DESARROLLO DE UN CONCENTRADOR DE DATOS DE SINCROFASORES BASADO EN EL ESTÁNDAR IEEE C37.118.2

JUAN CARLOS VARGAS RUBIO

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
PROYECTO CURRICULAR INGENIERÍA ELECTRÓNICA
LABORATORIO DE INVESTIGACIÓN EN FUENTES ALTERNATIVAS DE ENERGÍA
LIFAE
BOGOTA D.C. 2017

DESARROLLO DE UN CONCENTRADOR DE DATOS DE SINCROFASORES BASADO EN EL ESTÁNDAR IEEE C37.118.2

TRABAJO DE GRADO
MODALIDAD PASANTÍA

JUAN CARLOS VARGAS RUBIO
CÓDIGO: 20121005042

DIRECTOR EXTERNO
IVÁN DARÍO CLAROS GÓMEZ
GERENTE DE DESARROLLO TECNOLÓGICO
AXON GROUP LTDA

DIRECTOR INTERNO
OSCAR DAVID FLOREZ CEDIEL
DOCENTE DE PLANTA FACULTAD DE INGENIERÍA
UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
PROYECTO CURRICULAR INGENIERÍA ELECTRÓNICA
LABORATORIO DE INVESTIGACIÓN EN FUENTES ALTERNATIVAS DE ENERGÍA
LIFAE
BOGOTA D.C. 2017

CONTENIDO

1. INTRODUCCIÓN **1**
2. OBJETIVOS **2**
2.1. OBJETIVO GENERAL 2
2.2. OBJETIVOS ESPECÍFICOS 2
3. MARCO TEÓRICO **3**
3.1. BOOST ASIO 3
3.2. WPF 3
3.3. XAML 3
3.4. MVVM 4
3.5. ARQUITECTURA MULTIPROCESO 5
3.6. SOCKET 5
3.7. NPCAP 5
3.8. AKKA 5
4. DESARROLLO DEL CRONOGRAMA DE ACTIVIDADES **6**
5. DESARROLLO DE OBJETIVOS **8**
5.1. ESTÁNDAR IEEE C37.118.2 - 2011 9
5.1.1. Mensaje de Configuración 10
5.1.2. Mensaje de Datos 11
5.1.3. Mensaje de Cabecera 12
5.1.4. Mensaje de Comando 12
5.2. PROTOTIPO DEL CLIENTE IEEE C37.118.2 13
5.3. MÓDULO DE SINCROFASORES DEL RELÉ SEL 451 16
6. DESARROLLOS ADICIONALES **19**
6.1. INTERFAZ GRÁFICA DE USUARIO PARA EL PDC 19
6.2. PDC COMO CLIENTE DE MULTIPLES PMU 22
6.3. PRUEBAS DEL PDC CON EL RELÉ SEL 451 23
7. TRABAJO A FUTURO **35**
9. CONCLUSIONES **36**
BIBLIOGRAFÍA **37**

1. INTRODUCCIÓN

Las redes de sincrofasores son sistemas orientados a la adquisición de datos, supervisión, monitoreo y control en tiempo real de la red eléctrica. Son ampliamente utilizadas en redes eléctricas de área amplia ya que cada medida fasorial tomada de la red viene sincronizada, gracias a la tecnología GPS, por una estampa de tiempo dada según UTC. Esto permite ver el estado de la red en lugares geográficamente distantes en el mismo instante de tiempo para poder identificar el lugar y momento exacto en el que ocurrió un evento y tomar acciones de control para llevar el sistema a condiciones normales de operación. Algunas de las aplicaciones de la tecnología sicrofasorial incluyen control de área amplia, validación del modelo del sistema, determinación de márgenes de estabilidad, maximización de la carga estable del sistema, registro de perturbaciones, visualización de la respuesta dinámica del sistema, estimación de estados, entre otros.

Dentro de una red de sincrofasores se pueden identificar dos componentes principales, las unidades de medición fasorial (o PMU, por su nombre en inglés Phasor Measurement Unit) y los concentradores de datos de fasores (o PDC, por su nombre en inglés Phasor Data Concentrator). Las PMU son dispositivos que miden y entregan sincrofasores de voltaje y corriente, frecuencia del sistema, tasa de cambio de frecuencia (o ROCOF, por su nombre en inglés Rate Of Change Of Frequency), una estampa de tiempo y los metadatos necesarios para la interpretación de la información. Los PDC actúan como nodos en la red de comunicaciones donde se procesan los datos de sincrofasores de diferentes PMU u otros PDC y se envían como un flujo de datos a un PDC de más alto nivel. Algunas funciones de los PDC pueden ser: realizar chequeos de calidad de los datos, almacenamiento de datos para análisis posterior y visualización de datos, entre otros.

El presente documento expone las etapas de desarrollo y validación realizadas para el diseño e implementación de un concentrador de datos de sincrofasores basado en el estándar IEEE C37.118.2 como trabajo de grado en la modalidad de pasantía en la compañía Axon Group. El PDC puede enviar, recibir e interpretar datos de una o más unidades de medición sincrofasorial. Adicionalmente, cuenta con una interfaz gráfica de usuario que facilita su configuración y la visualización de la información proveniente de diferentes fuentes de forma dinámica.

El funcionamiento del concentrador fue probado con la PMU disponible en Axon Group y su desempeño fue sobresaliente. La conexión TCP/IP a través de ethernet funciona adecuadamente a altas tasas de transmisión, el protocolo de comunicación sincrofasorial probó estar correctamente implementado ya que fue posible la transmisión de datos entre el software y la PMU, la interfaz gráfica es amigable con el usuario ya que es bastante intuitiva y agradable a la vista, las gráficas dinámicas en tiempo real se actualizan rápidamente y ofrecen la información visual que el usuario necesita. Finalmente, el PDC puede atender varios servidores de manera completamente independiente.

2. OBJETIVOS

2.1. OBJETIVO GENERAL

Desarrollar un concentrador de datos de sincrofasores (PDC) capaz de enviar, recibir e interpretar datos de al menos una unidad de medición sincrofasorial (PMU).

2.2. OBJETIVOS ESPECÍFICOS

- Analizar y estudiar las normas IEEE que definen el funcionamiento de los elementos y el protocolo de comunicación en una red de sincrofasores.

- Diseñar y desarrollar un prototipo cliente del protocolo de transmisión sincrofasorial (IEEE Std. C37.118.2-2011) mediante un socket de comunicación TCP/IP.

- Validar el funcionamiento del PDC mediante herramientas de simulación de comunicación Cliente – Servidor.

- Explorar el funcionamiento de la PMU disponible en Axon Group.

- Probar el correcto funcionamiento del PDC utilizando la PMU disponible en la empresa y la herramienta de análisis de protocolos Wireshark.

- Realizar los ajustes finales al PDC, de ser necesario, para lograr un buen desempeño con el equipo real.

3. MARCO TEÓRICO

A continuación, se definen algunos de los conceptos que se nombrarán posteriormente en el documento y pueden facilitar su comprensión. La comprensión y utilización de estos conceptos fueron necesarias durante el desarrollo del proyecto para poder llevarlo a cabo.

3.1. BOOST ASIO

Asio es una librería multiplataforma de C++ para programación aplicada a redes y entradas/salidas de bajo nivel que provee a los desarrolladores un modelo asíncrono consistente usando un enfoque moderno de C++.

La mayoría de los programas interactúan con el exterior de algún modo, ya sea a través de un archivo, una red, un cable serial, o la consola. Algunas veces, como en el caso de las redes, operaciones entrada/salida individuales pueden tardar mucho tiempo en completarse, lo que propone retos particulares para el desarrollo de aplicaciones. Asio provee las herramientas para manejar estas operaciones de largo plazo, sin requerir programas que usen modelos de concurrencia basados en hilos y bloqueos explícitos [1].

3.2. WPF

WPF, por su nombre en inglés *Windows Presentation Foundation*, es un marco de interfaz de usuario que crea aplicaciones de cliente completas e interactivas. La plataforma de desarrollo de WPF admite un amplio conjunto de características de desarrollo de aplicaciones, incluido un modelo de aplicación, recursos, controles, gráficos, diseños, enlace de datos, documentos y seguridad. WPF usa el lenguaje de marcado de aplicaciones extensible (XAML) para proporcionar un modelo declarativo para la programación de aplicaciones [2].

3.3. XAML

XAML, por su nombre en inglés *eXtensible Application Markup Language*, es un lenguaje declarativo de marcado desarrollado por Microsoft. Es el lenguaje detrás de la presentación visual de una aplicación WPF, tal como HTML es el lenguaje detrás de la presentación de una página web. WPF usa XAML, en lugar de un lenguaje de programación, para construir interfaces de usuario de muy buena presentación. Se pueden crear elaborados documentos de interfaz de usuario utilizando XAML definiendo elementos como controles, textos, imágenes, formas, animaciones, entre otros. Sin embargo, ya que XAML es un lenguaje declarativo, requiere de código adicional para añadir lógica de ejecución a la aplicación. El código para una aplicación XAML se almacena en un archivo separado del documento XAML. Esta separación entre el diseño de la interfaz gráfica y el código detrás de la aplicación permite a los diseñadores y desarrolladores trabajar juntos en el mismo proyecto sin retrasar el progreso del otro [3].

3.4. MVVM

MVVM, por su nombre en inglés *Model-View-ViewModel*, es un patrón de arquitectura de software. Este patrón arquitectónico permite la separación entre la interfaz gráfica de usuario y la lógica detrás de la misma. De esta manera los cambios en los datos se visualizan de manera automática en la interfaz cuando estos sufren una alteración en el código que implementa la lógica [4]. La figura 1 representa la arquitectura general del modelo.

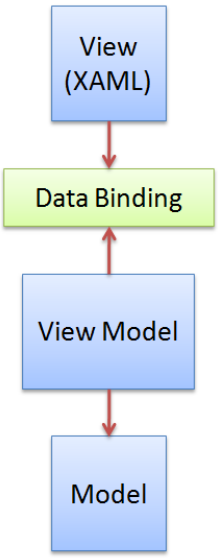

Figura 1. Arquitectura general MVVM.

La vista (*View*) es responsable de definir la estructura, diseño, y apariencia de lo que el usuario ve en pantalla. Idealmente, la vista está definida únicamente con XAML.

El modelo (*Model*) es la implementación de los objetos del dominio de la aplicación que incluye estructuras de datos junto con la lógica del programa.

El modelo de vista (*View Model*) actúa como un intermediario entre la vista y el modelo, y es responsable de manejar la lógica de la interfaz. El modelo de vista provee los datos desde el modelo de tal forma que puedan ser fácilmente usados por la vista. Adicionalmente, en el modelo de vista se realiza la implementación de los comandos que el usuario acciona desde la vista. Por ejemplo, cuando un usuario oprime un botón en la vista, esa acción dispara un comando en el modelo de vista.

El enlace de datos (Data Binding) permite la conexión entre el modelo de vista y la vista. Este enlace es el que permite relacionar, por ejemplo, el valor de una caja de texto en la interfaz con el valor de un string en el modelo de vista de tal forma que cuando se presente un cambio en el string la vista se actualice automáticamente.

3.5. ARQUITECTURA MULTIPROCESO

Es un modelo en el cual se pueden ejecutar dos o más procesos independientes de manera concurrente. Este modelo es muy común en sistemas grandes de tiempo real que recolectan información de diferentes dispositivos.

3.6. SOCKET

Socket es un concepto abstracto que representa una forma de comunicación entre procesos que se encuentran en diferentes dispositivos de una red, los sockets proporcionan un punto de comunicación por el cual se puede enviar o recibir información entre procesos. Los sockets tienen un ciclo de vida dependiendo si son sockets de servidor, que esperan a un cliente para establecer una comunicación, o socket cliente que busca a un socket de servidor para establecer la comunicación.

Las propiedades de un socket dependen de las características del protocolo en el que se implementan. Si se implementan con TCP los sockets son orientados a la conexión, se garantiza la transmisión de todos los octetos, se garantiza el orden de llegada de los mismos. Un socket TCP está formado por una pareja dirección IP y puerto.

3.7. NPCAP

Npcap es la actualización de la popular librería WinPcap, WinPcap sirve para realizar sniffing de paquetes en Windows con cualquier software analizador de paquetes tipo Wireshark. Npcap permite capturar paquetes de la interfaz de loopback, es decir, del tráfico que se intercambia entre los servicios del mismo ordenador, esto podremos hacerlo gracias a que usa Windows Filtering Platform. Después de la instalación Npcap creará un adaptador virtual llamado Npcap Loopback Adapter para poder realizar capturas de la loopback, por supuesto es compatible con Wireshark. Relacionada con esta característica tenemos también Loopback Packet Injection, que nos permitirá enviar paquetes a la loopback utilizando Winsock Kernel, ideal para utilizar Nping e inyectar paquetes. Por último, también tenemos la posibilidad de capturar paquetes 802.11, es decir, los paquetes que se intercambian a través de los adaptadores Wi-Fi [5].

3.8. AKKA

Akka es un conjunto de librerías de código abierto para diseñar sistemas escalables y flexibles que abarcan problemas relacionados con procesamiento de datos y redes [6].

Akka proporciona los siguientes beneficios:

- Procesamiento multi-hilo sin el uso de construcciones con concurrencia de bajo nivel.
- Comunicación remota transparente entre sistemas y sus componentes.

- Una arquitectura cerrada, de alta disponibilidad, elástica y escalable bajo demanda.

Todas estas características están disponibles a través de un modelo de programación uniforme: Akka explota el modelo de *Actor* para proporcionar un nivel de abstracción que facilita el desarrollo de sistemas concurrentes, paralelos y distribuidos.

El modelo de actor proporciona una abstracción que permite pensar en el código en términos de comunicación. La característica básica de los actores es que modelan el mundo como entidades con estado comunicándose entre ellas a través del paso explícito de mensajes.

Como entidades computacionales, los actores tienen las siguientes características:

- Se comunican con mensajes asíncronos en lugar de llamadas a métodos.
- Manejan su propio estado.
- Al responder a un mensaje, pueden:
    - Crear otros actores (hijos).
    - Enviar mensajes a otros actores.
    - Detener actores (hijos) o ellos mismos

La creación de actores sirve para delegar tareas, lo cual es útil para aliviar reponsabilidades y diseñar un buen modelo de actores: hay que intentar que cada uno tenga el menor número de competencias posible, de tal manera que quede claro qué mensajes debe procesar cada actor, cómo deben reaccionar a los mismos y cómo deben manejar errores [7].

Cada actor tiene un buzón, que es el lugar donde llegan todos los mensajes que recibe. Los actores están ociosos hasta que les llega un mensaje a su buzón; entonces lo procesan y, cuando terminan, siguen leyendo mensajes hasta que el buzón queda vacío. De nuevo, quedan ociosos hasta que llegue un nuevo mensaje.

4. DESARROLLO DEL CRONOGRAMA DE ACTIVIDADES

En la planificación del cronograma de actividades, expuesto en el anteproyecto presentado para este proyecto, se planeó cumplir los objetivos propuestos en 15 semanas durante las cuales se desarrollarían 6 etapas que se nombran a continuación:

Etapa 1: Estudio de los estándar IEEE C37.118 que definen las características de funcionamiento de los componentes y el protocolo de comunicación en una red de sincrofasores.

Etapa 2: Desarrollo de un socket de comunicación en Windows que permita la transmisión de datos usando el protocolo TCP-IP.

Etapa 3: Desarrollo de un concentrador de datos de sincrofasores capaz de enviar, recibir e interpretar información hacia y desde una unidad de medición sincrofasorial.

Etapa 4: Explorar el funcionamiento del RELÉ SEL 451 con funciones de sincrofasores, disponible en Axon Group.

Etapa 5: Probar el funcionamiento el PDC desarrollado utilizando la PMU disponible en Axon Group.

Etapa 6: Realizar los ajustes, de ser necesario, al PDC para lograr un buen desempeño con la PMU real.

Adicionalmente, para lograr los desarrollos adicionales que se nombrarán en el capítulo 6, fueron necesarias las siguientes etapas:

Etapa 7: Diseño y desarrollo de una interfaz gráfica de usuario para facilitar la configuración del PDC y la visualización de la información.

Etapa 8: Desarrollo de un componente gráfico que permita visualizar gráficas dinámicas en tiempo real de información sincrofasorial.

Etapa 9: Implementación de una arquitectura multiproceso que permita la conexión, transmisión y visualización de información proveniente de diferentes fuentes.

En la tabla 1 se muestra el desarrollo cronológico de cada una de las etapas del proyecto, incluyendo las etapas adicionales. Al comparar esta tabla con la presentada en el anteproyecto se comprueba que no solo se cumplió con las actividades planeadas, sino que dentro del mismo tiempo planteado inicialmente, se desarrollaron las etapas adicionales.

SEMANA	**ETAPAS**								
	1	2	3	4	5	6	7	8	9

1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									

Tabla 1. Cumplimiento del cronograma de actividades.

5. DESARROLLO DE OBJETIVOS

5.1. ESTÁNDAR IEEE C37.118.2 - 2011

El estándar IEEE para la transferencia de datos sincrofasoriales en una red eléctrica de potencia define 4 tipos de mensaje:

- **Mensaje de Datos:** Son las medidas tomadas por las PMU: sincrofasores, frecuencia, ROCOF, palabras analógicas y digitales.

- **Mensaje de Configuración:** Datos de configuración de los dispositivos, cantidad de PMU incluidas en la trama de datos, formato de los datos, cantidad de señales y sus nombres, factores de escala y demás metadatos necesarios para la interpretación de la información.

- **Mensaje de Cabecera:** Información descriptiva enviada por los dispositivos, pero proporcionada por el usuario. Se envía como caracteres ASCII.

- **Mensaje de Comando:** Enviados por los dispositivos para controlar la comunicación. Con este tipo de mensaje se inicia o termina la transmisión de datos y se solicitan las tramas de configuración y cabecera.

Los cuatro tipos de mensaje tienen la misma estructura base:

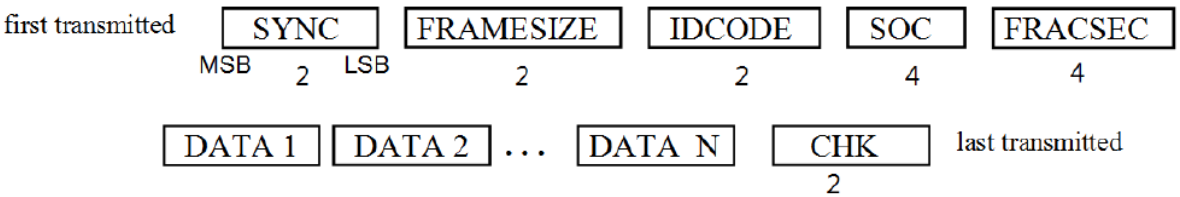

Figura 2. Estructura base de los mensajes en una red de sincrofasores [8].

SYNC: Palabra de sincronización, dos bytes que identifican el inicio de una trama y el tipo de mensaje.
FRAMESIZE: Cantidad de bytes del mensaje, máximo 65.535 bytes.
IDCODE: Identifica el destino de la trama si es un mensaje de comando y el origen si es otro tipo de mensaje.
SOC: Estampa de tiempo, inició el primero de enero de 1970 y se reinició en 2016.
FRACSEC: Fracción de segundo y calidad de tiempo. Instante de la medida para mensajes de datos e instante de la transmisión para otro tipo de mensaje.
DATA: Depende del tipo de mensaje.
CHK: Chequeo de redundancia cíclica. Sirve para determinar la integridad de la trama.

Para realizar las pruebas iniciales al PDC utilizando las herramientas de simulación de comunicación cliente-servidor se utilizan las tramas de ejemplo disponibles en la norma y

otras con algunas modificaciones. Con estas tramas de ejemplo resulta más fácil entender cómo funciona el protocolo y como debe ser implementado.

5.1.1. Mensaje de Configuración

Trama de Configuración			
Campo	**Tamaño (Bytes)**	**Valor**	**Descripción**
SYNC	2	AA 31	Byte de sincronización seguido del tipo de trama
FRAMESIZE	2	01 C6	Número de bytes de todo mensaje. 454.
IDCODE	2	1E 36	Identificación de la fuente. 7734.
SOC	4	44 85 27 F0	Segundo en que se envió en mensaje
FRACSEC	4	56 07 10 98	Fracción de segundo
TIME_BASE	4	00 0F 42 40	Divisor para la estampa de tiempo de los datos (1.000.000 µs)
NUM_PMU	2	00 01	Número de PMUs
STN	16	53 74 61 74 69 6F 6E 20 41 20 20 20 20 20 20 20	Nombre de la PMU. Station A.
IDCODE	2	1E 36	Identificación de la PMU. 7734.
FORMAT	2	00 04	Formato de los datos. Bits 15 - 4 no se usan. Bit 3: FREQ/DFREQ, **0=Entero 16 bits**, 1=Punto flotante Bit 2: ANALOG, 0=Entero 16 bits, **1=Punto flotante** Bit 1: PHASORS, **0=Entero 16 bits**, 1=Punto flotante Bit 0: PHASORS,**0=Fasor en rectangulares**, 1=Fasor en polares
PHNMR	2	00 04	Número de Fasores
ANNMR	2	00 03	Número de valores analógicos
DGNMR	2	00 01	Número de valores digitales
CHNAM	16 x (PHNMR + ANNMR + 16 x DGNMR)		Nombre de cada fasor, valor analógico y valor digital. ASCII.
PHUNIT	4 x PHNMR	00 0D F8 47 00 0D F8 47 00 0D F8 47 01 00 B2D0	Factores de conversión de cada fasor. Fasor_MAG = PHUNIT × 0.00001 Voltaje PHUNIT = (300 000/32768) × 10E+5=**915527** Corriente PHUNIT =

			(15 000/32768) × 10E+5=**45776**
ANUNIT	4 x ANNMR	00 00 00 01 01 00 00 01 02 00 00 01	Factor de conversión canales analógicos. Byte más significativo 0=Un punto en la señal, 1=valor rms, 2=valor pico. 3 Bytes dan el factor de conversión.
DIGUNIT	4 x DGNMR	00 00 FF FF	Enmascaramiento para las palabras digitales.
FNOM	2	00 00	Indica nominal de 60 Hz. 01 sería 50 Hz.
CFGCNT	2	00 16	Conteo de cambios en la conf. 22 es ej.
DATA_RATE	2	00 1E	Fuente entrega 30 mensajes por segundo
CHK	2	D5 D1	Chequeo de integridad.

Tabla 2. Mensaje de Configuración.

Para cada nombre en el campo CHNAM se asignan 16 bytes por lo cual no se pueden configurar nombres de más de 16 caracteres. Los nombres de los fasores en la trama de ejemplo proporcionada por la norma son VA, VB, VC e I1, los valores analógicos se nombran ANALOG1, ANALOG2 y ANALOG3 y las 16 palabras digitales son BREAKER X STATUS. Todos estos nombres hacen que la longitud de la trama de configuración sea tan extensa con respecto a los demás mensajes.

5.1.2. Mensaje de Datos

Trama de Datos			
Campo	**Tamaño (Bytes)**	**Valor**	**Descripción**
SYNC	2	AA 01	Byte de sincronización seguido de tipo de trama
FRAMESIZE	2	00 34	Tamaño de la trama. 52 Bytes.
IDCODE	2	1E 36	ID de PMU. 7734
SOC	4	44 85 36 00	Segundo en que se envió en mensaje
FRACSEC	4	00 00 41 B1	Fracción de segundo
STAT	2	00 00	Estado de los datos. Correcto.
PHASORS	4xPHNMR o 8xPHNMR	39 2B 00 00 E3 6A CE 7C E3 6A CE 7C 04 44 00 00	14635∠0° (134 kV ∠0°) 14635∠-120° (134 kV ∠-120°) 14635∠-120° (134 kV ∠-120°) 1092∠0° (500 A ∠0°)
FREQ	2 o 4	09 C4	(FNOM+2500 mHz) 62.5 Hz
DFREQ	2 o 4	00 00	0
ANALOG	2xANNMR o 4xANNMR	42 C8 00 00 44 7A 00 00 46 1C 40 00	100 1000 10000
DIGITAL	2 x DGNMR	3C 12	Palabras digitales válidas

CHK	2	D4 3F	Chequeo de integridad.

Tabla 3. Mensaje de Datos.

5.1.3. Mensaje de Cabecera

En la norma no hay ejemplo de la trama de cabecera por lo cual esta trama se creó para realizar pruebas. El mensaje de cabecera que se quiere transmitir es PMU_AXON. Este mensaje puede tener cualquier tipo de información, la que el usuario que configure la PMU desee proporcionar.

Trama de Cabecera			
Campo	**Tamaño (Bytes)**	**Valor**	**Descripción**
SYNC	2	AA 11	Byte de sincronización seguido de tipo de trama
FRAMESIZE	2	00 18	Tamaño de la trama. 24 Bytes.
IDCODE	2	1E 36	ID de PMU. 7734
SOC	4	44 85 36 00	Segundo en que se envió en mensaje
FRACSEC	4	00 00 41 B1	Fracción de segundo
DATA k	8	50 4D 55 5F 41 58 4F 4E	Información proporcionada. PMU_AXON.
CHK	2	5B EA	Chequeo de integridad.

Tabla 4. Trama de Cabecera.

5.1.4. Mensaje de Comando

Los 3 tipos de mensaje anteriores son enviados desde la PMU hacía el PDC, por lo tanto son los mensajes que van a ser enviados con la herramienta de simulación, por otro lado los mensajes de comando van a ser enviados desde el PDC por lo tanto deben ser construidos en el código. La tabla 5 muestra la estructura de un mensaje de comando.

Trama de Comando			
Campo	**Tamaño (Bytes)**	**Valor**	**Descripción**
SYNC	2	AA 41	Byte de sincronización seguido de tipo de trama
FRAMESIZE	2	00 12	Tamaño de la trama, 18 bytes
IDCODE	2	7734	Identifica destino del mensaje. 7734.
SOC	4	44 85 60 30	Segundo en que se envió en mensaje
FRACSEC	4	0F 0B BF D0	Fracción de segundo
CMD	2	00 02	Comando enviado. Enviar trama de Configuración 2
CHK	2	CE 00	Chequeo de integridad.

Tabla 5. Trama de Comando.

5.2. PROTOTIPO DEL CLIENTE IEEE C37.118.2

El prototipo inicial del PDC se desarrolló completamente en C++ utilizando el entorno de desarrollo Visual Studio Community 2017. Este prototipo puede intercambiar información con otro proceso, ya sea en el mismo dispositivo o en uno diferente, a través de un socket TCP/IP cuyos parámetros son configurables por el usuario. La interacción con el usuario se hace por medio de la aplicación de consola. Para simular el funcionamiento de este prototipo se utilizó el software Hércules, un simulador de comunicación cliente servidor para los protocolos TCP y UDP, y la librería Npcap para poder ver los paquetes transmitidos entre el hércules y el Cliente C37.188.2 en Wireshark.

En la figura 3 se muestra el Hércules configurado como servidor TCP, ya que va a actuar como PMU, y se escoge el puerto 4712 ya que es el puerto por defecto para sincrofasores según la norma.

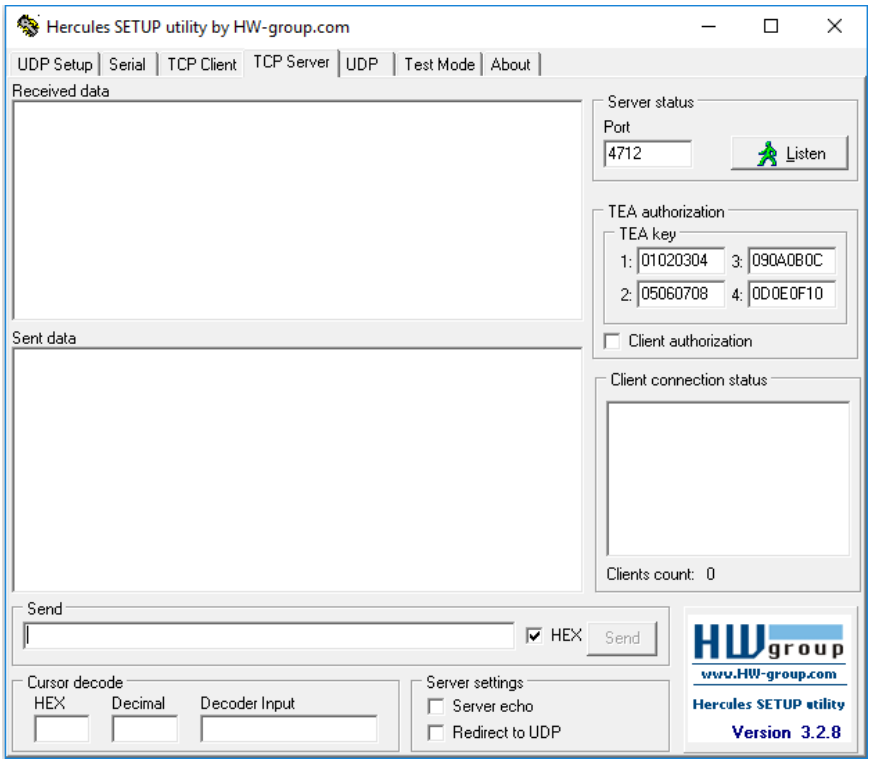

Figura 3. Hércules como servidor

Al dar clic en Listen, Hercules empieza a escuchar las peticiones del cliente. La ventaja de este simulador es que permite enviar tramas en formato Hexadecimal al habilitar la opción HEX mostrada en la figura. En la figura 4 se muestra el proceso de configuración del PDC desde la aplicación de consola.

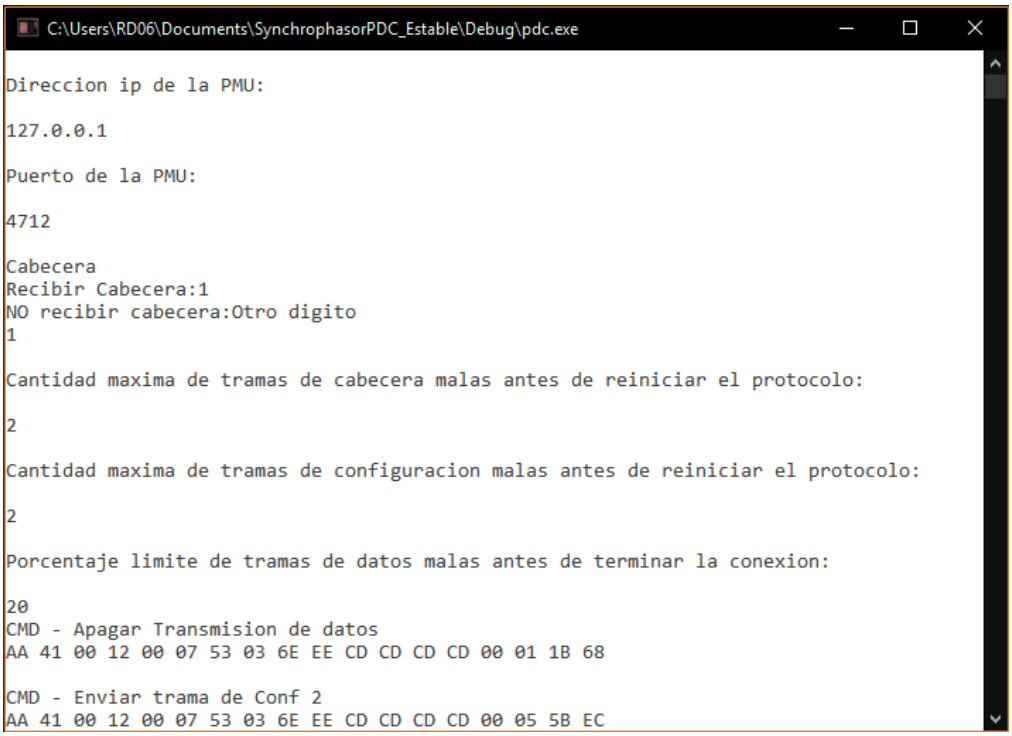

Figura 4. Configuración del PDC desde consola.

Los parámetros de conexión son la dirección IP, localhost ya que el Hercules está corriendo en el mismo dispositivo, y el puerto, luego el usuario escoge si quiere pedir o no la trama de cabecera. Adicionalmente, este PDC permite que el usuario determine la cantidad de veces que se va a pedir la retransmisión de tramas de cabecera o de configuración fallidas, en caso de ser necesario, antes de reiniciar el protocolo. Se define como trama fallida aquella para la cual el cliente detecta bytes de sincronización incorrectos, incoherencia entre la longitud de la trama y el campo FRAMESIZE o error en el chequeo de redundancia cíclica, CHK.

Al finalizar la configuración el PDC inicia la comunicación con el cliente, inicialmente envía un comando para apagar la transmisión de datos, en caso de que la PMU esté intentando enviar alguna trama en el momento de la conexión, y luego un comando pidiendo la trama de configuración. En la figura 5 se muestra el diagrama de secuencia básico de la comunicación entre una PMU y un PDC que permite comprender de mejor manera el comportamiento que se espera según el protocolo.

Como se aprecia en el diagrama, el cliente inicia la comunicación solicitando la trama de configuración. El PDC necesita tener la trama de configuración de la PMU para poder interpretar la trama de cabecera y las tramas de datos. Posteriormente el cliente solicita la trama de cabecera, si el usuario así lo determino, y luego inicia la transmisión de datos. La comunicación finaliza cuando el PDC envía el comando para apagar la transmisión de datos.

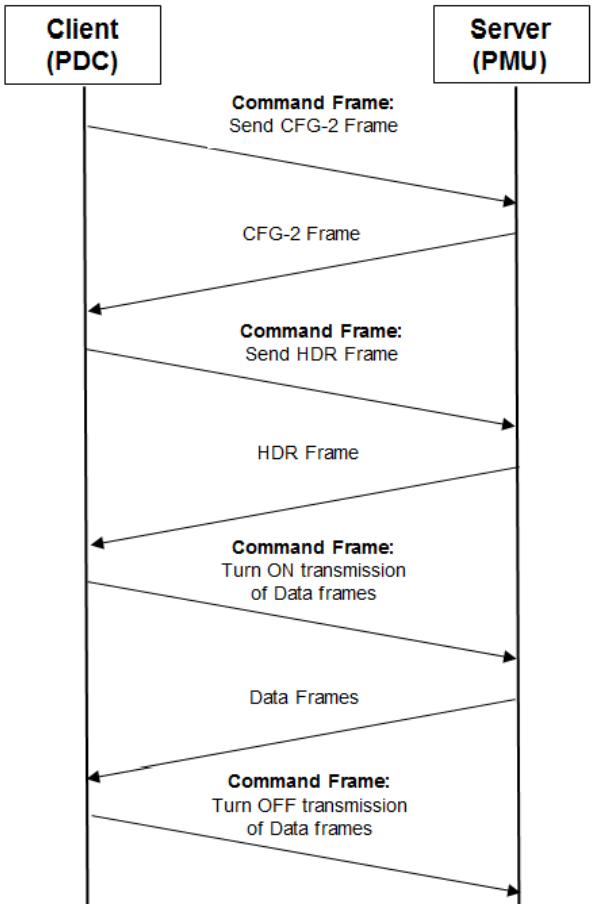

Figura 5. Diagrama de secuencia del protocolo.

En la figura 6 podemos ver las tramas capturadas por Wireshark, gracias a Npcap y el filtro synphasor, de los mensajes enviados entre el Cliente y el Servidor.

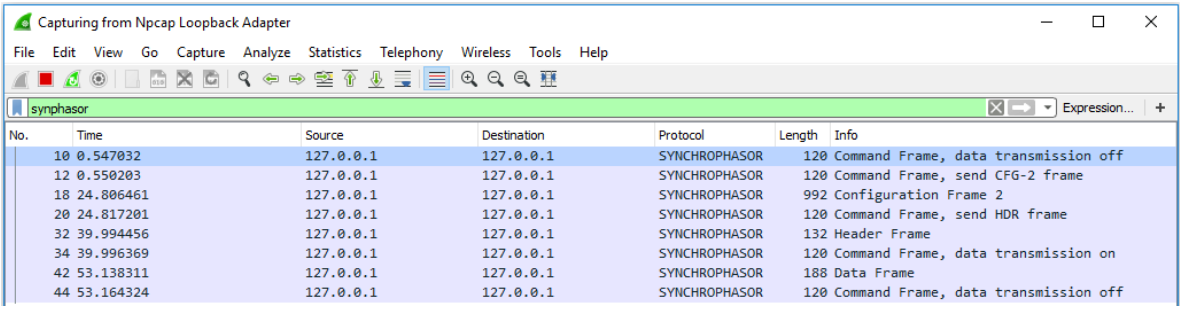

Figura 6. Tramas del protocolo capturadas por Wireshark.

Con Wireshark podemos verificar que el protocolo funciona correctamente según lo especificado por la norma, la fuente y el destino son localhost ya que los dos procesos se están ejecutando en el mismo ordenador. Las tramas desde el servidor se enviaron manualmente mientras que las tramas desde el PDC se enviaron de forma automática. En la figura 7 se ve como el PDC interpretó correctamente la información de la trama de cabecera y la trama de datos, lo que por ende significa que la trama de configuración también fue correctamente interpretada.

```
C:\Users\RD06\Documents\Visual Studio 2017\Projects\PdcPruebas\Debug\PDC.exe
Header Frame Sent by PMU
AA 11 00 18 1E 36 44 85 27 F0 56 07 10 98 50 4D 55 5F 41 58 4F 4E 5B EA

INFO: PMU_AXON

Turn ON Data Transmission
AA 41 00 12 1E 37 59 2D D5 78 00 09 08 80 00 02 F9 45

PMU: Station A
Data Frame sent by PMU
AA 01 00 34 1E 36 44 85 36 00 00 00 41 B1 00 00 39 2B 00 00 E3 6A CE 7C E3 6A 31 83 04 44 00 00 09 C4 00 00 42 C8 00 00
44 7A 00 00 46 1C 40 00 3C 12 D4 3F

Controller has received a new Data from client

MY PMU----------------------------------------
IDCODE: 7734
SOC: 1149580800
FRACSEC: 0.016817
VA              :133987|_ 0
VB              :134003|_ -120.001
VC              :133995|_ 120.003
I1              :499.874|_ 0
Frequency: 62.5 Hz
ROCOF: 1 Hz
ANALOG1         :100
ANALOG2         :1000
ANALOG3         :10000
Turn Off Data Transmission
AA 41 00 12 1E 37 59 2D D5 85 00 0B 98 C0 00 01 93 05
```

Figura 7. Trama de cabecera y de datos interpretadas por el Cliente.

Adicionalmente se aprecia como el cliente muestra todas las tramas enviadas y recibidas en formato hexadecimal.

5.3. MÓDULO DE SINCROFASORES DEL RELÉ SEL 451

Axon Group dispone del Relé SEL 451 con módulos funcionales de PMU mostrado en la figura 8.

Figura 8. Relé SEL 451 [9].

Más específicamente, este equipo es un controlador de bahía con funciones de relé de protección. Es un dispositivo electrónico inteligente (IED) programable que permite detectar y despejar perturbaciones en el sistema eléctrico. Debido a las funciones que ofrece, generalmente se utiliza sobre líneas de transmisión a nivel de subestaciones.

Para identificar los tipos de protecciones utilizados en subestaciones a nivel internacional y proporcionados por los IED, el Instituto Americano de Estándares Nacionales (ANSI) creó una nomenclatura universal que permite interpretar las diferentes funciones de protección de una forma más ágil. Dentro de las funciones de protección ofrecidas por el SEL 451 se encuentran la función 50 (sobrecorriente instantánea), la función 79 (recierre) y la función 81 (frecuencia), entre otras. Estas tres funciones destacan por ser de las más utilizadas a nivel de líneas de transmisión.

La función 50 mide el tiempo que la corriente permanece por encima de los límites permitidos. El ingeniero de protecciones programa en el equipo el tiempo, generalmente a escala de milisegundos, que el relé va a esperar antes de abrir el interruptor correspondiente. Este tipo de anomalía se puede dar, por ejemplo, por el contacto existente entre dos cables debido al movimiento de un árbol.

La función de recierre es muy común en líneas de transmisión que alimentan circuitos urbanos. Las empresas de energía no pueden cobrar por la potencia que es transmitida pero por alguna razón no llega al usuario final. Por esta razón, cuando ocurre alguna anomalía que causa la apertura de un interruptor se busca que el interruptor se cierre lo más rápido posible para que el usuario final no se vea afectado. Esto se logra programando en el relé la función de recierre, una vez el equipo detecta que la anomalía finalizó empieza a correr el tiempo de espera programado, generalmente milisegundos, para poder cerrar el interruptor y reestablecer el servicio de manera automática.

La función 81, frecuencia, detecta cuando la frecuencia de la señal medida se sale de los límites permitidos, generalmente 60 ± 0.5 Hz. Cuando la frecuencia es baja el relé empieza a quitar carga a la línea, esto se conoce como deslastre, hasta que la frecuencia vuelve a alcanzar su valor normal.

Además de las funciones anteriormente mencionadas, el SEL 451 ofrece el servicio de medición sincrofasorial. Esto permite a los operadores del sistema comprender el estado de la red con visualizaciones en tiempo real de los ángulos de fase y la frecuencia del sistema. Las mediciones fasoriales sincronizadas de alta exactitud proporcionan información y control para hacer coincidir la frecuencia y el ángulo de fase para actividades críticas, como conmutación, arranque y transferencia de potencia. En este proyecto se hace uso de esta función para probar el correcto funcionamiento del PDC.

Luego de configurar el relé, se conectó al PDC usando un cable Ethernet y se pudo comprobar gracias a Wireshark que el protocolo se estaba ejecutando correctamente, con lo cual se cumple el objetivo principal del proyecto, *enviar, recibir e interpretar datos de al menos una unidad de medición sincrofasorial*. Sin embargo, aunque el protocolo se implementó correctamente, la aplicación de consola no ofrece una buena experiencia al usuario, la impresión en pantalla de la información de cada trama de datos se hace de manera muy rápida y no es posible tener un registro visual en tiempo real de las variaciones de los fasores y

demás medidas entregadas por la PMU, además de que una vez ejecutado el protocolo el usuario no puede realizar ninguna acción sobre la aplicación aparte de detenerla. Por estas razones, entre otras, luego de estas pruebas se determina que para que el PDC sea funcional debe contar con una interfaz gráfica de usuario que permita observar los datos de manera dinámica y así poder analizar sus variaciones en tiempo real, además de permitir al usuario interactuar con la aplicación.

En la sección 6.3 se explica detalladamente el proceso de configuración del módulo de sincrofasores del relé y se muestran los resultados de las pruebas ya con la interfaz gráfica implementada.

6. DESARROLLOS ADICIONALES

6.1. INTERFAZ GRÁFICA DE USUARIO PARA EL PDC

El desarrollo de la interfaz gráfica se realizó en C#, por esta razón fue necesario crear una clase proxy en C++ cuya función es pasar a la aplicación en C# los datos que esta necesita manipular. Estos datos son aquellos que se requieren para interactuar con el usuario como los valores de los fasores y sus nombres, la frecuencia, las estampas de tiempo, entre otros. Para que C# pueda usar estos datos debe hacerse una conversión en su tipo ya que algunos tipos de datos de C++ no existen de la misma manera en C#. Por ejemplo los vectores de C++ se pasaron como colecciones a C#. Para este fin se creó una clase en C# cuyos atributos son los datos que se espera recibir de C++ pero, con tipos de dato manipulables por C#. En resumen, el proxy de C++ se encarga de actualizar los valores dicha clase en C#. Adicionalmente, ya que los datos de configuración del PDC son ingresados por el usuario a través de la interfaz, es necesario pasar estos datos al cliente en C++ para que pueda funcionar. En este caso también se realiza una conversión de los tipos de datos.

La aplicación desarrollada en C# es del tipo WPF, que se escogió en lugar de Windows Forms ya que permite un mejor aislamiento entre el diseño de la interfaz gráfica y la lógica detrás de la misma, además de ofrecer un mejor rendimiento ya que utiliza los recursos de la tarjeta gráfica del equipo donde está corriendo. Adicionalmente, WPF cuenta con librerías que ayudan a la creación de gráficos dinámicos como Dynamic Data Display [10], facilitando así el trabajo de graficar información sincrofasorial en tiempo real. El lenguaje que maneja WPF para la creación de la interfaz gráfica es XAML.

Para aprovechar todas las características que ofrece WPF se implementó el programa basado en el patrón de arquitectura MVVM. Este modelo permite un aislamiento limpio entre la interfaz gráfica y el código detrás de la misma. Bajo esta arquitectura cada documento XAML cuenta con un modelo de vista (*ViewModel*), una clase que se encarga de actualizar la interfaz a través de un enlace (*Binding*) entre los valores de los ítems (Label, TextBox, Graficas, etc.) y las variables que estos deben mostrar. Es decir, cuando cambia el valor de una variable durante la ejecución de la aplicación este cambio se ve reflejado de manera automática en la interfaz.

En la figura 9 se aprecia la interfaz diseñada para la configuración del PDC. Con respecto a la configuración realizada en el cliente como aplicación de consola se agregaron dos campos funcionales adicionales: PMU ID y PMU NAME. Estos dos parámetros se utilizan para validar el origen de los datos, si el nombre o el ID de la PMU no coinciden con los ingresados por el usuario entonces no va a ser posible ejecutar el protocolo ya que el ID es necesario para dar un ID de destino a las tramas de comando. Adicionalmente, si al llegar la trama de configuración desde la PMU el nombre no coincide con el ingresado por el usuario, el protocolo no se va a terminar de ejecutar. El uso de los campos adicionales que aún no son funcionales, como Transport Protocol, se explicará en el capítulo 8, Trabajo a Futuro.

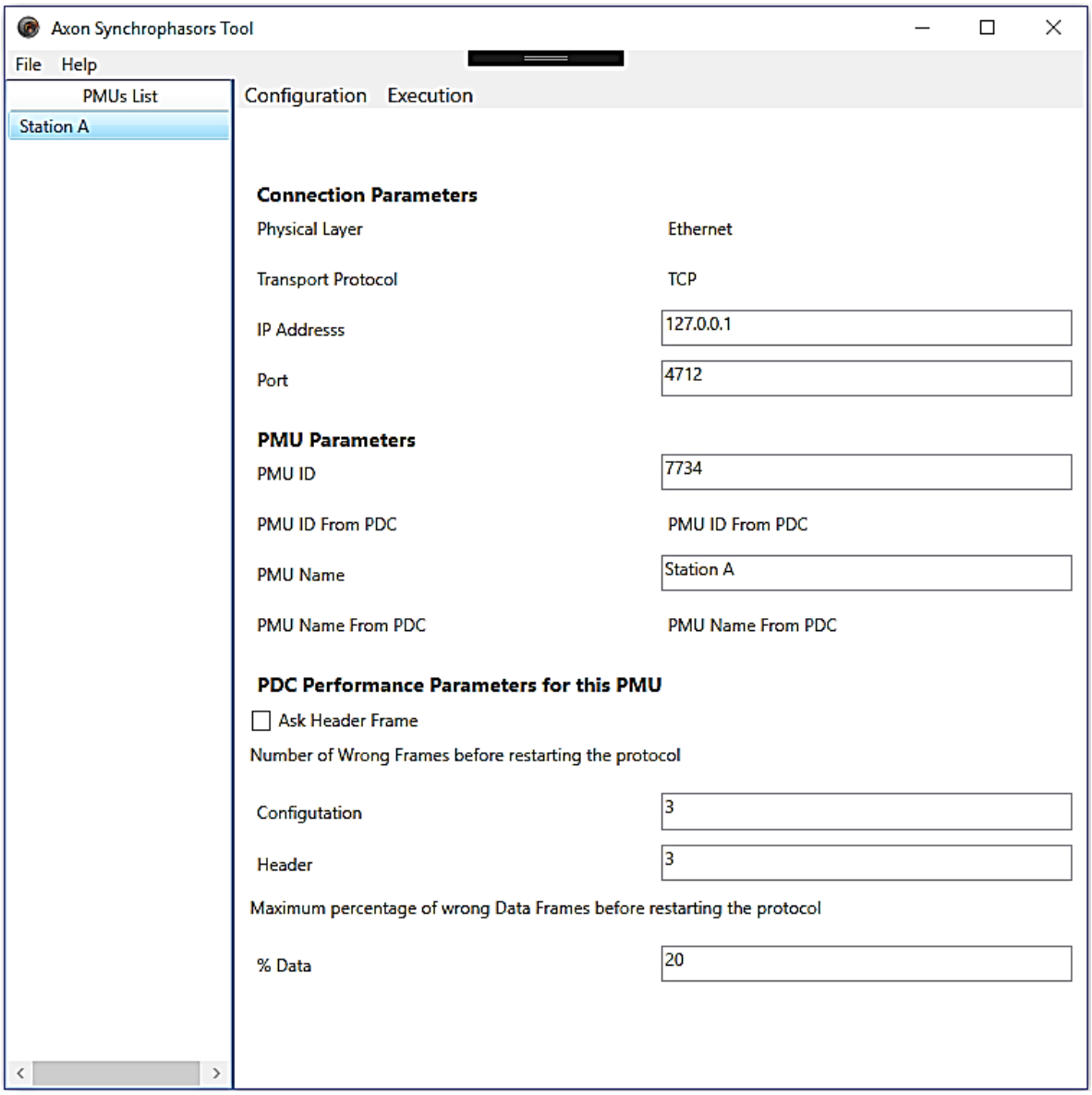

Figura 9. Interfaz para la configuración del PDC.

La interfaz diseñada para la ventana de ejecución general del protocolo se muestra en la figura 10. En esta se observan los botones que permiten iniciar o detener el protocolo de comunicación con la PMU. Además se aprecia la información general de ejecución como el nombre de la PMU, la información de cabecera, el ID Code, los valores analógicos, y en la parte de abajo los mensajes de sincrofasores enviados y recibidos en formato hexadecimal.

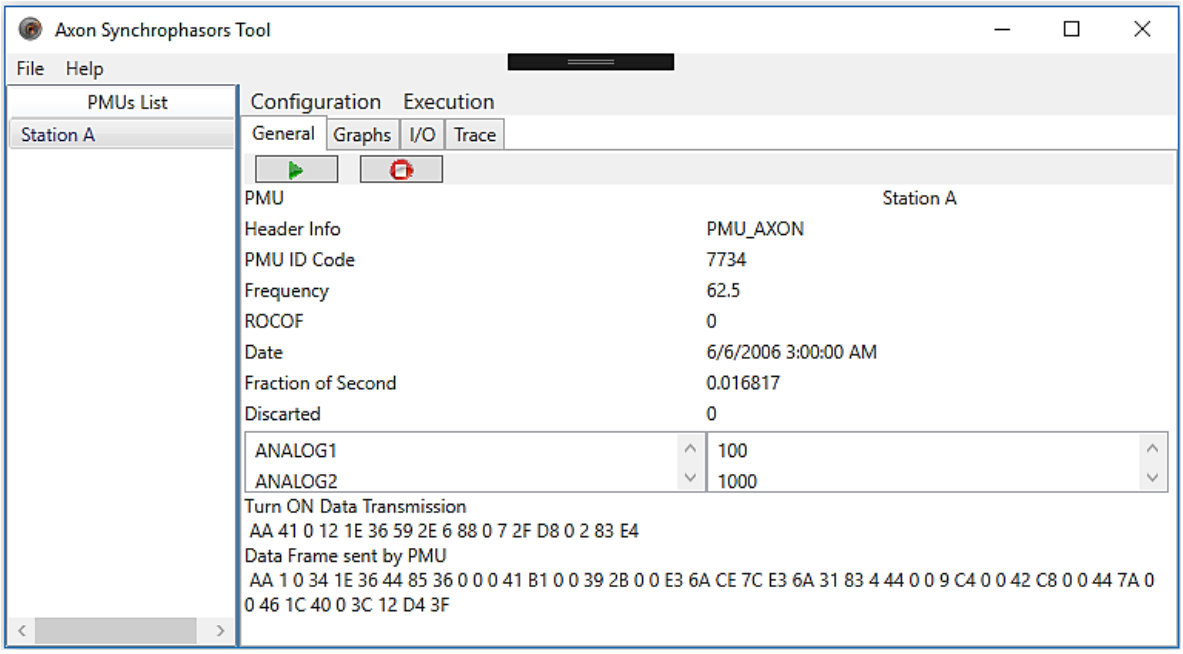

Figura 10. Interfaz General de Ejecución.

El contenido de la pestaña *Graphs* se muestra en la figura 11. Como su nombre lo indica en esta pestaña se muestran los gráficos de los fasores escogidos por el usuario y de la frecuencia.

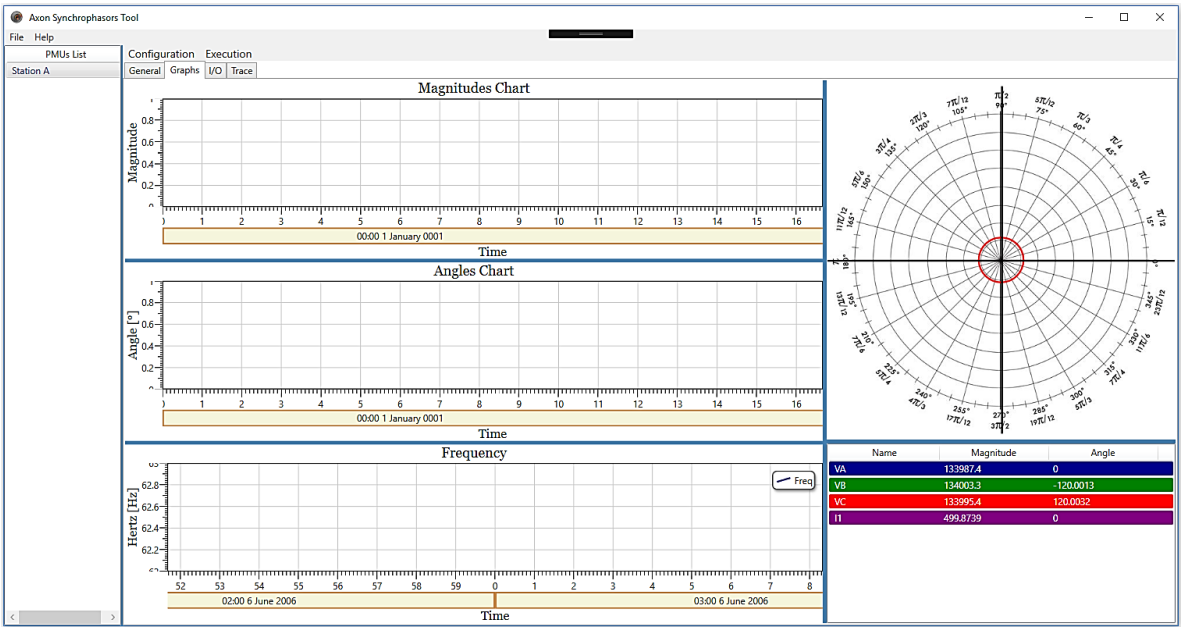

Figura 11. Interfaz para las gráficas dinámicas.

En esta interfaz se pueden identificar dos tipos de gráfica: las gráficas de tendencia y la carta de vectores. El usuario puede seleccionar por cada fasor si desea ver sus graficas de tendencia (magnitud y ángulo), su representación vectorial, o ambas y añadir o remover gráficas durante la ejecución del programa. La frecuencia se grafica automáticamente.

Adicionalmente, la carta de vectores no solo representa en ángulo sino que la longitud de los mismos es proporcional a su magnitud normalizada con respecto a la mayor magnitud entre los vectores graficados.

6.2. PDC COMO CLIENTE DE MULTIPLES PMU

Se puede hablar de una estructura jerárquica de PDCs que sigue la jerarquía de los sistemas de potencia: subestaciones, empresas de energía, áreas de control regional y nivel de interconexión. En este contexto, el fin último de un concentrador de datos de sincrofasores es recibir la información proveniente de múltiples servidores, PMUs o PDCs de más bajo nivel, y enviar esta información en un solo flujo de datos a un PDC de más alto nivel.

Luego de tener un cliente funcional para una PMU el paso a seguir fue desarrollar una arquitectura multiproceso que permitiera atender varios servidores de manera completamente independiente. Para este fin se utilizó el conjunto de librerías de Akka.

Para cada PMU debe existir una instancia de su configuración y su ejecución y estas deben ser completamente independientes con respecto a las de otra PMU, la comunicación y el envío de mensajes entre un servidor y el cliente debe ser transparente a los demás procesos. Para lograr esto fue necesario implementar el modelo de actores ofrecido por Akka, bajo este modelo se pueden crear procesos completamente independientes bajo demanda del usuario. Por cada PMU se crearán dos actores, un actor al que se llamará actor de host y otro al que se llamara actor de proceso, cada actor de host va a disparar un proceso y se va a comunicar con su actor intercambiando mensajes de tipos predefinidos.

En este caso, el usuario puede añadir varias PMU, cada una con sus respectivos parámetros de configuración. Cuando el usuario inicia el protocolo para un servidor, se crea un actor de host que se va a comunicar con el actor del proceso que ejecuta el protocolo. El actor de host se encuentra del lado más cercano al usuario, es decir, es el encargado de pasar a la aplicación WPF las instancias que recibe desde el proceso. El actor de proceso recibe la información necesaria para poder iniciar el proceso. En resumen, el actor de host pasa al actor de proceso un mensaje con la información de configuración de la PMU proporcionada por el usuario y el actor de proceso envía al actor de host los datos obtenidos desde la PMU al ejecutar el protocolo. Cada proceso crea un socket para la comunicación con su respectiva PMU, de esta manera cada PMU se va a trabajar con un proceso completamente independiente desde la interfaz gráfica hasta el socket TCP.

Para agregar una nueva PMU el usuario va a la pestaña *File – NewPMU* y al dar clic se despliega la pestaña emergente mostrada en la figura 12, el nombre ingresado por el usuario va a ser el que aparezca en la lista de PMUs mostrada en las figuras 9-11. No es necesario que este nombre coincida con el nombre del equipo, sin embargo el campo PMU Name de la ventana de configuración sí.

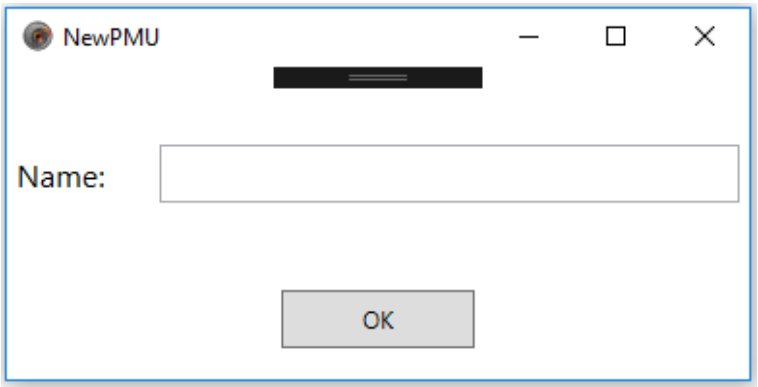

Figura 12. Ventana para agregar una nueva PMU.

Al dar clic en el nombre de la PMU en la lista, aparece la ventana de configuración y la de ejecución, por cada PMU que se agregue se crea una instancia diferente de estas dos interfaces. Durante la ejecución del PDC se pueden agregar o remover servidores según las necesidades del usuario sin interrumpir la ejecución del protocolo en los que aún estén transmitiendo información. Durante la etapa de pruebas se comprueba el correcto funcionamiento del software.

6.3. PRUEBAS DEL PDC CON EL RELÉ SEL 451

El software que permite configurar los servicios del relé es el *AcSELerator QuickSet*. El relé cuenta con un archivo de configuración que puede ser leído desde el equipo, modificado en un computador a través del software y cargado nuevamente en el equipo. Esta estrategia se usa para no modificar la configuración de los demás servicios que estén corriendo en el equipo sino solo, en este caso, activar y configurar el servicio de sincrofasores.

En la figura 13 se muestra la pantalla de configuración de parámetros de comunicación del servicio de sincrofasores del relé. Los principales parámetros configurados fueron:

- Habilitar Sincrofasores sobre Ethernet: Sí.
- Protocolo de Transporte: TCP.
- Dirección IP del equipo al que se enviarán los datos (PDC): 192.168.0.222.
- Puerto del servicio (PDC): 4712.

Con estos parámetros configurados es posible establecer la conexión entre el PDC y la PMU. Posteriormente se debe realizar la configuración de los parámetros de funcionamiento del módulo de sincrofasores del relé. La ventana de configuración de dichos parámetros de muestra en las figuras 14 y 15.

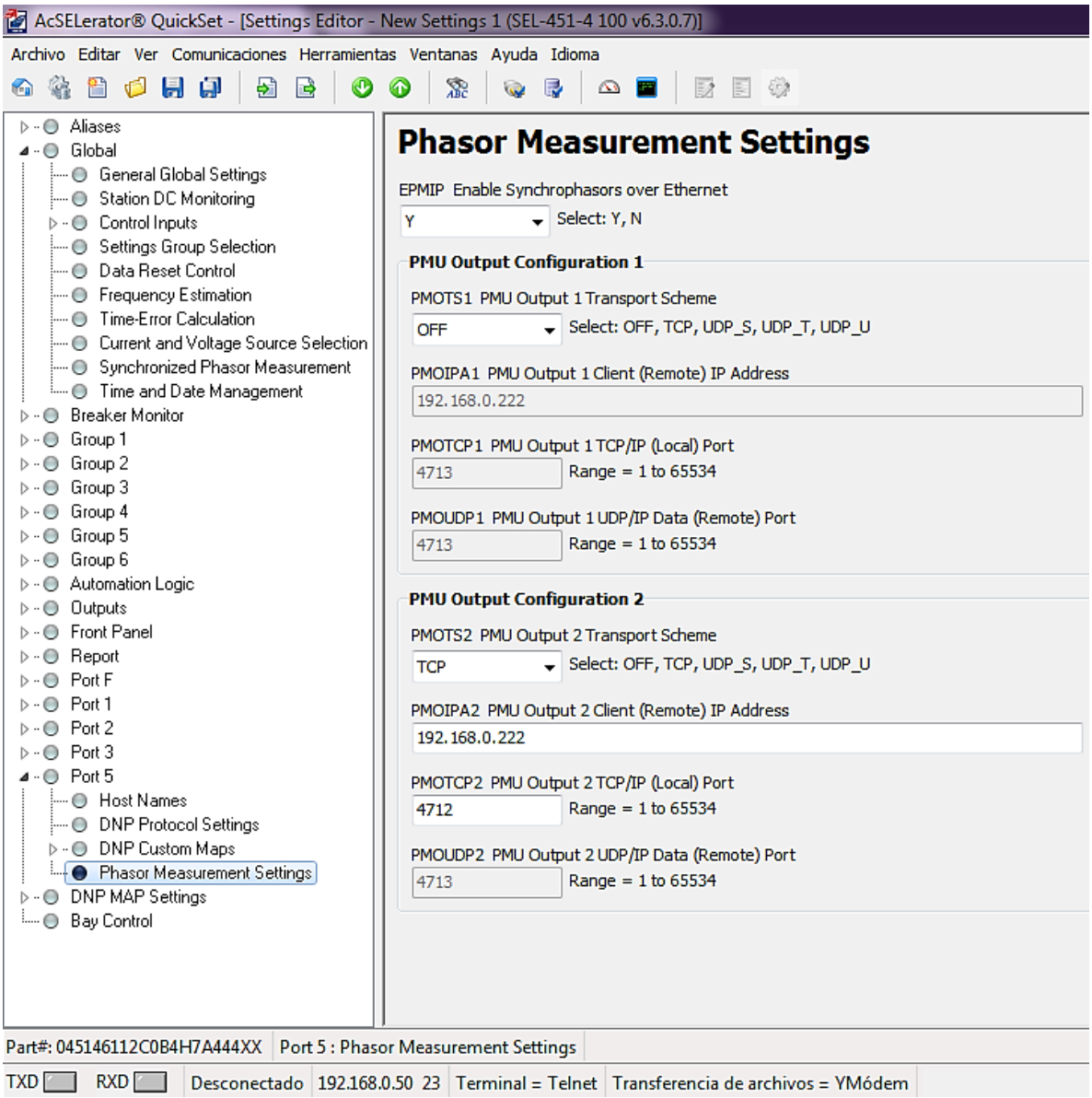

Figura 13. Configuración de parámetros de comunicación del servicio de sincrofasores del Relé SEL 451.

Los principales parámetros de funcionamiento configurados en el equipo fueron:

- Habilitar PMU: Sí.
- Formato de los mensajes (Protocolo): C37.118.
- Tasa de reporte: 30 Mensajes por segundo.
- Nombre de la PMU: PMU SEL.
- ID Code de la PMU: 1921.
- Fasores de Voltaje a Transmitir: Todos. Fasor de voltaje de secuencia positiva (V1) y voltajes de fase VA, VB y VB.
- Fasores de Corriente a Transmitir: Corriente de secuencia positiva (I1).
- Representación numérica de los Fasores: Entero de 16 bits.

- Formato de los Fasores: Coordenadas Rectangulares. Parte real y parte imaginaria.
- Representación numérica de la Frecuencia: Entero de 16 bits.
- Número de valores analógicos: 1.

Al terminar de establecer los parámetros de configuración del equipo se debe subir el archivo de configuración modificado al relé para que el servicio de sincrofasores quede activo con la configuración establecida.

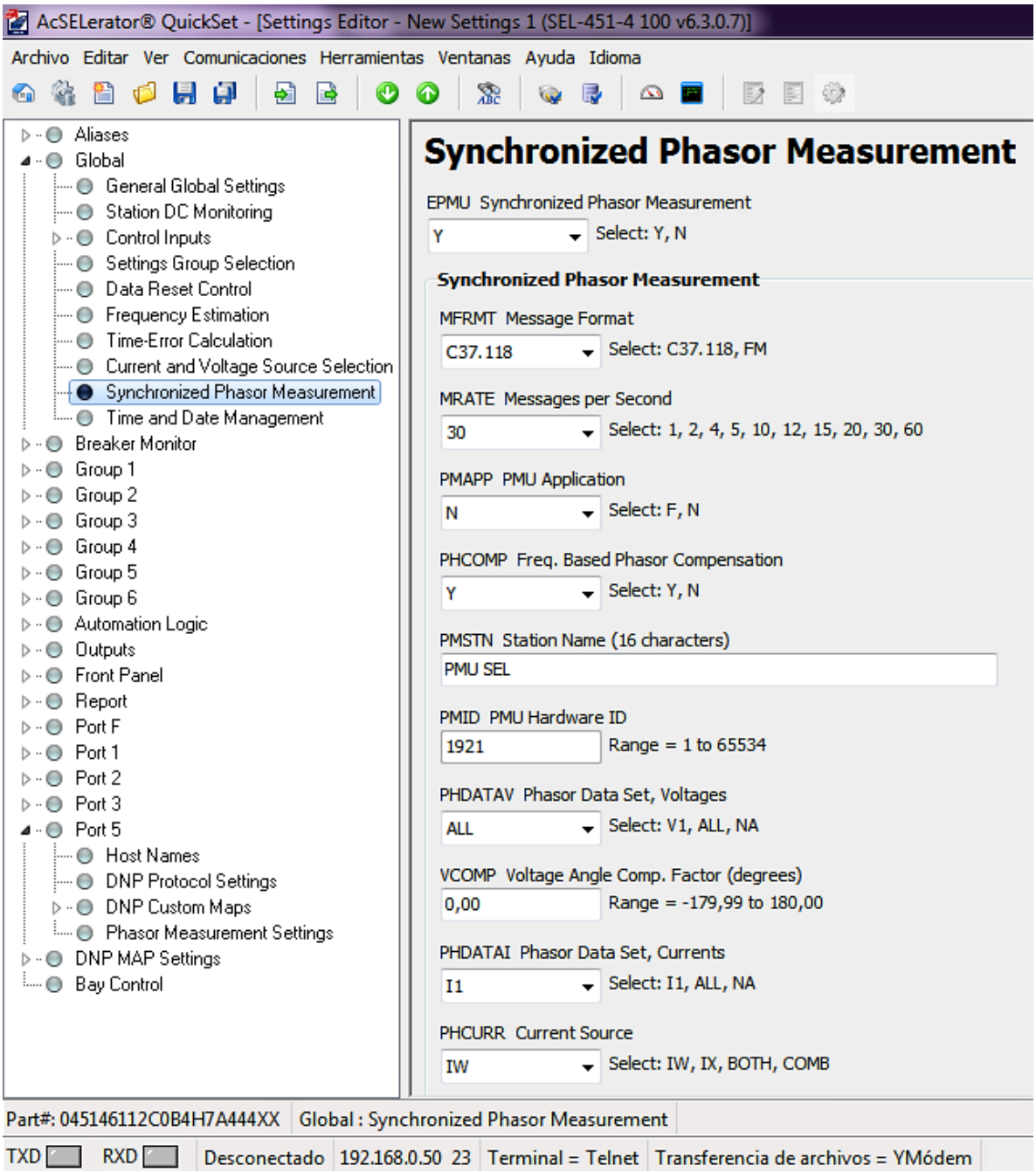

Figura 14. Configuración de los parámetros de funcionamiento del módulo de sincrofasores del relé SEL 451_1.

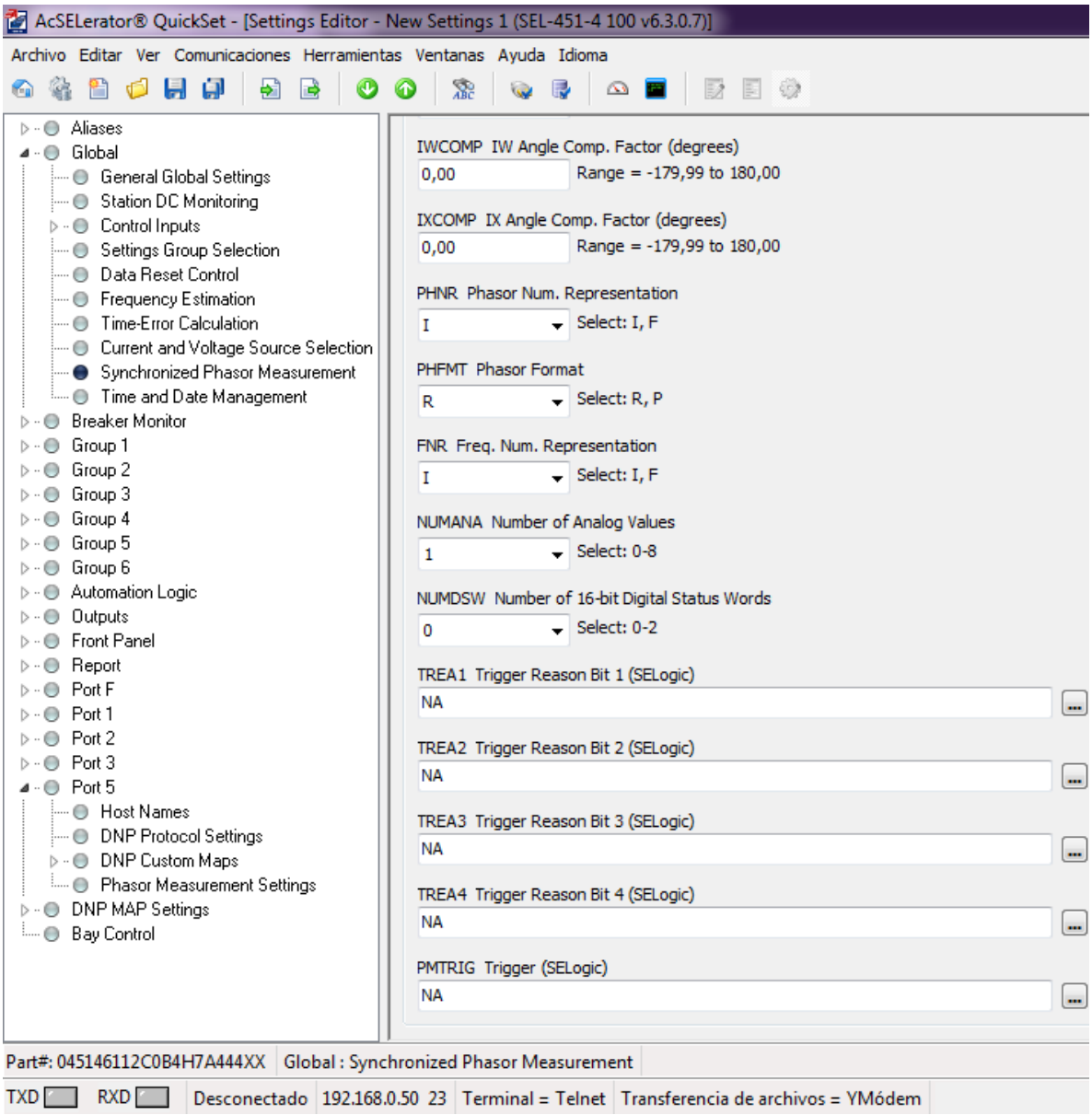

Figura 15.Configuración de los parámetros de funcionamiento del módulo de sincrofasores del relé SEL 451_2.

Para esta prueba el terminal VB del relé estaba midiendo el voltaje de la red eléctrica. Ya que en una subestación eléctrica los terminales del relé irían conectados a un transformador de corriente, existe un factor de conversión que multiplica el voltaje medido por el factor de reducción de voltaje del transformador. Por esta razón el dato de magnitud del voltaje VB entregado por el relé es mayor al voltaje de la red.

Una vez configurado el equipo se procede a ejecutar y configurar el PDC. Se añaden dos PMUs cuyos nombres son SEL 451 y HERCULES para identificar el equipo y el simulador. En las figuras 16 y 17 se aprecia la configuración para cada servidor.

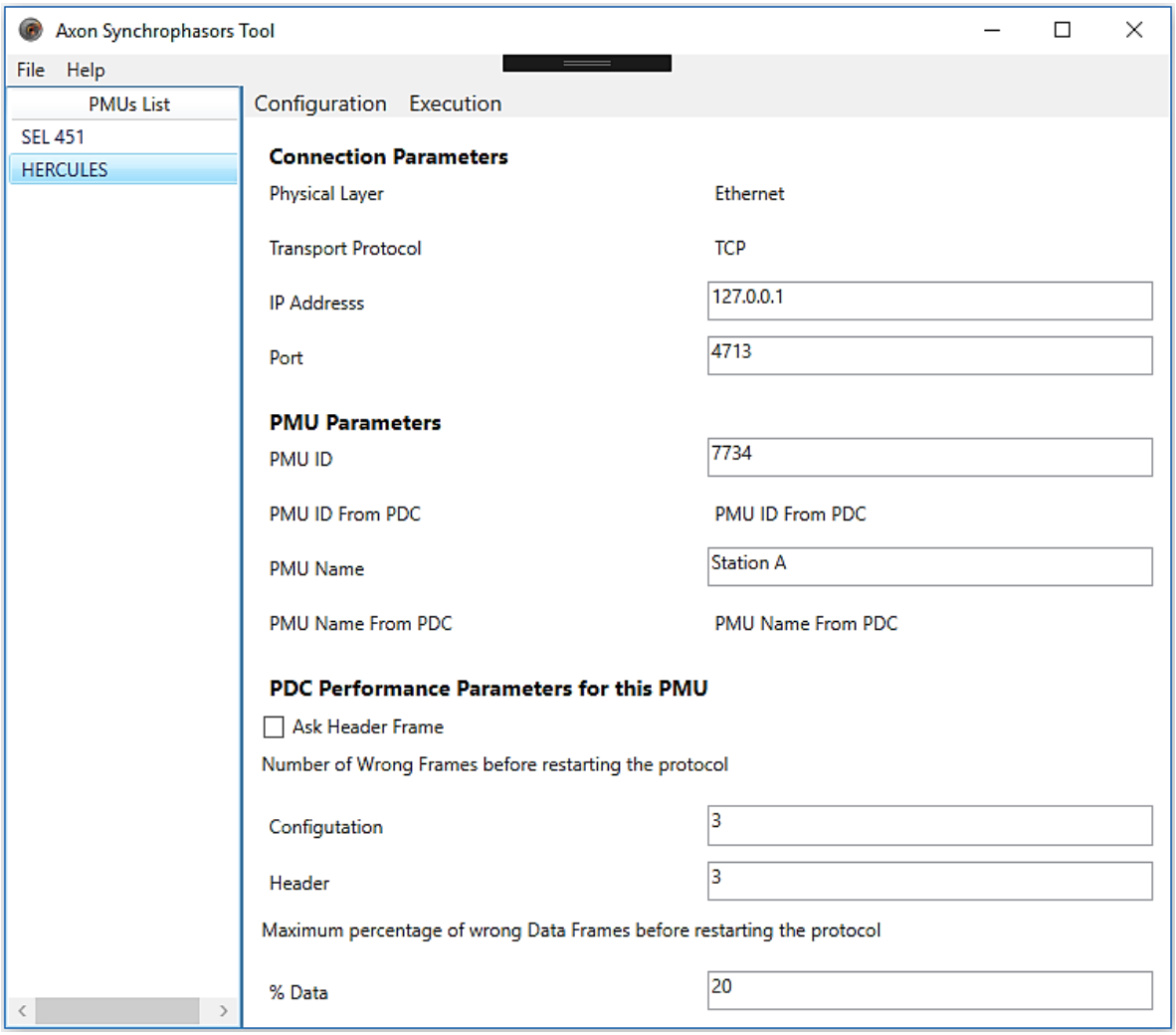

Figura 16. Parámetros de configuración del PDC para el simulador Hercules.

Los parámetros de configuración del PDC para establecer comunicación con el simulador y ejecutar el protocolo son:

- Dirección IP: 127.0.0.1, localhost.
- Puerto: 4713.
- ID Code de la PMU: 7734.
- Nombre de la PMU: Station A.
- Número máximo de pedidos de retransmisión de la trama de Configuración: 3.
- Número máximo de pedidos de retransmisión de la trama de Cabecera: 3.
- Porcentaje máximo de mensajes de datos erróneos: 20%.

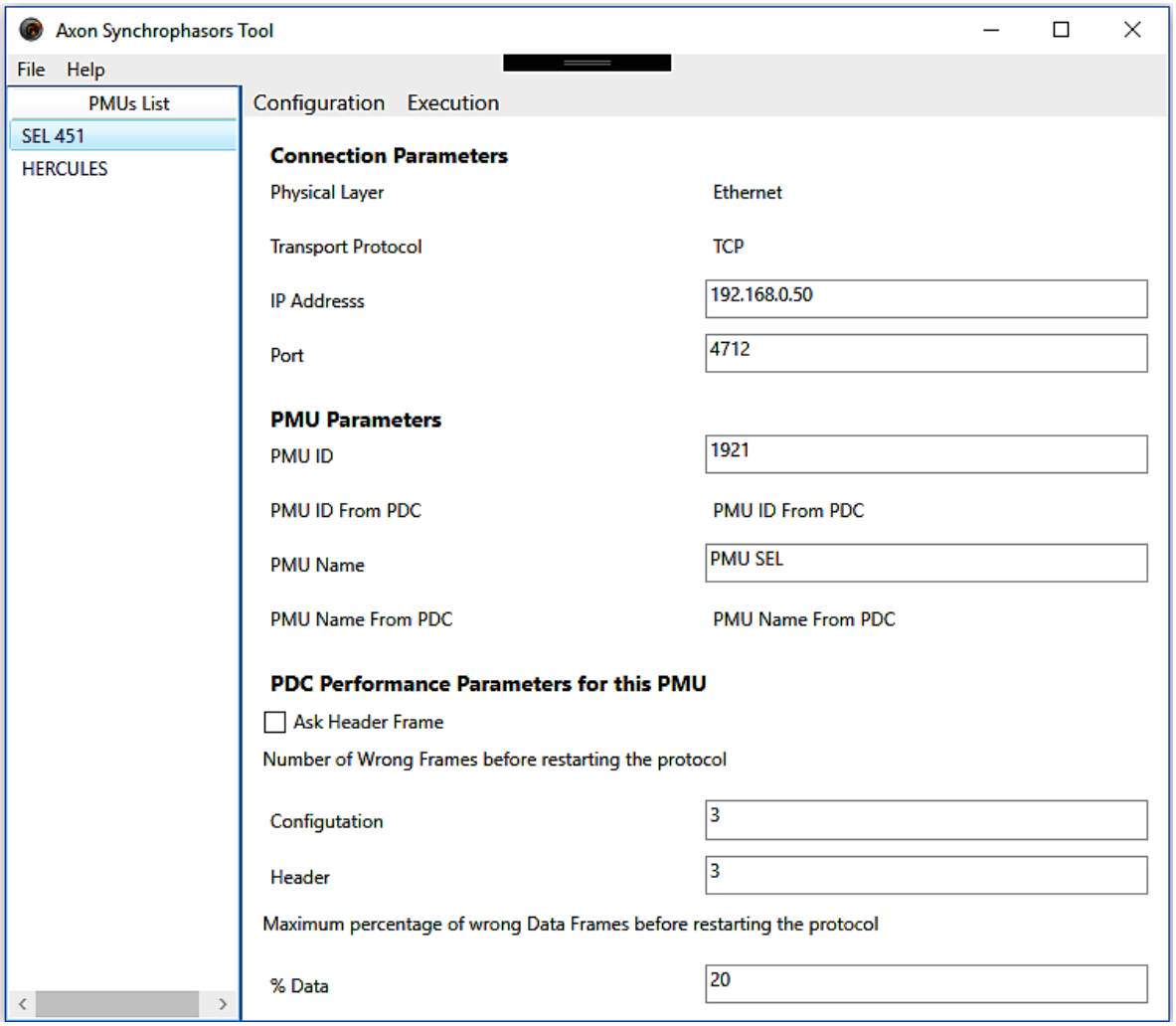

Figura 17. Parámetros de configuración del PDC para el Relé SEL 451.

Los parámetros de configuración del PDC para establecer comunicación con el relé y ejecutar el protocolo son:

- Dirección IP: 192.168.0.50. Dirección IP del equipo.
- Puerto: 4712.
- ID Code de la PMU: 1921. Igual que el configurado en el equipo.
- Nombre de la PMU: PMU SEL. Igual que el configurado en el equipo.
- Número máximo de pedidos de retransmisión de la trama de Configuración: 3.
- Número máximo de pedidos de retransmisión de la trama de Cabecera: 3.
- Porcentaje máximo de mensajes de datos erróneos: 20%.

En las figuras 18 y 19 se ven las ventanas de ejecución general para cada PMU. Los dos procesos se ejecutaron de manera independiente y están corriendo al mismo tiempo. El nombre de la PMU y su ID Code, en ambos casos, coinciden con los configurados.

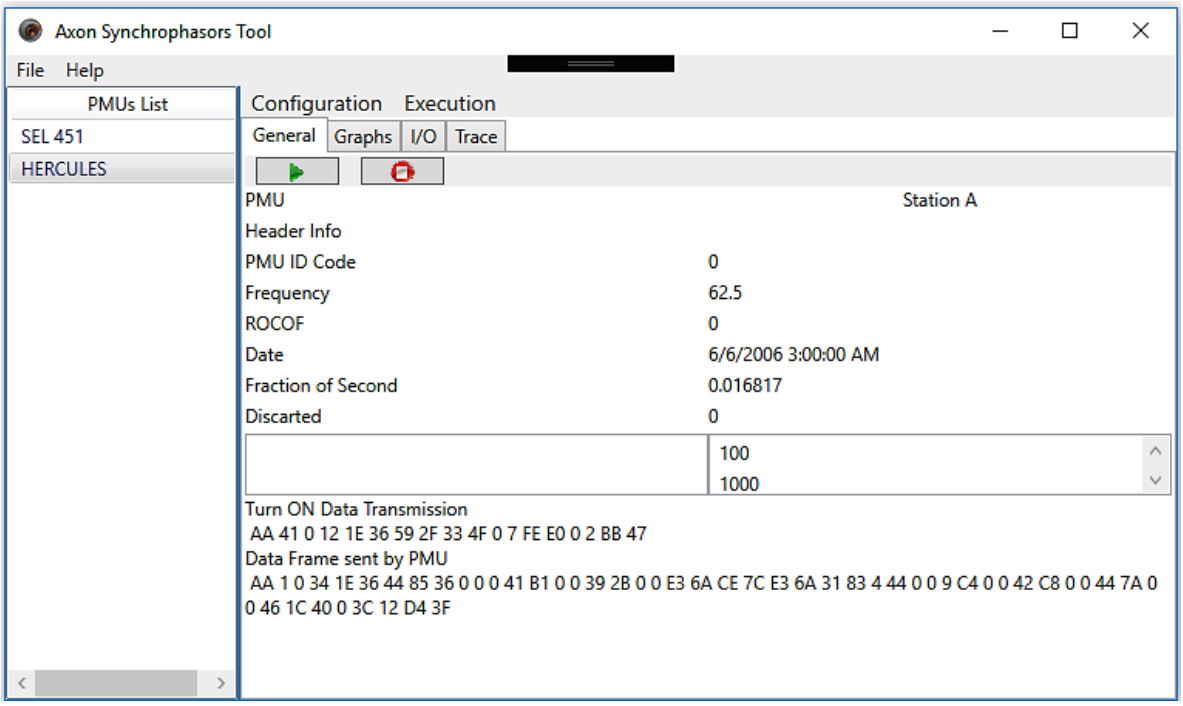

Figura 18. Ventana general de ejecución del protocolo para el simulador Hercules.

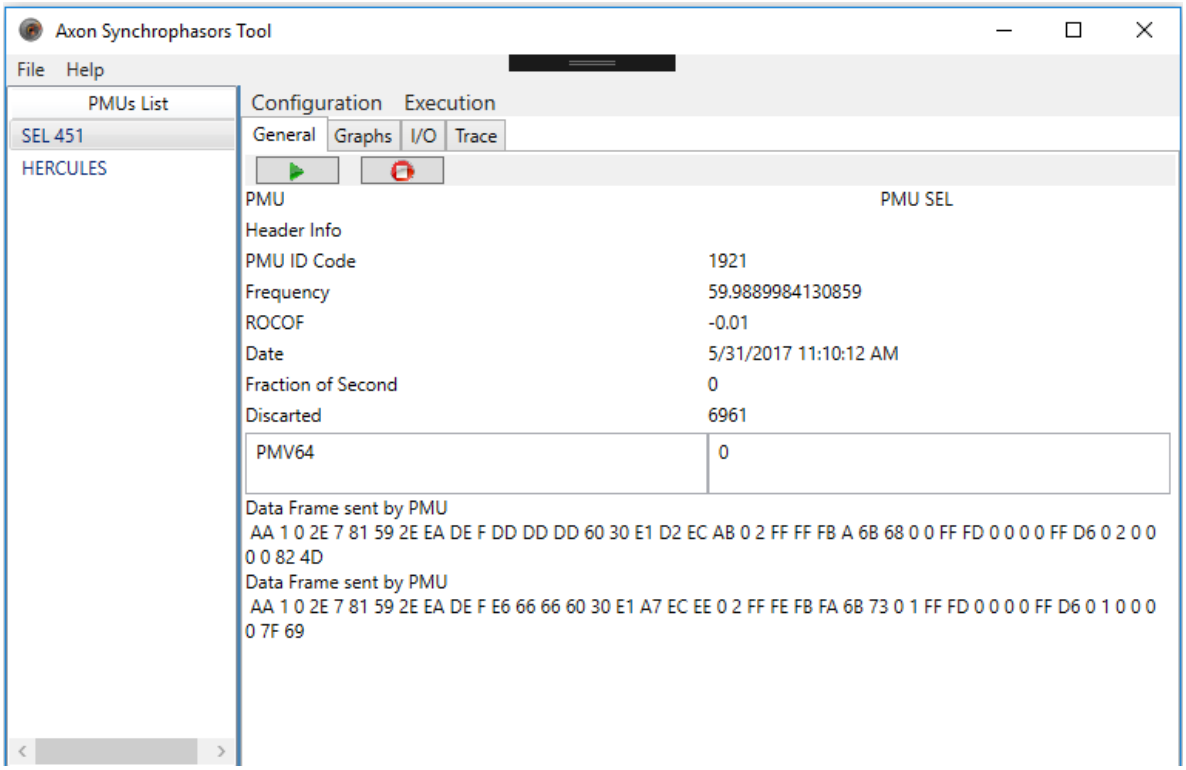

Figura 19. Ventana general de ejecución del protocolo para el Relé SEL 451.

En las figuras 20 y 21 se aprecian las ventanas de gráficas dinámicas para cada PMU. En el caso del simulador en la figura 20, se grafican los fasores VA y VB. Para poder graficar las líneas se creó una segunda trama de datos con una estampa de tiempo diferente pero, los demás valores se dejaron igual por los cual en las tres graficas de tendencia se aprecian líneas rectas. En la carta de vectores se graficaron también VA y VB. Ya que ambos fasores tienen la misma magnitud su longitud es igual, VA se encuentra en 0° y VB a -120°. La escala de las gráficas de tendencia se ajusta de manera automática de tal forma que se puedan visualizar todas las gráficas escogidas por el usuario. Los colores de las líneas de cada fasor vienen predefinidos de tal manera que exista un buen contraste entre las gráficas y el fondo.

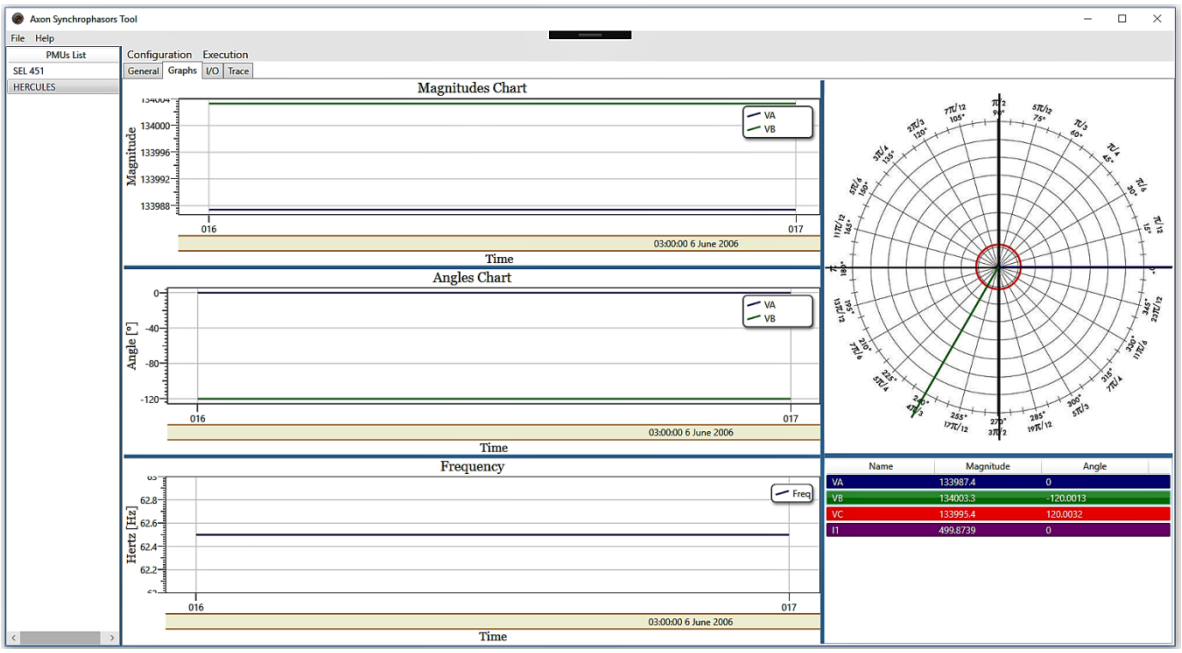

Figura 20. Graficas dinámicas para el simulador Hercules.

La imagen de la figura 21 muestra las gráficas de todos los fasores, cuatro de voltaje y uno de corriente, entregados por el equipo. Ya que la librería Dynamic Data Display autoajusta la escala para que se puedan ver todas las señales, las gráficas de los fasores cuya magnitud es casi cero se solapan. En la gráfica de fases si se pueden apreciar las tendencias de las fases de todos los fasores. En la gráfica de la parte baja de la imagen se ve como la frecuencia varía entre 50.94 Hz y 60.08 Hz. En la carta de vectores solo se ven los vectores correspondientes a V1 (Voltaje de secuencia positiva) y VB (Voltaje en la fase B) ya que sus magnitudes son muy grandes en comparación con las magnitudes de los demás vectores. Se ve como la longitud de V1 es aproximadamente la tercera parte de la longitud de VB, esto se debe a que la magnitud de V1 oscila alrededor de 7400 V mientras que la de VB alrededor de 22000 V.

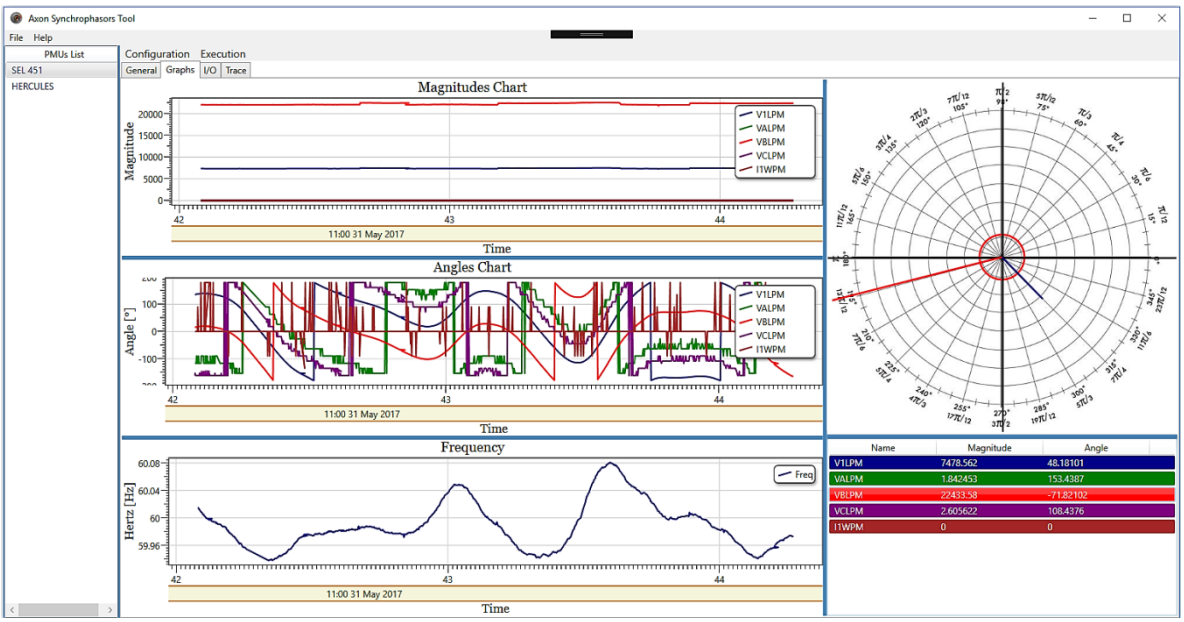

Figura 21. Graficas dinámicas de todos los fasores para el Relé SEL 451.

En la figura 22 se muestran las gráficas de tendencia y la representación vectorial para V1 y VB. En las gráficas de fase se aprecian variaciones aparentemente bruscas pero, esto se debe a que en el protocolo para sincrofasores los ángulos de los fasores deben representarse en valores de -180° a 180°. Se aprecia como la magnitud de VB se mantiene estable alrededor de 22500 V y la de V1 alrededor de 7500 V. Adicionalmente, se puede ver que entre estos dos fasores se mantiene un desfase estable de 120°.

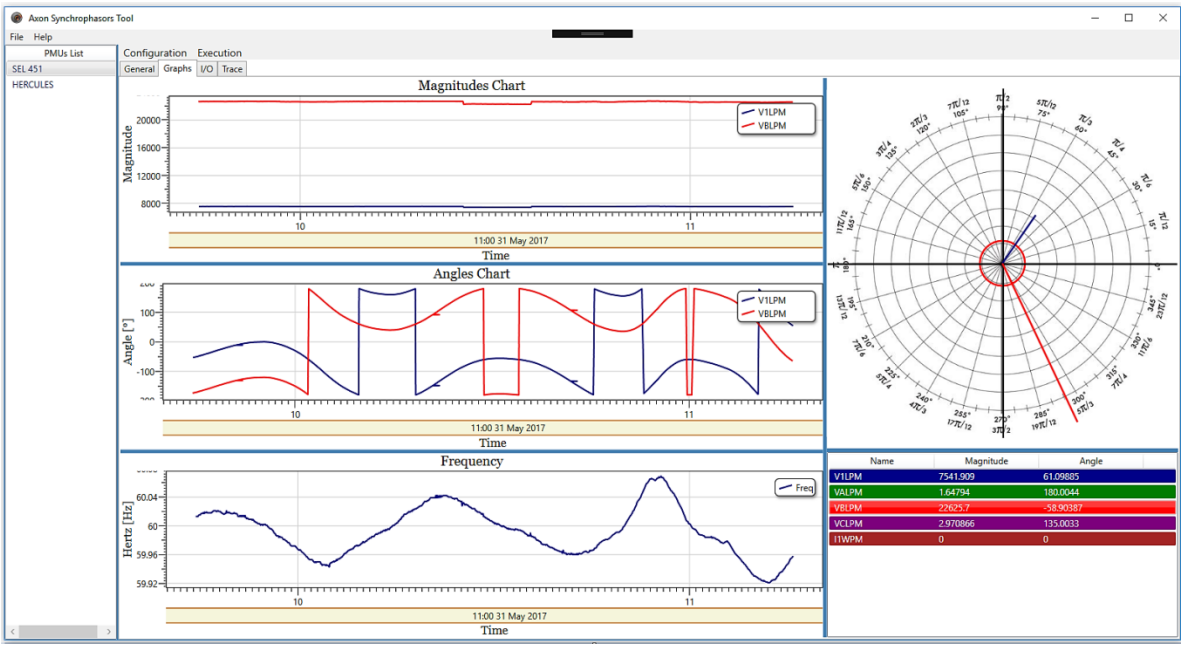

Figura 22. Graficas dinámicas de los fasores de voltaje de secuencia positiva y de la fase B.

En la figura 23 se pueden apreciar las gráficas de los fasores de voltaje que la PMU calcula a partir de las terminales del relé que no tienen nada conectado, VA y VC. En otras palabras, en la figura 23 podemos apreciar el ruido que generan las terminales libres del relé por eso su magnitud es tan baja, entre 0.5 V y 3 V. Hay que tener en cuenta que estos valores vienen multiplicados por un factor de conversión ya que en funcionamiento normal el relé toma las medidas de señales entregadas por transformadores de voltaje. Esta es la razón por la cual en una terminal vacía aparece un voltaje de hasta 3 V. En la carta de vectores es posible ver la representación para estas dos señales ya que su magnitud es comparable y no hay ningún otro vector graficado.

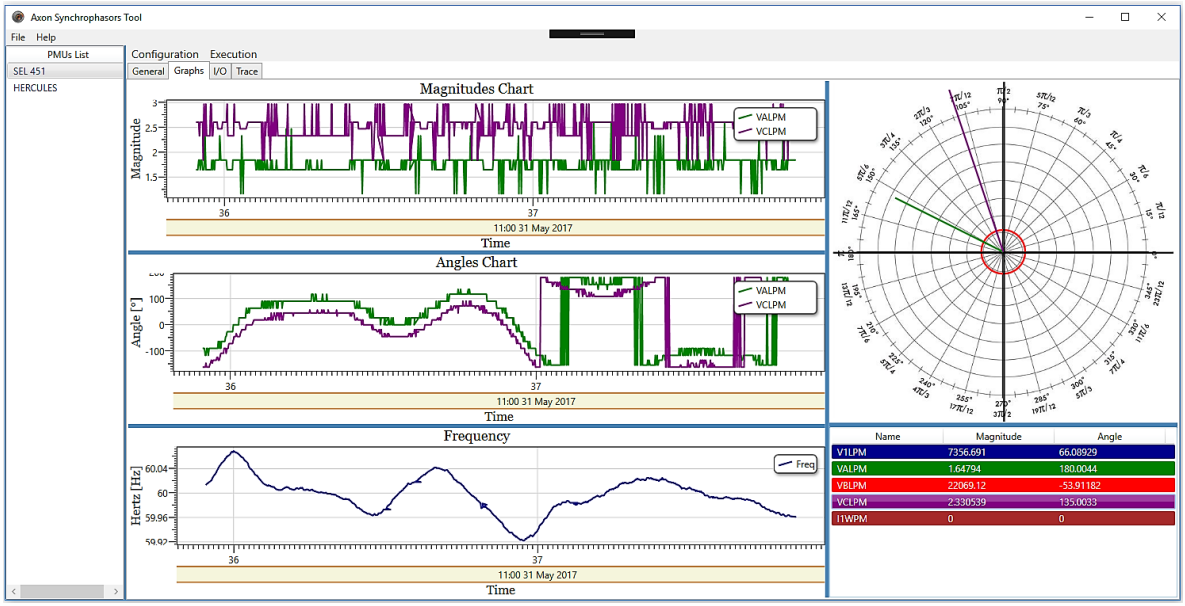

Figura 23. Graficas dinámicas de los fasores de voltaje de las fases A y C.

Para comprobar que efectivamente el PDC está realizando una interpretación correcta de las tramas entregadas por el equipo se va a utilizar Wireshark. El filtro synphasor de este sniffer captura las tramas correspondientes al protocolo de sincrofasores y las decodifica, de esta manera se puede revisar que Wireshark y el PDC estén interpretando la información de la misma manera. En la figura 24 se aprecian las tramas capturadas de la comunicación entre el PDC y el Hercules, ya que estas son las tramas de ejemplo de la norma, se sabe que el PDC si las interpreto bien sin necesidad de explorar a fondo en Wireshark.

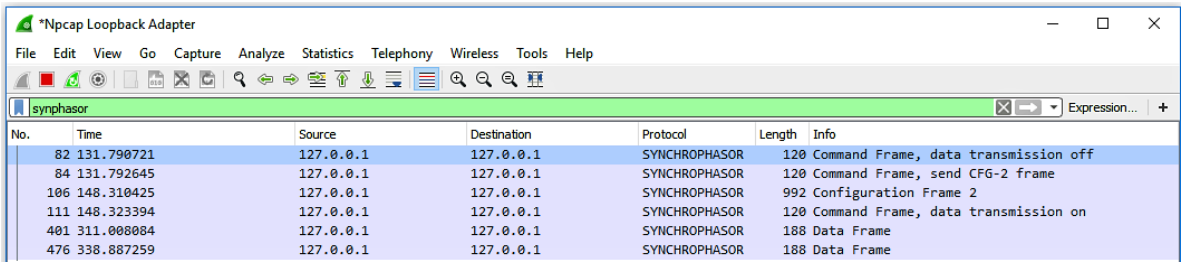

Figura 24. Tramas capturadas por Wireshark de la comunicación entre el PDC y el Simulador.

En la figura 25 se puede apreciar la información de la trama de configuración enviada por el equipo y decodificada por Wireshark. El ID y el nombre de la PMU interpretados por Wireshark coinciden con los configurados en el equipo. Además se ve que el sniffer identifica 5 fasores, 1 valor analógico, una frecuencia nominal de 60 Hz y una tasa de reporte de 30 tramas por segundo, todo según lo esperado e interpretado por el PDC.

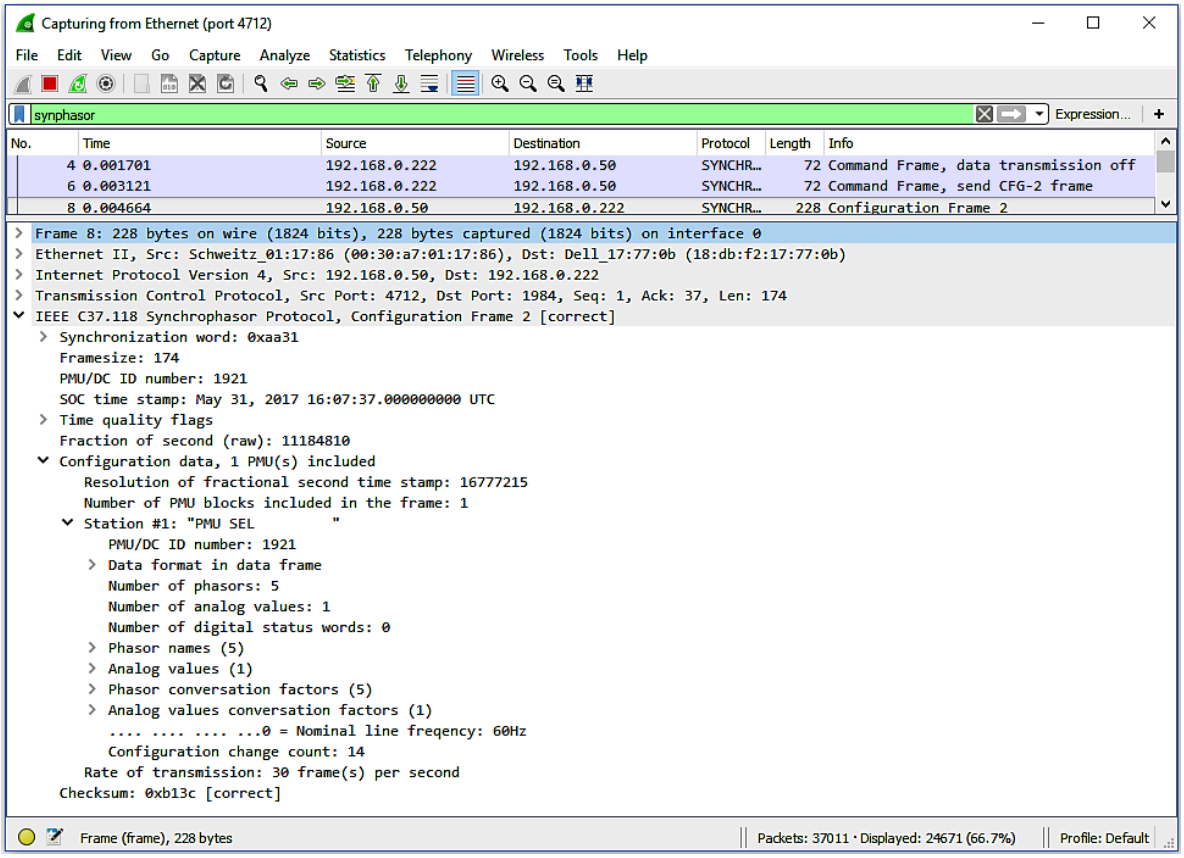

Figura 25. Mensaje de Configuración enviado desde el Relé capturado por Wireshark.

Finalmente, para comprobar que los datos entregados por la PMU si son los que el PDC muestra en pantalla, se revisa la interpretación de Wireshark a una de las tramas de datos. En la figura 26 se aprecia que los valores de magnitud y ángulo de los fasores coinciden con los valores mostrados en las figuras 21, 22 y 23.

Los resultados de estas pruebas demuestran que el concentrador de datos de sincrofasores desarrollado en este proyecto puede enviar, recibir, e interpretar datos de más de una unidad de medición sincrofasorial. Adicionalmente, gracias a la interfaz gráfica de usuario diseñada, el PDC se convierte en una herramienta versátil y de fácil uso. Este software permite al usuario visualizar, añadir o remover gráficas dinámicas de tendencia en tiempo real, ver el comportamiento vectorial de los fasores, revisar las variaciones en la frecuencia de la red eléctrica, todo esto para varios servidores de manera concurrente.

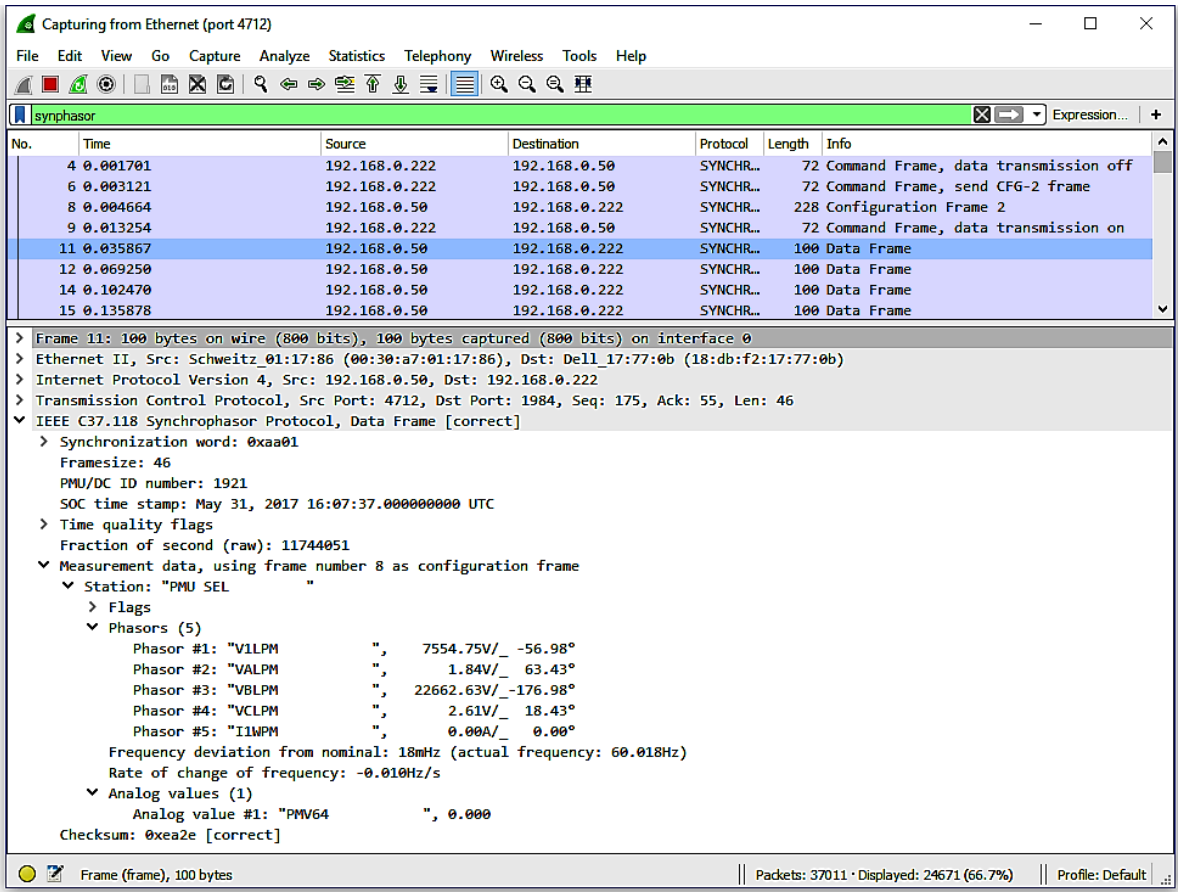

Figura 26. Mensaje de Datos enviado desde el Relé capturado por Wireshark.

7. TRABAJO A FUTURO

A continuación se nombran los pasos a seguir en el desarrollo del concentrador de datos de sincrofasores para que sea un producto comercializable de la compañía Axon Group, que le permita entrar a competir en el mercado de la tecnología sincrofasorial:

- Desarrollo del PDC como servidor. El software debe poder concentrar los datos de sincrofasores provenientes de diferentes fuentes y generar los mensajes de configuración, cabecera y datos que contengan toda esta información para enviarlos en un solo flujo de datos a PDCs de más alto nivel.

- Implementación de una base de datos para almacenar los datos de sincrofasores. Esta información puede utilizarse para el análisis post-falla de sistemas eléctricos.

- Realizar pruebas de estabilidad del software.

Se espera que al finalizar el proyecto, la arquitectura del PDC sea la mostrada en la figura 27.

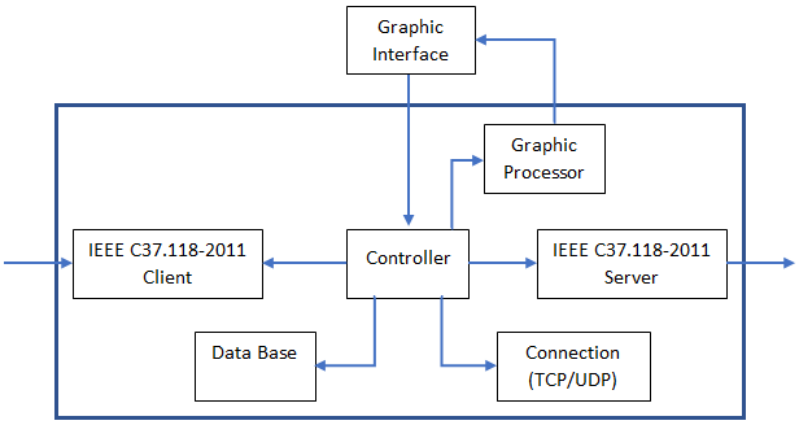

Figura 27. Arquitectura a futuro del PDC.

9. CONCLUSIONES

Un concentrador de datos de sincrofasores permite monitorear, almacenar y transmitir la información proveniente de múltiples PMUs en tiempo real. Debido a las múltiples aplicaciones de la tecnología sincrofasorial en el sector eléctrico, la implementación de redes de sincrofasores en Colombia va en aumento por lo cual se hace necesario desarrollar PDCs capaces de manejar grandes cantidades de información. En este contexto, el desarrollo de este proyecto abre las puertas de un mercado en expansión a la compañía, por lo que los objetivos alcanzados hasta el momento representan un avance muy importante para los propósitos de la misma. Al finalizar el proyecto se espera contar con una aplicación de fácil uso y de alta calidad que permita aprovechar al máximo las ventajas ofrecidas por la tecnología sincrofasorial para el control y supervisión de redes eléctricas.

BIBLIOGRAFÍA

[1] "Chapter 7. Boost.Asio - 1.63.0". *Boost.org*. Web:

[2] "Introducción (WPF)". *Msdn.microsoft.com*, 2017. Web.

[3] "What Is XAML?". *Msdn.microsoft.com*, 2017. Web.

[4] "The MVVM Pattern". *Msdn.microsoft.com*, 2017. Web.

[5] "Npcap Es La Nueva Librería Para Windows Que Nos Permite Sniffing De Paquetes". *RedesZone*, 2017. Web.

[6] "Introduction To Akka • Akka Documentation". *Doc.akka.io*, 2017. Web.

[7] "Introducción A Akka Para Java Con Ejemplos | Adictosaltrabajo". *Adictosaltrabajo.com*, 2017. Web.

[8] IEEE Standards Association: IEEE Standard for Synchrophasor Data Transfer for Power Systems - IEEE Std C37.118.2, 2011

[9] "SEL 451". *selinc.com*, 2017. Web.

[10] "Dynamic Data Display". *CodePlex*, 2017. Web.