EVALUACIÓN DE LA TRANSMISIÓN DE DATOS RESPECTO A LA NORMA IEC 61850 EN UNA RED DE DATOS LOCAL Y DE SU DESEMPEÑO ANTE ATAQUES A SU SEGURIDAD

DAVID ANDRÉS CASTRO RUIZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FISICOMECÁNICAS

ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

BUCARAMANGA

2012

EVALUACIÓN DE LA TRANSMISIÓN DE DATOS RESPECTO A LA NORMA IEC 61850 EN UNA RED DE DATOS LOCAL Y DE SU DESEMPEÑO ANTE ATAQUES A SU SEGURIDAD

DAVID ANDRÉS CASTRO RUIZ

Trabajo de grado para optar por el título de Ingeniero de Sistemas

Director

Msc. Pedro Javier Trujillo Tarazona

Codirector

Ph.D. Gabriel Ordoñez Plata

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FISICOMECÁNICAS

ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

BUCARAMANGA

2012

Dedicatoria

A quien mas podría dedicarle este trabajo sino ha Dios y a mi familia los principales ejes de mi vida y los que me hacen ser la persona que soy hoy.

A mis padres que con su amor, compañía y sabiduría me dieron la energía para culminar este trabajo. Gracias al apoyo de ustedes estoy donde estoy y soy lo que soy.

A mi hermano, al cual quiero devolver las siguientes palabras que me gustaron mucho, hermano usted también es "mi primer y mejor amigo" su apoyo y compañía constante hicieron que pudiera culminar con este trabajo.

Le doy gracias a Dios por tener la familia que tengo, a mis padres y a mi hermano, me queda decirles, Los Quiero Mucho.

TABLA DE CONTENIDO

PÁG

LISTA DE FIGURAS

PÁG

LISTA DE TABLAS

PÁG

LISTA DE ANEXOS

ACRÓNIMOS

ACSI	Abstract Communication Service Interface
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules of ASN.1
CAM	Content Addressable Memory
GSE	Generic Substation Events
GOOSE	Generic Object Oriented Substation Event
IEC	International Electrotechnical Commission
IED	Dispositivo Electrónico Inteligente
PDU	Unidad de datos de protocolo
SAS	Sistema de Automatización de Subestaciones
TLV	Tag – Length – Value

RESUMEN

TÍTULO: EVALUACIÓN DE LA TRANSMISIÓN DE DATOS RESPECTO A LA NORMA IEC 61850 EN UNA RED DE DATOS LOCAL Y DE SU DESEMPEÑO ANTE ATAQUES A SU SEGURIDAD*

AUTOR: DAVID ANDRÉS CASTRO RUIZ**

PALABRAS CLAVE: IEC 61850, GOOSE, Modelo OSI, Mensajes rápidos, ataques a la seguridad

CONTENIDO:

La norma IEC 61850 define en su contenido una serie de protocolos de comunicación los cuales son utilizados para el envío de diferentes tipos de mensajes al interior de una subestación eléctrica automatizada. Uno de los protocolos es GOOSE; a través de él se envían mensajes denominados rápidos, que indican un cambio de estado al interior de la subestación y poseen una alta exigencia de tiempo de transferencia (entre 3 y 10 ms) debido a la importancia de los mensajes al interior de la subestación; gracias a esto, el protocolo utiliza las capas de control del Enlace de datos y Física del modelo OSI, con el fin de hacer más rápida la transmisión de mensajes.

En este documento se describe una implementación del protocolo GOOSE, siguiendo todas las exigencias propuestas en la norma IEC 61850, con el fin de analizar su comportamiento en una red de datos local, comprobar el cumplimiento de los tiempos de transferencia exigidos en la norma IEC 61850 y observar el desempeño de los mensajes rápidos a ciertos ataques a la seguridad.

Para llevar cabo el proceso analítico se realizaron una serie de experimentos en los que se modificaron variables como: el tipo de transmisión, el tipo de configuración de la red local, la cantidad de computadores productores de mensajes GOOSE y el tráfico de red, con el fin de observar el desempeño de la red de datos local ante esos cambios.

* Proyecto de grado

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática Director Msc. Pedro Javier Trujillo Tarazona. Codirector Dr. Gabriel Ordoñez Plata

ABSTRACT

TITLE: EVALUATION OF DATA TRANSMISSION REGARDING THE STANDARD IEC 61850 IN A LOCAL DATA NETWORK AND ITS PERFORMANCE TO SAFETY ATTACKS*

AUTHOR: DAVID ANDRÉS CASTRO RUIZ**

KEY WORDS: IEC 61850, GOOSE, OSI Model, Quick messages, Security Attacks

CONTENT:

IEC 61850 defines in its content a number of communication protocols which are used for sending different types of messages within substation automation. One protocol is GOOSE, through it messages are sent faster, indicating a change of state within the substation and it has a high demand of transfer time (between 3 and 10 ms) due to the importance of the messages inside the substation, thanks to this, the protocol uses the layers of control of the data link and physical of the OSI model, in order to speed up the transmission of messages.

This paper describes a GOOSE protocol implementation, following all the proposed requirements in IEC 61850, in order to analyze their behavior on a local area network and also, verify compliance with transfer times required by the standard IEC 61850 and observe the performance of quick messages to certain security attacks.

To carry out the analytical process was performed a series of experiments in which certain variables were changed as: the transmission type, the type of local network configuration, the number of computers GOOSE message producers and network traffic, in order to observe the performance of the local data network to these changes.

* Degree work

** Faculty of physical and mechanical engineering. School of Systems and Computer Engineering. Director Msc. Pedro Javier Trujillo Tarazona. Director Assistant Dr. Gabriel Ordoñez Plata

INTRODUCCIÓN

Desde hace algunos años, en la Universidad Industrial de Santander se vienen trabajando una serie de proyectos relacionados con la actualización del sistema eléctrico,[5] buscando con ello la conversión del sistema vigente por uno que tenga aspectos inteligentes, y así lograr la implementación en el alma mater de las Redes Inteligentes o Smartgrids[6].

Estos proyectos se han desarrollado desde un punto vista primordialmente eléctrico y ejecutados en su totalidad por la Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones – E3T, a través de sus grupos de investigación, más es conveniente incluir otras áreas de la ingeniería, una de ella es la Ingeniería de Sistemas y la Computación, cuyo aporte es muy importante en las redes inteligentes.

Se plantea este proyecto para hacer partícipe a la Ingeniería de Sistemas, colaborando con una evaluación a la trasmisión de datos respecto a la norma IEC 61850[7] en una red de datos local, incluyendo además, un examen al desempeño de la transmisión de mensajes ante ataques a la seguridad.

La norma IEC 61850 dictamina en su interior una serie de protocolos de comunicación los cuales se utilizan al interior de la subestación automatizada. Uno de los más importantes por su velocidad en la transmisión de mensajes es el protocolo GOOSE, el cual se define al interior de la norma, y maneja las capas física y de control del Enlace de datos del modelo OSI para lograr esta velocidad.

[5] Información obtenida luego de una entrevista realizada al Dr. Gabriel Ordóñez Plata, Director del grupo de Grupo de Investigación en Sistemas de Energía Eléctrica (GISEL).

[6] La red eléctrica inteligente (smart grid en inglés) es una forma de gestión eficiente de la electricidad que utiliza las tecnologías informáticas y de las telecomunicaciones para optimizar el desempeño de la red eléctrica en sus diversas áreas.

[7] IEC 61850 es un estándar mundial, desarrollado por la Comisión Electrotécnica Internacional (IEC), para el diseño de una subestación eléctrica automatizada. Esta específica un modelo de datos y servicios abstractos a los cuales se les puede asignar una serie de protocolos.

En este proyecto se propone una implementación de una herramienta que envía mensajes por medio del protocolo GOOSE, los cuales son denominados en la norma IEC 61850 como mensajes rápidos. El software se realizó mediante el lenguaje de programación Python, prefiriendo este debido a que la labor de programación del protocolo fue más rápida y eficiente.

Además, se efectúan exámenes al tiempo de transferencia de la transmisión de mensajes rápidos y al desempeño del envío de mensajes ante ataques a su seguridad. Para llevar a cabo estas tareas se hizo uso de herramientas libres encontradas en Internet. Estas evaluaciones son apoyadas en una serie de experimentos, en los que se alteraron las variables que intervenían, logrando con tal procedimiento, obtener resultados que sirvieran como apoyo a la investigación.

OBJETIVOS

OBJETIVO GENERAL

Estudiar el comportamiento de la transmisión de datos en una red de datos local respecto a la norma IEC 61850 y su desempeño ante ataques a la seguridad.

OBJETIVOS ESPECÍFICOS

- Programar una herramienta que envíe mensajes rápidos por medio del protocolo GOOSE definido en la norma IEC 61850.

- Realizar la transmisión de mensajes rápidos definidos en la norma IEC 61850-5 en una red de datos local, para determinar si cumplen o no con las exigencias de la misma.

- Realizar ataques a la transmisión de mensajes rápidos para observar el comportamiento de una red de datos local frente a estos.

1. MARCO TEÓRICO

1.1 CONCEPTOS GENERALES

1.1.1 Modelo OSI[8]

El modelo OSI (Open Systems Interconection) se desarrolló por la Organización Internacional de Estandarización ISO (International Organization for Standarization) como una arquitectura para comunicaciones entre computadores, con el objetivo de ser el marco de referencia en el desarrollo de protocolos estándares. OSI considera siete capas:

- Aplicación
- Presentación
- Sesión
- Transporte
- Red
- Enlace de datos
- Física

La Tabla 1 muestra las capas del modelo OSI y la función que cada una realiza. La intención del modelo OSI es que los protocolos se desarrollen de forma tal que realicen las funciones de cada una de las capas.

[8] STALLINGS, William. Comunicaciones y redes de computadores, Prentice Hall, 2000. p. 18.

Tabla 1. Capas del modelo OSI y su función

Capa del Modelo OSI	Función
Aplicación	Proporciona el acceso al entorno OSI para los usuarios y también proporciona servicios de información distribuida.
Presentación	Proporciona a los procesos de aplicación independencia respecto a las diferencias en la representación de los datos (sintaxis).
Sesión	Proporciona el control de la comunicación entre las aplicaciones; establece, gestiona y cierra las conexiones (sesiones) entre las aplicaciones cooperadoras
Transporte	Proporciona seguridad, trasferencia transparente de datos entre los puntos finales; proporciona además procedimientos de recuperación de errores y control de flujo origen-destino.
Red	Proporciona independencia a los niveles superiores respecto a las técnicas de conmutación y transmisión utilizadas para conectar los sistemas; es responsable del establecimiento, mantenimiento y cierre de las conexiones.
Enlace de datos	Proporciona un servicio de trasferencia de datos seguro a través del enlace físico; envía bloques de datos (tramas) llevando a cabo la sincronización, el control de errores y de flujo necesarios.
Física	Se encarga de la transmisión de cadenas de bits no estructurados sobre el medio físico; esta relacionada con las características mecánica, eléctricas, funcionales y de procedimiento para acceder al medio físico.

Fuente: STALLINGS, Willliam. Comunicaciones y redes de computadores, Prentice Hall, 2000. p. 19.

1.1.2 Ethernet [9]

Ethernet es una familia de tecnologías de interconexión de redes que se define en los estándares 802.2 y 802.3. Los estándares de Ethernet definen los protocolos de la Capa 2 y las tecnologías de la Capa 1. Ethernet es la tecnología LAN más ampliamente utilizada y soporta anchos de banda de datos de 10, 100, 1000, o 10000 Mbps.

Ethernet proporciona servicio sin conexión y sin reconocimiento sobre un medio compartido utilizando CSMA/CD como métodos de acceso al medio. El medio compartido requiere que el encabezado del paquete de Ethernet utilice la dirección de la capa de enlace de datos para identificar los nodos de origen y destino. Como con la mayoría de los protocolos LAN, esta dirección se llama dirección MAC del nodo. Una dirección MAC de Ethernet es de 48 bits y generalmente se representa en formato hexadecimal.

En la parte superior de la Figura 1 se muestra un esquema de la trama estándar de Ethernet. En ella se pueden ver tanto los campos principales que la forman como la extensión de éstos en bytes.

a) Preámbulo (Preambule)
b) Delimitador de inicio (SFD)
c) Dirección de destino (DA)
d) Dirección de origen (SA)
e) Longitud / Tipo de protocolo (TYPE)
f) Datos (DATA)
g) Código de redundancia (CRC)

[9] LÁZARO, Jorge. MIRALLES, Marcelo. Fundamentos de Telemática, España: Valencia Editorial de la UPV, 2005. p. 157.

En la Tabla 2 se muestran las tecnologías Ethernet más difundidas y utilizadas. La notación empleada para denominar a estas tecnologías es la siguiente: Velocidad en Mbps, tipo de señal y longitud máxima de segmento.

Las tecnologías Ethernet que existen se diferencian en estos conceptos:

- Velocidad de transmisión: Velocidad a la que transmite la tecnología.
- Tipo de cable: Tecnología del nivel físico que usa la tecnología.
- Longitud máxima: Distancia máxima que puede haber entre dos nodos adyacentes (sin estaciones repetidoras).
- Topología: Determina la forma física de la red. Bus si se usan conectores T (hoy sólo usados con las tecnologías más antiguas) y estrella si se usan hubs (estrella de difusión) o switches (estrella conmutada).

Tabla 2. Tecnologías Ethernet

Tecnología	**Velocidad de transmisión**	**Tipo de cable**	**Distancia máxima**	**Topología**
10Base2	10 Mbps	Coaxial	185 m	Bus (Conector T)
10BaseT	10 Mbps	Par Trenzado	100 m	Estrella (Hub o Switch)
10BaseF	10 Mbps	Fibra óptica	2000 m	Estrella (Hub o Switch)
100BaseT4	100Mbps	Par Trenzado (categoría 3UTP)	100 m	Estrella. Half Duplex (hub) y Full Duplex (switch)
100BaseTX	100Mbps	Par Trenzado (categoría 5UTP)	100 m	Estrella. Half Duplex (hub) y Full Duplex (switch)
100BaseFX	100Mbps	Fibra óptica	2000 m	No permite el uso de hubs

Fuente: http://es.wikipedia.org/wiki/Ethernet

1.1.3 Red de área local virtual o VLAN

Una VLAN es una red de área local que agrupa un conjunto de equipos de manera lógica y no física. "Varias VLANs pueden coexistir en un único conmutador físico o en una única red física. Son útiles para reducir el tamaño del dominio de difusión y ayudan en la administración de la red separando segmentos lógicos de una red de área local (como departamentos de una empresa) que no deberían intercambiar datos usando la red local (aunque podrían hacerlo a través de un enrutador o un conmutador de capa 3 y 4).

Una VLAN es una red de computadores que se comportan como si estuviesen conectados al mismo conmutador, aunque pueden estar en realidad conectados físicamente a diferentes segmentos de una red de área local." [10]

1.1.4 IEEE 802.1Q

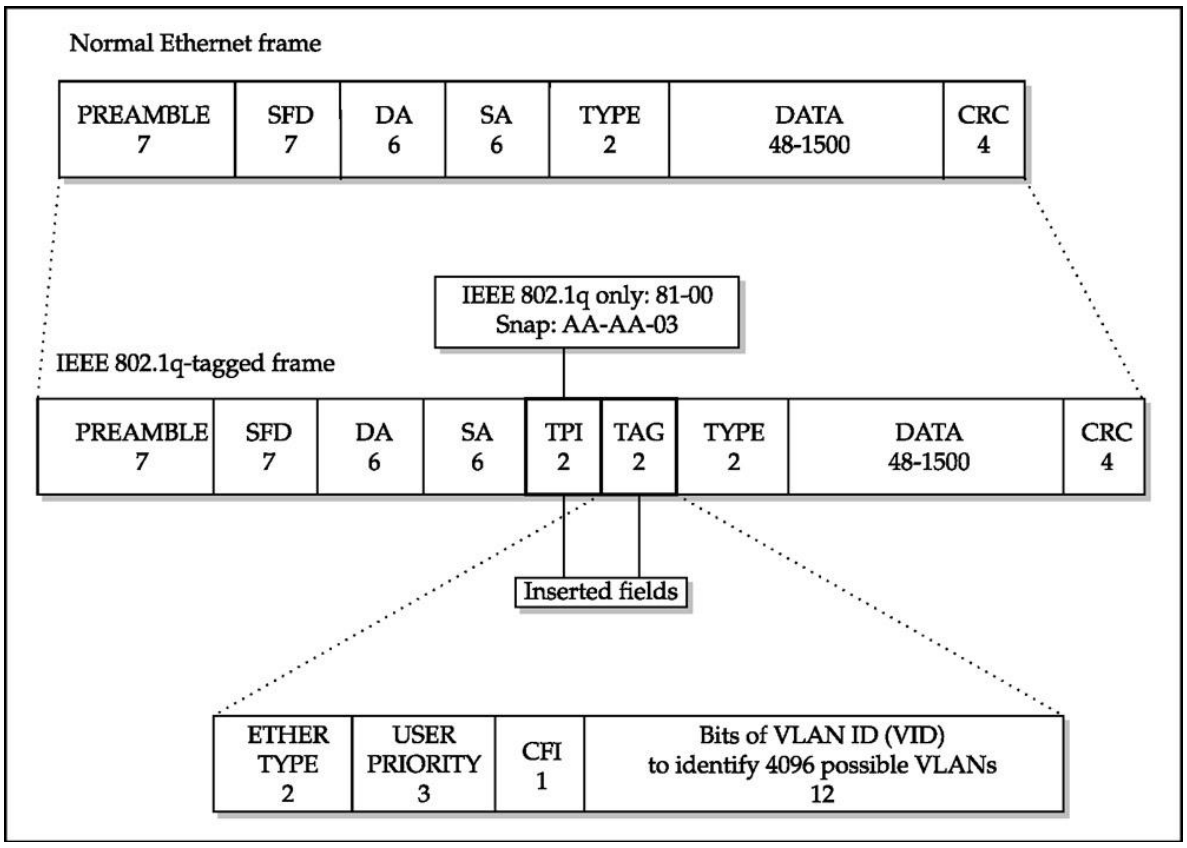
"El protocolo IEEE 802.1Q, fue un proyecto del grupo de trabajo 802 del IEEE para desarrollar un mecanismo que permita a múltiples redes compartir de forma transparente el mismo medio físico, sin problemas de interferencia entre ellas. Es también el nombre actual del estándar establecido en este grupo y se usa para definir el protocolo de encapsulamiento usado para implementar este mecanismo en redes Ethernet. Todos los dispositivos de interconexión que soportan VLAN deben seguir la norma IEEE 802.1Q que especifica con detalle el funcionamiento y administración de redes virtuales de área local."[11]

802.1Q en realidad no encapsula la trama original sino que añade 4 bytes al encabezado Ethernet original. El valor del campo Ethertype se cambia a 0x8100 para señalar el cambio en el formato de la trama. Obsérvese en la Figura 1.

---

[10] Tomado de: http://es.wikipedia.org/wiki/VLAN
[11] Tomado de: http://es.wikipedia.org/wiki/IEEE_802.1Q

Figura 1. Trama Ethernet e IEEE 802.1Q



Fuente: http://www.iphelp.ru/faq/35/files/fig12-6_0.jpg

1.1.5 Tipos de comunicación: Unicast, broadcast, multicast

En una red local, los host pueden comunicarse de tres maneras diferentes:

- **Unicast**: El proceso por el cual se envía de un host a un host individual.
- **Broadcast**: El proceso por el cual se envía un paquete de un host a todos los host de la red.
- **Multicast**: El proceso por el cual se envía un paquete de un host a un grupo seleccionado de hosts.

1.1.6 ASN.1 y BER

1.1.6.1 Abstract Syntax Notation One - ANS.1 [12]

La notación ASN.1 es un estándar recogido en la norma ISO/IEC 8825- 1, y fue desarrollado como parte de la capa de presentación de la pila de protocolos de comunicación OSI. Su notación básica puede encontrarse en el documento X.680 de la UIT-T[13].

Esta norma proporciona un nivel de abstracción similar a los lenguajes de alto nivel, y permite que dispositivos de distintos fabricantes o con representaciones de datos diferentes puedan disponer de una sintaxis de representación estándar.

ASN.1 define varias clases de datos, llamadas tipos. De entre ellos, son dos los que resultan interesantes a la hora de construir el encapsulado de la trama IEC 61850, y se definen a continuación:

- Tipos primitivos (Primitive): Representan las clases de datos básicas. Ejemplos de este tipo son INTEGER, BOOLEAN, OCTECT STRING.
- Tipos constructores o estructurados (Constructed): Esta clase de datos permite la formación de una tabla o lista a partir de los tipos primitivos.

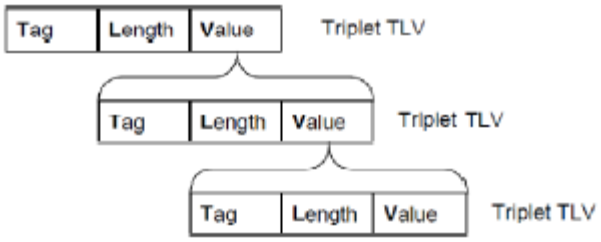
Una vez definidos los tipos de datos que en este caso alberga la trama física, se ilustra en la Figura 2 la estructura que debe tener un mensaje encapsulado según este estándar de interpretación de datos.

---

[12] RIVAS, Eduardo Alonso. Diseño de una plataforma de comunicaciones bajo la norma IEC 61850. p. 50.

[13] INTERNATIONAL TELECOMMUNICATION UNION. ITU-T X.680. Disponible en: <http://www.itu.int/rec/T-REC-X.680-200811-I/en>

Figura 2. Formato BER para el encapsulado de la trama



Fuente: Diseño de una plataforma de comunicaciones bajo la norma IEC 61850. <http://www.iit.upcomillas.es/pfc/resumenes/4aa7d65d31770.pdf>

1.1.6.2 Reglas de codificación básicas - BER

El conjunto de reglas de codificación básicas, o BER (Basic Encoding Rules), es uno de los formatos de codificación definidos como parte del estándar ASN.1. Las reglas, denominadas sintaxis de transferencia en el contexto de ASN.1, especifican las secuencias de octetos exactas para codificar un elemento de datos dado. La sintaxis BER, junto con dos subconjuntos de BER: Canonical Encoding Rules (CER) y Distinguished Encoding Rules (DER), están definidas por el documento X.690 de la UIT-T[14], el cual es parte de las series de documentos ASN.1.

La codificación BER especifica un formato de auto-descripción y auto-limitación para la llevar a cabo la codificación de las estructuras datos formadas desde ASN.1. Cada elemento de datos se codifica como un identificador de tipo, una descripción de longitud, los elementos de datos reales y en caso necesario, un marcador de fin de contenido. Estos tipos de codificaciones son comúnmente llamados tipo-longitud-valor (tag-length-value) o codificación TLV. Cada uno de

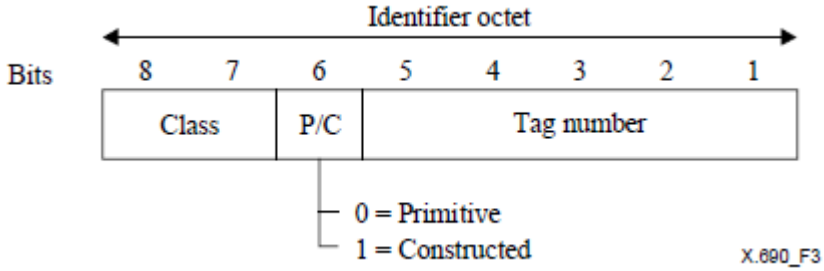
[14] INTERNATIONAL TELECOMMUNICATION UNION. ITU-T X.690. Disponible en: <http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf>

estos campos está formado por series de octetos y cumplen las siguientes especificaciones.

1.1.6.2.1 Campo tipo o tag

El campo tipo es un octeto que especifica las características de este campo.

Figura 3. Identificador de octeto



Fuente: INTERNATIONAL TELECOMMUNICATION UNION. ITU-T X.690: OSI networking and system aspects – Abstract Syntax Notation One (ASN.1) – Information technology – ASN.1 encoding rules: Specification of basic encoding rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

Si la clase (Bits 8 y 7) está definida en Universal, el valor es de tipo nativo en ASN.1 (por ejemplo, INTEGER). La clase Application es solo válida para una aplicación específica. La clase context-specific depende del contexto (por ejemplo, dentro de una secuencia, conjunto o selección) y la clase private, puede ser definida en especificaciones particulares.

Tabla 3. Codificación de la etiqueta class

Class	Bit 8	Bit 7
Universal	0	0
Application	0	1
Context-specific	1	0
Private	1	1

Fuente: INTERNATIONAL TELECOMMUNICATION UNION. ITU-T X.690: OSI networking and system aspects – Abstract Syntax Notation One (ASN.1) – Information technology – ASN.1 encoding rules: Specification of basic encoding rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

El bit 6 P / C especifica si el valor es primitive (0), como un INTEGER o Constructed (1), lo que significa que de nuevo tiene valores del tipo TLV.

El ultimo conjunto de bits llamado Tag number, especifica la etiqueta del tipo de dato. La siguiente tabla muestra las etiquetas más representativas.

Tabla 4. Etiquetas para el campo Tag number

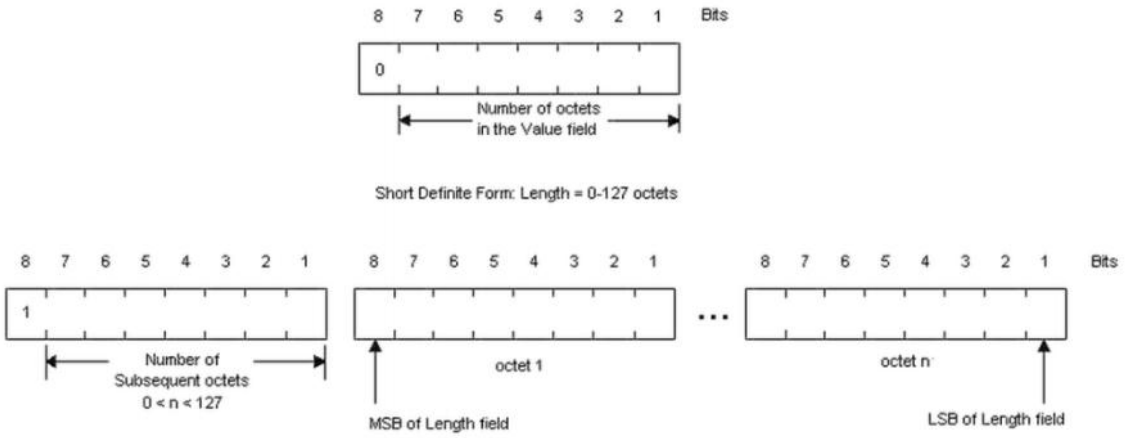
Nombre	P / C	Número (decimal)	Número (hexadecimal)
EOC (End-of-content)	P	0	0
BOOLEAN	P	1	1
INTEGER	P	2	2
BIT STRING	P/C	3	3
FLOAT	P	9	9
UTCTime	P/C	23	17

Fuente: Autor

1.1.6.2.2 Campo longitud o length [15]

Sirve para determinar el número de bytes que contendrá el siguiente campo. Puede estructurarse según dos formatos, como se puede observar en la Figura 4.

Figura 4. Formato de codificación del campo longitud



Fuente: Diseño de una plataforma de comunicaciones bajo la norma IEC 61850. <http://www.iit.upcomillas.es/pfc/resumenes/4aa7d65d31770.pdf>

Por un lado, la forma definida corta ocupa un único byte y representa una longitud de Valor de hasta 127 octetos, ya que para su identificación el bit 8 del campo longitud debe valer cero. Para la representación de datos que ocupen más de 127 bytes, se emplea la forma definida larga. El primer byte de esta estructura contiene el número de octetos que le siguen necesarios para representar toda la longitud del campo Valor, y está identificado por un uno en su bit más significativo. A partir del segundo octeto se encuentra el valor del campo Longitud, que puede representarse hasta con 127 bytes. Esta estructura puede representar longitudes más cortas que los 128 bytes, lo que será de utilidad para realizar la

[15] RIVAS, Eduardo Alonso. Diseño de una plataforma de comunicaciones bajo la norma IEC 61850. p. 52

implementación física, ya que permite reservar tres bytes para el campo de longitud aunque el tamaño pueda ser variable y no restringido a 127 bytes.

1.1.6.2.3 Campo valor o value [16]

Una vez identificado el tipo de dato y la longitud en bytes que ocupa, sólo queda agregar el valor representado a las cabeceras anteriores. Téngase en cuenta que este campo puede constituir una nueva estructura TLV.

Para completar la definición de la sintaxis, se ha de establecer el orden de envío de los datos. Así, ASN.1 presenta un formato Big Endian, de tal forma que cada byte transmite primero el bit más significativo (MSB), el cual se sitúa a la izquierda del octeto en la representación estructurada.

1.2 NORMA IEC 61850

IEC 61850 es un estándar mundial, desarrollado por la Comisión Electrotécnica Internacional (IEC), para el diseño de una subestación eléctrica automatizada. Esta especifica un modelo de datos y servicios abstractos a los cuales se les puede asignar una serie de protocolos, entre los cuales están: GOOSE (Generic Object Oriented Substation Event), MMS (Manufacturing Message Specification) y SV (Sampled Values). Estos protocolos usan Ethernet y algunos TCP/IP para la comunicación.

IEC 61850 consta de las siguientes partes separadas que se detallan en los documentos del estándar IEC 61850[17]:

---

[16] RIVAS, Eduardo Alonso. Diseño de una plataforma de comunicaciones bajo la norma IEC 61850. p. 52.

[17] INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 1: Communication Networks and Systems in Substations - Introduction and overview.

- **IEC 61850-1**: Introduction and overview
- **IEC 61850-2**: Glossary
- **IEC 61850-3**: General requirements
- **IEC 61850-4**: System and project management - Ed.2
- **IEC 61850-5**: Communication requirements for functions and device models
- **IEC 61850-6**: Configuration language for communication in electrical substations related to IEDs - Ed.2
- **IEC 61850-7**: Basic communication structure for substation and feeder equipment
    - **IEC 61850-7-1**: Principles and models - Ed.2
    - **IEC 61850-7-2**: Abstract communication service interface (ACSI) - Ed.2
    - **IEC 61850-7-3**: Common Data Classes - Ed.2
    - **IEC 61850-7-4**: Compatible logical node classes and data classes - Ed.2
- **IEC 61850-7-10**: Communication networks and systems in power utility automation - Requirements for web-based and structured access to the IEC 61850 information models
- **IEC 61850-8**: Specific communication service mapping (SCSM)
    - IEC 61850-8-1: Mappings to MMS (ISO/IEC9506-1 and ISO/IEC 9506-2) - Ed.2
- **IEC 61850-9**: Specific communication service mapping (SCSM)
    - IEC 61850-9-1: Sampled values over serial unidirectional multidrop point to point link
- **IEC 61850-9-2**: Sampled values over ISO/IEC 8802-3 - Ed.2
- **IEC 61850-10**: Conformance testing

1.2.1 Objetivos y alcance de la norma IEC 61850 [18]

Una de las principales necesidades, dadas por la experiencia en el desarrollo e implementación de Sistemas de Automatización de Subestaciones (SAS), es contar con estándares de protocolos de comunicaciones, que proporcionen la posibilidad de soportar la interoperabilidad entre Dispositivos Electrónicos Inteligentes (IED) de distintos fabricantes. En este caso el concepto de interoperabilidad, se relaciona con la propiedad de conectar IED de diferentes fabricantes, en una misma red LAN y que estos compartan información y comandos entre ellos sin ningún problema.

El objetivo de estandarizar los sistemas de comunicaciones dentro de un SAS es desarrollar un estándar que cuente con los requerimientos de funcionamiento y desempeño necesarios, además que sea capaz de soportar los avances tecnológicos venideros. Estos con el fin de que sea lo más beneficioso posible.

Este estándar debe cumplir con los requerimientos funcionales de una subestación, por lo tanto debe soportar las funciones de operación de la misma, sin caer en el error de limitar las funciones envueltas en la operación de una subestación.

Estas funciones deben ser identificadas y descritas claramente con el objeto de definir los requerimientos de comunicaciones de cada una de ellas, por ejemplo la cantidad de datos a ser intercambiados, retrasos de tiempo en las trasmisiones, etc.

---

[18] CEDIEL MARTINEZ, Julián Andrés. CHAPARRO RESTREPO, Jaime Hernán. Aspectos generales en la automatización de una subestación eléctrica y empleo del protocolo de comunicación IEC 61850. p.60.

1.2.2 Concepto de función [19]

Una función es una tarea que se lleva a cabo por el Sistema de Automatización de Subestaciones (SAS). Generalmente, las funciones intercambian datos con otras funciones y este es realizado por los IEDs (Dispositivos físicos). Las funciones pueden ser dividas en partes que residen en distintos IEDs pero estas pueden comunicarse una con la otra (función distribuida[20]) y con partes de otras funciones. Estas partes de comunicación son llamadas nodos lógicos.

En el contexto de esta norma, la descomposición de funciones o su granularidad es gobernado por el comportamiento de la comunicación únicamente. Por lo tanto, todas las funciones consideradas en esta consisten de nodos lógicos que intercambiar datos.

1.2.3 Concepto de nodo lógico (LN) [21]

Se define como la parte más pequeña de una función que intercambia datos. Un nodo lógico (LN) es un objeto definido por sus datos y métodos.

1.2.4 Concepto de PICOM [22]

PICOM (Piece of Information COMunication) es una descripción de una trasferencia de información en una conexión lógica dada, con atributos de comunicación dados entre dos nodos lógicos.

---

[19] INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 1: Communication Networks and Systems in Substations - Introduction and overview
[20] INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 2: Communication Networks and Systems in Substations – Glossary
[21] INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 1: Communication Networks and Systems in Substations - Introduction and overview
[22] INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 5: Communication Networks and Systems in Substations

Se utilizan para describir la información intercambiada entre nodos lógicos. Los componentes o atributos de un PICOM son los siguientes:

- Datos, es decir, el contenido de la información y su identificación, según sea necesario por las funciones (semántica).
- Tipo, que describe la estructura de los datos, es decir, si se trata de un análogo o un valor binario; de un valor único o un conjunto de datos, etc.
- El desempeño, es decir, el tiempo de transmisión permitida (definida por la clase de rendimiento), la integridad de los datos y el método o la causa de la transmisión (por ejemplo, evento periódico, impulsado, bajo petición).
- La conexión lógica, que contiene la fuente lógica (nodo lógico enviado) y el disipador de lógica (nodo lógico de destino o recepción).

NOTA: El PICOM no representa la estructura real o el formato de los datos que se transmiten a través de la red de comunicación.

1.2.5 Modelamiento de la información [23]

Los mecanismos de intercambio de la información están basados primariamente en modelos bien definidos de la misma. El IEC 61850 usa el acercamiento al modelamiento de información común, encontrada en el mundo tal y como se muestra en la Figura 5.

La definición de la información y su intercambio, está hecha de tal manera que sea independiente de una implementación concreta; para ello es usado el concepto de virtualización, el cual provee una visión de aquellos aspectos

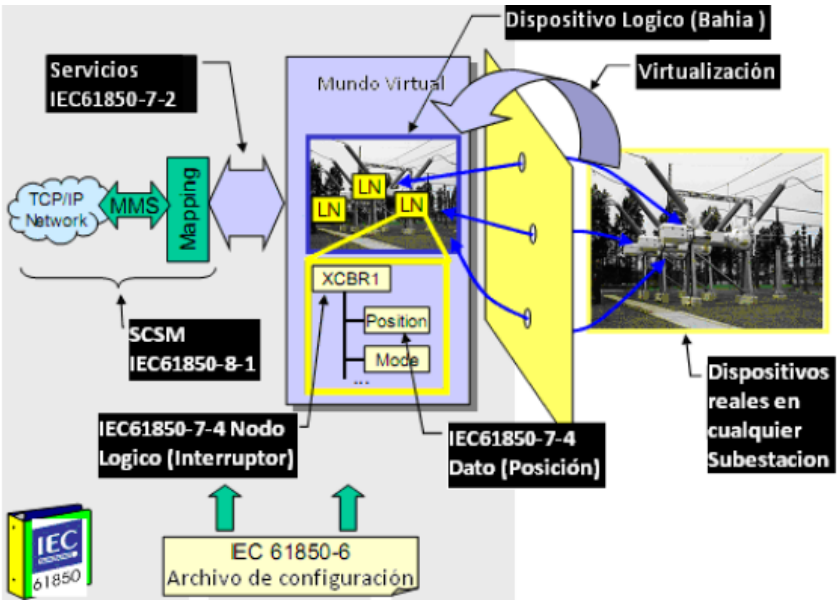
[23] CEDIEL MARTINEZ, Julián Andrés. CHAPARRO RESTREPO, Jaime Hernán. Aspectos generales en la automatización de una subestación eléctrica y empleo del protocolo de comunicación IEC 61850. p.65.

de un dispositivo real que son de interés para el intercambio de información con otros dispositivos.

Como se menciono anteriormente el estándar IEC 61850 descompone las funciones de aplicación en las más pequeñas entidades posibles, llamadas nodos lógicos (LN), los cuales son usados para el intercambio de información. Basado en su funcionalidad un LN contiene una lista de datos con sus atributos de datos dedicados, los cuales tienen una estructura y semántica bien definida.

La semántica de un nodo lógico esta representada por los datos y los atributos que contiene. Por ejemplo, dentro del LN *XCBR* (XCBR es un nodo lógico asociado a interruptores), el dato Pos indica la posición del interruptor, éste a su vez contiene el atributo *Pos.ctlVal* que representa la información de control (la cual puede ser ON u OFF), también el atributo *Pos.stVal* el cual indica la posición del interruptor (que podría ser ON, OFF, inválido o intermedio), etc. Este ejemplo muestra como está definida la estructura semántica anteriormente nombrada.

Figura 5. Acercamiento conceptual al modelamiento



Fuente: IEC 61850 part 7-1 Communication Networks and Systems in Substations – Basic communication structure for substation and feeder equipment- Principles and models

1.2.6 Requerimientos generales del estándar IEC 61850 [24]

De acuerdo al estándar IEC 61850 los requerimientos generales de un sistema de comunicaciones para un SAS, necesarios para cumplir con los objetivos del mismo, están presentados en dos grupos, que son: los requerimientos de desempeño, relacionados con los aspectos técnicos de tiempo de transmisión e integridad de la información y los requerimientos funcionales (los cuales no son del alcance de este proyecto, por ser exclusivos de la implementación de un SAS), que relacionan todos los aspectos de la calidad de la transmisión, condiciones ambientales de la red física y los servicios auxiliares.

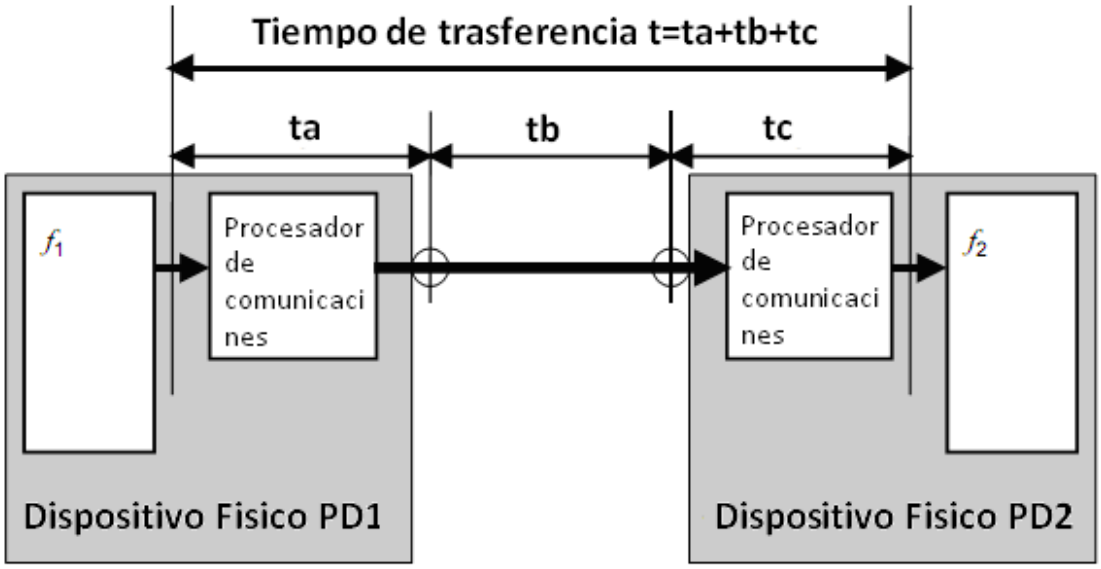
[24] CEDIEL MARTINEZ, Julián Andrés. CHAPARRO RESTREPO, Jaime Hernán. Aspectos generales en la automatización de una subestación eléctrica y empleo del protocolo de comunicación IEC 61850. p.70.

1.2.6.1 Requerimientos de desempeño[25]

El estándar IEC 61850 define varios requerimientos de desempeño para las funciones dentro de una subestación, independiente del tamaño de la misma, basados en dos grupos de desempeño y en el tiempo de transferencia.

El tiempo de transferencia es el tiempo necesario para enviar datos desde una función f1 hacia una función f2 ubicada en otro dispositivo físico. En este tiempo se debe incluir los retardos de transmisión además de los tiempos de procesamiento necesarios. Ver Figura 6.

Figura 6. Definición de tiempo de trasferencia



Fuente: IEC 61850 part 5: Communication Networks and Systems in Substations - Communication requirements for functions and device models

[25] CEDIEL MARTINEZ, Julián Andrés. CHAPARRO RESTREPO, Jaime Hernán. Aspectos generales en la automatización de una subestación eléctrica y empleo del protocolo de comunicación IEC 61850, p.70.

Los grupos de desempeño son independientes entre ellos y están definidos de acuerdo a las funcionalidades requeridas. Los grupos de desempeño son:

- Grupo de control y protección.
- Grupo de medición y calidad de la energía.

Los requerimientos para el grupo de control y protección son más exigentes, debido al efecto e importancia de despejar una falla en el menor tiempo posible, para la estabilidad del sistema de potencia.

El estándar IEC 61850 categoriza el grupo de control y protección en tres clases:

- **Clase de desempeño P1:** Aplica para las subestaciones de distribución o en casos donde los niveles de desempeño aceptados sean bajos.
- **Clase de desempeño P2:** Aplica típicamente a subestaciones de transmisión.
- **Clase de desempeño P3:** Aplica típicamente a subestaciones de trasmisión con altos requerimientos de sincronismo, como funciones de protección de barraje.

Las especificaciones de desempeño que se deben cumplir para cada clase dependen del tipo de mensaje a transmitir.

La norma IEC 61850 explica los requerimientos para el grupo de medición y calidad de la energía, citándolos como:

- Confiabilidad
- Disponibilidad

- Mantenibilidad y seguridad
- Integridad de los datos
- Distancia para las redes de comunicación
- Cantidad de IED integrados al SAS

1.2.7 Tipos de mensajes y clases de rendimiento [26]

Tipo 1 - Mensajes rápidos: Este tipo de mensaje suele contener un código binario simple que contiene datos, comandos o un mensaje simple, por ejemplo, "Trip", "Cerrar", "orden de reenganche", "start", "Stop", "Bloquear", "Desbloquear", "Trigger ","Suelta", "cambio de estado". Los mensajes de tipo 1 están divididos en dos tipos: tipo 1A y tipo 1B.

Tipo 1A "trip o disparo": El mensaje tipo 1A es el más importante de los "mensajes rápidos" en la subestación. Por lo tanto, este mensaje tiene requisitos más exigentes en comparación a todos los otros mensajes.

- Para la clase de desempeño P1, el tiempo de transmisión total será de un máximo de 10 ms.
- Para las clases de desempeño P2 y P3, el tiempo total de transmisión deberá ser inferior a la orden de un cuarto de ciclo. Por lo tanto, se define un máximo de 3ms.

Tipo 1B "Otros": Los mensajes de tipo 1B son importantes para la interacción entre el sistema de automatización (SAS) y el proceso, pero tienen requisitos menos exigentes en comparación con los de tipo 1A.

---

[26] INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 5: Communication Networks and Systems in Substations

- Para la clase de desempeño P1, el tiempo total de transmisión deberá ser inferior o igual a 100 ms.
- Para las clases de desempeño P2 y P3, el tiempo de transmisión total será de un máximo de 20ms.

Tipo 2 - Mensajes de velocidad media: Estos son los mensajes donde el tiempo en el cual se originó el mensaje es importante, pero donde el tiempo de transmisión es menos crítico. Se espera que los IED tengan sus propios relojes. El mensaje deberá incluir una etiqueta de tiempo fijado por el remitente y el receptor normalmente va a reaccionar después de un retardo de tiempo interno. El tiempo total de transmisión deberá ser inferior a 100 ms.

Tipo 3 - Mensajes de baja velocidad: Los mensajes de tipo 3 son usados para transmitir eventos, cambiar parámetros del SAS y otros mensajes complejos que requieran una etiqueta de tiempo. El tiempo total de transmisión deberá ser inferior a 500 ms.

Tipo 4 - Mensajes de datos en bruto: Este tipo de mensaje incluye datos de salida de los transductores de digitalización y transformadores digitales de instrumentos independientes de la tecnología de los transductores (magnético, óptico, etc.) Los datos consisten en secuencias continuas de datos sincronizados de cada IED, intercalados con los datos de otro IED. El tiempo total de transmisión para cada clase de desempeño es la misma que la definida para mensajes de tipo 1A.

Tipo 5 - Funciones de transferencia de archivos: Este tipo de mensaje se utiliza para transferir archivos grandes de datos, se utilizan para transferir archivos de información, configuraciones, datos, etc. Este tipo de mensaje debe ser dividido en bloques de longitud limitada, para permitir que otras actividades de comunicación en la red puedan realizarse. Los tiempos de transferencia no son críticos, no hay

límites específicos. Típicamente, los requisitos de tiempo son iguales o superiores a 1000 ms.

Tipo 6 - Mensajes de sincronización horaria: Este tipo de mensaje se utiliza para sincronizar los relojes internos de los IED en el SAS. Este tipo de mensaje no tiene requerimientos de desempeño.

Tipo 7 - Mensajes de comandos con el control de acceso: Este tipo de mensaje se utiliza para transferir las órdenes de control emitidas desde las funciones HMI locales o remotos, requiriendo un mayor grado de seguridad. Este tipo de mensajes está basado en los mensajes tipo 3, pero con la adición de control de acceso.

La Tabla 5 resume el tiempo de trasmisión de cada tipo de mensaje para las tres clases de desempeño.

Tabla 5. Tipos de mensajes, tiempos de transmisión y clases de desempeño

		Clase de desempeño		Observación
		P1	P2 / P3	
TIPO DE MENSAJE	1A	<10ms	<3ms	Tipo 1A (trip o disparo) Mensajes rápidos
	1B	<100ms	<20ms	Tipo 1B (otros) Mensajes rápidos
	2	<100ms		Tipo 2 - Mensajes de velocidad media
	3	<500ms		Tipo 3 - Mensajes de baja velocidad
	4	<10ms	<3ms	Tipo 4 - Mensajes de datos en bruto
	5	<1000ms		Tipo 5 - Funciones de transferencia de archivos
	6	No Aplica	No Aplica	Tipo 6 - Mensajes de sincronización horaria
	7	No Aplica	No Aplica	Tipo 7 - Mensajes de comandos con el control de acceso

Fuente: Autor

1.2.8 Diferencia entre PICOM y tipos de mensajes

La principal diferencia entre PICOMs y tipos de mensajes, se halla en que los PICOMs se refieren a la transferencia de información basada estrictamente en la funcionalidad; y los tipos de mensajes, son una agrupación de atributos relacionados con el rendimiento PICOM y, por lo tanto, definen los requisitos de rendimiento para ser admitidos. Dado que los requisitos de rendimiento se definen por mensaje, estos son independientes del tamaño de la subestación.

1.2.9 Mapeo de las comunicaciones

"Los modelos y servicios propios del estándar IEC 61850 se catalogan como modelos y servicios de tipo Abstract Communication Service Interface (ACSI por sus siglas en inglés) los cuales deben ser traducidos (mapeados) a modelos y servicios que empleen protocolos estándares como MMS y Ethernet."[27]

Como ya se había definido anteriormente el estándar IEC 61850 concreta siete tipos de mensajes, cada uno con distintas prioridades de transmisión. Los que se listan a continuación:

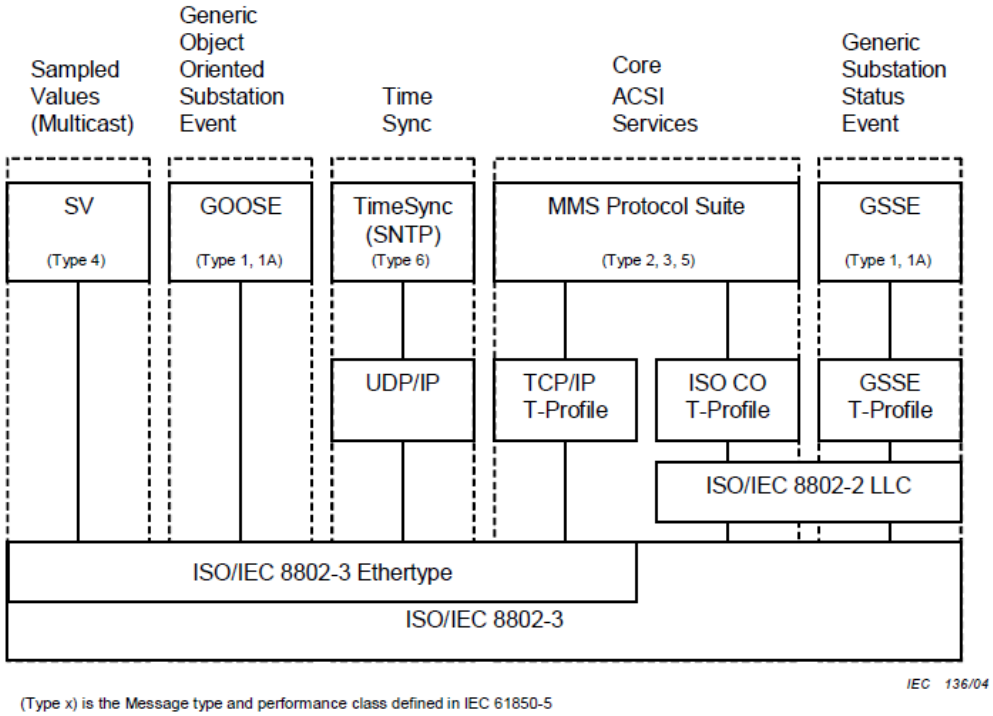
- Tipo 1 - Mensajes rápidos
- Tipo 2 - Mensajes de velocidad media
- Tipo 3 - Mensajes de baja velocidad
- Tipo 4 - Mensajes de datos en bruto
- Tipo 5 - Funciones de transferencia de archivos
- Tipo 6 - Mensajes de sincronización horaria
- Tipo 7 - Mensajes de comandos con el control de acceso

[27] CEDIEL MARTINEZ, Julián Andrés. CHAPARRO RESTREPO, Jaime Hernán. Aspectos generales en la automatización de una subestación eléctrica y empleo del protocolo de comunicación IEC 61850. p.102.

Los tipos 2, 3 y 5 son mapeados a mensajes MMS, debido a que este estándar (ISO 9506 Indutrial Automation Systems – Manufactuaring Message Specification – Part 1-Service definition) ofrece los servicios requeridos para su transmisión, mientras que los tipos 1, 1A, 4 y 6 emplean perfiles de comunicación no-MMS.

La Figura 7 ilustra la correspondencia entre los tipos de mensajes y los protocolos de comunicación identificados como adecuados para cada tipo de mensaje.

Figura 7. Tipos de mensajes y protocolos de comunicación



Fuente: IEC 61850 part 8-1: Communication Networks and Systems in Substations – Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3

El usuario del IEC 61850 se encarga de configurar y utilizar los modelos y servicios propios del estándar (modelos y servicios ACSI) de acuerdo a la aplicación que esté desarrollando, el IED que va trasmitir dichos modelos o

servicios, automáticamente se encarga de hacer el mapeo y adecuación necesaria para que el mensaje pueda ser trasmitido empleando el perfil de comunicación adecuado.

Un perfil de comunicación se define como el conjunto de protocolos y normas empleados en cada capa del modelo OSI. ISO reconoce la existencia de dos tipos de perfiles, de aplicación (Perfiles A) y de transporte (Perfiles T).

El perfil A abarca las capas de aplicación, presentación y sesión; el perfil T las capas de transporte, red, enlace y física.

Los diferentes tipos de mensajes, mencionados anteriormente, se pueden categorizar dentro de cuatro grupos de modelos y servicios ACSI:

- Modelos y servicios cliente/servidor
- Modelos y servicios de administración de GOOSE/GSE
- Modelos y servicios GSSE
- Modelos y servicios de sincronización

Para cada grupo de modelos y servicios se emplean diferentes combinaciones de perfiles A y perfiles T.[28]

1.2.10 Concepto GOOSE/GSE

Generic Substation Events (GSE) es un modelo de control que proporciona un mecanismo rápido y fiable de transferencia de datos de eventos a través de redes

[28] CEDIEL MARTINEZ, Julián Andrés. CHAPARRO RESTREPO, Jaime Hernán. Aspectos generales en la automatización de una subestación eléctrica y empleo del protocolo de comunicación IEC 61850. p.103.

de subestaciones completas. Cuando se implementa, este modelo asegura que el mensaje es recibido por varios dispositivos físicos que utilizan los servicios multicast / broadcast. El modelo de control GSE se subdivide en dos:

- GOOSE (Generic Object Oriented Substation Events)
- GSSE (Generic Substation State Events)

NOTA: El modelo de control GSSE no es del alcance de este proyecto.

1.2.11 Modelos y servicios de tipo administración GOOSE/GSE

La Tabla 6 contiene los modelos y servicios ACSI de tipo administración de GOOSE/GSE, con su respectiva categoría.

Tabla 6. Modelos y servicios ACSI GOOSE/GSE

Modelo	Servicio	Categoría
Generic Substation Event (GSE)	GetReference	UniCast
	GetGOOSEElementNumber	UniCast
	SendGOOSEMessage	Multicast

Fuente: Autor

1.2.11.1 SendGOOSEMessage[29]

El servicio SendGOOSEMessage proporciona la posibilidad de una rápida y fiable distribución de valores de entrada y salida de datos. Este SCSM (Specific Communication Service Mapping) utiliza un esquema especifico de re-transmisión para lograr el nivel adecuado de fiabilidad. Cuando un servidor GOOSE genera una petición SendGOOSEMessage, los valores de datos actuales se codifican en

[29] IEC 61850 part 8-1: Communication Networks and Systems in Substations – Specific Communication Service Mapping (SCSM) – Mappings to MMS

un mensaje GOOSE y se trasmiten por multidifusión. El evento que hace el servidor para invocar un servicio SendGOOSE es una cuestión de aplicación local tal como se define en la norma IEC 61805-7-2. La fiabilidad adicional se logra mediante la re-transmisión de los mismos datos (con SqNum aumentando gradualmente y el tiempo de re-transmisión, el campo SqNum será explicado con detalle más adelante).

Cada mensaje en la secuencia de re-transmisión lleva un parámetro timeAllowedToLive que informa al receptor el tiempo máximo de espera de la siguiente re-trasmisión. Si un nuevo mensaje no es recibido dentro de ese intervalo de tiempo, el receptor asume que la asociación se ha perdido.

El servicio SendGOOSEMessage, permite que un cliente envíe información de las variables de manera no solicitada y no confirmada.

1.2.12 Protocolo GOOSE

La Tabla 7 resume el protocolo GOOSE, usado para el mapeo de los servicios de administración GOOSE/GSE.

Se resalta el hecho que los servicios de administración de GOOSE/GSE no emplean servicios ni protocolos en las capas de sesión, trasporte y red; debido a la alta prioridad con que se deben transmitir, este tipo de mensajes son escritos directamente en la capa de enlace de datos.

Tabla 7. Protocolo GOOSE

Capa del Modelo OSI	Norma Utilizada
Aplicación	IEC 61850-8-1
Presentación	ASN.1 / BER
Sesión	No aplica
Transporte	No aplica
Red	No aplica
Enlace de datos	Ethernet / IEEE 802.1Q
Física	Fibra o Par trenzado

Fuente: Autor

NOTA: En cuanto al uso de tecnologías Ethernet el protocolo GOOSE define la utilización de las siguientes: 10Base-T/100Base-T (ISO/IEC 8802-3: 2001) y *Fibre optic transmission system* 100Base-FX (ISO/IEC 8802-3: 2001), referidas en la norma IEC 61850-8-1 Tabla 8. De las cuales se hizo uso de: 100Base-T.

1.2.12.1 Mensajes GOOSE[30]

Los mensajes GOOSE contienen información que permite que el dispositivo receptor pueda saber que un estado ha cambiado y en qué tiempo ocurrió este cambio. La hora del último cambio de estado permite a un dispositivo receptor establecer temporizadores locales relacionados con un acontecimiento determinado.

Un dispositivo recientemente activado o encendido debe enviar los datos actuales (Estado) o los valores iniciales como mensaje GOOSE. Así mismo todos los dispositivos que envían mensajes GOOSE deben continuar enviando mensajes

[30] IEC 61850 part 7-2: Communication Networks and Systems in Substations – Basic communication structure for substation and feeder equipment – Abstract communication service interface (ACSI)

con un tiempo de ciclo largo, incluso si no hay cambio de estado o no se a producido ningún nuevo valor. Esto asegura que los dispositivos que se han encendido recientemente conozcan los valores de estado actuales de otros dispositivos.

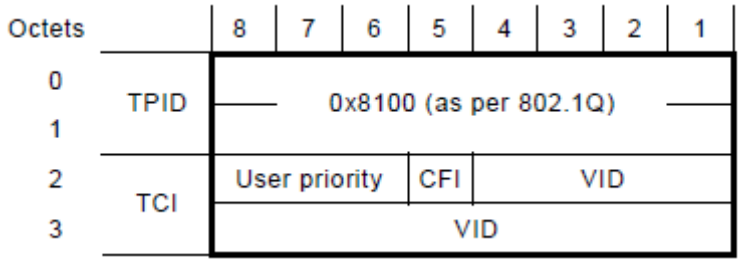
1.2.12.2 Descripción general de la estructura de la trama GOOSE/GSE [31]

Address fields (Campos de dirección): El campo destination address (dirección de destino) tiene que configurarse para la trasmisión de GOOSE/GSE como una dirección multidifusión.

Priority tagging/Virtual LAN: El campo Priority tagging de acuerdo a la norma IEEE 802.1Q se utiliza para separar el tráfico en tiempo crítico.

La estructura de la cabecera de la etiqueta VLAN se define en la Figura 8.

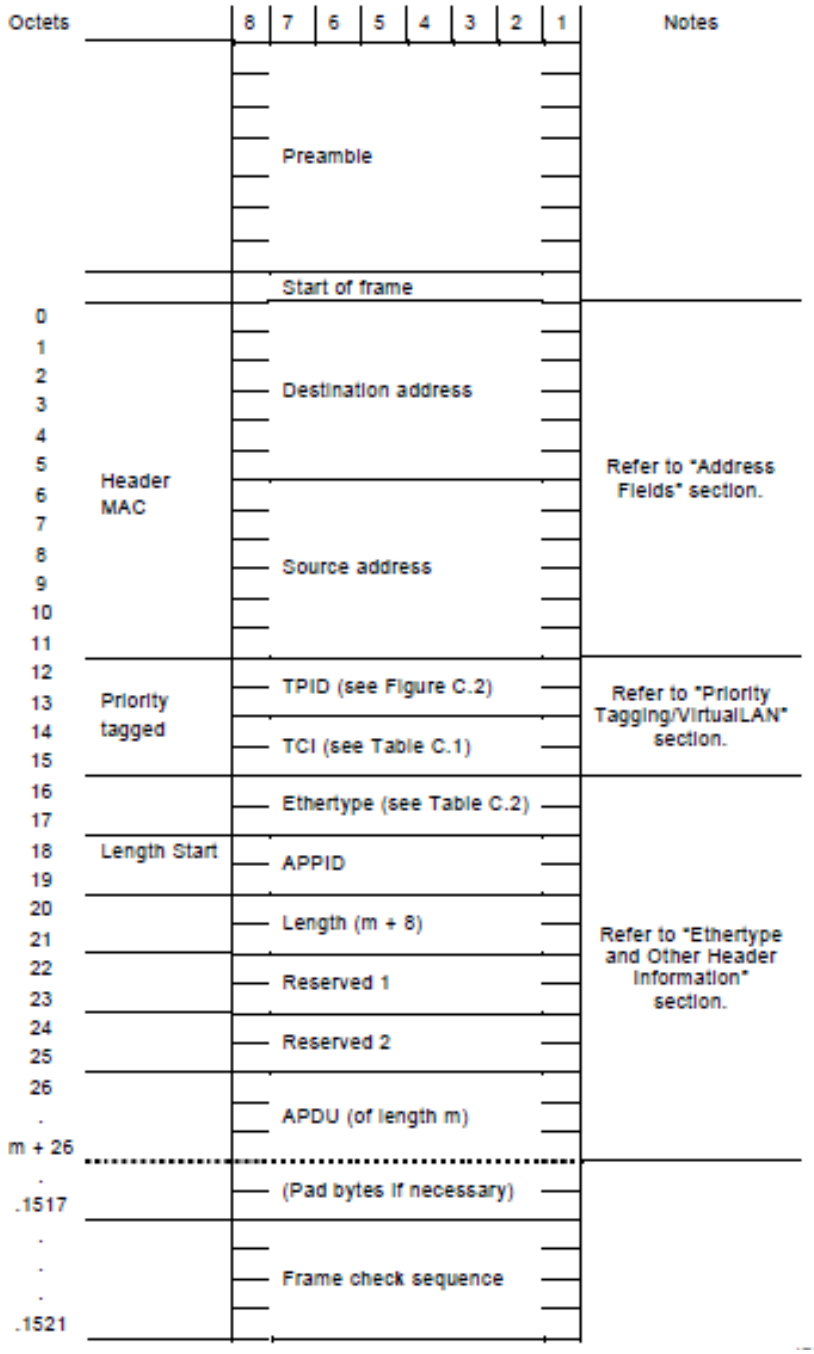
Figura 8. Etiqueta de virtual LAN



Fuente: IEC 61850 part 8-1: Communication Networks and Systems in Substations – Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3

[31] IEC 61850 part 8-1: Communication Networks and Systems in Substations – Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3

Figura 9. Formato de la trama GOOSE



Fuente: IEC 61850 part 8-1: Communication Networks and Systems in Substations – Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3

TPID (Tag Protocol Identifier): Indica el Ethertype asignado para la trama Ethernet 802.1Q. Este valor será 0x8100.

TCI (Tag Control Information): Se compone de tres campos User Priority, CFI y VID. El valor de User Priority se establecerá según la configuración para separar los valores de muestra y el tiempo crítico de mensajes GOOSE. Si este campo no esta configurado, entonces los valores de la tabla 7 se utilizarán.

Tabla 8. Valores por defecto de VID y User Prority

Service	Default VID	Default priority
GOOSE	0	4
GSE	0	1
Sampled Values	0	4

Fuente: IEC 61850 part 8-1: Communication Networks and Systems in Substations – Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3

CFI (Canonical Format Indicator): Para este estándar el valor de CFI esta configurado con un valor en cero (0).

VID: El uso de soporte para Virtual LAN es opcional. Si este mecanismo se utiliza el identificador de VLAN (VID), se fijará según la configuración, si no se utiliza, se pondrá en cero (0).

Ethertype: Los valores de Ethertype son basados en la norma ISO/IEC 8802-3. Los valores asignados para GOOSE, GSE y Sampled Values se definen en la tabla 9.

Tabla 9. Asignación de valores Ethertype

Use	Ethertype value (hexadecimal)	APPID type
IEC 61850-8-1 GOOSE	88-B8	0 0
IEC 61850-8-1 GSE Management	88-B9	0 0
IEC 61850-9-2 Sampled Values	88-BA	0 1

Fuente: IEC 61850 part 8-1: Communication Networks and Systems in Substations – Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3

APPID (application identifier): El APPID es usado para seleccionar las tramas que contienen GSE y mensajes GOOSE. El valor APPID es la combinación del APPID Type definido como los dos bits más significativos (Como los define la Tabla 9) y el ID actual.

El APPID tiene un rango entre 0x0000 y 0x3ffff. Si APPID no esta configurado, el valor por defecto es 0x0000. El valor por defecto se reserva para indicar la falta de configuración.

Los campos definidos como Reserved1 y Reserved2 están reservados para futuras aplicaciones y por defecto se colocan en 0.

1.2.12.3 Selección de la dirección de multidifusión[32]

Las direcciones de multidifusión utilizadas en la norma IEC 61850 deberán tener la siguiente estructura:

- Los tres primeros octetos son asignados por el IEEE con el 01-0C-CD.

[32] IEC 61850 part 8-1: Communication Networks and Systems in Substations – Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3

- El cuarto octeto será 01 para GOOSE, 02 para GSSE y 04 para SV (Sampled Values).
- Los últimos dos octetos se utilizan como direcciones individuales asignadas por el rango definido en la tabla 10.

Tabla 10. Ejemplo recomendado de direccionamiento multicast

Servicio	Rango de direcciones recomendadas	
	Dirección Inicial (Hexadecimal)	Dirección Final (Hexadecimal)
GOOSE	01-0C-CD-01-00-00	01-0C-CD-01-01-FF
GSSE	01-0C-CD-02-00-00	01-0C-CD-02-01-FF
Multicast Sampled Values	01-0C-CD-04-00-00	01-0C-CD-04-01-FF

Fuente: IEC 61850 part 8-1: Communication Networks and Systems in Substations – Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3

1.2.12.4 PDU GOOSE

El PDU (Unidad de datos de protocolo) GOOSE está compuesto por los siguientes campos, definidos en el anexo A de la norma IEC 61850 part 8-1: Communication Networks and Systems in Substations – Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3:

A continuación se hace la representación del PDU GOOSE mediante ASN.1:

```
IECGoosePdu ::= SEQUENCE {
        gocbRef                 [0] IMPLICIT VISIBLE-STRING,
        timeAllowedtoLive       [1] IMPLICIT INTEGER,
        datSet                  [2] IMPLICIT VISIBLE-STRING,
        goID                    [3] IMPLICIT VISIBLE-STRING OPTIONAL,
        t                       [4] IMPLICIT UtcTime,
        stNum                   [5] IMPLICIT INTEGER,
        sqNum                   [6] IMPLICIT INTEGER,
```

```
        test                    [7] IMPLICIT BOOLEAN DEFAULT FALSE,
        confRev                 [8] IMPLICIT INTEGER,
        ndsCom                  [9] IMPLICIT BOOLEAN DEFAULT FALSE,
        numDatSetEntries        [10] IMPLICIT INTEGER,
        allData                 [11] IMPLICIT SEQUENCE OF Data,
        security                [12] ANY OPTIONAL,
}
```

Cada campo describe alguna característica del mensaje o da indicaciones del estado del mismo, como se explica enseguida:

- **gocbRef:** Este atributo muestra el nombre de la ruta de un bloque de control GOOSE en el LLN0 y posee la siguiente estructura general: LDName/LLN0.GoCBName.
- **timeAllowedtoLive:** Informa al receptor el tiempo máximo de espera de la siguiente re-trasmisión. Las unidades de este atributo deberán ser dadas en milisegundos. Este atributo puede tomar un valor de 1 a 4 294 967 295.
- **datSet**: El valor de este atributo será el mismo que se encuentra en bloque de control GOOSE especificado por *gocbRef*. Difiere en la estructura general: LDName/LLN0.
- **goID:** El valor de este atributo será el mismo que se encuentra en bloque de control GOOSE especificado por *gocbRef.* Con la diferencia que tendrá la siguiente estructura general: LDName.
- **t:** Este atributo representa el tiempo en el cual el mensaje fue enviado. El formato es el siguiente: "*mes día, año hora:minutos:segundos UTC*"
- **stNum:** El parámetro *stNum* contendrá un contador que aumenta cada vez que un mensaje GOOSE se ha enviado y un cambio de valor se ha detectado en el conjunto de datos especificado por *datSet*. Este atributo tendrá un valor entero de 1 a 4 294 967 295. El valor 0 se reservará.
- **sqNum:** Este parámetro contendrá un contador que se incrementará cada vez que se ha enviado un mensaje GOOSE. Tendrá un valor entero de 0 a

4 294 967 295. El valor 0 esta reservado para primera transmisión de un cambio *stNum*. *sqNum* se incrementa por cada transmisión.

- **test:** Este parámetro indica con el valor TRUE que los valores del mensaje no se utilizan para fines operativos. Tendrá un valor booleano de FALSE o TRUE.
- **confRev**: Este atributo representa un recuento del número de veces que la configuración del DATA-SET ha sido cambiada. El contador se incrementa cuando hay cambios en la configuración. Puede tomar valores de 0 a 4 294 967 295.
- **ndsCom**: Este atributo tiene un valor TRUE si el *datSet* tiene un valor NULL. Se utiliza para indicar que el GoCB requiere una configuración adicional.
- **numDatSetEntries:** Este parámetro especifica el número de miembros de la NamedVariableList MMS que se especifica en el bloque de control GOOSE que controla el servicio GOOSE.
- **allData:** Este parámetro contiene una lista de la información definida por el usuario de la NamedVariableList MMS que se especifica en el bloque de control GOOSE.

NOTA: La clase “control de bloque GOOSE o GoCB por su definición en ingles” es la reunión de una serie de atributos y servicios relacionados con los mensajes de tipo GOOSE.[33]

[33] IEC 61850 part 7-2: Communication Networks and Systems in Substations – Basic communication structure for substation and feeder equipment – Abstract communication service interface (ACSI)

1.3 ATAQUE INFORMÁTICO

"Un ataque informático es un método por el cual un individuo, mediante un sistema informático, intenta tomar el control, desestabilizar o dañar otro sistema informático (computador, red privada, etcétera)."[34]

1.3.1 Tipos de ataques informáticos

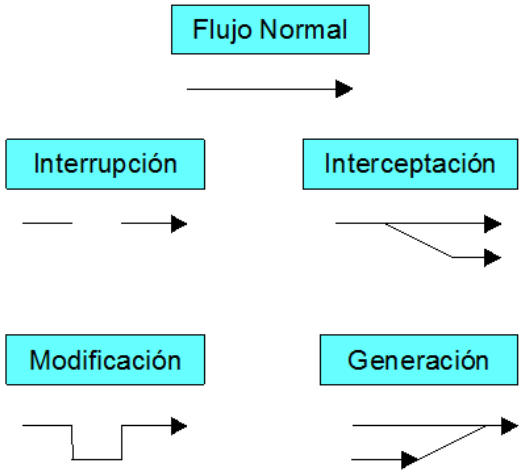
A la hora de estudiar los distintos tipos de ataques informáticos, hay que diferenciar entre los ataques activos, que producen cambios en la información y en la situación de los recursos del sistema, y los ataques pasivos, que se limitan a registrar el uso de los recursos y/o a acceder a la información guardada o transmitida por el sistema.

Entre los distintos tipos de ataques en una red de computadores tenemos fenómenos de:

- Interrupción: Se daña, pierde o deja de funcionar un punto del sistema.
- Interceptación: Acceso a la información por parte de personas no autorizadas. Uso de privilegios no adquiridos.
- Modificación: Acceso no autorizado que cambia el entorno para su beneficio
- Generación: Creación de nuevos objetos dentro del sistema.

[34] Tomado de: http://es.wikipedia.org/wiki/Ataque_inform%C3%A1tico

Figura 10. Tipos de ataques informáticos



Fuente: Autor

1.4 PHYTON

"Python es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico, es fuertemen te tipado y multiplataforma". [35]

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1.

[35] Tomado de: http://www.python.com.co/

Algunas de sus características más relevantes son[36]:

- Sintaxis clara y legible.
- Orientación a objetos intuitiva.
- Modularidad completa, soporte a paquetes jerárquicos.
- Muy alto nivel de los tipos de datos dinámicos.
- Extensas bibliotecas estándar y módulos de terceros prácticamente para todas las tareas.
- Extensiones y módulos fácilmente escritos en C y C++.
- Integrable dentro de las aplicaciones como una interfaz scripting.

1.4.1 Scapy

Scapy es un potente programa interactivo de la manipulación de paquetes, escrito en Python por Philippe Biondi. Con este se pueden modificar o decodificar paquetes de un gran numero de protocolos, para enviarlos y capturarlos. También puede manejar tareas tales como: scannnig, tracerouting, unit tests, ataques y detección de redes.[37]

> Con Scapy no es necesario aprender un lenguaje de programación nuevo, solamente es necesario tener conocimientos sobre la sintaxis y estructura de los programas escritos en Python. Su uso es simple, puede hacerse de forma interactiva (del mismo modo que se hace con el intérprete de Python) o directamente desde una rutina completa de código utilizando un fichero (*.py*). Por otro lado, el funcionamiento principal de Scapy también es muy simple y puede resumirse en dos pasos claramente definidos, la creación de un conjunto de paquetes (1 o muchos) y la captura de respuestas

[36] Tomado de: http://www.python.org/about/
[37] Tomado de: http://www.secdev.org/projects/scapy/

emitidas desde el destinatario de dichos paquetes, es así de simple, dichas respuestas posteriormente pueden ser analizadas y tratadas.

También es posible crear cualquier tipo de paquete especificando valores para cada uno de los campos disponibles en dicho tipo de paquete, así como también crear un tipo de paquete nuevo que herede y extienda características propias de un tipo determinado (por ejemplo TCP, UDP, ICMP, etc.). Dado que la herramienta se encuentra escrita en Python su uso resulta muy simple, y con muy pocas líneas de código pueden hacerse rutinas que en otros lenguajes pueden resultar complejas.[38]

Hay que resaltar que scapy permite el envío de tramas no válidas y la creación de nuevas.

1.5 WIRESHARK

"Wireshark, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones de datos, para desarrollar software y protocolos, y como una herramienta didáctica para la educación. Cuenta con todas las características estándar de un analizador de protocolos".[39]

"Wireshark implementa una amplia gama de filtros que facilitan la definición de criterios de búsqueda para los más de 1100 protocolos soportados hasta la versión 1.4.3; y todo ello por medio de una interfaz sencilla e intuitiva que permite desglosar por capas cada uno de los paquetes capturados. Gracias a que Wireshark interpreta la estructura de los protocolos, se pueden visualizar los campos de cada una de las cabeceras y capas que componen los paquetes

[38] Tomado de: http://thehackerway.com/2012/05/07/manipulacion-de-paquetes-utilizando-scapy-conceptos-basicos-sobre-scapy-parte-i/
[39] Tomado de: http://es.wikipedia.org/wiki/Wireshark

monitorizados, proporcionando un gran abanico de posibilidades al administrador de redes a la hora de abordar varias tareas en el análisis de tráfico".[40]

1.6 OSTINATO [41]

Ostinato es una herramienta de código abierto, multiplataforma que permite generar paquetes de red, además de generar y analizar tráfico de red, con una interfaz gráfica amigable. Permite al usuario enviar paquetes de diferentes protocolos a velocidades distintas.

Entre las características principales de este software encontramos:

a) Se ejecuta en Windows, Linux, BSD y Mac OS X.
b) Soporta los protocolos estándares más comunes:
    a. Ethernet/802.3/LLC SNAP.
    b. VLAN.
    c. ARP, IPv4, IPv6.
    d. TCP, UDP, ICMPv4, ICMPv6 IGMP, MLD.
    e. Cualquier texto basado en protocolo (HTTP, SIP, RTSP, NNTP etc.).
c) Modificar cualquier campo de cualquier protocolo.
d) Crear y configurar múltiples flujos.
e) Configurar las tasas de flujo y el número de paquetes.

1.7 BACKTRACK [42]

BackTrack es una distribución GNU/Linux en formato LiveCD[43] pensada y diseñada para la auditoría de seguridad y relacionada con la seguridad informática

[40] Tomado de: http://cert.inteco.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seguridad_analisis_trafico_wireshark.pdf
[41] Tomado de: http://code.google.com/p/ostinato/
[42] Tomado de: http://es.wikipedia.org/wiki/BackTrack

en general. Actualmente tiene una gran popularidad y aceptación en la comunidad que se mueve en torno a la seguridad informática y se encuentra en la versión 5, entrega 3 Final (Kernel 3.2.6).

Incluye una larga lista de herramientas de seguridad listas para usar, entre las que destacan numerosos escáner de puertos y vulnerabilidades, husmeadores (sniffers), herramientas de análisis forense y herramientas para la auditoría comunicación inalámbrica.

Backtrack le ofrece al usuario una extensa colección de herramientas completamente usables desde un Live CD o un Live USB por lo que no requiere una instalación para poder utilizarse. O bien, se ofrece la opción de instalar en un disco duro. Entre las herramientas ofrecidas se encuentran:

a) Aircrack-ng: Herramientas para auditoría inalámbrica.
b) Kismet: Sniffer inalámbrico.
c) Ettercap: Interceptor/Sniffer/Registrador para LAN.
d) Wireshark: Analizador de protocolos.
e) Medusa: Herramienta para Ataque de fuerza bruta.
f) Nmap: Rastreador de puertos.

[43] Una distribución live o Live CD o Live DVD, es un sistema operativo almacenado en un medio extraíble, tradicionalmente un CD o un DVD (de ahí sus nombres), que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro de una computadora.

2. PROGRAMACIÓN DEL ENVÍO DE MENSAJES RÁPIDOS MEDIANTE EL PROTOCOLO GOOSE

La finalidad del programa explicado a continuación será el envío de mensajes rápidos cumpliendo el protocolo GOOSE.

El lenguaje de programación elegido para esta tarea fue Python 2.7.1, por traer disponible una herramienta que facilita la elaboración y manipulación de paquetes de red, Scapy 2.1.0, por ser un lenguaje con una sintaxis muy limpia que favorece un código legible y por tener disponible gran variedad de bibliotecas que favorecen el uso del lenguaje de programación.

Python y Scapy fueron instalados sobre el sistema operativo Ubuntu 11.04, gracias a que éste facilitaba la instalación y ejecución de los mismos. Para realizar el proceso de instalación adecuado se consultaron las fuentes de los desarrolladores[44].

Después de la instalación correcta, las dos herramientas pueden ser ejecutadas en el terminal mediante el comando *python* para ejecutar el lenguaje o *scapy* para la herramienta.

La idea general que gobernó la programación del envío de mensajes rápidos mediante protocolo GOOSE fue realizar un paquete de datos codificados llamado *Packet_Goose* que se añadiría al final a una trama Ethernet (IEEE 802.1Q). Este paquete fue obtenido gracias a un proceso de consulta de la norma IEC 61850 y otras fuentes bibliográficas citadas en este documento y al análisis de un archivo ofrecido en la página web de Wireshark[45], el cual muestra unos mensajes GOOSE

[44] Consultar en: http://www.secdev.org/projects/scapy/doc/installation.html y http://www.python.org/getit/
[45] Consultar en: http://wiki.wireshark.org/Protocols/IEC61850GOOSEGSE

que cumplen con las especificaciones del protocolo. Este archivo de extensión *.pcap* fue examinado con detalle byte a byte con el fin de obtener toda la información adicional o complementaria a la norma IEC 61850.

Como se mencionó en el marco teórico el protocolo GOOSE se compone solo de cuatro capas del modelo OSI: Aplicación, Presentación, Enlace de datos y Física; esta última no se programó por ser la capa encargada de definir las características físicas por las cuales viajará el mensaje. Los procedimientos realizados para la programación de cada capa son explicados a continuación.

Para la capa de Aplicación detallada en la norma IEC 61850-8-1, se predefinieron una serie de mensajes siguiendo la norma; fue realizado de esta manera porque el contenido de los mensajes GOOSE, es producido por los IEDs dentro de la subestación.

Los mensajes elaborados fueron diseñados siguiendo la sintaxis presentada en la norma IEC 61850-8-1, la cual define que el mensaje en su primera parte debe estar compuesto por el nombre del dispositivo físico "*IED007"* y el nombre del dispositivo lógico "*LD0"*, separado con una barra inclinada "/" de la segunda parte. La siguiente sección del mensaje empieza con el nombre nodo lógico *"$$XCBR$$"* (encerrado por los caracteres $$, que son indicadores del nodo lógico), seguido de un punto, el nombre de la clase de dato "*Pos*", seguido de un punto y finalmente el nombre del atributo del dato *"stVal"*. Dando como resultado un mensaje del siguiente tipo:

IED007LD0/$$XCBR$$.Pos.stVal

Este mensaje fue utilizado para todas las pruebas realizadas en el Anexo C.

Para realizar la capa de Presentación la cual utiliza ASN.1 y BER, dentro del paquete de datos elaborado se codificaron los mismos utilizando la librería *struct* de Python que permite llevar a cabo esta tarea, ejecutándola a todos los componentes del mensaje, desde la cabecera hasta todas las variables definidas en el PDU GOOSE.

Para llevar a cabo la última capa programable del protocolo, la de Enlace de datos, se utilizó una serie de comandos de la libreria Scapy para formar la trama IEEE 802.1Q, estas funciones fueron *Ether()* y *Dot1Q().* La primera sirve para elaborar una trama Ethernet Normal, como la expuesta en la parte superior de la Figura 1, en donde se definen campos como las direcciones de destino y de origen y el campo *TPID* (predeterminado como 0x8100 para la trama IEEE 802.1Q); y la segunda instrucción sirve para definir la trama IEEE 802.1Q, específicamente los campos exclusivos de esta norma como *User Priority* y *Ethertype*, por nombrar algunos ejemplos.

Como se había mencionado con anterioridad 802.1Q añade campos extras a la trama de Ethernet Normal. Entonces para lograr armar la trama 802.1Q completa, con todos sus campos, se hizo la integración de las mismas mediante el operador de composición[46] "/" definido por Scapy, logrando así la trama IEEE 802.1Q como se muestra en la parte inferior de la Figura 1.

Teniendo definidas todas las capas del protocolo y las respectivas codificaciones como son precisadas en la norma, el último paso a seguir fue la composición del paquete GOOSE (*Packet),* la cual se ejecutó mediante el operador de composición "/", teniendo como resultado un mensaje GOOSE, como lo indica la norma.

[46] El operador de composición "/" sirve para encapsular el argumento del lado derecho en el del lado izquierdo.

Finalmente para el envío del paquete GOOSE (Con nombre: *Packet*) se utilizó la instrucción *sendp()* de Scapy, la cual ayuda a enviar paquetes modificados desde la capa de Enlace de datos.

NOTA: Se debe tener en cuenta que para la ejecución del comando *sendp(),* utilizado para el envío de paquetes realizados por el usuario, este debe iniciar sesión como superusuario, para lograr todos los privilegios que esta cuenta tiene.

La programación anteriormente explicada se guardó con el nombre de *goose.py*, siendo *py* la extensión de un archivo del lenguaje Python. Para luego ser ejecutada desde el terminal de Ubuntu por medio de la instrucción *python goose.py.*

Para elaborar la interfaz gráfica del programa se utilizó *glade*. Glade es una herramienta RAD (Rapid Application Development) que permite el desarrollo rápido y fácil de interfaces de usuario. Los diseños hechos en Glade se guardan como XML y mediante el uso del objeto GtkBuilder pueden ser cargados en numerosos lenguajes de programación como C, C++, Python y otros.

El código fuente puede ser consultado en el Anexo A.

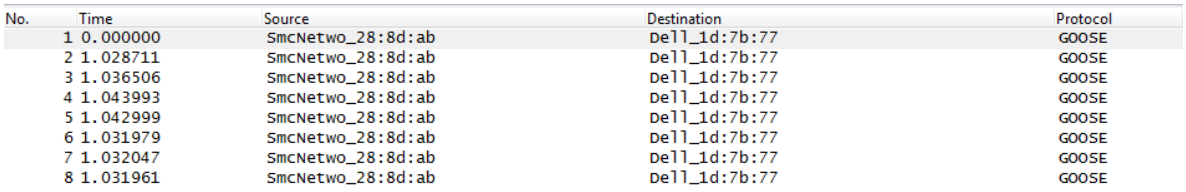
2.1 PRUEBA DEL CUMPLIMIENTO DEL PROTOCOLO GOOSE

Para realizar la comprobación del protocolo GOOSE se realizó un experimento que consistió en el envío de 100 mensajes GOOSE (unicast) de un PC a otro, en una topología sencilla. Esta prueba se documenta y detalla en el Anexo C – Experimento 1. Se concluyó que el programa realizado cumple con el protocolo y la estructura del mensaje GOOSE.

Es de vital importancia recordar que el programa Wireshark muestra el protocolo por el cual es enviado determinado mensaje y además desglosa la información de los diferentes parámetros existentes en el mismo y si el mensaje capturado por Wireshark no cumple con lo establecido en un determinado protocolo, el mismo muestra el siguiente mensaje: *Malformed Packet: Nombre del protocolo.*

En la Figura 11, sección derecha de la imagen se puede apreciar como cada mensaje enviado cumple con el protocolo GOOSE.

Figura 11. Comprobación del protocolo GOOSE

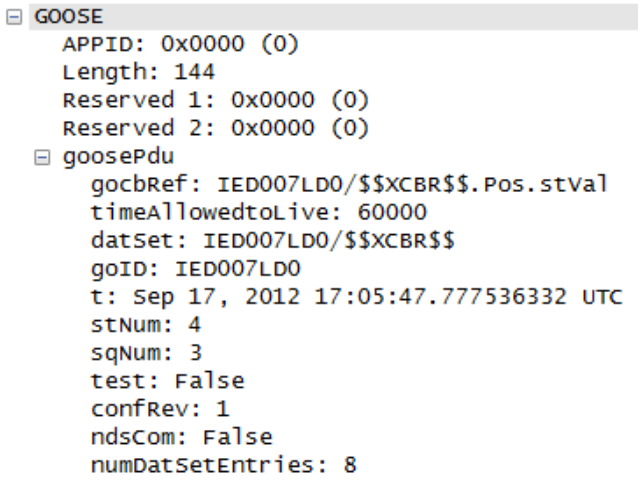


No.	Time	Source	Destination	Protocol
1	0.000000	SmcNetwo_28:8d:ab	Dell_1d:7b:77	GOOSE
2	1.028711	SmcNetwo_28:8d:ab	Dell_1d:7b:77	GOOSE
3	1.036506	SmcNetwo_28:8d:ab	Dell_1d:7b:77	GOOSE
4	1.043993	SmcNetwo_28:8d:ab	Dell_1d:7b:77	GOOSE
5	1.042999	SmcNetwo_28:8d:ab	Dell_1d:7b:77	GOOSE
6	1.031979	SmcNetwo_28:8d:ab	Dell_1d:7b:77	GOOSE
7	1.032047	SmcNetwo_28:8d:ab	Dell_1d:7b:77	GOOSE
8	1.031961	SmcNetwo_28:8d:ab	Dell_1d:7b:77	GOOSE

Fuente: Autor

En la Figura 12 se muestra el PDU GOOSE con toda la información pertinente en cada parámetro del mensaje, acatando con la estructura del mismo, sin mostrar el error *Malformed Packet: GOOSE.*

Figura 12. Comprobación de la estructura del mensaje GOOSE



Fuente: Autor

2.2 ANÁLISIS DE UN MENSAJE GOOSE

A continuación se mostrará un mensaje GOOSE visto en formato hexadecimal y comentado byte a byte, en donde se puede ver el significado de cada conjunto de bytes.

Figura 13. Mensaje GOOSE en hexadecimal

```
01 0c cd 01 00 00 00 22  2d 28 8d ab 88 b8 00 00
00 90 00 00 00 00 61 81  86 80 1c 49 45 44 30 30
37 4c 44 30 2f 24 24 58  43 42 52 24 24 2e 50 6f
73 2e 73 74 56 61 6c 81  03 00 ea 60 82 12 49 45
44 30 30 37 4c 44 30 2f  24 24 58 43 42 52 24 24
83 09 49 45 44 30 30 37  4c 44 30 84 08 50 58 90
9e c7 0c 9f fd 85 01 03  86 01 02 87 01 00 88 01
01 89 01 00 8a 01 08 ab  20 83 01 00 84 03 03 00
00 83 01 00 84 03 03 00  00 83 01 00
```

Fuente: Autor

Cada parámetro del mensaje GOOSE se relaciona a continuación, con su respectivo conjunto de bytes:

- **Dirección de destino:** (01 0c cd 01 00 00)
- **Dirección de origen:** (00 22 2d 28 8d ab)
- **Ethertype:** (88 b8)
- **ID de aplicación o APPID:** (00 00)
- **Longitud o Length:** (00 90)
- **Reserved 1:** (00 00)
- **Reserved 2:** (00 00)
- **APDU longitud:** (61 81 86) El byte 23 "61" está asignado por APDU. "81 86" es la longitud de datos de los comandos APDU.

Se puede notar que en la siguiente sección del mensaje inicia el PDU GOOSE allí se ve la estructura de la codificación BER (tipo-longitud-valor), explicada anteriormente. En cada parámetro mostrado a continuación el primer byte corresponde al tipo de dato, el segundo a la longitud de bytes que posee el mismo y el resto bytes corresponden al valor dado para ese parámetro.

Considérese un ejemplo, en el parámetro *gocbRef*; el primer byte (80) corresponde al tipo de dato, el segundo a la longitud (1c). En este caso particular se está indicando que el campo valor posee 28 bytes. Finalmente desde el tercer byte (49) al final (6c) es el campo valor, cuya significación es el mensaje "*IED007LD0/$$XCBR$$.Pos.stVal*".

- **gocbRef:** (80 1c 49 45 44 30 30 37 4c 44 30 2f 24 24 58 43 42 52 24 24 2e 50 6f 73 2e 73 74 56 61 6c)
- **timeAllowedtoLive:** (81 03 00 ea 60)
- **dataSet:** (82 12 49 45 44 30 30 37 4c 44 30 2f 24 24 58 43 42 52 24 24)
- **goID:** (83 09 49 45 44 30 30 37 4c 44 30)
- **t:** (84 08 50 58 90 9e c7 0c 9f fd)
- **stNum:** (85 01 03)

- **sqNum:** (86 01 02)
- **test:** (87 01 00)
- **confRev:** (88 01 01)
- **ndsCom:** (89 01 00)
- **numdatSetEntries:** (ab 20 83 01 00 84 03 03 00 00 83 01 00 84 03 03 00 00 83 01 00)

El significado de cada campo para el ejemplo mencionado es el siguiente:

- **timeAllowedtoLive:** Corresponde al tiempo máximo de espera de la siguiente retrasmisión el cual se definió en: 60000ms.
- **datSet:** Mensaje definido previamente, siguiendo la sintaxis de la norma: IED0007LD0/$$XCBR$$.
- **goID:** Mensaje definido previamente, siguiendo la sintaxis de la norma: IED0007LD0.
- **t:** Indica el tiempo en el cual el mensaje fue enviado, para este ejemplo corresponde a la siguiente fecha, en su formato prestablecido: Sep 18, 2012 15:17:45.7775363 UTC
- **stNum:** Contador del mensaje, en este caso indica que el mensaje es el número 3.
- **sqNum:** Contador del mensaje, en este caso indica que el mensaje es el número 2.
- **test:** Parámetro definido como Falso.
- **confRev:** Predefinido como el valor 1.
- **ndsCom:** Parámetro definido como Falso.
- **numdatSetEntries: :** Predefinido como el valor 8.

3. EVALUACIÓN DE LA TRANSMISIÓN DE MENSAJES RÁPIDOS SEGÚN LA NORMA IEC 61850

Para realizar la evaluación de la transmisión de mensajes rápidos en una red local se realizaron una serie de experimentos con los cuales se pudo determinar el cumplimiento del tiempo estipulado en la norma IEC 61850-5. En esta se consigna que los mensajes rápidos deben transmitirse en un tiempo mínimo de entre 3 y 10 ms, dependiendo del tipo de subestación, menor a 3ms para una subestación con clase de desempeño P1 y menor a 10ms para una subestación con clase de desempeño P2 o P3.

Estos experimentos se ejecutaron sobre una diversidad de escenarios para observar si existían diferentes tipos de comportamientos en la transmisión de los mensajes rápidos y para medir el tiempo de transferencia de los mismos entre los computadores que intervenían en determinado experimento. Para llevar a cabo esta medición se utilizó el software Wireshark, porque éste informa al usuario el tiempo de todos los mensajes que captura. Además, ayudó a observar el comportamiento de la red mientras viajaban los mensajes rápidos a través de ésta.

Las variaciones realizadas a los experimentos se hicieron: en el tipo de transmisión: unicast y multicast; en el tipo de configuración de la red local: con y sin VLAN; en la cantidad de computadores que producían mensajes GOOSE y en el tráfico de red: ninguno, bajo, medio y alto. Todos los experimentos realizados se encuentran detallados en el Anexo C (Informes de los experimentos realizados para la evaluación de mensajes rápidos en una red local).

Para generar tráfico de red se utilizó el software Ostinato el que permitió realizar variedad de tipos de tráficos en la red de datos local, presentada en cada experimento propuesto en el Anexo C.

El software Ostinato permite elaborar paquetes de red de la mayoría de protocolos de comunicación conocidos (Ethernet, TCP, UDP etc.) y enviarlos a través de un puerto de red, permitiendo además la modificación del número de paquetes que pueden ser enviados a través de la red, la alteración de parámetros de los mismos y la velocidad con la cual los mensajes viajan a través de cierta red, dada en: Paquetes/Segundos.

En la Tabla 11 se presenta la cantidad de mensajes de cada protocolo y los niveles de tráfico propuestos: Bajo, medio y alto. Estos niveles fueron definidos de la manera como se presentan en la tabla 10 porque al interior de una subestación eléctrica automatizada, hay bajos niveles de tráfico. Ello es debido a que el tipo de red manejada al interior de la subestación es dedicada solamente al uso propio de la misma. Por lo que el tráfico dentro de la subestación tendrá niveles bajos a comparación de una topología de red más grande, elaborada y con conexión a Internet.

Tabla 11. Tráfico de red

Protocolo	Porcentaje	Tráfico Bajo	Tráfico Medio	Tráfico Alto
Ethernet II	25%	125	500	2000
Ethernet VLAN	25%	125	500	2000
TCP	20%	100	400	1600
UDP	20%	100	400	1600
IPv4	10%	50	200	800
Total de mensajes		**500**	**2000**	**8000**

Fuente: Autor

La velocidad de cada nivel de tráfico fue definida de la siguiente manera: Tráfico bajo: 1 paquete/segundo; tráfico medio: 4 paquetes/segundo; tráfico alto: 16 paquetes/segundo. Se delimitó de esta forma dado que la mayoría de los experimentos realizados se basaron en el envío de 500 y 1000 mensajes GOOSE con velocidad de 1 mensaje/segundo (velocidad definida por la norma IEC 61850-

5 Anexo B.3)[47]. Por lo tanto se definió que un tráfico bajo es la misma cantidad de mensajes enviados, el tráfico medio es doble del bajo y tráfico alto es el doble del tráfico medio.

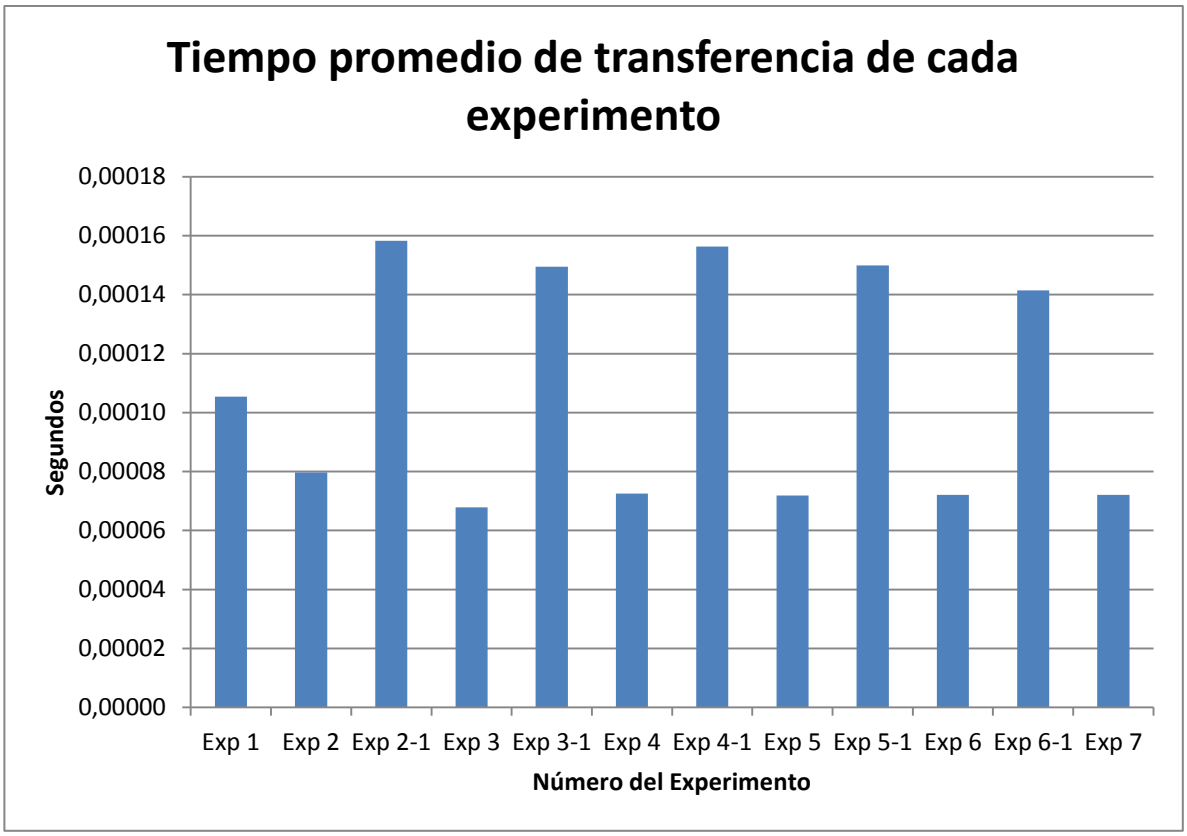
Adicionalmente se realizó un experimento con una velocidad de tráfico de 500 paquetes/segundo (Protocolo: Ethernet II) y una totalidad de 250000 mensajes enviados (Experimento 7). Este se efectuó con el fin de verificar la correcta transmisión de mensajes GOOSE ante un tráfico de red más alto que el presentado en la Tabla 11.

A lo anterior hay que añadir que en el actual estudio se realizaron pruebas con niveles más altos que los presentados en la tabla 10 y el experimento 7, tal como se muestra en el experimento A 1-1 en la tabla D1 y que corresponde al anexo D. En los cuales se realizaron pruebas hasta con 10’000.000 de paquetes con una velocidad de 20000 paquetes/segundo.

Después de realizar la totalidad de los experimentos planteados en el Anexo C, en general se obtuvieron resultados exitosos, en los cuales los mensajes se trasmitieron en promedio a un tiempo de transferencia mucho más rápido que el propuesto en la norma IEC 61850-5. En la Figura 14 se puede ver el tiempo promedio de cada experimento, con éste se halló el promedio general para la totalidad de pruebas realizadas, el cual fue de: 0,00011 segundos.

[47] INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 5: Communication Networks and Systems in Substations - Communication requirements for functions and device models.

Figura 14. Tiempo promedio de cada experimento



Fuente: Autor

Igualmente todos los mensajes rápidos enviados a través de cada red local propuesta en cada uno de los experimentos realizados en el anexo C, llegaron exitosamente en su totalidad al receptor que estaba a la espera de los mismos.

Es importante resaltar, que en los experimentos cumplidos con tráfico de red se obtuvieron resultados muy cercanos o iguales, como se puede ver en la tabla 11; los experimentos 4, 5 y 6 realizados con tráfico bajo, medio y alto respectivamente, obtuvieron resultados iguales y en las pruebas 4-1, 5-1 y 6-1 tiempos de transferencia muy cercanos. Esto significa que el protocolo es afectado de manera muy leve por tráficos de red con velocidades inferiores a 500 paquetes/segundo.

Tabla 12. Resumen del tiempo promedio de cada experimento

No. Experimento	Promedio (Seg)
Exp 1	0,000105
Exp 2	0,000080
Exp 2-1	0,000158
Exp 3	0,000068
Exp 3-1	0,000150
Exp 4	0,000072
Exp 4-1	0,000156
Exp 5	0,000072
Exp 5-1	0,000150
Exp 6	0,000072
Exp 6-1	0,000141
Exp 7	0,000072

Fuente: Autor

4. COMPORTAMIENTO DE LA TRANSMISIÓN DE MENSAJES RÁPIDOS EN UNA RED LOCAL ANTE ATAQUES A LA SEGURIDAD

La norma IEC 61850 en toda su extensión hace referencia a algunas amenazas; tan solo en el anexo A de la sección tres (IEC 61850-3) se especifica, que al implementar un SAS se deben efectuar características de seguridad que las contrarresten, ya sea por "denegación del servicio" o "uso ilegítimo[48]", como así se denominan.

De estas dos amenazas se tuvo en cuenta solo el ataque señalado como denegación del servicio, ya que el uso ilegitimo definido en la norma trata sobre los privilegios que tiene el determinado usuario para acceder al sistema implementado para un SAS.

Además del ataque mencionado precedentemente, se hizo la elección de otros dos ataques que complementaron el estudio de la seguridad de la transmisión de los mensajes GOOSE. Esta escogencia se basó en dos criterios: el primero, que los ataques estuvieran suficientemente documentados en diferentes fuentes bibliográficas (tanto a nivel teórico como a nivel técnico); y el segundo, que los ataques fueran basados en la capa 2 (enlace de datos) del modelo OSI. Los ataques a la seguridad escogidos fueron los siguientes:

- Denegación del servicio (DoS)
- CAM Table Overflow o MAC Flooding
- ARP Spoofing

[48] Es cuando un atacante intenta hacer uso del sistema SAS de un modo no autorizado.

4.1 DENEGACIÓN DEL SERVICIO (DoS)

Es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima, o sobrecarga de los recursos computacionales del sistema de la víctima.

4.1.1 Inundación ICMP

Es una técnica DoS que pretende agotar el ancho de banda de la víctima. Consiste en enviar de forma continuada un número elevado de paquetes *ICMP Echo request* (ping) de tamaño considerable a la víctima, de forma que esta ha de responder con paquetes *ICMP Echo reply* (pong) lo que supone una sobrecarga tanto en la red como en el sistema de la víctima.

Dependiendo de la relación entre capacidad de procesamiento de la víctima y el atacante, el grado de sobrecarga varía, es decir, si un atacante tiene una capacidad mucho mayor, la víctima no puede manejar el tráfico generado.

4.1.2 Resultados obtenidos después de ejecutado el ataque a la transmisión de mensajes GOOSE

Después de realizados los diferentes experimentos, desglosados en el anexo D, se pudo observar que el ataque “denegación del servicio” solamente retardó los mensajes GOOSE por unos segundos, entre 2 y 10s. Pero nunca causó corte en la conexión existente entre los PCs (ya que los mensajes se siguieron transmitiendo) que intervenían en las diferentes pruebas realizadas.

4.2 CAM TABLE OVERFLOW O MAC FLOODING

Los switchs guardan las asociaciones MAC – Puerto a medida que las aprenden en una tabla llamada tabla CAM (Content addressable memory). Esta tabla tiene un tamaño fijo y finito (de 256 a más de 16.000 direcciones).

Cuando la tabla CAM no tiene espacio para almacenar más asociaciones MAC – Puerto envía a todos los puertos las tramas que tengan dirección MAC destino no guardada en la tabla CAM.

Para realizar el ataque sólo falta enviar un gran número de tramas con direcciones MAC distintas (usualmente generadas al azar) a cualquier puerto del switch hasta que se llene la tabla CAM.

Este ataque tiene dos efectos adversos:

- Un dispositivo intruso puede conectarse a cualquier puerto del switch y capturar tráfico que normalmente no se ve en ese puerto.
- El tráfico saliente del switch es ineficiente y voluminoso y se puede producir un ataque de denegación del servicio (DoS).

Para realizar el ataque existe una herramienta llamada *macof* que es parte del paquete *dsniff* (GNU/Linux), actualmente incluida en la distribución *BackTrack*.

4.2.1 Resultados obtenidos después de ejecutado el ataque a la transmisión de mensajes GOOSE

De los experimentos realizados, la herramienta *macof* no causó ningún efecto adverso de los descritos anteriormente , los mensajes rápidos enviados en las pruebas (descritas en el anexo D), llegaron de manera exitosa a su receptor sin

ser intervenidos. Además el ataque no ocasionó retardos en la transmisión de mensajes GOOSE.

4.3 ARP SPOOFING

El ARP Spoofing, también conocido como ARP Poisoning o ARP Poison Routing, es una técnica usada para infiltrarse en una red ethernet conmutada (basada en switches y no en hubs), que puede permitir al atacante leer paquetes de datos en la LAN (red de área local), modificar el tráfico, o incluso detenerlo.

El principio del ARP Spoofing es enviar mensajes ARP[49] falsos (falsificados, o spoofed) a la Ethernet. Normalmente la finalidad es asociar la dirección MAC del atacante con la dirección IP de otro nodo (el nodo atacado), como por ejemplo la puerta de enlace predeterminada (gateway). Cualquier tráfico dirigido a la dirección IP de ese nodo, será erróneamente enviado al atacante, en lugar de a su destino real. El atacante, puede entonces elegir, entre reenviar el tráfico a la puerta de enlace predeterminada real (ataque pasivo o escucha), o modificar los datos antes de reenviarlos (ataque activo). El atacante puede incluso lanzar un ataque de tipo DoS (Denegación de Servicio) contra una víctima, asociando una dirección MAC inexistente con la dirección IP de la puerta de enlace predeterminada de la víctima.

[49] ARP Son las siglas en inglés de Address Resolution Protocol (Protocolo de resolución de direcciones). Es un protocolo de la capa de enlace de datos responsable de encontrar la dirección hardware (Ethernet MAC) que corresponde a una determinada dirección IP.

4.3.1 Resultados obtenidos después de ejecutado el ataque a la transmisión de mensajes GOOSE

En el experimento realizado el ataque no afectó la transmisión de los mensajes GOOSE.

Esto sucedió porque la transmisión de los mensajes GOOSE no actualiza la tabla ARP. El protocolo GOOSE no utiliza la capa de red y aunque el protocolo ARP pertenece a la capa 2 (enlace de datos) del modelo OSI, éste se relaciona con la capa 3 (red). Hay que recordar el funcionamiento del protocolo: "ARP, es un protocolo de la capa de enlace de datos responsable de encontrar la dirección hardware (Ethernet MAC) que corresponde a una determinada dirección IP. Para ello se envía un paquete (ARP request) a la dirección de difusión de la red (broadcast MAC = FF:FF:FF:FF:FF:FF) que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina (u otra) responda (ARP reply) con la dirección Ethernet que le corresponde. Cada máquina mantiene una caché con las direcciones traducidas para reducir el retardo y la carga".[50]

Al no afectarse la tabla ARP, no se pueden iniciar el ataque ARP Spoofing y todos los ataques que tienen como base el mismo, ya que al no hallarse las tablas ARP éstos no se pueden realizar.

Conveniente resaltar que durante el experimento se hizo *ping*[51] desde un PC a otro y las tablas ARP si se crearon o se actualizaron, lo cual ocurre porque los mensajes ICMP están enmarcados en la capa 3 y 4 del modelo OSI y el protocolo GOOSE no maneja estas capas.

---

[50] Tomado de: http://es.wikipedia.org/wiki/Address_Resolution_Protocol#Tablas_ARP

[51] Como programa, ping es una utilidad diagnóstica en redes de computadoras que comprueba el estado de la conexión del host local con uno o varios equipos remotos de una red TCP/IP por medio del envío de paquetes ICMP de solicitud y de respuesta.

5. CONCLUSIONES

- Después de la revisión teórica realizada al protocolo GOOSE se elaboró una herramienta implementada en Python que envía mensajes rápidos cumpliendo el protocolo GOOSE a diferentes destinos, tanto a multicast como a unicast.

- El uso lenguaje Python y sus bibliotecas, así como la herramienta Scapy fueron instrumentos importantes en la realización de un programa óptimo con pocas líneas de código.

- La transmisión de mensajes GOOSE es afectada de manera muy leve por el tráfico de red, observándose perturbada sólo cuando el tráfico excedió los límites normales.

- Los mensajes rápidos enviados con la herramienta desarrollada tuvieron un tiempo de transferencia inferior al exigido por la norma IEC 61850-5.

- Los experimentos llevados a cabo con más de un productor de mensajes GOOSE aumentaron el tiempo de transferencia en relación al tiempo que generó un solo productor de mensajes en un factor cercano a los 0.00008 segundos.

- De los ataques a la seguridad escogidos para modificar el envío de los mensajes rápidos solo se obtuvo resultado del ataque “Denegación del servicio”, el cual hizo que los mensajes demoraran más en su tiempo de transmisión.

- Los ataques CAM Table Overflow y ARP Spoofing no ocasionaron ningún efecto adverso a la transferencia de los mensajes GOOSE de un computador a otro.

- Con el actual proyecto se aportó al grupo GISEL de la Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones E3T, un estudio formal de una temática referente a la línea de investigación manejada por el grupo desde un ámbito de la Ingeniería de Sistemas.

- La experiencia de utilizar el lenguaje de programación Python para el autor fue de gran beneficio, pues al utilizarlo durante la realización del proyecto, lo instruyó sobre un lenguaje con gran potencial, con una sintaxis muy limpia y de fácil aprendizaje.

- Las consultas e investigaciones realizadas, permitieron al autor afianzar y adquirir nuevos conocimientos, no solo en el área de redes de computadores sino en la norma IEC 61850 y su manejo, herramientas importantes para ser más competitivo.

6. RECOMENDACIONES

- Se propone a la Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones – E3T, la realización de un proyecto de grado enfocado al análisis de un dispositivo electrónico inteligente (IED) y la relación de éstos con los diferentes protocolos de comunicación propuestos en la norma IEC 61850, con el fin de avanzar a una segunda fase el actual trabajo.

- Se sugiere a la Escuela de Ingeniería de Sistemas e Informática, desarrollar un proyecto de grado para modelar el funcionamiento de diversos IEDs (dispositivo electrónico inteligente) y de varios dispositivos eléctricos que integran una subestación, siguiendo las directrices de la norma IEC 61850 y elaborar la programación necesaria para llevar a cabo la comunicación de estos IEDs a través de una red de área local.

- Efectuar un proyecto de grado encaminado hacia la misma temática propuesta en la presente investigación, pero realizando pruebas con equipos IED (dispositivo electrónico inteligente) reales, ya que las pruebas efectuadas en esta investigación fueron ejecutadas en escenarios ideales y controlados de laboratorio.

- El IEC ha anunciado que a comienzos del año 2013 publicará la segunda edición de la norma IEC 61850-5 y otras normas del juego IEC 61850 y que actualmente se encuentra en procesos de votación para su plena aprobación. Esa nueva edición en su interior presenta la adición de nuevas capas del modelo OSI a los protocolos de comunicación (GOOSE y SV), por esto se propone realizar un estudio formal acerca de dicha normatividad analizando los efectos que pueden tener los protocolos ante esos cambios.

7. BIBLIOGRAFÍA

1. AGUDELO HERNÁNDEZ, Laura Liliana. FLÓREZ ACOSTA, Ovidio Alfonso. Evaluación de desempeño de las tecnologías Power Line Carrier (PLC) y Wireless LAN para la implementación al prototipo de sistema metropolitano de telemetría para la monitorización de parámetros eléctricos en las redes de distribución de media y baja tensión. Bucaramanga, 2010, 104h. Trabajo de Grado (Ingeniera Electrónica e Ingeniero Electrónico). Universidad Industrial de Santander. Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones.

2. ALI, Ikbal. THOMAS, Mini. GOOSE based protection scheme implementation & testing in laboratory. Disponible en: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5759133&contentType=Conference+Publications&sortType%3Dasc_p_Sequence%26filter%3DAND(p_IS_Number%3A5759123)>

3. CEDIEL MARTINEZ, Julián Andrés. CHAPARRO RESTREPO, Jaime Hernán. Aspectos generales en la automatización de una subestación eléctrica y empleo del protocolo de comunicación IEC 61850. Bucaramanga, 2010, 170h. Trabajo de Grado (Ingeniero Electricista e Ingeniero Electrónico). Universidad Industrial de Santander. Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones.

4. DE OLIVEIRA, Julio Cezar. VARELLA, Walter Augusto. MARQUES, Antonio Eduardo. FORSTER, Gustavo. Real time application using multicast ethernet in power substation automation according to IEC61850. En UNICSUL – Universidade Cruzeiro do Sul. Disponible en: <http://www.pacw.org/fileadmin/doc/Multicast_Ethernet_em_Automacao_de_Subestacoes-released-Eng1_.pdf>

5. DINH, Nhat. KIM, Gwan-Su. LEE, Hong-Hee. A study on GOOSE communication based in IEC 61850 using MMS Ease Lite. En University of Ulsan. Korea. Disponible en: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4406651&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D4406651>

6. DU, Liang. LIU, Qun-ying. The design of communication system on te real-time relay protection based on GOOSE. Disponible en: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6307735&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F6304951%2F6306868%2F06307735.pdf%3Farnumber%3D6307735>

7. HERNÁNDEZ SAMPIERI, Roberto. FERNÁNDEZ-COLLADO, Carlos. BAPTISTA LUCIO, Pilar. Metodología de la Investigación. Ricardo A. del Bosque Alayón, Cuarta Edición. México D.F: McGraw-Hill Interamericana Editores S.A, 2006.

8. HONG, Sugwon. SHIN, Dae-Yong. LEE, Seung-Jae. Experimenting security algorithms for the IEC 61850-based substation communication. En Department of Computer Software, Electrical Engineering Myongji University. Disponible en: <http://www.computer.org/portal/web/csdl/doi/10.1109/EmbeddedCom-ScalCom.2009.63>

9. HOYOS, Juan. DEHUS, Mark. BROWN Timthy. Exploiting the GOOSE Protocol: A practical Attack on Cyber-infrastructure. Interdisciplinary Telecommunications program, University of Colorado Boulder USA.

10. INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 1: Communication Networks and Systems in Substations - Introduction and overview. Geneve, Suiza: IEC, 2003. 34h

11. INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 2: Communication Networks and Systems in Substations – Glossary. Geneve, Suiza: IEC, 2003. 48h

12. INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 5: Communication Networks and Systems in Substations - Communication requirements for functions and device models. Geneve, Suiza: IEC, 2003. 34h

13. INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 7-1: Communication Networks and Systems in Substations – Basic communication structure for substation and feeder equipment - Principles and models. Geneve, Suiza: IEC, 2003. 116h.

14. INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 7-2: Communication Networks and Systems in Substations – Basic communication structure for substation and feeder equipment – Abstract communication service interface (ACSI). Geneve, Suiza: IEC, 2003. 178h.

15. INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61850 part 8-1: Communication Networks and Systems in Substations – Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3. Geneve, Suiza: IEC, 2003. 140h.

16. INTERNATIONAL TELECOMMUNICATION UNION. ITU-T X.690: OSI networking and system aspects – Abstract Syntax Notation One (ASN.1) –

Information technology – ASN.1 encoding rules: Specification of basic encoding rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER). Geneve, Suiza: ITU-T, 2002. 36h. Disponible en: <http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf>

17. INTERNATIONAL TELECOMMUNICATION UNION. ITU-T X.680: OSI networking and system aspects – Abstract Syntax Notation One (ASN.1) – Information technology - Abstract Syntax Notation One – ASN.1: Specification of basic notation. Geneve, Suiza: ITU-T, 2008, 194h. Disponible en: <http://www.itu.int/rec/T-REC-X.680-200811-I/en>

18. JIANZHONG, Fan. QIANLI, Ma. GOOSE and its application study. En International Conference on Electrical Enginnering (ICEE). Disponible en: <http://www.icee-con.org/papers/2007/Oral_Poster%20Papers/04/ICEE-195.pdf>

19. LÁZARO LAPORTA, Jorge. MIRALLES AGUIÑIGA, Marcelo. Fundamentos de Telemática, España: Valencia Editorial de la UPV, 2005.

20. NOWAK, Dominik. Bridge between industrial wireless sensor network and wired Ethernet network. Kraków (Poland), 2012, 117h. Master's Thesis (Electronics and Telecommunications). University of Science and Technology. Faculty of Electrical Engineering, Automatics, Computer Science and Electronics. Disponible en: <http://www.smartgrid.agh.edu.pl/pdf/Master_Thesis_DominikNowak.pdf>

21. PÉREZ VILLALÓN, Elena. Diseño y optimización de una arquitectura IEC61850. Madrid, 2008, 269 h. Proyecto de fin de carrera (Ingeniero Industrial). Universidad Pontificia Comillas. Disponible en: <http://www.iit.upcomillas.es/pfc/resumenes/486a07817e45c.pdf>

22. PIIRAINEN, Jukka. Applications of horizontal communication in industrial power networks. Hervanta (Finland), 2010, 80h. Master of Science Thesis (Master's Degree Programme in Electrical Engineering). Tampere University of Technology. Disponible en: <http://webhotel2.tut.fi/units/set/opetus/pdf%20julkiset%20dtyot/Piirainen_Jukka_julk.pdf>

23. PREMARATNE, Upeka. SAMARABANDU, Jagath. SIDHU, Tarlochan. BERESH, Robert. TAN, Jian-Cheng. An intrusion detection system for IEC61850 automated substations. En IEEE Transactions on power delivery, Vol. 25, No. 4. Disponible en: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5570110&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5570110>

24. PREMARATNE, Upeka. SAMARABANDU, Jagath. SIDHU Tarlochan. BERESH Robert. TAN, Jian-Cheng. Security analysis and auditing of IEC61850-based automated substations. En IEEE Transactions on Power Delivery, Vol. 25, No. 4, October 2010. Disponible en: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5451151&contentType=Journals+%26+Magazines&searchField%3DSearch_All%26queryText%3DSecurity+Analysis+and+Auditing+of+IEC61850-Based+Automated+Substations>

25. RIVAS, Eduardo Alonso. Diseño de una plataforma de comunicaciones bajo la norma IEC 61850. Madrid, 2009, 155h. Proyecto de fin de carrera (Ingeniero Industrial). Universidad Pontificia Comillas. Disponible en: <http://www.iit.upcomillas.es/pfc/resumenes/4aa7d65d31770.pdf>

26. STALLINGS, Willliam. Comunicaciones y redes de computadores, Sexta Edición. Prentice Hall, 2005.

27. SEELEY, Nicholas C. Automation at protection speeds: IEC 61850 GOOSE messaging as a reliable, high-speed alternative to serial communications. En Schweitzer Engineering Laboratories, Inc. Disponible en: <http://www.selinc.com/WorkArea/DownloadAsset.aspx?id=3532>

28. VILLARREAL SOLANO, Ariel Yezid. Arquitectura de red local para la transmisión de datos en banda ancha mediante la red eléctrica de baja tensión. Bucaramanga, 2011, 97h. Trabajo de Grado (Maestría en Ingeniería de sistemas e informática). Universidad Industrial de Santander. Escuela de Ingeniería de Sistemas e Informática.

29. YANG, Hyo. KIM, Sang. JANG, Hyuk. Optimized security algorithm for IEC 61850 based power utility system. En Journal of Electrical Engineering & Technology Vol 7, No 3. Disponible en: <http://www.dbpia.co.kr/Journal/ArticleDetail/1922051>

30. YINGYI, Liang. CAMPBELL, Roy H. Understanding and simulating the IEC 61850 standard. En University of Illinois at Urbana-Champaign. 2008. Disponible en: <http://www.ideals.illinois.edu/bitstream/handle/2142/11457/Understanding%20and%20Simulating%20the%20IEC%2061850%20Standard.pdf?sequence=>

31. XIN, Peng. IEC 61850 Testing and documentation. En University of Applied Sciences. Finland. 2010. Disponible en: <http://publications.theseus.fi/bitstream/handle/10024/17035/Peng_Xin.pdf?sequence=1>

8. ANEXOS

ANEXO A: CÓDIGO FUENTE DE LA HERRAMIENTA PARA EL ENVÍO DE MENSAJES RÁPIDOS Y EXPLICACIÓN DE USO DE LA HERRAMIENTA.

Código fuente de la herramienta *goose.py*

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

#Importación de las librerias necesarias

import struct
import time
from scapy.all import *

import sys
try:
    import pygtk
    pygtk.require("2.0")
except:
    pass
try:
    import gtk
except:
    print("GTK Not Available")
    sys.exit(1)

#En la primera sección del programa se codifican los parámetros
correspondientes al PDU GOOSE de acuerdo
#al formato VER, esto se realiza mediante la función struct

def cabecera(APPID , Length, Reserved_1, Reserved_2):
     z0 = struct.pack('!h', APPID)
     z1 = struct.pack('!h', Length)
     z2 = struct.pack('!h', Reserved_1)
     z3 = struct.pack('!B', Reserved_2)

     My_Packet = (z0 + z1 + z2 + z3)
     return My_Packet
```

```
def ini_goose():
     x0 = struct.pack('!h', 97)
     x1 = struct.pack('!B', 129)
     x2 = struct.pack('!B', 134)

     My_Packet = (x0 + x1) + x2
     return My_Packet

def fun_gocbRef(gocbRef):
     a0 = struct.pack('!B', 128)
     a1 = struct.pack('!B', len(gocbRef))
     a2 = gocbRef

     My_Packet = (a0 + a1 + a2)
     return My_Packet

def fun_timeAllowedtoLive(timeAllowedtoLive):

     b0 = struct.pack('!B', 129)
     b2 = struct.pack('!B', 0)

     byte_aux = (len(bin(timeAllowedtoLive)) - 2 ) / 8
     byte = (len(bin(timeAllowedtoLive)) - 2 ) % 8

     if byte >= 0:
          byte_aux += 1

     b1 = struct.pack('!B', byte_aux)

     if timeAllowedtoLive <= 127:
          b1 = struct.pack('!B', 1)
          b3 = struct.pack('!B', timeAllowedtoLive)
          My_Packet = (b0 + b1 + b3)
     elif timeAllowedtoLive > 127 and timeAllowedtoLive <= 65535:
          b3 = struct.pack('!H', timeAllowedtoLive)
          if byte_aux % 2 == 0:
               My_Packet = (b0 + b1 + b3)
          else:
               My_Packet = (b0 + b1 + b2 + b3)
     elif  timeAllowedtoLive  >  65535  and  timeAllowedtoLive  <=
4294967295:
          b3 = struct.pack('!I', timeAllowedtoLive)
          b1 = struct.pack('!B', 4)
```

```
            My_Packet = (b0 + b1 + b3)

      return My_Packet

def fun_datSet(datSet):
      c0 = struct.pack('!B', 130)
      c1 = struct.pack('!B', len(datSet))
      c2 = datSet

      My_Packet = (c0 + c1 + c2)
      return My_Packet

def fun_goID(goID):
      d0 = struct.pack('!B', 131)
      d1 = struct.pack('!B', len(goID))
      d2 = goID

      My_Packet = (d0 + d1 + d2)
      return My_Packet

def fun_t(t1,t2):
      e0 = struct.pack('!B', 132)
      e1 = struct.pack('!B', 8)
      e2 = struct.pack('!i', t1)
      e3 = struct.pack('!f', t2)

      My_Packet = (e0 + e1 + e2 + e3)
      return My_Packet

def fun_stNum(stNum):
      f0 = struct.pack('!B', 133)
      f2 = struct.pack('!B', 0)

      byte_aux = (len(bin(stNum)) - 2 ) / 8
      byte = (len(bin(stNum)) - 2 ) % 8

      if byte >= 0:
            byte_aux += 1

      f1 = struct.pack('!B', byte_aux)

      if stNum <= 127:
            f1 = struct.pack('!B', 1)
            f3 = struct.pack('!B', stNum)
```

```
            My_Packet = (f0 + f1 + f3)
        elif stNum > 127 and stNum <= 65535:
            f3 = struct.pack('!H', stNum)
            if byte_aux % 2 == 0:
                My_Packet = (f0 + f1 + f3)
            else:
                My_Packet = (f0 + f1 + f2 + f3)
        elif stNum > 65535 and stNum <= 4294967295:
            f3 = struct.pack('!I', stNum)
            f1 = struct.pack('!B', 4)
            My_Packet = (f0 + f1 + f3)

        return My_Packet

    def fun_sqNum(sqNum):
        g0 = struct.pack('!B', 134)
        g2 = struct.pack('!B', 0)

        byte_aux = (len(bin(sqNum)) - 2 ) / 8
        byte = (len(bin(sqNum)) - 2 ) % 8

        if byte >= 0:
            byte_aux += 1

        g1 = struct.pack('!B', byte_aux)

        if sqNum <= 127:
            g1 = struct.pack('!B', 1)
            g3 = struct.pack('!B', sqNum)
            My_Packet = (g0 + g1 + g3)
        elif sqNum > 127 and sqNum <= 65535:
            g3 = struct.pack('!H', sqNum)
            if byte_aux % 2 == 0:
                My_Packet = (g0 + g1 + g3)
            else:
                My_Packet = (g0 + g1 + g2 + g3)
        elif sqNum > 65535 and sqNum <= 4294967295:
            g3 = struct.pack('!I', sqNum)
            g1 = struct.pack('!B', 4)
            My_Packet = (g0 + g1 + g3)

        return My_Packet

    def fun_test(test):
```

```
        h0 = struct.pack('!B', 135)
        h1 = struct.pack('!B', 1)
        h2 = struct.pack('!?', test)

        My_Packet = (h0 + h1 + h2)
        return My_Packet

    def fun_confRev(confRev):
        i0 = struct.pack('!B', 136)
        i1 = struct.pack('!B', 1)
        if confRev <= 255:
            i2 = struct.pack('!B', confRev)
        else:
            i2 = struct.pack('!H', confRev)

        My_Packet = (i0 + i1 + i2)
        return My_Packet

    def fun_ndsCom(ndsCom):
        j0 = struct.pack('!B', 137)
        j1 = struct.pack('!B', 1)
        j2 = struct.pack('!?', ndsCom)

        My_Packet = (j0 + j1 + j2)
        return My_Packet

    def fun_numDatSetEntries(numDatSetEntries):
        k0 = struct.pack('!B', 138)
        k2 = struct.pack('!B', 0)

        byte_aux = (len(bin(numDatSetEntries)) - 2 ) / 8
        byte = (len(bin(numDatSetEntries)) - 2 ) % 8

        if byte >= 0:
            byte_aux += 1

        k1 = struct.pack('!B', byte_aux)

        if numDatSetEntries <= 127:
            k1 = struct.pack('!B', 1)
            k3 = struct.pack('!B', numDatSetEntries)
            My_Packet = (k0 + k1 + k3)
        elif numDatSetEntries > 127 and numDatSetEntries <= 65535:
            k3 = struct.pack('!H', numDatSetEntries)
```

```
            if byte_aux % 2 == 0:
                  My_Packet = (k0 + k1 + k3)
            else:
                  My_Packet = (k0 + k1 + k2 + k3)
      elif  numDatSetEntries  >  65535  and  numDatSetEntries  <=
4294967295:
            k3 = struct.pack('!I', numDatSetEntries)
            k1 = struct.pack('!B', 4)
            My_Packet = (k0 + k1 + k3)

      return My_Packet

def fun_allData():
      l0 = struct.pack('!B', 171)
      l1 = struct.pack('!B', 32)
      l2 = struct.pack('!BB', 131,1)
      l21 = struct.pack('!?', False)
      l4 = struct.pack('!BB', 132,3)
      l41 = struct.pack('!Bh', 3,0)

      My_Packet = (l0 + l1 + l2 + l21 + l4 + l41 + l2 + l21 + l4 +
l41 + l2 + l21)
      return My_Packet

#En la función tiempo Se restan 18000 segundos para encontrar la
hora Colombiana
#(UTC - 5horas = Hora Colombiana)

def tiempo():
      t1 = time.time() - 18000
      aux0 = time.gmtime()
      aux1 = time.mktime(aux0)
      t2 = t1 - aux1
      return t1, t2

class GOOSE:
      print ""
      print "                    G O O S E"
      print ""

      def __init__(self, destino, origen, mensajes):

            #Constantes definidas por la norma IEC 61850
            TPID = 0x8100
```

```
            UserPriority = 4
            CFI = 0
            Ethertype = 0x88b8
            APPID = 0

            #Creación de la trama Ethernet
            #Para esta versión v2.0 las variables Et.dst y Et.src
pueden ser modificadas desde la interfaz gráfica
            Et = Ether()
            Et.dst = destino
            Et.src = origen
            Et.type = TPID

            #Creación de la trama IEEE 802.1Q
            vlan = Dot1Q()
            vlan.prio = UserPriority
            vlan.id = APPID
            vlan.vlan = CFI
            vlan.type = Ethertype

            APPID = 0
            Length = 144
            Reserved_1 = 0
            Reserved_2 = 0

            gocbRef = 'IED007LD0/$$XCBR$$.Pos.stVal'
            timeAllowedtoLive = 60000
            datSet = 'IED007LD0/$$XCBR$$'
            goID = 'IED007LD0'
            t1, t2 = tiempo()
            stNum = 1
            sqNum = 0
            test = False
            confRev = 1
            ndsCom = False
            numDatSetEntries = 8

            msn = int (mensajes)

            for aux in range(0,msn):

                  p1    =    cabecera(APPID,    Length,    Reserved_1,
Reserved_2)
                  p2 = ini_goose()
```

```
                p3 = fun_gocbRef(gocbRef)
                p4 = fun_timeAllowedtoLive(timeAllowedtoLive)
                p5 = fun_datSet(datSet)
                p6 = fun_goID(goID)
                p7 = fun_t(t1, t2)
                p8 = fun_stNum(stNum)
                p9 = fun_sqNum(sqNum)
                p10 = fun_test(test)
                p11 = fun_confRev(confRev)
                p12 = fun_ndsCom(ndsCom)
                p13 = fun_numDatSetEntries(numDatSetEntries)
                p14 = fun_allData()

                stNum += 1
                sqNum += 1
                t1,t2 = tiempo()

                Packet_goose = p1 + p2 + p3 + p4 + p5 + p6 + p7 +
p8 + p9 + p10 + p11 + p12 + p13 + p14

                Packet = Et / vlan / Packet_goose
                #Armado del Paquete Goose

                Length = len(Packet)
                #Actualizar lenght del Packet

                hexdump(Packet)
                #Ver el paquete en hexadecimal

                #Packet.show()
                #Ver el la composición del protocolo completo

                sendp(Packet)
                #Enviar mensaje GOOSE

                time.sleep(1)
                #Suspende la ejecución por t segundos

                print ""
                print "Mensaje enviado exitosamente"
                print ""

class Gui:
```

```
      def __init__( self ):
            self.builder = gtk.Builder()
            self.builder.add_from_file("MyInterfaz.ui")

            dic = {
                "on_buttonSalir_clicked" : self.quit,
                "on_buttonEnviar_clicked" : self.enviar,
                "on_buttonAyuda_clicked" : self.ayuda,
                "on_windowMain_destroy" : self.quit,
            }

            self.builder.connect_signals( dic )

      def enviar(self, widget):
            entry1 = self.builder.get_object ("entry1")
            entry2 = self.builder.get_object ("entry2")
            entry3 = self.builder.get_object ("entry3")
            Goo   =   GOOSE(   entry1.get_text(),   entry2.get_text(),
entry3.get_text() )

      def ayuda(self, widget):
            Ayuda = GuiAyuda()

      def quit(self, widget):
            sys.exit(0)

class GuiAyuda:

      def __init__( self ):
            self.builder = gtk.Builder()
            self.builder.add_from_file("Interfaz_Ayuda.ui")

Gui = Gui()
gtk.main()
```

Explicación de uso de la herramienta *goose.py*

La funcionalidad del programa realizado es enviar mensajes rápidos cumpliendo el protocolo GOOSE definido en la norma IEC 61850.

Al ejecutar el código mediante la instrucción *python goose.py* en el terminal de Ubuntu el usuario se encontrará con una interfaz como la expuesta en la Figura A1 de este anexo. En esta se encuentran 3 casillas de texto y 3 botones. Las casillas permiten cambiar los parámetros MAC Destino, MAC Origen y cantidad de mensajes GOOSE y los botones dan la opción de enviar mensajes GOOSE, recibir una ayuda de manejo de la interfaz y salir del programa.

Figura A1. Interfaz gráfica del programa GOOSE

Fuente: Autor

Las tres casillas de texto se describen a continuación:

- MAC Destino: En este campo se define la dirección MAC destino de los mensajes, en formato (01:0C:CD:01:00:00) esta se define por la norma IEC 61850 como una dirección Multicast.

- MAC Origen: En este campo se define la dirección MAC de donde es emitido el mensaje. En formato (00:OC:29:CA:B5:B7).

- Cantidad de mensajes (Número Entero): En este campo se define la cantidad de mensajes GOOSE que el usuario desea enviar.

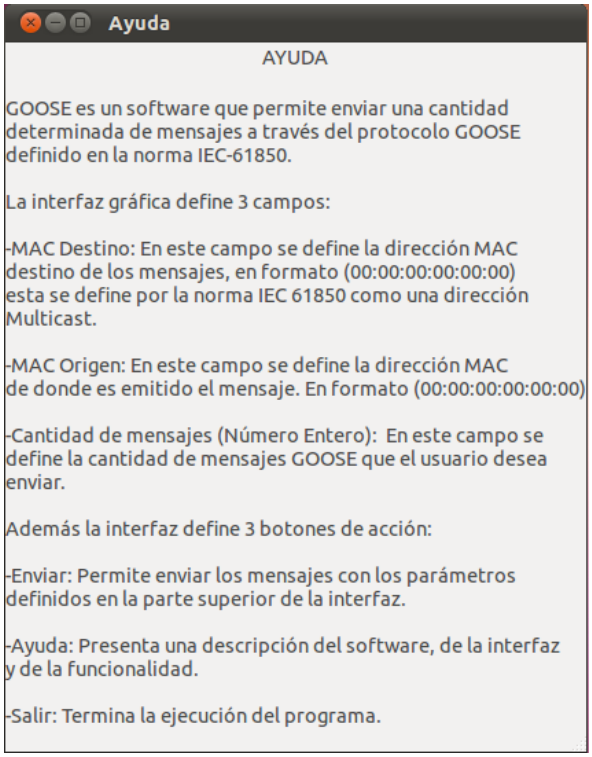
Los tres botones de acción se explican a continuación:

- Enviar: Permite enviar los mensajes con los parámetros definidos en la parte superior de la interfaz.

- Ayuda: Presenta una descripción del software, de la interfaz y de la funcionalidad.

- Salir: Termina la ejecución del programa.

Figura A2. Ayuda del programa GOOSE



Fuente: Autor

ANEXO B: CARACTERÍSTICAS FÍSICAS DE LOS EQUIPOS UTILIZADOS PARA REALIZAR LOS EXPERIMENTOS

El PC0 y el PC1 (AMD Athlon 64 X2 Dual Core Processor 5600+ RAM: 4GB Sistema Operativo: Debian GNU/Linux 6.0.5), PC2 (Intel Core i7 Q720 RAM: 2GB Sistema Operativo: Windows 7 64 Bits).

Tarjeta de Red PC0 y PC1 (Broadcom NetLink NetXtreme 57xx: Clock 66MHz Maximum, Gigabit Ethernet, Logical Link Control, Layer-2 Priority Encoding, Standar Ethernet frame size) y Tarjeta de Red PC2 (Gigabit Ethernet PCI-E de la familia Realtek RTL8168D/8111D: Clock 25MHz, Supports Layer-2 Priority Encoding, Supports VLAN tagging, Supports Full Duplex flow control). [52]

Switch-1 (TRENDnet TE100-S24WS: 24 puertos Fast-Ethernet Auto-MDIX y auto-Negociación a 10/100Mbps, modo de transferencia Full/half dúplex para cada puerto, Buffer de datos RAM de 768K bytes y tabla de dirección MAC con 4k entradas, VLAN 25 Estaciones basado en puerto y QoS IEEE 802.1p basado en puerto). [53]

Switch-2 (D-Link DES3326S: 24 10/100Mbps Ports for Workstation Connection, Wire-speed IP Routing, VLANs for Enhanced Security & Performance, Advanced QoS Support, Optional Ports, Supports 802.1Q VLAN, IGMP snooping, 802.1p Priority, Administrator definable port security).[54]

NOTA: La velocidad de transmisión utilizada para todos los experimentos realizados en los anexos C y D es de 100Mbps. (Fast Ethernet – 100Base-T).

---

[52] Ver www.broadcom.com/docs/support/ethernet_nic/Broadcom_NetLink-NetXtreme_DTM_15.pdf http://www.realtek.com.tw/search/default.aspx?keyword=RTL8168
[53] Ver http://www.trendnet.com/downloads/list_subcategory.asp?SUBTYPE_ID=351
[54] Ver ftp.dlink.se/Datasheets/des-3326s.pdf

ANEXO C: INFORMES DE LOS EXPERIMENTOS REALIZADOS PARA LA EVALUACIÓN DE MENSAJES RÁPIDOS EN UNA RED LOCAL.

INFORME
EXPERIMENTO 1

Fecha: Septiembre 17 / 2012

Hora: 5PM a 5:05PM

Tráfico: Ninguno

Distancia entre los equipos: Menor a 2 metros

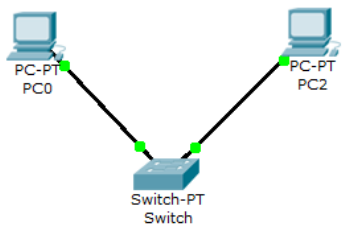
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

Se enviaron 100 mensajes GOOSE (Unicast) a través de la topología descrita en el informe, desde el PC0 con dirección MAC 00:22:2d:28:8d:ab hacia el PC2 con dirección MAC 00:26:b9:1d:4b:77. Este experimento se realizó para comprobar el cumplimiento del protocolo GOOSE en una red controlada y el buen funcionamiento del programa realizado.

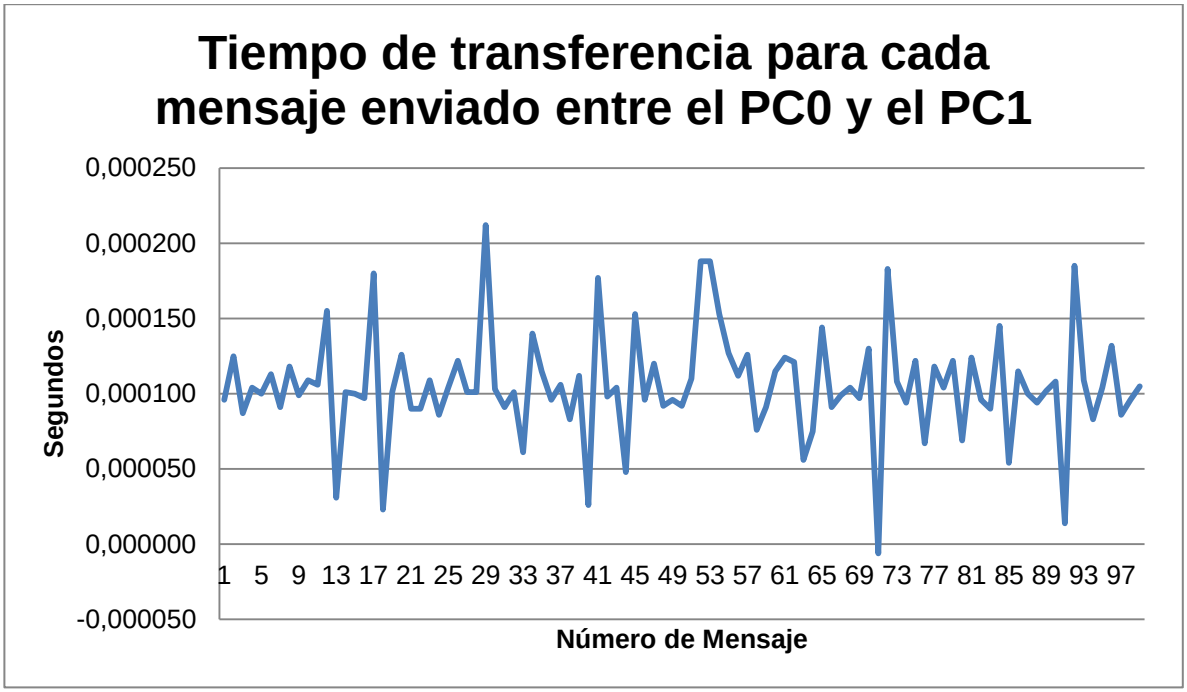
Topología de Red:

Figura C1. Topología de red del experimento 1



Fuente: Autor

Figura C2. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC1



Fuente: Autor

Resultados:

- Los mensajes enviados a través del protocolo se transmitieron de manera exitosa y rápida.
- El promedio de transferencia de los mensajes enviados fue de 0,000105 segundos.

INFORME

EXPERIMENTO 2

Fecha: Septiembre 19 / 2012

Hora: 8AM a 8:10AM

Tráfico: Ninguno

Distancia entre los equipos: Menor a 2 metros

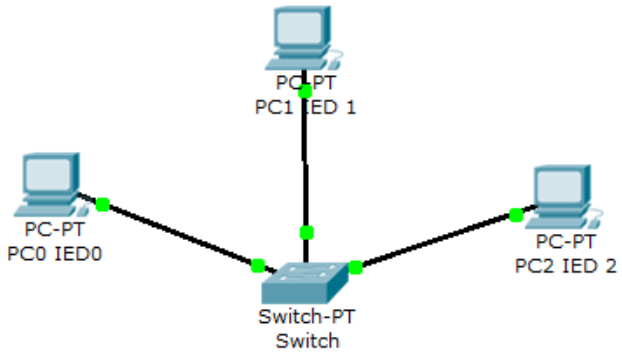
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. Este experimento se realizó para comprobar el funcionamiento correcto de la trasmisión multicast de mensajes GOOSE en una red controlada.

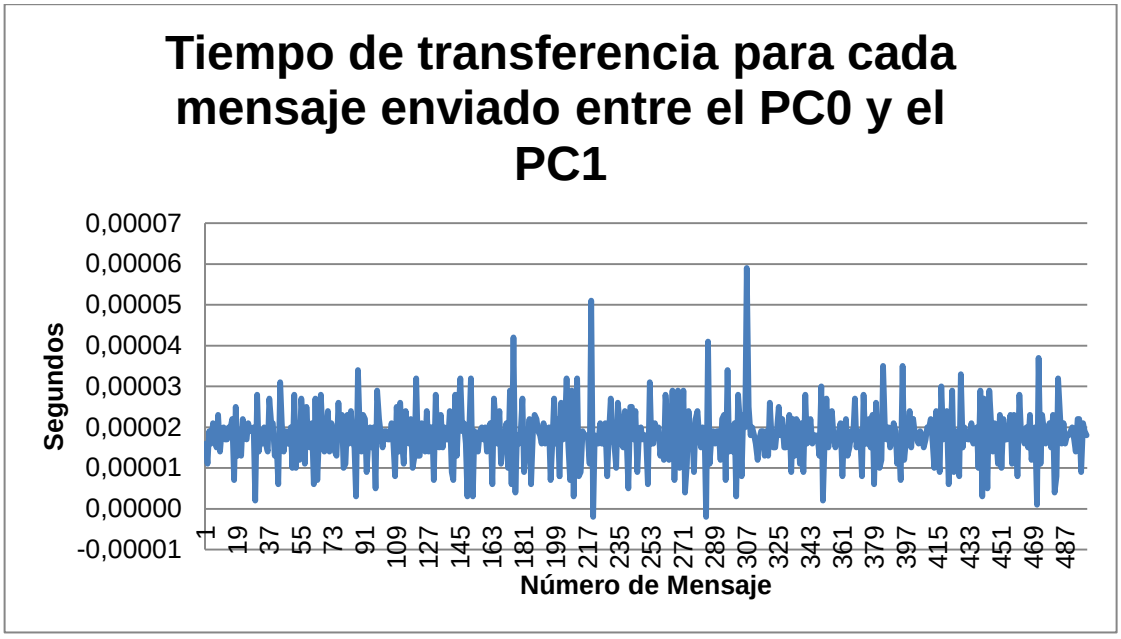
Topología de Red:

Figura C3. Topología de red del experimento 2



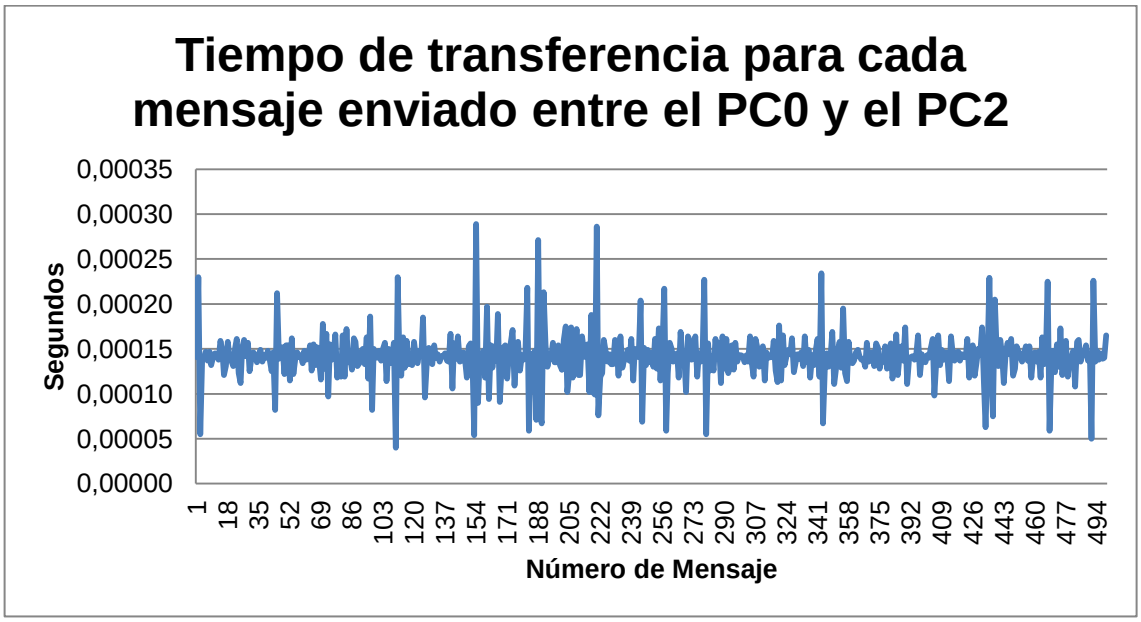
Fuente: Autor

Figura C4. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC1



Fuente: Autor

Figura C5. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC2



Fuente: Autor

Resultados:

- Los mensajes enviados a través del protocolo se transmitieron de manera exitosa y rápida.
- El tiempo promedio de transferencia de los mensajes desde el PC0 hacia el PC1 fue de 0,00002 segundos y desde el PC0 hacia el PC2 fue de 0,00014 segundos.

INFORME
EXPERIMENTO 2-1

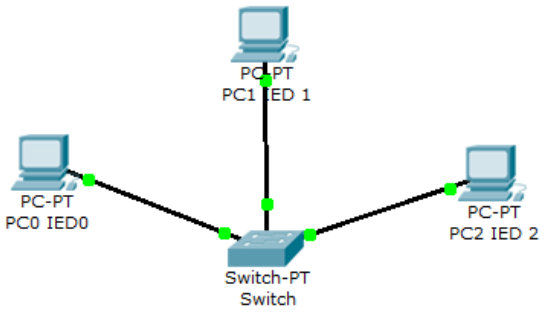
Fecha: Septiembre 19 / 2012
Hora: 9AM a 9:15AM
Tráfico: Ninguno
Distancia entre los equipos: Menor a 2 metros
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. Y a su vez se enviaron 5 segundos después 500 mensajes GOOSE (Multicast) más desde el PC1 (IED1) con dirección MAC 00:22:19:1d:06:43 hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. Este experimento se realizó para comprobar el funcionamiento correcto de la trasmisión multicast desde varios productores PC0 y PC1.

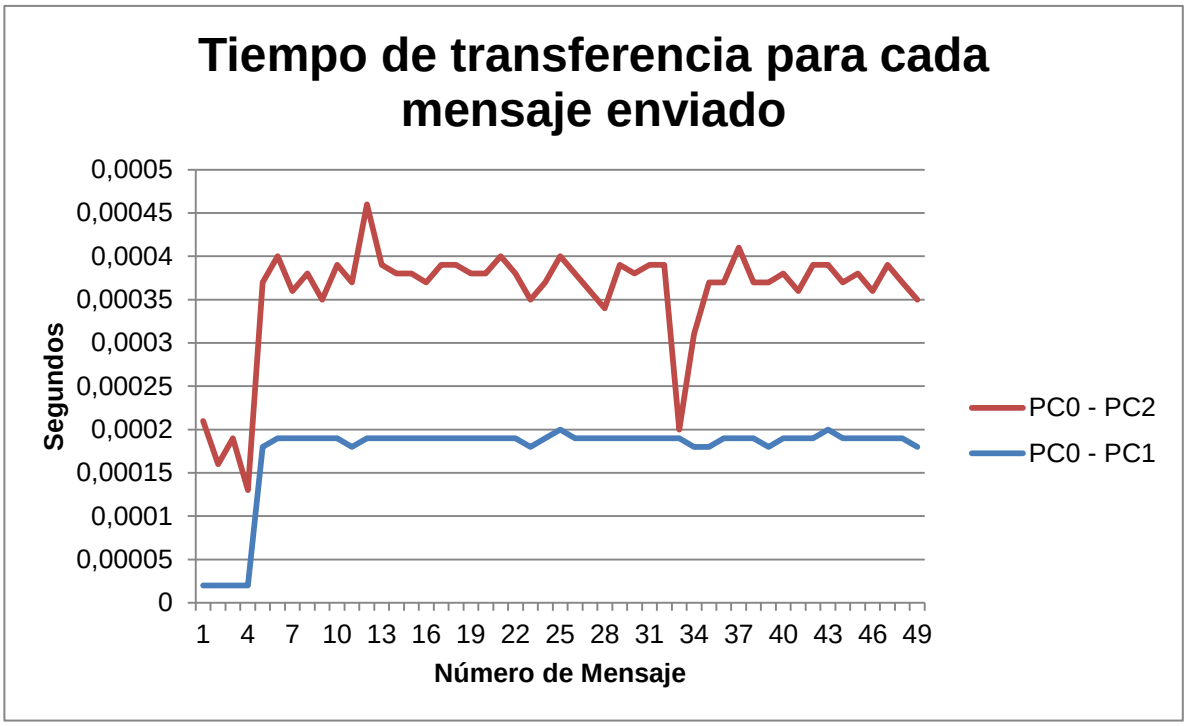
Topología de Red:

Figura C6. Topología de red del experimento 2-1



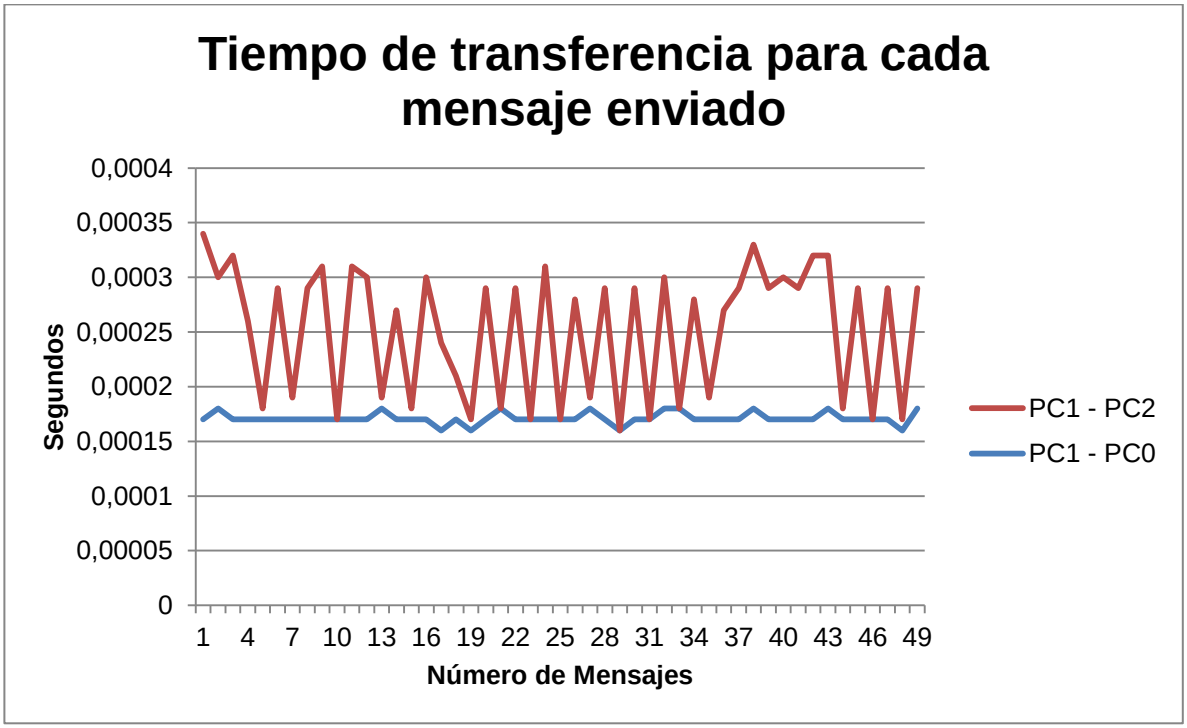
Fuente: Autor

Figura C7. Tiempo de transferencia para cada mensaje enviado entre el PC0 - PC1 y el PC0 – PC2



Fuente: Autor

Figura C8. Tiempo de transferencia para cada mensaje enviado entre el PC1 – PC0 y el PC1 – PC2



Fuente: Autor

Resultados:

- Los mensajes se trasmitieron con éxito, de manera correcta y rápida cumpliendo el protocolo.
- El tiempo promedio para las transmisiones fue: Desde PC0 hacia PC1: 0,00019 segundos, desde PC0 hacia PC2: 0,00018 segundos, desde PC1 hacia PC0: 0,00017 segundos, desde PC1 hacia PC2: 0,0009 segundos.

Observaciones:

- Para llevar a cabo las gráficas anteriormente mostradas se tomo una muestra de los 50 primeros mensajes, ya que al existir tantos datos (1000

en el caso de este experimento) no se podía apreciar con claridad el comportamiento de la gráfica.

- La inclusión de otro productor de mensajes GOOSE al experimento, demoró unos segundos más, la trasmisión de los mensajes de los PC0 y PC1, en esta ocasión el PC2 tuvo un promedio de tiempo menor al de los dos primeros debido a que el PC2 no es productor de mensajes GOOSE.

INFORME

EXPERIMENTO 3

Fecha: Septiembre 20 / 2012

Hora: 11AM a 11:10AM

Tráfico: Ninguno

Distancia entre los equipos: Menor a 2 metros

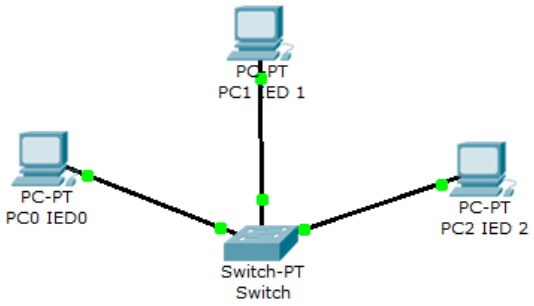
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. En esta ocasión la configuración del switch fue cambiada creando una VLAN en los puertos en donde fueron conectados los diferentes equipos. La VLAN tiene el nombre *vlan2* y los puertos pertenecientes a esta fueron: 5, 6, 7, 8, 17, 18, 19, 20. Este experimento se realizó para comprobar el funcionamiento correcto de la trasmisión multicast en una VLAN.

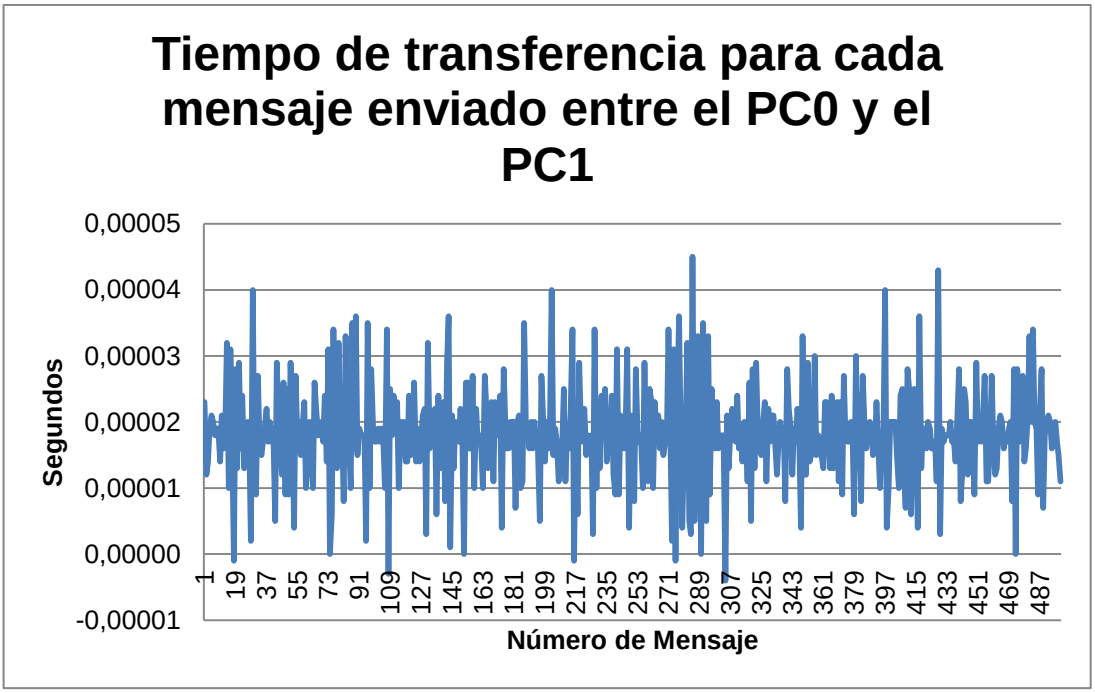
Topología de Red:

Figura C9. Topología de red del experimento 3



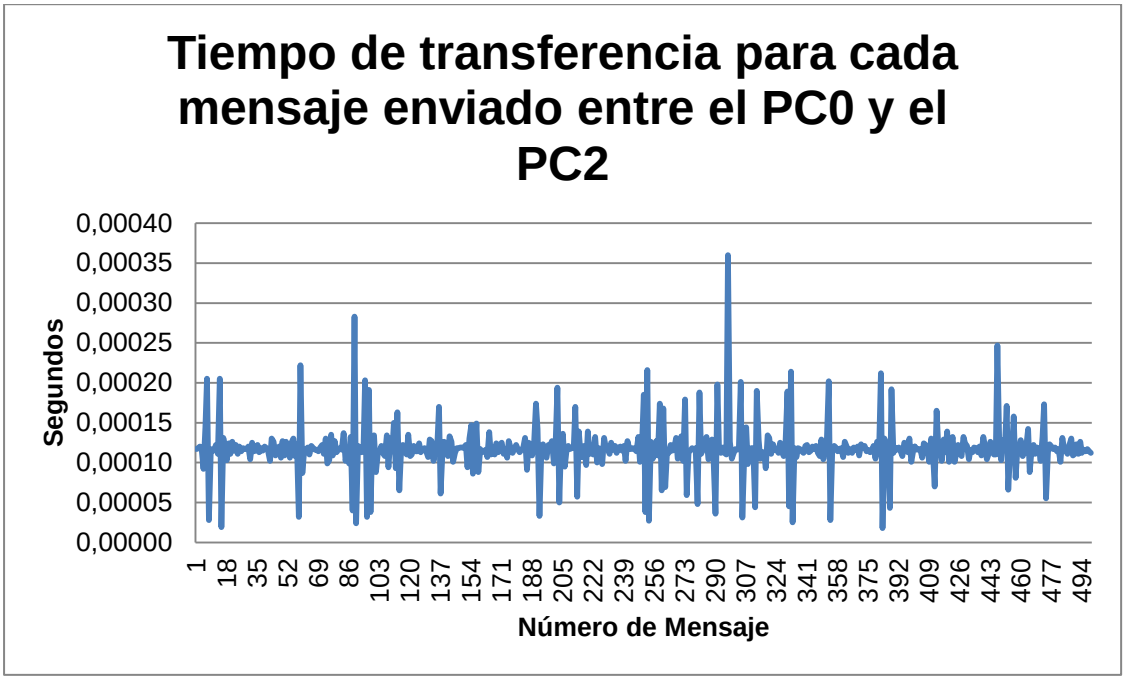
Fuente: Autor

Figura C10. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC1



Fuente: Autor

Figura C11. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC2



Fuente: Autor

Resultados:

- Los mensajes enviados a través del protocolo se transmitieron de manera correcta y rápida.
- El tiempo promedio de transferencia de los mensajes desde el PC0 hacia el PC1 fue de 0,00002 segundos y desde el PC0 hacia el PC2 fue de 0,00012 segundos.

INFORME

EXPERIMENTO 3-1

Fecha: Septiembre 20 / 2012

Hora: 12PM a 12:15PM

Tráfico: Ninguno

Distancia entre los equipos: Menor a 2 metros

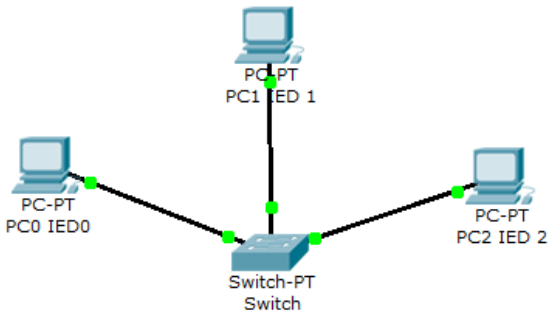
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. Y a su vez se enviaron 5 segundos después 500 mensajes GOOSE (Multicast) más desde el PC1 (IED1) con dirección MAC 00:22:19:1d:06:43 hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. En esta ocasión la configuración del switch fue cambiada creando una VLAN en los puertos en donde fueron conectados los diferentes equipos. La VLAN tiene el nombre *vlan2* y los puertos pertenecientes a esta fueron: 5, 6, 7, 8, 17, 18, 19, 20. Este experimento se realizó para comprobar el funcionamiento correcto de la trasmisión multicast desde dos productores PC0 y PC1 en una VLAN.

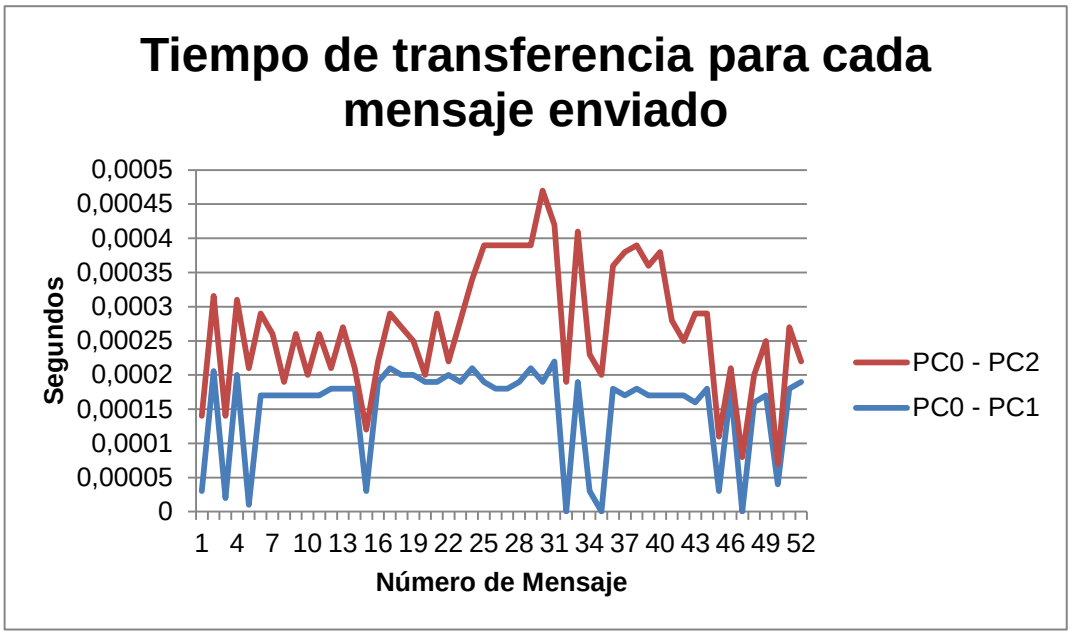
Topología de Red:

Figura C12. Topología de red del experimento 3-1



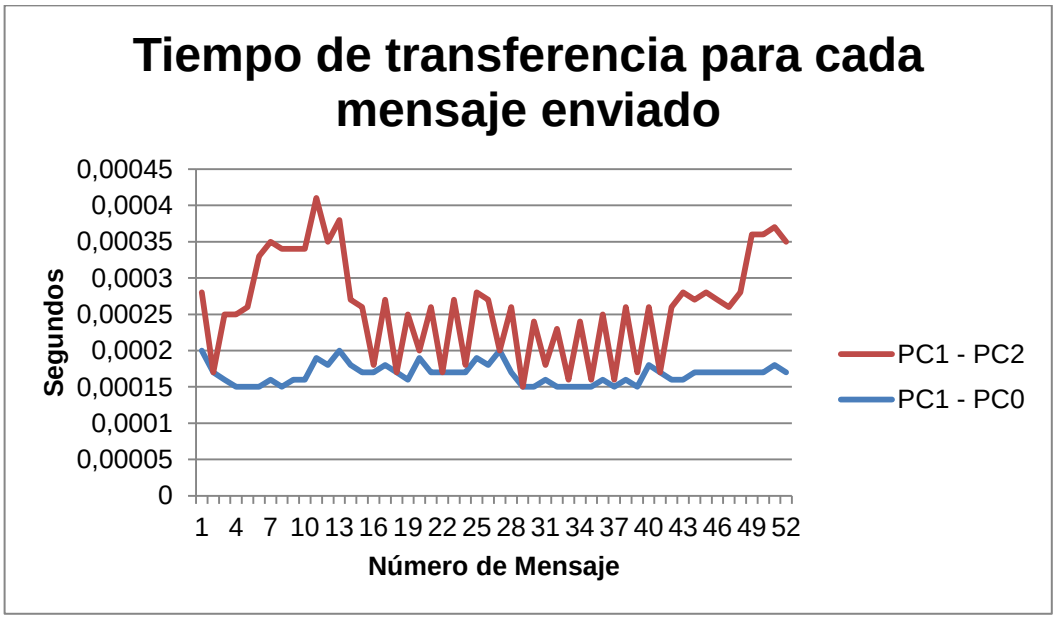
Fuente: Autor

Figura C13. Tiempo de transferencia para cada mensaje enviado entre el PC0 - PC1 y el PC0 – PC2



Fuente: Autor

Figura C14. Tiempo de transferencia para cada mensaje enviado entre el PC1 – PC0 y el PC1 – PC2



Fuente: Autor

Resultados:

- Los mensajes se trasmitieron con éxito, de manera correcta y rápida cumpliendo el protocolo.
- El tiempo promedio para las transmisiones fue: Desde PC0 hacia PC1: 0,00019 segundos, desde PC0 hacia PC2: 0,00015 segundos, desde PC1 hacia PC0: 0,00018 segundos, desde PC1 hacia PC2: 0,0008 segundos.

Observaciones:

- Para llevar a cabo las gráficas anteriormente mostradas se tomó una muestra de los 50 primeros mensajes, ya que al existir tantos datos (1000 en el caso de este experimento) no se podía apreciar con claridad el comportamiento de la gráfica.
- La inclusión de otro productor de mensajes GOOSE al experimento demoró unos segundos más, la trasmisión de los mensajes de los PC0 y PC1. En esta ocasión el PC2 tuvo un promedio de tiempo menor al de los dos primeros debido a que el PC2 no es productor de mensajes GOOSE.

INFORME

EXPERIMENTO 4

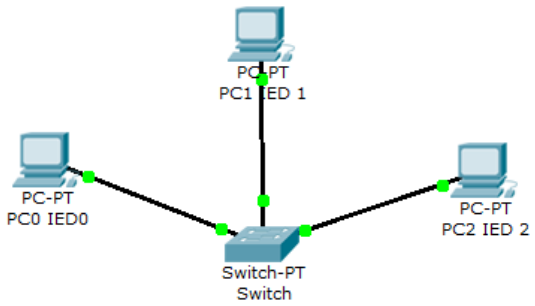
Fecha: Octubre 01 / 2012
Hora: 10AM a 10:10AM
Tráfico: Bajo
Distancia entre los equipos: Menor a 2 metros
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. En esta ocasión la configuración del switch fue cambiada creando una VLAN en los puertos en donde fueron conectados los diferentes equipos. La VLAN tiene el nombre *vlan2* y los puertos pertenecientes a esta fueron: 5, 6, 7, 8, 17, 18, 19, 20. Este experimento se realizó para comprobar el comportamiento de la trasmisión multicast en una VLAN con tráfico de red bajo. El tráfico de red es generado desde el PC2.

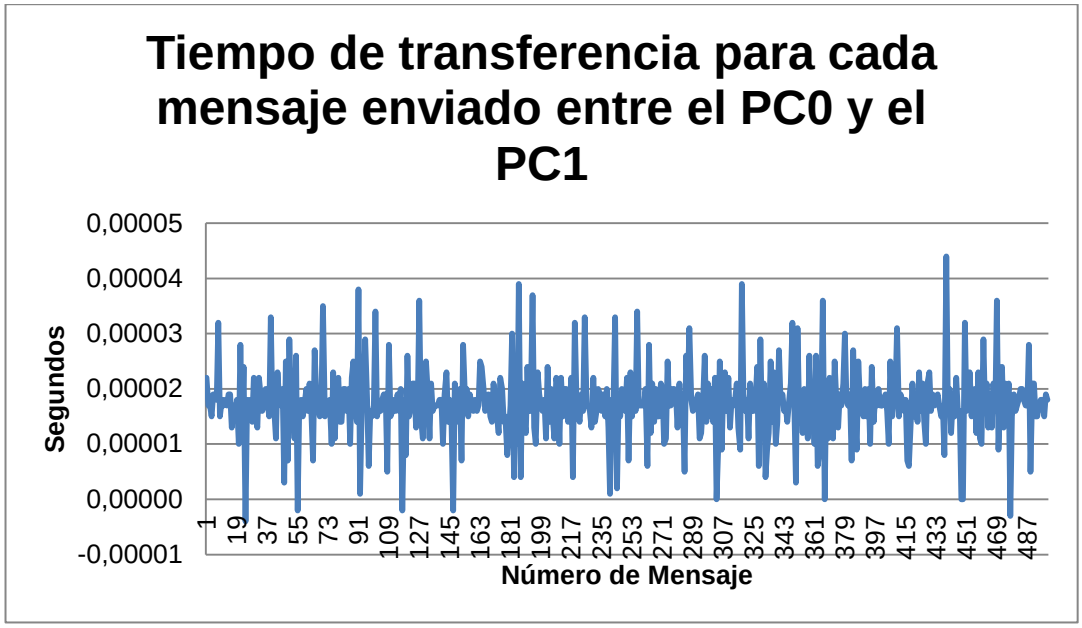
Topología de Red:

Figura C15. Topología de red del experimento 4



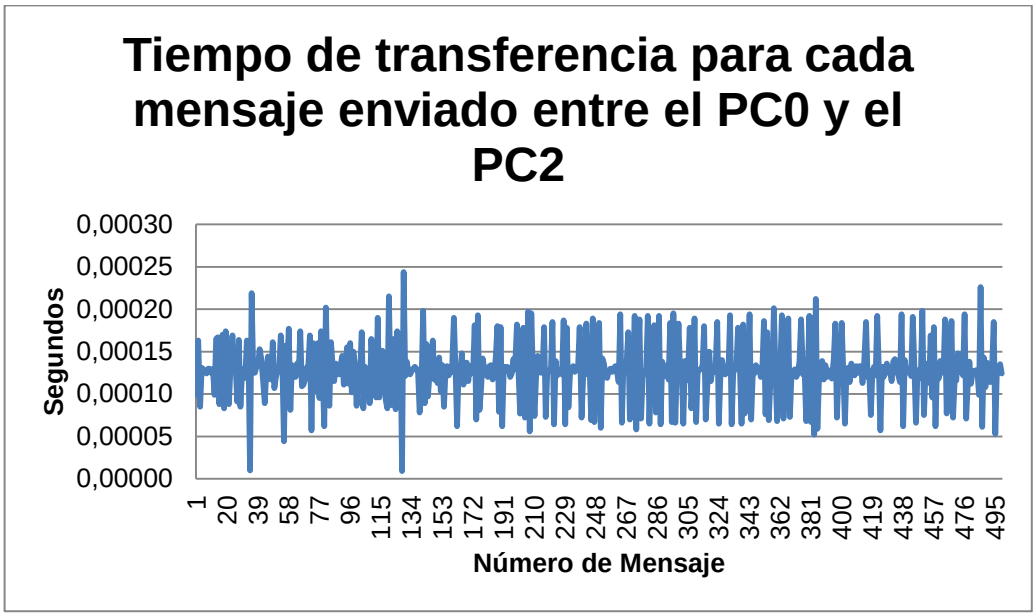
Fuente: Autor

Figura C16. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC1



Fuente: Autor

Figura C17. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC2



Fuente: Autor

Resultados:

- Los mensajes enviados a través del protocolo se transmitieron de manera correcta y rápida.
- El tiempo promedio de transferencia de los mensajes desde el PC0 hacia el PC1 fue de 0,00002 segundos y desde el PC0 hacia el PC2 fue de 0,00013 segundos.

INFORME

EXPERIMENTO 4-1

Fecha: Octubre 01 / 2012

Hora: 10:50 AM a 11:05AM

Tráfico: Bajo

Distancia entre los equipos: Menor a 2 metros

Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

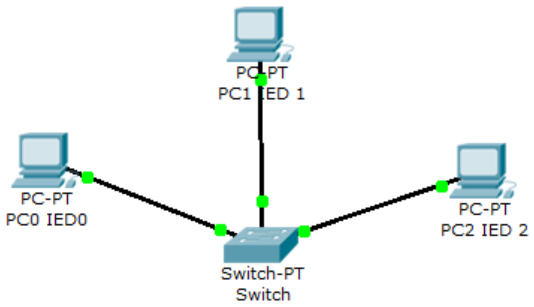
Descripción del Experimento

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. Y a su vez se enviaron 5 segundos después 500 mensajes GOOSE (Multicast) más desde el PC1 (IED1) con dirección MAC 00:22:19:1d:06:43 hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. En esta ocasión la configuración del switch fue cambiada creando una VLAN en los puertos en donde fueron conectados los diferentes equipos. La VLAN tiene el nombre *vlan2* y los puertos pertenecientes a esta fueron: 5, 6, 7, 8, 17, 18, 19, 20. Este experimento se realizó para comprobar el comportamiento de la trasmisión

multicast desde varios productores PC0 y PC1 en una VLAN con tráfico bajo. El tráfico de red es generado desde el PC2.

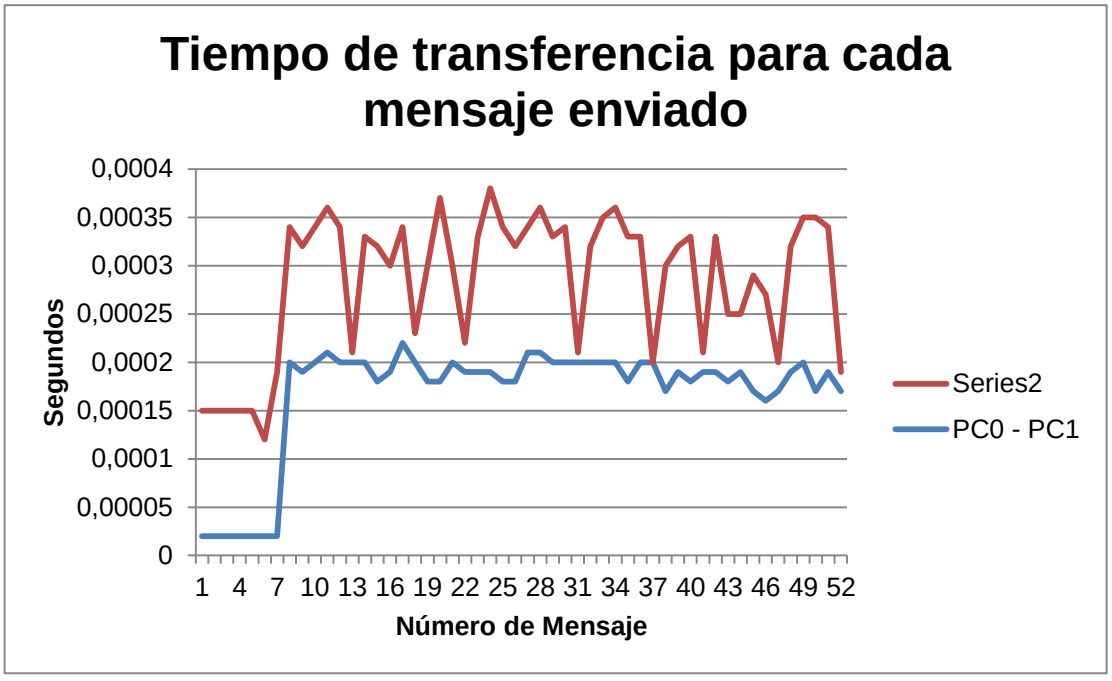
Topología de Red:

Figura C18. Topología de red del experimento 4-1



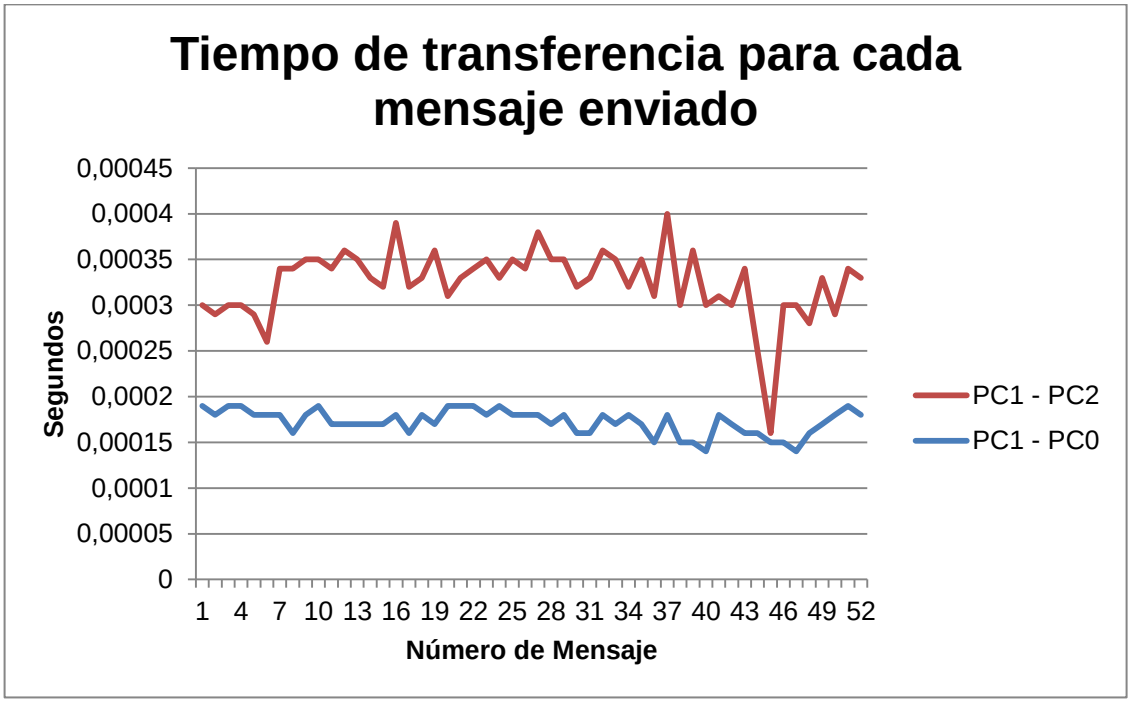
Fuente: Autor

Figura C19. Tiempo de transferencia para cada mensaje enviado entre el PC0 - PC1 y el PC0 – PC2



Fuente: Autor

Figura C20. Tiempo de transferencia para cada mensaje enviado entre el PC1 – PC0 y el PC1 – PC2



Fuente: Autor

Resultados:

- Los mensajes se trasmitieron con éxito, de manera correcta y rápida cumpliendo el protocolo.
- El tiempo promedio para las transmisiones fue: Desde PC0 hacia PC1: 0,00019 segundos, desde PC0 hacia PC2: 0,00012 segundos, desde PC1 hacia PC0: 0,00018 segundos, desde PC1 hacia PC2: 0,0014 segundos.

Observaciones:

- Para llevar a cabo las gráficas anteriormente mostradas se tomo una muestra de los 50 primeros mensajes, ya que al existir tantos datos (1000

en el caso de este experimento) no se podía apreciar con claridad el comportamiento de la gráfica.

- La inclusión de otro productor de mensajes GOOSE al experimento demoró unos segundos más, la trasmisión de los mensajes de los PC0 y PC1. En esta ocasión el PC2 tuvo un promedio de tiempo menor al de los dos primeros debido a que el PC2 no es productor de mensajes GOOSE.

INFORME

EXPERIMENTO 5

Fecha: Octubre 01 / 2012

Hora: 12PM a 12:10PM

Tráfico: Medio

Distancia entre los equipos: Menor a 2 metros

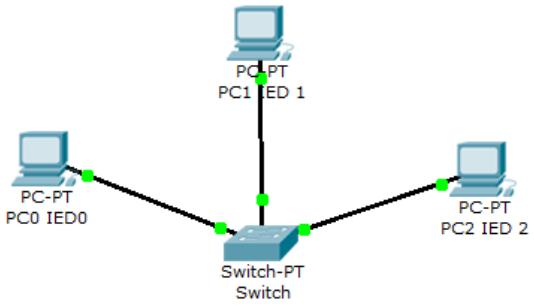
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. En esta ocasión la configuración del switch fue cambiada creando una VLAN en los puertos en donde fueron conectados los diferentes equipos. La VLAN tiene el nombre *vlan2* y los puertos pertenecientes a esta fueron: 5, 6, 7, 8, 17, 18, 19, 20. Este experimento se realizó para comprobar el comportamiento de la trasmisión multicast en una VLAN con tráfico de red medio. El tráfico de red es generado desde el PC2.

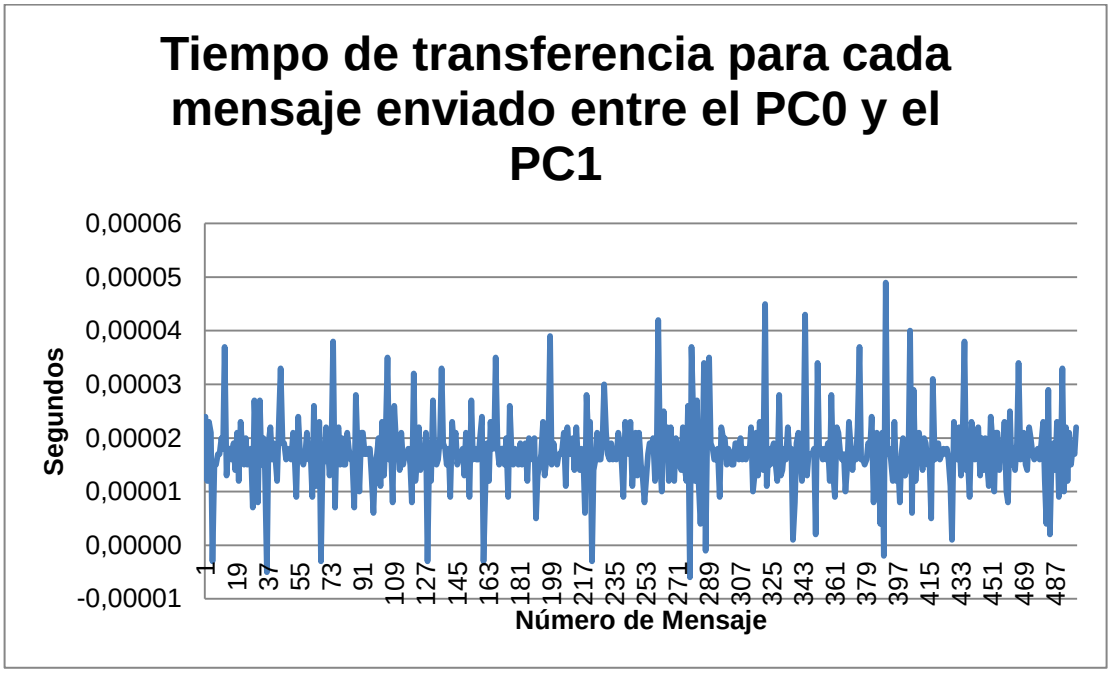
Topología de Red:

Figura C21. Topología de red del experimento 5



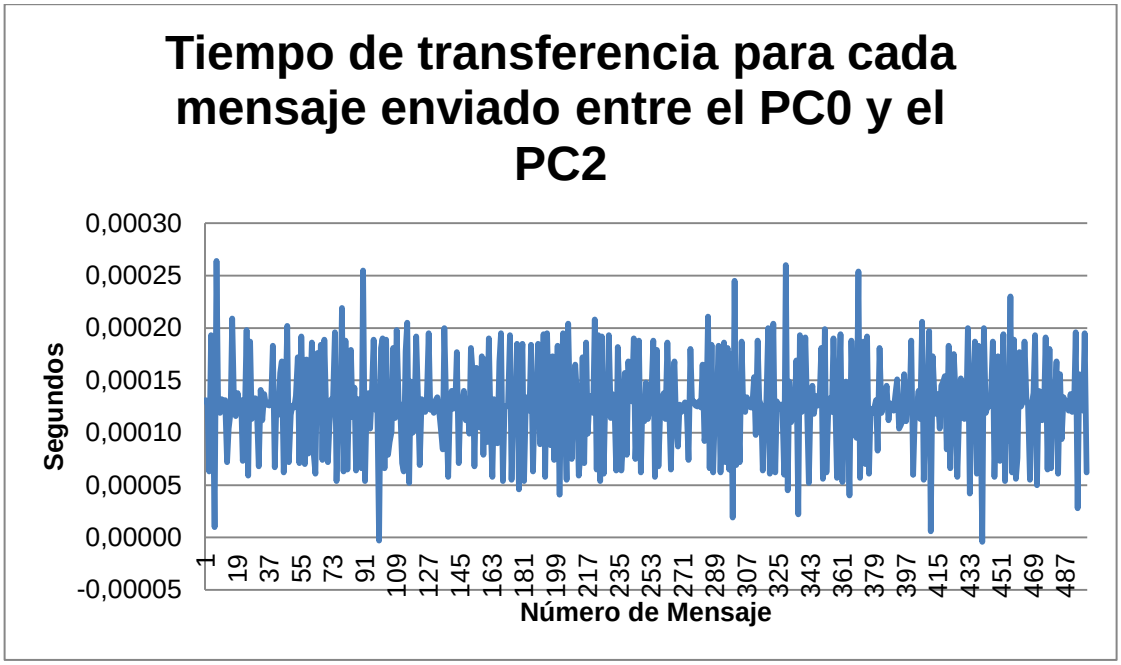
Fuente: Autor

Figura C22. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC1



Fuente: Autor

Figura C23. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC2



Fuente: Autor

Resultados:

- Los mensajes enviados a través del protocolo se transmitieron de manera correcta y rápida.
- El tiempo promedio de transferencia de los mensajes desde el PC0 hacia el PC1 fue de 0,00002 segundos y desde el PC0 hacia el PC2 fue de 0,00013 segundos.

INFORME

EXPERIMENTO 5-1

Fecha: Octubre 01 / 2012

Hora: 1PM a 1:15PM

Tráfico: Medio

Distancia entre los equipos: Menor a 2 metros

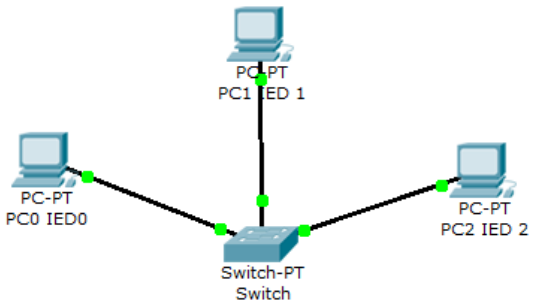
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. Y a su vez se enviaron 5 segundos después 500 mensajes GOOSE (Multicast) más desde el PC1 (IED1) con dirección MAC 00:22:19:1d:06:43 hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. En esta ocasión la configuración del switch fue cambiada creando una VLAN en los puertos en donde fueron conectados los diferentes equipos. La VLAN tiene el nombre *vlan2* y los puertos pertenecientes a esta fueron: 5, 6, 7, 8, 17, 18, 19, 20. Este experimento se realizó para comprobar el comportamiento de la trasmisión multicast desde varios productores PC0 y PC1 en una VLAN con tráfico medio. El tráfico de red es generado desde el PC2.

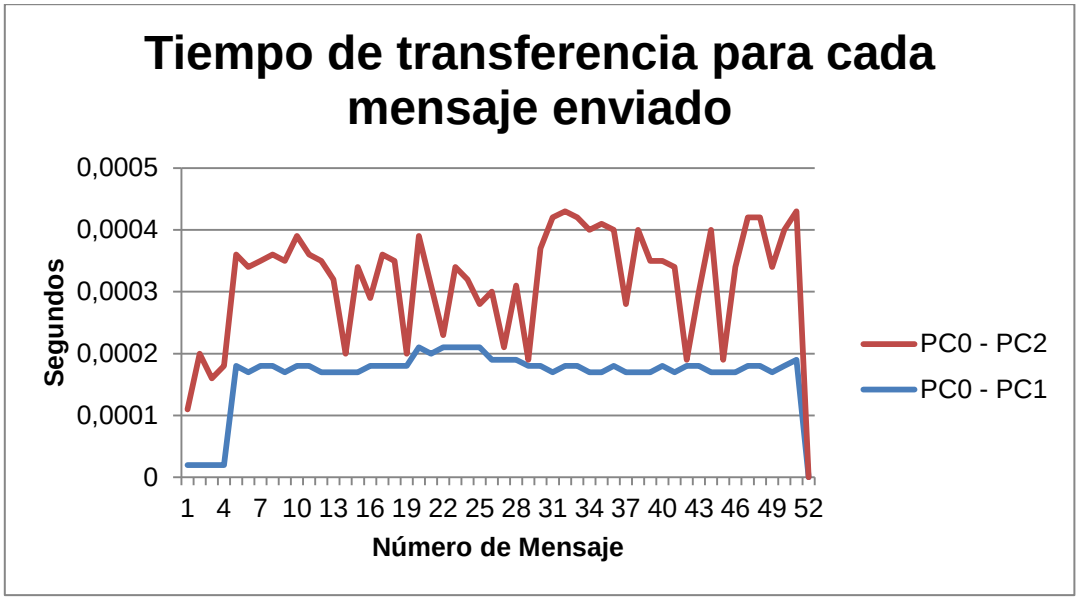
Topología de Red:

Figura C24. Topología de red del experimento 5-1



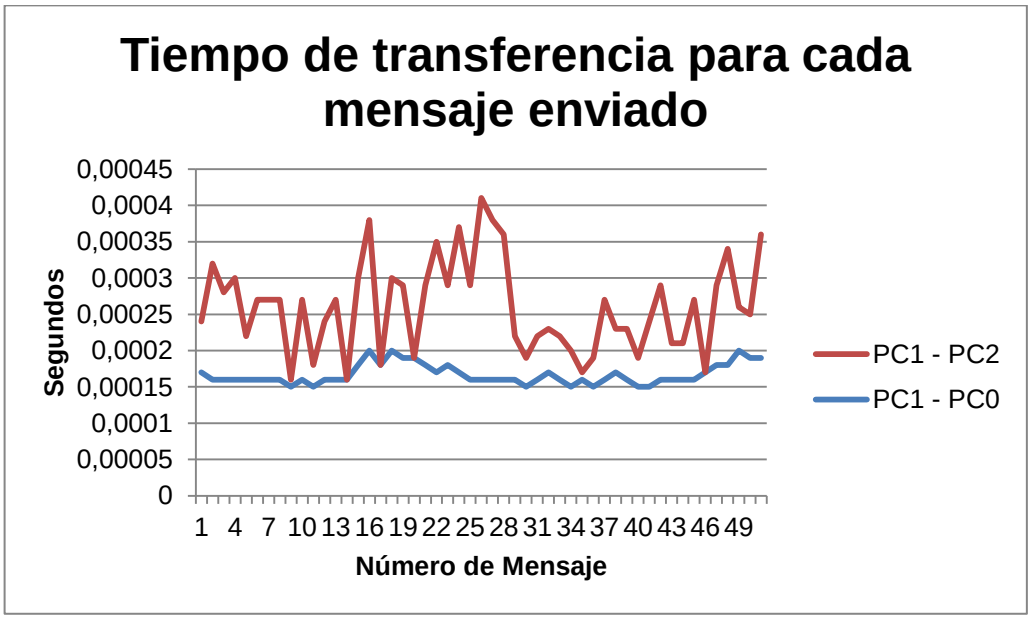
Fuente: Autor

Figura C25. Tiempo de transferencia para cada mensaje enviado entre el PC0 - PC1 y el PC0 – PC2



Fuente: Autor

Figura C26. Tiempo de transferencia para cada mensaje enviado entre el PC1 – PC0 y el PC1 – PC2



Fuente: Autor

Resultados:

- Los mensajes se trasmitieron con éxito, de manera correcta y rápida cumpliendo el protocolo.
- El tiempo promedio para las transmisiones fue: Desde PC0 hacia PC1: 0,00018 segundos, desde PC0 hacia PC2: 0,00016 segundos, desde PC1 hacia PC0: 0,00017 segundos, desde PC1 hacia PC2: 0,0008 segundos.

Observaciones:

- Para llevar a cabo las gráficas anteriormente mostradas se tomo una muestra de los 50 primeros mensajes, ya que al existir tantos datos (1000 en el caso de este experimento) no se podía apreciar con claridad el comportamiento de la gráfica.
- La inclusión de otro productor de mensajes GOOSE al experimento demoró unos segundos más, la trasmisión de los mensajes de los PC0 y PC1. En esta ocasión el PC2 tuvo un promedio de tiempo menor al de los dos primeros debido a que el PC2 no es productor de mensajes GOOSE.

  .

INFORME

EXPERIMENTO 6

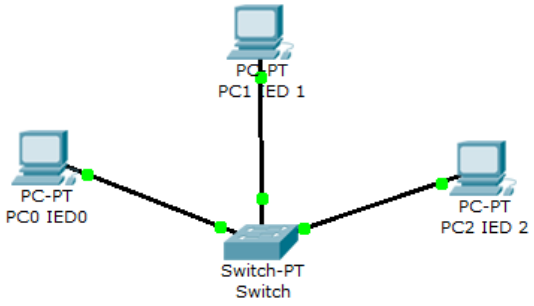
Fecha: Octubre 01 / 2012
Hora: 1:30PM a 1:40PM
Tráfico: Alto
Distancia entre los equipos: Menor a 2 metros
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. En esta ocasión la configuración del switch fue cambiada creando una VLAN en los puertos en donde fueron conectados los diferentes equipos. La VLAN tiene el nombre *vlan2* y los puertos pertenecientes a esta fueron: 5, 6, 7, 8, 17, 18, 19, 20. Este experimento se realizó para comprobar el comportamiento de la trasmisión multicast en una VLAN con tráfico de red alto. El tráfico de red es generado desde el PC2.

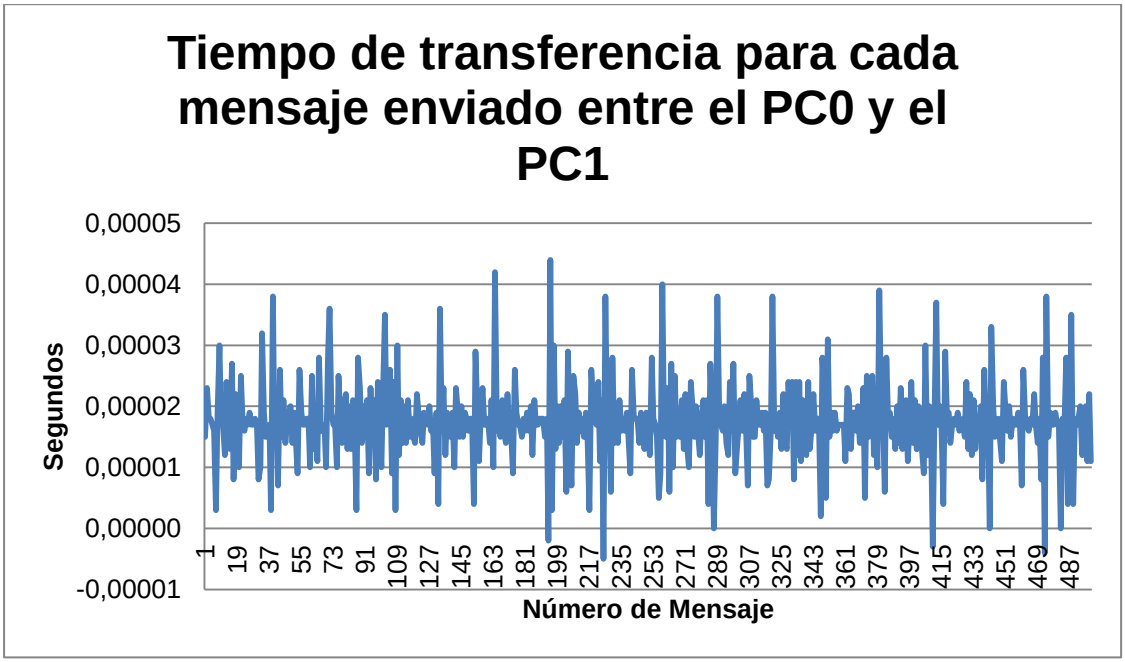
Topología de Red:

Figura C27. Topología de red del experimento 6



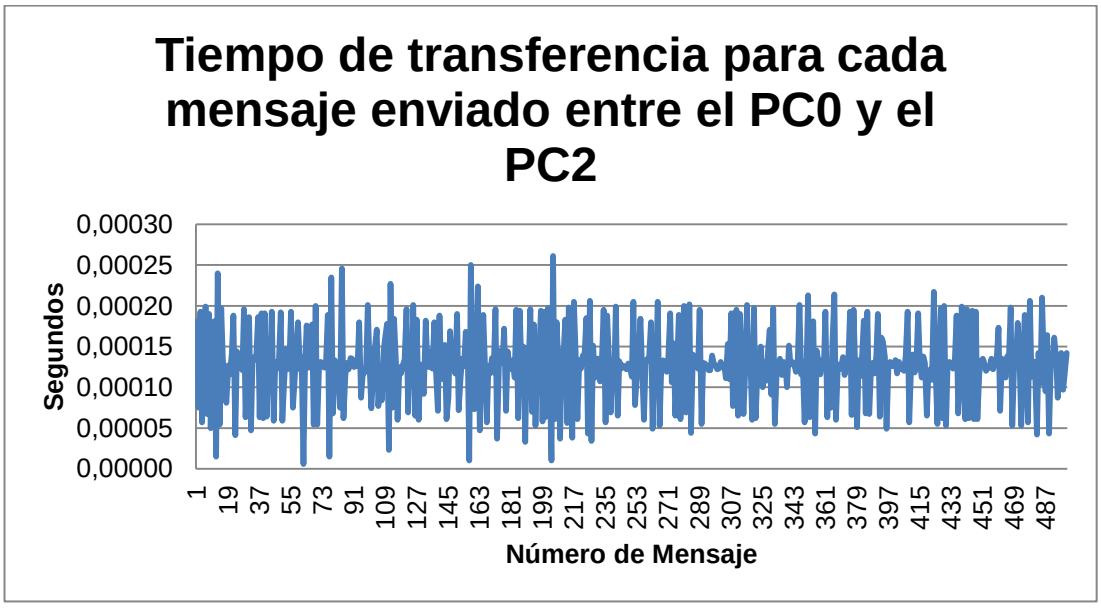
Fuente: Autor

Figura C28. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC1



Fuente: Autor

Figura C29. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC2



Fuente: Autor

Resultados:

- Los mensajes enviados a través del protocolo se transmitieron de manera correcta y rápida.
- El tiempo promedio de transferencia de los mensajes desde el PC0 hacia el PC1 fue de 0,00002 segundos y desde el PC0 hacia el PC2 fue de 0,00013 segundos.

INFORME

EXPERIMENTO 6-1

Fecha: Octubre 01 / 2012

Hora: 2PM a 2:15PM

Tráfico: Alto

Distancia entre los equipos: Menor a 2 metros

Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

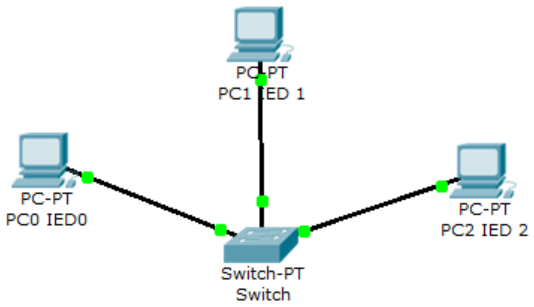
Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. Y a su vez se enviaron 5 segundos después 500 mensajes GOOSE (Multicast) más desde el PC1 (IED1) con dirección MAC 00:22:19:1d:06:43 hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. En esta ocasión la configuración del switch fue cambiada creando una VLAN en los puertos en donde fueron conectados los diferentes equipos. La VLAN tiene el nombre *vlan2* y los puertos pertenecientes a esta fueron: 5, 6, 7, 8, 17, 18, 19, 20. Este experimento se realizó para comprobar el comportamiento de la trasmisión

multicast desde varios productores PC0 y PC1 en una VLAN con tráfico alto. El tráfico de red es generado desde el PC2.

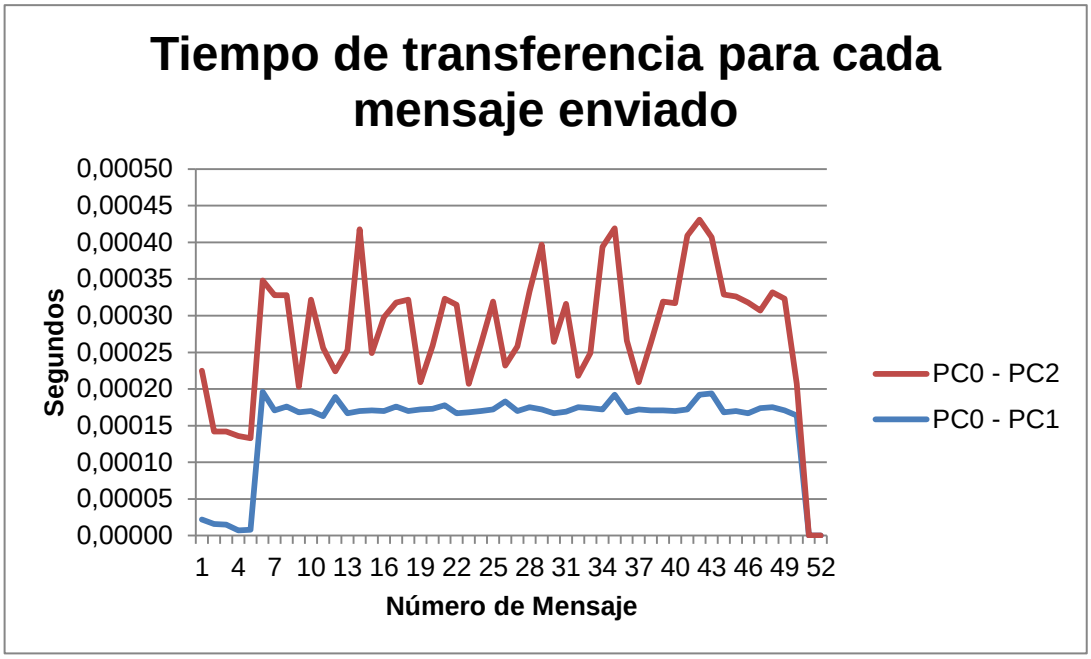
Topología de Red:

Figura C30. Topología de red del experimento 6-1



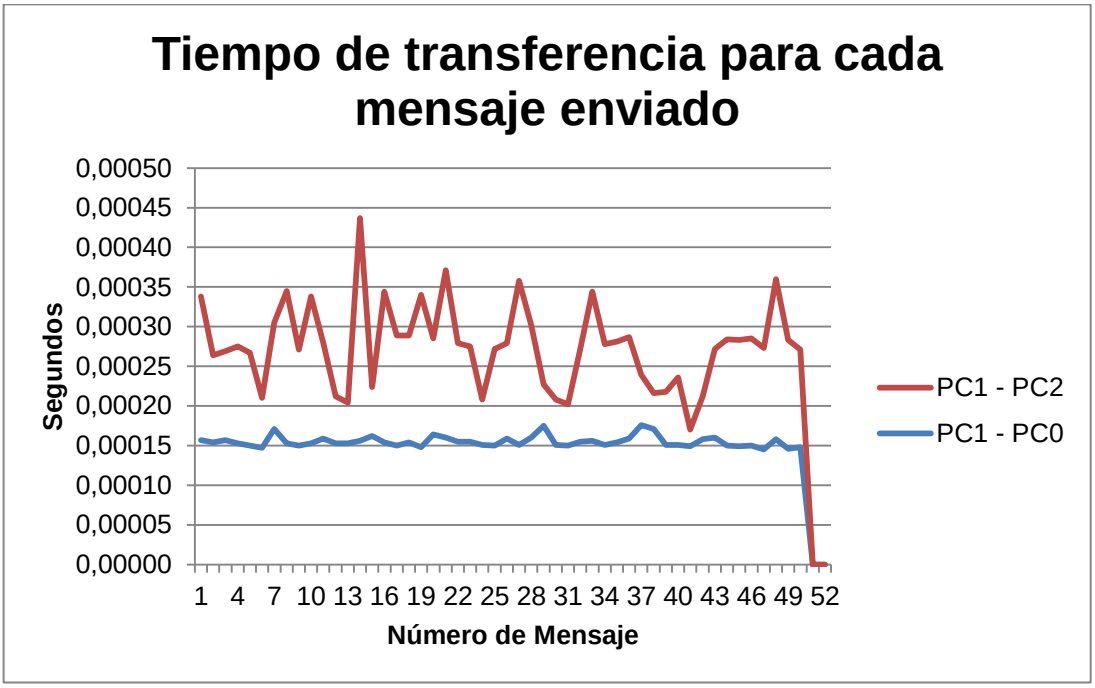
Fuente: Autor

Figura C31. Tiempo de transferencia para cada mensaje enviado entre el PC0 - PC1 y el PC0 – PC2



Fuente: Autor

Figura C32. Tiempo de transferencia para cada mensaje enviado entre el PC1 – PC0 y el PC1 – PC2



Fuente: Autor

Resultados:

- Los mensajes se trasmitieron con éxito, de manera correcta y rápida cumpliendo el protocolo.
- El tiempo promedio para las transmisiones fue: Desde PC0 hacia PC1: 0,00017 segundos, desde PC0 hacia PC2: 0,00013 segundos, desde PC1 hacia PC0: 0,00016 segundos, desde PC1 hacia PC2: 0,0012 segundos.

Observaciones:

- Para llevar a cabo las gráficas anteriormente mostradas se tomo una muestra de los 50 primeros mensajes, ya que al existir tantos datos (1000

en el caso de este experimento) no se podía apreciar con claridad el comportamiento de la gráfica.

- La inclusión de otro productor de mensajes GOOSE al experimento demoró unos segundos más, la trasmisión de los mensajes de los PC0 y PC1. En esta ocasión el PC2 tuvo un promedio de tiempo menor al de los dos primeros debido a que el PC2 no es productor de mensajes GOOSE.

INFORME

EXPERIMENTO 7

Fecha: Octubre 01 / 2012

Hora: 2:30PM a 2:40PM

Tráfico: 250000 Mensajes con velocidad de 500 Paquetes / Segundo

Distancia entre los equipos: Menor a 2 metros

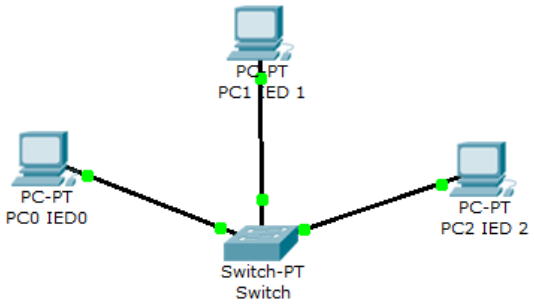
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. En esta ocasión la configuración del switch fue cambiada creando una VLAN en los puertos en donde fueron conectados los diferentes equipos. La VLAN tiene el nombre *vlan2* y los puertos pertenecientes a esta fueron: 5, 6, 7, 8, 17, 18, 19, 20. Este experimento se realizó para comprobar el comportamiento de la trasmisión multicast en una VLAN con tráfico de red de 500 Paquetes/segundo. El tráfico de red es generado desde el PC2.

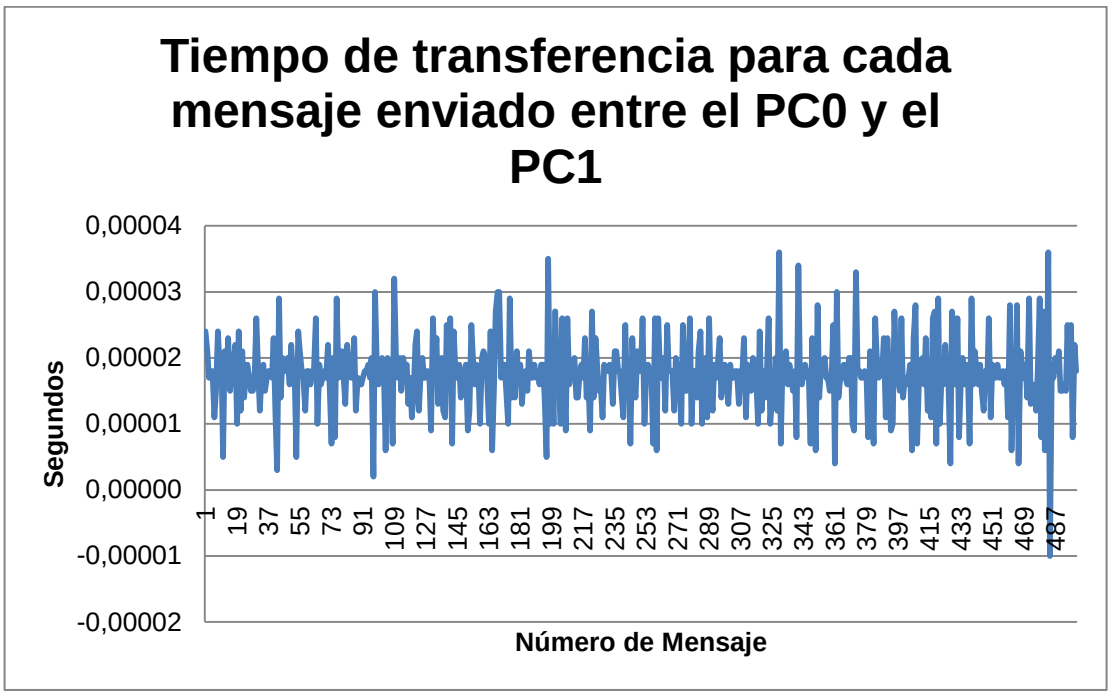
Topología de Red:

Figura C33. Topología de red del experimento 7



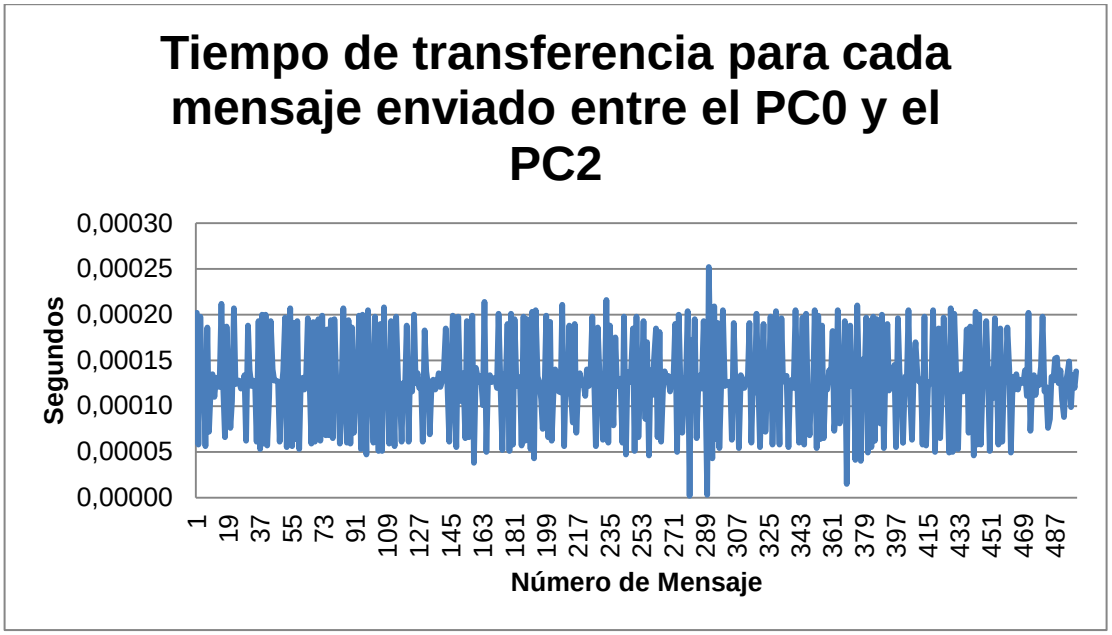
Fuente: Autor

Figura C34. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC1



Fuente: Autor

Figura C35. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC2



Fuente: Autor

Resultados:

- Los mensajes enviados a través del protocolo se transmitieron de manera correcta y rápida.
- El tiempo promedio de transferencia de los mensajes desde el PC0 hacia el PC1 fue de 0,00002 segundos y desde el PC0 hacia el PC2 fue de 0,00013 segundos.

INFORME

EXPERIMENTO 8

Fecha: Octubre 25 / 2012

Hora: 5:40PM a 5:50PM

Tráfico: Ninguno

Distancia entre los equipos: Menor a 2 metros

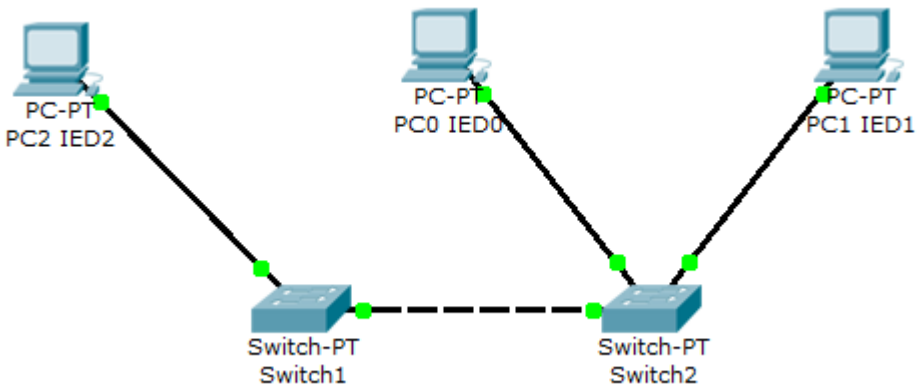
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. Este experimento se realizó para comprobar la transmisión de mensajes GOOSE en una topología con dos switchs y un solo productor de mensajes GOOSE.

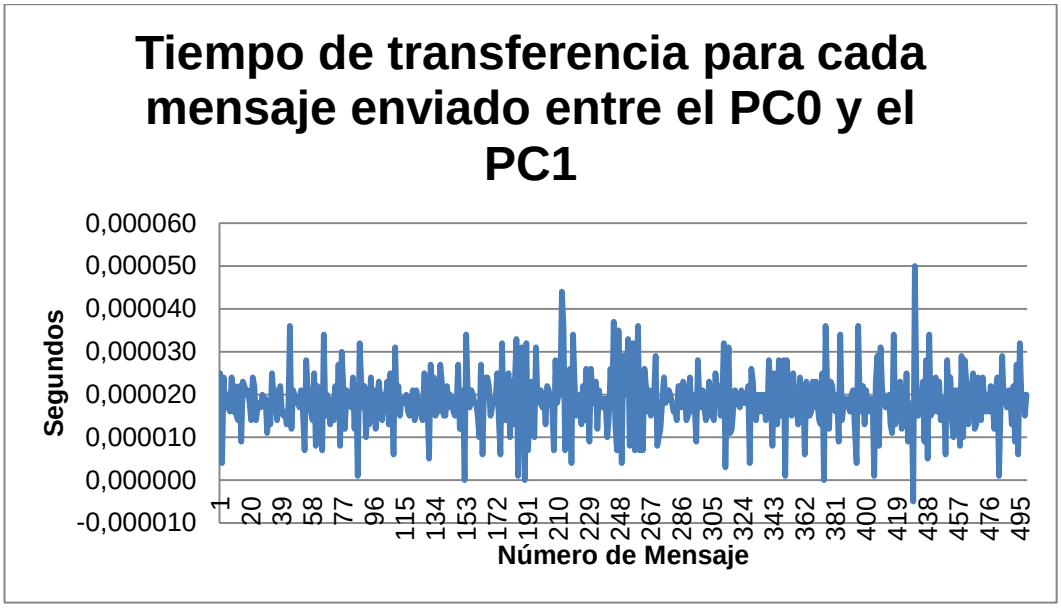
Topología de Red

Figura C36. Topología de red del experimento 8



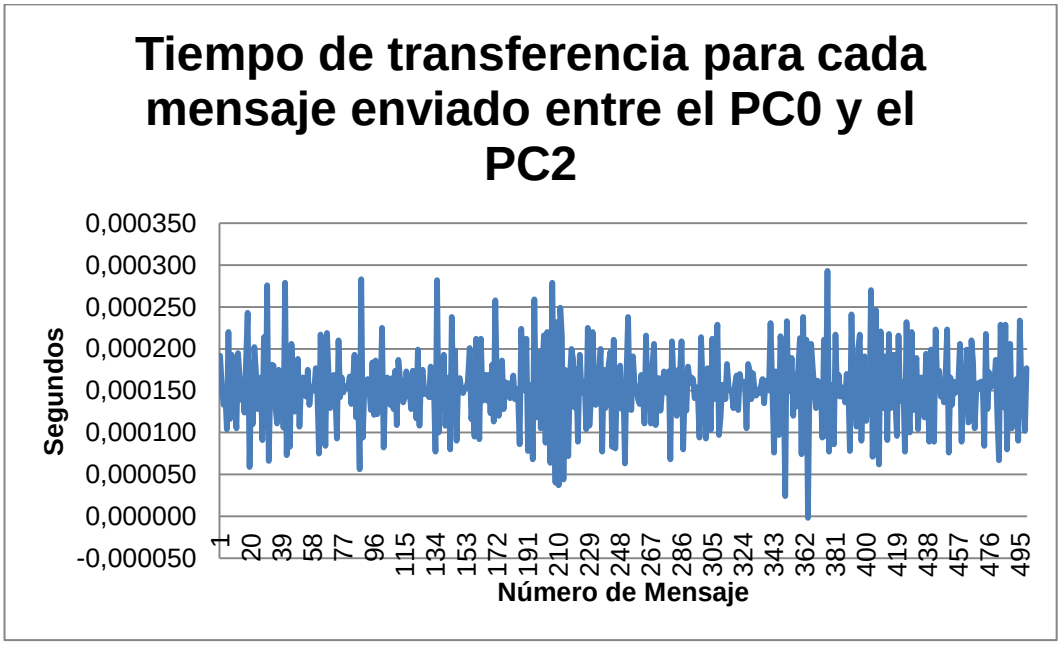
Fuente: Autor

Figura C37. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC1



Fuente: Autor

Figura C38. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC2



Fuente: Autor

Resultados

- Los mensajes enviados a través del protocolo se transmitieron de manera exitosa y rápida.
- El tiempo promedio de transferencia de los mensajes desde el PC0 hacia el PC1 fue de 0,000018 segundos y desde el PC0 hacia el PC2 fue de 0,000153 segundos.

INFORME

EXPERIMENTO 8-1

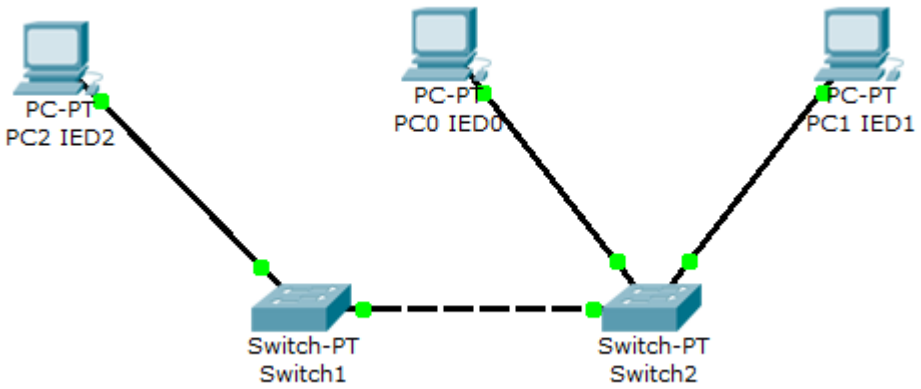
Fecha: Octubre 26 / 2012
Hora: 11AM a 11:15AM
Tráfico: Ninguno
Distancia entre los equipos: Menor a 2 metros
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. Y a su vez se enviaron 5 segundos después 500 mensajes GOOSE (Multicast) más desde el PC1 (IED1) con dirección MAC 00:22:19:1d:06:43 hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. Este experimento se realizó para comprobar la transmisión de mensajes GOOSE en una topología con dos switchs y dos productores de mensajes GOOSE.

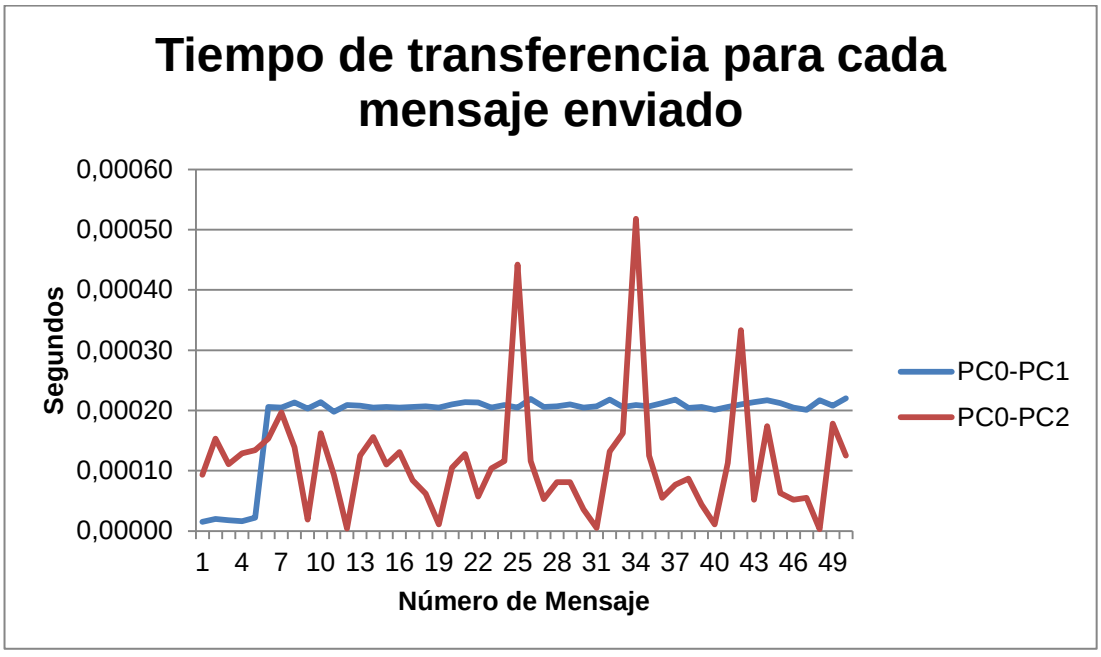
Topología de Red

Figura C39. Topología de red del experimento 8-1



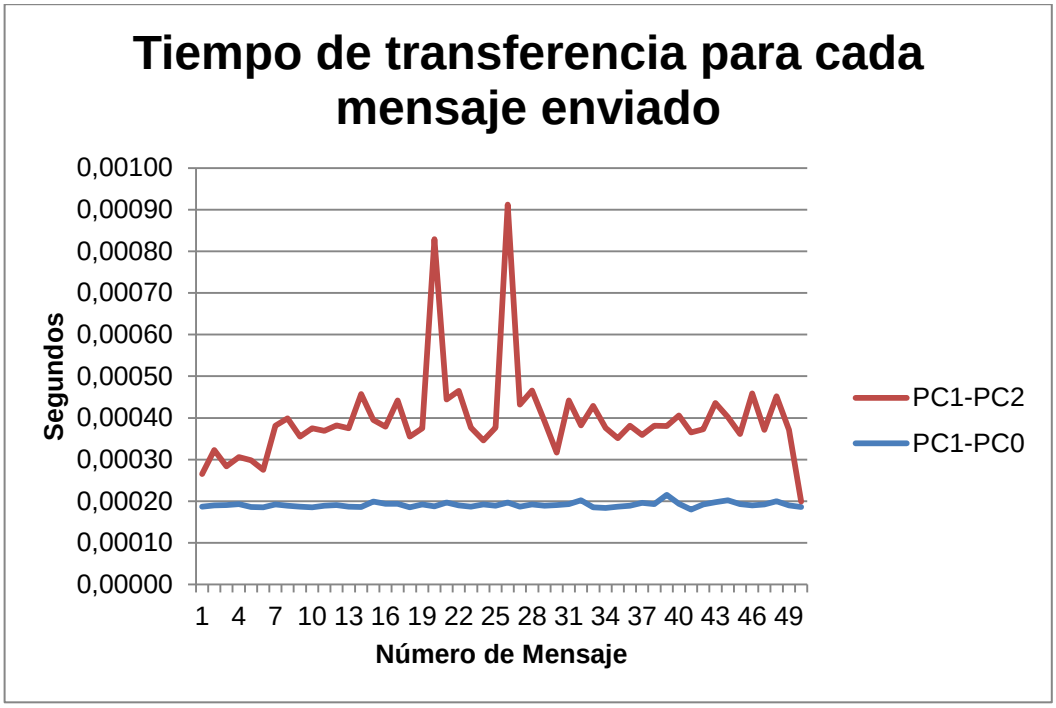
Fuente: Autor

Figura C40. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC1



Fuente: Autor

Figura C41. Tiempo de transferencia para cada mensaje enviado entre el PC0 y el PC2



Fuente: Autor

Resultados:

- Los mensajes se trasmitieron con éxito, de manera correcta y rápida cumpliendo el protocolo.
- El tiempo promedio para las transmisiones fue: Desde PC0 hacia PC1: 0,00021 segundos, desde PC0 hacia PC2: 0,00014 segundos, desde PC1 hacia PC0: 0,00020 segundos, desde PC1 hacia PC2: 0,0017 segundos.

Observaciones:

- Para llevar a cabo las gráficas anteriormente mostradas se tomo una muestra de los 50 primeros mensajes, ya que al existir tantos datos (1000

en el caso de este experimento) no se podía apreciar con claridad el comportamiento de la gráfica.

ANEXO D: INFORMES DE LOS EXPERIMENTOS REALIZADOS CON EL FIN DE OBSERVAR EL COMPORTAMIENTO DE LA TRANSMISIÓN DE MENSAJES RÁPIDOS EN UNA RED LOCAL ANTE ATAQUES A LA SEGURIDAD

INFORME
EXPERIMENTO A 1-1

Fecha: Octubre 08 / 2012
Hora: 2PM a 3:30PM
Tráfico: Ninguno
Ataque realizado: Denegación del servicio DoS – Inundación ICMP
Configuración VLAN: PC0, PC1 y PC2 (*vlan2*)
Distancia entre los equipos: Menor a 2 metros
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. A la vez se enviaron desde el PC2 (con la herramienta Ostinato) paquetes ICMP, hacia el PC0 con una velocidad determinada, realizando en total 6 pruebas. Ver la tabla 1.

Este experimento se realizó con el fin de provocar pérdida de conectividad por alto tráfico de red.

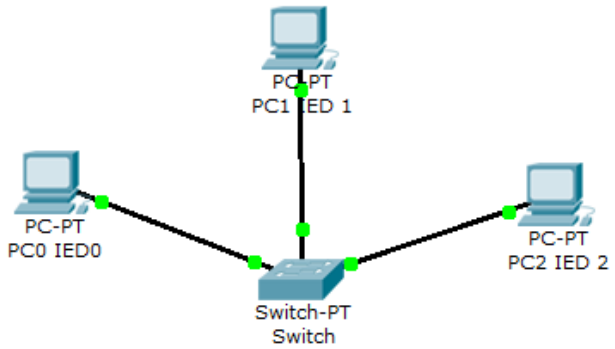
Tabla D1. Parámetros del experimento A1-1

N°	Número de paquetes	Velocidad (Paquetes/segundo)
1	500	1
2	1.000	2
3	10.000	20
4	100.000	200
5	1’000.000	2.000
6	10´000.000	20.000

Fuente: Autor

Topología de Red:

Figura D1. Topología de red del experimento A 1-1



Fuente: Autor

Resultados:

- Para las pruebas 1, 2 ,3 ,4 y 5 la transmisión de mensajes GOOSE fue exitosa. Los tráficos propuestos en estos experimentos no afectaron de ninguna manera la transmisión.
- El ataque realizado surgió un efecto hasta la prueba 6 (10’000.000 paquetes, con velocidad de 20.000 paquetes/segundo), el cual fue el

retardo de los mensajes GOOSE entre 1 y 3 ms más, al tiempo promedio de envío (1,00011 segundos). El ataque no denegó el servicio.

INFORME
EXPERIMENTO A 1-2

Fecha: Octubre 08 / 2012
Hora: 3:50PM a 4PM
Tráfico: Ninguno
Ataque realizado: Denegación del servicio DoS – Inundación ICMP
Configuración VLAN: PC0, PC1 y PC2 (*vlan2*)
Distancia entre los equipos: Menor a 2 metros
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

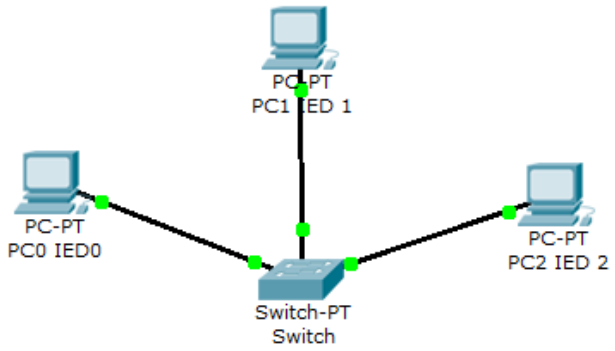
Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. A la vez se enviaron desde el PC2 (con la herramienta Ostinato) paquetes ICMP, hacia el PC0 con una velocidad de 20.000 paquetes/segundo y una cantidad de 10’000.000 paquetes. También se produjo un *ping* continuo (*Instrucción: ping –i 0.00000000001 IP-PC0)* desde el PC1 hacia el PC0

Este experimento se realizó con el fin de provocar pérdida de conectividad por alto tráfico de red.

Topología de Red:

Figura D2. Topología de red del experimento A 1-2



Fuente: Autor

Resultados:

- Los mensajes fueron trasmitidos de manera correcta y exitosa, el ataque hizo que cada mensaje GOOSE tardara entre 2 y 9 ms más que el tiempo promedio: 1,00011 segundos.

Observaciones:

- En la realización de este ataque el retardo que se causó en algunos mensajes fue variante durante la transmisión.

INFORME

EXPERIMENTO A 1-3

Fecha: Octubre 08 / 2012

Hora: 4PM a 4:10PM

Tráfico: Ninguno

Ataque realizado: Denegación del servicio DoS

Configuración VLAN: PC0, PC1 y PC2 (*vlan2*)

Distancia entre los equipos: Menor a 2 metros

Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

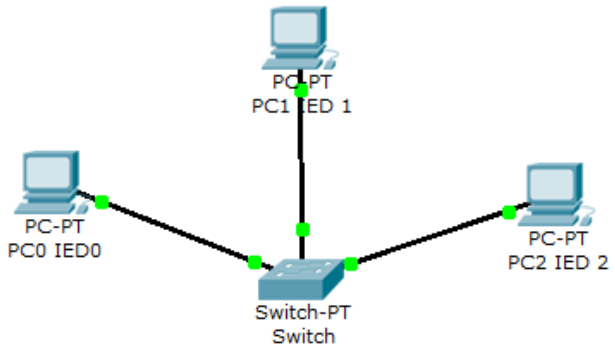
Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. A la vez se enviaron desde el PC2 (con la herramienta Ostinato) 12’000.000 paquetes Ethernet II dirigidos hacia el PC0 (MAC 00:22:2d:28:8d:ab) con una velocidad de 6’000.000 paquetes/segundo.

Este experimento se realizó con el fin de provocar pérdida de conectividad por alto tráfico de red y bloquear el switch.

Topología de Red:

Figura D3. Topología de red del experimento A 1-3



Fuente: Autor

Resultados:

- Los mensajes fueron transmitidos de manera correcta y exitosa. El ataque hizo que cada mensaje GOOSE tardara entre 3 y 10 ms más, que el tiempo promedio: 1,00011 segundos.

Observaciones:

- En la realización de este ataque el retardo que se causó en algunos mensajes se mantuvo constante entre mensajes.

INFORME
EXPERIMENTO A 2-1

Fecha: Octubre 08 / 2012
Hora: 4:40PM a 5PM
Tráfico: Ninguno
Ataque realizado: CAM Table Overflow o MAC Flooding
Configuración VLAN: PC0 y PC1 (*vlan2*) – PC2 (*vlan3*)
Distancia entre los equipos: Menor a 2 metros
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. El ataque CAM Table Overflow fue realizado desde el PC2 con dirección MAC 00:26:b9:1d:4b:77 con ayuda de la herramienta *macof* incluida en BackTrack. El

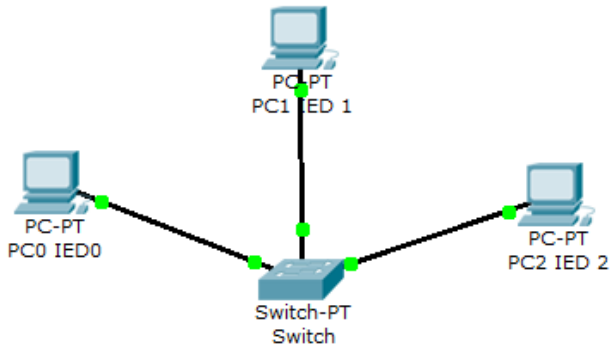
envío de mensajes GOOSE y el ataque CAM Table Overflow fueron iniciados a la vez.

La única instrucción ejecutada en BakcTrack fue *macof.*

Este experimento se realizó con el fin de interceptar los mensajes GOOSE con el ataque CAM Table Overflow.

Topología de Red:

Figura D4. Topología de red del experimento A 2-1



Fuente: Autor

Resultados:

- En este experimento el ataque realizado no obtuvo resultado exitoso. Se esperaba que con éste se lograran interceptar los mensajes GOOSE desde el PC2 pero el efecto no resultó. Esto se debe a que el ataque CAM Table Overflow se realizó desde una VLAN distinta a donde se enviaban los mensajes GOOSE y el ataque no evadió esta configuración del switch.

Observaciones:

- El ataque no se realizó desde la misma VLAN donde se enviaron los mensajes rápidos porque estos son enviados a una dirección multicast / brodcast; entonces no tendría sentido hacer un ataque de interceptación de datos a unos mensajes que llegan al PC atacante.

INFORME
EXPERIMENTO A 2-2

Fecha: Octubre 08 / 2012
Hora: 5PM a 5:20PM
Tráfico: Ninguno
Ataque realizado: CAM Table Overflow o MAC Flooding
Configuración VLAN: PC0 y PC1 (*vlan2*) – PC2 (*vlan3*)
Distancia entre los equipos: Menor a 2 metros
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

Descripción del Experimento:

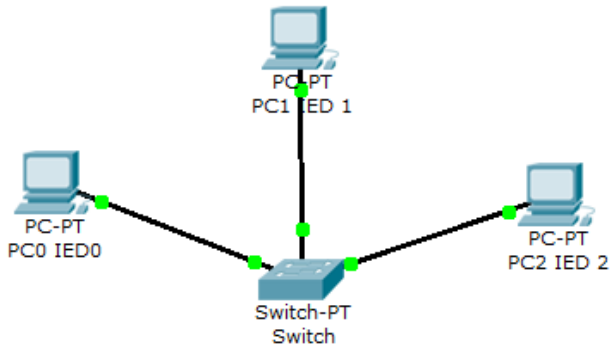
Se enviaron 500 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. El ataque CAM Table Overflow fue elaborado desde el PC2 con dirección MAC 00:26:b9:1d:4b:77 con ayuda de la herramienta *macof* incluida en BackTrack. El envío de mensajes GOOSE y el ataque CAM Table Overflow fueron iniciados a la vez.

La instrucción ejecutada en BakcTrack fue *macof –e 00:22:2d:28:8d:ab,* en esta se especifica la dirección MAC del dispositivo a la se quiere realizar el ataque.

Este experimento se cumplió con el fin de interceptar los mensajes GOOSE con el ataque CAM Table Overflow, especificando la MAC del equipo al cual se iba a atacar.

Topología de Red:

Figura D5. Topología de red del experimento A 1-2



Fuente: Autor

Resultados:

- En este experimento el ataque realizado no obtuvo resultado exitoso. Se esperaba que con éste se lograra interceptar los mensajes GOOSE desde el PC2 pero el efecto no resultó. Esto se debe a que el ataque CAM Table Overflow se realizó desde una VLAN distinta a donde se enviaban los mensajes GOOSE y el ataque no evadió esta configuración del switch.

Observaciones:

- El ataque no se realizó desde la misma VLAN donde se enviaron los mensajes rápidos porque estos son enviados a una dirección multicast / brodcast; entonces no tendría sentido hacer un ataque de interceptación de datos a unos mensajes que llegan al PC atacante.

NOTA: Otro efecto que se esperaba al ejecutar el ataque MAC Table Overflow (en los experimentos A 1-1 y A 1-2), según la fuente consultada[55] era denegación del servicio; recordemos que la herramienta *macof* envía una cantidad considerable de paquetes con distintas direcciones MAC, pero esta denegación del servicio tampoco sucedió. Lo anterior se debe a que la herramienta no envía las tramas suficientes para hacer que el servicio deje de funcionar o demore la transmisión de los mensajes rápidos.

INFORME

EXPERIMENTO A 3-1

Fecha: Octubre 10 / 2012
Hora: 11AM a 12PM
Tráfico: Ninguno
Ataque realizado: ARP Spoofing
Configuración VLAN: PC0, PC1 y PC2 (*vlan2*)
Distancia entre los equipos: Menor a 2 metros
Ubicación: Edificio LP Salón 203 (Sala de Redes) – Universidad Industrial de Santander

[55] Consultado en: http://www.undec.edu.ar/PDF/seguridad_capa2.pdf

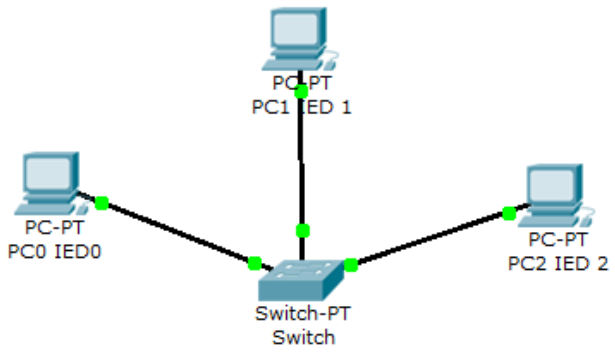
Descripción del Experimento:

Se enviaron 1000 mensajes GOOSE (Multicast) a través de la topología descrita en el informe, desde el PC0 (IED0) con dirección MAC 00:22:2d:28:8d:ab hacia la dirección multicast MAC 01:0C:CD:01:00:00 definida en la norma IEC 61850-8-1. El PC2 fue configurado para ser el atacante mediante la herramienta *ettercap-ng* incluida en *BackTrack*.

Este experimento se realizó con el fin de interceptar los mensajes GOOSE que viajaban por la topología descrita a continuación.

Topología de Red:

Figura D6. Topología de red del experimento 3-1



Fuente: Autor

Resultados:

- El ataque realizado no tuvo ningún efecto sobre los mensajes GOOSE. Se esperaba que con éste se lograra interceptar los mensajes rápidos desde el PC2 pero el ataque no resultó. Esto se debe a que el protocolo GOOSE no maneja la capa 3 del modelo OSI, y aunque el ataque ARP Spoofing está basado en la capa 2, éste se relaciona con la capa 3 (protocolo ARP) del

mismo modelo. Por lo anterior la tabla ARP no se actualiza con la transmisión de mensajes GOOSE y no permite efectuar el ataque.

- Los mensajes GOOSE enviados a través de la topología se transmitieron de manera correcta y exitosa.