer concept. Should a single relay or relay scheme fail or misoperate, a majority vote of relays would restrain the protection system from sending a tripping signal to relays. Using the output of the real-time assessment of the system from the DT, the arming of the voting scheme can be adaptively applied to the wide-area system conditions.

This chapter provides an understanding of data mining; particularly building, trimming, and optimizing decision trees. It also details how the theory of data mining techniques is physically applied to the power system. Furthermore, it provides a link between the theory and application of an adaptive voting scheme through wide-area measurement techniques.

2.1 Overview of Decision Trees

As defined by [16], a decision tree (DT) is "a collection of 'test nodes' and 'terminal nodes', organized in a tree, structured upside down." This definition can be expanded upon by explaining the overall structure and purpose of decision trees and the basic functionality of each tree node. The process of building, trimming, and optimizing the trees based on a 'learning set' will also be explored. This discussion will form the basis for applying decision trees to adaptive protection using wide-area measurement, particularly to an adaptive voting scheme.

Figure 2.1 shows an example decision tree; this tree is provided to explain the different components of a DT. The tree consists of root, splitting, and terminal nodes. The root node makes the first split in the data using the entire sample data set. Splitting nodes are those nodes in the tree that further split the data into subsets s_L and s_R. The process of determining additional splits in the data is performed until a terminal node is reached. The terminal nodes in the tree provide the classification of the data.

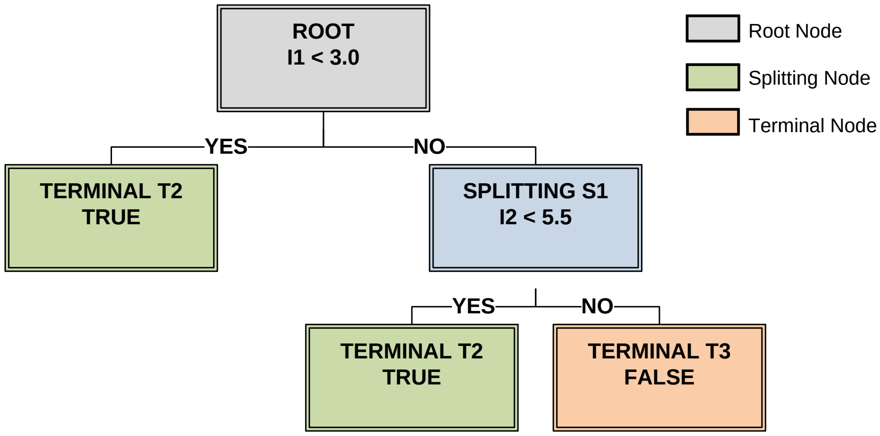


Figure 2.1: Decision Tree Elements

Both the root node (the initial split of the sample data set)and splitting nodes (further splits towards increased homogeneity of the data) of a decision tree test an 'attribute', a_i, of the data against a threshold value, $v_{t,i}$.

$$a_i < v_{t,i}$$

Referring back to Figure 2.1, the splitting node S1 shows the current on some line, I2, being compared against a threshold value of 5.5. The outcome of this inequality provides the determination of the classification of the data as Secure or Insecure. Although decision tress can have splitting nodes that split the data into multiple subsets, [12] uses only binary splits for constructing DTs consisting of true/false or yes/no answers. Binary, rather than multi-way splitting was chosen because a binary algorithm is more efficient. Each of the two subsets created by a splitting node are more homogeneous in classification that the splitting node itself.

The tree is constructed through recursive splitting of the learning sample set L, which is defined as the sample used for building the decision tree. Each split attempts to increase the homogeneity of each subset. The process of building the tree intelligently uses this recursive nature to test attributes against thresholds such that the splitting node minimizes the impurity of the two created subsets. Typically an impurity index is used to determine whether the data is more or less homogeneous. This process is continued until either the classifications are entirely homogeneous, meaning the impurity index is maximum, or it is not possible to further enhance the homogeneity of the subset [16].

Decision trees are built using a sample of data called the learning set. This sample provides the necessary data for the tree to 'learn', or build the tree offline. Building the tree is iterative and recursive and requires a significant amount of computational power and time. After the tree is generated, the results can be used in a real-time manner to determine, or classify, the state of the system with regard to the testing performed by the root and splitting nodes.

The benefit of using this form of data mining towards a problem with significant amounts of data is that implementation of the decision trees require very little computational effort. Although generation of the decision tree requires an enormous amount of data and computation time, this is performed offline prior to implementation. During real-time operation of the decision trees, only a few key attributes and their threshold values are required to make a statistically significant assessment of the system. These attributes and their corresponding thresholds are generated during the building process of the decision trees, and provide useful information when applied to power system protection.

2.2 Decision Trees for Power System Protection

The nature of decision trees provide a simple, efficient, and accurate representation of the operating state of the data provided based on key attributes and their comparison to predetermined threshold values. These trees can be used in real-time applications due to this simplicity and processing power. This section provides a connection between the DT output and the physical power system measurements and components.

The decision of where to install this type protection is prompted by the idea of 'critical locations' in the power system. For the context of this work, 'critical locations' are those locations where a N-2 contingency (second simultaneous disturbance) caused by protective relay mis-operation, such as hidden failure, would results in an unstable operating condition. In general, the power system is designed with high reliability such that these events will not cause catastrophic failure should they occur. This becomes more of a concern during highly stressed system operating conditions where the additional outage of an important line on the system would move the operating condition of the power system closer to instability. Locations where this is more of a prevalent concern are critical locations for implementing an adaptive protection system.

As discussed in section 2.1, the decision root node and tree splitting nodes determine key attributes and their respective threshold values. Attributes from the DT directly correspond to measurements in the power system, and the threshold values set limits for these measurements. The attributes provided for CART to build the optimal tree are bus voltages and line currents at certain locations. Power system stability is most commonly a system-wide or regional problem involving multiple generators; therefore, an overall system perspective is required in the learning set for CART to use. This situation defines the idea of Wide Area Measurement Systems (WAMS).

Measurements requiring real-time monitoring in the power system are included as possible attributes used by the decision tree. The final decision trees, as produced by CART®, corresponded to physical measurements taken in the substations, requiring equipment necessary to monitor bus voltages or line currents. Phasor measurement units (PMUs) are the only device currently available to provide a wide-area view of the system at a given instant in time; they provide a snapshot picture of the power system measurements. The classification process using the decision tree requires all measurements to be time-aligned, the function of a phasor data concentrator (PDC). The decision tree then becomes an application in a wide-area measurement system, requiring time-aligned synchronized measurements of the power system. PMUs measuring voltages and currents are only required at locations determined by the DT splitting nodes; these attributes (power system measurements) will be compared with threshold values (measurement limits) in the decision tree implementation.

At the time of this writing, two means of assessing the operating condition of the system exist: a PC connected to a PDC collecting the data or a PDC performing the assessment itself. Few manufacturers of PDCs allow user-programmable post-processing of the data, but the benefits are evident. With the PDC performing the assessment there is no need for an additional device such as a PC; this results in reduced latency and added reliability since there are less points of failure. Following aggregation and time alignment of the data, required measurements can be checked with their thresholds and the decision tree can be traversed and processed. Implementation of this setup is discussed in more detail in Chapter 4.

The location of the PDC is based on the time requirements of the system and the capabilities of the communications network. Ideally, the PDC would be installed at the critical location in the system, rather than at a centralized location such as a control center. The main advantage of using the critical location in the system is the reduced communications delays. Aggregation at the critical location would eliminate the transmission of the data following the alignment of the PMU packets; the classification status of the state of the system could be sent over the local substation Ethernet or hard-wired as a screw terminal input contact to the relays protecting the critical line. This signal is referred to as the arming signal of the protection scheme, and is the input of the voting scheme employed for reliability of that line.

2.3 Application of Decision Trees

With a foundation of decision trees and the connection to power system laid out, the question that arises is "how will this system be integrated?" This question is multifaceted, is focused primarily on the implementation of the decision tree, and is the main contribution of the work performed for this Thesis.

Programs such as CART® use the learning set of data provided and generate an optimal decision tree to classify the data based on attributes' values as compared with threshold values. The output of CART® is a decision tree figure, and does not provide a useful means of implementing the concept of DTs into a functional system. Fortunately, binary decision trees are an application of if-else statements in software programming languages.

The program is simply an input-output function, analyzing inputs and determining whether to enable or disable an output. The inputs to this program are the PMU measurements pertaining to the attributes of the root and splitting nodes. These measurements are compared against predetermined threshold values. A set of nested if-else statements are traversed, just as the tree is traversed visually, to determine the state of the system. If the attribute values are such that the system is classified as “stressed”, then the arming output is enabled; if the system is classified as “safe”, then the arming output is disabled. A simple digital bit is necessary to set the output signal as 1 or 0, corresponding to stressed or safe operating conditions on the system.

For the adaptive voting scheme implementation, the output is sent to a voting processor. This processor will read the digital output bit to determine if the relays should be in a voting state or a normal operation state. The functionality of the voting is described later in this chapter.

2.4 Conversion of Decision Tree to If-Else Statements

Offline studies are performed to build and trim a decision tree to optimal size. Software such as CART® provides a graphic of the decision tree, but this output must be converted to an implementable format. Figure 2.2 provides an example of a basic decision tree. This tree consists of a root node, three splitting nodes, and 5 terminal nodes. Based on the attributes within the decision nodes, it can be determined which measurements are required; for simplicity, these have been referred to as PMU1.I, PMU2.I, and PMU3.I. This corresponds to the current phasor magnitudes for three separate PMUs.

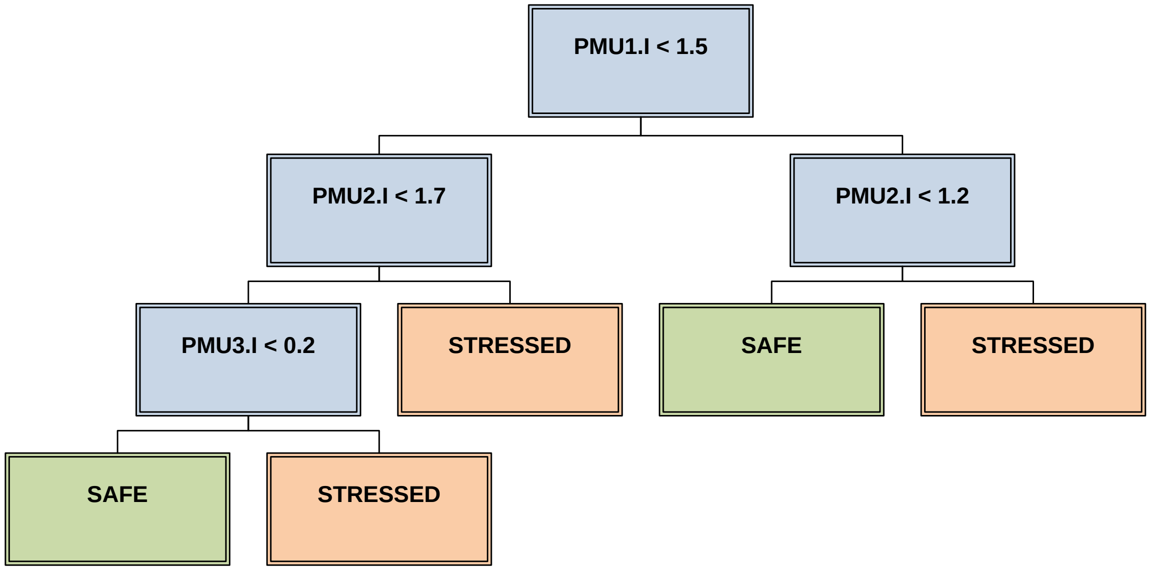


Figure 2.2: Example Decision Tree

Any form of decision tree can be converted to "if-else" (binary tree) or "if-elseif-else" (multi-subset tree) statements for programming. Each splitting node in the tree represents an "if" statement; the left branch representing a true statement and the right branch representing a false statement. The terminal nodes in the tree provide a classification of the system state. The left decision node represents a safe condition and the right decision node represents a stressed condition. The code below provides the if-else code that would be generated for the tree shown in Figure 2.2.

```
IF ( PMU1.I < 1.5 ) {
      IF ( PMU2.I < 1.7 ) {
            IF ( PMU3.I < 0.2 )
                  STATE = SAFE;
            ELSE
                  STATE = STRESSED;
      }
      ELSE
            STATE = STRESSED;
ELSE {
      IF ( PMU2.I < 1.2 )
            STATE = SAFE;
      ELSE
            STATE = STRESSED;
}
```

Programming if-else statements is a simple task for an engineer to perform. Given a decision tree output figure, an accompanying program can be created to match that tree. This is convenient if multiple trees were to be implemented, for example to accommodate seasonal conditions. [12] generated two decision trees for the California system using a Heavy Winter model and a Heavy Summer model for different operating conditions. A MATLAB program was created to produce a set of if-else statements modeling the trees generated, and is provided with the input files in Appendix A.

2.5 Relay Voting Scheme

The end-use application of a voting scheme seeks to increase the reliability of the protection system. During normal operating conditions of the power system, the voting scheme is disabled and the protective devices operate as expected. The decision is based on an 'or' function; that is, any one relay or scheme can trip the breaker(s) protecting the equipment. This defines an increase in dependability for this system. Should a tripping condition occur, there exists a bias to remove that component from service with a greater chance of misoperation at any given time. A redundancy check is not present to verify that indeed a tripping condition has occurred. On the other end of the reliability spectrum, during stressed system conditions, the voting scheme is armed. During an armed state, the devices vote on tripping the breaker(s) prior to the action being taken. The voting process ensures that a misoperation does not occur, which defines an increase security on the system. During stressed conditions, a misoperation of a relay and breaker would cause heavier loading of the system and could potentially lead to a cascading failure or worse a blackout condition. The voting scheme is a means of increasing the certainty that an event such as this will not occur due to misoperation.

The arming signal generated at the real-time analysis stage of the adaptive protection system is provided to the field devices. This signal can be sent to a standalone device for arming or disarming the voting scheme, or the signal may be imbedded into the code of the protective relays themselves [17]. Figures 2.3 and 2.4 show both configurations.

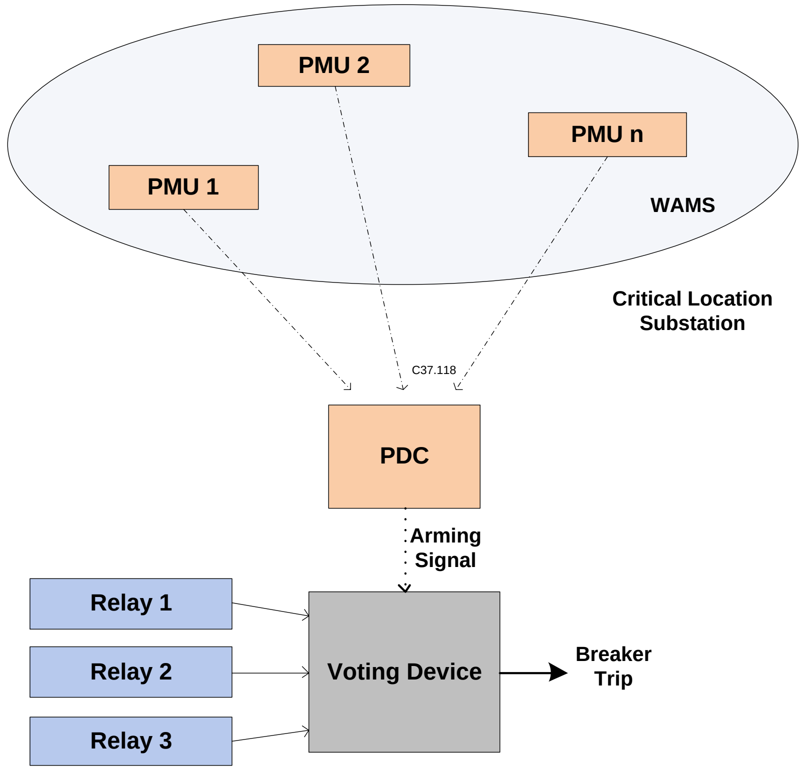


Figure 2.3: Voting Device Configuration

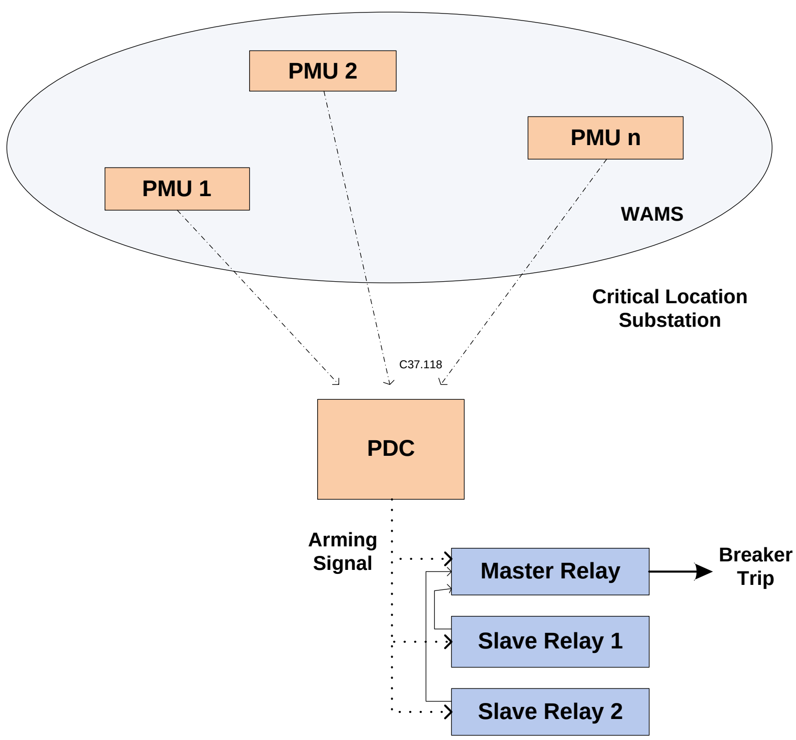


Figure 2.4: Embedded Master-Slave Configuration

With an additional device providing the voting mechanism, the existing relays can remain untouched because their functionality remains unchanged. Each device monitors the power system quantities and provides an output signal for tripping. The only required change would be changing where that output signal is sent. A problem with this setup is that during normal, safe operating conditions, the relay trip signal must be sent through an additional device increasing latency prior to breaker operation.

Rather than changing the configuration of the system, the relays can be set up in a Master-Slave approach. Only the internal logic of the relay trip settings is changed to account for signals coming from the other protective relays monitoring the common element. This setup does not include an additional device and also does not require a change in hardware configuration. It does however require the changes in the relay settings to be documented, and also requires a change in maintenance and testing procedures for the devices. This is often a difficult obstacle to overcome in the utility industry.

Chapter 3

Data Latency in Wide Area Measurement Systems

Wide-area measurement system applications can be categorized based on their input requirements:

- Accuracy: How much can the incoming phasor data be trusted? Is it accurate during transient conditions or only during steady state? Can the data be trusted for metering purposes, protection purposes, or both?
- Reliability: How often do data packets go missing, and can the application using the data handle missing packets?
- Latency: Does the application need the data before a certain amount of time? What is the expected delay of incoming measurements, and what is the maximum allowable time for the application to wait before proceeding? What is the added latency to the existing protection scheme?

One of the primary focuses of this paper is to explore where latency is introduced in a Wide-Area Measurement and Control System (WAMCS). This was done by investigating the effects it has on applications using the synchronized phasor data, and attempting to quantify latency of WAMCS components.

Latency has a number of definitions based on the application; for the context of this work, latency is defined as the time between when an input action occurs and when the desired output corresponding to that action occurs. This chapter will explain in detail the major contributing factors of latency in a WAMCS and examine the reasons behind these delays. Latencies have been separated into the following categories:

- Phasor Measurement Latency
- Device Latency
- Wide-Area Network Latency
- Total Wide-Area Measurement and Control System (WAMCS) Latency

Each component in the WAMCS measurements system has a delay associated with it, but the overall latency of the WAMCS is defined as the difference in time between the state change of an input signal at the measurement location and the response to a signal at the control location.

3.1 Definition of Latency

The definition of latency is often ambiguous and generally varies based on the system or process being performed. A common engineering definition of latency for a system is defined as *the time between an initiating event at a source and the response at some destination*. This definition can be applied to Wide-Area Measurement and Control Systems (WAMCS). The discussion below identifies the locations that latency is apparent in a WAMCS, and defines each subsection; Figure 3.1 provides a visual perspective of these latencies.

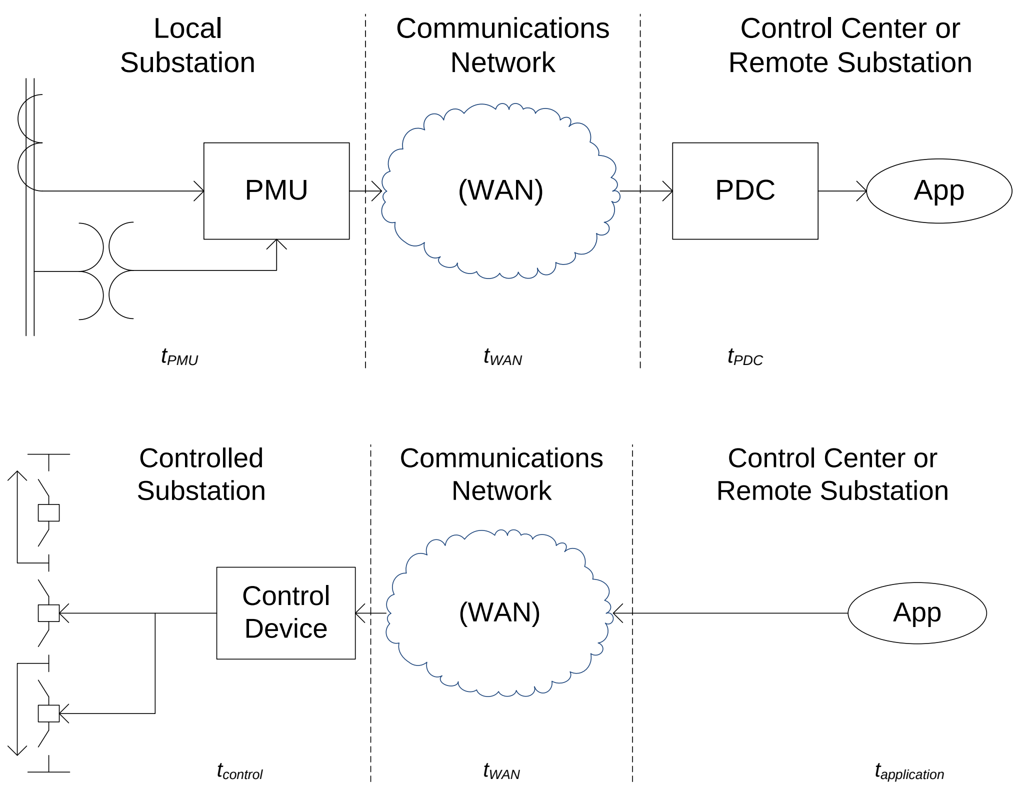


Figure 3.1: Wide-Area Measurement and Control System Configuration

The measurement of power system quantities originates at the instrument transformers in the power system substations. The secondary values of these devices are sampled by the PMU and used to compute discrete phasor values with a corresponding time stamp. Device latency is defined as *the time required for a WAMS device to receive, process, and output the packets of data*. For example, PMUs must convert analog quantities to discrete phasors and send them as packets of data while PDCs must aggregate and time-align these packets; each device has its own function and associated latency. The data packets are sent from remote substations throughout the system to a central location such as another substation or control center over a wide-area network (WAN). WAN latency is defined as *the time difference between when the source transmitter at the remote substation sends the data and when the receiver at the local destination receives the data*. Once the data has reached its destination, most commonly a PDC at a control center, the data must be processed. Application latency is defined as *the time required to process the data and make meaningful decisions based on the measurements received*. The functions performed in these applications vary in the complexity and therefore the time required to perform them. For example, changing the digital status of a bit may take less than a millisecond, while visual displays for situational awareness may take hundreds of milliseconds. Fortunately, control applications have been designed efficiently and require a very small amount of time. These control applications may often require a control signal to be sent to a device in the field for operation of a breaker for switching in or out a capacitor bank or transmission line. Control latency is defined as *the time required for the controlling device to operate after receiving the control signal*.

Each subsection of a WAMCS has a different definition of latency, and the total system latency of the system is defined as the aggregate total of all subsystem latencies. More clearly, the total WAMCS latency is *the time required for operation of a control or protection device following instigation at the measuring device on the system*.

3.2 Sources of Latency

A discussion of the sources of latency and an understanding of each is discussed in the following section. Latency can be classified based on where it occurs in a Wide-Area Measurement and Control System (WAMCS), as follows:

- Phasor Measurement Unit
- Phasor Data Concentrator
- Wide-Area Network
- Application
- Control Device
- Overall System

3.2.1 Phasor Measurement Unit Latency

A PMU rapidly samples the analog power system quantities from the secondary side of instrument transformers, generating discrete samples of data. The latency of this phasor measurement process can be broken down into two components: relative time stamping and response time. Both these forms of latency are described in this section.

A PMU requires a window of data samples to generate a phasor representation of the signal. A single sample is not sufficient for determining the phase angle of the quantity being measured with enough accuracy. Recursive phasor algorithms are commonly used to generate phasors at rates near 1 kHz. For example, a PMU may sample the analog values through the A/D converter at a very high rate and create phasor samples at 720 Hz. This would create 12 phasor values within one cycle for a 60 Hz system. For a 1 cycle window length, the PMU would create a single output packet using these 12 recursive phasor values; commonly an average phasor is calculated. Figure 3.2 shows an example of the process described above. The black line is the analog signal applied to the PMU, the red dots are the sampled values through the A/D converter, and the shaded area is the 1 cycle window.

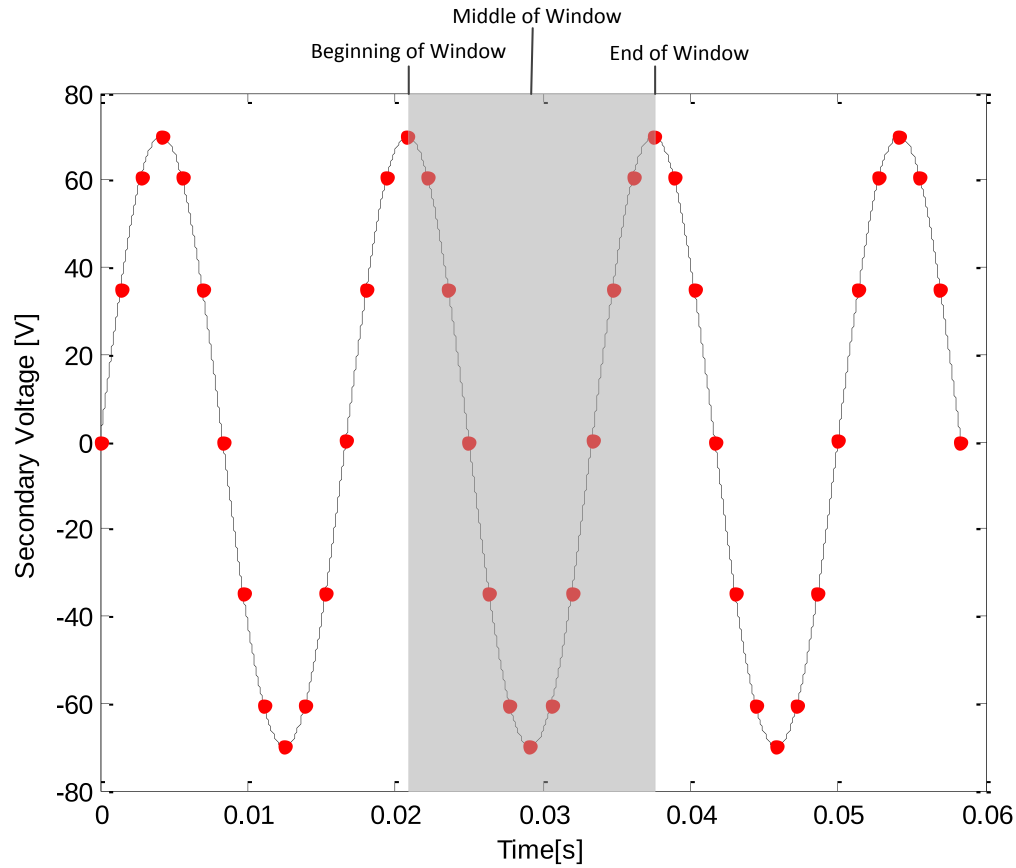


Figure 3.2: PMU A/D Conversion and Windowing

The output phasor created for the window of data collected must be synchronized to a universal time system. The question becomes where to place the time stamp for the given window of data measured. Each input sample was recorded at a different time, but a single time must represent the window. The placement of this time creates "latency" when observed by an application requiring the data. There are three logical locations to for placing the timestamp: beginning, middle, and end of the measurement window.

Regardless of the placement of the time stamp in the measurement window, the PMU must wait for the full window of data to be sampled. Where the time stamp is placed determines how much delay is expected from the PMU measurements. Placing the time stamp at the beginning of the window, the delay of the data would seem as though it was a full window length. The packet sent out for the example above would be delayed 1 cycle, or approximately 16.67ms. Conversely, placing the time stamp at the end of the measurement window would seem like the PMU had no latency. The packet would be sent out right on the time stamp; the example above would have a delay of 0ms. Likewise, the middle of the window placement would be observed as a half-cycle delay, or approximately 8.33ms.

One could argue that the end of the measurement window would be the best option for placing the timestamp because the delay would be zero. This may not be true because the time stamp would be a full window of time away from a measurement it must represent. For this reason, another may argue that the middle of the window is the better option because it better represents each measurement within the window of data.

Neither the prior or current Standard for Synchrophasors for Power Systems require the time stamp to be placed in a certain location relative to the data window. The previous standard, IEEE Std 1344, recommended the end of the window for placing the time stamp [18]. The current standard, C37.118-2005, recommends the middle of the window. Following C37.118-2005, each phasor reported by the PMU would have a delay of

$$Delay_{PMU,Meas} \geq \frac{L}{2F_r}$$

where F_r is reporting frequency and L is the window length, for example 60Hz reporting frequency with a 3 cycle window length would result in a delay of 25ms. Although the standard does not specify a window, it does require that if the 1PPS occurs at a zero crossing, phase angle of the phasor should have the corresponding +90˚ or -90˚ shift.

The other important aspect of PMU latency is based on the measurement of the PMU itself rather than the processing time of the device. During a transient condition, the PMU reporting phasors may experience errors from the actual events occurring. A contributing factor to this is the response of the PMU based on its filtering characteristics. This concept is best understood by observing the step response of a PMU, using either a magnitude or phase angle step change. This test is a standard test performed on PMUs to conform to the C37.118 standard. The response to the PMU due to an instantaneous step change gives the largest delay in measurement of the step that the PMU would experience; this is a worst case latency calculation.

3.2.2 Phasor Data Concentrator Latency

Multiple PMU measurements are aggregated and time aligned at a phasor data concentrator. Ideally, each PMU packet would arrive at the PDC at the same instant but this is rarely, if ever, the case. Data sent from certain PMUs often must traverse hundreds of miles of communications channels to a PDC, while other PMUs may only be sending data from the substation nearby. This results in each PMU measurement arriving at the PDC at a different time instant. For the PDC to aggregate PMU measurements and time-align them based on their time tag, the PDC must allow for varying wide-area network delays. Typically a Wait Time is set such that the PDC will wait for incoming PMU packets for a maximum of this set time. A time aligned C37.118-2005 output packet from the PDC for a given time stamp will be sent either once all the expected PMU data has arrived or the Wait Timer is met. Data arriving after the timer has expired is either discarded or sent to a historian for offline analysis. PDC latency must often meet a time requirement, which is usually set by the application requiring the data. For real-time control applications, the Wait Time must be minimized such that the application can determine and initialize a control action in a timely manner. Therefore the Wait Time of the PDC must lie between the limits of

$$T_{MaxPMU} \leq T_{WaitTime} \leq T_{Application}$$

In addition to data aggregation and time alignment, PDCs perform additional functions on the data such as buffering, data validation, and format conversion. These functions require time for computation and processing. Engineers must be conscientious of both the Wait Time setting as well as the additional processing time of the PDC. The time alignment and buffering of PMU data of most concern, rather than the processing time of the PDC.

3.2.3 Wide-Area Network Latency

The PMU measurements sent from remote substations in the power system must be collected and time-aligned for measurement and control applications. Transmission of this data depends on reliable and efficient operation of the Wide Area Network (WAN) interconnecting the different networks in the power system. Although reliability is a high priority for synchrophasor technology, the primary concern for this thesis is the question: "how fast will the data arrive?" The applications at the control center are often limited by the latency of the WAMS, and more directly the WAN itself.

Throughput in a communications network is often defined as the average rate of delivery for successful messages over a communications network. This quantity is often measured in bits/second or packets/second, and is primarily controlled by the bandwidth, or available communications resources, of the network. The available resources, throughput, and therefore communications medium of the WAN determine the latency of the subsystem.

A number of different communications systems are used to connect measurement devices at substations in the field to collection devices at different substations or control centers. The main mediums for this data acquisition include the following:

- Microwave Radio
- Power Line Carrier
- Fiber Optic Cable

Microwave communications are heavily ingrained in today's power system, extensively used for SCADA communications. This is due to their large bandwidth or capacity to transport the data, reduced costs due to no wire conductors or right of way, and resistance to electromagnetic interference in the form of noise. These systems do require a line-of-sight path between transmitter and receiver, therefore requiring frequent repeaters.

Power line carrier (PLC) technology uses the existing power system architecture to transmit data between locations but overlaying a high frequency signal on the low frequency power lines. Some relaying schemes use this technology but the bandwidth is limited and the loss of a transmission line will result in lost data.

The most preferred means of data transmission, especially for Wide-Area Measurement Systems, is fiber optic communication. Section 1.1.2 introduced the concept of fiber optic communication, and the benefits are apparent. Although fiber systems require repeaters and a physical cable between source and destination, existing right of way can be used in conjunction with the transmission lines. The nature of optics, rather than electrical charge makes this method extremely immune to substation and atmospheric noise, and also creates a virtually unlimited path for digital data transmission.

Synchronous Optical Network (SONET) is the American system for optical communications in electrical utilities under American National Standards Institute (ANSI) T1.105 and Bellcore GR Standards [19]. SONET systems use a ring topology where data can be transmitted in either direction. This configuration allows for a break in the ring with outage of the communications system because the break will be detected and the data rerouted. Figure 3.3 shows the basic concept of this ring topology [20]. The fully working ring of fiber optic communication provides data to all nodes, as shown by the blue flows. If a disturbance or failure occurs between for example MUX 3 and MUX 5, the data is rerouted as shown by the red pathways. Even following a disturbance, all nodes in the ring have maintained operation. If data was being sent from MUX 3 to MUX 5, and a disruption in transmission occurred between them, the data would be forced around the ring through all other repeaters before arriving at the intended MUX 5.

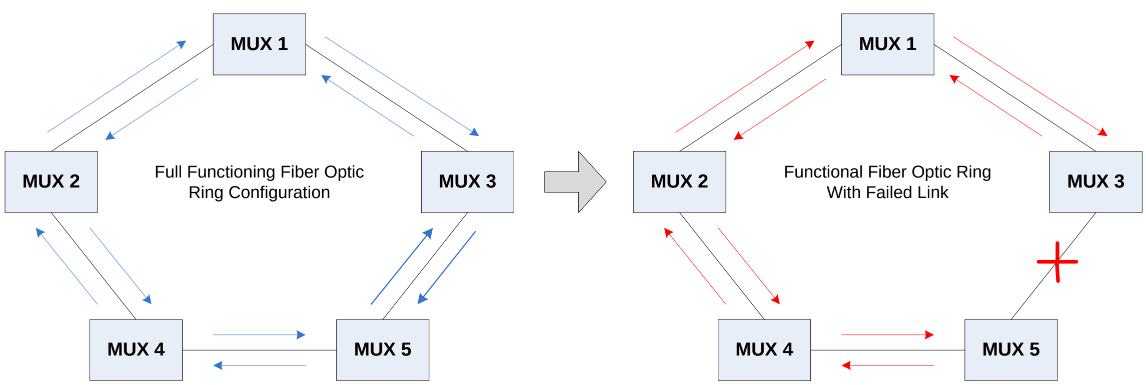


Figure 3.3: Fiber Optic SONET Ring Configuration

C37.118-2005 suggests the use of either RS232 or IP protocols, with IP protocol being the more popular choice. Under the IP protocol, either Transport Communications Protocol (TCP) or User Datagram Protocol (UDP) can be used. TCP is a protocol for data transmission which requires confirmation of both the presence of a receiving unit and the reception of data for transmission to occur. On the other hand, UDP is a protocol in which data can be sent from one device to one or multiple devices without acknowledge of data reception [21]. With that said the benefit of UDP is that it does not require time for confirmation and is thus better suited for real-time applications. Meanwhile, TCP confirms packet arrival, controls network congestion, and would be suited for applications requiring highly reliable data to the destination.

The WAN is one of the most difficult parts of the WAMS to quantify latency in a laboratory setting. The large number of variables contained within this subsystem has an effect on the latency, and many are hard to construct within the lab. For example, the communications path length is an important factor in the latency of a phasor measurement. Each measurement must traverse a different distance to the central processing location, often varying hundreds of miles for each PMU. Furthermore, the communications medium, transmission protocol, bandwidth, and equipment all contribute to the variability of this system.

One the goals of this thesis is to better understand the components and protocols of a WAN, and to determine how they affect the latency of the WAN.

3.2.4 Application Latency

Often overlooked, the applications requiring timely arrival of the data also have a latency and throughput capability associated with them. Programmable logic controllers (PLCs) and other microcomputer devices often decode the data and provide real-time applications such as visualization or power system protection and control using the phasor data. Relative to the WAMS latency requirements, the applications' latency is minimal, although a concern based on the type of application.

Time-criticality is a means of classifying applications of a wide-area measurement and control system. Three subsets of data applications can be defined based on the latency associated with each:

- Offline applications include data archiving, model validation, and post-mortem analysis. These applications do not require the data to arrive in a minimal amount of time; rather, they emphasize accuracy and reliability of the data over a latency requirement.
- Monitoring applications such as operator situational awareness and data visualization require the data to arrive quickly, but not "real-time". More important that instantaneous data is the clear and concise display and understanding of the data. This requires both a screen for display and a well-trained operator able to make quick decisions. The decision making of the operator is obviously slow compared with the real-time measurement of the power system and is thus the limiting factor for this type of application.
- Protection and Control applications such as line protection, modal analysis, adaptive protection, etc. all require a small latency of the data should they be used in real-time. For WAMS-based protection to be used in the power system, it must respond to a disturbance faster than the current protection system deployed. During a transient event on the system such as system oscillations following a fault or disturbance, the PMUs must provide valid and accurate data for control action to be taken.

The classifications provided above for the applications using the wide-area measurements illustrate the importance of understanding and quantifying the latency of a WAMCS. A control application deployed, with no heed towards latency, may be unjustified and invalid because the system cannot be controlled within a time threshold. Conversely, an offline application may not require latency to be a concern based on the end-use of the data.

3.2.5 Control Device Latency

For applications using the synchrophasor data for control applications, a control signal such as an arming or trip signal may be sent to devices in the field. Understanding the architecture of a WAMCS, this signal may need to traverse the WAN before being applied to a device. If possible, the control devices and application could be located within the same location to avoid additional delays; this could include placing the PDC at a critical location in the system rather than at the control center. The devices reacting to the control signal sent from the application also have a delay. These delays can vary based on the type of device using the signal. Literature review and laboratory testing were performed to determine what these latencies are for different devices.

For controls involving a breaker operation, the time requirement varies based on the voltage level of the breaker. [22] and [23] give examples of rated interrupting time for breakers at 230kV and 500kV: 2 or 3 cycles for 230kV and 2 cycles for 500kV systems. This could result in up to 50 ms of additional delay for a breaker operation at 230kV.

Shorter delays exist for applications applying an arming scheme to relays. The only delay associated with this is the acknowledgement of the relay or relay scheme of a state change of an input signal. This was tested in the lab using a relay contact and observing the time between when the relay initiated a command and the state change of the input contact.

Chapter 4

Implementation of Adaptive Protection Scheme

Implementation of the security/dependability adaptive voting scheme uses decision trees to make a classification of the operating condition of the system. As described in Chapter 2, classification was made at the PDC based on its ability to time align and aggregate system-wide measurements. PMU data collected by the PDC was processed through an additional user-defined set of functions that determined whether to arm or disarm the voting scheme based on stressed or safe operating condition, respectively. Many manufacturers will work with power engineers to develop firmware for their specific applications, but few PDCs are programmable by the user of the device. This chapter discusses two commercially available options for implementing this scheme at the PDC level, using equipment that provides the capability of programming user-defined functions into the device.

The first option explored was a hardware based PDC, the Schweitzer Engineering Laboratories SEL-3378 Synchrophasor Vector Processor (SVP) [24]. The SEL-3378 was programmed using the accompanying SVP Configurator software from SEL. The other option explored was a software based PDC, openPDC, which is a software platform from Grid Protection Alliance, running on a Windows-based computer [25]. This open source program was downloaded from the Internet and programmed using C# and Visual Studio 2010.

This chapter explains two independent configurations of the entire adaptive voting scheme. First, the decision tree functionality was tested using the hardware-based PDC. This work elaborates upon and completes the work performed in [17]. Second, the decision tree was implemented in the software-based PDC. The ability to implement the decision trees into this type of PDC was also explored. Lastly, the adaptive voting scheme in its entirety was implemented using openPDC and a programmable logic controller (PLC). This configuration expands upon the work done thus far on the

Security-Dependability Adaptive Voting Scheme. Topics discussed for both configurations include the network architecture, programming of the devices, testing methods, and results obtained.

4.1 Hardware PDC Implementation

The focus of the work performed using the hardware based PDC was to implement the Heavy Winter and Heavy Summer decision trees from [12]. This chapter uses the Heavy Winter model for explanation of the setup, testing, and results, but the methodology holds for the Heavy Summer model also. The decision tree used in the Heavy Winter model is shown in Figure 4.1 for clarity.

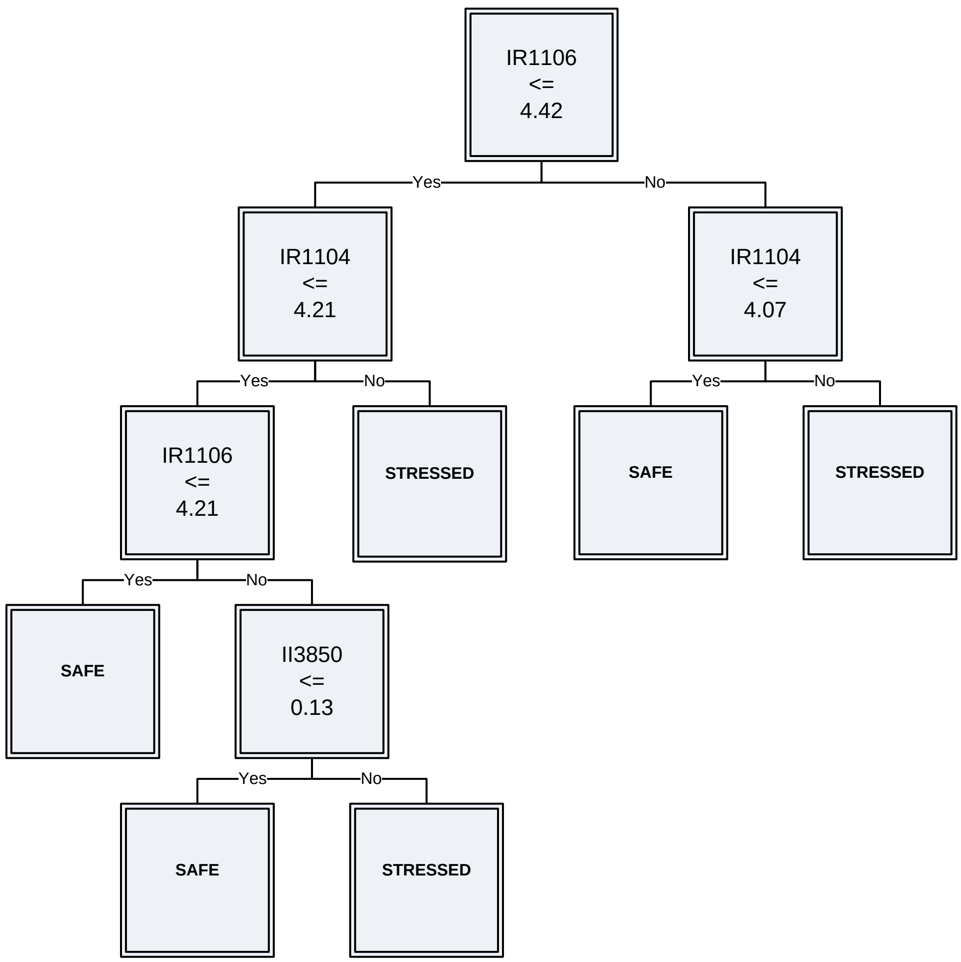


Figure 4.1: Heavy Winter Decision Tree

As the tree shows, there are three distinct line currents requiring monitoring by phasor measurement units (1104, 1106, and 3850). The physical location of these three lines is of no interest for testing purposes, but the number of measurements and required PMUs is of interest. The decision tree provided the measurement requirements for three distinct PMUs, each monitoring a power system quantity that is compared against a per unit threshold for classification. This requirement added the need of a per unit measurement system for testing. For field implementation of this system, an engineer would need the base value used to determine the quantity to enter into the tree in actual units rather than per unit values. For example, 4.21 may correspond to 2105A on a 500A base. The tree uses "IR" and "II" to refer to the real and imaginary components of currents in the system, where "IR" represents the real component of the complex current phasor and "II" represents the imaginary component of the complex current phasor.

4.1.1 Network Architecture

A test system was set up to emulate how the devices would interact with each other when deployed in the field. A relay scheme was reduced to a single relay to capture and record events from the PDC. The adaptive protection scheme was tested using the network configuration shown in Figure 4.2.

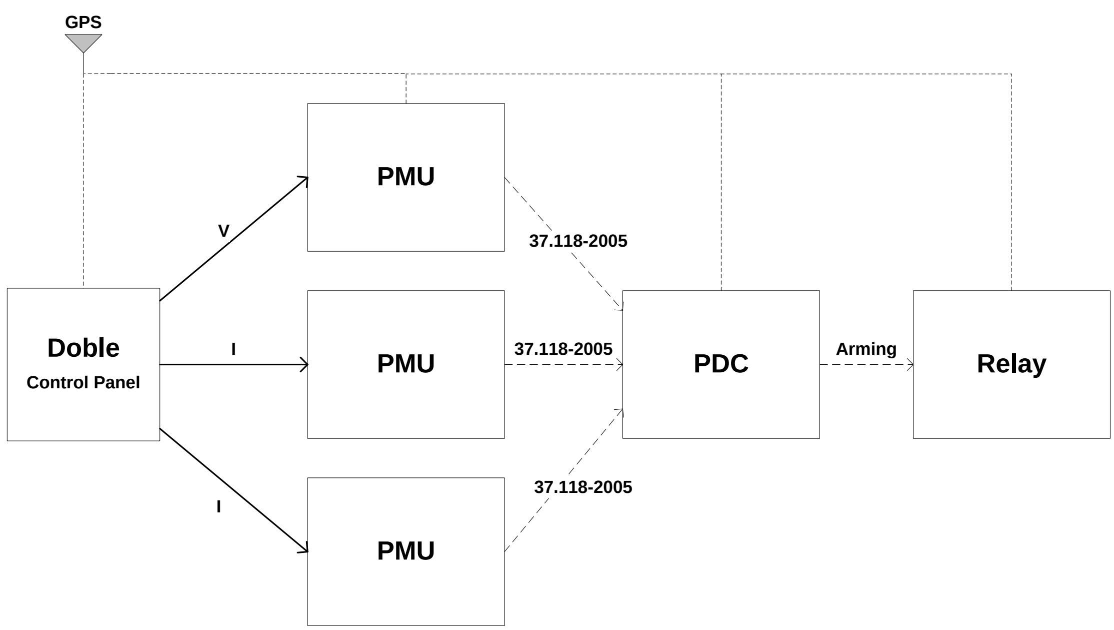


Figure 4.2: Hardware PDC Network Configuration

The Doble 6150 Power System Simulator is described in more detail later in Chapter 5, but for the purposes of testing implementation of the decision trees in the PDC, it was used as a user-selectable signal generator, with simultaneous 3-phase voltage and current output channels [26]. The Doble Control Panel is a computer user interface used to set and alter signals generated by the device. Since the device was time synchronized to UTC, it was used to select the appropriate voltage magnitude and phase angle to obtain the required real and imaginary components. Due to the structure of the tree and the limited output channels, the only way to test all possible conditions was to use a voltage as the origin node in the tree. All inputs were set to the same base values of V_{base} = 10V and I_{base} = 1A. The resulting tree used for testing the Heavy Winter decision tree is shown in Figure 4.3.

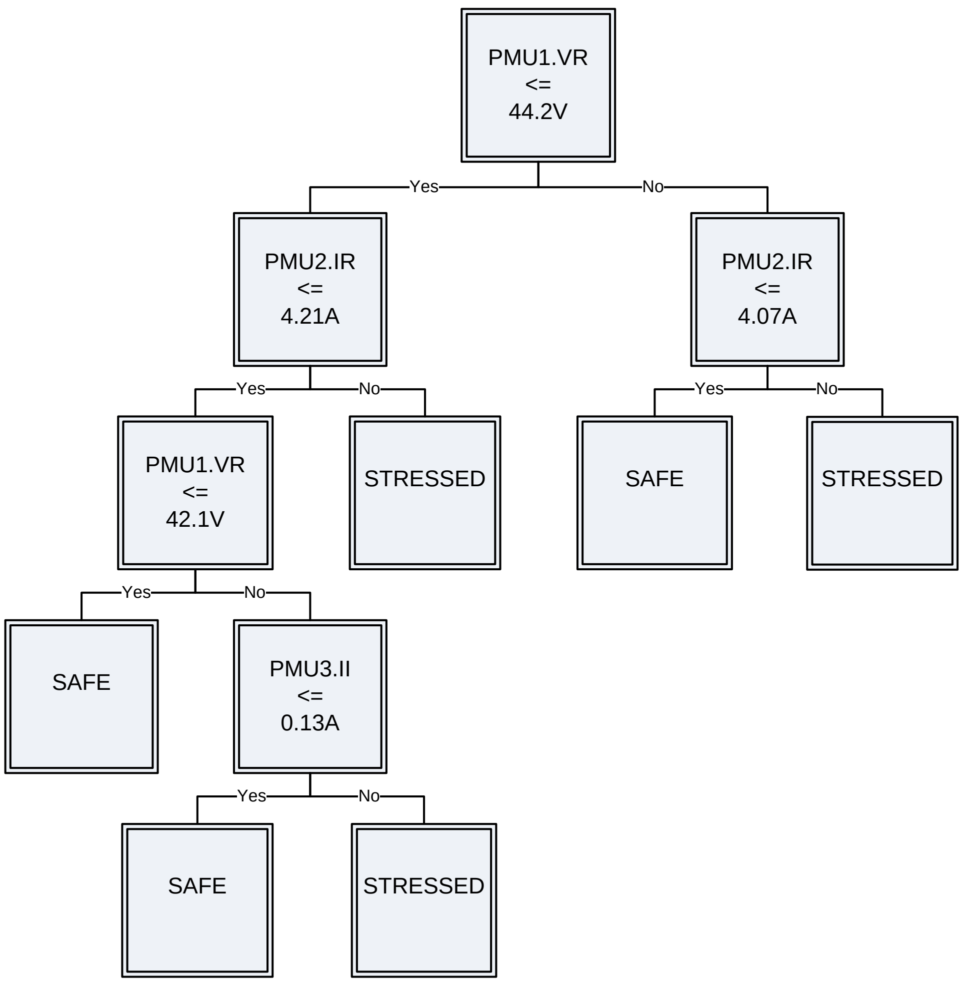


Figure 4.3: Hardware PDC Heavy Winter Testing Tree

Three separate PMUs measured 3-phase current or voltage quantities, and reported the positive sequence phasors for each. These phasors were sent over Ethernet channels to the hardware PDC using C37.118-2005 protocol. The PDC aggregated and time aligned these PMU measurements and reported a single C37.118-2005 output stream to a computer for observation. Additionally, the PDC also performed the decision tree functionality and provided an arming output signal to a relay. The actual implementation of this system deploys a voting scheme between three relays to decide on a majority trip or single trip. The implementation for testing the hardware PDC functionality was limited to sending the arming signal to a single relay rather than a set of relays. Of interest is the complete and accurate operation of the PDC, not the voting scheme. The work in [17] shows the operation of the voting portion of the overall adaptive voting scheme.

4.1.2 SEL-3378 and SVP Configurator Programming and Setup

The SEL-3378 Synchrophasor Vector Processor provides the features of a phasor data concentrator (PDC) as well as additional functions of a programmable logic controller (PLC). It aggregates and time aligns synchrophasor data as a traditional PDC would, and also allows for user-defined programmable logic. Up to 16 PMUs can be aggregated by the SEL-3378 using C37.118-2205 protocol or SEL's proprietary Fast Message protocol, and these messages can be time aligned at up to 60 messages per second. C37.118-2005 messages can be output to up to 6 external clients and one internal client with a throughput of 2 ms [27]. The PDC can also send Fast Operate or SEL Mirrored Bits messages to perform control action. SEL offers interface hardware such as the SEL-2515 Remote I/O module for interfacing with non-SEL products [28].

SVP Configurator is the SEL software package for configuring the SEL-3378. It provides a visual platform for setting communication parameters and programming protection applications into the device. The SEL-3378 consists of different internal components performing operations in real-time; the SVP Configurator software allows for managing each of these components. Programming of the SEL-3378 is performed using standard IEC 61131-3 logic such as Structured Text, Function Block Diagrams, Ladder Logic, and Instruction Lists. These functions were created in the SEL SVP Configurator software provided with the device. Once the program is created in software by the user, it is uploaded to the SEL-3378 and run in real-time.

The Time Alignment Client Server (TCS) is responsible for aggregation and time alignment of the incoming synchrophasor data, and is the first module programmed for the SEL-3378. Figure 4.4 shows the Structured Text programmed for the Heavy Winter Model, consisting of three input PMUs and a single output stream to the PC. PMU2 (IDCODE 1) is also used as the "relay" in this situation for the arming signal acknowledgement. In the field implementation, this would be replaced with the relays monitoring the critical line where the voting scheme is to be deployed.

```
* 487E Station A -- PMU*)
PMCU_INPUT[1].EN := TRUE;
PMCU_INPUT[1].IDCODE:= 2;
PMCU_INPUT[1].C37_118_CLIENT:=TRUE;
PMCU_INPUT[1].FASTOP:=1;
PMCU_INPUT[1].CONNECTION := 'E';
PMCU_INPUT[1].EPORT.SERVER_IP := '10.222.3.171';
PMCU_INPUT[1].EPORT.SERVER_CMD_PORT := 4712;
PMCU_INPUT[1].EPORT.TRANSPORT_SCHEME := 'TCP';
PMCU_INPUT[1].EPORT.CLIENT_IP := '10.222.3.181';
PMCU_INPUT[1].EPORT.CLIENT_DATA_PORT := 4712;

(* 487E #2 -- Relay*)
PMCU_INPUT[2].EN := TRUE;
PMCU_INPUT[2].IDCODE:= 1;
PMCU_INPUT[2].C37_118_CLIENT:=TRUE;
PMCU_INPUT[2].FASTOP:=1;
PMCU_INPUT[2].CONNECTION := 'E';
PMCU_INPUT[2].EPORT.SERVER_IP := '10.222.3.178';
PMCU_INPUT[2].EPORT.SERVER_CMD_PORT := 4728;
PMCU_INPUT[2].EPORT.TRANSPORT_SCHEME := 'TCP';
PMCU_INPUT[2].EPORT.CLIENT_IP := '10.222.3.181';
PMCU_INPUT[2].EPORT.CLIENT_DATA_PORT := 4728;
PMCU_INPUT[2].EPORT.FASTOP_PORT := 23;
PMCU_INPUT[2].EPORT.FASTOP_PORT_TELNET_EN := TRUE;

(*Fiber PMU*)
PMCU_INPUT[3].EN := TRUE;
PMCU_INPUT[3].IDCODE:= 6;
PMCU_INPUT[3].C37_118_CLIENT:=TRUE;
PMCU_INPUT[3].FASTOP:=0;
PMCU_INPUT[3].CONNECTION := 'E';
PMCU_INPUT[3].EPORT.SERVER_IP := '10.222.3.176';
PMCU_INPUT[3].EPORT.SERVER_CMD_PORT := 4713;
PMCU_INPUT[3].EPORT.TRANSPORT_SCHEME := 'UDP';
PMCU_INPUT[3].EPORT.CLIENT_IP := '10.222.3.181';
PMCU_INPUT[3].EPORT.CLIENT_DATA_PORT := 4713;

(* OUTPUT01*)
PMCU_OUTPUT[1].EN := TRUE;
PMCU_OUTPUT[1].MRATE := 60;
PMCU_OUTPUT[1].CLIENT_IP := '10.222.3.185';
PMCU_OUTPUT[1].CLIENT_DATA_PORT := 4750;
PMCU_OUTPUT[1].TRANSPORT_SCHEME := 'UDP_S';
PMCU_OUTPUT[1].SERVER_IP := '10.222.3.181';


TCSconfigOK := TCS_CONFIG(EN := TRUE,
PDC_IDCODE := 1000,
pHID := ADR(HEADER_118),
NFREQ := 60,
MRATE := 60,
MISSING_MESSAGE_THRESHOLD := 6,
TIME_UNSYNC_BLOCK := TRUE,
pCMD_OUT_DATA_IN := ADR(PMCU_INPUT),
pCMD_IN_DATA_OUT := ADR(PMCU_OUTPUT),
pERROR := ADR(TCS_ERROR_OUT),
pSTATUS := ADR(TCS_STATUS_OUT));
```

Figure 4.4: SEL-3378 TCS Structured Text

Figure 4.5 elaborates on the SEL-3378 architecture and the internal interaction between processes following PMU data frame arrival.

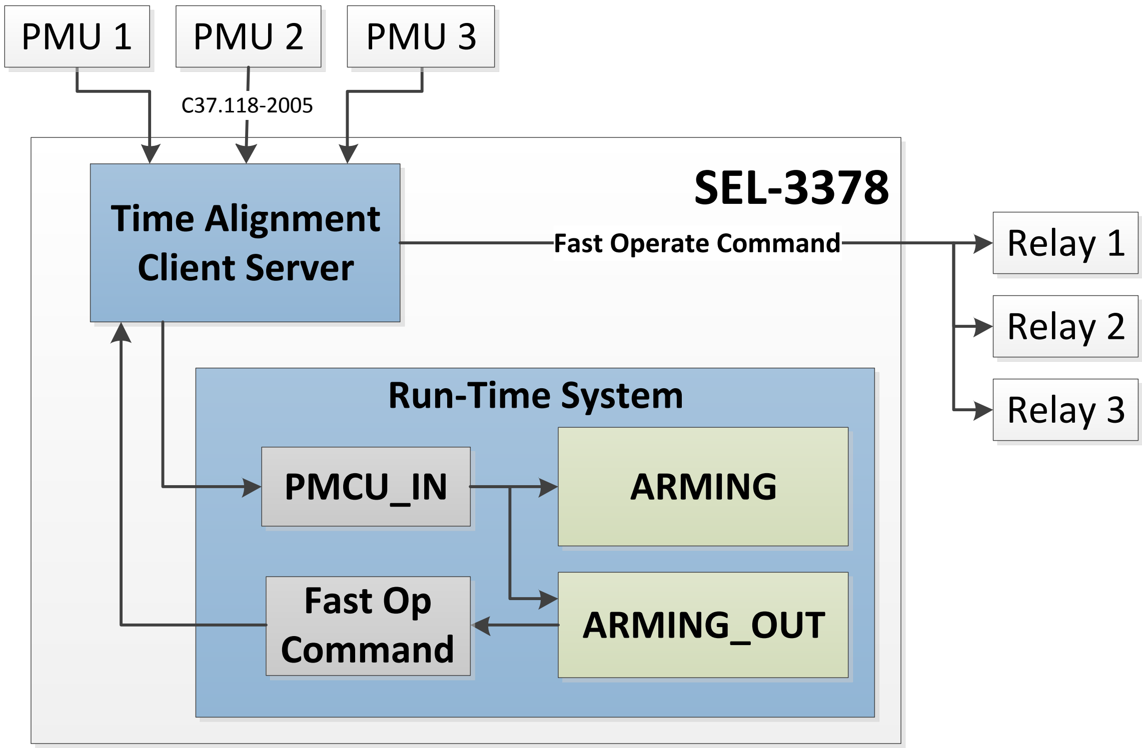


Figure 4.5: SEL-3378 System Architecture

The TCS running internally in the SEL-3378 sends the time aligned C37.118-2005 data to the Run-Time System (RTS) and up to 6 external C37.118-2005 clients. Of interest for wide-area protection applications was the RTS functionality, which allows for processing of the time aligned data using IEC 61131-3 standard logic. The next function block programmed into the SEL-3378 was the Phasor Measurement and Control Unit block (PMCU_IN). A PMCU_IN block was generated for each incoming PMU or PDC. Figure 4.6 shows the PMCU_IN block for PMU with ID = 1. This PMU was currently enabled, so the PDC expected synchrophasor data from the device.

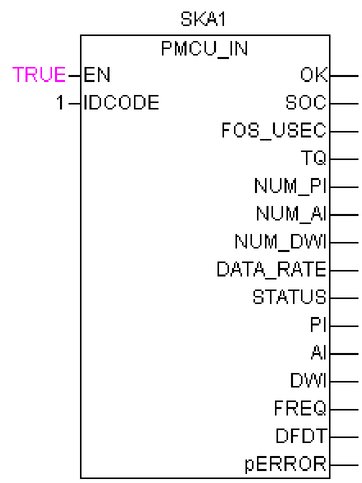


Figure 4.6: PMU Input Function Block

Each PMCU_IN variable (SKA1 in the block above) was set as a global variable, so that it could be accessed by other functions in the program. With the PMCU_IN and TCS functions created, the PDC was able to aggregate and time align PMU measurements and output a single C37.118-2005 data stream. The following sections describe the functions created for running the decision tree and sending a Fast Operate Message based on the state of the system.

4.1.3 Decision Tree and Arming Signal Implementation

Aside from the RTS performing the PDC tasks, an additional ARMING function was created to use the parsed PMUs packets for state determination using the decision trees. This function contained all the logic for the decision tree if-else statements as discussed in Chapter 2, and passed a STRESSED variable to the output function.

The Heavy Summer decision tree was programmed into the PDC using the testing threshold values. A digital bit, STRESSED, was used as the classification of the system. If STRESSED := TRUE, then the arming signal was be asserted; if STRESSED := FALSE, then the arming signal was restrained. The following structured text block was programmed into the SEL-3378 using SVP Configurator to model the decision tree output figure.

```
(*Heavy Winter Model Test Decision Tree*)
IF SKA1.PI[1].RE <= 44.2
THEN
   IF SKA2.PI[2].RE <= 4.21
   THEN
      IF SKA1.PI[1].RE <= 42.1
      THEN
         STRESSED := FALSE;
      ELSE
         IF SKA3.PI[2].IM <= 0.13
         THEN
            STRESSED := FALSE;
         ELSE
            STRESSED := TRUE;
         END_IF;
      END_IF;
   ELSE
       STRESSED := TRUE;
   END_IF;
ELSE
   IF SKA2.PI[2].RE <= 4.07
   THEN
       STRESSED := FALSE;
   ELSE
      STRESSED := TRUE;
   END_IF;
END_IF;
```

As discussed above, the ARMING function is where the STRESSED bit is set to TRUE or FALSE based on the values and their thresholds of phasor measurements. The STRESSED bit from the ARMING function was used to enable or disable function blocks in the ARMING_OUT function.

The first method explored for sending the arming signal to a relay was the SEL Fast Operate commands. This communications protocol uses a binary data stream over Telnet that an external device such as a relay uses to perform trip, close, set remote bit, or clear remote bit operations [29]. Since the adaptive protection scheme provides a signal to the relays for voting determination, the Remote Bit Set and Clear functions were used. The remote bit Fast Operate command has a total message length of 6 bytes, where the fourth byte into the message is the operate code for the external relay. This byte changes from 0x00-0x0F (hex) when the remote bit is cleared to 0x20-0x2F (hex) when the remote bit is set, for remote bits RB01-R16 respectively. Using the RB01, the operate byte in the stream was monitored for change from 0x00 to 0x20 for the transition from disable to enabled.

Figure 4.7 shows the function blocks used for setting and clearing the remote bit at the external device. The STRESSED bit and its binary inverse are sent to the ARM1 and DISARM1 blocks in the function. Therefore, when the system is classified as stressed based on the traversal of the decision tree, the FAST_OP_REMOTE_BIT_SET block will enable and set RB01 (REMOTE_BIT = 1) of the relay with ID of 1 (IDCODE = 1). When this occurs, the disarm function is simultaneously disabled due to the enable bit of DISARM1 being false. Conversely, when the system is classified as normal operating conditions based on the tree results, the FAST_OP_REMOTE_BIT_CLEAR block is enabled and RB01 is cleared.

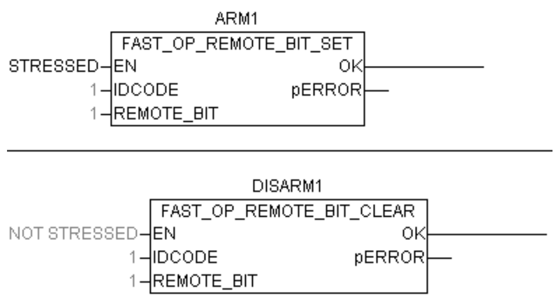


Figure 4.7: Fast Operate Control Function Blocks

Fast Operate commands are a very simple method for controlling remote devices of SEL manufacture. The only problem encountered was integrating this control signal into a system of non-SEL devices. The work in [17] discusses the use of the SEL-2515 Remote I/O Module in conjunction with the SEL-2812 Fiber Optic Transceiver/Modem to convert the Fast Operate messages to contact closure. The contact closure completes a DC circuit to close in a contact on a relay input contact terminal.

Other options also exist for communicating the arming bit to external devices. SEL also provides a patented Mirrored Bits communication technology that exchanges the status of eight internal logic points called Mirrored Bits, encoded in a digital message [30]. The bits sent from one device (TMB) are "mirrored" by the received bits of the remote device (RMB). A newer technology currently being developed by SEL is the use of the C37.118-2005 command frames as a carrier of the Fast Operate control signal, rather than Telnet, to expedite transmission of the signal. The latest SEL 421, 451, 411L, and 487V versions will decode the fast operate commands using this methodology. This is again an SEL-based technology but may be expanded upon if proven effective.

The Fast Operate commands were implemented for this protection scheme because of the simple and effective means of transmitting a control signal to other devices, and the proved use with manufacturers of other devices.

4.1.4 Task Configuration

The priority of tasks was an essential component of setting up both the TCS as well as the decision tree program within the PDC. The various programs created were grouped into tasks as shown in Figure 4.8. Three task groups were created: Configuration_Task, HighSpeed, and ArmingSignal. The Configuration_Task was set to run every 2 seconds due to the low priority of the tasks. The HighSpeed task was the PMCU_Assign task, which performed the PDC alignment functionality for reading in PMU data. This had the highest priority and ran at an interval of 4 milliseconds. Lastly, the ArmingSignal task performed the decision tree functionality and control output. Based on advice from an SEL representative, this task was run at an interval of 16 ms.

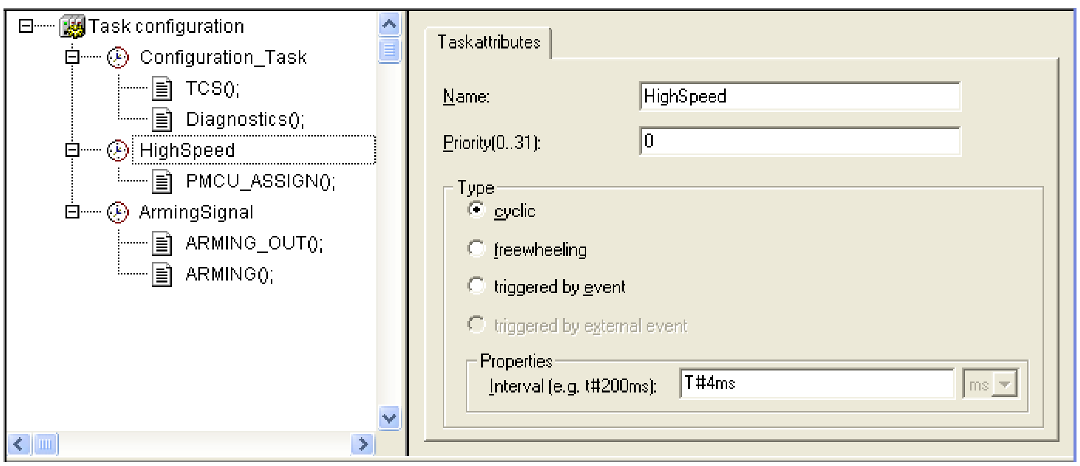


Figure 4.8: SEL-3378 Task Configuration

4.1.5 Hardware PDC Testing

A test procedure was devised to test all possible scenarios of the security/dependability adaptive protection scheme within the PDC. The focus of testing was to ensure that PMUs reporting the correctly monitored values to the PDC were parsed, the tree traversed, and the correct control action taken. The three PMU measurements of key locations on the system were used as the inputs and monitored using Wireshark. SVP Configurator was used to observe the STRESSED arming signal state in real-time, and the relay receiving the control signal over Telnet communications recorded SER Events to ensure that the signal was received by the device.

Six distinct states were created using combinations of the voltage and current available to test each classification for the Heavy Winter Model decision tree, starting from the bottom left classification of the tree moving right. The Doble F6150 was used to generate positive sequence phasors by setting the magnitude and phase angle of phase quantities. Values were chosen such that the real and imaginary components, as required by the decision tree, fell above or below the thresholds being tested. Important to note was that the Doble had a -92.8° angle offset on both the current and voltage channels that was accounted for during testing. The Doble Control Panel was used for setting the steady state conditions, as shown in Figure 4.9.

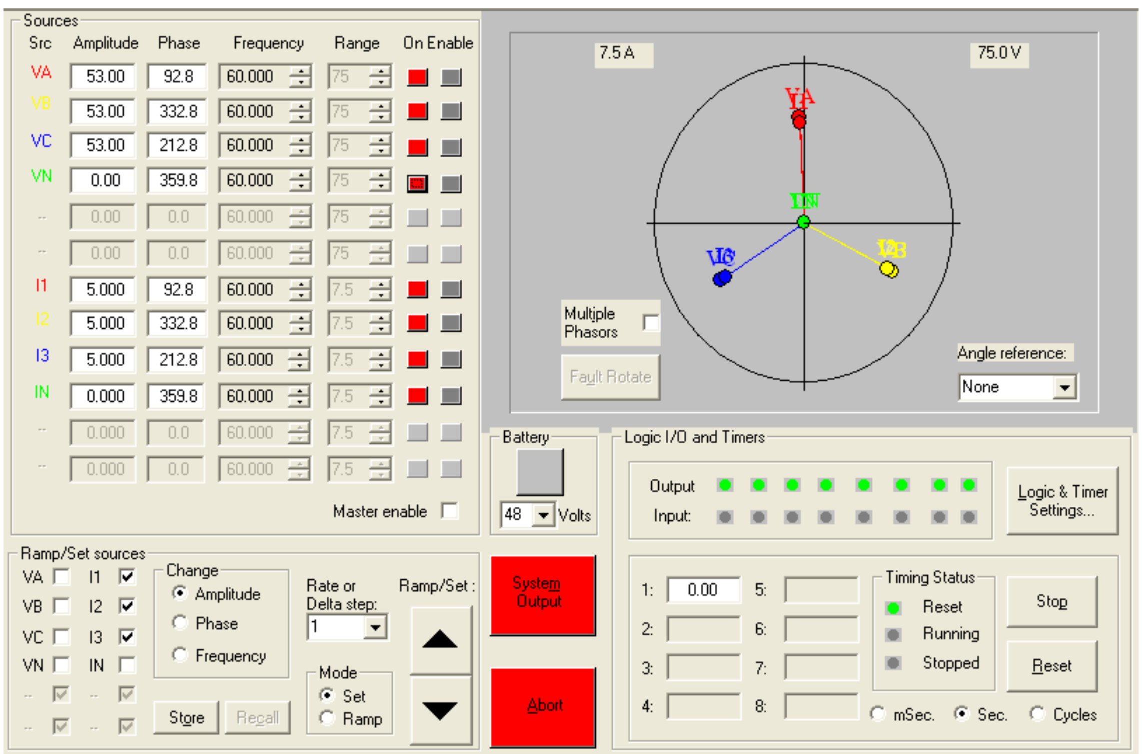


Figure 4.9: Doble Control Panel

The different voltages and currents applied to the PMUs are provided in Table 4.1. It was found that both the PDC Boolean STRESSED signal and the Remote Bit

(RB01) in the relay receiving the control signal were set correctly for all expected output conditions. It was concluded based on this test that the PDC was functioning as expected for implementation of the decision tree for a set of steady state conditions.

Further work was performed to ensure that the STRESSED signal was being set accordingly by the correct classification in the tree, as an extra measure of functionality. Output variables, OUT1-6, were assigned to each classification terminal in the tree, and the test was repeated. As expected, the PDC functioned properly, only setting the correct output for a given set of system conditions.

Table 4.1: Hardware PDC Functionality Test Results

State		Inputs						Outputs		
		Doble Voltage			Doble Current					
#	Op. Cond.	V	V_r	V_i	I	I_r	I_i	PDC Out	RB 01 Pickup	OUT1-6 Check
1	Safe	33/0°	33	0	3/0	3	0	FALSE		Y
2	Safe	43/0°	43	0	3/0	3	0	FALSE		Y
3	Stressed	43/0°	43	0	3/10°	2.95	0.52	TRUE	X	Y
4	Stressed	33/0°	33	0	4.4/0°	4	0	TRUE	X	Y
5	Safe	53/0°	53	0	3/0°	3	0	FALSE		Y
6	Stressed	53/0°	53	0	5/0°	5	0	TRUE	X	Y

These tests confirmed expectations of how the PDC would operate during normal conditions, i.e. steady state and no bad data. Transient analysis of the PDC is embedded in the Chapter 5 discussion on latency testing, which also requires correct functionality. The topic unexplored thus far was how the PDC would react to uncertain WAMS conditions such as a missing PMU measurements or a PMU with invalid IRIG-B.

The phasor measurements in a C37.118-2005 data frame for a given PMU are each preceded by a status word, STAT, that provides critical information about the quality and condition of the data as well as the measuring device. The STAT word originates at the PMU, but may be altered by other devices such as the PDC. Mainly of interest in the STAT word are the most significant bits, Bits 15-13. Bit 15 is the Data Valid bit, which is set to 1 when the PMU deems the data invalid. Bit 14 is the PMU Error bit which is set when the PMU experiences an internal error such as an A/D converter error [8]. Bit 13 is the PMU Sync Error which is set to 1 when the PMU experiences a loss of synchronization; this can occur if either the time source fails or if the IRIG-B input fails.

The decision tree implementation was tested on the SEL-3378 with bad data to determine how it would react to these types of situations. In today's power system, with the increasingly complex and growing number of devices and networks, it is not uncommon to lose a single device in the field. Measures must be taken to allow for these single failures while repairs or replacements occur. Two tests were performed to "force" the bad data: remove the IRIG-B input to a PMU and entirely remove a PMU from the system.

A PMU was removed from the network by disconnecting the Ethernet connection to the hub. As expected, these packets did not arrive at the PDC, and the PDC responded by filling the phasor data with "0" in the PDC output packets. For the decision tree application, this posed a slight problem because the tree could misclassify the correct state of the system. For example, the voltage and current applied to the PMUs placed the operating condition into State 4 (OUT4 := TRUE). PMU2 was removed from the network, so the PMU2.IR measurement needed in the tree compared its phasor of "0" to the threshold value. This changed the operating state of the system to either State 2 (OUT2 := TRUE) or State 3 (OUT3 := TRUE) based on the measurement of PMU3.II. This example demonstrates how the reported operating condition of the system can be easily altered based on which PMUs are actually reporting phasor data. It was also noted that based on the structure of the Heavy Winter decision tree, there was no possible way of losing a PMU measurement and going into a STRESSED condition from a SAFE condition; only the opposite scenario held.

A PMU lacking a source of time synchronization reported its phasors as expected, but acknowledged this problem in the STAT word. The PDC picked up this error and filled the respective PMU's phasor measurements with "0". This caused significant problems with the decision tree functionality of the device. Voltages and currents were applied to the system using the Doble such that the decision tree fell into classification 6 (far right decision node). The IRIG-B was disconnected from PMU1 and the results were observed: no change in the STRESSED output signal. This is because the traversal of the tree moved from classification 6 to classification 4, still a stressed condition. The IRIG-B was replaced for PMU1, but then disconnected from PMU2: change in

STRESSED signal. The tree traversal again changed, but from classification 6 to classification 5. The origin node moved the traversal to the right of the tree, but the next decision was using an erroneous “0” as its input. Therefore, the tree misclassified the operating condition of the system when bad data was present.

The fix to this problem involved adding simple logic to the decision tree to determine the state of the PMUs prior to traversing the tree. If any single measurement device reported a STAT bit that was not congruent with a functioning system, the decision tree was bypassed, and the STRESSED output was set to FALSE. Therefore, the voting scheme application would perform as normal with any one device or protection scheme allowed to send a tripping signal. The structured text below shows how this additional layer of security was implemented in the tree.

```
(*Heavy Winter Model Test Decision Tree*)
IF SKA1.STATUS.PMCU_OK  AND SKA2.STATUS.PMCU_OK AND SKA3.STATUS.PMCU_OK
THEN
        [Decision Tree Implementation…]
ELSE
        STRESSED := FALSE;
END_IF;
```

The results obtained from hardware PDC testing using the SEL-3378 showed positive results. The device was able to fully implement the decision trees using synchrophasor measurements from PMUs. The state of the system was classified as stressed or safe and an output bit was set within the PDC. This bit was sent to the set of relays using Fast Operate Messages for SEL devices, and additional hardware was used to communicate between the SEL-3378 and non-SEL relays. Each operating condition was tested as well as injection of bad measurements in the form of lost PMU data or bad IRIG-B signal.

4.2 Software PDC Implementation

The Tennessee Valley Authority (TVA) developed a phasor data concentrator in 2004, and made it available as open-source software free to download from [25]. This software, openPDC, has been proven operable with over 120 PMUs on a single computer platform; this specification depends on the computer's hardware limitations and capabilities. openPDC is compatible with most synchrophasor protocols and a number of databases for data storage. Database storage of synchrophasor data, statistics data, and other user-defined data allows for retrieval from other devices with the capability to communicate with a server database. TVA PMU Connection Tester [31] is used in conjunction with openPDC to provide a simple means of connecting and communicating with phasor measurement units or other phasor data concentrators on the network.

This section details implementation of openPDC as a viable option for the adaptive voting scheme. It also describes how the full protection scheme was devised and applied.

4.2.1 openPDC Implementation Network Architecture

The configuration used for testing openPDC as a viable option for the adaptive voting scheme is shown in Figure 4.10. Three PMUs were aggregated and time aligned at the computer running openPDC. C37.118-2005 protocol over TCP connection was used to send the data from the PMUs to the PC. The real-time determination of operation state of the system was performed in the PC using C# programming. A bit was stored in an SQL database reflecting this classification from the decision tree. A Rockwell Automation PLC was used to retrieve the arming bit in the database on the server and determine whether to vote or execute the relays as they normally would be deployed [32]. Three SEL relays were hard-wired to the PLC inputs, communicating their trips signals based on instantaneous overcurrent faults. The PLC performed the actual 'breaker tripping' in the circuit regardless of the state of the Arming signal. A visual display was used to ensure that the scheme operated correctly as designed.

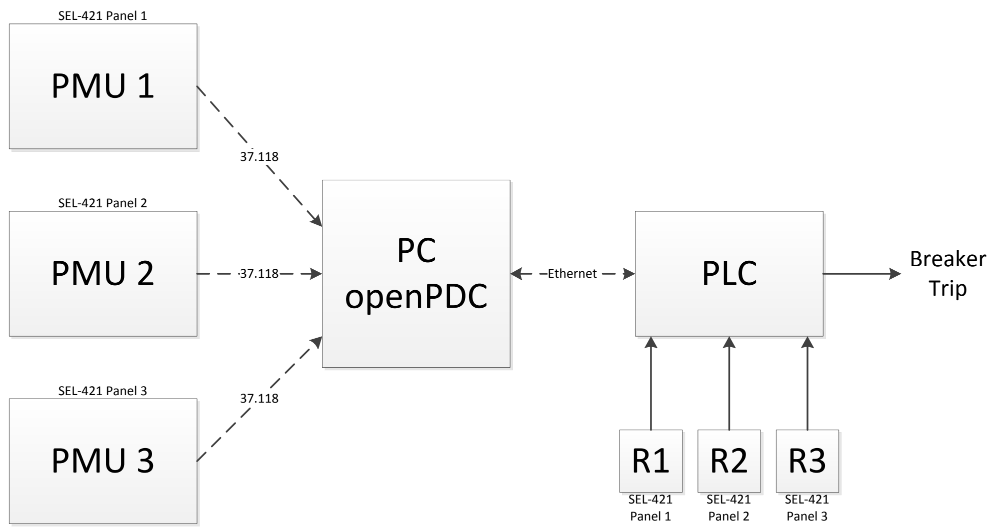


Figure 4.10: Software PDC Network Configuration

Due to available resources at the time of testing, the actual Heavy Winter and Heavy Summer trees were altered slightly. Only one synchronized source was available to a single PMU such that only one PMU could report phasors with exact real and imaginary current values. Other 3-phase sources were applied to the other PMUs but they were not synchronized to UTC and did not have a true 60Hz frequency. Therefore, the phasors would rotate over a period of time, making it impractical to test the actual HW/HS decision trees. Figure 4.11 shows the Heavy Winter and Heavy Summer trees used for testing purposes. A classification node in the tree was selected to place the third PMU with exact real and imaginary components such that the functionality could still be tested. It was inferred that the rectangular form decision would function identically regardless of where it was placed in the decision tree.

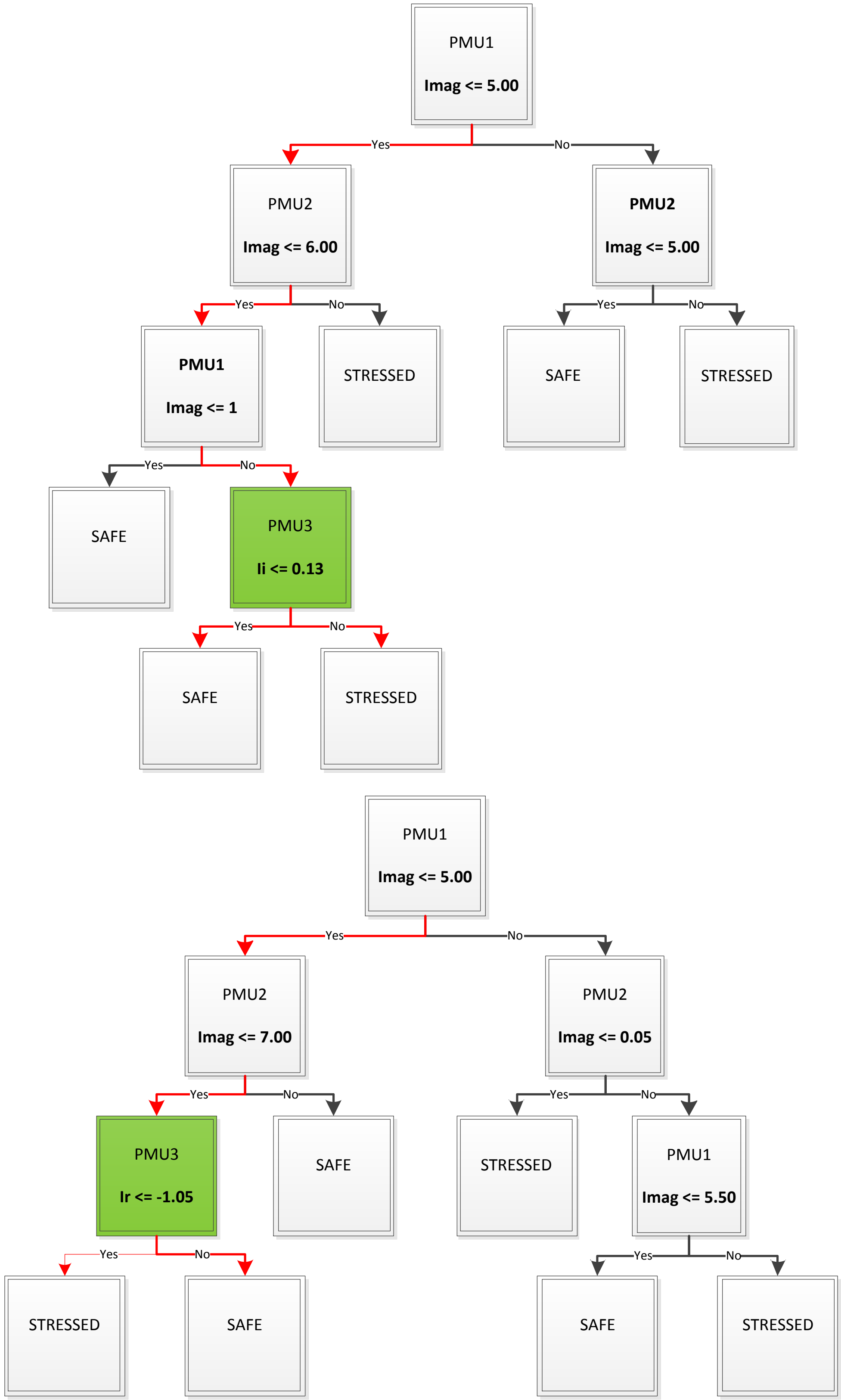


Figure 4.11: Software PDC HW and HS Testing Trees

4.2.2 openPDC Setup and PMU Configuration

Following installation, open PDC was configured for three separate PMU inputs. The PMU Connection file and Configuration files were extracted from PMU Connection Tester, and each PMU was configured to send at least positive sequence current and voltage phasors at 60 frames per second. A historian database was set up to archive PMU data following alignment in openPDC. Each individual measurement made by an input PMU was given a unique identification number within the database such that only the necessary measurements for the decision tree could be extracted and provided to the functions requiring the data.

openPDC is built on three types of adapters: input, action, and output adapters. The input and output adapters allow openPDC to communicate and remotely configure I/O such as subscribers/publishers, or historians and real-time data. The action adapters, on the other hand, perform functions or actions on the data once it has been time aligned. For implementation of the decision tree within openPDC, the action adapter layer was the only layer of importance. Figure 4.12 shows the openPDC architecture within the computer, and the connecting devices communicating with the computer. Upon time alignment, the phasor data was stored in a MySQL database, Primary Phasor Archive (PPA) [33]. The measurements were passed to the PPA at 30 frames per second. Each frame was then sent to the adaptive protection code running the decision tree. The results of the decision tree were then pushed to a Microsoft SQL database, storing the Arming signal [34].

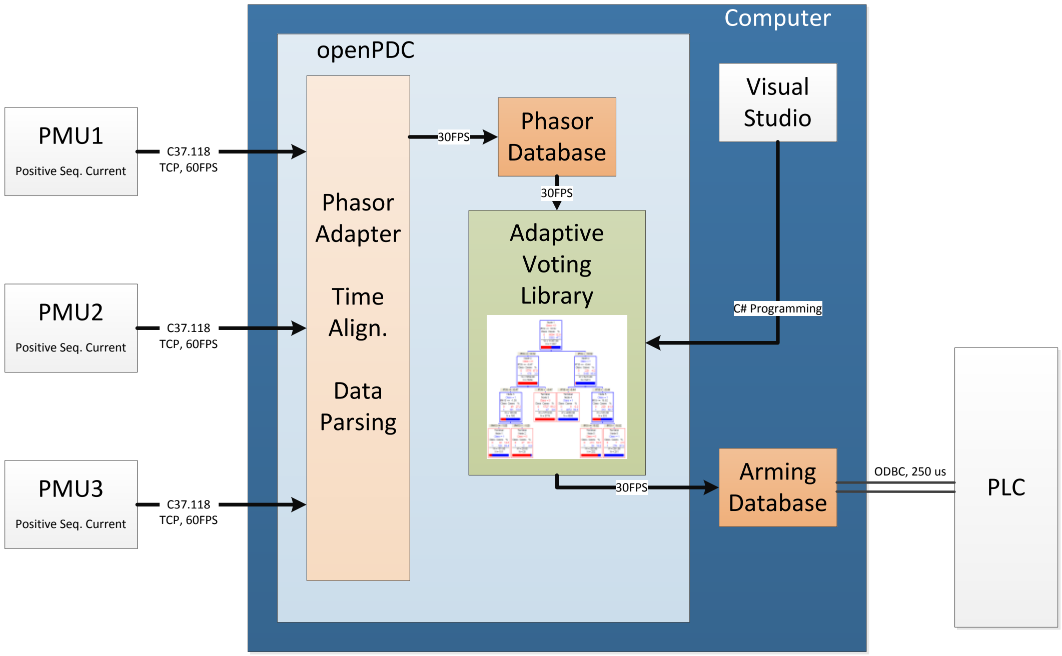


Figure 4.12: openPDC Software Architecture

4.2.3 Arming Decision Tree Programming

Implementation of the decision tree running in openPDC was performed using C# programming in Visual Studio 2010. A Visual Studio library was created that consisted of code that the openPDC would execute each time a new measurement frame was received. Phasor measurements stored in frames were called using the PublishFrame() function from the TimeSeriesFramework library. Within the PublishFrame() function, a for loop was used to extract all measurements from the current frame. These measurements were stored in temporary variables. Since the decision tree requires real and imaginary components of the phasors, or rectangular form, the data was converted from polar to rectangular since openPDC reported the phasors in polar form.

```
I3imag = I3mag*Math.Sin(I3phase*pi/180);
I3real = I3mag*Math.Cos(I3phase*pi/180);
```

Both decision trees were implemented in the same function. Each tree was built using if-else programming logic, and the State was stored as either “Safe” or “Stressed”. The Heavy Winter tree is shown below for demonstrative purposes. Both trees are provided in Appendix B. The trees were dynamically switched based on the current models being used for operation of the power system. A selector switch was implemented that provided the user of the program the ability to easily switch between trees. This was done by providing a 24VDC signal to the input of a PMU sending synchrophasor data. The input was programmed into the C37.118-2205 data frame as a digital word. openPDC was then able to parse the incoming frames to read the status of this word. The action adapter library implemented used this value to determine which decision tree to perform.

```
        OnStatusMessage("Decision Tree: Heavy Winter");
        if (I1mag <= 5.00) {
            if (I2mag <= 6.00) {
                if (I1mag <= 1.00) {
                    State = "Safe";
                }
                else {
                    if (I3imag <= 0.13) {
                        State = "Safe";
                    }
                    else {
                        State = "Stressed";
                    }
                }
            }
            else {
                State = "Stressed";
            }
        }
        else {
            if (I2mag <= 5.00) {
                State = "Safe";
            }
            else {
                State = "Stressed";
            }
        }
    OnStatusMessage("System Operating Condition: {0}", State);
    Update(State);
```

Following classification of the operating condition of the system, the data was stored in an SQL server for later use. A connection to the SQL server was established and the State variable was updated within the database table. The table consisted of a single important cell called ‘vote’ that was either set to 0 (meaning “Safe”) or 1 (meaning “Stressed”). This bit was retrieved by the PLC and is discussed later in this chapter. Each Update() function within the library created would open a connection to the table, update the table based on the classification of the system from the decision tree, then close the connection to the server.

4.2.4 Arming Signal Extraction from SQL Server

A Rockwell Automation ControlLogix 1756-L73 Controller was used as the PLC retrieving the vote signal from the database and performing the breaker trip logic based on the state of the vote signal. The ControlLogix controller is part of the Logix5000 family of Rockwell Automation controllers, and provides I/O modules linked by a common chassis or backplane [35]. The controller used was equipped with both digital input and output cards, each with 16 points at 5VDC TTL signal levels. Also included was an EtherNet/IP communications module for connecting to the network where the SQL Database Server resided. The 1756-L73 Controller is capable of 100 programs per task, and 32 tasks. It supports up to 500 connections to different points in the system, and also provides Ethernet communication [32]. Programming options include relay ladder logic, structured test, function blocks, and structured flowchart. All programming of the PLC was done in ladder logic.

The PLC served two critical purposes in this system: query the SQL database residing on the computer running openPDC, and perform the logic based on the operating condition results obtained. Based on the integer value of the 'vote' bit retrieved, the PLC would perform 2-out-of-3 voting if the bit was 1 and normal operation for relay tripping if the bit was 0.

Query of the SQL database was done using FactoryTalk Transaction Manager. Two connectors were established between the SQL database and the ControlLogix controller. A connection was established between each of the two ends of the tunnel for passing data, and the 'vote' bit was read out of the database. Based on all the testing performed, all transactions between both ends of the connection performed at 100% passing rate for successful transactions. Upon retrieval of the bit by Transaction Manager, the integer value obtained was stored in a Vote tag in the controller itself.

In essence, the database was continuously queried by Transaction Manager. Each query would update the Vote tag within the PLC itself, and the PLC would use this integer value obtained to determine whether the scheme should vote or perform normal operation.

4.2.5 PLC Voting Logic

Programming the PLC was performed using ladder logic in RSLogix5000; RSLogix5000 is the software that accompanies the ControlLogix controller [36]. A set of ladder rungs were sequenced such that the PLC would determine what the operating condition was based on the decision tree output, and perform voting or normal operation for relay tripping. Figure 4.13 shows the ladder logic programmed into the PLC. Rung 1 determines the state of the VOTE tag. If the VOTE tag is set to integer value 1, then the ARM bit goes true. Rungs 2 and 3 use this ARM bit to determine which rung to execute. If ARM = TRUE, then Rung 2 will execute and if ARM = FALSE, then Rung 3 will execute. When ARM = TRUE, the PLC performs voting of the relay trip signals using a parallel combination of 2 out of 3 relays trips in series. For any 2 out of 3 tripping scenario, the series elements will assert the input contacts and the VOTE_TRIP signal will go to TRUE. On the other hand, if ARM = FALSE and Rung 3 is executed, any individual relay input contact could complete the sequence and the NON-VOTE_TRIP element would go TRUE. Rung 4 shows that conditions in which the BREAKER TRIP output would assert. These include the VOTE_TRIP, NON-VOTE_TRIP, and LAMP_TEST contacts. The BREAKER TRIP output is the physical output contact that would apply a trip signal to the breaker itself. The LAMP_TEST input was used for testing to assure that the breaker would trip when required.

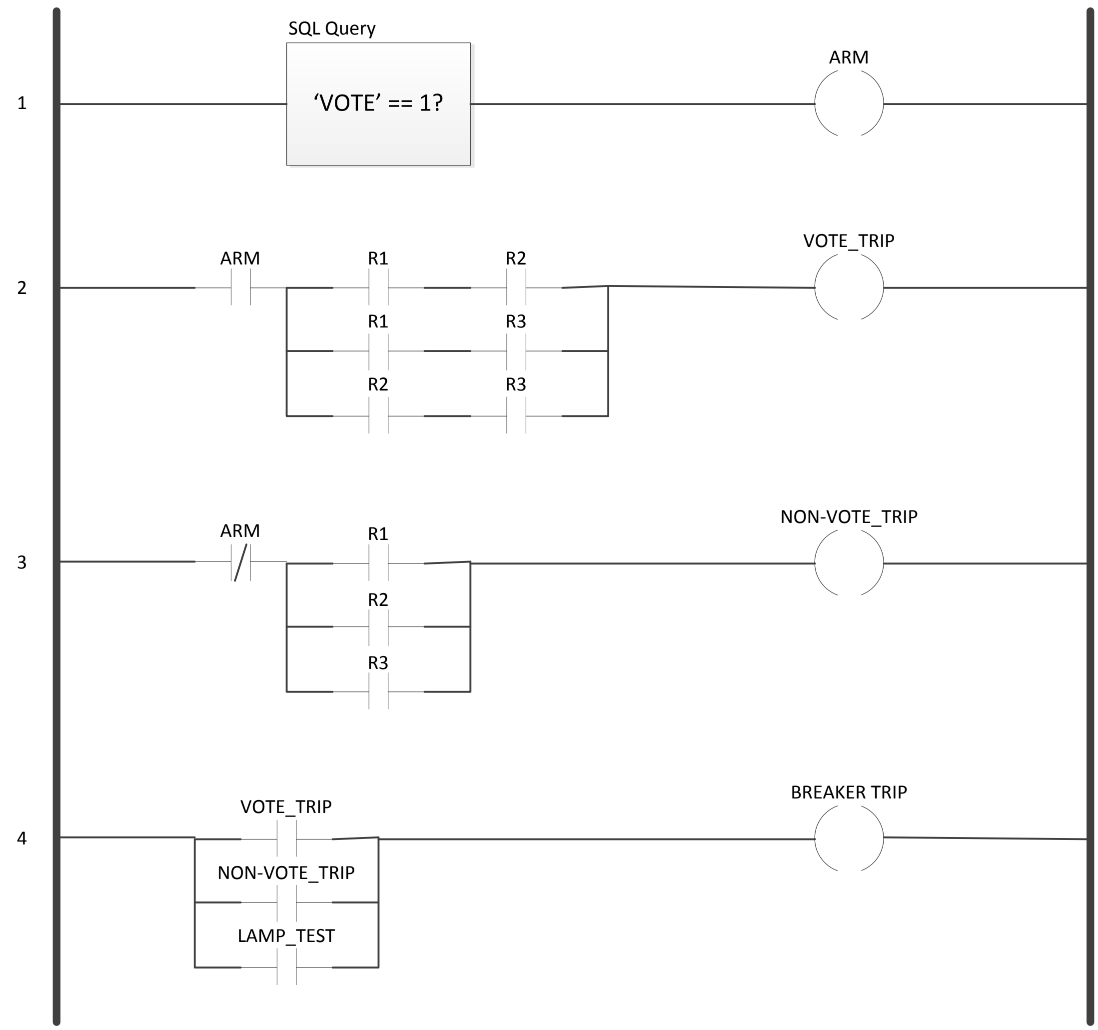


Figure 4.13: PLC Voting Ladder Logic

4.2.6 Software PDC Testing

The implementation of the decision tree using a software-based PDC, openPDC, was performed similarly to testing of the hardware PDC. Prior to testing if the voting scheme functioned properly in the PLC, the openPDC decision tree was tested to ensure that the outputs were operating as expected. Table 4.2 shows the results for functionality testing of the Heavy Winter and Heavy Summer decision trees in openPDC. For each operating condition the PDC Console provided the correct output, the database contained the correct value for 'vote', and the PLC arming signal was set to TRUE. This triplicated the confirmation that the tree was operating as expected.

Table 4.2: Software PDC Functionality Testing Results

	State		Input Currents				Outputs	
	#	Op. Cond.	I1mag [A]	I2mag [A]	I3r [A]	I3i [A]	PDC Console	PLC Arming
Heavy Winter	1	Safe	0.5	3	3	0	vote=0	FALSE
	2	Safe	3	3	0	-3	vote=0	FALSE
	3	Stressed	3	3	0	3	vote=1	TRUE
	4	Stressed	3	6.5	3	0	vote=1	TRUE
	5	Safe	5.5	3	3	0	vote=0	FALSE
	6	Stressed	5.5	5.5	3	0	vote=1	TRUE
Heavy Summer	1	Stressed	3	3	-3	0	vote=1	TRUE
	2	Safe	3	3	3	0	vote=0	FALSE
	3	Safe	3	8	3	0	vote=0	FALSE
	4	Stressed	5.5	5	3	0	vote=1	TRUE
	5	Safe	5.5	8	3	0	vote=0	FALSE
	6	Stressed	5.5	8	6	0	vote=1	TRUE

The software-based PDC application of decision trees for the adaptive voting scheme focused on integration of the trees into openPDC as well as physical hardware configuration for the relay voting. The decision trees were both realizable within the openPDC platform itself using C# programming. Communication between the computer running openPDC and the PLC performing the trip logic was a critical component of the system. A Rockwell Automation ControlLogix controller was used to query the SQL database set up on the computer to retrieve the classification of the state of the system based on the PMU measurements. Regardless of a stressed or safe operating condition, the PLC always had responsibility of tripping the breaker whether voting or performing normal operation. Each relay was hard-wired to the PLC, and ladder logic was programmed into the device to coordinate relay operation and breaker tripping.

Chapter 5

Testing Latency of Wide Area Measurement and Control Systems

Numerous methodologies can be designed and implemented for testing data latency, including software simulations, laboratory testing procedures, and field measurement collection. Ultimately the exact measure of latency in a wide-area system is best determined once installed, although an understanding of latency prior to implementation helps determine the limitations for protecting the power system using wide-area measurements. The procedure discussed in this chapter uses laboratory testing to determine latency associated with WAMS measurement and control.

5.1 Required Components

Latency testing performed in the lab used a number of additional devices to the WAMCS components discussed. A description of these devices is provided for clarification on the capabilities of these devices and the reasons for using them.

5.1.1 GPS Time Synchronized Clock

The synchronized time source used for acquiring Universal Time Coordinated (UTC) was the SEL-2407 Satellite-Synchronized Clock. The SEL-2407 was used to provide the high accuracy timing of ± 100 nanosecond average, ± 500 nanosecond peak accuracy [37]. Inter Range Instrumentation Group mod B, also known as IRIG-B, was output from the SEL-2407 to the WAMS devices as well as the PC through BNC cable. The input signal used for latency testing was the 1 pulse-per-second (1PPS) acquired from the SEL-2407 clock. The 1PPS signal indicates the start of a second, and is the most accurate signal acquired from the GPS receiver [8].

5.1.2 Doble F6150 Power System Simulator

The Doble F6150 is a test simulator used primarily for relay and protection scheme testing [26]. It provides six independently controlled, direct-coupled sources, each rated at 150VA at 0-30A. These sources were used as 3-phase voltage and current signals. It also performs steady-state and transient simulations for testing accuracy, performance, and functionality of power system devices. The F6150 allows for user programmed simulations using COMTRADE files. These files were generated using a MATLAB program to create the signals and a converter file to convert the signals to COMTRADE format. The files were uploaded to the F6150 as playback files, which were executed at a user-defined time.

The key benefit of the Doble F6150 for latency testing was the time-synchronization associated with generating the signals. The GPS accuracy using the G6895 Antenna and Receiver was +/- 50 nanoseconds. The start of the test was set by the user using the Doble ProTest software package, which usually occurred at the top of a minute. This ensured that steps, ramps, etc., occurred at the correct time intervals and could be traced to an accurate time. This allowed for accurate comparison of input versus output of WAMS devices.

5.1.3 Wireshark Packet Analyzer

Open source network packet analyzer software, Wireshark, was used for examining the incoming packets of PMU and PDC data [38]. Wireshark measured all packets on the network that the PC running the application was connected to. The data within each packet was examined and decoded to find, for example, the TOS packet.

Wireshark was used for capturing Ethernet packets from the network devices for testing latency of WAMS components. Prior to testing the components, Wireshark itself was tested for precision during a capture. Spirent SmartBits Traffic Generation and Analysis Test Tool was used to generate packets for Wireshark to analyze [39]. Based on the loading percentage provided to the SmartBits Tester, 1480 total frames were generated over the course of 20 seconds. The frame size was set to match the packet size; therefore, an average of 74.4 packets per second was generated by the SmartBits Tester.

This resulted in an expected inter-packet time of 13.44086 ms. This time was chosen because of its proximity to a 60Hz reporting rate of 16.667 ms; setting the inter-packet time exact to this value was not important. During the 20 second generation of packets from the SmartBits Tester, Wireshark captured approximately 15 ms of packets for analysis. Figure 5.1 shows the results of this capture, both in a time domain form as well as a statistical histogram.

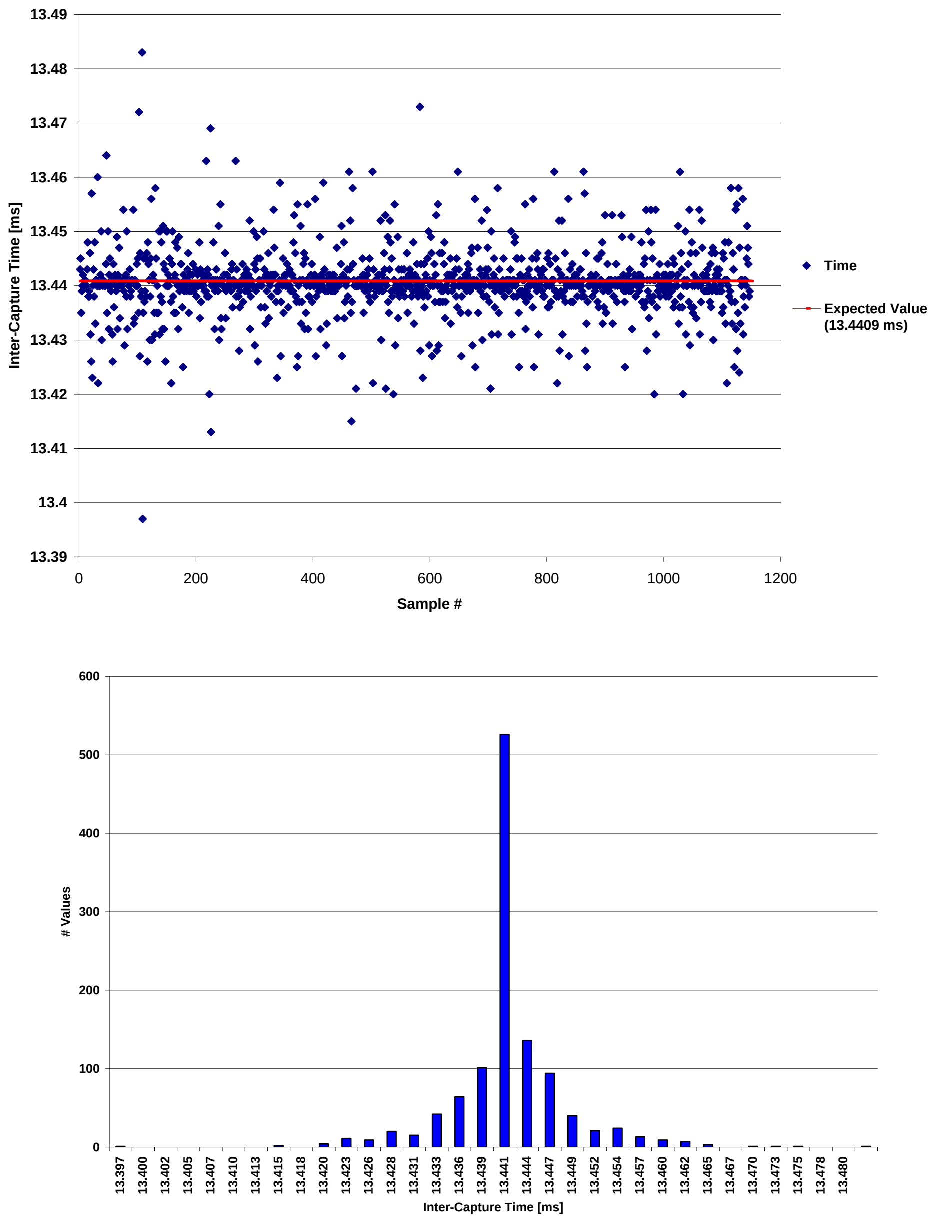


Figure 5.1: Wireshark Inter-capture Precision

Wireshark had a maximum deviation of approximately 0.042 ms, or 42 μs, from the expected value of 13.4409 ms. For the purposes of measuring WAMS device latency, this error was well within an acceptable range for using Wireshark as a means of capturing latency. This error was small enough that it could not be attributed to inaccuracies in Wireshark or within the SmartBits Tester itself.

5.1.4 BPA StreamReader

StreamReader is a program created at the Bonneville Power Administation (BPA) for reading, decoding, and archiving synchrophasor data [40]. This application runs on a PC and collects the incoming phasor packets on the given ports. It requires an .ini file to decode the packets correctly, and displays the packets as they are collected in real-time. StreamReader also stores 5-minute segments of data into a temporary file for a given period of time for offline analysis of the data. Therefore, running a Doble F6150 playback file at the top of a 5-minute interval provided a full capture of the Doble test performed on the PMU. StreamReader was used primarily for PMU connection validity and offline PMU response testing.

5.2 Testing Methodology

As mentioned, the 1PPS is the most accurate signal available from the synchronized timing source. This signal, coupled with a clock accurate to 1μs, yields a combined error of ± 2 microseconds [8]. It therefore was integral to use this signal as a reference point for measuring latency. The WAMS components send C37.118-2005 protocol packets of data time stamped with a UTC time. Measurement of latency using the 1PPS signal involved using the packet corresponding to the top of the second (TOS). At the TOS, a C37.118-2005 packet is sent with a fraction of second (FRACSEC) value of 0.000000. Ideally, this packet would arrive at the collection device at the same time, but this is of course not the case. The latency of the device or collection of devices is the time difference between the reported time by the device and the time it was actually acquired by the PC.

$$Delay_{Device} = T_{Reported} - T_{Actual}$$

All latency testing was performed using a reference signal; attention was given to determine what the correct reference should be and why. For example, to measure PMU latency the delay is between when the PMU reports the TOS and the actual TOS. For a PDC, similarly, the latency is the delay between the last PMU packet to arrive at the PDC and the PDC output of the packet with the same time stamp. The PMU latency requires a time synchronized collection device while the PDC latency does not. This is because the PDC latency is relative to PMU time reported rather than real-time UTC.

For testing of a full WAMCS, a highly accurate means of generating signals and collecting the time reported by various devices was required. Both these capabilities were available through the Doble F6150 and the SER Event Reports on microprocessor relays. Each relay can be configured such that a given event will trigger an assertion or deassertion of a digital point. For example, when the relay observes the trip signal from the PDC, the SER will record an event. Since all components are time synchronized to UTC through the same SEL-2407 clock, the time between the provided input to the Doble and the assertion of a SER event report was the latency of the whole system.

Important to note is the network configuration used for testing latency. Since the network in which the devices reside can have an impact on the timing of the synchrophasor packets, a symmetric network was used to minimize these delays. A star-connected Ethernet network was used for all latency testing because the delays required for the PMUs and PDCs reporting their data frames to the computer observing the traffic would be equal. Also, since the network distances were equal and the switch used for testing was capable of handling all traffic, we could neglect network latency. Figure 5.2 shows a configuration of the latency testing network.

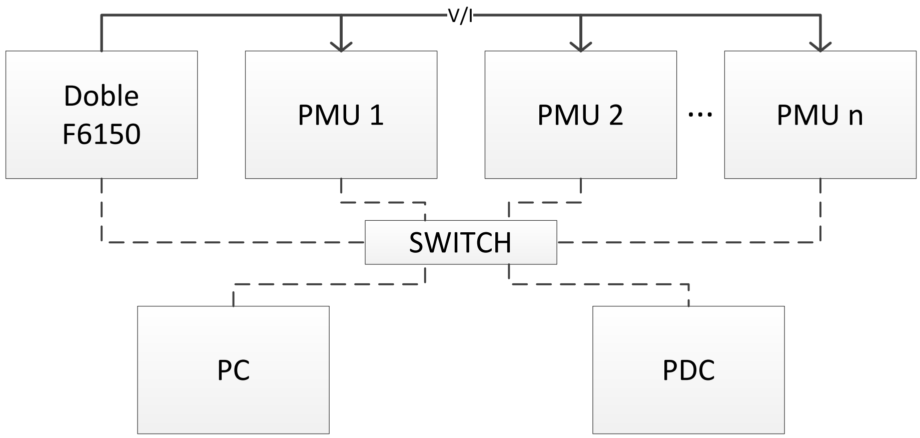


Figure 5.2: Latency Testing Network Configuration

5.3 Phasor Measurement Unit Testing

The PMU overall latency consists of both the response time and the reporting time. During steady-state conditions in the power system, the response time has little effect on the overall latency while the real-time collection and computation of phasors has an inherent delay. During transient conditions, the reporting delay is still apparent but the response delay also becomes a factor that must be taken into account. Both forms of latency were explored in the work presented.

The phasor measurement network implemented was set up as shown in Figure 5.3 below. Playback files were programmed into the Doble F6150 using a MATLAB program and COMTRADE converter. These files were provided by Tony Faris, Electrical Engineer at the Bonneville Power Administration, for use for latency testing (discussion of playback file follows). The signals generated by the Doble were applied to the PMU input terminals, simulating secondary current and voltage quantities. The PMU reported synchrophasor data at 60 frames per second over an Ethernet TCP connection. The data was sent to a computer, where BPA Streamreader was used to capture and archive the synchrophasor data for offline analysis.

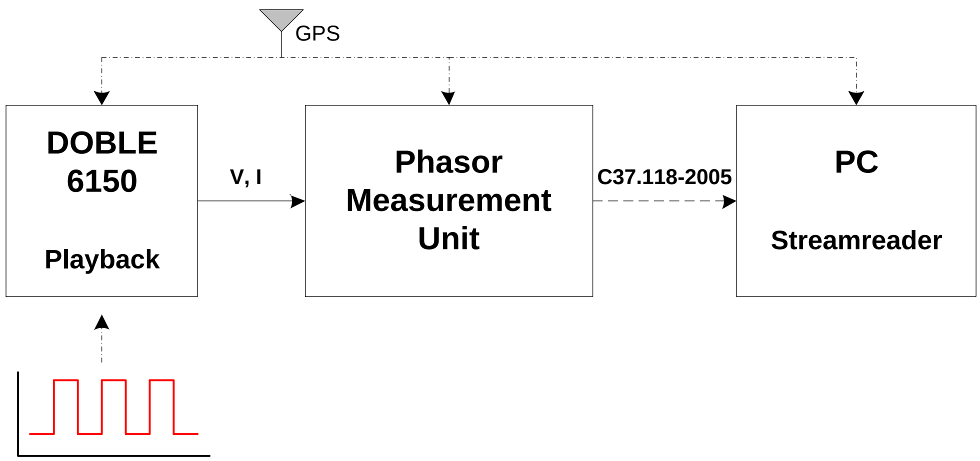


Figure 5.3: Phasor Measurement Unit Testing Network Architecture

Response time was measured based on a step input to the PMU at a given time *t=0*. Following the C37.118-2005 standard, the response time was measured by applying a positive 10% step in magnitude with the input signal at nominal magnitude and rated frequency. The response time was defined by the standard as "*the time between the instant the step change is applied and the time tag of the first phasor measurement for which the TVE enters and stays in the specified accuracy zone corresponding to the compliance level (1% TVE)*" [8]. This definition was altered slightly due to the nature of the adaptive protection system to "*the time between the instant the step change is applied and the time tag of the first phasor measurement for which the magnitude of the phasor reaches or surpasses the 10% step change*". The thresholds set by the decision tree application within the PDC use basic values and are not interested in whether the devices are within 1% TVE.

The Doble 6150 was programmed to generate a three-phase signal at nominal voltage experiencing a 10% step change. The performed test generated 17 step inputs, each 1/16 of a 60 Hz cycle apart. This gave a step input occurring at 1/16 intervals throughout the measurement window to fully observe the step response of the PMU. The first step occurred at the top of the second, the second step at 1/16 of a cycle away from the top of the second, etc. Appendix C provides the .m file corresponding to the amplitude step test performed. With a step occurring at the beginning of the phasor measurement window, the PMU responded as shown in Figure 5.4.

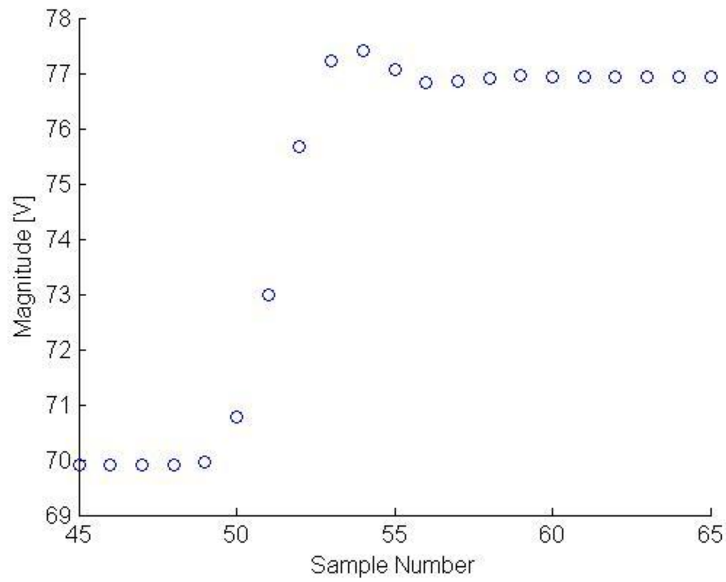


Figure 5.4: Beginning of Window Step Response

Similarly if the step occurred 1/16th of a cycle later in the measurement window, the PMU response would be shifted slightly as shown in green in Figure 5.5.

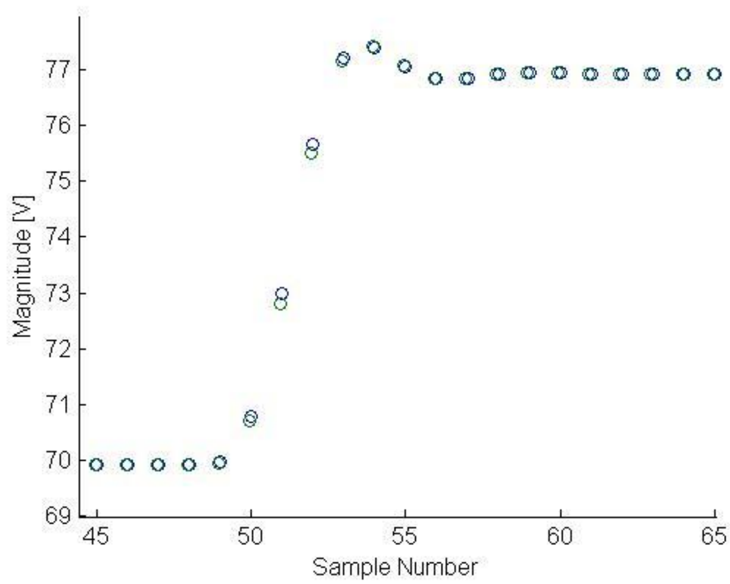


Figure 5.5: Beginning of Window + Shift Step Response

By combining each of the 16 steps for the 16 different locations of the step occurring within the test file, a better response of the PMU was determined. Overlaying each step response gives a detailed visualization of the response of the device. The complete process is shown in Figure 5.6 (code in Appendix C).

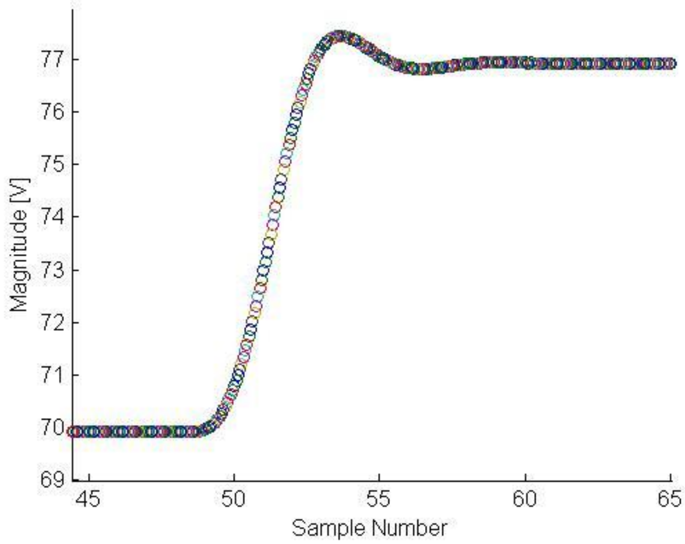


Figure 5.6: Phasor Measurement Unit Step Response

The accumulation of data points for the step test can be represented as a continuous function in time. This response and the step input are shown in Figure 5.7. Based on these results, the PMU has a response time, $t_{response}$, of 33 ms. This is the time difference between the step change in magnitude and the time tag reporting the full step change. The 10% step represents the threshold set in the PDC decision tree.

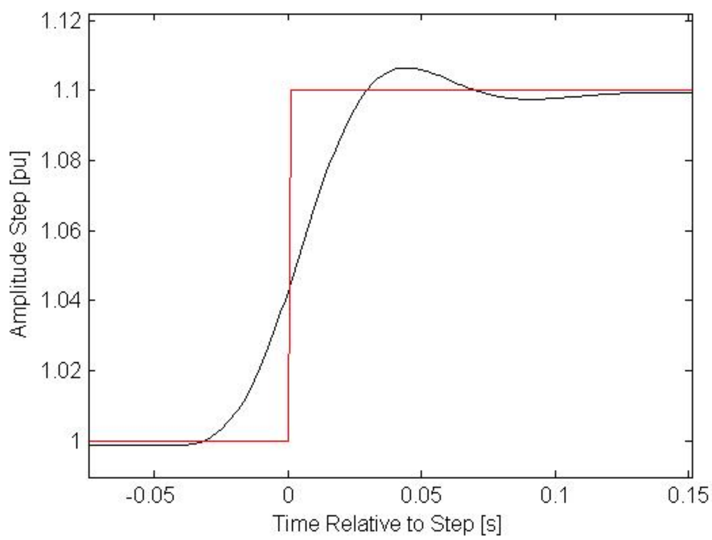


Figure 5.7: Input Step Change and PMU Response

The effect of a measurement window on the latency of a PMU measurement was also explored. Using a window of data to compute a set of recursive phasors creates a more accurate representation of the system quantities, but also requires time to wait for collecting all these quantities. For applications requiring the data in real-time, the effect of the measurement window is apparent because the data is delayed by half the measurement window for a middle of the window time stamp. Figure 5.8 shows the effect of adding the reporting latency to the response of the PMU. This example illustrates the delay associated with the PMU reporting at 60 samples per second. For each measurement, the additional shift to the right corresponds to an increase in the window length from one cycle, to two cycles, to three cycles, respectively.

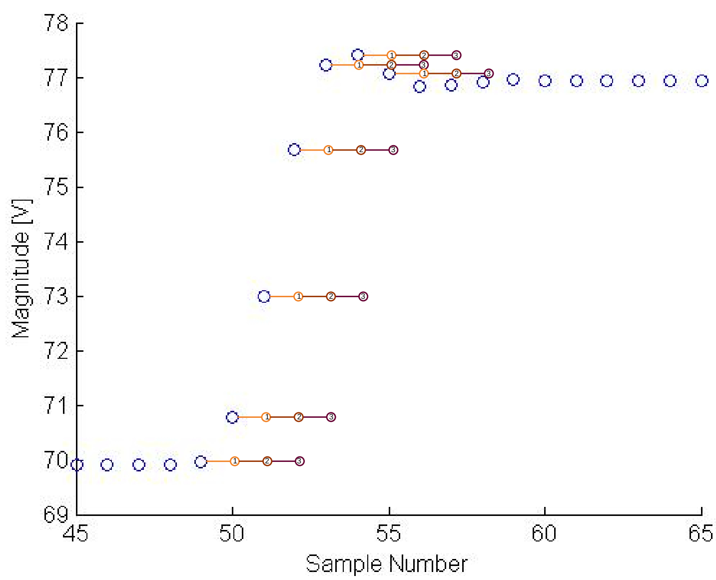


Figure 5.8: Step Response with PMU Reporting Latency

Figure 5.9 shows the combination of both the response and reporting latencies to get an overall picture of major contributing factors to PMU latency. The step response of the PMU was derived using the methodology discussed above in MATLAB. Overlaid on the response are the additions of the reporting times to make up the total delays, t_1, t_2, t_3, which are based on the window length of the PMU as well as the response time if the measurements are being used in a real-time application.

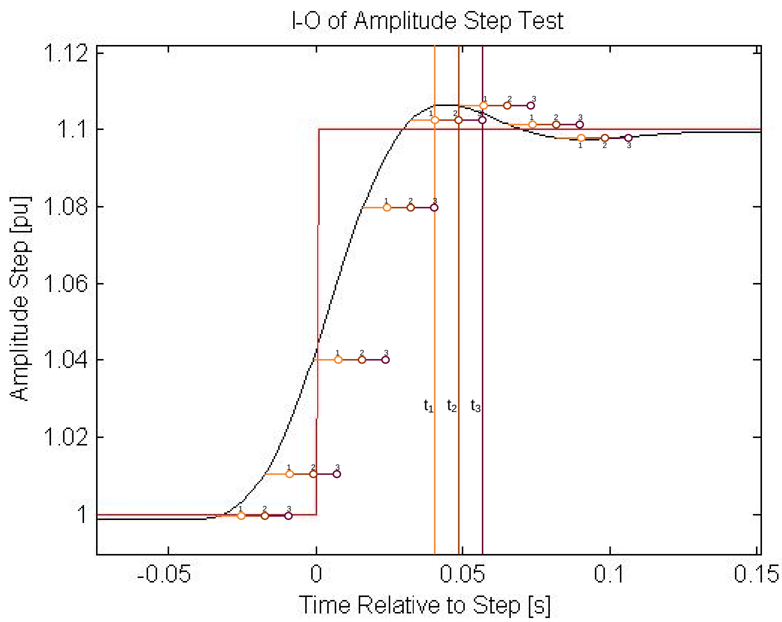


Figure 5.9: Combined PMU Response and Reporting Latencies

The time required for the PMU to respond to the change in the input signal, along with the time required to sample the data and report a phasor comprise the total PMU latency. The first PMU measurement to cross the 10% threshold value would occur at approximately 41 ms, 49 ms, and 57 ms after the step change was initiated for 1 cycle, 2 cycle, and 3 cycle window lengths, respectively.

Using a one cycle measurement window, the response time contributes more to the overall delay than the windowing effect of the PMU. As the length of the measurement window grows for applications requiring a more accurate response, the reporting delays become more apparent. This is a key aspect of determining total latency of a WAMS for real-time applications compared with offline analysis. Table 5.1 shows a comparison of the worst-case total latency of the PMU for these scenarios for a step change in the input signal.

Table 5.1: Total PMU Latencies

	1 Cycle	2 Cycle	3 Cycle
Off-line	33 ms	33 ms	33 ms
Real-Time	41 ms	49 ms	57 ms

5.4 Phasor Data Concentrator Testing

The phasor data concentrator latency is defined *as the time difference between the PMU TOS packet and the PDC TOS packet*. These packets were monitored using Wireshark running on a PC. Since the PMU reports time stamped phasors to the PDC and the PDC also has access to IRIG-B, the PMU time stamp can be used as the reference time for the test. The TOS packet was used because the 1PPS signal from the GPS provided the highest accuracy of packet timing.

5.4.1 Throughput Capability

The minimum throughput of the PDC was determined based on the processing of a single PMU rather than the time alignment of a large number of PMUs. Figure 5.10 below shows the system setup for testing PDC throughput.

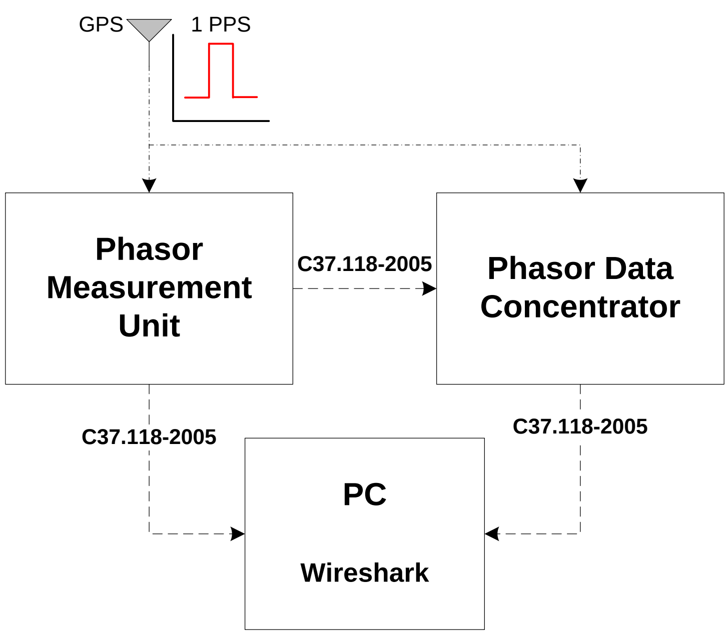


Figure 5.10: Phasor Data Concentrator Testing Network Architecture

The GPS receiver provided the 1PPS signal to both the PMU and PDC for time stamping the measurements. C37.118-2005 packets were sent from the PMU to the PDC and from the PDC to the computer. Since all devices were located on the same network, Wireshark running on the PC was able to monitor the Ethernet traffic on the network between each device.

A Wireshark capture was started, and C37.118-2005 packets were collected for a period of approximately 20 seconds. Figure 5.11 shows an example of data packets for both the PMU and the PDC, with the FRACSEC values highlighted.

```
00 30 a7 00 9c cc 00 30  a7 00 be 9e 08 00 45
00 54 34 eb 00 00 40 06  28 97 0a de 03 b2 0a de
03 b5 12 78 95 1f 4e 51  31 93 06 b0 32 aa 80
21 ba 6e 43 00 00 01 01  08 0a 60 bb 59 f8 00 06
42 bb aa 01 00 20 00 01  4e 3b 07 3c 00 44 44 44
00 00 00 00 99 d7 00 00  fd 1c 00 00 00 00 00 00
90 13

00 07 e9 ab 54 0b 00 30  a7 00 9c cc 08 00 45 00
00 3c 00 00 40 00 40 11  1d 88 0a de 03 b5 0a de
03 b9 80 05 12 6f 00 28  70 8e aa 01 00 20 03 e8
4e 3b 07 3c 00 04 11 ab  00 00 00 00 99 d7 00 00
fd 1c 00 00 00 00 00 00  33 4d
```

Figure 5.11: PMU (Top) and PDC (Bottom) Data Frames

The FRACSEC values for the PMU and PDC were matched up for each sample as shown above. The PMU TIME BASE was set to FF FF FF in hexadecimal while the PDC TIME BASE was set to 1E6. Dividing their FRACSEC value by the TIME BASE gave the millisecond value of the time stamp. The two equations below illustrate this alignment computation.

$$44\ 44\ 44 / \mathrm{FF\ FF\ FF} = 0.266667 = 26.667\ \mathrm{ms}$$
$$04\ 11\ \mathrm{ab} / 1000000 = 0.266667 = 26.667\ \mathrm{ms}$$

The latency, or time required for the PDC to output a C37.118-2005 packet following an input of the PMU C37.118-2005 packet with the same time stamp, was computed by taking the time difference between packet arrival as reported by Wireshark. Figure 5.12 shows the results for the SEL-3378 PDC. The absolute maximum latency of the PDC was less than 1.6 ms; the average latency was 0.77 ms.

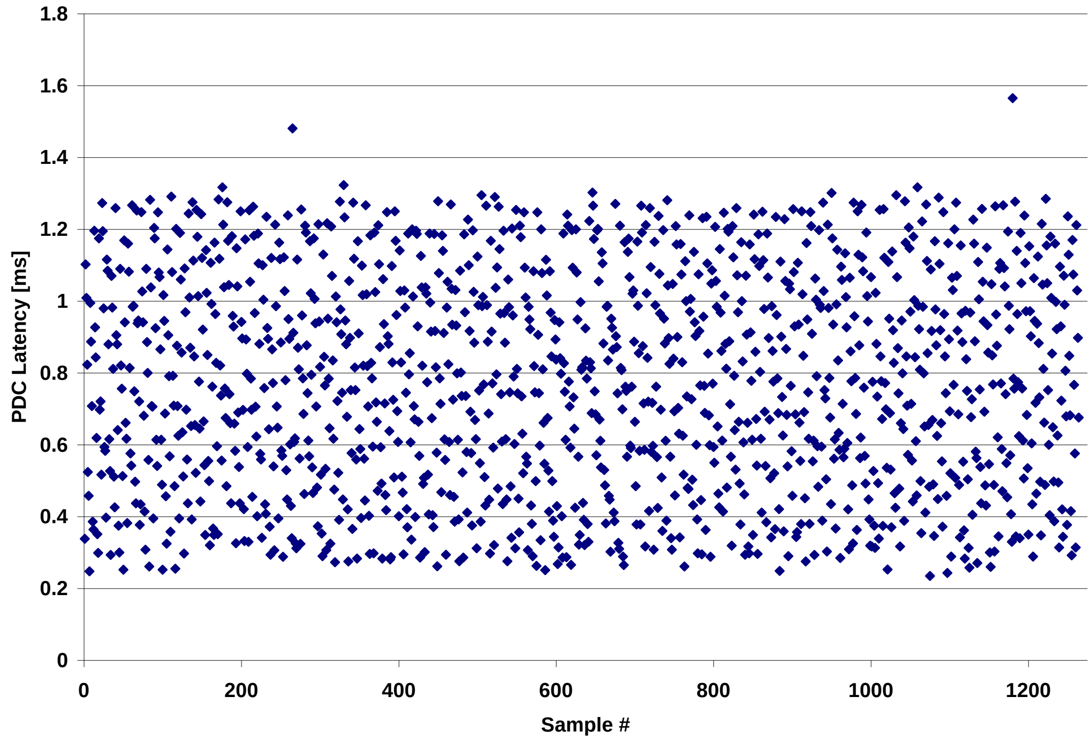


Figure 5.12: SEL-3378 Latency Results - Single PMU

In a real system, multiple PMUs send data to a PDC for time alignment, and a consolidated 37.118 packet is sent to various applications. This test, with only a single PMU sending packets to the PDC, was used to test the PDC for throughput capability. With no need to time align the data, the PDC simply waited for incoming PMU measurements and immediately processed and output the data. The latencies observed for this test provided a bare minimum for PDC latency, and the following section uses multiple PMUs to observe if the latency increased for additional PMUs on the network.

5.4.2 Multiple PMU Test

Two additional PMUs were added to the network and aggregated by the SEL-3378, to determine if a change in the number of time stamps requiring time alignment would have an effect on the latency of the PDC. Table 5.2 shows an example Wireshark TOS capture for the three-PMU network.

Table 5.2: Three-PMU Wireshark Capture

No.	Time	Source	Destination	Protocol
489	0.694736	10.222.3.171	10.222.3.181	SYNCHROPHASOR
503	0.717650	10.222.3.178	10.222.3.181	SYNCHROPHASOR
513	0.733329	10.222.3.176	10.222.3.181	SYNCHROPHASOR
516	0.735458	10.222.3.181	10.222.3.185	UDP

The C37.118-2005 TOS data packet from PMU 2 (.171) is the first to arrive at both the PC (Wireshark time not shown for simplicity) and the PDC (.181). PMU 1 (.178) is the next PMU to reports its phasors to the PDC, 22.9 ms after PMU 1. Lastly, PMU 3 (.176) reports its phasor data packet, 40.7 ms after PMU 1. The difference in latencies from the different PMUs, although not an issue for PDC latency testing, was due to the filtering settings of each PMU. Each PMU reported its data packets over Ethernet through a simple hub to the PDC on the network using TCP protocol.

In the example above, the PDC reported its C37.118-2005 packet to the PC after the final PMU reported its packet, 2.13 ms later. This is only the first example TOS in a set of captures; the overall results for testing the latency of the SEL-3378 are shown in Figure 5.13.

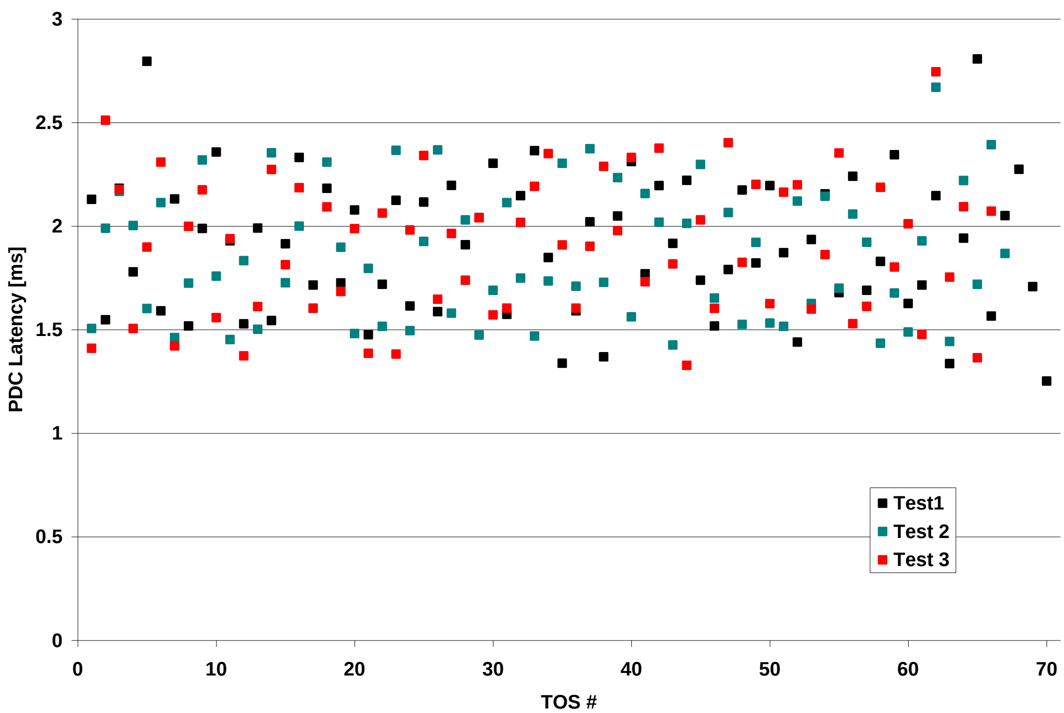


Figure 5.13: SEL-3378 Latency - Multiple PMUs

The maximum latency observed for the three separate tests performed was 2.81 ms; the average latency was 1.89 ms. These values are slightly above the PDC latency for a single PMU. It is both apparent and expected that as the additional PMUs were aggregated by the PDC, the PDC required more processing time to align the time stamps and generate an output packet.

Although it is unknown what the latency would be when the maximum number of PMUs are connected to the PDC, it is expected that it would slowly rise for each additional device. The SEL-3373 Station Phasor Data Concentrator advertises that with 40 PMUs connected, reporting at 60 messages per second, and a single C37.118-2005 output, the average latency is <15 ms; with 20 PMUs and three outputs, it is < 10 ms [41]. A similar test as performed for the SEL-3378 was also performed for the SEL-3373. Starting with a single PMU connected to the PDC, additional PMUs were added to the system to see the incremental effect on latency that each PMU would have on the PDC. Up to five PMUs were connected to the SEL-3373 and the results are shown in Figure 5.14. As the figure shows, each additional PMU had a nearly linear effect on the latency of the PDC. The addition of a PMU being time aligned by the SEL-3373 caused the PDC latency to increase by an average of 0.13 ms. These times are expected to be faster than the SEL-3378 since the SEL-3373 is simply a PDC, with no added functionality or complexity.

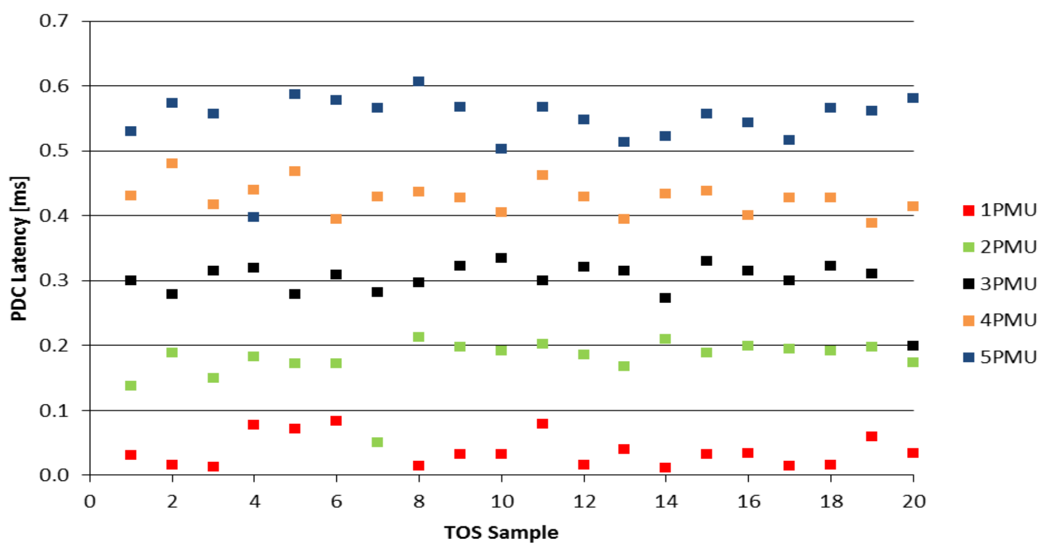


Figure 5.14: SEL-3373 Incremental Latencies

5.4.3 Disconnect Test

Two tests were performed with the SEL-3378 PDC to determine the effects of losing a PMU stream on the latency of the PDC. These tests included changing the MISSING_MESSAGE_THRESHOLD setting of the PDC from 30 to 6 to observe the difference in delay. For each test, two PMUs were connected to the PDC, sending C37.118-2005 packets for time alignment. In steady state, PMU 1 experienced approximately a 1ms delay while PMU 2 arrived later at approximately 26ms. At a given time within each test, PMU 2 was disconnected from the network. The capture of C37.118-2005 packets using Wireshark was continued for approximately one second and the results were observed. Figure 5.15 shows the results from these two tests, aligning the times when PMU 2 was disconnected from the network. The black data points represent the PDC latency using PMU 2 as a reference to the delay and the red data points represent the PDC latency using PMU 1 as a reference. The PDC latency should be classified as the black dots during the steady state conditions, but the reference PMU was disconnected mid-test, so the reference changed. Both are included for clarity.

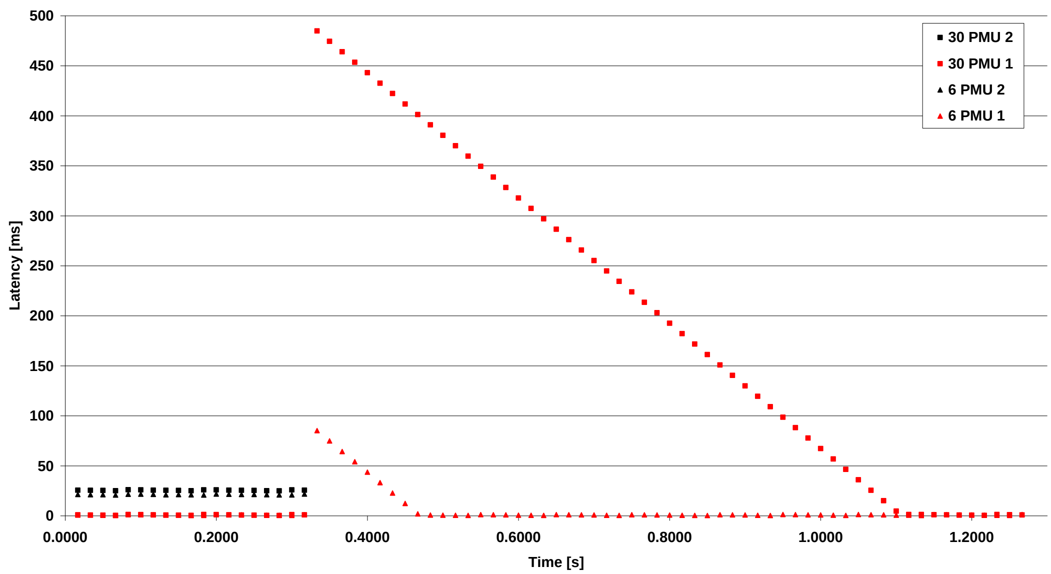


Figure 5.15: PDC Latency Response to Lost PMU Measurements

The test provided some valuable into how the PDC would respond to lost PMU measurements. It was observed that the PDC MISSING_MESSAGE_THRESHOLD was the contributing factor to latency when measurements were discontinued. This threshold was based on the reporting rate of the PMUs and the number of missing messages allowed. The PMUs reported phasors at 60 samples per second; therefore each missing message increment added approximately 16.667ms of latency plus the time required to empty the internal buffer of data in the PDC. Figure 5.15 illustrates how the 6 message threshold would require approximately 150ms to stabilize back to minimal PDC latency while the 30 message threshold would require approximately 800ms to stabilize. The length of time allowable is determined by the application using the data. This test demonstrates the importance of taking the PDC missing message threshold into account.

5.5 Adaptive Protection Scheme Latency Testing

The adaptive protection scheme consists of PMUs measuring system quantities, the PDC aggregating those measurements and deciding on a control action, and a control signal triggered during certain system conditions. The relays, based on the arming signal applied to the protection schemes, perform either a voting or normal operation. The following sections describe how the latency of the adaptive protection scheme was tested for its performance against different system events. The hardware PDC was used to test the arming signal latency while the software PDC was used to test the relay operation latency.

5.5.1 Hardware PDC Threshold Step Response

A simple adaptive protection scheme with two PMUs measuring 3-phase voltages was set up as shown in Figure 5.16; the PDC used was the SEL-3378.

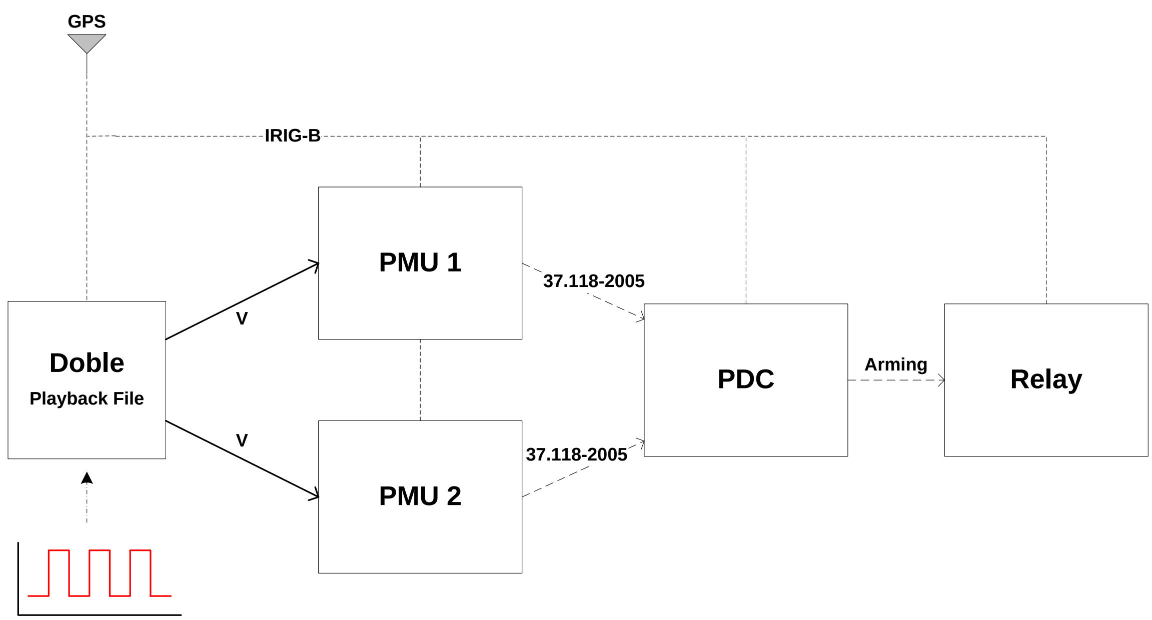


Figure 5.16: Adaptive Protection Scheme (Decision Tree) Configuration

A magnitude step change was programmed into the Doble 6150 to occur at time-synchronized intervals. The Doble had a 10kHz discrete output reporting rate; therefore, the change appeared instantaneous to the PMU reporting measurements. The playback file was set to run at the top of a minute, and the times that the step changes occurred were therefore known. Voltage magnitude steps from 70V to 77V RMS were used and this signal was applied to the input terminals of one of the PMUs. This value was assumed to be the secondary voltage from an instrument transformer. The PMU reported positive sequence phasor data at 60 frames per second, and the PDC simply collected that PMU measurement. This test was used to measure the smallest expected delay in response to a transient condition. The PDC was programmed to parse the incoming PMU stream, extract the positive sequence voltage phasor, and compare it to a threshold value in real-time. The threshold was set at 73.5V, 50% of the total step. If the voltage magnitude reported was less than the threshold, the PDC set an arming flag to STRESSED := FALSE; if the voltage magnitude reported was greater than the threshold, the PDC set the flag to STRESSED := TRUE. This basic function is shown below.

```
IF SKA1.PI[1].MAG <= 73.5
THEN
        STRESSED := FALSE;
ELSE
        STRESSED := TRUE;
END_IF;
```

The resulting flag was used to enable the Fast Operate commands FAST_OP_REMOTE_BIT_SET and FAST_OP_REMOTE_BIT_CLEAR, as discussed in Chapter 4.

The relay was used as a measure of the time required to acknowledge the control signal. The RB01 remote bit sent from the PDC as a control command was used to trigger a Sequential Event Report (SER). When the remote bit signal was received by the relay, RB01 was asserted and the SER event recorded. Likewise, when the RB01 was deasserted, the SER also recorded this event. Therefore, each step change recorded an asserted SER event.

Analysis of the time difference between the input step change and the output control action is dependent solely on a universal time for all the system devices. Fortunately, each of the devices used was synchronized to UTC using a common GPS receiver. The GPS signal was sent to each of the devices through an SEL-2407 Satellite-Synchronized Clock reporting IRIG-B. Therefore, the time reported by the relay SER originated from the same time source used for setting the Doble F6150 to perform the step changes.

An important detail to note is that two time references were used for testing 1) the overall protection system latency and 2) individual component latencies. Each device except the computer received IRIG-B from the same time source. Therefore, the time reported by the computer was not accurate to UTC. Conversely, the time difference between the reported packets was accurate because Wireshark reports accurate timing during inter-capture periods. For example, the SER from the relay receiving the control signal could be compared to UTC since it was reported by a synchronized device. The Wireshark capture, on the other hand, could be used to determine the latency of the PDC to send a control signal and output a C37.118-2005 output packet. These two times could not be aligned together since there was a discontinuity due to the timing problem. This issue could be resolved if Wireshark had no delay in requesting local time on a highly accurate, time synchronized PC.

Three separate tests were run on the simple threshold configuration, each test consisting of 8 steps from 70V to 77V RMS. These step changes occurred at the top-of-second (TOS), each 1/16 further into a 60 Hz cycle. This had little, if no, effect on the overall latency since the next packet after the TOS would send the phasor crossing the threshold value. A plot of the overall latency is shown in Figure 5.17. The SER Event Reports from the Relay are provided in Appendix D.

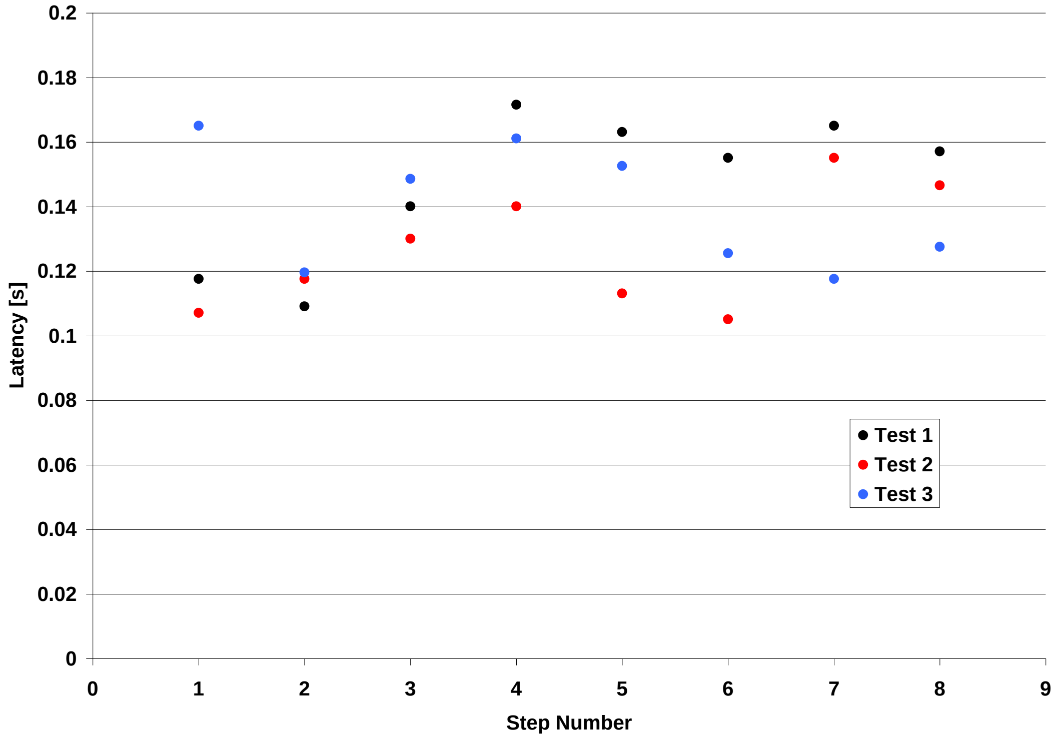


Figure 5.17: Basic Threshold Step Response Latency

Figure 5.17 shows that the maximum delay observed from the time between the generated step change and the relay operation was 171.6 ms, while the minimum delay was 105.1 ms.

The overall delay was dissected to determine where the latencies originated and the cause of the variance in delays from step to step. The SER Events were tested to observe how quickly the SER could capture an event. The Doble F6150 also provides digital outputs, and Digital Out 1 was sent to the IN101 screw terminal of the relay. This digital bit was set high at the first step change in the test. Additionally, a SER Event was triggered based on the 591P1 Overvoltage pickup within the relays. The step test was run again and the results were that the IN101 SER Event reported 2.1ms after the step change while the 591P1 SER Event reported 16.8ms after the step change. This was repeated a number of time to ensure precision, and it was concluded that the SER Event reporting accounted for a minute component of the delays associated with the control action.

Wireshark was used to compare the PDC C37.118-2005 output as well as the Telnet control signal to the PMU reported packets. Tables 5.3 and 5.4 show the results for the PDC output packet and control packet latencies. These were computed by subtracting their respective times from the time that the final PMU reported its corresponding packet.

Table 5.3: C37.118-2005 Packet Delay

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Test 1	0.665	1.255	0.518	0.721	0.889	1.287	0.539	0.608
Test 2	1.058	0.544	1.174	1.333	0.357	1.168	0.961	1.303
Test 3	0.9	0.998	0.455	0.721	0.936	1.194	0.821	1.122

Table 5.4: Control Signal Packet Delay

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Test 1	22.411	22.921	23.361	24.658	24.795	25.992	26.333	27.397
Test 2	30.064	30.625	31.023	32.259	32.378	33.167	34.181	33.519
Test 3	31.831	31.931	33.501	33.868	35.157	35.181	36.098	20.68

Although it was not measurable with the equipment available, it is expected that the PMU delays would remain fairly constant throughout the testing process. Further exploration and discussion with the manufacturer of the PDC provided insight about the variance in the delays observed. The Fast Operation commands sent from the PDC to the relays use Telnet communications, which has a low priority transmission, thus the control signal could be handled at slightly different times for each test [29]. This holds for both the PDC control action on the sending end and the relay responding to the action at the receiving end. The PDC control action also includes the parsing of the PMU data, the threshold decision, and the communication of the control. This is a much more complex algorithm than the simple C37.118-2005 time alignment output, thus requiring more time. Since the PMUs and PDC were reporting phasors at 60 samples per second, or 16.667 ms, this could explain the slightly drift in control signal output. The tasks within the PDC to perform the decision tree logic and set/clear the remote bits were run on a 16 ms time interval. The slight difference between these two values would compound until they realigned.

This methodology for exploring the latency of each component is not exact, due to the lack of synchronization across the entire setup of devices including Wireshark. Although, this is the case, the overall latencies reported are accurate to UTC, and the exploration shown above does provide insight as to what is actually occurring during the control action.

A similar test was performed on the test system, using a ramping signal rather than a step change; this is discussed in the following section.

5.5.2 Hardware PDC Heavy Winter Step and Ramp Response

The Heavy Winter model for the security/dependability adaptive voting scheme used in Chapter 4 is shown in Figure 5.18, with slight modification. This is a laboratory implementation of the decision tree. The red decisions in the tree show the path tested for latency. These decisions represent the longest traversal of the decision tree, which would be the longest required time for computation. This does not take into account any delay in the measurements themselves.

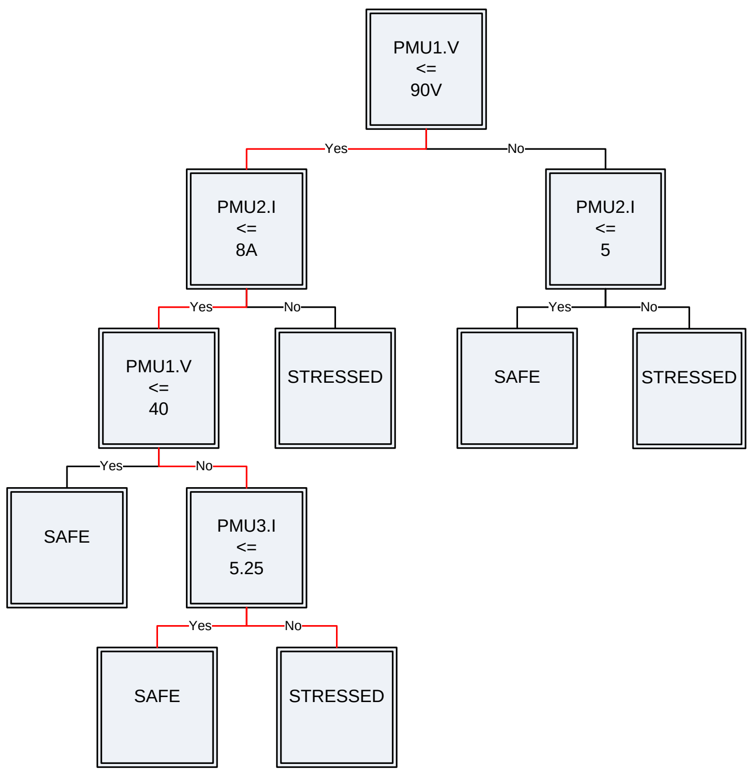


Figure 5.18: Heavy Winter Step Test Tree for Latency Testing

The real and imaginary components used in the actual tree were converted to magnitude for simplicity during testing. Both voltage and currents were used due to the limited signals created by the Doble test set. As the decision tree shows, the voltage was used to make the first determination at the origin node in the tree. The voltage level was set to some value less than 90V for the entirety of the test. A 3-phase current was then applied to both PMU 2 and PMU 3 in series since only one current source was available.

Two tests were performed on the Heavy Winter Model decision tree to determine the latency of its operation: a step response and a ramp response test. Both tests moved the classification of the operating conditions of the system from SAFE to STRESSED. The time between when this occurred on the input and then time when it was acknowledged by the relay on the output is considered the latency of the adaptive protection scheme. Figure 5.19 shows the laboratory setup for this test; this is the same physical network setup that was used for the hardware PDC testing.

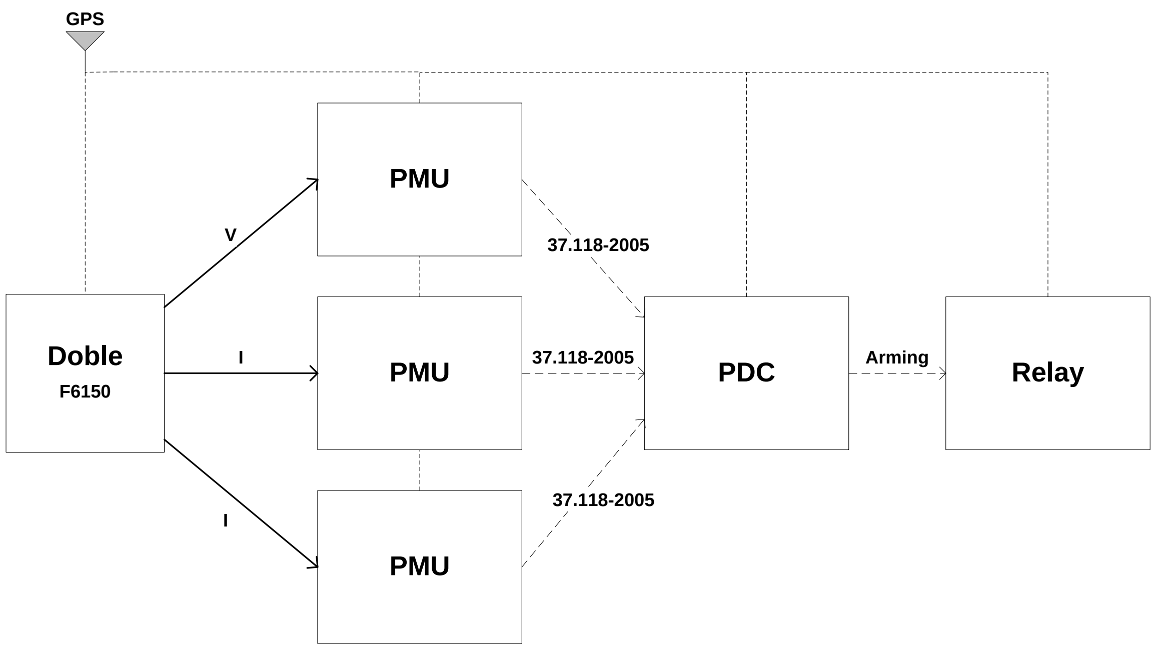


Figure 5.19: Adaptive Protection Scheme Latency Test Configuration

The step test generated a step change on the input current signal at the TOS multiple times during each test, stepping the current from 5A to 5.5A. The results of these tests are shown in Figure 5.20.

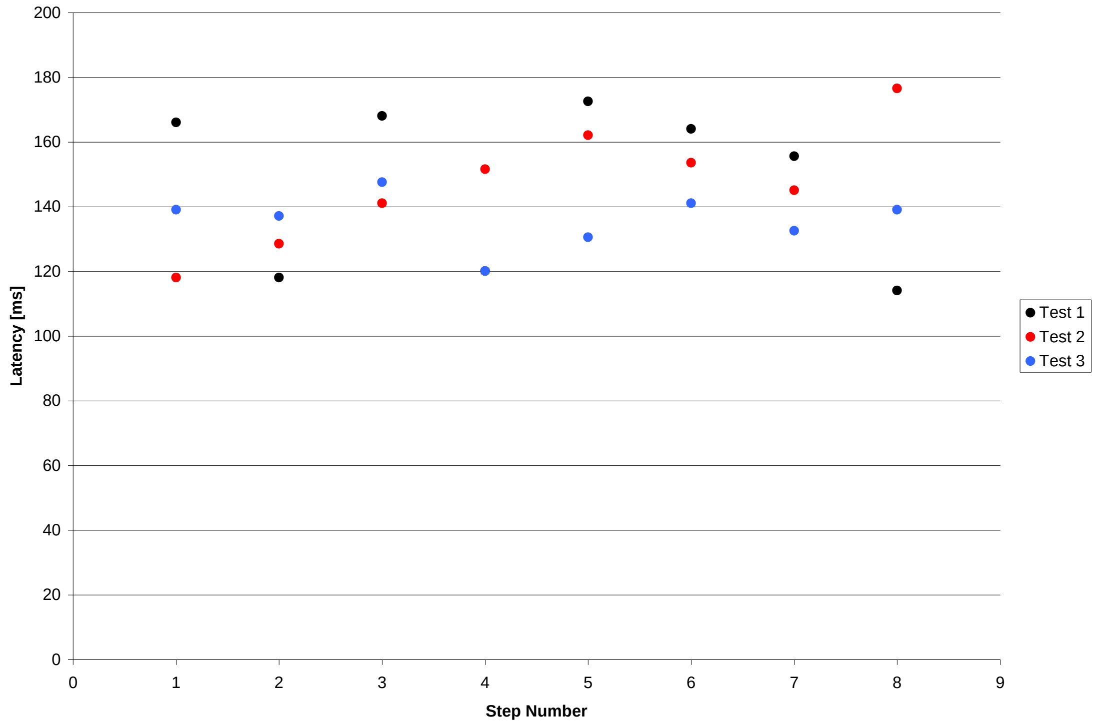


Figure 5.20: Heavy Winter Step Response Latency

The results for the Heavy Winter Model Step Test are very similar to the Simple Threshold Step Test results. The maximum overall latency was 176.6 ms, while the minimum latency was 114.1 ms. The maximum and average latencies were only about 5 ms greater, while the minimum latency was about 10 ms greater. This small increase in latency is most likely attributed to the additional extraction and comparison of PMU measurements in the decision tree. The SER Events for this test are provided in Appendix D. Tables 5.5 and 5.6 show a breakdown of the latencies, explored in the same way as the Simple Threshold test.

Table 5.5: C37.118-2005 PDC Packet Delay

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Test 1	1.99	1.8	1.961	1.406	1.865	1.637	1.569	2.224
Test 2	1.442	2.345	1.71	2.342	2.229	1.852	1.953	2.308
Test 3	2.407	2.238	1.849	1.282	1.435	2.174	2.045	2.112

Table 5.6: PDC Control Action Delay

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Test 1	21.774	23.488	24.74	25.254	24.627	25.467	26.336	28.016
Test 2	20.159	21.154	21.857	21.044	22.02	22.523	23.731	25.146
Test 3	20.064	20.953	22.471	23.025	22.257	23.865	24.822	25.894

The latencies for the C37.118-2005 output packet from the PDC are greater than those for the Simple Threshold Test. This is due to the greater number of PMUs being aggregated by the PDC in this test. The decrease in Telnet Fast Operate command latencies for the Heavy Winter test was unexpected. These were generally a few milliseconds less than the Simple Threshold test.

A ramp test was also performed to generate a change in the current at a slower rate. This would better reflect a sequence of steady-state conditions rather than a transient event. The 3-phase current was ramped from 0A to 5A RMS, with the current passing the decision tree threshold value at 0.5% change in nominal current (25 mA) every 4 seconds. The Heavy Winter Model tree used for ramp testing is shown in Figure 5.21.

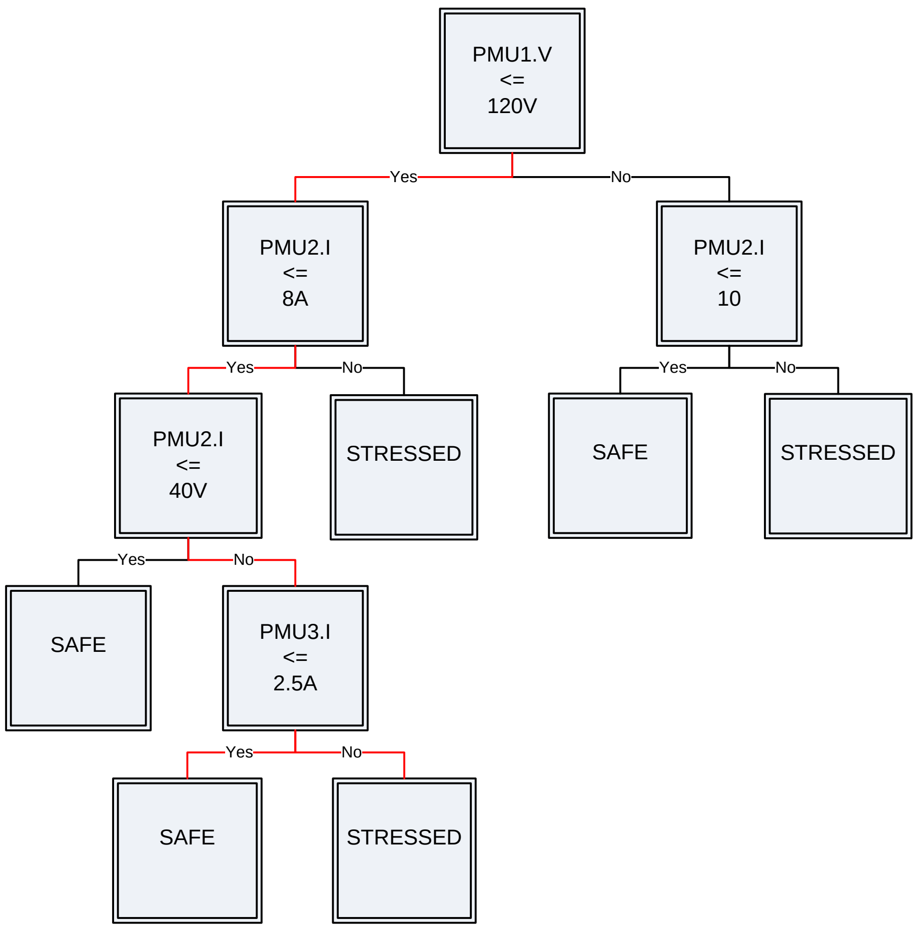


Figure 5.21: Heavy Winter Ramp Test Tree for Latency Testing

The SER Event Report from the relay recording state changes is shown below, along with a graph of the overall latency in Figure 5.22.

#	DATE	TIME	ELEMENT	STATE
18	08/05/2011	20:39:52.1331	RB01	Asserted
17	08/05/2011	20:40:40.4146	RB01	Deasserted
16	08/05/2011	20:48:52.1351	RB01	Asserted
15	08/05/2011	20:49:19.0061	RB01	Deasserted
14	08/05/2011	20:52:52.1601	RB01	Asserted
13	08/05/2011	20:53:12.9666	RB01	Deasserted
12	08/05/2011	20:55:52.1416	RB01	Asserted
11	08/05/2011	20:56:00.9121	RB01	Deasserted
10	08/05/2011	20:58:52.1626	RB01	Asserted
9	08/05/2011	20:59:03.2271	RB01	Deasserted
8	08/05/2011	21:10:52.1601	RB01	Asserted
7	08/05/2011	21:10:56.0686	RB01	Deasserted
6	08/05/2011	21:13:52.1456	RB01	Asserted
5	08/05/2011	21:13:54.9831	RB01	Deasserted
4	08/05/2011	21:16:52.1271	RB01	Asserted
3	08/05/2011	21:17:01.2851	RB01	Deasserted
2	08/05/2011	21:19:52.1831	RB01	Asserted
1	08/05/2011	21:19:57.9581	RB01	Deasserted

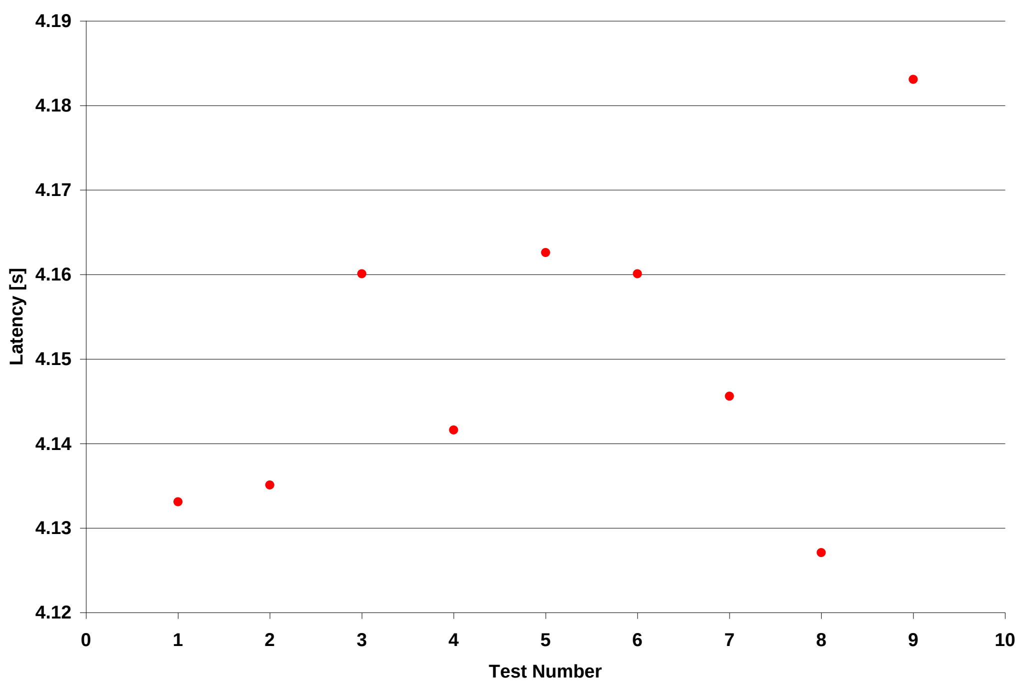


Figure 5.22: Heavy Winter Ramp Test Overall Latency

The latency results for the Heavy Winter Ramp Test were expected to be similar to the Heavy Winter Step Test, but this was not the case. As Figure 5.21 shows, the overall latency of the protection scheme was far greater for a slow ramping event than for an instantaneous step change in the current signal. This is due to the testing procedure as well as the characteristics of the PMU measurements.

The excessive delays in this test are due to the attenuation of the analog signals from the Doble by the PMUs converting those signals to discrete digital samples. The reported positive sequence phasors were monitored visually during the ramp to determine what the PMUs were reporting as compared with the Doble input signal at the transition point. At second 108 into the test, when the Doble output was 2.5A, the PMUs measuring the current reported 2.474A and 2.4888A, with the latter PMU being used for classification. At the next minute step change in the input signal at second 112 into the test, the Doble output was 2.525A. The PMUs then reported 2.503A and 2.508A, with the latter being used for classification. Therefore, the additional four seconds in the results above are attributed to the PMU attenuating the signal and not surpassing the threshold until the next small step change. Removing the four second delay, the results are again shown in Figure 5.23.

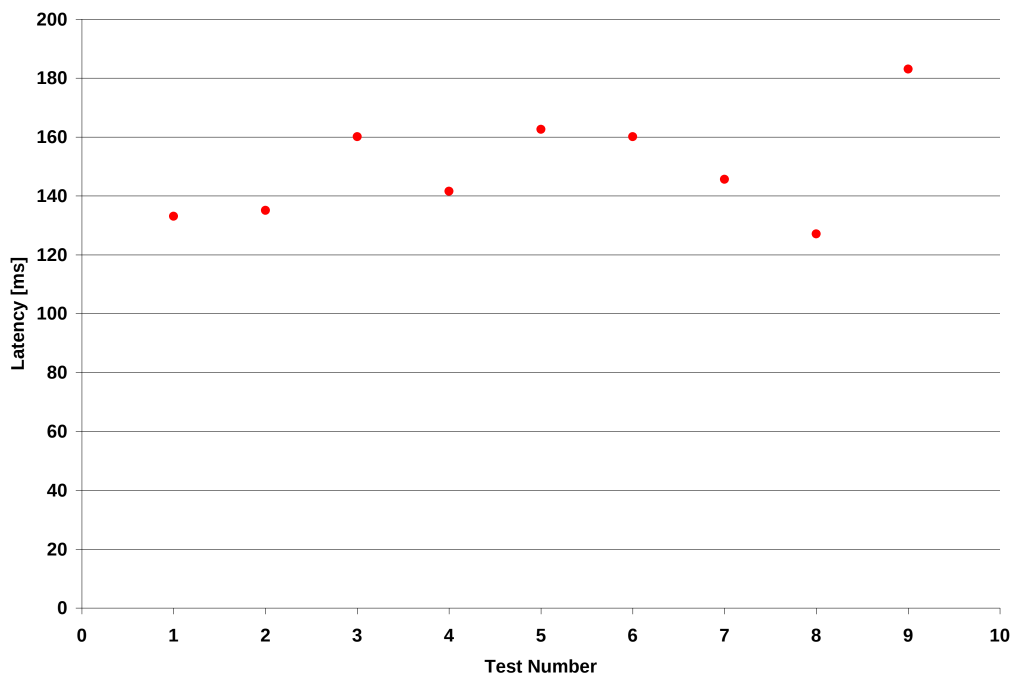


Figure 5.23: Heavy Winter Ramp Test Overall Latency - Time Shift Removed

The latencies are similar to the Heavy Winter Step Test, with the maximum latency at 183.1 ms and the average latency at 149.8 ms.

5.5.3 Hardware PDC Breaker Operation Latency

The hardware-based solution presented in Chapter 4 was tested for relay operating latency. This was done using a simplified WAMS setup using a single PMU to Arm or Disarm the relay voting scheme. The work in [17] describes in detail how the PMUs vote or operate normally in a Master-Slave configuration based on the Arming signals applied. Figure 5.24 shows the configuration of the Master-Slave voting scheme used in the hardware PDC solution.

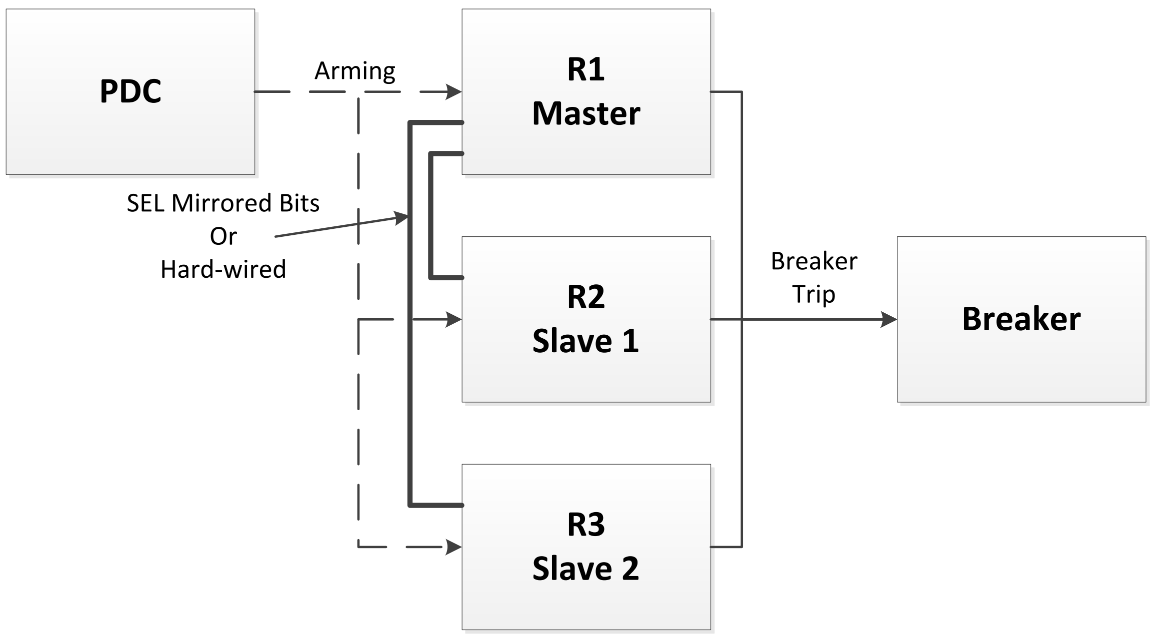


Figure 5.24: Master-Slave Relay Voting Configuration

The relays were configured such that there was one Master relay, and the others Slave relays. When the voting scheme was armed, the Master relay performed the voting and was the only device capable of tripping the breaker. The trip signals for the other two relays were restrained but the protection elements were communicated to the master using either SEL Mirrored Bits protocol or hard-wired contacts at the relay input terminals. During disarmed operation, the relays performed normal operation where any one relay could trip the breaker. For the purposes of testing, the trip signal was assigned to an instantaneous overcurrent element with different pickups for each relay.

Latency is apparent in this configuration during Stressed conditions because the Slave relays must send their trip signal to the Master device prior to breaker tripping. This configuration was tested for its added latency to the normal tripping of the breaker using a time synchronized step input to the terminals of one of the relays. Figure 5.25 shows how latency testing was set up for this protection scheme. The darker lines were the communications between Slave relays and the Master relay – SEL Mirrored Bits or hard-wired I/O contacts – and both were tested.

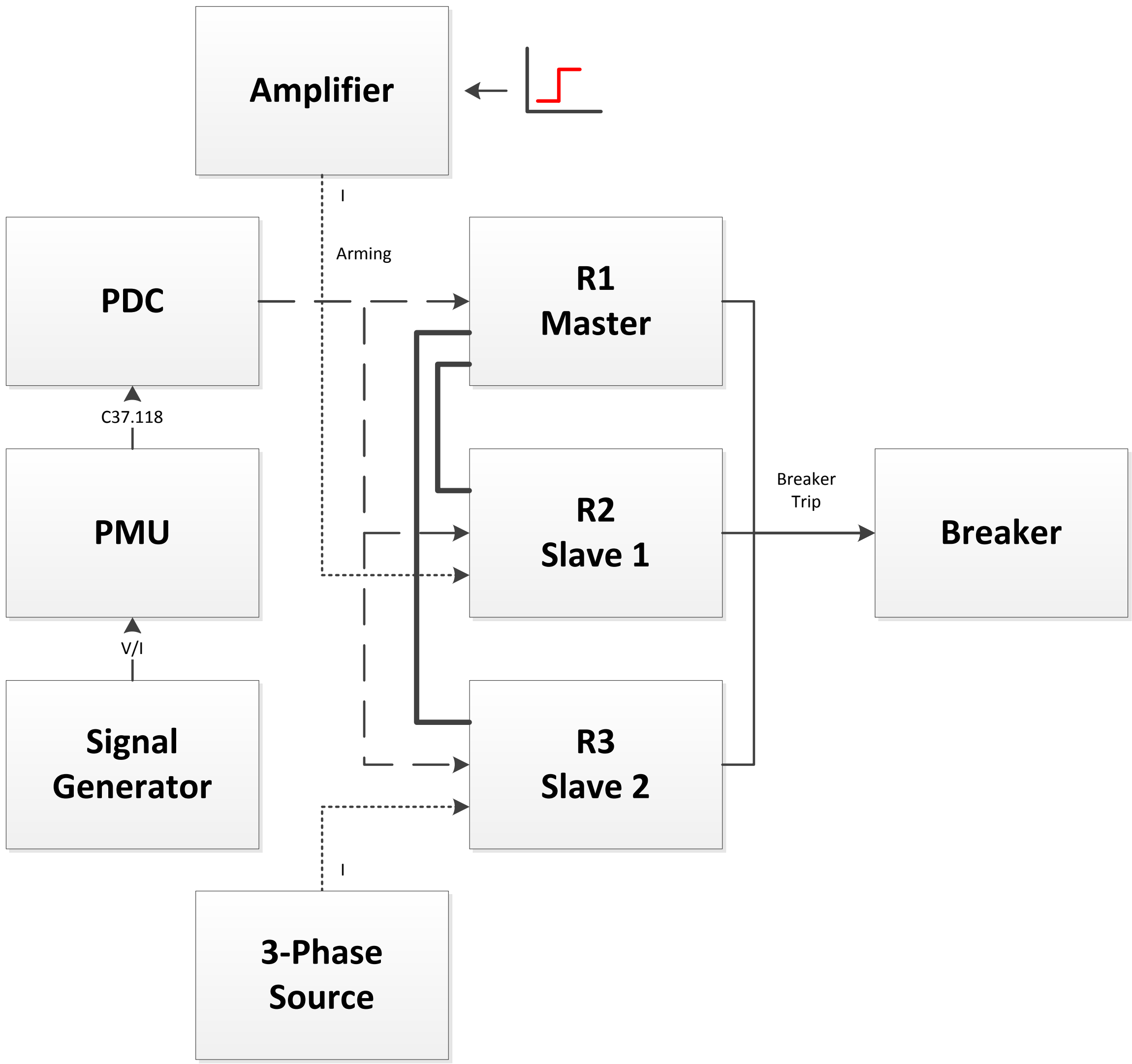


Figure 5.25: Master-Slave Relay Latency Test Configuration

First, the signal generator was used to apply a specified current amplitude to the PMU. This amplitude was set above the threshold in the arming decision tree in the PDC, causing the PDC to apply an Arming signal set to Stressed to all three relays. This current was held throughout the test to maintain voting operation. Next, the 3-phase source was increased to apply a current above the overcurrent pickup setting of relay R3. Since the voting scheme was in a Stressed condition, the overcurrent element was sent to the Master, but tripping of the breaker was restrained. Lastly, a time synchronized step change in current amplitude was applied to relay R2. The step was a 10% increase of a 5A nominal current, so the overcurrent setting of R2 was set to 5.25A, the middle of the step change.

Sequential Event Reports (SER) were used to capture the time at which each element asserted and when the breaker trip signal was actually applied. Since all the relays were synchronized to UTC using a GPS clock, all had the same time reference and could be compared against UTC time. Figure 5.26 shows the results of testing the relay operation latencies using SEL Mirrored Bits communication between the Slave and Master relays. Figure 5.27 shows the results for the same test using hard-wired I/O contacts to communicate between the relays. All the times were referenced to the top of second (TOS) when the step change was applied to the current input of R2. SER events for both tests are provided in Appendix D.

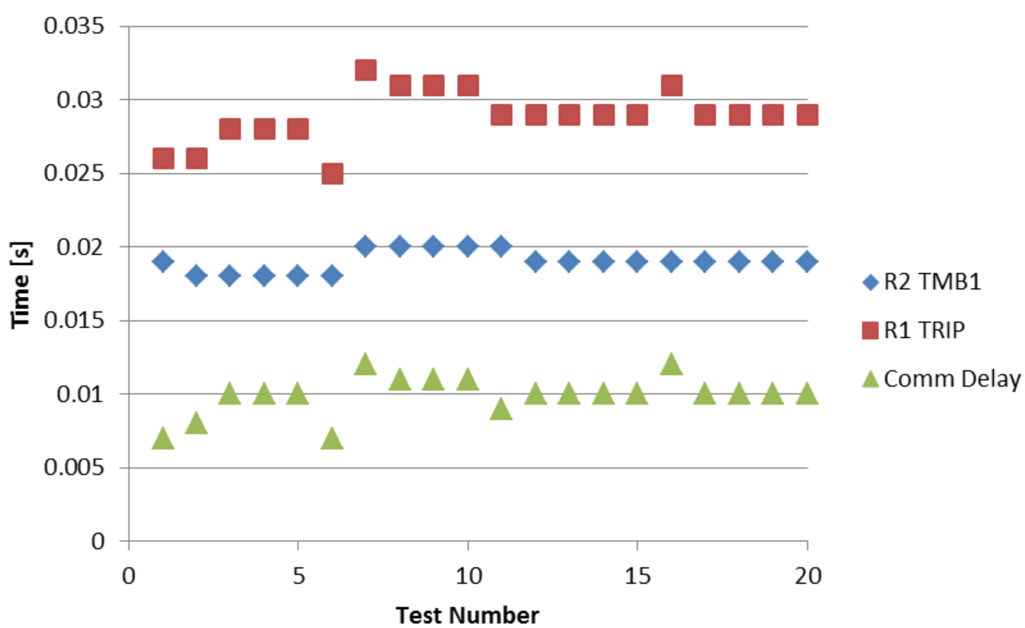


Figure 5.26: Master-Slave Relay Latency – SEL Mirrored Bits

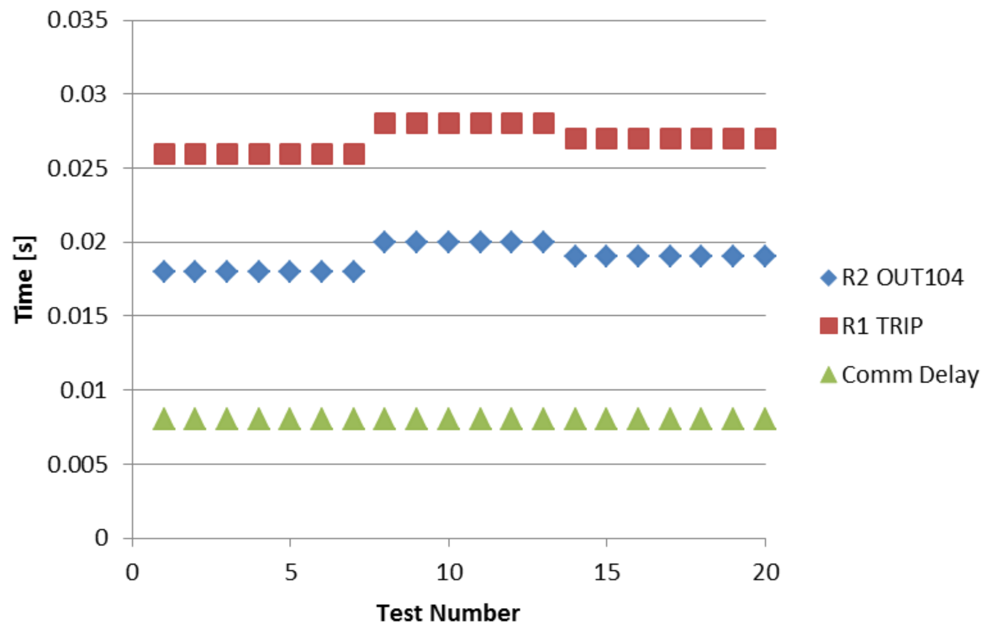


Figure 5.27: Master-Slave Relay Latency – Hard-wired

Important to note from the SER events provided in Appendix D is that for every test performed, the Slave relay 50P1 instantaneous overcurrent element asserted at the same time that the output signal was applied. Therefore, there was no delay between the pickup and the output. Also, acknowledgement at the Master relay of the input from R2 occurred at the same time as the Master relay applied its Trip signal to the breaker. There was no delay between these times.

Both figures show the critical components of the latency test. R2 output in the form of a Transmitted Mirrored Bits (TMB) signal and a hard-wired output contact are shown. The maximum delay was 20ms for both forms of output. This time corresponded to the time required for the relay to pick up the instantaneous overcurrent and apply an output signal. The time when the Master relay tripped is also provided as R1 TRIP. The time that Master relay receives the input from the Slave relay is not shown because it was the same as the TRIP signal for all tests. It can be concluded that there was no noticeable additional latency for the Master relay performing the voting function in its TRIP signal.

The communications delays were the largest contributing factor to added delays since the Slave relay had to output its trip signal not to the breaker but to the Master relay. The communications delay was determined as the time difference between when the Slave relay SER Event recorded the output enables and when the Master relay SER Event recorded the input enabled. Since both devices had the same UTC time reference, the SER events could be compared against each other. For the hard-wired configuration, the communications delay was a constant 8ms. The communications delay for the SEL Mirrored Bits configuration varied between 7ms and 12ms. Based on the results for the two tests performed, it was concluded that the hard-wired setup would add an additional 8ms of delay onto the normal configuration of relays and the Ethernet-based setup would add up to 12ms.

5.5.4 Software PDC – PLC Operation Latencies

For the software-based adaptive voting scheme, the relays perform normally for both armed and disarmed states. This is because the PLC performs the breaker trip signal based on the inputs from the relays. Of interest for latency testing was the added time required to perform breaker tripping with the PLC injected into the protection schemes. The PLC was tested for its efficiency in processing the inputs and performing the voting logic.

A Sequence of Events (SOE) module was installed on the PLC backplane to monitor input status changes and time stamp those inputs [42]. The SOE module was configured such that it would latch the time when the input status changed from "Off→On", meaning a voltage was applied to the module. The three relay trip signals were also wired to the SOE module and the PLC 24V output module was hard-wired back to the SOE module to time stamp the breaker trip signal. Latency of the PLC was defined as "*the time required for the PLC output to enable following the correct set of inputs enabling.*" The timing reference between the output and the inputs can change for each test based on which relays trip first. For example, in a disarmed state the output should enable as soon as any single input enables. Conversely, for an armed state the output should enable only when any two relay signals enable.

One hundred tests were performed, with the first half in the armed state and the second half in the disarmed state. For each test, the arming signal was applied beforehand and a fault was induced on the relay schemes. For all tests, the instantaneous overcurrent thresholds were surpassed in two of the three relays. Therefore, for both test the PLC should operate as quickly as possible. Figure 5.28 shows the results for all tests performed.

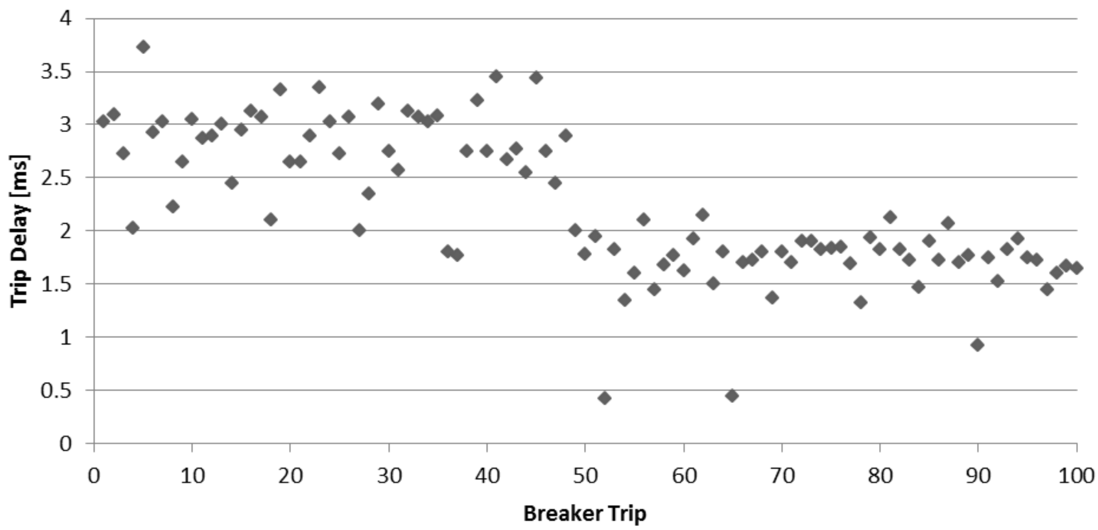


Figure 5.28: Breaker Trip Delays

As the figure shows, there is a difference between breaker trip delays in the armed and disarmed states. The data suggests that the average latency of the disarmed scheme is 2.15ms, while the average latency of the armed scheme is 3.73ms. Table 5.7 shows the basic statistical information relating to the two sets of data.

Table 5.7: Breaker Trip Latency Statistics

	Minimum [ms]	Average [ms]	Maximum [ms]
Armed	1.77	2.81	3.73
Disarmed	0.42	2.15	1.69

The delays induced by the addition of the PLC between the relays and the breaker increase from the disarmed to the armed state. Although the maximum delay observed was upwards of 4ms, this delay is still a relatively small amount of time for power system protection applications. If the protection applications can withstand an addition of a quarter cycle increase in the latency, then this option is feasible.

Chapter 6
Conclusions and Future Work

The work described in the previous chapters of this thesis detail two main topics: implementation of a Security/Dependability Adaptive Voting Scheme using decision trees, and latency testing of wide-area measurement system components. The conclusions drawn from the data collected and the work performed will be explained in this chapter, along with the limitations of the testing procedures and discussion of future work in these areas.

6.1 Conclusions

The Security/Dependability Adaptive Protection Scheme introduced by [12] provided a means of improving the reliability of the power system using wide-area measurement. The scheme determined the necessary locations in the power system requiring real-time monitoring using PMUs and the critical location in the system requiring additional protection to avoid catastrophic failures. Decision trees were used to categorize real-time classifications of the power system operating conditions into two states: Safe or Stressed. During Safe conditions, the protection system should operate towards Dependability while during Stressed conditions the protection system should operate towards Security. Based on the classification of the decision tree, an Arming signal was applied to the relays at the critical location to change their operation from a voting scheme to ‘or’ functionality.

This theory was successfully implemented in a laboratory setting using different configurations and equipment. Due to the ability to program user-defined functions along with performing time alignment and aggregation of phasor measurements, the SEL-3378 Synchrophasor Vector Processor and openPDC phasor data concentrators were used for implementing the decision trees. CART® output trees were converted to if-else logic statements to perform the classification of the operating state of the system based on key PMU measurements. An Ethernet network was set up in the lab, with the PMUs reporting their phasors at 60 data frames per second. The PDC executed both the time alignment and aggregation functionality along with the decision tree application. Based on the measurements from the incoming PMUs, the decision tree residing within the PDC would apply a control signal to the relaying scheme or PLC performing the breaker trip operation.

It was determined that both PDCs, hardware PDC (SEL-3378) and software PDC (openPDC), successfully ran the decision tree following time alignment and aggregation of incoming PMU data. Although each implementation was realizable, there were advantages and disadvantages for the different configurations. The SEL-3378 implementation of the adaptive voting scheme used a hardware-based PDC. This device could easily read in PMU measurements, apply the decision tree, and output an Arming signal to the relays all within a standalone device. The control signal sent to the relays was an SEL-proprietary Fast Operate Message protocol, and therefore required either all SEL devices or an extra device to convert these messages to DC voltages. openPDC was an effective device for performing PDC functionality and also provided a simple platform for programming C# logic into the software. This software PDC is a free open-source product that can run on a standard Windows-based PC. The drawback of this setup was the difficulty and potential cost increase to get the required output signal to the PLC. The PLC was required to read an SQL database within the computer running openPDC to retrieve the ARM signal. A substation-hardened hardware PDC has greater capabilities for interoperability between devices and vendors.

Latency of wide area measurement systems, and its effects on the adaptive protection scheme was the other major focus for this thesis. For synchrophasor measurement systems, data latency occurs at all locations within the architecture. PMUs require time to sample and convert the analog system quantities, process the data into phasor form, and output the data over C37.118-2005 protocol. The data frames must then traverse some sort of wide-area network where a PDC aggregates the data. This also requires some time since the PDC latency is dependent on the slowest arriving PMU measurement. The PMU and PDC latencies are intertwined in that the throughput power of the PDC cannot be utilized when PMU measurements with the same time stamp arrive at different times. It is important that the PDC be able to accurately process all incoming PMU measurements, and generate an output C37.118-2005 packet as quickly as possible. The PDC output packet of time aligned data is further sent to some sort of application. This could be at another remote location, on a local area network, or even within the PDC itself; this could add to additional delays. The application using the synchrophasor data also requires some time to process and perform its computations. When a control application is being deployed, latencies for sending the resulting signal back out to the power system devices requires even more time. These are all contributing factors to wide-area measurement system latency.

The latency testing was a quantification of delays from wide area measurement system devices. Phasor measurement units were tested on steady state and transient conditions and phasor data concentrators were tested based on their ability to expeditiously time align incoming synchrophasor data. The adaptive protection scheme was tested both on the latency of the arming signal to be deployed to the devices as well as the physical devices to trip the breaker for the software PDC configuration. Two major factors were considered for PMU delays: response time and reporting time. It was determined that the reporting time is most significantly affected by the window length of the PMU. If the middle of the window is used for time stamping the phasor measurement, the inherent delay due to reporting the phasor in real-time is half that measurement window. The response of the PMU due to sudden changes in the inputs was the other contributing factor to latency. A 10% step increase was applied to the PMU and the output was observed. This was done offline, demonstrative of the issues due to PMU response. When the step change was applied, it was observed that the PMU takes greater than two cycles to respond. This can be reduced or lengthened based on the filtering used, which is dependent on the types of applications that the data is being used for. The PDC was tested for its throughput capabilities. The time required to align the data based on time stamps was relatively small compared with the PMU delays. Using a hardware PDC, the SEL-3373, it was concluded that an incremental latency occurs for each PMU aggregated by the PDC. When measurements were removed or bad data was identified, the PDC had a period of time where the measurements became delayed, with a maximum of almost half a second with a 30 message waiting period. This amount of time could also be reduced or increased based on whether timeliness of the data or entirety of the data was more important.

The adaptive voting scheme was tested both for its delays in the arming signal transmission and relay operation. Using the hardware PDC, the protection scheme was tested for its ability to determine the operating state of the system using the decision tree application in real-time. Latency of the PDC to traverse the decision tree and output a control signal was tested using a laboratory setup. It was determined that the decision tree arming signal applied to the protective devices would require less than 180ms. For the purposes of the voting scheme, the delays were not important since it was assumed that the state of the system would be set prior to an event taking place due to the unlikeliness of the a significant event occurring simultaneously with a state change.

The adaptive voting scheme was also tested for the time required for the breaker trip signal to be deployed with the addition of a voting device such as the PLC. A Sequence of Events module was installed in the PLC to record I/O triggering. Time stamping was performed on the rising edge of the relay trip signals, and timing was based on relative times between inputs and outputs. It was determined that the state of the arming scheme had an impact on the relay operation delays due to the additional logic required. The maximum latency of the PLC to send the breaker trip signal following appropriate relay operation was 2.15 ms for the disarmed state and 3.73 ms for the armed state.

The area where the adaptive protection scheme and latency in WAMCS intertwine is the idea of using wide-area measurements, particularly synchrophasor technology, for protection and/or control of the power system. The latency results acquired provide a collective interpretation of where the latencies in WAMS technology arise. The PMU reporting its phasors to the PDC or control center is no faster than half the length of the measurement window used for collecting phasors. This becomes problematic when 3-cycle or even 5-cycle windows are used, because this results in a minimum of 25 to 42 ms delay. If a PDC is used to collect these measurements it generally requires a minimal amount of time, less than 2 ms in normal operation. The latency of a wide-area network (WAN) is highly variable based on the medium of communication, the distance between communication nodes in the system, and the protocol used for transmitting the data.

With the PDC located at a critical location in the power system where the voting scheme would be deployed, the major latency factors would be the PMU delays, PDC time alignment delays, communications network delays, and PLC voting delays when applicable. Since all the WAMS delays are considered to occur prior to an event occurring, the only delays affecting the protection scheme would be the PLC delays which are relatively small.

A critical consideration for these delays is, "How would these delays affect a protection scheme where the WAMS data was needed continuously? i.e., the state determination was not assumed to occur prior to an event." A transient condition on the power system could send the system into a state where quick classification of the system is highly useful. This information would need to be transmitted as quickly as possible to a protection scheme that would be more dependent on this state. An example of this type of situation would be changing the relay group settings dynamically depending on the state of the power system. Rather than use a binary state assessment, a range of operating conditions could be selected such that the internal protection settings of the relays at the critical location could be updated in real-time to bias the scheme towards dependability or security.

6.2 Main Contributions

Based on the research presented and the conclusions drawn from the results collected, the significant contributions from this thesis are listed below.

1. Implementation of both Heavy Winter and Heavy Summer decision trees in a hardware-based phasor data concentrator: SEL-3378.
2. Implementation of both Heavy Winter and Heavy Summer decision trees in a software-based phasor data concentrator: openPDC. Dynamic switching of decision trees through C37.118-2005 communications from phasor measurement unit.
3. Development and implementation of voting logic using a programmable logic controller for the software-based solution.
4. Development and testing of wide area measurement system device latency. Phasor measurement unit response and reporting latency, phasor data concentrator reporting latency and response to bad data.
5. Testing of additional latencies induced from voting logic and configuration in both the hardware- and software-based implementations.

6.3 Future Work

The use of decision trees for adaptive protection such as the Security-Dependability Adaptive Voting Scheme was realized in this thesis using wide area measurement technology. Using phasor measurement units, phasor data concentrators, programmable logic controller, and relays, a practical protection scheme was developed and deployed in a laboratory setting.

Two manufacturers of PDC were used for implementing the adaptive protection scheme, but this leaves room for improvement. As WAMS technology continues to improve and expand, a number of new products are coming onto the market that have the capabilities and capacities to perform the necessary functions to deploy the scheme presented. Utilities are exploring options other than the SEL-3378 and openPDC for implementation of this scheme for awareness and monitoring purposes. The advantage of these devices chosen for testing was that they were PDCs with the added functionality of a PLC. Similar but different would be a PLC that is capable of PDC functionality. This would have a different effect on the configuration and latency of the scheme. As the industry continues to evolve phasor technology, controller devices with the ability to read C37.118-2005 packets are becoming cheaper and more diverse.

Another venue for improvement would be to expand upon the control signal technology available and deploy more efficient and widely used protocols or options. For example, if the PDC is located at the critical location where the voting takes place, hard-wired copper could be used as an arming signal rather than Ethernet communications. This would most likely reduce the time required to act on the signal provided to the relays. Similarly, the GE N60 Universal Relay using DNP V3.00 has a Binary Input scanning period of 8 scans per power system cycle, which would be about 2 ms on a 60Hz system [44]. If the PDC could output binary statuses over DNP V3.00, the control signal latency would be reduced drastically. Along with reducing the time requirements, these options would allow interoperability of devices between manufacturers (an ancillary objective of standards).

The voting scheme implemented using openPDC and the Rockwell Automation PLC has room for expansion. A user could benefit from a human-machine interface (HMI) that could display the current state of the system. This could be in the form of a dynamically updated decision tree, with the current state highlighted to indicate to the user what the current attribute values are and where the system lies relative to Stressed or Safe operating condition. Furthermore, the PLC could also be equipped with a database such that the incoming relay trip signals, breaker trip output, and arming state changes from openPDC could be time stamped and stored for offline analysis.

Phasor measurement unit latency testing fell short due to lack of resources, primarily the lack of time synchronization of a PC. With this tool, one could analyze top-of-second (TOS) packets from the PMU as they arrive at the PC to determine how long the PMU actually required to collect and transmit the data in the packet. This requires a highly accurate time-synchronized PC with a miniscule time of querying the system time. This was not realizable on a Windows-based PC but could be programmed into a Linux-based machine, and run using a Symmetricom Bus Level Timing Card [45]. Another option could be a programmable logic controller that can read C37.118-2005 data and accurately time stamp when the measurements are received.

This work is part of a funded project by the California Institute for Energy and Environment, Pacific Gas and Electric, and Southern California Edison. Following demonstration of the implementations in the lab, their intent is to reproduce a similar arrangement to what has been presented. This will require some amount of future consultation and guidance as the laboratory implementations take place. Also, upon successful implementation, it is hoped that useful data will be generated using simulations such that data analysis will confirm the benefits and further areas for improvement with this configuration.

References

1. Horowitz, S.H., and A.G. Phadke, *Power System Relaying*, 3rd ed., Chichester, England: Wiley/Research Studies , 2008.

2. IEEE PES PSRC, WG C-6, Wide Area Protection and Emergency Control, http://www.pes-psrc.org/Reports/.

3. Trhan, N. K., Saran, R., *Electric Utility Deregulation: Failure or Success*, Nuclear Science Symposium Conference Record, IEEE, Vol. 7, pp. 4614-4617, 2004.

4. North American SynchroPhasor Initiative (NASPI), http://www.naspi.org/.

5. Western Electricity Coordinating Council (WECC), Western Interconnect Synchrophasor Program (WISP), http://www.wecc.biz/awareness/Pages/WISP.aspx.

6. Schweitzer Engineering Laboratories, Corporate News, *BPA Selects SEL-487E PMU for WISP Program*, http://www.selinc.com/news.aspx?id=8919.

7. Phadke, A. G., Thorp, J. S., Adamiak, M. G., *A New Measurement Technique for Tracking Voltage Phasors, Local System Frequency, and Rate of Change of Frequency*, IEEE Trans. on Power Apparatus and Systems, Vol. PAS-102, No. 5, May 1983.

8. *IEEE Standard for Synchrophasors for Power Systems*. IEEE Std C37.118-2005 (Revision of IEEE Std 1344-1995), 22 March 2006.

9. Phadke, A. G., *Synchronized Phasor Measurement in Power Systems*, IEEE Computer Applications in Power, April 1993.

10. Li, J., Xie, X., Xiao, J., *The Framework and Algorithm of a New Phasor Measurement Unit*, IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies, April 2004.

11. NASPI Performance & Standards Task Team (PSTT), *Phasor Data Concentrator Functional and Performance Requirements*, Draft, 3 May 2011.

12. Bernabeu, E., *Methodology for a Security-Dependability Adaptive Protection Scheme based on Data Mining*, in ECE. 2009, Virginia Polytechnic Institute and State University: Blacksburg.

13. IEEE PES PSRC, WG C-12, *Performance of Relaying during Wide-Area Stressed Conditions*, 2008.

14. Wang, G., et al., *Granular Computing Based Data Mining in Views of Rough Set and Fuzzy Set*, Chongqing Univ. of Posts and Telecomm., China.

15. Mori, H., *State-of-the-Art Overview on Data Mining in Power Systems*, Power Systems Conference and Exposition, PDCE, pp. 33-34, 2006.

16. Van Cutsem, T., et al., *Decision tree approaches to voltage security assessment*, IEE Proceedings-C, Vol. 140, No. 3, May 1993.

17. Thomas, M. K., *Implementation of the Security-Dependability Adaptive Voting Scheme*, in ECE. 2011, Virginia Polytechnic Institute and State University: Blacksburg.

18. *IEEE Standard for Synchrophasors for Power Systems*. IEEE Std. 1344-1995 (R2001).

19. IEEE PES PSRC, WG C-6, *Wide Area Protection and Emergency Control*, Accessed June 2011.

20. Luss, H., Rosenwein, M. B., Wong, R. T., *Topological Network Design for SONET Ring Architecture*, IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 28, No. 6, November 1998.

21. IEEE PSRC, WG H-14, *Glossary of Communications Terms For Relay Engineers*, 2008.

22. ABB, *Type PM, 550kV, up to 80kA, Power Circuit Breaker*, May 2009, http://www.abb.com/highvoltage.

23. Mitsubishi Electric Power Products, Inc., *SF6 Circuit Breaker*, Accessed July 2011, http://www.meppi.com/.

24. Schweitzer Engineering Laboratories, *SEL-3378 Synchrophasor Vector Processor Instruction Manual*, 2010.

25. openPDC, http://openpdc.codeplex.com/.

26. Doble, *F6150 Power System Simulator*, http://www.doble.com/products/f6150.html.

27. Schweitzer Engineering Laboratories, *SEL-3378 Synchrophasor Vector Processor Data Sheet*, 2008.

28. Schweitzer Engineering Laboratories, *SEL-2515 Remote I/O Module Instruction Manual*, 2008.

29. Schweitzer Engineering Laboratories, *Fast Operate Command Processing*, SEL Application Guide, Volume II, Accessed July 2011.

30. Schweitzer Engineering Laboratories, http://www.selinc.com.

31. PMU Connection Tester, http://www.pmuconnectiontester.codeplex.com.

32. Rockwell Automation, *1756 ControlLogix Controllers Specifications*, Technical Data, 2011.

33. MySQL, Oracle Corporation, 2011, http://www.mysql.com.

34. Microsoft Corporation, Microsoft SQL Server, 2011, http://www.microsoft.com/sqlserver/en/us/default.aspx.

35. Rockwell Automation, ControlLogix 1756 System, http://www.ab.com/en/epub/catalogs/12762/2181376/2416247/360807/360809/.

36. Rockwell Automation, RSLogix 5000, http://www.rockwellautomation.com/rockwellsoftware/design/rslogix5000/.

37. Schweitzer Engineering Laboratories, *SEL-2407 Satellite-Synchronized Clock Instruction Manual*, 2011.

38. Wireshark, Wireshark Foundation, http://www.wireshark.org.

39. SmartBits, Spirent, http://www.spirent.com/Solutions-Directory/Smartbits.aspx.

40. Bonneville Power Administration, FTP, http://www.bpa.gov/ftp/ftpretrieve.shtml.

41. Schweitzer Engineering Laboratories, *SEL-3373 Station Phasor Data Concentrator Data Sheet, 2011*.

42. Allen-Bradley, *ControlLogix Sequence of Event Module 1756-IB16ISOE User Manual*, April 2004.

43. General Electric Digital Energy Multilin, *GE N60 Network Stability and Synchrophasor Measurement System UR Series Instruction Manual*, N60 Revision 5.7x, pp. E-2.

44. Symmetricom, Inc., *PCI / PCIe Time and Frequency Processor User's Guide*, Revision A, 2009.

45. Western Electricity Coordinating Council (WECC), http://www.wecc.biz/.

46. Bonneville Power Administration, http://www.bpa.gov/corporate/.

Appendix A

MATLAB If-Else Logic Generator

```
%This function reads in the CART file manually created, and generates an
%if/else representation of the decision tree.
%
%The output of this function is a .m file that will be called continuously
%by the Security Assessment function for determinig the system state.
function [] = buildTree(CARTFile)

fid = fopen(CARTFile); % CARTFile must be formatted to a specific standard
C = textscan(fid, '%s %f %s %f');
fclose(fid);

LTCount = 0; RTCount = 0; IfCount = 0;
TotLTCount = 0; TotRTCount = 0; TotIfCount = 0;

outFile = fopen('treeOutput.m','w');

% Determine counts for IFs RTs and LTs for the whole tree.
for x = 1:length(C{1})
    if strcmpi(C{1}(x),'O') || strcmpi(C{1}(x),'L') || strcmpi(C{1}(x),'R') || strcmpi(C{1}(x),'RI')
        TotIfCount = TotIfCount + 1;
    elseif strcmpi(C{1}(x),'LT')
        TotLTCount = TotLTCount + 1;
    elseif strcmpi(C{1}(x),'RT')
        TotRTCount = TotRTCount + 1;
    end
end

endC = TotIfCount - TotRTCount;
```

```
s = sprintf('function [ FLAG ] = treeOutput(PDCstream)\n');
fprintf(outFile,s);

% Build tree from input file.
for x = 1:length(C{1})
    % Options for node types are:
    % O:  Origin Node
    % L:  Left Child Node
    % LT: Left Terminal Node
    % R:  Right Child Node
    % RT: Right Terminal Node
    % RI: Initial Right Child Node (used for counter)

    if strcmpi(C{1}(x),'O')
        s = sprintf('if( %s <= %f )\n',char(C{3}(x)),C{4}(x));
        IfCount = IfCount + 1;

    elseif strcmpi(C{1}(x),'L')
        s = sprintf('if( %s <= %f )\n',char(C{3}(x)),C{4}(x));
        IfCount = IfCount + 1;

    elseif strcmpi(C{1}(x),'LT')
        s = sprintf('%s\n',char(C{3}(x)));
        LTCount = LTCount + 1;

    elseif strcmpi(C{1}(x),'R')
        s = sprintf('else\nif( %s <= %f )\n',char(C{3}(x)),C{4}(x));
        IfCount = IfCount + 1;

    elseif strcmpi(C{1}(x),'RT')
        for y = 1:abs(C{2}(x)-C{2}(x-1))-1
            s = sprintf('end\n');
            fprintf(outFile,s);
            endC = endC - 1;
        end

        s = sprintf('else\n%s\nend\n',char(C{3}(x)));
```

```
        RTCount = RTCount + 1;

    elseif strcmpi(C{1}(x),'RI')

        if TotIfCount - TotRTCount > 0 && LTCount > RTCount
            s = sprintf('end\n');
            fprintf(outFile,s);
            if endC > 0
                endC = endC - 1;
            end
        end

        s = sprintf('else\nif( %s <= %f )\n',char(C{3}(x)),C{4}(x));
        IfCount = IfCount + 1;

    else
        error('Incorrect node classification in input txt file.')
    end

    fprintf(outFile,s);

end

while endC >= 1
    for x = 1:endC
        s = sprintf('end\n');
        fprintf(outFile,s);
        endC = endC - 1;
    end
end

if x == length(C{1})
    s = sprintf('Tree built')
end

end
```

Heavy Summer Decision Tree Input .txt File

O	1	IR19	16.52
L	2	II735	-0.47
L	3	IR415	-1.05
LT	4	STRESSED=1;	0.0
RT	4	STRESSED=0;	0.0
RT	3	STRESSED=0;	0.0
R	2	II735	-0.44
LT	3	STRESSED=1;	0.0
R	3	IR19	16.92
LT	4	STRESSED=0;	0.0
RT	4	STRESSED=1;	0.0

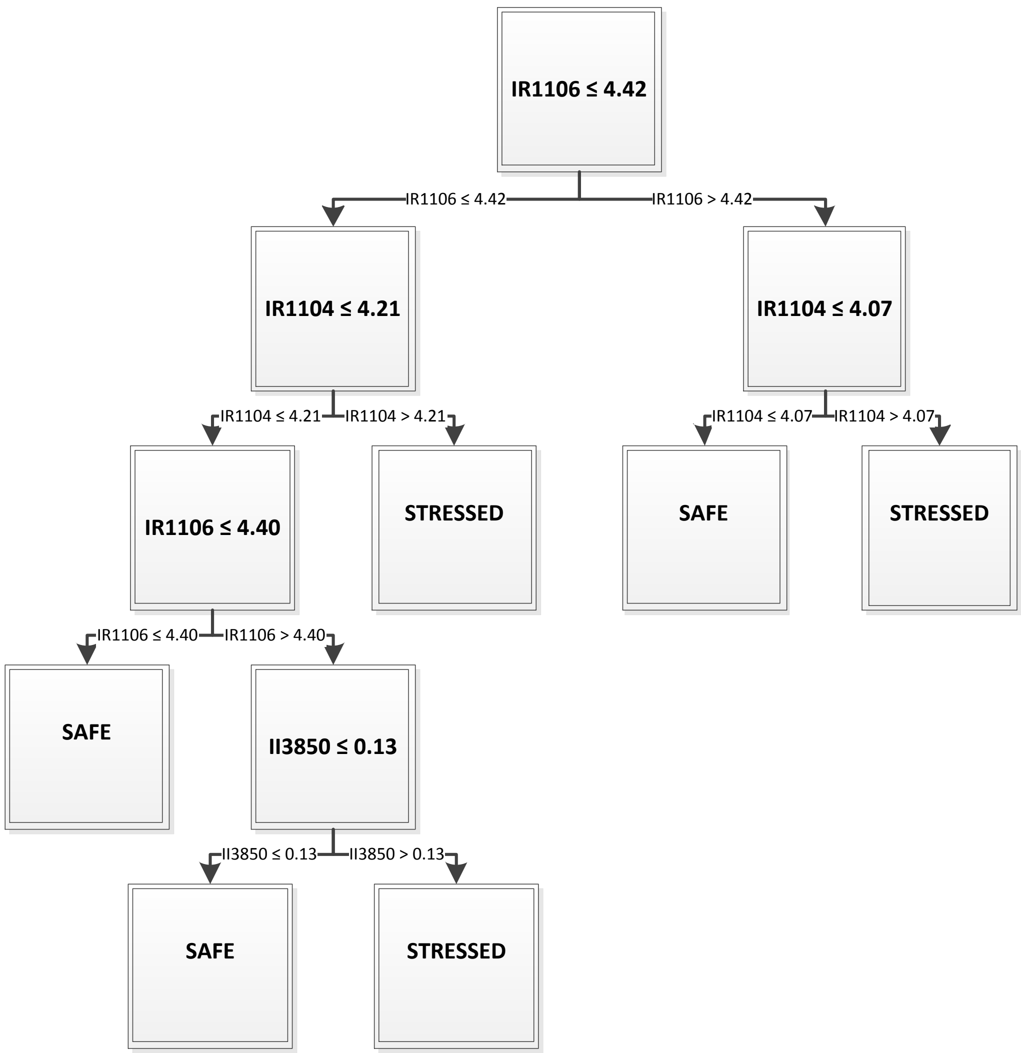
Heavy Summer Decision Tree Input .txt File

O	1	IR1106	4.42
L	2	IR1104	4.21
L	3	IR1106	4.40
LT	4	STRESSED:=0;	0.00
R	4	II3850	0.13
LT	5	STRESSED:=0;	0.00
RT	5	STRESSED:=1;	0.00
RT	3	STRESSED:=1;	0.00
RI	2	IR1104	4.06
LT	3	STRESSED:=0;	0.00
RT	3	STRESSED:=1;	0.00

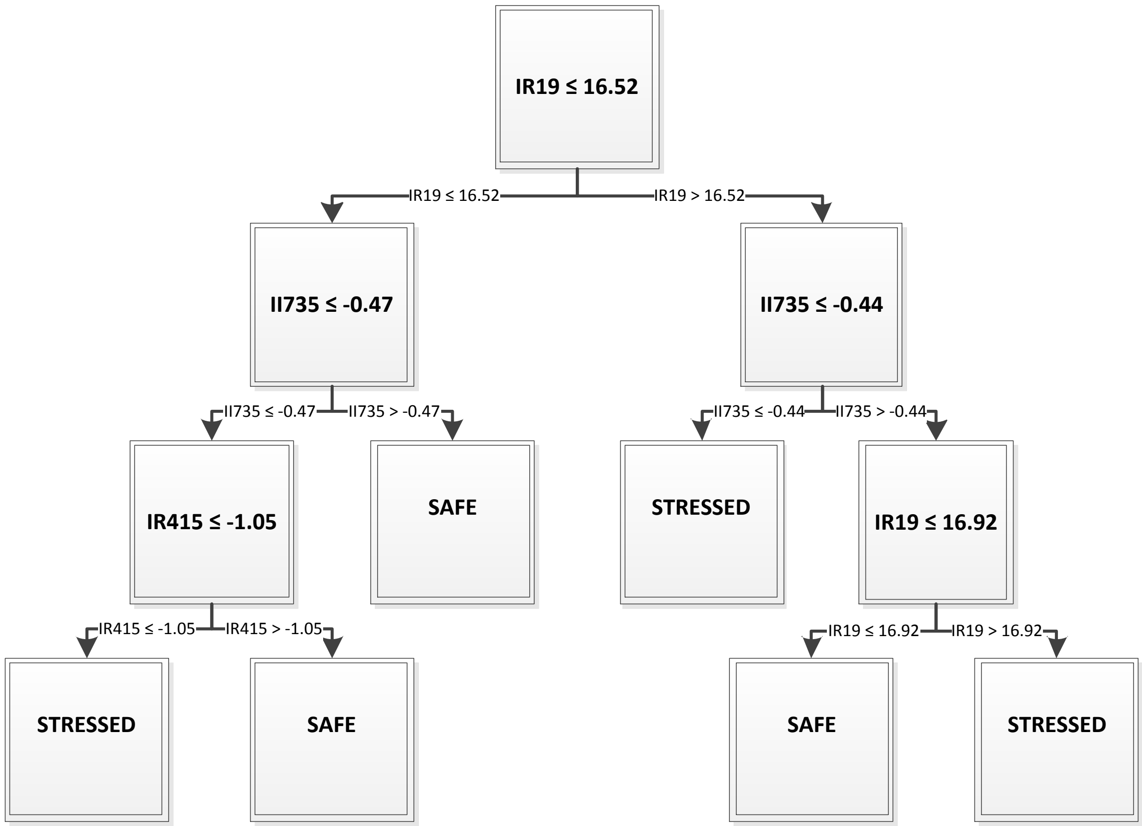
Appendix B

Heavy Winter and Heavy Summer Decision Trees

Heavy Winter Decision Tree:



Heavy Summer Decision Tree:



Appendix C

MATLAB Amplitude Step Test

```
% ----------------------------------------
% AstepVI.m
%
% Amplitude step from 100-110% of nominal
% for voltage and current
%
% ----------------------------------------

disp('Generating test file');

clear all
close all

nomFrq=60;           %Nominal frequency
numPhases=6;         %Total number of phases
sampRate=5760;       %Sample rate of generated file
step=1/sampRate;     %Time increment of one sample

%Define voltage levels, current levels, and time at each level
Vlevels=[1 1.1 1 1.1 1 1.1 1 1.1 1 1.1 1 1.1 1 1.1 1 1.1 1 1.1 1 1.1 1 ...
    1.1 1 1.1 1 1.1 1 1.1 1 1.1 1 1.1 1 1.1 1];
Ilevels=Vlevels;
numVlevels=length(Vlevels);
Dwells = [2 4*ones(1,34)];

%Shift each step within the measurement window
cyc=1/nomFrq;
timeShift=[0 0  .125*cyc  -.125*cyc  .25*cyc  -.25*cyc  .375*cyc  -.375*cyc  .5*cyc  -.5*cyc...
```

```
                .625*cyc  -.625*cyc  .75*cyc  -.75*cyc  .875*cyc  -.875*cyc     cyc     -cyc...
               1.125*cyc -1.125*cyc 1.25*cyc -1.25*cyc 1.375*cyc -1.375*cyc 1.5*cyc -1.5*cyc...
               1.625*cyc -1.625*cyc 1.75*cyc -1.75*cyc 1.875*cyc -1.875*cyc   2*cyc   -2*cyc 0];
timeShift=timeShift*.5;
Dwells = Dwells + timeShift;
shiftSteps=16;

%Determine number of samples in file
numPoints = sum(Dwells) * sampRate;
numSampPts = sum(Dwells(2:end)) * sampRate;

%Initialize flags
startFlag = zeros(numPoints,1);
sampFlag = zeros(numPoints,1);

%Create array for signals
signals(:,1)=[0:1/sampRate:sum(Dwells)];
mark=1;
F=1;
trig=0;
ang=[0 -2*pi/3 2*pi/3 0 -2*pi/3 2*pi/3];    %Angle shifts per phase

%Run loop to fill arrays with signals
for K=1:numVlevels
    DwellTime=Dwells(K)*sampRate;
    mark2=mark+DwellTime-1;
    points=signals(mark:mark2,1)-((mark-1)/sampRate);
    for P=1:numPhases
        if P<4
            signals(mark:mark2,P+1)=Vlevels(K)*sin(2*pi*nomFrq*signals(mark:mark2,1)+ang(P));
        else
            signals(mark:mark2,P+1)=Ilevels(K)*sin(2*pi*nomFrq*signals(mark:mark2,1)+ang(P));
        end
    end

    %sample flag
    if Vlevels(K) ~= 1
        lowBound=F+timeShift(K)*sampRate;
```

```
        upBound=F-1+sampRate*Dwells(K);
        sampFlag(lowBound:upBound)=1;
        trig=1;
    end

    %start flag
    if trig==1
        startFlag(F:F-1+sampRate*Dwells(K)) = 1;
    end

    F = F+sampRate*Dwells(K);

    mark=mark2+1;

end

%Adjust flags for use in analysis
signals=signals(1:end-1,:);
sampFlag=sampFlag(numPoints-numSampPts+1:end);

%Save relevant variables of test file
test_type = 'Astep';
savefile = 'C:\PMU_Matlab\SourceFiles\IEEE37118Tests\AstepVI.mat';
save(savefile, 'signals', 'Dwells', 'startFlag', 'sampFlag', 'test_type', 'nomFrq',...
    'Vlevels', 'Ilevels', 'numVlevels', 'timeShift', 'shiftSteps', 'sampRate');

disp('Test file generated');
disp([savefile ' saved']);

clear all
```

Appendix D

Sequential Events Reports

Below are the SER Event Reports generated for the latency tests performed.

Simple Threshold – Step Test

#	DATE	TIME	ELEMENT	STATE
16	07/18/2011	21:50:02.1176	RB01	Asserted
15	07/18/2011	21:50:06.1051	RB01	Deasserted
14	07/18/2011	21:50:10.1091	RB01	Asserted
13	07/18/2011	21:50:14.1151	RB01	Deasserted
12	07/18/2011	21:50:18.1401	RB01	Asserted
11	07/18/2011	21:50:22.1651	RB01	Deasserted
10	07/18/2011	21:50:26.1716	RB01	Asserted
9	07/18/2011	21:50:30.1571	RB01	Deasserted
8	07/18/2011	21:50:34.1631	RB01	Asserted
7	07/18/2011	21:50:38.1301	RB01	Deasserted
6	07/18/2011	21:50:42.1551	RB01	Asserted
5	07/18/2011	21:50:46.1591	RB01	Deasserted
4	07/18/2011	21:50:50.1651	RB01	Asserted
3	07/18/2011	21:50:54.1131	RB01	Deasserted
2	07/18/2011	21:50:58.1571	RB01	Asserted
1	07/18/2011	21:51:02.1051	RB01	Deasserted

#	DATE	TIME	ELEMENT	STATE
16	07/18/2011	21:57:02.1071	RB01	Asserted
15	07/18/2011	21:57:06.1321	RB01	Deasserted
14	07/18/2011	21:57:10.1176	RB01	Asserted
13	07/18/2011	21:57:14.1236	RB01	Deasserted
12	07/18/2011	21:57:18.1301	RB01	Asserted
11	07/18/2011	21:57:22.1341	RB01	Deasserted
10	07/18/2011	21:57:26.1401	RB01	Asserted
9	07/18/2011	21:57:30.1071	RB01	Deasserted
8	07/18/2011	21:57:34.1131	RB01	Asserted
7	07/18/2011	21:57:38.1196	RB01	Deasserted
6	07/18/2011	21:57:42.1051	RB01	Asserted
5	07/18/2011	21:57:46.1091	RB01	Deasserted
4	07/18/2011	21:57:50.1551	RB01	Asserted
3	07/18/2011	21:57:54.1426	RB01	Deasserted
2	07/18/2011	21:57:58.1466	RB01	Asserted
1	07/18/2011	21:58:02.1526	RB01	Deasserted

#	DATE	TIME	ELEMENT	STATE
16	07/18/2011	22:03:02.1651	RB01	Asserted
15	07/18/2011	22:03:06.1131	RB01	Deasserted
14	07/18/2011	22:03:10.1196	RB01	Asserted
13	07/18/2011	22:03:14.1051	RB01	Deasserted
12	07/18/2011	22:03:18.1486	RB01	Asserted
11	07/18/2011	22:03:22.1736	RB01	Deasserted
10	07/18/2011	22:03:26.1611	RB01	Asserted
9	07/18/2011	22:03:30.1651	RB01	Deasserted
8	07/18/2011	22:03:34.1526	RB01	Asserted
7	07/18/2011	22:03:38.1196	RB01	Deasserted
6	07/18/2011	22:03:42.1256	RB01	Asserted
5	07/18/2011	22:03:46.1111	RB01	Deasserted
4	07/18/2011	22:03:50.1176	RB01	Asserted
3	07/18/2011	22:03:54.1216	RB01	Deasserted
2	07/18/2011	22:03:58.1276	RB01	Asserted
1	07/18/2011	22:04:02.1341	RB01	Deasserted

Heavy Winter Model – Step Test

#	DATE	TIME	ELEMENT	STATE
16	07/26/2011	20:51:02.1661	RB01	Asserted
15	07/26/2011	20:51:06.1306	RB01	Deasserted
14	07/26/2011	20:51:10.1181	RB01	Asserted
13	07/26/2011	20:51:14.1621	RB01	Deasserted
12	07/26/2011	20:51:18.1681	RB01	Asserted
11	07/26/2011	20:51:22.1536	RB01	Deasserted
10	07/26/2011	20:51:26.1201	RB01	Asserted
9	07/26/2011	20:51:30.1476	RB01	Deasserted
8	07/26/2011	20:51:34.1726	RB01	Asserted
7	07/26/2011	20:51:38.1576	RB01	Deasserted
6	07/26/2011	20:51:42.1641	RB01	Asserted
5	07/26/2011	20:51:46.1101	RB01	Deasserted
4	07/26/2011	20:51:50.1556	RB01	Asserted
3	07/26/2011	20:51:54.1411	RB01	Deasserted
2	07/26/2011	20:51:58.1661	RB01	Asserted
1	07/26/2011	20:52:02.1141	RB01	Deasserted

#	DATE	TIME	ELEMENT	STATE
16	07/26/2011	20:59:02.1181	RB01	Asserted
15	07/26/2011	20:59:06.1431	RB01	Deasserted
14	07/26/2011	20:59:10.1286	RB01	Asserted
13	07/26/2011	20:59:14.1351	RB01	Deasserted
12	07/26/2011	20:59:18.1411	RB01	Asserted
11	07/26/2011	20:59:22.1451	RB01	Deasserted
10	07/26/2011	20:59:26.1516	RB01	Asserted
9	07/26/2011	20:59:30.1576	RB01	Deasserted
8	07/26/2011	20:59:34.1621	RB01	Asserted
7	07/26/2011	20:59:38.1286	RB01	Deasserted
6	07/26/2011	20:59:42.1536	RB01	Asserted
5	07/26/2011	20:59:46.1411	RB01	Deasserted
4	07/26/2011	20:59:50.1451	RB01	Asserted
3	07/26/2011	20:59:54.1516	RB01	Deasserted
2	07/26/2011	20:59:58.1766	RB01	Asserted
1	07/26/2011	21:00:02.1431	RB01	Deasserted

#	DATE	TIME	ELEMENT	STATE
16	07/26/2011	21:02:02.1391	RB01	Asserted
15	07/26/2011	21:02:06.1431	RB01	Deasserted
14	07/26/2011	21:02:10.1496	RB01	Asserted
13	07/26/2011	21:02:14.1371	RB01	Deasserted
12	07/26/2011	21:02:18.1411	RB01	Asserted
11	07/26/2011	21:02:22.1476	RB01	Deasserted
10	07/26/2011	21:02:26.1536	RB01	Asserted
9	07/26/2011	21:02:30.1201	RB01	Deasserted
8	07/26/2011	21:02:34.1246	RB01	Asserted
7	07/26/2011	21:02:38.1306	RB01	Deasserted
6	07/26/2011	21:02:42.1371	RB01	Asserted
5	07/26/2011	21:02:46.1411	RB01	Deasserted
4	07/26/2011	21:02:50.1661	RB01	Asserted
3	07/26/2011	21:02:54.1326	RB01	Deasserted
2	07/26/2011	21:02:58.1391	RB01	Asserted
1	07/26/2011	21:03:02.1641	RB01	Deasserted

Heavy Winter Model – Ramp Test

#	DATE	TIME	ELEMENT	STATE
18	08/05/2011	20:39:52.1331	RB01	Asserted
17	08/05/2011	20:40:40.4146	RB01	Deasserted
16	08/05/2011	20:48:52.1351	RB01	Asserted
15	08/05/2011	20:49:19.0061	RB01	Deasserted
14	08/05/2011	20:52:52.1601	RB01	Asserted
13	08/05/2011	20:53:12.9666	RB01	Deasserted
12	08/05/2011	20:55:52.1416	RB01	Asserted
11	08/05/2011	20:56:00.9121	RB01	Deasserted
10	08/05/2011	20:58:52.1626	RB01	Asserted
9	08/05/2011	20:59:03.2271	RB01	Deasserted
8	08/05/2011	21:10:52.1601	RB01	Asserted
7	08/05/2011	21:10:56.0686	RB01	Deasserted
6	08/05/2011	21:13:52.1456	RB01	Asserted
5	08/05/2011	21:13:54.9831	RB01	Deasserted
4	08/05/2011	21:16:52.1271	RB01	Asserted
3	08/05/2011	21:17:01.2851	RB01	Deasserted
2	08/05/2011	21:19:52.1831	RB01	Asserted
1	08/05/2011	21:19:57.9581	RB01	Deasserted

Hardware PDC – Master-Slave Mirrored Bits Latency Testing

#	DATE	TIME	ELEMENT	STATE
12	10/27/2011	16:26:06.019	50P1	Asserted
11	10/27/2011	16:26:06.019	OUT104	Asserted
10	10/27/2011	16:26:06.019	TMB1A	ASSERTED
5	10/27/2011	16:26:06.026	TRIP	Asserted
4	10/27/2011	16:26:06.026	IN101	Asserted
3	10/27/2011	16:26:06.026	RMB1A	ASSERTED

3	10/27/2011	16:31:21.018	50P1	Asserted
2	10/27/2011	16:31:21.018	OUT104	Asserted
1	10/27/2011	16:31:21.018	TMB1A	ASSERTED
5	10/27/2011	16:31:21.026	TRIP	Asserted
4	10/27/2011	16:31:21.026	IN101	Asserted
3	10/27/2011	16:31:21.026	RMB1A	ASSERTED
18	10/27/2011	16:32:40.018	50P1	Asserted
17	10/27/2011	16:32:40.018	OUT104	Asserted
16	10/27/2011	16:32:40.018	TMB1A	ASSERTED
19	10/27/2011	16:32:40.026	IN101	Asserted
18	10/27/2011	16:32:40.028	TRIP	Asserted
17	10/27/2011	16:32:40.028	RMB1A	ASSERTED
6	10/27/2011	16:33:46.018	50P1	Asserted
5	10/27/2011	16:33:46.018	OUT104	Asserted
4	10/27/2011	16:33:46.018	TMB1A	ASSERTED
5	10/27/2011	16:33:46.026	IN101	Asserted
4	10/27/2011	16:33:46.028	TRIP	Asserted
3	10/27/2011	16:33:46.028	RMB1A	ASSERTED
3	10/27/2011	16:35:52.018	50P1	Asserted
2	10/27/2011	16:35:52.018	OUT104	Asserted
1	10/27/2011	16:35:52.018	TMB1A	ASSERTED
5	10/27/2011	16:35:52.026	IN101	Asserted
4	10/27/2011	16:35:52.028	TRIP	Asserted
3	10/27/2011	16:35:52.028	RMB1A	ASSERTED
3	10/27/2011	16:37:19.018	50P1	Asserted
2	10/27/2011	16:37:19.018	OUT104	Asserted
1	10/27/2011	16:37:19.018	TMB1A	ASSERTED
5	10/27/2011	16:37:19.025	IN101	Asserted
4	10/27/2011	16:37:19.028	TRIP	Asserted
3	10/27/2011	16:37:19.028	RMB1A	ASSERTED
3	10/27/2011	16:38:27.018	50P1	Asserted
2	10/27/2011	16:38:27.018	OUT104	Asserted
1	10/27/2011	16:38:27.018	TMB1A	ASSERTED
5	10/27/2011	16:38:27.025	TRIP	Asserted
4	10/27/2011	16:38:27.025	IN101	Asserted
3	10/27/2011	16:38:27.025	RMB1A	ASSERTED
3	10/27/2011	16:39:32.020	50P1	Asserted
2	10/27/2011	16:39:32.020	OUT104	Asserted
1	10/27/2011	16:39:32.020	TMB1A	ASSERTED
5	10/27/2011	16:39:32.027	IN101	Asserted
4	10/27/2011	16:39:32.032	TRIP	Asserted
3	10/27/2011	16:39:32.032	RMB1A	ASSERTED

6	10/27/2011	16:40:47.020	50P1	Asserted
5	10/27/2011	16:40:47.020	OUT104	Asserted
4	10/27/2011	16:40:47.020	TMB1A	ASSERTED
5	10/27/2011	16:40:47.027	IN101	Asserted
4	10/27/2011	16:40:47.031	TRIP	Asserted
3	10/27/2011	16:40:47.031	RMB1A	ASSERTED
6	10/27/2011	16:41:59.020	50P1	Asserted
5	10/27/2011	16:41:59.020	OUT104	Asserted
4	10/27/2011	16:41:59.020	TMB1A	ASSERTED
5	10/27/2011	16:41:59.027	IN101	Asserted
4	10/27/2011	16:41:59.031	TRIP	Asserted
3	10/27/2011	16:41:59.031	RMB1A	ASSERTED
3	10/27/2011	16:43:23.020	50P1	Asserted
2	10/27/2011	16:43:23.020	OUT104	Asserted
1	10/27/2011	16:43:23.020	TMB1A	ASSERTED
5	10/27/2011	16:43:23.027	IN101	Asserted
4	10/27/2011	16:43:23.031	TRIP	Asserted
3	10/27/2011	16:43:23.031	RMB1A	ASSERTED
3	10/27/2011	16:44:27.020	50P1	Asserted
2	10/27/2011	16:44:27.020	OUT104	Asserted
1	10/27/2011	16:44:27.020	TMB1A	ASSERTED
5	10/27/2011	16:44:27.027	IN101	Asserted
4	10/27/2011	16:44:27.029	TRIP	Asserted
3	10/27/2011	16:44:27.029	RMB1A	ASSERTED
6	10/27/2011	16:46:43.019	50P1	Asserted
5	10/27/2011	16:46:43.019	OUT104	Asserted
4	10/27/2011	16:46:43.019	TMB1A	ASSERTED
5	10/27/2011	16:46:43.027	IN101	Asserted
4	10/27/2011	16:46:43.029	TRIP	Asserted
3	10/27/2011	16:46:43.029	RMB1A	ASSERTED
3	10/27/2011	16:48:03.019	50P1	Asserted
2	10/27/2011	16:48:03.019	OUT104	Asserted
1	10/27/2011	16:48:03.019	TMB1A	ASSERTED
5	10/27/2011	16:48:03.027	IN101	Asserted
4	10/27/2011	16:48:03.029	TRIP	Asserted
3	10/27/2011	16:48:03.029	RMB1A	ASSERTED
3	10/27/2011	16:49:12.019	50P1	Asserted
2	10/27/2011	16:49:12.019	OUT104	Asserted
1	10/27/2011	16:49:12.019	TMB1A	ASSERTED
5	10/27/2011	16:49:12.027	IN101	Asserted
4	10/27/2011	16:49:12.029	TRIP	Asserted
3	10/27/2011	16:49:12.029	RMB1A	ASSERTED

3	10/27/2011	16:50:25.019	50P1	Asserted
2	10/27/2011	16:50:25.019	OUT104	Asserted
1	10/27/2011	16:50:25.019	TMB1A	ASSERTED
5	10/27/2011	16:50:25.027	IN101	Asserted
4	10/27/2011	16:50:25.029	TRIP	Asserted
3	10/27/2011	16:50:25.029	RMB1A	ASSERTED
3	10/27/2011	16:51:50.019	50P1	Asserted
2	10/27/2011	16:51:50.019	OUT104	Asserted
1	10/27/2011	16:51:50.019	TMB1A	ASSERTED
5	10/27/2011	16:51:50.027	IN101	Asserted
4	10/27/2011	16:51:50.031	TRIP	Asserted
3	10/27/2011	16:51:50.031	RMB1A	ASSERTED
3	10/27/2011	16:52:48.019	50P1	Asserted
2	10/27/2011	16:52:48.019	OUT104	Asserted
1	10/27/2011	16:52:48.019	TMB1A	ASSERTED
5	10/27/2011	16:52:48.027	IN101	Asserted
4	10/27/2011	16:52:48.029	TRIP	Asserted
3	10/27/2011	16:52:48.029	RMB1A	ASSERTED
3	10/27/2011	16:53:49.019	50P1	Asserted
2	10/27/2011	16:53:49.019	OUT104	Asserted
1	10/27/2011	16:53:49.019	TMB1A	ASSERTED
5	10/27/2011	16:53:49.026	IN101	Asserted
4	10/27/2011	16:53:49.029	TRIP	Asserted
3	10/27/2011	16:53:49.029	RMB1A	ASSERTED
3	10/27/2011	16:54:48.019	50P1	Asserted
2	10/27/2011	16:54:48.019	OUT104	Asserted
1	10/27/2011	16:54:48.019	TMB1A	ASSERTED
5	10/27/2011	16:54:48.026	IN101	Asserted
4	10/27/2011	16:54:48.029	TRIP	Asserted
3	10/27/2011	16:54:48.029	RMB1A	ASSERTED

Hardware PDC – Hard-wired Latency Testing

#	DATE	TIME	ELEMENT	STATE
6	10/27/2011	17:05:08.018	50P1	Asserted
5	10/27/2011	17:05:08.018	OUT104	Asserted
4	10/27/2011	17:05:08.018	TMB1A	ASSERTED
5	10/27/2011	17:05:08.026	TRIP	Asserted
4	10/27/2011	17:05:08.026	IN101	Asserted
3	10/27/2011	17:05:08.030	RMB1A	ASSERTED
3	10/27/2011	17:06:38.018	50P1	Asserted
2	10/27/2011	17:06:38.018	OUT104	Asserted
1	10/27/2011	17:06:38.018	TMB1A	ASSERTED
5	10/27/2011	17:06:38.026	TRIP	Asserted
4	10/27/2011	17:06:38.026	IN101	Asserted
3	10/27/2011	17:06:38.030	RMB1A	ASSERTED

3	10/27/2011	17:07:35.018	50P1	Asserted
2	10/27/2011	17:07:35.018	OUT104	Asserted
1	10/27/2011	17:07:35.018	TMB1A	ASSERTED
5	10/27/2011	17:07:35.026	TRIP	Asserted
4	10/27/2011	17:07:35.026	IN101	Asserted
3	10/27/2011	17:07:35.030	RMB1A	ASSERTED
3	10/27/2011	17:08:32.018	50P1	Asserted
2	10/27/2011	17:08:32.018	OUT104	Asserted
1	10/27/2011	17:08:32.018	TMB1A	ASSERTED
5	10/27/2011	17:08:32.026	TRIP	Asserted
4	10/27/2011	17:08:32.026	IN101	Asserted
3	10/27/2011	17:08:32.030	RMB1A	ASSERTED
3	10/27/2011	17:09:29.018	50P1	Asserted
2	10/27/2011	17:09:29.018	OUT104	Asserted
1	10/27/2011	17:09:29.018	TMB1A	ASSERTED
5	10/27/2011	17:09:29.026	TRIP	Asserted
4	10/27/2011	17:09:29.026	IN101	Asserted
3	10/27/2011	17:09:29.030	RMB1A	ASSERTED
3	10/27/2011	17:10:26.018	50P1	Asserted
2	10/27/2011	17:10:26.018	OUT104	Asserted
1	10/27/2011	17:10:26.018	TMB1A	ASSERTED
5	10/27/2011	17:10:26.026	TRIP	Asserted
4	10/27/2011	17:10:26.026	IN101	Asserted
3	10/27/2011	17:10:26.030	RMB1A	ASSERTED
3	10/27/2011	17:11:23.018	50P1	Asserted
2	10/27/2011	17:11:23.018	OUT104	Asserted
1	10/27/2011	17:11:23.018	TMB1A	ASSERTED
5	10/27/2011	17:11:23.026	TRIP	Asserted
4	10/27/2011	17:11:23.026	IN101	Asserted
3	10/27/2011	17:11:23.030	RMB1A	ASSERTED
3	10/27/2011	17:12:20.020	50P1	Asserted
2	10/27/2011	17:12:20.020	OUT104	Asserted
1	10/27/2011	17:12:20.020	TMB1A	ASSERTED
5	10/27/2011	17:12:20.028	TRIP	Asserted
4	10/27/2011	17:12:20.028	IN101	Asserted
3	10/27/2011	17:12:20.030	RMB1A	ASSERTED
6	10/27/2011	17:13:24.020	50P1	Asserted
5	10/27/2011	17:13:24.020	OUT104	Asserted
4	10/27/2011	17:13:24.020	TMB1A	ASSERTED
5	10/27/2011	17:13:24.028	TRIP	Asserted
4	10/27/2011	17:13:24.028	IN101	Asserted
3	10/27/2011	17:13:24.028	RMB1A	ASSERTED

3	10/27/2011	17:14:24.020	50P1	Asserted
2	10/27/2011	17:14:24.020	OUT104	Asserted
1	10/27/2011	17:14:24.020	TMB1A	ASSERTED
5	10/27/2011	17:14:24.026	RMB1A	ASSERTED
4	10/27/2011	17:14:24.028	TRIP	Asserted
3	10/27/2011	17:14:24.028	IN101	Asserted
3	10/27/2011	17:15:26.020	50P1	Asserted
2	10/27/2011	17:15:26.020	OUT104	Asserted
1	10/27/2011	17:15:26.020	TMB1A	ASSERTED
19	10/27/2011	17:15:26.028	TRIP	Asserted
18	10/27/2011	17:15:26.028	IN101	Asserted
17	10/27/2011	17:15:26.030	RMB1A	ASSERTED
3	10/27/2011	17:16:23.020	50P1	Asserted
2	10/27/2011	17:16:23.020	OUT104	Asserted
1	10/27/2011	17:16:23.020	TMB1A	ASSERTED
5	10/27/2011	17:16:23.028	TRIP	Asserted
4	10/27/2011	17:16:23.028	IN101	Asserted
3	10/27/2011	17:16:23.030	RMB1A	ASSERTED
3	10/27/2011	17:17:47.020	50P1	Asserted
2	10/27/2011	17:17:47.020	OUT104	Asserted
1	10/27/2011	17:17:47.020	TMB1A	ASSERTED
5	10/27/2011	17:17:47.028	TRIP	Asserted
4	10/27/2011	17:17:47.028	IN101	Asserted
3	10/27/2011	17:17:47.030	RMB1A	ASSERTED
3	10/27/2011	17:20:16.019	50P1	Asserted
2	10/27/2011	17:20:16.019	OUT104	Asserted
1	10/27/2011	17:20:16.019	TMB1A	ASSERTED
5	10/27/2011	17:20:16.027	TRIP	Asserted
4	10/27/2011	17:20:16.027	IN101	Asserted
3	10/27/2011	17:20:16.027	RMB1A	ASSERTED
3	10/27/2011	17:21:13.019	50P1	Asserted
2	10/27/2011	17:21:13.019	OUT104	Asserted
1	10/27/2011	17:21:13.019	TMB1A	ASSERTED
5	10/27/2011	17:21:13.027	TRIP	Asserted
4	10/27/2011	17:21:13.027	IN101	Asserted
3	10/27/2011	17:21:13.027	RMB1A	ASSERTED
3	10/27/2011	17:22:10.019	50P1	Asserted
2	10/27/2011	17:22:10.019	OUT104	Asserted
1	10/27/2011	17:22:10.019	TMB1A	ASSERTED
5	10/27/2011	17:22:10.027	TRIP	Asserted
4	10/27/2011	17:22:10.027	IN101	Asserted
3	10/27/2011	17:22:10.027	RMB1A	ASSERTED

3	10/27/2011	17:23:12.019	50P1	Asserted
2	10/27/2011	17:23:12.019	OUT104	Asserted
1	10/27/2011	17:23:12.019	TMB1A	ASSERTED
5	10/27/2011	17:23:12.025	RMB1A	ASSERTED
4	10/27/2011	17:23:12.027	TRIP	Asserted
3	10/27/2011	17:23:12.027	IN101	Asserted
3	10/27/2011	17:24:12.019	50P1	Asserted
2	10/27/2011	17:24:12.019	OUT104	Asserted
1	10/27/2011	17:24:12.019	TMB1A	ASSERTED
5	10/27/2011	17:24:12.025	RMB1A	ASSERTED
4	10/27/2011	17:24:12.027	TRIP	Asserted
3	10/27/2011	17:24:12.027	IN101	Asserted
3	10/27/2011	17:26:12.019	50P1	Asserted
2	10/27/2011	17:26:12.019	OUT104	Asserted
1	10/27/2011	17:26:12.019	TMB1A	ASSERTED
5	10/27/2011	17:26:12.025	RMB1A	ASSERTED
4	10/27/2011	17:26:12.027	TRIP	Asserted
3	10/27/2011	17:26:12.027	IN101	Asserted
3	10/27/2011	17:27:11.019	50P1	Asserted
2	10/27/2011	17:27:11.019	OUT104	Asserted
1	10/27/2011	17:27:11.019	TMB1A	ASSERTED
5	10/27/2011	17:27:11.027	TRIP	Asserted
4	10/27/2011	17:27:11.027	IN101	Asserted
3	10/27/2011	17:27:11.027	RMB1A	ASSERTED