The Virginia Tech Phasor Data Concentrator Analysis & Testing System

Kunal S. Dekhane

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 Reelopez

Lamine M. Mili

December 12, 2011

Blacksburg, VA

Keywords: pdc testing, pdc, synchrophasor, latency, pmu simulator

The Virginia Tech Phasor Data Concentrator Analysis & Testing System

Kunal S. Dekhane

ABSTRACT

The development of Smart Grid and an increased emphasis on Wide Area Measurement, Automation, Protection and Control (WAMPAC) has lead to the substantial increase in the development and use of Synchrophasor Systems. The Department of Energy having realized its importance in the Power System has encouraged its deployment through the Smart Grid Investment Grant. With many utilities beginning to implement a large number of PMUs over their respective power systems, Phasor Data Concentrators (PDCs) play a crucial part in accurately relaying data from the point of measurement to the operators at the control center. The current Synchrophasor standard, IEEE C37.118-2005 covers adequately the steady state characterization of PMUs but does not specify requirements for PDCs. Having recognized the need for such a standard for PDCs, the North American Synchrophasor Initiative (NASPI) has developed a guide outlining some of its objectives, functions and tests requirements. Virginia Tech has developed a PDC Test System under these guidelines and as per the requirements of the PJM Synchrophasor Systems Deployment Project. This thesis focuses on the testing tools developed and the procedures implemented in the Virginia Tech PDC Test System.

Table of Contents

List of Figures

List of Tables

Definitions and Acronyms

This section provides some useful definitions and acronyms pertinent to this document.

DUT – Device Under Test. The device being tested by the test system.

GPS – Global Positioning System. A satellite based system for providing position and time. The accuracy of GPS based clocks can be better than 1 microsecond.

IEEE C37.118 – The new IEEE phasor data protocol that replaced the IEEE 1344 and the BPA/PDCStream protocols. Typically data is streamed in this format over UDP/IP or across a serial link.

IRIG-B – Time transmission formats developed by the Inter-Range Instrumentation Group (IRIG). The most common version is IRIG-B, which transmits day of year, hour, minute, and second once per second, over a 1 kHz carrier signal.

MATLAB – Commercial numerical computing environment and programming language.

PDC – Phasor Data Concentrator. A logical unit that collects phasor data, and discrete event data from PMUs and possibly from other PDCs, and transmits data to other applications. PDCs may buffer data for a short time period but do not store the data. [Note: PDC should not be confused with a PDU (or alternate acronyms) which is a similar device function for a hub of data packets in the EMS SCADA environment, concentrating SCADA data.]

PDC Processing Completion Time – The amount of time PDC uses to complete the production of an output stream. PDC Processing Completion Time is measured either between the time of

arrival of the last relevant PMU data or the end of the "Wait Time," whichever earlier, and the time PDC output stream starts leaving the PDC.

Phasor – A complex equivalent of a simple single-frequency cosine wave quantity such that the complex modulus is the cosine wave amplitude and the complex angle (in polar form) is the cosine wave phase angle. This complex number may be represented in a polar or rectangular form. In A.C. power systems, the phasor usually represents the power frequency (typically 50 Hz or 60 Hz) signal only.

PPS – Pulse-Per-Second. A signal consisting of a train of square pulses occurring at a frequency of 1 Hz, with the rising edge synchronized with UTC seconds. This signal is typically generated by GPS receivers.

PMU – Phasor Measurement Unit. A device that samples analog voltage and current data in synchronism with a GPS-clock. The samples are used to compute the corresponding phasors. Phasors are computed based on an absolute time reference (UTC), derived from a built in GPS receiver or time reference signals from an external GPS receiver. (See IPMU)

Relay – An electromechanical or electronic device applied to the purpose of power apparatus protection. A relay typically monitors voltages and currents associated with a certain power system device and may trip appropriate breakers when a potentially damaging condition is detected.

TCP/IP – TCP/IP is a low-level protocol for use mainly on Ethernet or related networks. Most of the higher-level protocols use TCP/IP to transport the data. TCP/IP provides a highly reliable connection over unreliable networks, using checksums, congestion control, and automatic

resending of bad or missing data. TCP/IP requires time to handshake new connections and will block if missing data is being resent.

UDP/IP – UDP/IP is a low-level IP protocol that provides low-latency communication across Ethernet or related networks. UDP/IP does not provide any error-control or resending of missing or bad data. The Application will need to check data for correctness. UDP/IP however, does not require time for handshaking and will not block, making it a good choice for *real-time* data communications.

UTC – Coordinated Universal Time (initials order based on French). UTC represents the time-of-day at the Earth's prime meridian (0° longitude).

Wait Time – The amount of time a PDC waits before it considers expected PMU data (or other relevant data) that is not yet received as "missing data." This time is measured with reference to either the PDC source of time (whether it is an internal PDC clock or a GPS referenced source of time used by the PDC) or the time of arrival of the first PMU data received.

WAMS – Wide area measurement system. Generally features one or more PMU networks as a "backbone," but may also include local recorders, legacy equipment, or advanced technologies that are GPS synchronized to the PMU networks while recording non-phasor data.

1. INTRODUCTION

Over the past two decades, a novel technique of measurement and monitoring of the wide area power system known as Synchrophasor systems has been in development and its need and use is growing not just in the United States of America, but all over the world. Synchrophasor systems, which first began to be known as Phasor Measurement Unit (PMU) takes real-time synchronized measurements of voltage, current & frequency at specific locations on the power system. PMUs must provide these voltage and current phasors in positive sequence but single phase, three phase, negative sequence, or zero sequence forms are optional.

The phasor data may be in integer or floating point format. All this information is streamed in pre-determined reporting rates, such as 10, 15, 30, 60 or 120 frames per second for a 60 Hz power system [1]. The novelty of the technique lies in the time synchronization of the measurements provided by a Global Positioning System (GPS). With every unit being time synchronized and placed at specific and strategic locations over a designated power system, the operator is provided with the luxury of observing a snapshot of the entire system after every sampled measurement.

With further development of phasor processing and acquisition systems built around the working of the PMU, Synchrophasor systems became broadly associated with not just wide-area measurement, but also real-time monitoring, control and protection. These systems typically consist of large number of PMUs, which generate multiple real-time data streams, and a number of applications that process and consume this data. [2]

These systems although being installed and commissioned by the respective entities are connected to each other so as to facilitate exchange of data amongst these systems. For example, a utility may transmit data from its own power system to an ISO that it belongs to and exchange data with neighboring utilities to gain a better view and understanding of a wider interconnected power grid. These Wide Area Monitoring, Protection, and Control (WAMPAC) systems integrate information from selected local networks to a remote location to stem the wide-spread effect of major disturbances. [3]

A Synchrophasor measurement data although taken at the same sampling instant by various PMUs typically will not reach the destination at that exact instant, mainly due to delays in communication paths used by the data streams known as network delays or transmission latency and due to differences in the processing and generation delays of the PMU known as PMU calculation latency [4]. Hence, before a real-time user application can decode and assimilate this data, it needs to be time-aligned, validated and processed. This is done by a Phasor Data Concentrator (PDC).

In 2007, the North America efforts in phasor technology were combined and the North American Synchro Phasor Initiative (NASPI) emerged with the intent to coordinate phasor activities in the entire North American grid. The increased role for industry collaborations of the NASPI working group and task teams has already extended to a more global collaboration of industry best practices, while the DOE continues to support phasor research. Today, there are seven task teams focusing on various aspects of phasor activities. [5]

Among the task teams is the Performance and Standards Task Team (PSTT). The PSTT is chartered to coordinate and act as liaison to standardization efforts and to determine consistent

and satisfactory performance of synchronized measurement devices and systems by creating guidelines and reports in accordance with best practices. Many of the PSTT members are active in many international industry activities which help the Task Team members to coordinate the development of phasor-related standards both within the NASPI as well as outside of North America. [6] The PSTT team developed two complementary documents, The Synchrophasor Accuracy Characterization [6] and Guide for Phasor Data Concentrator Requirements for Power System Protection, Control and Monitoring [5].

As a part of the DOE Smart Grid Investment Grant, PJM Interconnection, L.L.C funded the Synchrophasor Technology Deployment Project at the Power & Energy Lab at Virginia Tech. The primary objective of this project was to set up a Test Facility here in the Power Lab to be able to test PMUs and PDCs under the concurrence of both the IEEE C37.118 Standard and PJM's own requirements for installation and operation at required locations over their power system.

This thesis will describe in detail the operation, working and functionality of a Phasor Data Concentrator and its various types and characteristic requirements. It will also explain in the detail, the various procedures, techniques and tests which were developed and conducted by me here at Virginia Tech for the characterization of PDCs and their certification. It provides a complete overview of one of the PDC testing tools such as the PMU & PDC Simulator, one of the first of its kind to have been developed using the C# programming language by me at Quanta Technology LLC., explaining its features, importance and some of the difficulties faced during its execution, which although currently in its nascent stages, has the scope to be of the most important tool in the automated testing and certification of PDCs.

2. Phasor Data Concentrator (PDC)

2.1 Working

Data Concentration involves aligning all PMU data corresponding to one time tag into one data stream [5]. If all PMU data arrive at the PDC at the same time, at the same rate, with the same format, at the exact same time and without any missing data, the aggregation function may be relatively convenient.

However, this is almost never the case in actual systems. Time alignment involves aligning data arriving from different PMUs at different times, but having the same time-tag into one record. The time alignment process is further complicated when data from various PMUs with different reporting rates and different latencies (including both PMU calculation latencies and network communication latencies) need to be time aligned.

PDCs usually process the incoming PMU data in order to control, to a certain extent, the quality of the PMU data being reported by using validation processes such as checking the status of the transmitting PMU and its associated GPS clock, checking the CRC or other communication error checking processes, as well as more sophisticated data validation functions.

The process of time alignment usually involves buffering the incoming PMU data for certain amount of time and, after a certain waiting period, selecting all relevant data with the same time-tag as one time-aligned data set [7]. This buffering and waiting time may have to be limited to maximum values based on the limitations of the PDC (and the associated memory and processing power) and the requirements of the functions/application that use the output of the PDC. When all relevant data related to one time tag have arrived at the PDC or when the

maximum wait time is reached, the PDC will complete data validation, time alignment, and aggregation process as soon as possible and transmits the output data stream. This final processing time as well as the Wait Time should be short enough (while the buffering time should be long enough) to meet the needs of the functions/applications using the output data stream. In a system, the overall latency including latencies associated with the PMU functions, PDC functions, and the communication network have to be considered. [8]

2.2 Objectives

PDCs are mainly implemented to achieve the following objectives in a Synchrophasor interconnection system.

2.2.1 Communication Bandwidth Reduction The standard requires all PMU data to be transmitted over an IP based connection, which brings with it, certain amount of congestion [9]. With the help of a PDC, many Synchrophasor data streams can be aggregated and assimilated to become one single stream, thus reducing the network overhead significantly for a data stream. This becomes very helpful and rather necessary while sending data from substations to main control centers where there's a limited communication bandwidth.

2.2.2 Interface Simplification In a small system, it is possible for the control center to interface with each PMU but this task becomes highly complicated in applications where Synchrophasor data from hundreds and thousands of PMUs is to be used. PDCs reduce this problem by interfacing with the PMUs themselves first before sending the data to the control center. System Isolation Security: PDCs also act as a gateway between two networks. During the exchange of data between different entities, PDCs can facilitate network isolation so as to provide increased security among these networks.

2.3 Functions

This section describes some of the main functions of a PDC

2.3.1 Data re-sampling Although Synchrophasor streams are time-tagged at the same instant by the PMU, due to PMU processing delays & network latencies, the data packets will reach the destination at different times [10]. Hence, one of the important functions of a PDC is to align all the received PMU packets with the same time tag in correspondence with the desired output rate. If the data rate of some of the PMU streams is different from that of the others, the PDC's data re-sampling function should adjust the data rates to be the same as that of the PDC output rate [2]. Increasing the data rate of the PMU stream to the desired rate is called up-sampling, where as decreasing the data rate of the PMU stream to the desired rate is called down-sampling.

Care should be taken during down-sampling using proper filtering techniques so that no aliasing occurs. Similarly in up-sampling, proper interpolation methods need to be used so that no additional errors are introduced.

2.3.2 Data Validation and error-checking This may include checking the basic communication errors embedded in the IEEE C37.118 standard format [1] code, such as CRC Check and time quality check. Apart from such data validation mechanisms provided by the IEEE C38.118 standard [1], many advanced PDCs may also perform extensive data accuracy checks, such as implementation of a voting scheme in a system where several redundant Synchrophasor measurements are present [11]. PDCs can also make analysis of measurements at different locations at the power system and may use it to make an accurate approximation of any one measurement.

2.4 Types of PDCs

This section defines different types of PDCs to be referenced in this document.

2.4.1 Hardware and software PDCs

The phasor data concentration function of a PDC could be implemented to run on either specialized hardware platforms or on off-the-shelf hardware platforms. An "embedded" PDC is defined as a PDC that runs on a specialized, custom designed hardware platform. Such PDC implementation can optimize specific performance parameters because the interactions of the hardware and software components can be tightly controlled/optimized [5]. For example, real-time throughput can be optimized. An embedded PDC may also be used to achieve other objectives, such as operation in a harsh substation environment that require the PDC hardware to be designed for such environment. A software PDC (or stand alone PDC) is defined as a PDC that can be run on off-the-shelf hardware platforms. The software PDC typically runs on standard PC type hardware using a standard operating system such as Windows, Linux, or MAC OS.

2.4.2 PDCs used at different locations

PDCs may also be categorized based on the location where they are used, such as substation PDC, control center PDC, etc. A substation PDC is defined as a PDC that is used in a harsh substation environment. A substation PDC could be an embedded PDC or a software PDC. Embedded substation PDC's are usually based on typical substation IED (Intelligent Electronic Devices) platforms, such as relays and digital fault recorders. If software PDC is used, it is important that its off-the-shelf hardware platform must be able to withstand the harsh substation

environment. This involves compliance with a number of environmental / industry standards that other substations IEDs (Intelligent Electronic Devices) have to comply with. Usually substation "hardened" computers are used for hosting software PDCs in substations [11]. In general, it is recommended that any substation PDC to be compliant with all the environmental, safety, and EMC requirements and standards of typical substation devices such as relays. A control center PDC is defined as a solution that is used in a power grid control center environment. Control center PDCs are typically software PDCs, are designed to run on generic hardware platforms (and operating systems), and are more readily expandable. A high level of availability is usually achieved using multiple redundancy (or multi-host techniques). [5]

2.4.3 PDCs used in different levels of a hierarchical system

In a multi-tiered Synchrophasor system, PDCs may be used at different levels in the tiered system. An example could be an ISO Synchrophasor system that is the interconnection of all Synchrophasor systems of its transmission owners (TO). In such a system, the PMU data may be first concentrated at the substation level by substation PDCs, then at the TO's control center by TO control center PDCs, and finally at ISO control center by the ISO's own control center PDCs that at times may be called Super PDCs. [12]

2.5 Performance Requirements

This section specifies the performance requirements for main PDC functions.

2.5.1 PDC Processing Time

The total processing time required for a PDC to receive, validate, re-sample and time align all input data stream packets and repack and send the data to an output port for one data

record is defined as PDC data processing time for a real time output stream. PDC Data Processing Completion Time is the portion that occurs after last relevant data arrives at the PDC or the maximum wait time is reached, whichever is sooner [4]. The PDC manufacturer shall specify the maximum Processing Completion Time for received PMU and PDC data for each output data stream. The PDC shall make the time-aligned data ready for transmission within maximum Processing Completion Time.

2.5.2 Data Re-sampling

As specified earlier, the data re-sampling function of a PDC shall include up-sampling and down-sampling.

PDC's down-sampling function, as a minimum, should be able to down-sample a received data stream with a data rate specified in IEEE C37.118 Standard [1] to any lower data rate specified in that same standard. Similarly, a PDC's up-sampling function should be able to up-sample a received data stream with a data rate specified in the standard to any higher data rate specified in the same standard.

The PDC manufacturer shall specify and provide capability of up-sampling and down-sampling of the incoming data to time align data arriving at different rates and to make the output stream compatible with the downstream Synchrophasor devices as well as any application that may use these data.

2.6 PDC Communication Requirements

PDC shall be capable of supporting its communication needs through multiple physical ports and/or virtual channels. It shall be configured to use both TCP/IP and UDP/IP for its messaging,

data communications and data streaming operations mode configuration with each PMU or PDC. It shall support IPv4 and IPv6 and be capable of IP multicast for streaming data communication and input data from multiple data sources [9]. The primary data protocols which a PDC shall, as a minimum support, include IEEE C37.118-2005 and IEE 1344.

One of the primary functions of a PDC is to receive data from several PMUs and PDCs within the wait time as specified by the application function and transmit the accumulated and processed data as output. If the data has not arrived within that wait time, the PDC must aggregate what data it has and send it as output, without the delayed data [13]. It must also tag this data as missing in its outgoing data and correspondingly accept such tags if one of the inputs is from another PDC.

3. Virginia Tech PDC Testing Program

Phasor Data Concentrators (PDCs) are part of PJM's three year project to implement a wide area monitoring system. To guarantee correct operation of this system, the individual PDCs must undertake various types of tests to ensure its reliable and accurate operation. The application of PDCs for wide area measurement requires minimum delays, precise alignment and time tagging of the reported data, and consistent performance across all units in the PJM system. These requirements demand a set of functional performance tests to ensure that the performance of an installed PDC system can meet its desired objectives. The tests described in this report were developed according to the PJM requirements for minimum PDC performance and are partially based on the Guide for Phasor Data Concentrator Requirements for Power System Protection, Control and Monitoring [5]. The complete report of the first device tested is in Appendix A & B.

3.1 PDC Testing System

The operational diagram of the Virginia Tech National Instrument PDC Testing system developed to perform the tests listed in this report is shown in Figure 3.1. This simple set up consists of a base PMU which is connected to the DUT PDC, which in-turn is connected to a data analyzer or the NI System depending on the type of test to be performed. The number of PMUs in the base PMU can be changed depending on the requirement.

The primary goal of this set up is to observe if the DUT PDC is successfully able to process and align the data coming from the base PMU and output the time-aligned data to the user (in this case, the data analyzer) without any errors.

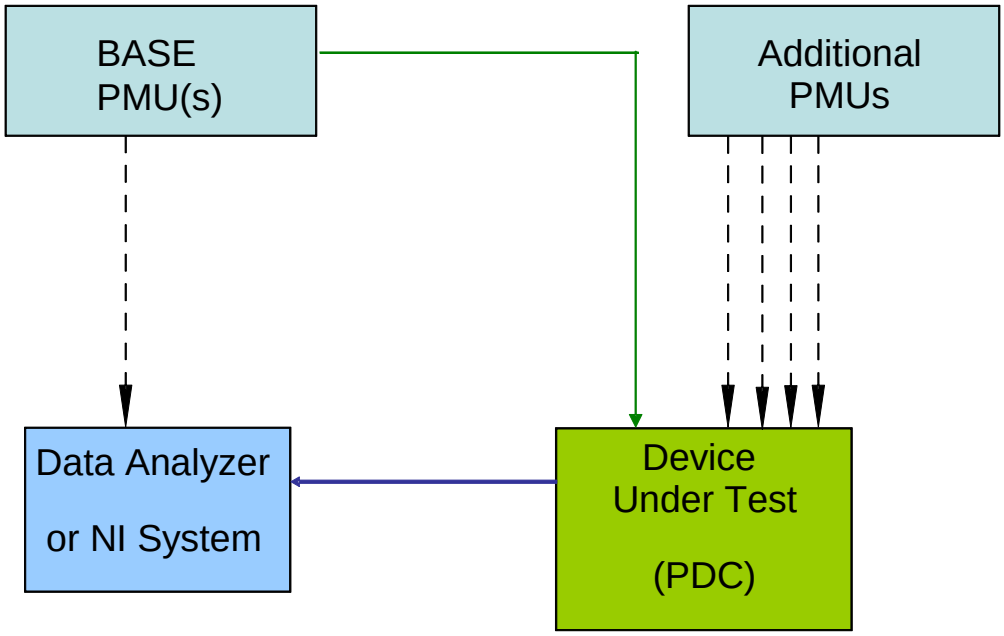

Figure 3.1 Operational Diagram of PDC Testing System

3.2 Virginia Tech PDC Test Procedures

3.2.1 Procedure 1: DUT Unpacking

This section describes the PDC unpacking procedures for all PDCs sent by manufacturers for testing as part of the PJM testing project. This procedure was developed to maintain a certain level of procedural uniformity for every DUT being tested. The tests described in this section are general and assume a simple package per device. For multiple packages, the procedure must be repeated for each package.

3.2.1.1 Prepare a DUT Folder

Before unpacking DUT prepare a folder for the DUT to keep an official record of the tests performed and their dates and results. All received packages contain a packing list attached to the package or inside the box. Collect packing list and attach it to the DUT folder. Use Packing list to create a folder ID page (see last page of procedure records). Record telephone number or email contact from the manufacturer.

3.2.1.2 Package Unpacking

3.2.1.2.1 Inspect Package for Damage

Any damage on the received boxes should be recorded and damage to the equipment should be reported to the manufacturer. Before opening the box inspect it for any sign of external damage. Describe the location and characteristics of any observed damage.

3.2.1.2.2 Device Unpacking

Open package and compare box content with packing list. List any missing or additional component in the package. Contact manufacturer and inform them of any missing item. The following items must be part of the package: DUT, Operational Manual, Interface Software, power cord, and dedicated connectors (if required).

Inspect DUT for any visible damage. Contact manufacturer if any damage is detected.

3.2.2 Procedure 2: DUT Wiring and Communication

This section describes the PDC wiring and communication procedure. Steps described in this section are general and require the use of the manufacturers user and/or installation manual(s).

3.2.2.1 Install DUT on Test Rack

Review user or installation manual procedures for device mounting or software installation. Using the manufacturer's instructions, if possible, mount the DUT hardware above the test terminal board and fix tightly to the rack. If mounting is not possible position the device securely on the test bench. If software installation is required follow the instructions provided by the manufacturer.

3.2.2.2 Synchronizing Signal Wiring

If synchronization is provided to an internal or external GPS source use manufacturer's manual to wire the synchronizing signal and/or install GPS antenna.

3.2.2.3 Communication Wiring

Follow manufacturer's instructions to connect a serial or Ethernet cable between the DUT and the lab's base PMU. If available install the DUT interface software in the lab's interface computer.

Follow manufacturer's instructions to connect a serial or Ethernet cable between the DUT and the NI PMU.

3.2.2.4 DUT Configuration

If the device has been configured by the manufacturer skip this step limit configuration to the fix number of communication channels. Power DUT and follow manufacturer's instruction to establish communication with the lab's base PMU. From the software user interface make the following settings:

- Set message format to C37.118
- Set message rate to 30 frames/sec
- Set PDC ID No.
- Enable Synchrophasor broadcast from base PMU.
- Enter IP Address of base PMU to DUT
- Enable data collection on the PDC
- Save or record settings before collecting DUTR data

3.2.2.5 Data Availability Verification

Compare the DUT data with the data of the base PMU data stream verify compliance with C37.118 format.

3.2.3 Procedure 3: Operational Tests

This section described PDC operational test to be performed in all units to determine correct operation. The tests described in this section are limited to PDC specific operations for all other test the manufacturer recommended test must be performed prior to the tests described in this section.

3.2.3.1 Manufacturer recommended Test

All pre-energization inspection, basic functionality and operational test recommended by the manufacturer must be performed and passed before the test described in this section to be performed.

3.2.3.2 GPS Clock Tests

If available time synchronization to the microsecond is specific to PDCs and it is required for its correct operation. In terms of their synchronizing signals, PDCs are classified in Internal GPS Clock and External GPS clock units. The described GPS antenna signal must be performed on the PDC or the external GPS clock. The IRIG-B input signal test is specific to the PDCs without an internal GPS clock.

3.2.3.3 GPS antenna signal

GPS clocks require connection to an antenna to acquire the signals from the GPS satellites. Before performing the test set the antenna in a location with a clear view of the horizon. Follow the manufacturer's recommendations for the cable length and type. Use of a low attenuation cable can extend the usable cable length as can an antenna signal amplifier.

3.2.3.3.1 Methodology

After installing the antenna in the PDC or external GPS clock allow 30 minutes for the GPS receiver to acquire signal, download almanac data and fix its location. If available use the instruction provided by the manufacturer to monitor the status, signal level and number of satellites track by the GPS clock.

3.2.3.3.2 Test Devices

Use the GPS clock interface, PDC interface, or provided software to obtain the GPS clock status.

3.2.3.3.3 Test Criteria

If the unit fails to acquire signals after 5 minutes or if it fails to lock its time signal after 30 minutes the unit fails the test.

3.2.3.4 IRIG-B Input Signals

PDCs that do not have a built-in GPS clock may and external synchronization signal

3.2.3.4.1 Methodology

Perform and pass the GPS antenna signal test on the external GPS clock before connecting the external synchronizing signal. Follow the PDC and GPS clock manufacturer instructions for the installation of the synchronizing signal. Wait five minutes for the PDC to indicate that its time has locked with the external GPS clock. Once lock verified that the PDC time agrees with the GPC clock time.

3.2.3.4.2 Test Devices

Use the GPS clock interface or provided software to obtain the GPS clock status. Use the PDC display or provided software to verify the locked status of the timing signal.

3.2.3.4.3 Test Criteria

If the PDC fails to lock its time signal after 5 minutes the unit fails the test.

3.2.3.5 Time Quality Indicator Flag Test

The IEEE C37.118-2005 Standard required that the 32-bit (4-byte) FRACSEC be divided into two components: a 24-bit integer that is the actual fraction of second count and an 8-bit Time

Quality flag. The Time Quality indicator code is contained in the lowest 4 bits and indicates the maximum time error as determined by the PMU clock function. Bits 0-3 shall be all cleared to 0 when the time function is locked to the GPS. Bits 0-3 shall be all set to 1 when there is either a clock error or the clock has never been initially set. Conditions of accuracy between these extremes are defined in Table 8 of the Standard.

3.2.3.5.1 Methodology

Use the NI-VIs to monitor bits 0 to 3 of the FRACSEC flag to determine the status of the time quality reporting functionality of the PMU. of the PMUGPS clock [15]. Verify operation of the communication channels and enable on-line data on the PMU. Allow the DUT to lock to the GPS signal. Verify that the bits 0 to 3 are zero when the unit is locked. With the unit locked removed the GPS signal and monitor bits 0 to 3 of FRACSEC to complied with the quality definitions on Table 8 of the standard. Re-insert the GPS antenna and verify that the DUT resets the quality bits when the unit locks to the GPS.

3.2.3.5.2 Test Equipment

This test requires that the DUT receives data from the VT base PMU and that the operator verifies the values of the time quality bits when the GPS signal is lost and recovered.

3.2.3.5.3 Test Criteria

If the time quality bits do not change in accordance with Table 8 of the Standard the test is failed. Record comments for partial operation of the time quality bits in order to make the end-user of the report be cognizant of the error/problem and act on it accordingly. In many cases, the partial operation of the time quality bits may be within the user's criteria of requirements.

3.3 Virginia Tech Test Program Report

3.3.1 Introduction

The Virginia Tech Test program consists of three main groups of tests: PDC unpacking and functionality pre-test, PDC general operation tests and PDC conformance tests.

3.3.2 Unpacking and functionality pre-test

The PDC unpacking and functionality tests are performed when the unit is received at Virginia Tech. The main objectives of these tests are to:

- Determine if there is any damage in the unit due to transportation
- Confirm that all the required software and hardware interfaces to connect with a computer are received with the unit
- Verify simple operation of the device

3.3.3 PDC general operation tests

The PDC general operation tests are used to establish communication between the PDC and a computer system and include frames check, flags check, and reporting rate check. The frames check is performed to ensure that the format of the data, configuration, command, and header frames conform to the IEEE C37.118-2005 Standard specifications. The flags check validates the operation of the timing quality flag in accordance with the timing source. The reporting rate check performs a large step frequency variation test for 30 and 60 frames per second (fps) to verify that the PDC can produce phasor data streams at the reporting rates required by PJM.

3.3.4 PDC Application test

The PDC Application test includes the PDC Latency test. This test is not performed for compliance with the PJM PMU requirements but in fact only as parameter to measure the performance of the system. The results of the tests are display in terms of the delay in seconds as defined in the Guide for PDC Requirements.

3.3.5 General PASS/FAIL criteria description

For pre-test, the aim is to determine if the DUT is ready to be tested. If the PDC is received without any apparent damage, operates normally for basic operations, and connects to the test system; it is considered that it has passed the pre-test and it is ready for the full test program.

For general operation tests and conformance test, if the criterion has been defined by the IEEE C37.118-2005 Standard, the PASS/FAIL criteria will follow the Standard. For tests not defined by the Standard and required by PJM there is no PASS/FAIL criterion but comments are provided on the relative performance of the equipment.

3.3.6 General Tests

The general tests include message frames check, flags check, and reporting rates check. All general tests are steady-state tests that are performed with balanced three-phase voltage and current inputs to the DUT using the NI PMU testing system. The nominal magnitudes for the SEL 421 are 70 volts and 5 amps rms. Message frames and flags are checked at the two PJM required reporting rates of 30 and 60 fps.

3.3.6.1 Message frames check

3.3.6.1.1 Test description

The following message frames are checked for IEEE C37.118-2005 [1] compliance:

- Configuration-1 (CFG-1), Configuration-2 (CFG-2)
- Data
- Header

The following commands are sent to the DUT during these tests:

- Turn off transmission of data frames,
- Turn on transmission of data frames,
- Send HDR file,
- Send CFG-1 file, and
- Send CFG-2 file.

The tests are conducted with voltage and current magnitudes at 100% nominal, frequency at 60 Hz, and harmonic and out-of-band injection within their limits (< 0.2%).

Message frames must fully conform to data frame, configuration frames, and header frame of IEEE C37.118-2005 Standard to be considered compliant.

3.3.6.1.2 Test configuration

This test is conducted using the test setup shown in Figure 3.1. The message frames from the DUT (PDC) are captured on the same NI Test system and checked to see if they conform to the

IEEE C37.118-2005 Standard. A single stream of Synchrophasor data is captured and stored for analysis and reporting using the NI Test System as shown in figures 3.2 & 3.3.

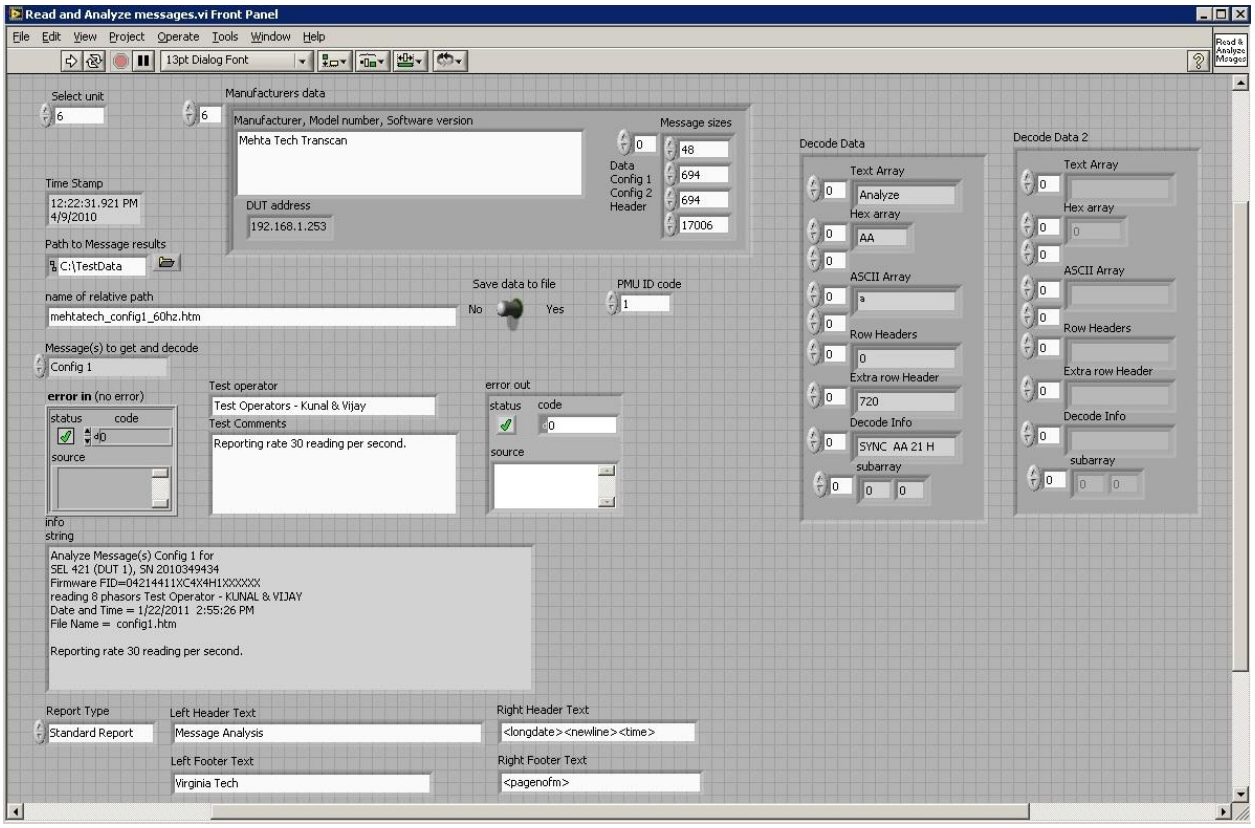

Figure 3.2 NI LabView layout for capturing data frames

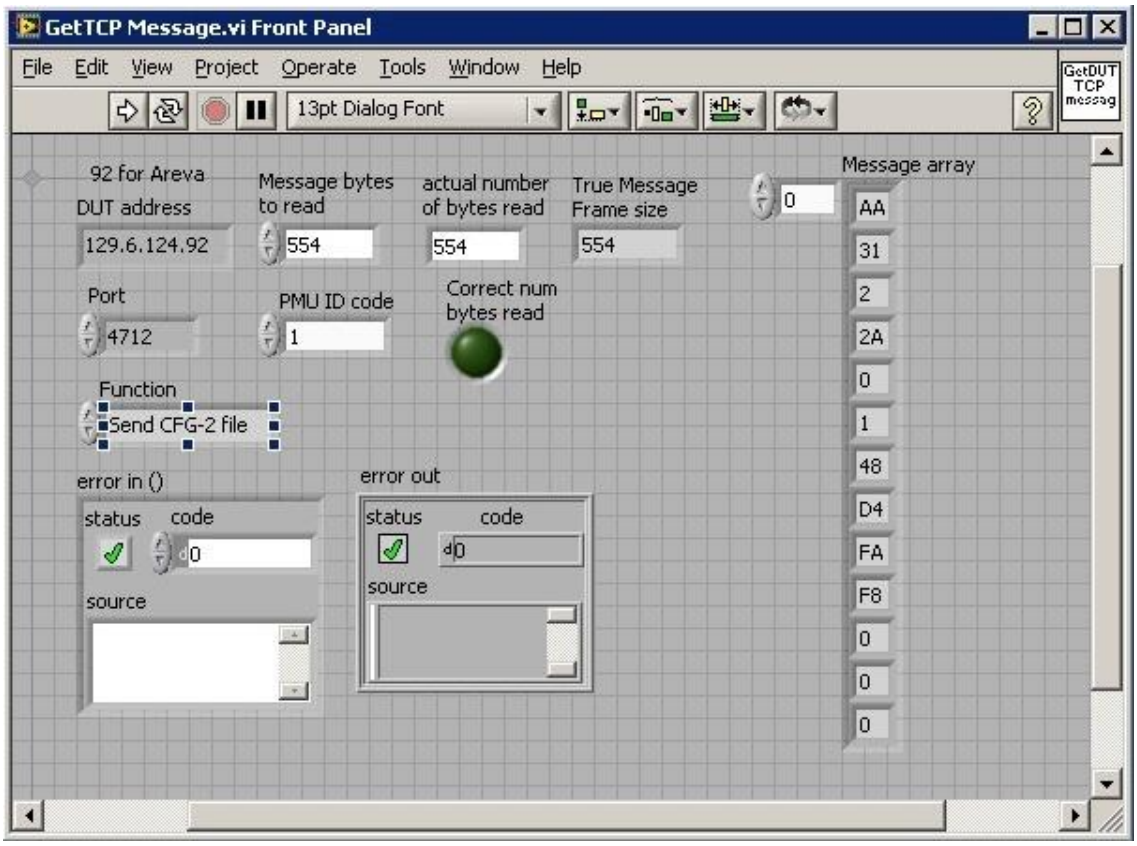

Figure 3.3 NI LabView layouts to determine frame size

3.3.6.1.3 Results

All frame messages were confirmed to comply with IEEE C37.118-2005 Standard's definition.

3.3.6.1.4 Conclusion

Based on the test results for message frames check tests, the SEL – 3373 PDC **PASSED** the Message Frame check test.

3.3.6.2 Flags Check

3.3.6.2.1 Test description

This test checks the DUT for proper reporting, by the time quality flags, of the loss of UTC synchronization. The GPS clock's antenna is purposely removed, and the timing quality flags of the DUT are checked to see if they change appropriately.

The tests are conducted with voltage and current magnitudes at 100% nominal, frequency at 60 Hz, and harmonic and out-of-band injection within their limits (< 0.2%).

3.3.6.2.2 Test configuration

Up to 16 GPS synchronized PMU connections present in the Power Lab at Virginia Tech with streaming Synchrophasor data are sent to the SEL 3373 PDC for processing.

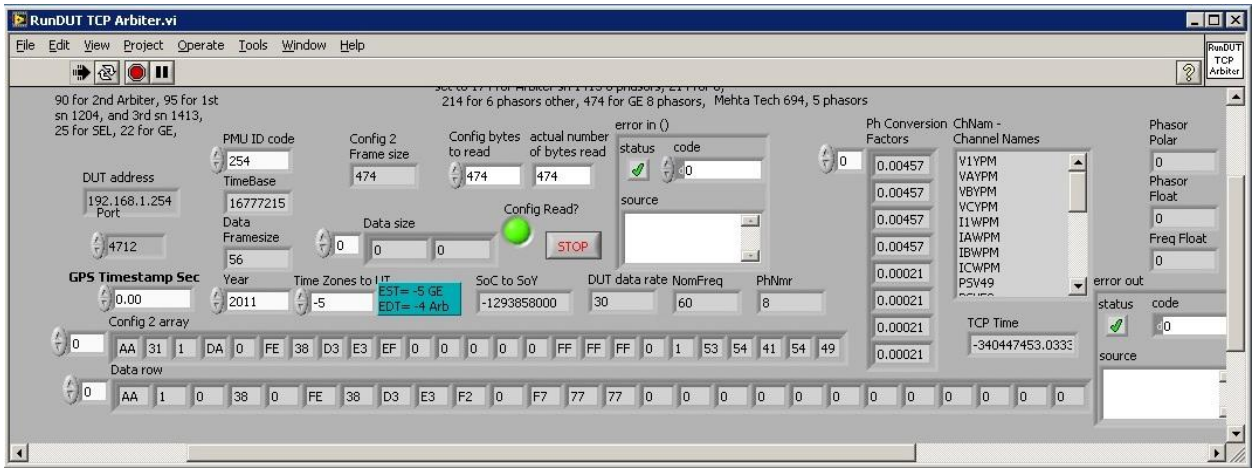

Figure 3.4 NI LabView layout to observe changes in data frame

The SEL – 3373 PDC uses an external GPS clock through an IRIG-B connection for synchronization. For this test and all tests performed in the SEL-3373 an IRIG-B signal from an Arbiter 1094B GPS Substation Clock is used for synchronization.

3.3.6.2.3 Results

The SEL–3373 unit was tested for GPS antenna-disconnect at a reporting rate of 30 frames per second after the unit reported synchronization to the GPS signal. When the antenna was removed the message frame from the DUT displayed in figure 3.4 showed **no change** what so ever, indicating that the GPS (IRIG-B) input signal to the DUT was quite redundant, but when the PMU which was connected to the DUT PDC was disconnected from IRIG-B a 1 bit flag check was observed in the status bits and time quality bits. The DUT set the unlocked time bits (bits 4 and 5) in agreement with Table 8 of the C37.118-2005 Standard for the 10 seconds, 100 seconds and 1000 seconds after antenna disconnection.

3.3.6.2.4 Conclusion

Based on the Test results for the time quality flag check the SEL – 3373 PDC **FAILED** the Flags Check test**.**

3.3.6.3 Reporting Rate Check

3.3.6.3.1 Test description

This test checks the DUT performance at reporting rates of 30 and 60 fps as part of the Steady State Tests. DUT data is expected at the correct rates and data format. Incorrect Rates and/or data format results in large errors in the application of the system.

3.3.6.3.2 Test configuration

The three-phase voltage and the three-phase current input channels of a PMU are energized which in turn is connected to the DUT PDC. The streaming Synchrophasor data of the DUT PDC is captured and stored for analysis.

3.3.6.3.3 Results

No specific test results are shown for this test. The NI PMU Test system collected data at 30 and 60 frames per second. Data was received at the correct rates with the correct data format.

3.3.6.3.4 Conclusion

Based on the Test results for the time quality flag check the SEL-3373 PDC **PASSED** the Reporting Rate test.

3.3.6.4 General Tests Summary Results

The results for the General Tests of the SEL-421 unit are summarized in Table 3.1. The SEL-421 PMU passed all the General Tests.

Table 3.1 General Test Summary

General Test	Test Result
Configuration-1 Frame	Passed
Configuration-2 Frame	Passed
Data Frame Request	Passed
Header Frame Request	Passed
Loss of Sync Flag	FAILED*
Unlocked Time Bits	FAILED*
30 fps Data Frame	Passed
60 fps Data Frame	Passed

*See comments in 3.3.6.2.3

3.3.7 Application Test

3.3.7.1 Introduction

The Application test includes the Latency Test. One of the primary tools used in calculating the latency of the system is the Wireshark network analysis software [16]. With the help of this software, two different data streams, one from a base PMU and the other from the DUT PDC can be systematically captured and stored for analysis. The test set up is as shown in the figure 3.1.

3.3.7.2 Test Configuration

As you can see in figure 3.1, the base PMU sends two identical data streams, one to the DUT PDC and the other to the computer containing the network analyzing software called Wireshark. The PDC after processing and aligning the data from the PMU sends an output stream to the same computer as the base PMU.

The Wireshark software has been programmed to capture and store the data from the streams as they arrive and notes the exact time at which each data was captured as shown in figure B.4. The important parameter here is the time delay between two data frames. With the help of this set up, we can easily calculate the time difference in obtaining the data from the base PMU and the DUT PDC. This time difference in milliseconds is nothing but the latency of the DUT PDC because of which its data frame will arrive slightly later than that of the base PMU's data frame [17]. Since both the base PMU and the DUT PDC are part of the same local area network, the network delays get cancelled out. This way, the latency of the PDC only is tested. The network latency does not play any part in this experiment and is not tested or accounted for.

The Wireshark software is installed on a Microsoft Windows XP operating system. Now, from the technical specifications of Microsoft Windows the maximum accuracy which an application can obtain on this operating system through the system clock is between the range of 0.7 to 1 millisecond. With the resources available to us, this accuracy is adequate to measure up to a sufficient level of certainty the time delay between the arrivals of the two data frames. For a higher level of accuracy, Linux operating system can be used. By hard coding the source code of the system clock in Linux to be synchronized with an external GPS receiver, time stamp

accuracy similar to that of a PMU can be obtained but due to limited resources and budget constraints, such options could not explored.

As shown in the figure below (fig 3.5), the Base PMU is represented by the IP Address '192.168.1.254' and the DUT PDC is represented by the IP Address '192.168.1.2'. With the help of a pre-existing filter available in Wireshark called 'synphasor', all the other network communication packets are filtered out and only the Synchrophasor data can be observed.

Figure 3.6 shows a screenshot of the operating console of the DUT PDC SEL 3373. As you can see, a total of 16 different PMU connections have been made. The PMU with the IP Address '192.168.1.254' which is DUT Station 1 is used as the base PMU to observe the latency of the DUT when a PMU connection is sequentially removed.

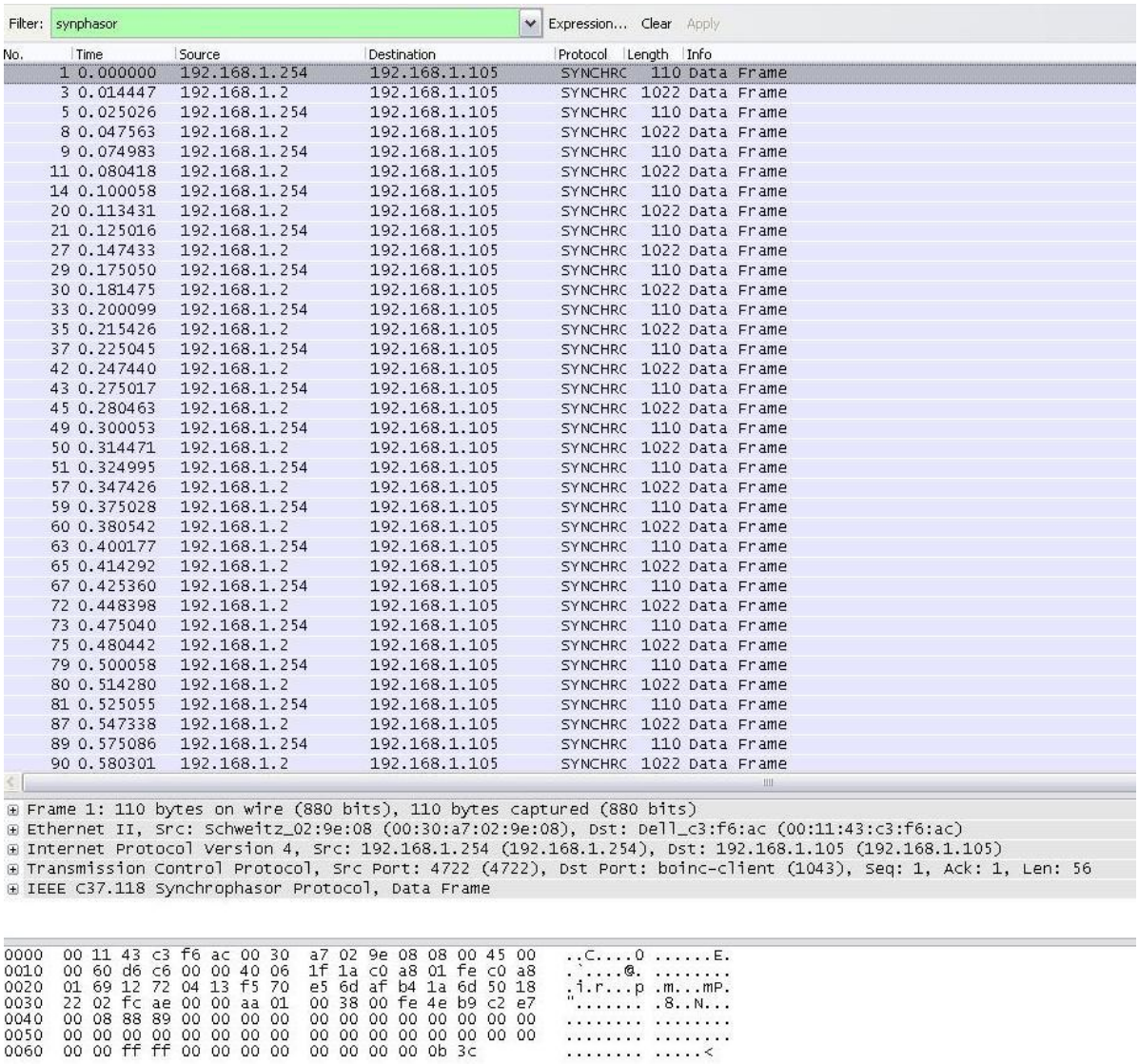

Figure 3.5 PDC Testing Set-up on Wireshark Network Analysis Software

Input Connections

Name	PDC ID	Connection State	Time Quality	Received Data Frames
VCDAU-1	119	Receiving Data	Within 10^-2 s	2512
Station A	118	Receiving Data	Normal	2512
Station A'	118	Receiving Data	Normal	2512
VT Panel 1	1	Receiving Data	Normal	2512
VT Panel 1'	1	Receiving Data	Normal	2512
VT Panel 2	2	Receiving Data	Normal	2512
VT Panel 2'	2	Receiving Data	Normal	2512
VT Panel 3	3	Receiving Data	Normal	2512
VT Panel 3'	3	Receiving Data	Normal	2512
VT Panel 4	4	Receiving Data	Normal	2512
VT Panel 4'	4	Receiving Data	Normal	2512
GE-UR PMU	25	Receiving Data	Normal	2512
1133 PMU-1	1	Receiving Data	Normal	2512
IDM-E T3-T5	500	Receiving Data	Normal	2333
DUT Station 1	254	Receiving Data	Normal	2512
PJM Interconnect	253	Receiving Data	Normal	2509

Figure 3.6 Operating Console of the DUT

3.3.7.3 Results

The test results are shown in Figure 3.7 are for 5000 message frames captured from both base PMU and DUT PDC. The average latency was calculated to be **0.857 msec**.

The Figure 3.7 below shows the first of four different tests performed on the DUT PDC to observe its latency. In this test, only one PMU connection was made to the DUT PDC.

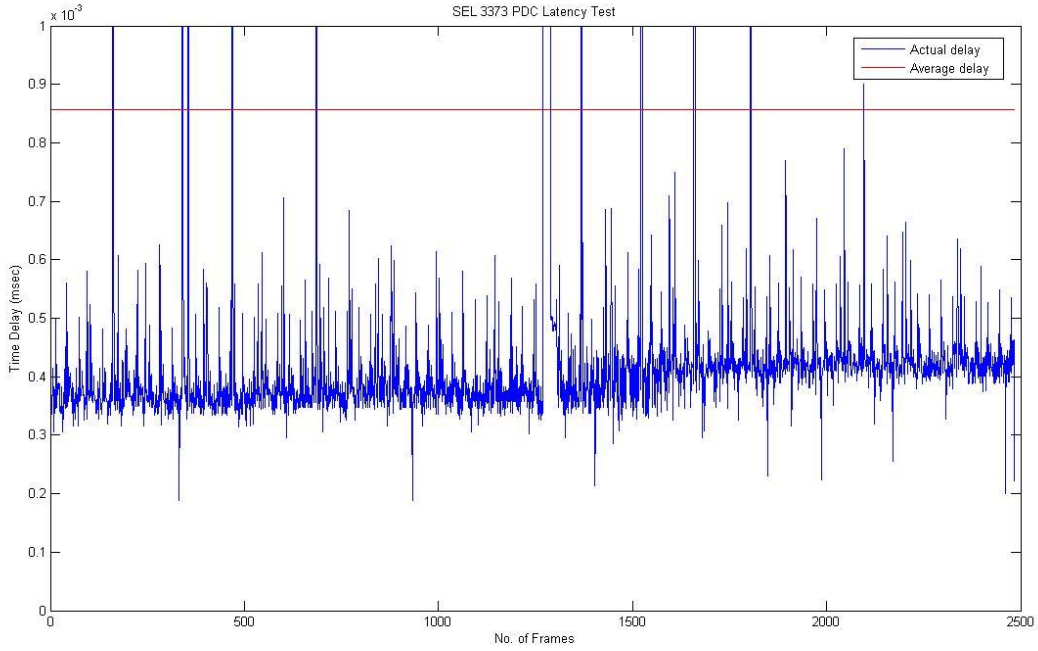

Figure 3.7 Latency Test 1: Plot 1 PMU

In the subsequent three tests, up to 16 PMU connections were made depending on their availability, and systematic plot of the delay between the base PMU and the DUT PDC was observed after ever PMU connection was sequentially removed.

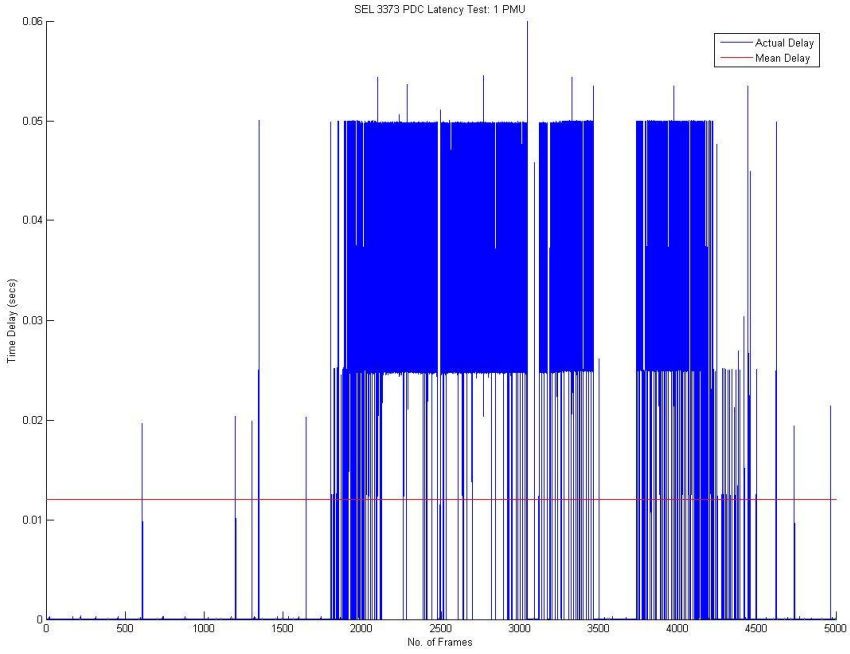

Figure 3.8 Latency Test 2: Plot 1 PMU

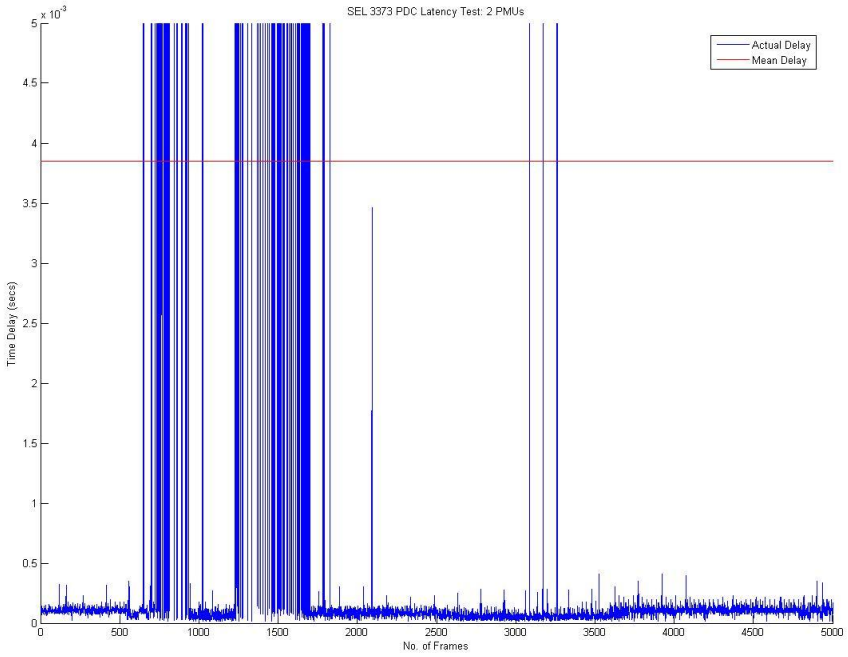

Figure 3.9 Latency Test 2: Plot 2 PMU

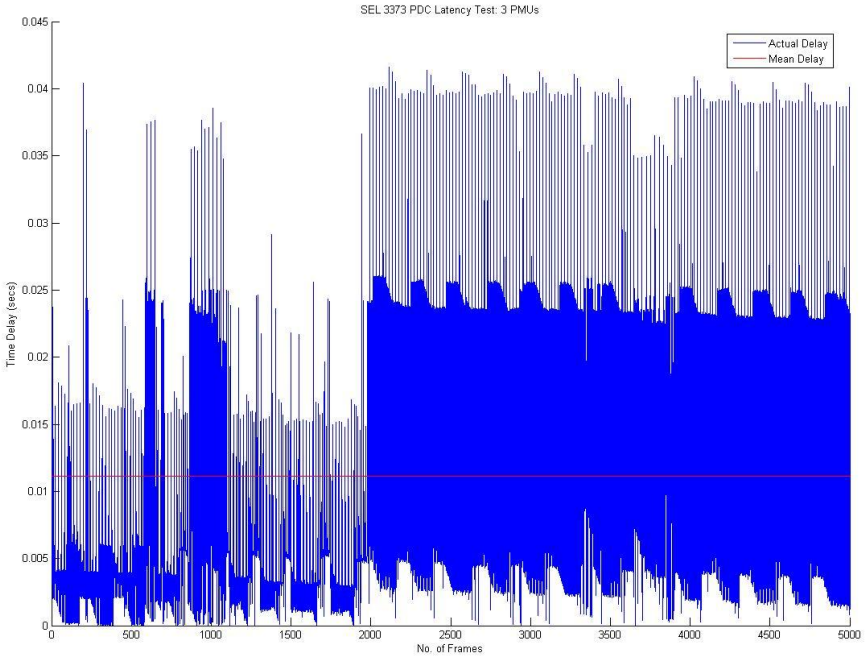

Figure 3.10 Latency Test 2: Plot 3 PMU

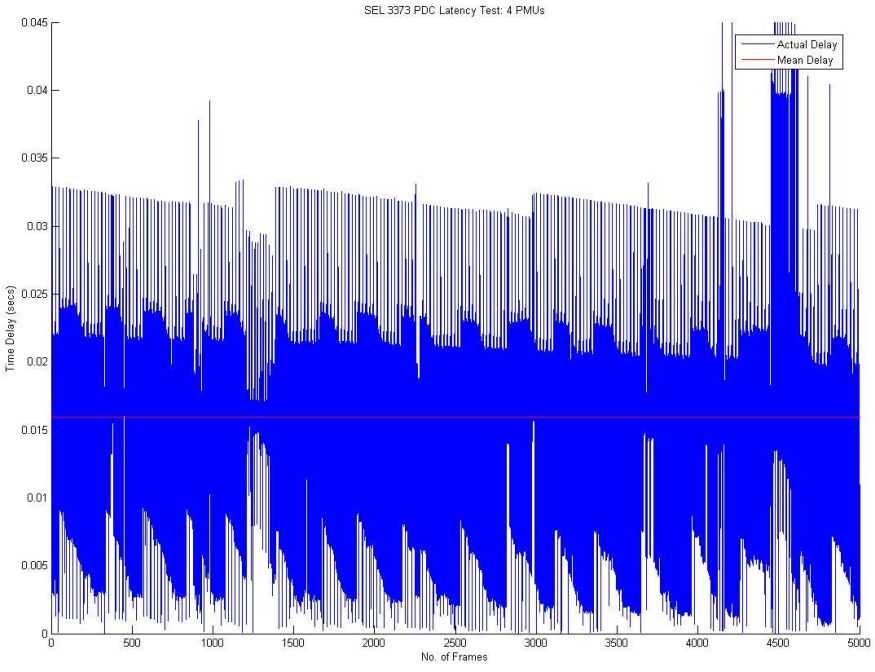

Figure 3.11 Latency Test 2: Plot 4 PMU

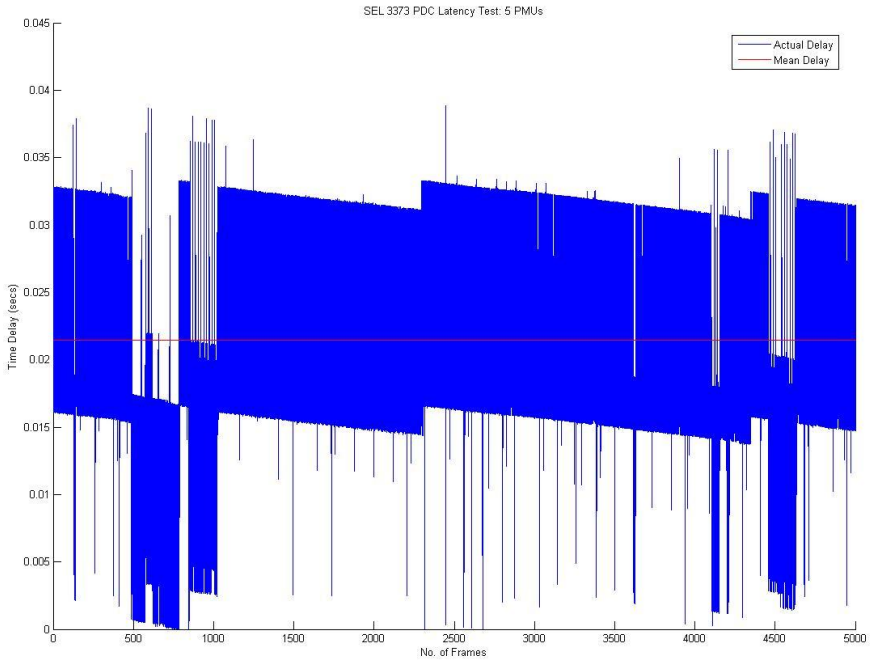

Figure 3.12 Latency Test 2: Plot 5 PMU

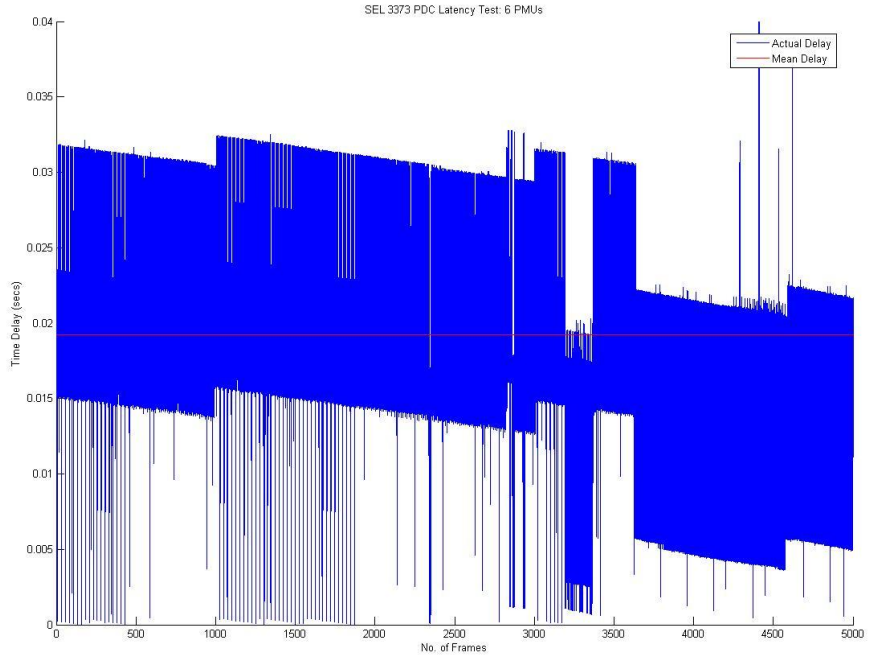

Figure 3.13 Latency Test 2: Plot 6 PMU

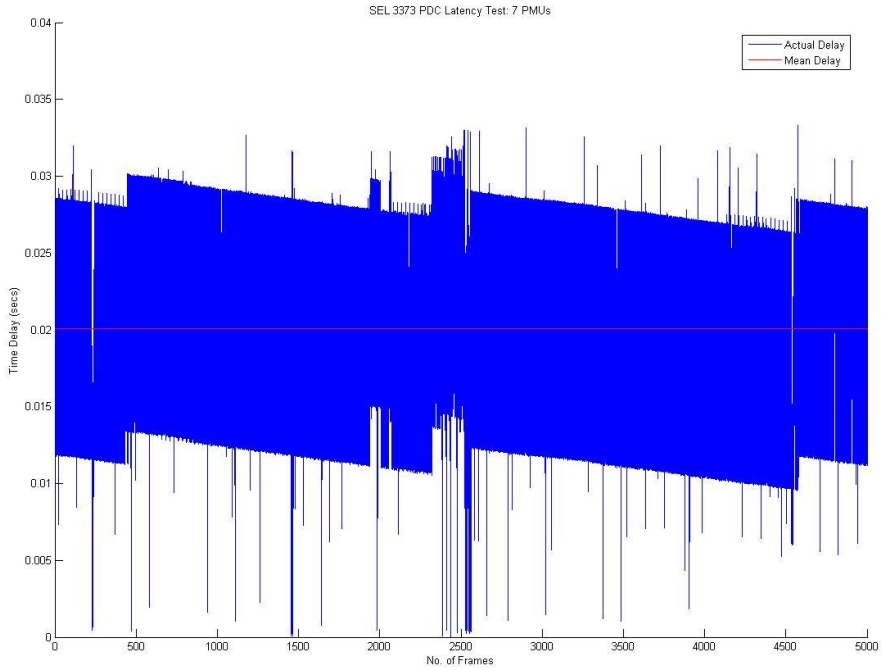

Figure 3.14 Latency Test 2: Plot 7 PMU

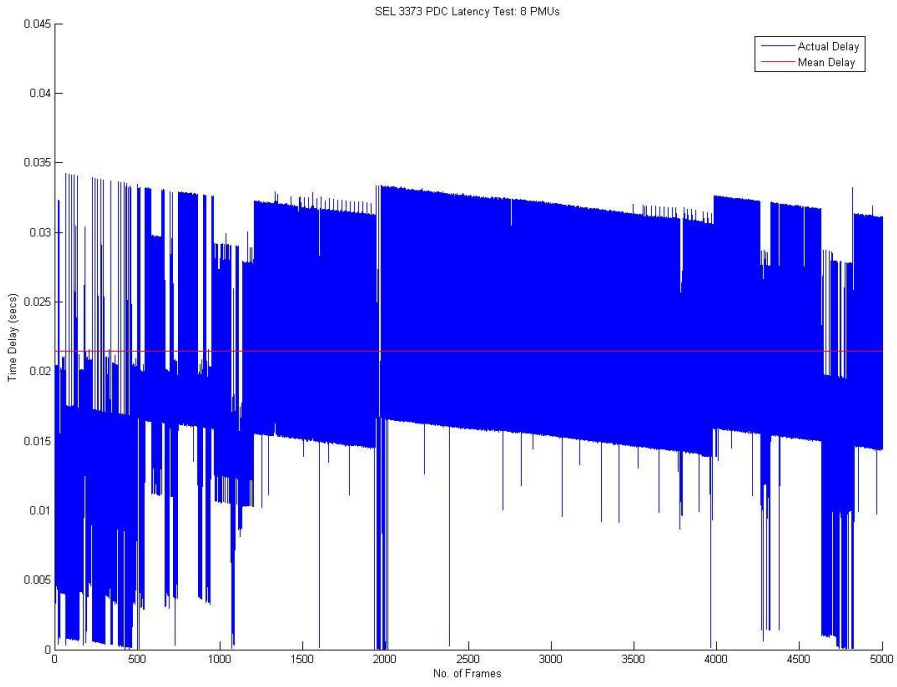

Figure 3.15 Latency Test 2: Plot 8 PMU

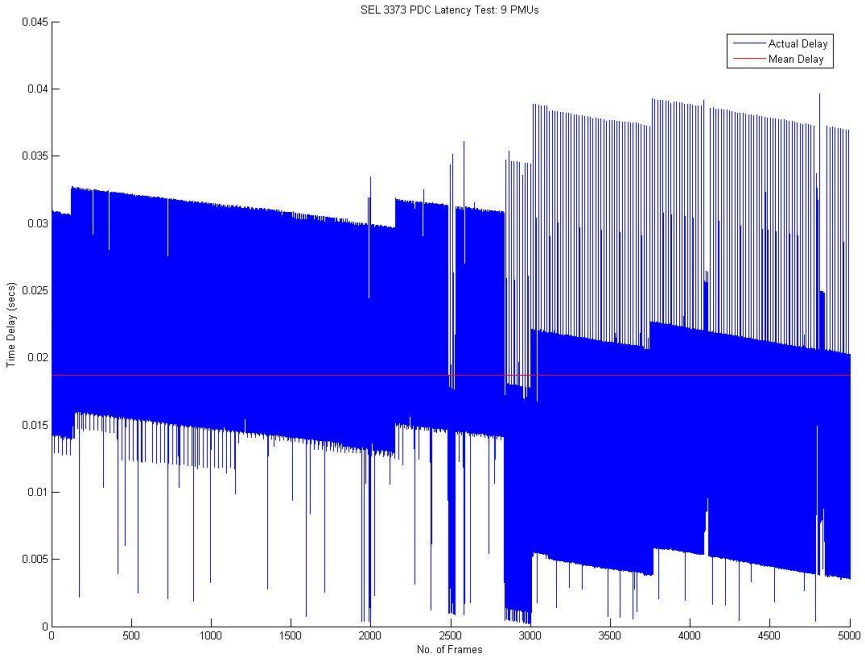

Figure 3.16 Latency Test 2: Plot 9 PMU

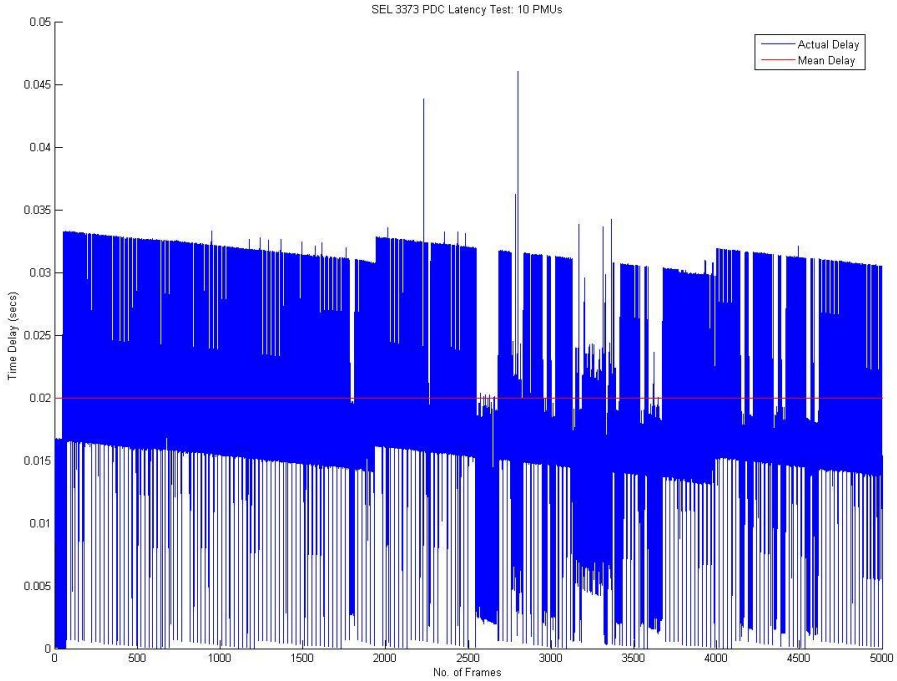

Figure 3.17 Latency Test 2: Plot 10 PMU

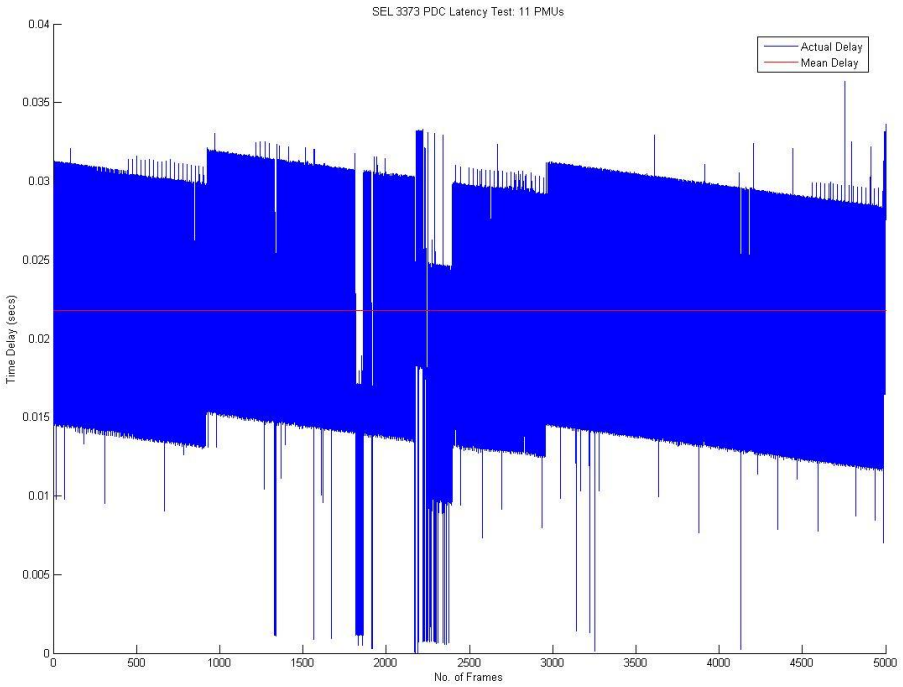

Figure 3.18 Latency Test 2: Plot 11 PMU

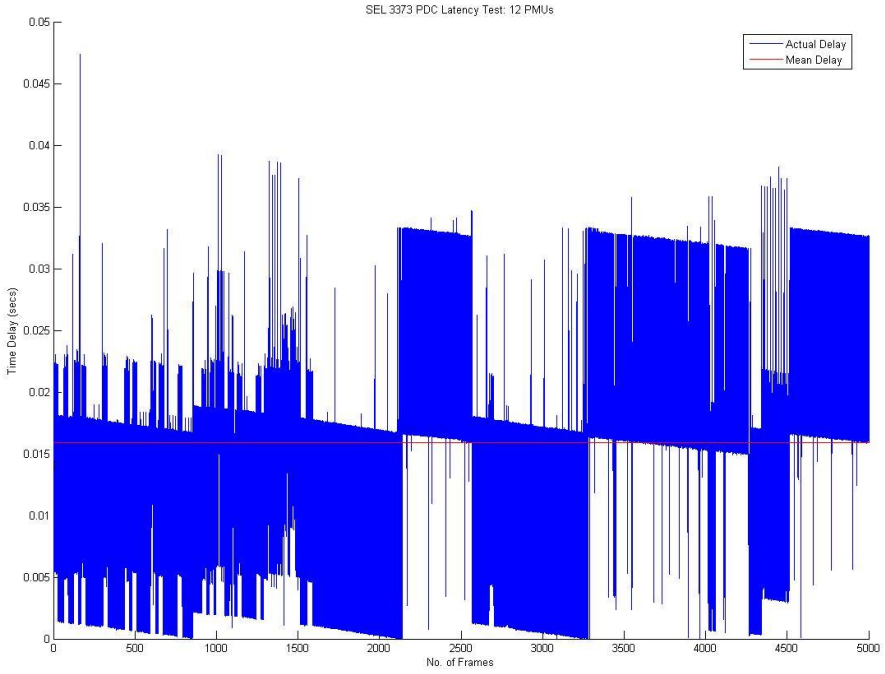

Figure 3.19 Latency Test 2: Plot 12 PMU

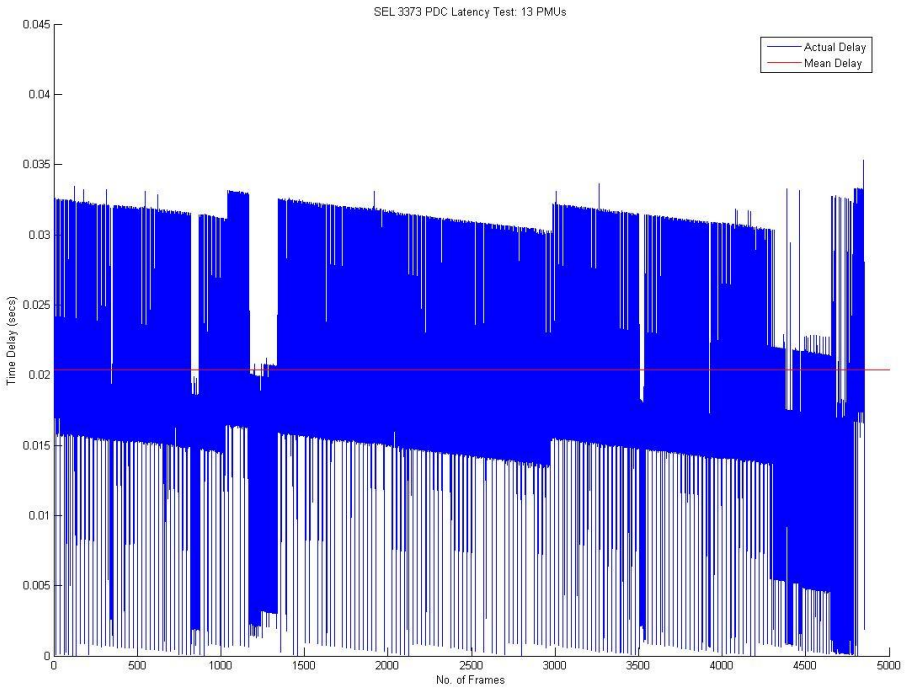

Figure 3.20 Latency Test 2: Plot 13 PMU

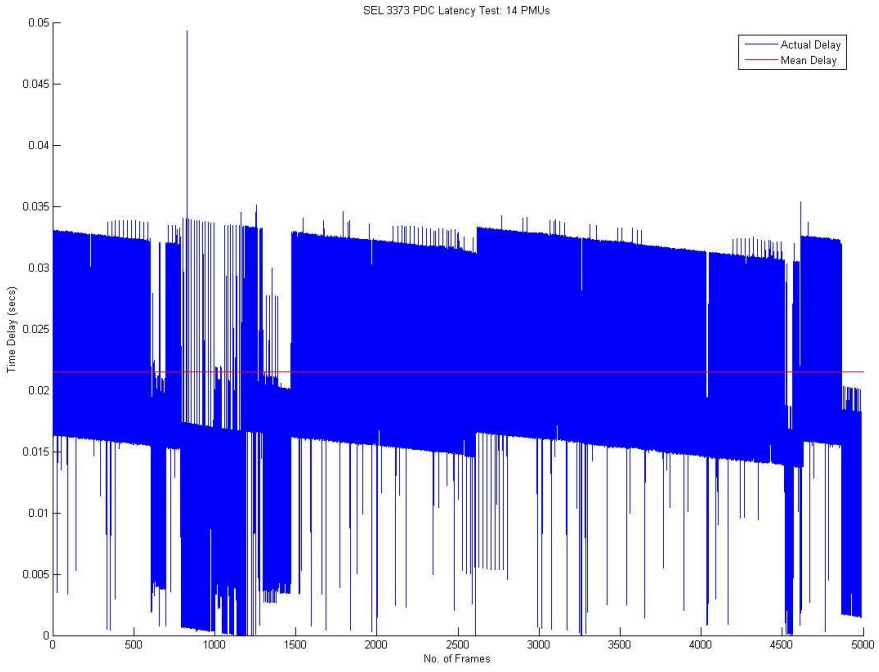

Figure 3.21 Latency Test 2: Plot 14 PMU

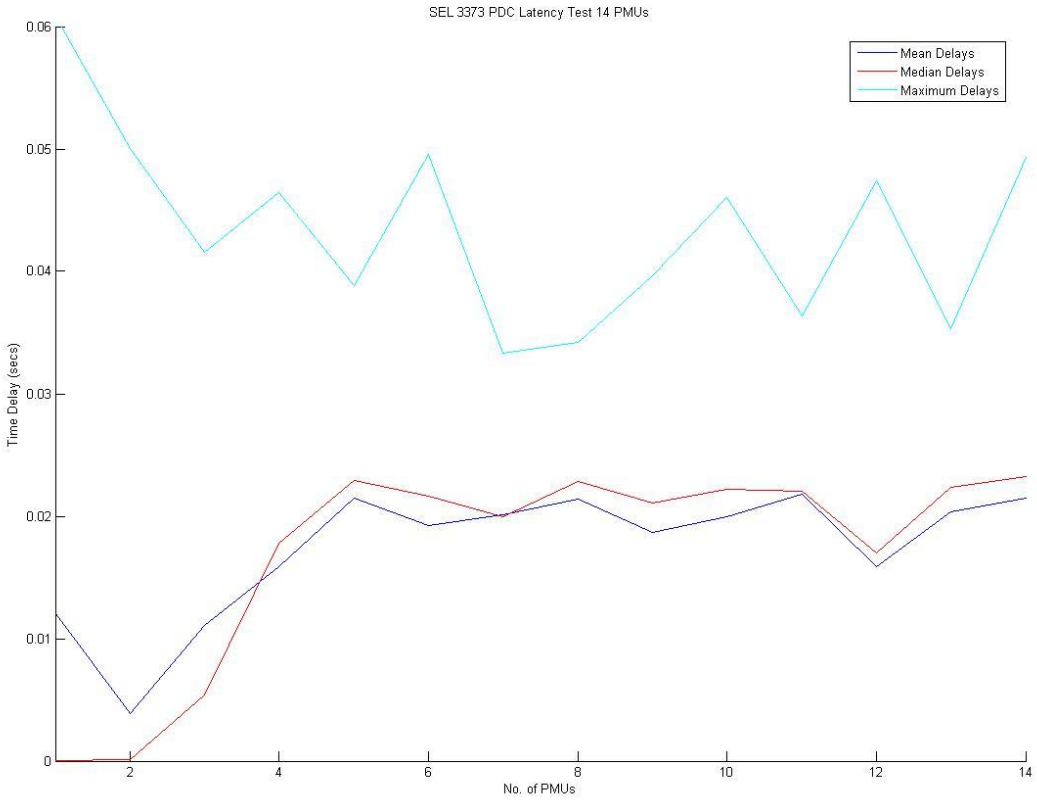

Figure 3.22 Latency Test 2: Cumulative Plot 14 PMUs

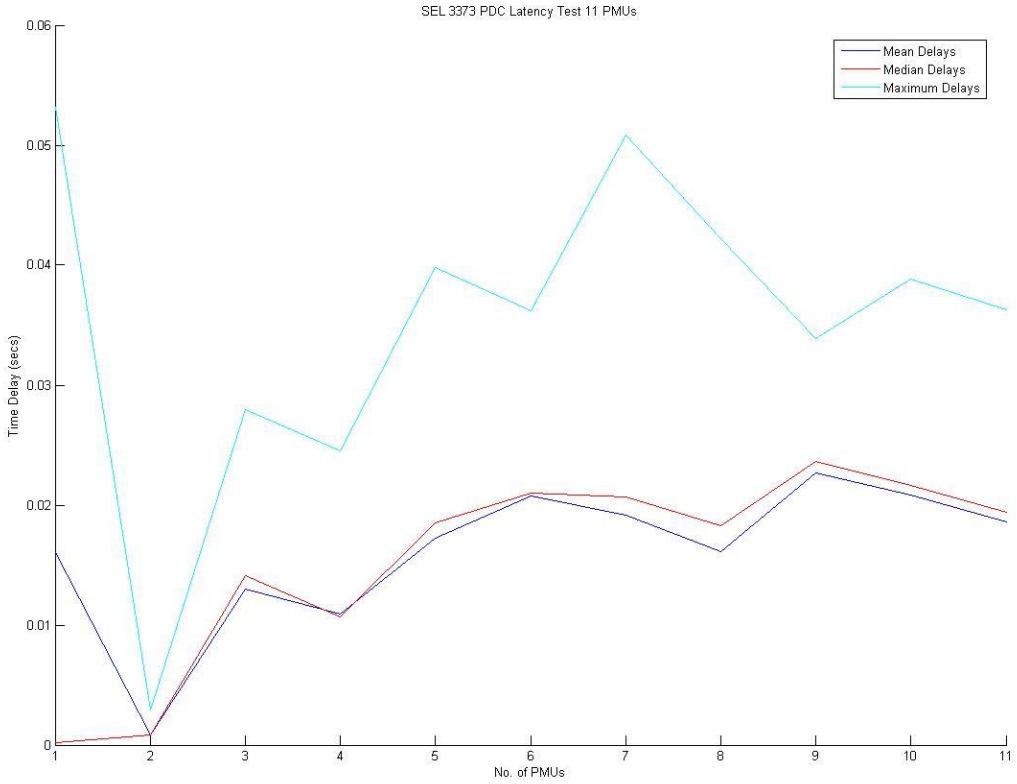

Figure 3.23 Latency Test 3: Cumulative Plot 11 PMUs

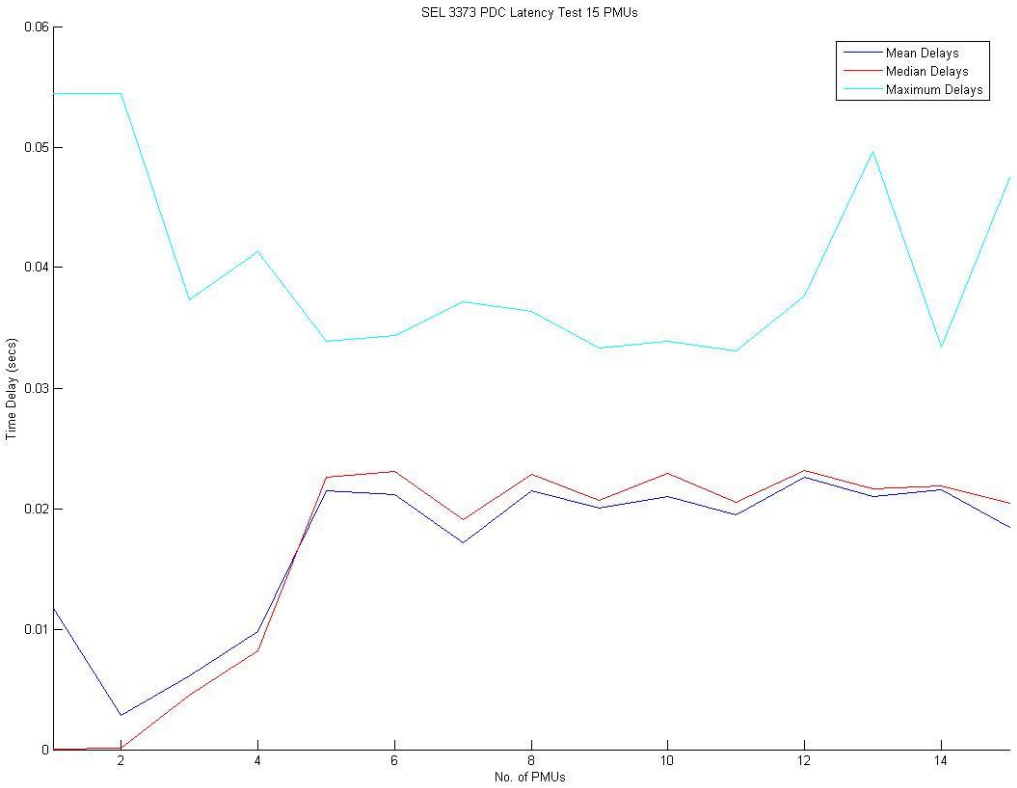

Figure 3.24 Latency Test 2: Cumulative Plot 15 PMUs

3.3.7.4 Conclusion

The first of the 4 tests performed on SEL 3373 PDC shown in figure -- was to observe its base latency with just 1 PMU connection. Figure 3.22, 3.23 and 3.24 show the response of the DUT PDC as the number of PMU connections to it is systematically increased.

One of the important plots in the above three figures is depicting the median of delay values for each no. of PMU connections. As expected, the median delay gradually increases as the no. of PMU connections are increased but the PDC delay stabilizes after a certain extent, after which the delays are fairly constant.

Hence, the SEL 3373 PDC was tested to observe its latency.

4 PDC test system

A majority of the test tools are some combination of software tools that allow manual and/or automated testing per given PDC requirements. As Figure shows, a very basic approach would be to analyze PDC requirements, express them as requirements by example, and then create acceptance tests based on the requirements. The PJM Synchrophasor Deployment Project on PMU & PDC Testing is based on this same idea. The acceptance tests are implemented using testing tools that allow interfacing to the PDC under test and test reporting. Test tools can be in a form of individual applications or software libraries combined in different scenarios.

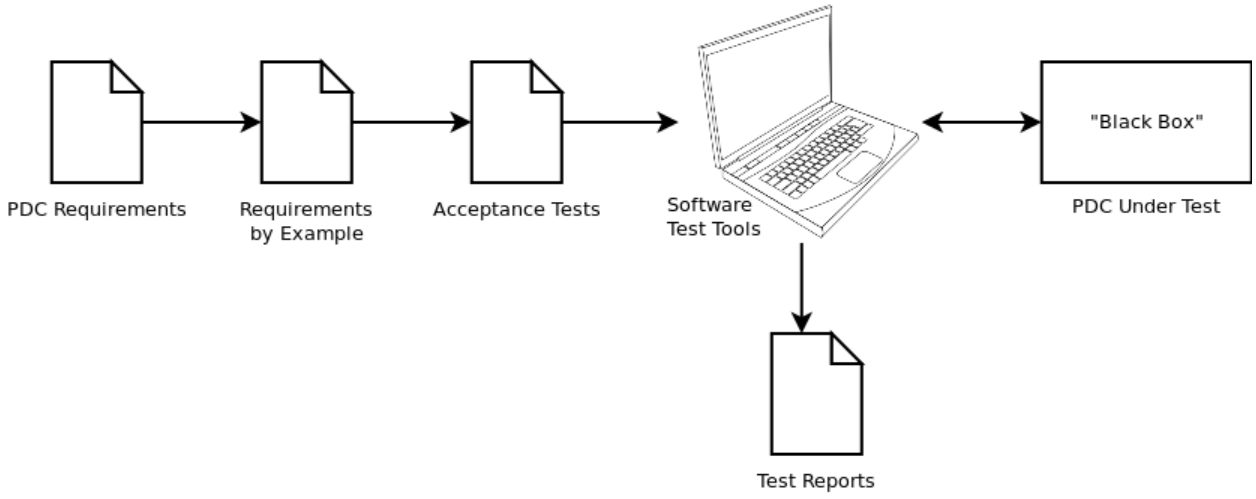

Figure 4.1 A generic set up for PDC testing using software test tools [5]

The tools allow for “off-line” and “on-line” testing and the testing can take place in labs or in the field. The “off-line” testing as illustrated before assumes that the test tools directly interface to the PDC under test and generate PMU/PDC input data streams, as well as capture the output data stream. Based on the generated input streams and received output streams the test tools are used for analysis of data and test reporting. The off-line tests can be performed both in lab or field environment. On-line tools are expected to interface to “live” PMU/PDC set ups in using data

stream “listeners” that should not affect the PMU/PDC operation. Figure 4.2 describes an example how the test data can be collected while running the test cases from the PDC test system. In this case the criterion to capture the test data is a time window defined by the time stamps of interest. Both sets of collected data frames represent the test data that will be used with PDC test system.

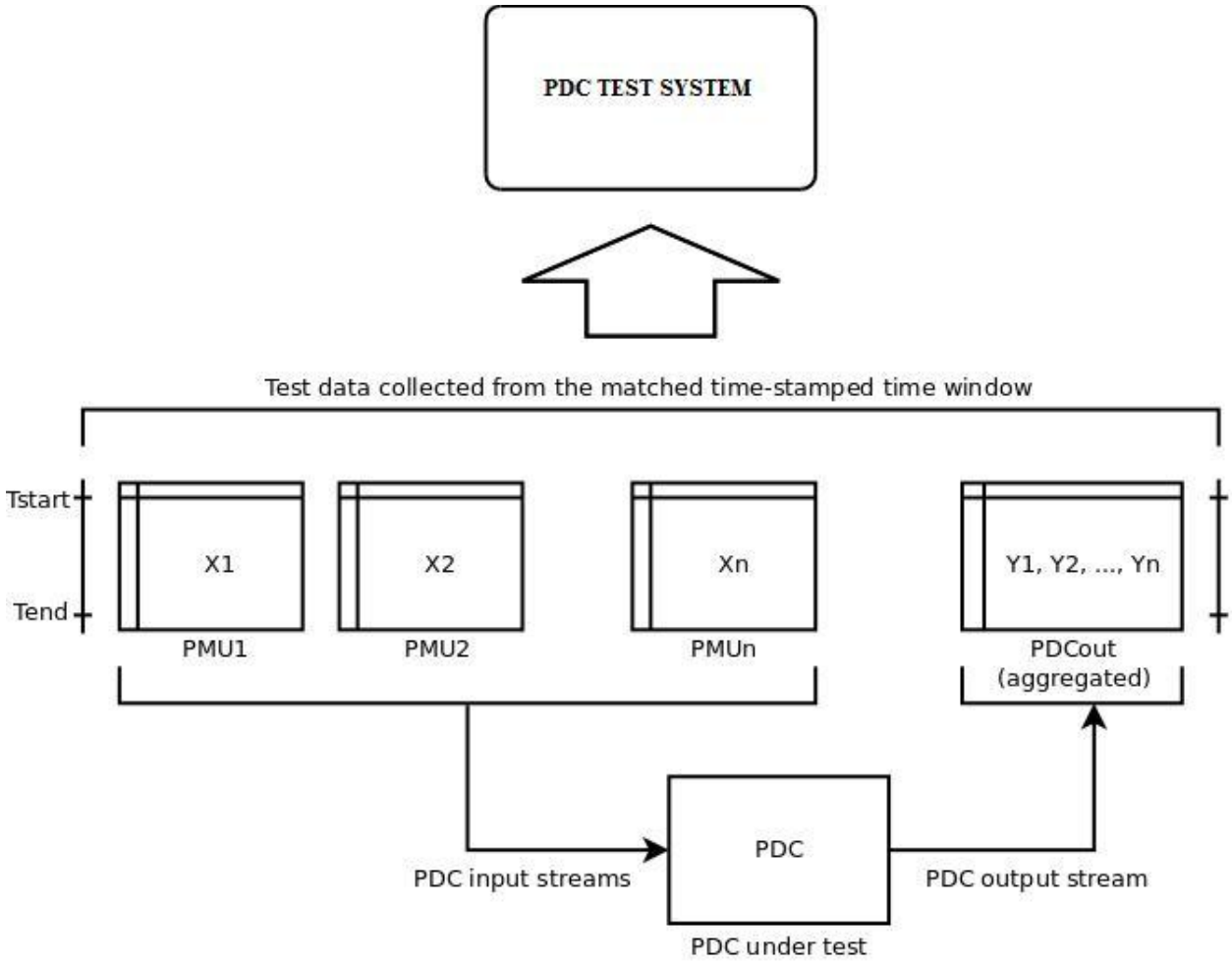

Figure 4.2 Collecting test data to be used with PDC test suite [5]

Test tools for on-line testing should be able to allow capturing data frames in predefined or desired time frames so that data frames from the input data streams can be compared with corresponding output data frames of the PDC under test. [5]

4.1 General System Architecture

The overall system architecture of the developed PDC Test System is shown in Figure 4.3.

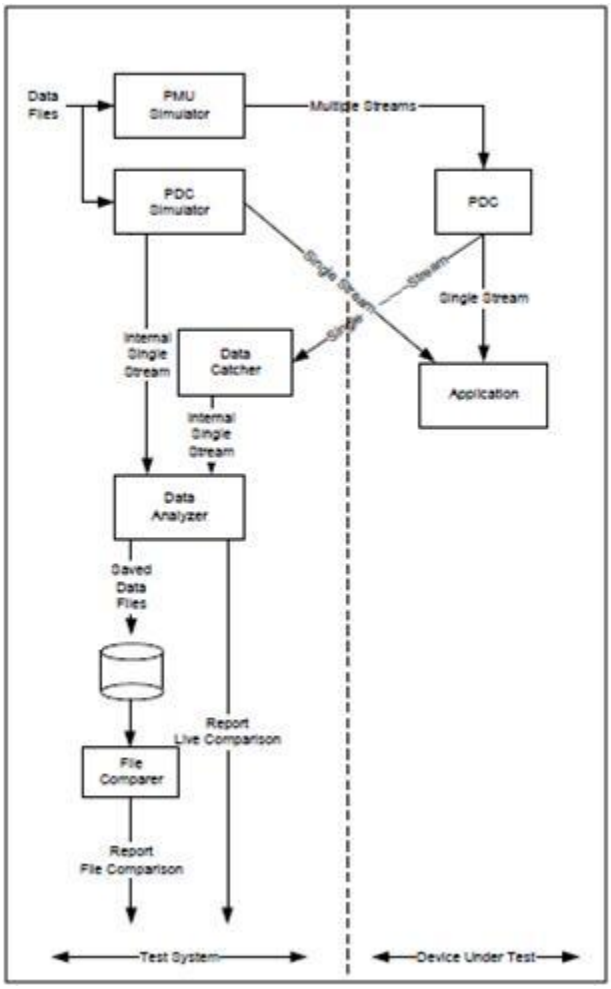

Figure 4.3 PDC Test System Diagram

The test system (shown on the left side of the figure) can be used to test PDCs or Application functions (shown on the right side of the figure). The test system consists of the following components.

4.1.1 PMU Simulator

PMU simulator function is critical for performing off-line testing of PDCs. This is an important tool in testing its data re-sampling, validation and error checking functions. The PMU simulator provides realistic PMU data streams, in the standard IEEE 37.118-2005 format, with various parameters (i.e. data rate, TCP/UDP). This is used to simulate actual working conditions for a PDC in which each unit is connected over a large power system to hundreds of PMUs. Hence, the developed PMU emulator provides functionality for simulation of up to a hundred PMU input data streams at 60 frames per second.

- The PMU simulator accepts various forms of data files, such as PSS/E simulation data, PSLF simulation data and simulates multiple PMU data streams.

- This PSS/E simulation data file consists of instantaneous voltage, angle and frequency data over a period of four seconds at an interval of $1/240^{th}$ of a second, for 100 buses with other supporting information such as bus no. and bus name which is correspondingly used by the program as the simulated measurement data for the 100 PMUs.

- The details of PMUs to be simulated such as frame rate, PMU ID, IP address, UDP & TCP port numbers and other requisite information is included in a PMU configuration file.

- The PMU data streams so simulated are transmitted to the PDC under test and deliberate errors can be simulated into the data stream to observe the response of the PDC under test.

- Detailed logs of all its operations being performed are maintained in a separate text file.

The basic overview of the PMU Simulator operation is shown in the flow chart in figure below.

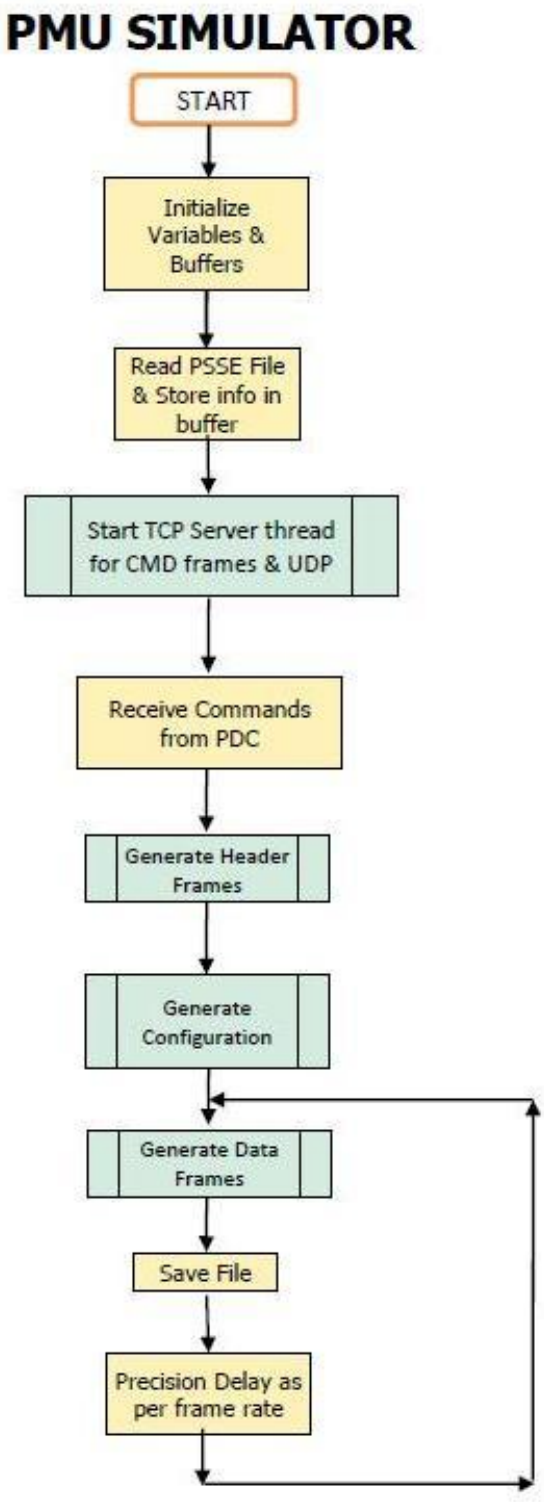

Figure 4.4 PMU Simulator Flow Chart

The source code was implemented in C# Programming language on Microsoft .NET application platform. Since the main objectives of the program was to simulate multiple PMU streams (up to 100 streams) with different data but similar standard, a technique called multi-thread programming was used. A thread in programming terms is defined as a complete single sequence of processes being performed by the program. For example, in the above figure 4.4 the sequence of events from start to end (the infinite loop of sending the data frames) can be considered as a single thread. Thus, by using multi-thread programming structures in C# the single thread depicted in figure 4.4 was replicated automatically in to 100 similar threads, in which all the global variables like buffers & initial variables are shared from the parent process. This also leads to convenient separation of multiple tasks within a process. And since most of the resources like the buffers and variables are shared from the parent process, fewer resources in terms of actual system memory are required by the entire program.

Since all threads of a process share the same global variables, a problem arises with synchronization of access to global variables. For example, let's assume you have a gobal variable X and two threads A and B. Let's say threads A and B will merely increment the value of X. When thread A begins execution, it copies the value of X into the registers and increments it. Before it gets a chance to write the value back to memory, this thread is suspended. The next thread starts, reads the same value of X that the first thread red, increments it and writes it back to memory. Then, the first thread finishes execution and writes its value from the register back to memory. After these two threads finish, the value of X is incremented by 1 instead of 2 as you would expect.

To prevent this from occurring, the 'Lock' function in C# was used. This function isolates the first thread when it is trying to access a variable and transfer control only to that thread. All the

other threads which try to access that variable at the same time are temporarily put on hold till the first thread has completed its operation with the variable. This way, threads are given access to the variable one by one.

Now, with 100 threads created and executing their various processes, the problem of synchronization of the entire program, as in the 100 PMU streams arises and one of the most critical aspects of a set of PMUs is its ability to be time synchronized to one another. Without any synchronization of the program, some of the simulated PMUs would have its thread execute at a faster rate and some would execute much slower as compared to the rate required causing different time stamps for data frames generated at the same time. This problem was solved using the 'Monitor.Wait' and 'Monitor.PulseAll' functions in C#. The Monitor.Wait function when called by a thread temporarily deactivates that particular thread and puts it on hold unto a trigger function called Monitor.Pulse or Monitor.PulseAll is called by the parent process. Thus, if you observe figure 4.4, the data frame generated but just before it is transmitted, the Monitor.Wait function is called and the thread is deactivated and stopped from executing further. This occurs for all the 100 threads. Now, when the 100^{th} thread has arrived at this point, the function Monitor.PulseAll is called and all the threads are triggered to execute at the exact same instant and data frames are transmitted to the PDC with the exact same time stamp.

4.1.2 PDC Simulator

The PDC simulator is similar to the PMU simulator, except that it generates a single data stream exactly as that of a PDC, consisting of all the signals included in all the PMUs simulated by the PMU simulator, hence constructing an ideal PDC stream in accordance with the IEEE C37.118 Standard format, against which the output of the PDC DUT can be compared for accuracy.

The output of this function is internally sent to a Synchrophasor application, for application testing, or to an internal data analyzer function as per requirement. The generation of the PDC stream is configured to be synchronized with that of the multiple PMU streams. This effect is useful in determining the latency of the PDC DUT. Although the PDC Simulator is described as a separate entity being used for different functions, the operation and working of the PDC Simulator is integrated into that of the PMU simulator so as to have a much more efficient compilation and execution of the program.

4.1.3 Data Catcher

This function accepts a Synchrophasor data stream from a PDC under test. It sends this data internally to the Data Analyzer. It is one of the critical programs on the receiving end of the PDC Test system since it is responsible for accurate capturing the PDC Output stream from the network and buffering the saved data for use by the data analyzer.

The captured data frames and configuration frames are then decoded from hexadecimal format to decimal (human readable format) and categorically separated into designated and pre-existing data fields or arrays as per the IEEE C37.118 standard and saved to a comma separated variable (.csv) file format for use by the data analyzer. The flowchart describing the basic functionality is shown in figure 4.5 below.

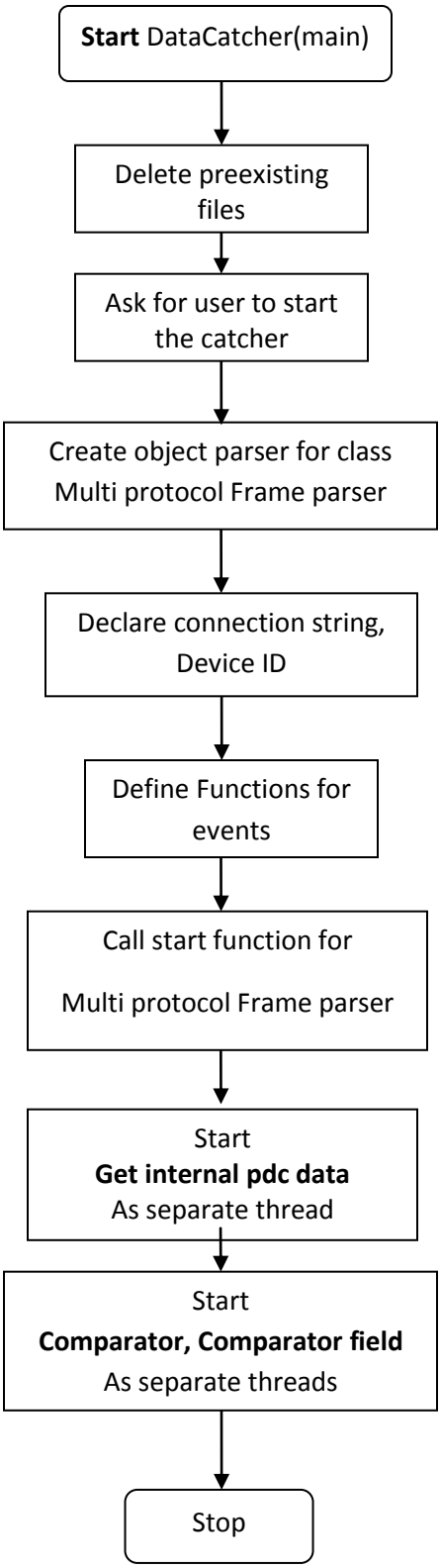

Figure 4.5: Data Catcher Flow Chart

4.1.4 Data Analyzer

This function accepts data from the internal PDC simulator and from the Data Catcher. It compares the two data streams and generates a report, consisting of the differences in the two data streams. This function also archives the data received from the two data stream inputs, for later analysis by the file comparer.

5. Conclusions and Future Work

A test system for accurate characterization and certification of PDCs complete with the techniques, procedures and test tools was developed at the Power & Energy Lab at Virginia Tech, The Virginia Tech Phasor Data Concentrator Analysis & Testing System.

The NI LabView Platform was selected for a part of the network analysis procedures and data processing functions. Wireshark network analyzer software was used to capture and catalog the data frames from the Base PMU and the DUT PDC and the data exported out to text files for processing. Microsoft .NET C# platform was used in processing the data and the software MathWorks MATLAB 7.0 was used in obtaining the plots.

This test system which has been developed is in its nascent stages and has a broad scope for future development. The primary test techniques of testing a DUT PDC with single & multiple PMU connections can be complemented with secondary test procedures to observe the response of the PDC, such as the Adaptive Protection Scheme Tests [18].

In this test, a DUT PDC with multiple PMU connections can be configured to parse the incoming PMU streams for a step change in the voltage measured by the PMUs and compare the positive sequence voltage observed with that of a threshold value in real time. If the positive sequence voltage magnitude observed is greater than the threshold value, the PDC shall transmit a trip signal to a relay. The relay is used as a measure of the time required to acknowledge the trip signal from the PDC. Each device being used has to be synchronized to UTC reference using a common GPS receiver. This way, by calculating the time difference in the observed changes in the system by each of the devices, the PDC Latency can be accurately measured. [18]

The developed PMU & PDC Simulator is one of the most promising testing tools in terms of automated PDC tests. The current version of the source code has the capability to simulate up to 100 PMUs at 60 frames per second. This value can be increased up to 300 PMUs after minor changes in the code and in the supporting PSSE files. This would serve as a very powerful and reliable tool for testing of commercial PDCs which have a capability of up to 300 incoming PMU connections.

The Simulator will also have the capability to incorporate deliberate errors such as late data, missing data, CRC errors, time quality bit errors etc. in the simulated PMU streams which will be sent to the DUT PDC. The DUT PDC should be able to recognize these errors, set error flags, archive the observed errors and relay them to the user.

References

[1] *IEEE Standard for Synchrophasors for Power Systems, IEEE Standard C37.118-2005, March, 2006.*

[2] Jim Y. Cai, et al., *Current Status and Experience of WAMS Implementation in North America.* 2005 IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific, Dalian, China.

[3] T. Babnik, et al., *Wide Area Measurement System in Action.* IEEE Power Tech, 2007 Lausanne.

[4] Biju N., et al., *Communication Delays in Wide Area Measurement Systems.* Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory, 2002.

[5] *Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, and Monitoring.* Technical Report for the North American SynchroPhasor Initiative, Performance & Standards Task Team, Working group chair Galina Antonova.

[6] *Synchrophasor Measurement Accuracy Characterization.* Technical Report for the North American Synchrophasor Initiative, Performance and Standard Task Team, team leader G. Stenbakken.

[7] Virgilio A. Centeno, et al., *Synchronized Phasor Measurement Applications in Power Systems.* IEEE Transactions on Smart Grid Vol. 1, No. 1, June 2010.

[8] Andrew Armenia, et al., *A Flexible Phasor Data Concentrator Design Leveraging Existing Software Technologies*. IEEE Transactions on Smart Grid, Vol. 1, June 2010.

[9] *IEC 61850-3 - Communication Networks and Systems in Substations - Part 3: General Requirements.*

[10] D. Tholomier, et al., *Phasor Measurement Units: Functionality and Applications.* Advanced Metering, Protection, Control, Communication and Distributed Resources, PES 2009.

[11] Bharat Bhargava, *Synchronized Phasor Measurement System Project at Southern California Edison Co.* IEEE Power Engineering Society Summer Meeting, 1999.

[12] Jian Zuo, et al., *Development of TVA SuperPDC: Phasor Applications, tools, and Event Replay*. Power and Energy Society General Meeting – Conversation and Delivery of Electrical Energy in the 21st Century, 2008 IEEE.

[13] *IEEE C37.111-1999, IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems, June 1999.*

[14] *IEEE C37.118.1, Draft 1.8 "Standard for Synchrophasor Measurements for Power Systems", May 2011.*

[15] *PMU System Testing and Calibration Guide.* Technical Report for the North American Synchrophasor Initiative, Performance and Standard Task Team, team leader G. Stenbakken.

[16] Wireshark Network Protocol Analysis software. *www.wireshark.org/*

[17] IEEE Standard 1588-2002, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.*

[18] Ryan Quint, *Practical Implementation of a Security-Dependability Adaptive Voting Scheme Using Decision Trees.* Master's Thesis, Virginia Tech, Nov 2011.

Appendix A. Device Under Test

This report presents the result of the PDC testing performed at Virginia Tech on the following device:

SEL – 3373 (Figure A.1 & A.2).

P/N: 3373#422014

S/N: 1111650153

RMA#: 70693

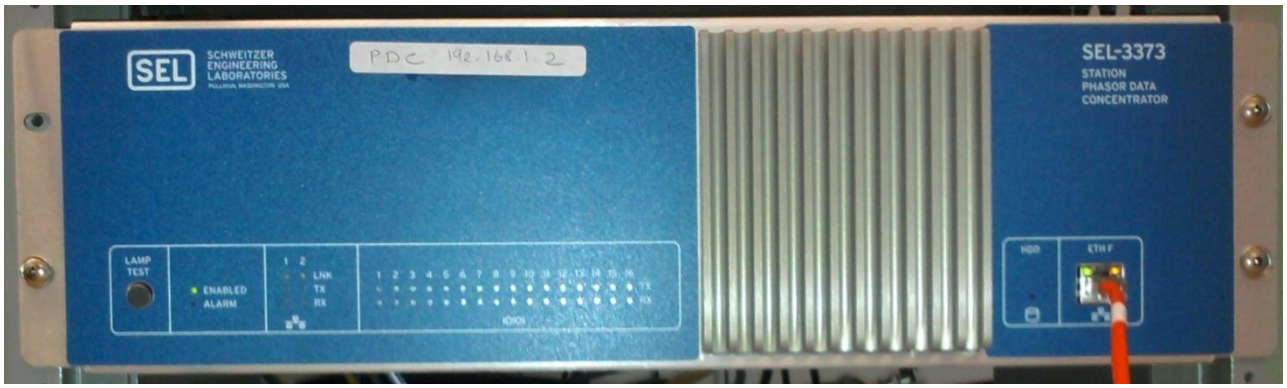

Figure A.1 SEL 3373 PDC Front View

Figure A.2 SEL 3373 PDC Rear View