ÁLVARO JOSÉ CERVELIÓN BASTIDAS

ANÁLISIS Y EVALUACIÓN DE DESEMPEÑO DE LOS PROTOCOLOS MODBUS Y DNP3 EN LA RED DE COMUNICACIONES DE UNA MICRORRED ELÉCTRICA

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Maestría en Electrónica y Telecomunicaciones

Popayán
2019

ÁLVARO JOSÉ CERVELIÓN BASTIDAS

ANÁLISIS Y EVALUACIÓN DE DESEMPEÑO DE LOS PROTOCOLOS MODBUS Y DNP3 EN LA RED DE COMUNICACIONES DE UNA MICRORRED ELÉCTRICA

Trabajo de Grado presentado a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la Universidad del Cauca
para la obtención del Título de

Magister en
Electrónica y Telecomunicaciones.

Director:
Ph.D. Guefry Leider Agredo Méndez
Co-Director:
Ph.D. Edgardo Javier Revelo Fuelagán

Popayán
2019

Dedicatoria

A Dios por brindarme la oportunidad de vivir esta experiencia,

a mis padres Aura y Álvaro por su apoyo incondicional,

a mi esposa Mary por su aliento y paciencia,

a mi hermano Oscar Javier

a mi familia y amigos

Agradecimientos

Al director y co-director de este trabajo de grado de maestría, doctores Guefry Agredo Méndez y Javier Revelo Fuelagán, por su dedicación y apoyo, sus orientaciones y sugerencias fueron fundamentales en el desarrollo de este trabajo.

Al Magister Harold Romo y al Doctor Pablo Jojoa, que como coordinadores del programa de maestría estuvieron pendientes del desarrollo del trabajo. A todos los docentes y compañeros con los que tuve la oportunidad de compartir.

Al Doctor Andrés Pantoja, Universidad de Nariño, por permitir desarrollar el estudio y la pasantía de investigación en el grupo de investigación GIIEE.

A todos aquellos que aportaron en el desarrollo de este trabajo.

Resumen

Las microrredes o *microgrids* se perfilan como la solución al crecimiento en la demanda de energía eléctrica, mejorando el control, eficiencia y fiabilidad en el suministro de electricidad. Estas funcionalidades y ventajas son posibles gracias a la infraestructura de medición avanzada (AMI, *Advanced Metering Infrastructure*) que permite la interacción de sistemas de medida y de comunicación con la red eléctrica convencional permitiendo ejecutar funciones como monitoreo en tiempo real y control en la red de distribución. Para el intercambio de información entre los componentes de la *microgrid* se utilizan diferentes tecnologías y protocolos de comunicación, que generalmente se validan en la instalación física, o de manera alternativa por simulación. En este trabajo se presenta una comparativa de desempeño de los protocolos Modbus y DNP3 encapsulados en TCP bajo tecnologías Ethernet y Wi-Fi para aplicaciones en *microgrids*, por medio de medidas reales y simulación. Para ello se diseñó e implementó la infraestructura AMI para la *microgrid* de la Universidad de Nariño, se utilizó el software de simulación NS-2 para el modelado de los protocolos y se analizaron parámetros de calidad de servicio como *delay* y *throughput* en diversos escenarios de aplicación.

Palabras Clave: AMI, DNP3, *microgrids*, Modbus, simulador NS-2.

x

Abstract

The microgrids are emerging as the solution to the growth in the demand for electricity, improving control, efficiency, and reliability in the supply of electricity. These functionalities and advantages are possible thanks to the advanced metering infrastructure (AMI) that allows the interaction of measurements and communication systems with the conventional electrical network allowing to execute functions such as real-time monitoring and control in the distribution network. For the exchange of information between the components of the microgrid, different technologies and communication protocols are used, which are generally validated in the physical installation or alternatively by simulation. This document presents, the outcomes of a performance comparison of the Modbus and DNP3 protocols encapsulated in TCP under Ethernet and Wi-Fi technologies for microgrids applications, by means of measurements and simulation. For this purpose, first the AMI infrastructure for the microgrid of the University of Nariño was designed and implemented, then the NS-2 simulation software was used for modeling the protocols and quality of service parameters, such as delay and throughput were analyzed in various application scenarios.

Keywords: AMI, DNP3, microgrids, Modbus, NS-2 simulator.

Contenido

Lista de Tablas

Lista de Figuras

Lista de Abreviaturas

ACK *Acknowledgment,* Reconocimiento.

ADU *Application Data Unit,* Unidad de Datos de Aplicación.

AMI *Advanced Metering Infrastructure*, Infraestructura de Medición Avanzada.

bps *Bits Per Second*, Bits Por Segundo.

CRC *Cyclic Redundancy Check*, Verificación de Redundancia Cíclica.

DER *Distributed Energy Resources*, Fuentes de Energía Distribuida.

DNP3 *Distributed Network Protocol* 3, Protocolo de Red de Distribución 3.

EMS *Energy Management System,* Sistema de Gestión de Energía.

EPA *Enhanced Performance Architecture,* Arquitectura de Rendimiento Mejorado.

FCS *Frame Check Sequence*, Secuencia de Verificación de Trama.

Gbps *Giga Bits Per Second*, Giga Bits Por Segundo.

HAN *Home Area Network*, Red de Área Doméstica.

HDLC *High-Level Data Link Control*, Control de Enlace de Datos de Alto Nivel.

HMI *Human-Machine Interface*, Interfaz Hombre-Máquina.

IED *Intelligent Electronic Device*, Dispositivo Electrónico Inteligente.

IP *Internet Protocol*, Protocolo de Internet.

LAN *Local Area Network*, Red de Área Local.

MAC *Media Access Control,* Control de Acceso al Medio.

Mbps *Mega Bits Per Second*, Mega Bits Por Segundo.

MIMO *Multiple-Input & Multiple-Output,* Múltiple entrada múltiple salida.

MTU *Master Terminal Unit*, Unidad Terminal Maestra.

NAM *Network AniMator,* Animador de Red.

NAN *Neighborhood Area Network,* Red de Área de Vecindario.

NS-2 *Network Simulator 2*, Simulador de Redes versión 2.

NS-3 *Network Simulator 3*, Simulador de Redes versión 3.

OFDMA *Orthogonal Frequency-Division Multiple Access*, Acceso Múltiple por División de Frecuencia Ortogonal.

OMS *Outage Management System,* Sistema de Gestión de Interrupciones.

OSI *Open System Interconnection,* Interconexión de Sistemas Abiertos.

PDU *Protocol Data Unit*, Unidad de Datos de Protocolo.

PLC *Power Line Communications*, Comunicaciones por Línea de Potencia.

QoS *Quality of Service*, Calidad de Servicio.

RTT *Round Trip Time,* Tiempo de Ida y Vuelta.

RTU *Remote Terminal Unit*, Unidad Terminal Remota.

SCADA *Supervisory, Control And Data Acquisition*, Supervisión, Control y Adquisición de Datos.

SFD *Start Frame Delimiter*, Delimitador de Inicio de Trama.

TCP *Transmission Control Protocol,* Protocolo de Control de Transmisión.

THD *Total Harmonic Distortion,* Distorsión Armónica Total.

ToU *Time of Use*, Tiempo de Uso.

UDP *User Datagram Protocol*, Protocolo de Datagramas de Usuario.

WAN *Wide Area Network,* Red de Área Amplia

Wi-Fi *Wireless Fidelity*, Fidelidad Inalámbrica.

WIMAX *Worldwide Interoperability for Microwave Access,* Interoperabilidad Mundial para Acceso por Microondas.

WLAN *Wireless Local Area Network*, Red de Área Local Inalámbrica.

Lista de Símbolos

P_t	Potencia de señal transmitida
$P_r(d)$	Potencia media recibida
$P_r(d_0)$	Potencia de referencia
G_t	Ganancia antena del transmisor
G_r	Ganancia antena del receptor
d	Distancia entre emisor y receptor
L	Constante de pérdidas del sistema
λ	Longitud de onda
h_t	Altura antena del transmisor
h_r	Altura antena del receptor
β	Exponente de pérdida de trayectoria
σ_{dB}	Desviación de sombreado
n	Cantidad mínima de eventos
$Z_{\alpha/2}$	Valor de la distribución normal
w	Ancho del intervalo de confianza

Capítulo 1

Introducción

La estructura del actual sistema de energía eléctrica fue desarrollada hace varias décadas y con el crecimiento en la demanda de los últimos años se ha convertido en un sistema ineficiente, propenso a las fallas que provocan interrupciones en el servicio, incapacidad para aumentar la generación, pérdidas en la transmisión, sobrecargas y caídas de tensión, entre otras [1]. Lo anterior ha motivado a que se generen grandes transformaciones en el sistema eléctrico con la aplicación de nuevas tecnologías en electrónica de potencia y de las Tecnologías de la Información y la Comunicación (TIC). Entre los dominios que componen la red eléctrica, el sistema de distribución es el que está sufriendo mayores cambios con la implantación de las *microgrids* [2,3].

Una *microgrid* es una red eléctrica de próxima generación que integra fuentes de energía distribuida, generalmente de energías renovables como solar, biomasa y eólica, entre otras [4]; en la que la distribución y gestión de la energía eléctrica se actualiza incorporando comunicaciones bidireccionales y herramientas informáticas para mejorar el control, eficiencia, fiabilidad y seguridad en el suministro de electricidad [5]. Estas funcionalidades no serían posibles sin un nuevo sistema de medida y de comunicación denominado Infraestructura de Medición Avanzada (AMI, *Advanced Metering Infrastructure*), que recopila los datos de consumo de energía en un centro de gestión, ejecuta procesos de monitoreo y control y que permite el despliegue de nuevas aplicaciones como conexión y desconexión remota, detección de interrupciones, identificación temprana de posibles fallas, administración de energía, entre otras [6].

Para garantizar un rendimiento apropiado de la AMI se debe diseñar cuidadosamente la red de comunicaciones y elegir los equipos de medida y protocolos de comunicación que permitan el intercambio de información dentro de una ventana de tiempo predefinido, es así, que un rendimiento inapropiado no solo limita la eficiencia energética y la calidad del servicio, sino que también genera posibles daños al sistema eléctrico [7]. La norma IEEE 2030 clasifica los eventos de intercambio de información en los sistemas de energía en varias categorías y recomienda que el retardo máximo para datos de protección sea de 16 ms, para monitoreo en tiempo real de 100 ms y de 2 segundos para sistemas SCADA [8].

Por otra parte, en el diseño de la red de comunicaciones generalmente se integran diferentes tecnologías, considerando factores como ubicación geográfica, accesibilidad, costos, equipos a interconectar, entre otros [9]. Además, actualmente existe una gran variedad de medidores inteligentes que se deben seleccionar por sus características de medición y por el protocolo de comunicación que soporta. Dos de los protocolos más utilizados son Modbus y DNP3, su expansión se debe a que son de tipo abierto, lo que permite su integración con equipos de diversas marcas y con cualquier sistema de gestión. La elección de uno de estos protocolos es importante, pues generalmente un medidor que soporta DNP3, por ser más robusto, tiene mayor costo que uno que soporte Modbus; este es un aspecto importante a considerar en el diseño de la AMI, más aún cuando el número de puntos de medición es elevado.

Para determinar si la red y el protocolo de comunicaciones cumplen con los requerimientos de la AMI se debe realizar una evaluación de desempeño. Para ello en [10] y [11] se plantea que este proceso debe combinar medidas reales y de simulación. Las simulaciones tienen algunas ventajas como: los escenarios de red se pueden construir y modificar fácilmente, los datos se pueden recopilar con facilidad, se pueden modelar topologías de red a gran escala que podrían ser muy costosas de implementar, tienen bajos costos de experimentación en el caso de simuladores de licencia libre. Por otro lado, también tienen algunas desventajas como: en algunas ocasiones la ausencia de factores externos como errores de transmisión o interferencias se traduce en que los resultados de las simulaciones puedan alejarse demasiado de los reales. Además, cuando se desea simular una topología con gran cantidad de nodos y durante un largo periodo de tiempo, esto puede consumir grandes recursos del sistema donde se simula [12].

Algunos de los simuladores ampliamente utilizados son OPNET, NS-2 y NS-3. OPNET es capaz de simular una gran variedad de redes, con características como flujos de datos, paquetes perdidos, caída de enlaces, entre otras. Este es un simulador utilizado primordialmente por grandes compañías de telecomunicaciones por sus altos costos de licenciamiento. Por su parte, NS-2 y NS-3 son simuladores de código libre y a pesar de que NS-3 es una versión más actual, no dispone de todos los modelos con los que cuenta NS-2. Además, NS-2 cuenta con una bibliografía extensa indispensable para el desarrollo de nuevas investigaciones. NS-2 es un simulador de eventos discretos ampliamente utilizado en la academia por sus características, como código abierto, fácil configuración de escenarios para tecnologías cableadas e inalámbricas y la posibilidad de añadir nuevas funciones en su biblioteca [13]. Por las características anteriores, el presente estudio se realizó con el simulador NS-2.

Por lo anterior, en este documento se presentan los resultados de la evaluación de desempeño de los protocolos Modbus y DNP3 sobre tecnologías Ethernet y Wi-Fi en un escenario de *microgrid* bajo los parámetros de Calidad de Servicio (QoS, *Quality of Service*) de *delay* y *throughput.* Lo anterior busca estudiar las características, funcionalidades y potencialidades que permitan determinar la viabilidad de cada protocolo en la infraestructura AMI. Para las pruebas de campo, fue necesario el diseño e implementación de una AMI según los requerimientos de la *microgrid* del Campus de la Universidad de Nariño. Por su parte, para la evaluación de desempeño por simulación se utilizó la herramienta *Network Simulator* 2 (NS-2) que permite añadir nuevas funciones en su biblioteca [14], esta característica permitió modelar Modbus y DNP3 en el núcleo de NS-2, ya que estos protocolos no están disponibles en la versión original; para la validación se compararon los resultados con los datos reales. Una vez los modelos son validados, se realizan diferentes estudios, como modificar la distancia entre nodos y aumentar la cantidad de Bytes transmitidos

1.1. Trabajos relacionados

A continuación se describen algunos estudios referentes a la evaluación de desempeño de redes de comunicaciones y de los protocolos Modbus y DNP3 en ambientes de redes inteligentes.

Performance Testing Framework in a Heterogeneous and Hybrid Smart Grid Communication Network [15]. En este estudio se presenta un procedimiento para realizar la evaluación del rendimiento del sistema de comunicación en una red inteligente. Se especifica los tipos de medios de comunicación y tecnologías, los criterios de evaluación y herramientas de software para llevar a cabo las pruebas. Para este fin se hace uso de herramientas como Nagios, Iperf y Bing.

Testing Methodology for Performance Evaluation of Communication Systems for Smart Grid [16]. En este trabajo, se propone una metodología para la prueba y evaluación de sistemas de comunicación en redes inteligentes, especialmente para sistemas de Comunicaciones mediante Línea de Potencia (PLC, *Power Line Communications*). La metodología formula diversos escenarios y fue probada en una red eléctrica real, para lo cual se utilizaron herramientas como FlowTester, Flowping, Iperf y UDA.

On Network Performance Evaluation toward the Smart Grid: A Case Study of DNP3 over TCP/IP [17]. En el artículo se realiza un estudio para determinar el rendimiento del protocolo DNP3/TCP en un SCADA, para comprobar si cumple con los requisitos de tiempo que las aplicaciones de redes inteligentes requieren. Para este fin se implementa una *microgrid* inteligente denominada *Green Hub* en la cual se realizan las mediciones. Los resultados muestran que aunque DNP3/TCP es ampliamente utilizado en redes inteligentes, en *Green Hub* no se puede usar para aplicaciones con limitaciones de retardo menores de 16 ms, como en relés de protección.

An Empirical Study of Communication Infrastructures Towards the Smart Grid: Design, Implementation, and Evaluation [18]. En el documento, se aborda por medio del estudio de caso de un proyecto de demostración de redes inteligentes, los sistemas de gestión y la entrega de energía eléctrica renovable. Se investigan los escenarios de comunicación y se adopta el protocolo DNP3 sobre TCP/IP. En el banco de pruebas, se mide el rendimiento en la entrega de mensajes DNP3 dentro de la infraestructura de comunicación. Se concluye que aunque DNP3 sobre TCP/IP es ampliamente considerado como una solución de comunicación de red inteligente, no puede satisfacer los requisitos de comunicación en algunos escenarios de tiempo

crítico, como las protecciones de retransmisión, que reclaman una mayor optimización en la eficiencia del protocolo de DNP3.

Performance Evaluation of Future AMI Applications in Smart Grid Neighborhood Area Networks [19]. En este artículo, se estudia el funcionamiento de AMI en Colombia. Teniendo en cuenta que ya se han implementado las primeras aproximaciones a AMI que se enfocan principalmente en lecturas de consumo automatizadas, no hay certeza de que el tráfico de diferentes naturalezas, correspondientes a futuras aplicaciones AMI, pueda ser soportado por estas implementaciones iniciales. Por lo anterior, se desarrolla la evaluación de las tecnologías actuales que entregan tráfico de futuras aplicaciones AMI a través de simulaciones. Se realiza una revisión de las tecnologías de comunicación que se emplean actualmente para AMI en Colombia, y en una caracterización de las futuras aplicaciones de AMI. Con el estudio se busca una mejor comprensión de los principales desafíos que tendrán que afrontarse para el desarrollo y la mejora de las redes AMI en el país.

Performance Evaluation of the DNP3 Protocol for Smart Grid Applications over IEEE 802.3/802.11 Networks and Heterogeneous Traffic [20]. En este artículo, se evalúa el rendimiento de DNP3 encapsulado sobre TCP/IP en el simulador de código abierto NS-2. El escenario es una red Ethernet IEEE 802.3, el propósito del trabajo es investigar la viabilidad de utilizar DNP3 en una red que transporte tráfico heterogéneo, para información de monitoreo y protección, midiendo el retraso necesario para cada proceso.

Simulation in NS-2 of DNP3 Protocol Encapsulated over TCP/IP in Smart Grid Applications [21]. Este trabajo, describe la implementación del Protocolo DNP3 en un simulador de eventos discretos. El simulador elegido es NS-2 porque es *open source,* facilita el desarrollo de escenarios de redes de comunicaciones considerando los protocolos involucrados, en tecnologías inalámbricas o cableadas. El objetivo de este trabajo es desarrollar el encapsulado del protocolo DNP3 sobre TCP/IP y analizar su comportamiento en una red de dimensiones mediana a grande para aplicaciones de redes inteligentes.

Performance Analysis of Smart Grid Communication Protocol DNP3 over TCP/IP in a Heterogeneous Traffic Environment [22]. En el trabajo se presentan los resultados de simulación del protocolo DNP3/TCP para aplicaciones en redes inteligentes, utilizando el simulador NS-2. El objetivo del trabajo es verificar el rendimiento del protocolo DNP3 en una red LAN heterogénea. Los resultados indican que para una capacidad de canal entre 60 y 85 por ciento, DNP3 funciona correctamente con una baja pérdida de paquetes y un bajo retardo, sin embargo, para valores de tráfico superiores al 85 por ciento, el uso de DNP3 es inviable a causa de que la pérdida de información, las retransmisiones y el *delay* se incrementan significativamente.

Análise de Desempenho de Rede Smart Grid Protocolo de Comunicação DNP3 Sobre IEEE 802.11 [23]. Este documento, presenta un estudio sobre el uso de IEEE 802.11 WLAN y IEEE 802.3 LAN en el sistema de comunicaciones de redes inteligentes a través del simulador NS-2. El objetivo de este trabajo es establecer y ejecutar escenarios experimentales del protocolo de comunicación DNP3 encapsulado en TCP/IP en una red que contiene cinco nodos inalámbricos, un punto de acceso y un nodo cableado, con el fin de proponer una alternativa para reducir los costos de enlace entre los medidores inteligentes y las utilidades que utilizan los servicios de banda ancha existentes. Los resultados de la simulación muestran que el uso de este servicio de banda ancha para la automatización de los medidores inteligentes es prometedor.

DNP3 integration performance with IEEE 802.11 in Smart Grid Applications Through the CDF of delay and Jitter [24]. El articulo presenta los resultados de simulación realizada en NS-2 del protocolo DNP3 sobre TCP/IP que opera en una red inalámbrica IEEE 802.11b Ad-hoc, en aplicaciones de redes inteligentes. En el trabajo se analiza el rendimiento de la comunicación de datos del protocolo DNP3 en una red WLAN Ad-hoc en relación con la distancia y los saltos. Los resultados muestran que DNP3 funciona correctamente en una red inalámbrica en donde un nodo realice hasta tres saltos, cumpliendo con los requisitos de entrega de información en relación con el retraso para la implementación del monitoreo en redes inteligentes.

Simulação Computacional do Protocolo de Comunicação em Aplicações de Smart Grid [25]. El articulo describe la simulación del protocolo DNP3/TCP para aplicaciones en redes inteligentes, utilizando NS-2. El objetivo es evaluar el funcionamiento del Protocolo DNP3 en el simulador NS-2 utilizando algunos escenarios pequeños con topología punto a punto.

Proposal DNP3 Protocol Simulation on NS-2 in IEEE 802.11g Wireless Network Ad Hoc Over TCP/IP in Smart Grid Applications [26]. En el estudio, se propone una evaluación del rendimiento de DNP3 a través de una red inalámbrica 802.11g ad hoc, encapsulado en TCP/IP utilizando el NS-2. El objetivo es investigar la factibilidad de usar DNP3 a través de una red ad hoc, para monitoreo y la tele protección, midiendo el retraso requerido para completar el envío de mensajes en un sistema de auto recuperación.

Simulation of the DNP3 Protocol over TCP/IP on a Network IEEE 802.11g Ad-hoc With Smart Meter [27]. En el trabajo se realiza una evaluación del rendimiento de DNP3/TCP a través de una red inalámbrica IEEE 802.11g ad hoc, utilizando el simulador NS-2. El objetivo del trabajo es investigar la viabilidad de utilizar el protocolo en una red residencial con la conexión de medidores inteligentes. Con los resultados de la simulación se conoce si la combinación de red una ad hoc inalámbrica 802.11g y DNP3 es viable para aplicaciones de redes inteligentes de bajo costo para monitorear el consumo de energía eléctrica en tiempo real.

Análise e desempenho do protocolo de comunicação DNP3 sobre redes wireless em aplicações smart grid [28]. El trabajo presenta los resultados de simulación del protocolo DNP3 sobre TCP/IP que opera en una red inalámbrica IEEE 802.11b para aplicaciones de red inteligente. Las simulaciones se realizaron en NS-2 y se analiza y verifica el rendimiento de comunicación de datos de DNP3 en una red inalámbrica ad hoc, considerando el retraso y la tasa de error de paquetes de acuerdo con la distancia y el número de saltos. A pesar de que los resultados de la simulación muestran un amplio rango de retardo entre los nodos, todas las estaciones remotas obtienen un rendimiento factible para el monitoreo en tiempo real.

A NS-2 simulation model for DNP3 protocol over IEEE 802.15.4 wireless protocol toward low cost simulation of Smart Grid applications [29]. En el artículo se propone el modelamiento de la integración del protocolo DNP3 y el estándar inalámbrico IEEE 802.15.4. Se utiliza NS-2 para realizar simulaciones de redes de bajo costo con aplicaciones en redes inteligentes. Se evalúa de forma eficiente un escenario punto a punto, teniendo en cuenta conectividad y *delay*. Se plantea realizar simulaciones con topologías más complejas para que otras características de la red inteligente puedan ser evaluadas.

Performance Analysis of the DNP3 Protocol Over IEEE 802.11g Wireless Network in Smart Grid Application through of the Simulation in the NS-2 [30]. El objetivo de este trabajo fue simular en NS-2 el protocolo DNP3/TCP sobre una red inalámbrica IEEE 802.11g ad hoc en aplicaciones de red inteligente con velocidades de transmisión de hasta 54 Mb/s. El escenario simulado se compone de una estación maestra y ocho estaciones externas. Los resultados demuestran que el tiempo de demora en la transmisión de información es muy bajo, con tiempos de demora en el peor de los casos con un 90% de paquetes transmitidos en 4 ms, y en el mejor caso con el 90% de los paquetes transmitidos en 1,1 ms. Estos resultados muestran que DNP3 sobre una red inalámbrica 802.11g puede ser utilizado en sistemas de protección y monitoreo en tiempo real.

Simulação Aplicada à Infraestrutura da Rede Elétrica com Smart Grid Empregando Comunicação Sem Fio [31]. Este artículo presenta un estudio sobre el uso de redes WLAN IEEE 802.11 en el sistema de comunicaciones de redes inteligentes, utilizado el simulador NS-2. Se ejecutan simulaciones del protocolo DNP3 encapsulado sobre TCP/IP en redes WLAN, con el fin de evaluar el desempeño del sistema de monitoreo. Los resultados demuestran que DNP3 presenta la funcionalidad esperada cuando se utiliza en redes inalámbricas 802.11 sin movilidad y sin saltos.

Simulation of the DNP3 communication protocol over tcp/ip protocol on a 802.11g wireless network in smart grid applications [32]. En el trabajo se simula en NS-2 el protocolo DNP3/TCP en una red inalámbrica 802.11g para aplicaciones de red inteligente. El objetivo es verificar el rendimiento en la transmisión de la información DNP3 en relación con los retardos de transmisión de paquetes de datos,

de acuerdo con la distancia y el número de saltos. En los resultados se observa que las terminales con un solo salto obtuvieron mejores resultados en comparación con otras con más saltos. Los resultados demostraron un gran rendimiento con hasta cuatro saltos y retrasos de menos de cinco milisegundos, ideales para aplicaciones de protección de red y monitoreo en tiempo real.

Ns-Modbus: Integration of Modbus with ns-3 Network Simulator [33]. El trabajo se enfoca en el protocolo Modbus, un protocolo de comunicaciones estándar ampliamente utilizado en la industria que inicialmente fue diseñado para redes de comunicación serie. El protocolo surgió cuando se integró a Ethernet con lo que ganó popularidad y fue introducido al campo de las redes inteligentes. El objetivo de la investigación es integrar Modbus TCP y UDP con el simulador de red NS-3, con el fin de comparar la sobrecarga de las dos implementaciones.

Performance Evaluation for Modbus/TCP Using Network Simulator NS-3 [34]. En este trabajo se analiza el rendimiento del protocolo Modbus/TCP en el simulador NS-3. La evaluación del rendimiento se centra en el tiempo de respuesta según la cantidad de nodos y la topología implementada. Los resultados indican que la topología de bus tiene un mejor rendimiento que la topología en estrella para la interconexión de hasta 50 nodos. Después de los 50 nodos, el jitter y las retransmisiones aumentan en la topología bus, mientras que la topología en estrella tiene un tiempo de respuesta constante, que es adecuado para un sistema en tiempo real.

1.2. Referentes Teóricos

Esta sección presenta los fundamentos teóricos necesarios para proponer el diseño de la infraestructura AMI, centrándose en la red de comunicaciones, donde se consideran aspectos como topologías y tecnologías de comunicación. Además, se estudian las características de los protocolos Modbus y DNP3 para su modelado y la revisión de las herramientas necesaria para el desarrollo del estudio.

1.2.1. Infraestructura de Medición Avanzada (AMI)

Una Infraestructura de Medición Avanzada (AMI, *Advanced Metering Infrastructure*) se define como un sistema capaz de recolectar información de medición cada cierto periodo, con el fin de transmitirla adecuadamente por un canal de comunicaciones para su correcta recepción en un sistema de recolección y análisis de datos [35]. En la figura 1.1 se puede observar la arquitectura general de la AMI que consta de tres componentes principales clasificados en niveles, los cuales se describen a continuación [36].

Nivel 1: es la capa superior de la infraestructura y corresponde al centro de gestión de datos. En esta capa se recolecta la información de los medidores, para su respectivo análisis, procesamiento y almacenamiento, lo cual se hace con la ayuda de un *software* de Supervisión, Control y Adquisición de Datos (SCADA, Supervisory *Control and Data Acquisition*). Además, se pueden implementar otras aplicaciones como el Sistema de Gestión de Energía (EMS, *Energy Management System*) que se enfoca en aplicaciones de control y programación de generación y el Sistema de Gestión de Interrupciones (OMS, *Outage Management System*), un sistema utilizado para restablecer la energía después de cortes inesperados.

Nivel 2: es la capa de acceso, la cual provee los canales de comunicación que hacen posible la comunicación bidireccional entre el centro de gestión y los medidores inteligentes.

Nivel 3: corresponde a los medidores inteligentes, que se encargan del registro de variables asociadas al consumo eléctrico.

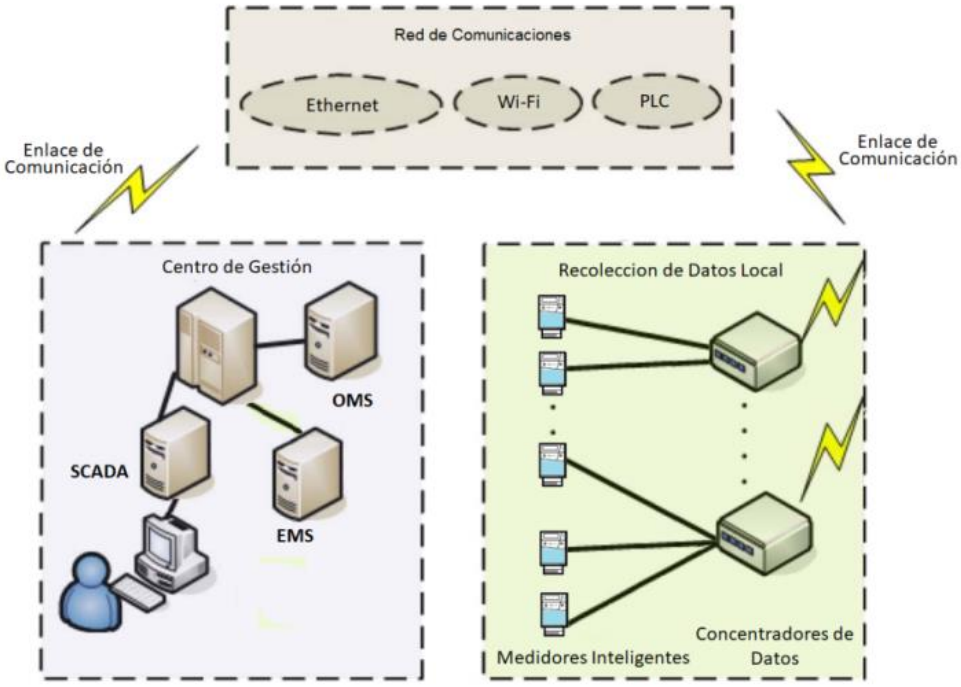


Figura 1.1. Diagrama de bloques AMI. Adaptada de: [37].

Una vez definidos los componentes de la AMI, se presentan aspectos detallados de cada nivel.

1.2.2. SCADA

Es un *software* de control que se comunica con los dispositivos de campo, en este caso los medidores inteligentes, que permite monitorear y controlar un proceso de forma automática desde un servidor principal. Los elementos que componen el sistema SCADA se identifican a continuación [38]:

Interfaz Hombre-Máquina (HMI, *Human-Machine Interface*): es el *software* encargado de presentar la información a través de una interfaz gráfica, con el propósito de hacer más amigable la interacción entre el operador y el sistema, permite supervisar el estado de un proceso, interrumpir manualmente las operaciones de control automático, modificar valores de referencias, entre otras.

Unidad Terminal Maestra (MTU, *Master Terminal Unit*): es la unidad central, generalmente es un ordenador ubicado en un centro de gestión donde se ejecutan las acciones de control (programado), con base en los valores actuales de las variables. Además, es la responsable de reunir la información de los diferentes dispositivos, del almacenamiento y procesamiento de los datos de forma que otra aplicación o dispositivo pueda acceder a ellos como el *software* HMI.

Unidad Terminal Remota (RTU, *Remote Terminal Unit*): son las estaciones ubicadas en campo y responsables de recoger y enviar la información del proceso a la MTU. En el ambiente de *microgrids* el encargado de esta función son los medidores inteligentes, denominado Dispositivo Electrónico Inteligente (IED, *Intelligent Electronic Device*) por sus funcionalidades adicionales más potentes como rutinas de control, entre otras.

Por otro lado, las funciones y prestaciones principales de un SCADA se presentan a continuación:

- Adquisición de datos: para el procesamiento y almacenamiento de la información recibida en una base de datos.
- Supervisión: para observar desde un monitor la evolución de las variables de control.
- Control: para modificar la evolución del proceso actuando sobre el proceso mediante salidas conectadas a actuadores.
- Presentación: representación gráfica de los datos mediante la HMI.
- Procesamiento: los datos adquiridos se procesan para gestión de la calidad, control estadístico, gestión de la producción y gestión administrativa y financiera.
- Alarmas: se crean paneles de alarma con registro de incidencias, que exigen la presencia del operador, mediante el monitoreo de valores y umbrales determinados.
- Reportes: genera reportes históricos a partir de los datos almacenados, estos se generan periódicamente o de forma manual.

1.2.3. Medidores inteligentes

Un medidor inteligente es un dispositivo que mide y registra variables eléctricas, como el consumo o producción de energía y que soporta comunicación bidireccional permitiendo que los datos sean leídos remotamente y recibir información de configuración y control. Algunas de sus funciones son las siguientes [39]:

- Registro y almacenamiento de datos: los medidores inteligentes están equipados con memorias no volátiles que permiten el almacenamiento de variables y eventos que pueden seleccionarse, al igual que el intervalo de almacenamiento.
- Control de carga: algunos medidores inteligentes están equipados con salidas digitales que permiten habilitar/deshabilitar el suministro de energía a una carga específica.
- Tiempo de uso: los medidores inteligentes pueden registrar un consumo detallado del Tiempo de Uso (ToU, *Time of Use*), que se refiere a la capacidad de un medidor para registrar cuándo se produce el consumo en lugar de cuánto se consume.

Por otro lado, para la elección del medidor inteligente se deben considerar los siguientes aspectos:

- Precisión: se designa por un número (índice de clase) que expresa el límite de error porcentual admisible de la magnitud medida. De acuerdo al punto de instalación, carga a medir, nivel de tensión, tipo de cliente y otros parámetros, debe utilizarse un medidor de una clase determinada. Comúnmente las clases de medidores utilizados son: 2, 1, 0.5, 0.5S, 0.2 y 0.2S, siendo este último el de mayor exactitud.
- Variables eléctricas: se debe verificar las variables que soporta el medidor, entre las cuales las más comunes son: voltaje, corriente, potencia activa, potencia reactiva, factor de potencia, frecuencia, entre otras
- Rangos de medición: indican la capacidad de medición que soporta el medidor. Para el voltaje se encuentran disponibles rangos que van desde 120v, 220v, 380v y 800v y que además pueden ampliarse utilizando trasformadores de potencia. Para la corriente, algunos medidores miden

directamente hasta 100 amperios o hasta varios miles de amperios utilizando transformadores de corriente.

- Puertos de comunicación: los medidores inteligentes comúnmente se encuentran equipados con puertos ópticos, puerto RS-485, puerto Ethernet, Wi-Fi, GSM, entre otros. La elección se fundamenta en la tecnología que se implementa en la red de comunicaciones.
- Módulos adicionales: muchos medidores inteligentes permiten la conexión de módulos extra que permiten realizar tareas adicionales. Los módulos más comunes son: módulos de comunicación, módulos de salidas/entradas digitales, módulos de salidas/entradas análogas, entre otros.
- Protocolos de comunicación: los protocolos se dividen en abiertos y propietarios. Los abiertos permiten la integración del equipo en cualquier sistema de gestión, por su parte los protocolos propietarios solo permiten la gestión del equipo desde un *software* propietario.

1.2.4. Red de comunicaciones

La red de comunicaciones es el componente principal de la AMI, pues debe brindar los canales de comunicación que se ajusten a las necesidades de cada aplicación. Para su diseño se deben considerar aspectos como el tamaño de la red, según el área de cobertura requerida, topología de comunicación elegida por el tipo de conexión de los elementos de la red y de la topografía del terreno y finalmente, los protocolos de comunicación que ofrecen el conjunto de pautas que posibilitan la comunicación entre los elementos que forman parte del sistema. En las siguientes secciones se describe cada uno de los aspectos anteriormente mencionados.

1.2.5. Jerarquía de las redes de comunicaciones

Según el área de cobertura, las redes de comunicaciones para AMI se pueden dividir jerárquicamente en tres. A continuación se describe cada una de ellas.

Red de Área Doméstica. La Red de Área Doméstica (HAN, *Home Area Network*) pertenece al dominio del cliente y se compone de electrodomésticos y sensores. Estos dispositivos reportan información del uso de energía al medidor inteligente que a su vez la envía a la red central. Los medidores inteligentes también reciben instrucciones sobre el uso de electricidad y pueden activar o desactivar cargas [40].

Red de Área de Vecindario. La Red de Área de Vecindario (NAN, *Neighborhood Area Network*) tiene como función permitir la comunicación de los medidores inteligentes con el centro de gestión. Las NAN se despliegan al aire libre y pueden constar de varios medidores inteligentes hasta miles de ellos en una cobertura de hasta 10 kilómetros cuadrados [41].

Red de Área Amplia. La Red de Área Amplia (WAN, *Wide Area Network*) conecta varias NAN con la red de *backhaul* o *Gateways* para remitir la información al centro de gestión. El área de cobertura es mucho mayor de alrededor de miles de kilómetros cuadrados [41]. En la figura 1.2 se aprecia el área de cobertura de cada red.

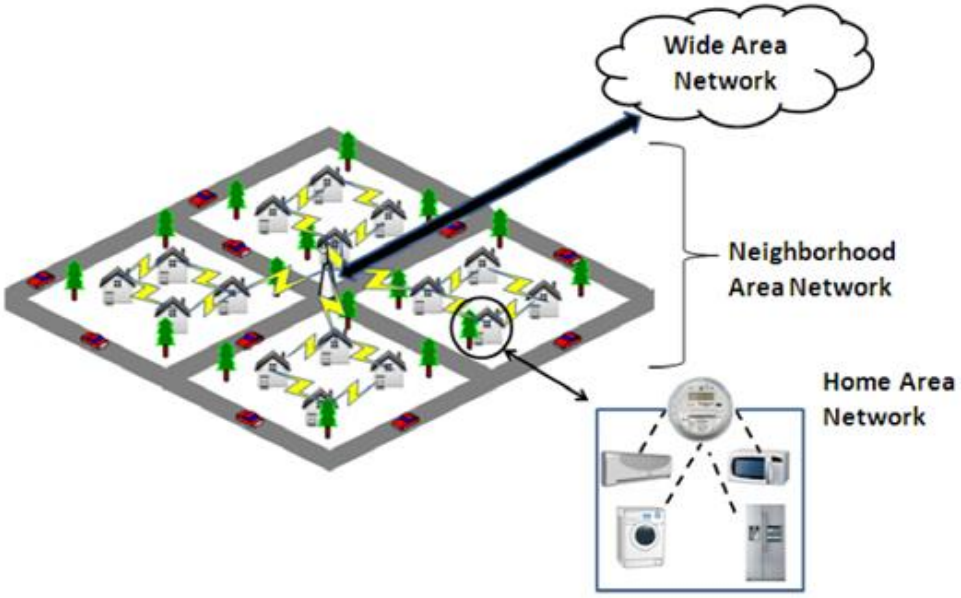


Figura 1.2. Red de comunicaciones HAN, NAN y WAN en redes inteligentes [41].

1.2.6. Topologías de Comunicación

La topología es el arreglo físico de los elementos dentro de una red y define la cadena de comunicación usada por los nodos que conforman la red para comunicarse. Las topologías usualmente empleadas en AMI son la topología estrella, bus, árbol y malla [7]. A continuación se presentan las características, ventajas y desventajas de cada una de ellas.

Topología en Estrella. Posee un nodo central que se conecta directamente con los nodos periféricos, lo que permite una mayor velocidad de comunicación entre estos nodos. El fallo de un nodo periférico no afecta el funcionamiento de la red. Las desventajas de esta topología son que toda la carga de tráfico recae sobre el nodo central y un fallo sobre este nodo puede dejar inoperante a toda la red.

Topología en Bus. Tiene un único canal de comunicaciones denominado bus al cual se conectan los diferentes nodos. De esta forma todos los nodos comparten el mismo canal para comunicarse entre sí. Requiere menos cable que una topología estrella y es fácil conectar nuevos nodos, lo que permite su extensibilidad y su fácil expansión. Las desventajas son que es difícil detectar el origen de un problema ante un fallo de la red y que toda la red fallaría si hubiera una ruptura en el cable principal o bus de datos.

Topología en Árbol. Esta topología combina características de la topología en estrella con la de bus. Consiste en un conjunto de subredes estrella conectadas a un bus. Esta topología facilita el crecimiento de la red. Los nodos periféricos que requieren transmitir y recibir solamente de otro nodo no necesitan de repetidores o regeneradores. Si falla un enlace que conecta con un nodo hoja, ese nodo hoja queda aislado, pero si falla un enlace con un nodo que no sea hoja, la sección entera queda aislada del resto.

Topología en Malla. En esta topología un nodo periférico puede comunicarse con otros nodos, por lo que no requiere de un nodo central. La ventaja de esta topología es que la información de un nodo puede tomar diferentes rutas, por lo tanto en un eventual fallo de un nodo la información podrá tomar otra ruta. La desventaja de esta

topología es que cada nodo se sobrecarga con la información de los nodos anteriores a él, lo cual aumenta los retardos en la comunicación.

1.2.7. Tecnologías de Comunicación

Son los mecanismos que permiten la comunicación entre dos nodos y se clasifican en dos tipos: cableadas e inalámbricas. En aplicaciones de redes AMI, existen diferentes ventajas y desventajas asociadas a estas tecnologías. Para elegir la tecnología correcta se requiere un análisis de las necesidades, evaluar la infraestructura existente, tener en cuenta factores como el impacto en equipos antiguos, la funcionalidad, requerimientos técnicos y el factor económico de la implementación [42].

En ciertas situaciones las tecnologías inalámbricas tienen ventajas sobre las tecnologías cableadas, tales como bajo costo y facilidad de conexión, pero sufren de interferencia y atenuación de la señal. Por otra parte, las tecnologías de comunicación cableadas son más fiables, menos propensas a la interferencia, pero más costosas de desplegar especialmente si se requiere de una nueva infraestructura [43]. A continuación se presenta una descripción general de las tecnologías más relevantes para implementar la infraestructura de comunicaciones de los sistemas AMI, inicialmente se presentan cuatro tecnologías cableadas y posteriormente cuatro inalámbricas.

Ethernet. Es una tecnología con muchas ventajas, incluyendo versatilidad, seguridad, velocidad y compatibilidad que la convierten en una buena opción para ser utilizada en múltiples aplicaciones en las redes AMI, como la automatización de subestaciones de distribución y sistemas de protección [44]. Cuando se utiliza sobre tecnología 100BaseTX la velocidad máxima es de 100 Mbit/s y se transmite mediante par trenzado UTP categoría 5, la desventaja que presenta es que la comunicación se puede realizar a máximo 100 metros sin la utilización de repetidores. Es ideal para uso en redes NAN [45]. En la sección 1.2.11 se presentan detalles de esta tecnología.

PLC (*Power Line communications*). Es una tecnología que utiliza la infraestructura de la red eléctrica como red de comunicaciones. Lo anterior se convierte en la mayor

ventaja de esta tecnología, pues no requiere la inversión de nuevos cables para la comunicación. Mejora la rentabilidad de las líneas rurales, hace posible trabajar en zonas remotas o a distancias más grandes. Las principales desventajas de esta tecnología son los errores de transmisión debidos a la atenuación y la interferencia electromagnética. La velocidad de transmisión al igual que el ancho de banda es muy baja. Es ideal para redes HAN y NAN [42].

RS-485. Es un estándar de comunicación serial, usado como sistema de interconexión entre dispositivos a grandes distancias diseñado para funcionar en ambientes eléctricamente ruidosos. Algunas ventajas del estándar es que soporta hasta 1000 metros de distancia entre un dispositivo maestro y los esclavos, posibilita conectar más dispositivos a la misma red, es decir, se puede conectar a los dos hilos del RS-485 una gran cantidad de dispositivos reduciendo los costos de cableado. Su desventaja es que el ancho de banda que soporta es bajo. Es ideal para redes HAN y NAN [46].

Fibra Óptica. Esta tecnología se puede utilizar para implementar redes WAN. Generalmente se usa para interconectar subestaciones con el centro de gestión. Las fibras ópticas proporcionan una capacidad de transmisión muy alta de 10 Gbps utilizando una sola longitud de onda y de 40 Gbps a 1600 Gbps utilizando la multiplexación por división de longitud de onda. Además ofrecen alto rendimiento y alta confiabilidad [6].

Wi-Fi. Se usa comúnmente como la abreviatura del estándar IEEE 802.11, esta es una tecnología de comunicación inalámbrica que soporta diferentes topologías, como punto a punto, punto-multipunto o malla, utilizando frecuencias que se encuentran entre los 800 MHz y 60 GHz. Wi-Fi tiene algunas ventajas, tales como gran ancho de banda, bajo *delay*, puede cubrir distancias más largas, es más beneficioso implementar una Red de Área Local (LAN, *Local Area Network*) inalámbrica que cableada porque son más fáciles de instalar, su costo-eficiencia y proporciona movilidad de los dispositivos.. También tiene algunas desventajas, como que es susceptible a interferencias, la topografía puede generar atenuaciones o impedir la comunicación. En AMI se utiliza particularmente en redes NAN usualmente en el rango de frecuencias de 2.4 GHz [47, 42]. En la sección 1.2.12 se presentan detalles de esta tecnología.

WiMax. Interoperabilidad Mundial para Acceso por Microondas (*WiMax, Worldwide Interoperability for Microwave Access*) es una tecnología inalámbrica que proporciona conexiones de banda ancha de alto rendimiento a largas distancias. Se puede configurar para conexiones punto a punto o punto-multipunto y proporcionar servicios de datos principalmente basados en Ethernet / IP. En redes AMI se puede utilizar para la comunicación entre subestaciones o *microgrids* y como *backbone*. Es ideal para redes WAN [6].

ZigBee. Está diseñado para aplicaciones de bajo costo y bajo consumo, ofrece una red inalámbrica fiable, que permite la creación de redes utilizando múltiples topologías, como estrella, árbol y malla. ZigBee se basa en el estándar IEEE 802.15.4 y se utiliza para tasas de transferencia bajas ideales para aplicaciones de utilidad dentro de una HAN [48].

Red Celular. A pesar de que las redes celulares fueron diseñadas con otro objetivo como es la comunicación de voz y enlace de datos, ha encontrado aplicación en las redes AMI, especialmente en sistemas SCADA para subestaciones y monitoreo de dispositivos remotos gracias a la gran cobertura que ofrece. La ventaja de esta tecnología es que ya cuenta con la infraestructura por lo que no requiere de instalaciones adicionales. Una limitación importante de la tecnología celular es que la comunicación de datos no es priorizada en caso de eventos especiales y falta de cobertura en algunas regiones. Es ideal para aplicaciones WAN [9, 49].

1.2.8. Protocolos de Comunicación

Un protocolo de comunicación es un conjunto de reglas que establecen la semántica y la sintaxis que debe emplearse para la comunicación de los elementos que forman parte de una red. Los protocolos se dividen en propietarios y abiertos, a continuación se describe cada uno de ellos [50].

Protocolos propietarios: son denominados propietarios o cerrados porque son diseñados exclusivamente para operar con determinadas marcas de equipos y sistemas de gestión, es el caso de protocolos propietarios de las marcas como: SIEMENS, ABB, GE, Allen Bradley, entre otras. La desventaja de los protocolos

propietarios es que se obliga a los usuarios a utilizar una misma marca en los diferentes equipos y sistema de gestión, imposibilitando la integración con equipos de otras marcas.

Protocolos Abiertos: son denominados abiertos o libres, debido a que están diseñados para operar indistintamente de la marca de los equipos a integrar. Algunos ejemplos de protocolos abiertos son: Modbus, DNP3, OPC, ICCP, IEC 61850, entre otros. Existen diversas variaciones de un mismo protocolo abierto que los fabricantes realizan para sus productos, como es el caso del protocolo IEC 60870 que es la versión europea de DNP3, pero es el cumplimiento de los estándares lo que hace que los protocolos abiertos sean muy utilizados en la actualidad. Dos de los protocolos abiertos más empleados son Modbus y DNP3, que se utilizan en el desarrollo de este trabajo y se presentan en detalle en las siguientes secciones.

1.2.9. Modbus

Modbus es un protocolo de comunicaciones que fue diseñado en 1979 por Modicon y que continúa en vigencia por su gran versatilidad y su carácter abierto, lo que ha permitido que esté ampliamente extendido entre múltiples fabricantes de dispositivos. Este protocolo proporciona una comunicación tipo cliente/servidor que ofrece servicios especificados por códigos de función [51]. Por su parte, la configuración cliente/servidor (maestro / esclavo) indica que el cliente (maestro) envía un mensaje de solicitud (solicitud de servicio) al servidor (esclavo), y el servidor responde con un mensaje de respuesta. Si el servidor no puede procesar una solicitud, en su lugar devolverá un código de función de error (respuesta de excepción) que es el código de función original más 80H (es decir, con el bit más significativo establecido en 1) [52].

Las funciones Modbus operan en los registros de memoria del dispositivo para configurar, monitorear y controlar las Entradas/Salidas del dispositivo. El modelo de datos Modbus tiene una estructura simple que solo diferencia entre cuatro tipos de datos básicos:

- Entradas discretas
- Bobinas (Salidas)

- Registros de entrada (datos de entrada)
- Registros de retención (datos de salida)

En la tabla 1.1 se describen las principales funciones Modbus que permiten acceder a los registros descritos en el mapa de registro del dispositivo para enviar y recibir datos [58].

Tabla 1.1. Funciones básicas y códigos de operación Modbus [53].

Código	Función	Descripción
01H	*Read Coil*	Leer estado de salida discreta
02H	*Read Discrete Inputs*	Leer estado de entradas discretas
03H	*Read Holding Registers*	Leer registros de retención
05H	*Write Single Coil*	Permite modificar el valor de una sola salida discreta
06H	*Write Register*	Escribe un valor en un registro
0FH	*Write Multiple Coils*	Permite modificar el valor de múltiples salidas discretas
10H	*Write Multiple registers*	Escribe múltiples registros

Descripción del protocolo. Modbus está ubicado en la capa 7 del modelo OSI, su arquitectura se presenta en la figura 1.3. Además, Modbus se puede implementar de tres formas, las cuales se describen a continuación [54].

- TCP: Utilizando Ethernet como enlace de datos y acceso al medio.
- Transmisión serie asíncrona sobre diversos medios: RS232/422/485.
- Modbus: Red de alta velocidad con paso de testigo. Utiliza HDLC como protocolo de enlace de datos.

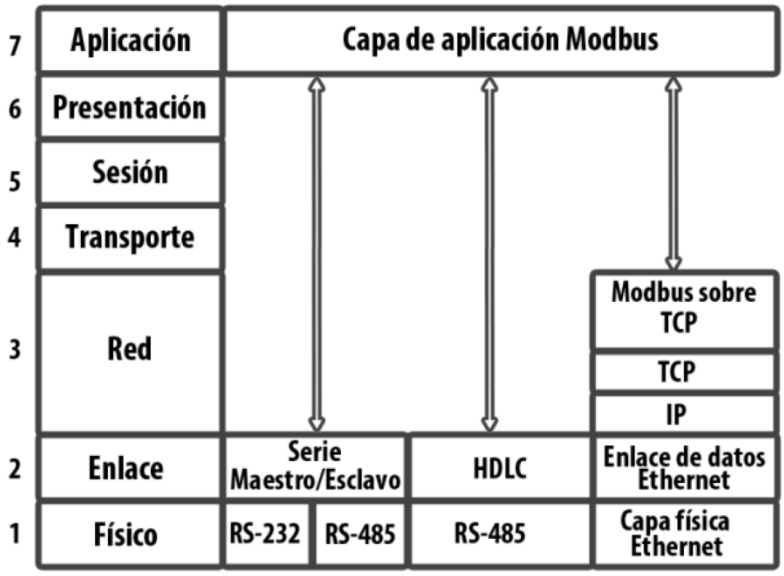


Figura 1.3. Arquitectura de Modbus según el modelo OSI [54].

El protocolo Modbus define una Unidad de Datos de Protocolo (PDU, *Protocol Data Unit*) independiente de las capas de comunicación subyacentes, el mapeo del protocolo en buses o redes específicas puede introducir algunos campos adicionales en la Unidad de Datos de la Aplicación (ADU, *Application Data Unit*). En la figura 1.4 se observa la trama general del protocolo Modbus.

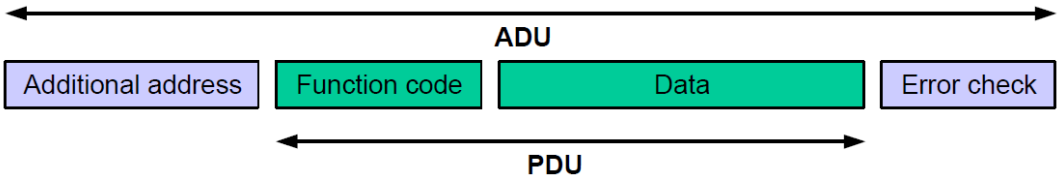


Figura 1.4. Trama General Modbus [51].

La Unidad de Datos de Protocolo (PDU) es la unidad de datos básica en el protocolo Modbus y consta de dos campos [54]:

- Código de Función: codifica el tipo de acción a realizar por parte del servidor. Ocupa un Byte.
- Campo de Datos: es el mensaje. Si ha sido enviado por el cliente hacia el servidor, contendrá información que el servidor necesita para ejecutar la acción indicada por el código de función. Si el mensaje ha sido enviado por el servidor puede contener los datos solicitados por el cliente o un código de

error, que indicará que la acción solicitada no se ha podido llevar a cabo y la causa.

La unidad de datos de aplicación ADU es construida por el cliente que inicia una transacción. Por su parte, el campo código de función de la unidad de datos PDU está codificado en un byte e indica al servidor qué tipo de acción debe realizar. Además, el campo de datos de los mensajes enviados desde el cliente al servidor puede contener información adicional que el servidor usa para tomar la acción definida por el código de la función. Esto puede incluir elementos como direcciones discretas y de registro, la cantidad de elementos a manipular y el conteo de bytes de datos actuales en el campo. En la figura 1.5 se presenta un ejemplo de trama Modbus, en donde se muestra que la trama está dirigida al servidor de dirección 10, el código de función indica que se leerán registros de retención, a partir de dirección cero, el número de registros es 10, por lo tanto se leerán las direcciones cero a nueve y finalmente se presenta el código de verificación de errores CRC [58].

Dirección servidor	Código de función	Dirección de inicio	N° de registros	CRC
10	03	00 00	00 0A	----

Figura 1.5. Trama Modbus [52].

Si no se produce ningún error y la función solicitada en una ADU es recibida correctamente, los datos solicitados se encuentran en el campo de datos de la respuesta del servidor. Por el contrario, si se produce un error relacionado con la función solicitada, el campo contiene un código de excepción que la aplicación del servidor puede usar para determinar la siguiente acción a realizar. Cuando el servidor responde al cliente, utiliza el campo de código de función para indicar una respuesta normal (sin errores) o que se produjo algún tipo de error (llamada respuesta de excepción). Para una respuesta normal, el servidor simplemente hace eco a la solicitud del código de función original. En la figura 1.6 se presenta una transaccion sin errores y con errores.

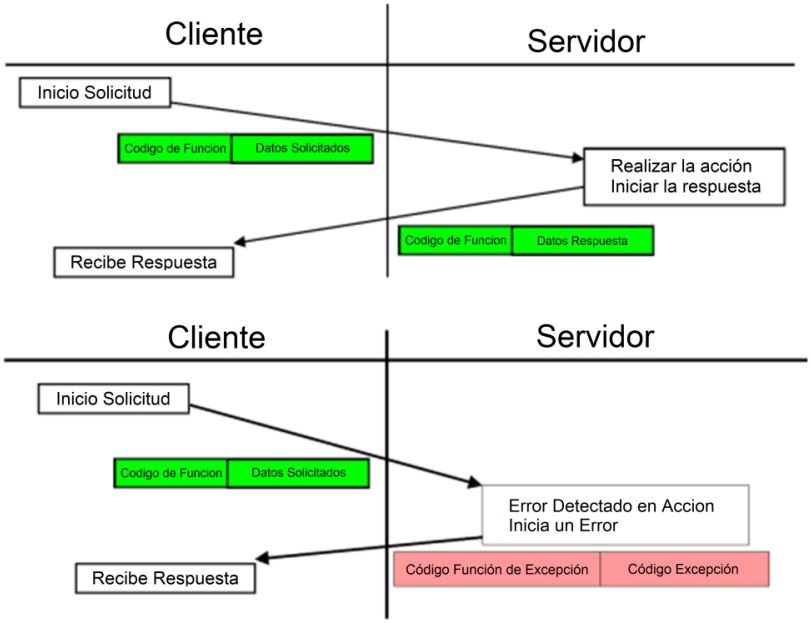


Figura 1.6. Transacción Modbus sin errores y con errores [51].

Por otro lado, es importante destacar que el tamaño de la PDU está limitado por la restricción de tamaño heredada de la primera implementación de Modbus serial que estipula lo siguiente:

- PDU Modbus (comunicación serial) = 256 - dirección del servidor (1 byte) - CRC (2 bytes) = 253 bytes.

Por consiguiente el tamaño de la ADU de las implementaciones actuales es la siguiente:

- ADU RS232 / RS485 = 253 bytes + dirección del servidor (1 byte) + CRC (2 bytes) = 256 bytes.
- ADU Modbus/TCP = 253 bytes + MBAP (7 bytes) = 260 bytes.

Modbus/TCP. Modbus sobre TCP es una variante del protocolo que se incluyó para hacerlo compatible con redes Ethernet y al igual que Modbus serial provee una comunicación Cliente/Servidor entre dispositivos conectados a la red Ethernet TCP utilizando el puerto 502. El modelo Cliente/Servidor se muestra en la figura 1.7 y está basado en cuatro tipos de mensajes que se describen a continuación [53]:

- Solicitud Modbus (*Request*): mensaje enviado por el cliente para iniciar una transacción.
- Indicación Modbus (*Indication*): mensaje de solicitud recibido en el lado del servidor.
- Respuesta Modbus (*Response*): mensaje de respuesta enviado por el servidor.
- Confirmación Modbus (*Confirmation*): mensaje de respuesta recibido en el lado del cliente.

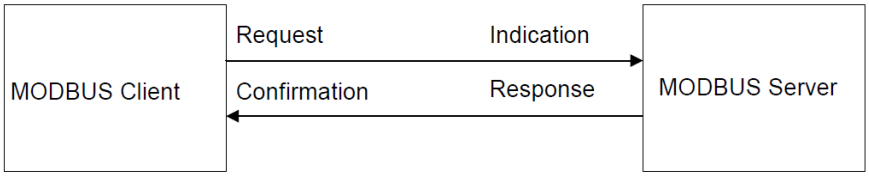


Figura 1.7. Modelo Modbus/TCP [53].

Por su parte, en la Figura 1.8 se describe el encapsulado de una petición o respuesta, cuando se realiza por medio de Modbus/TCP.

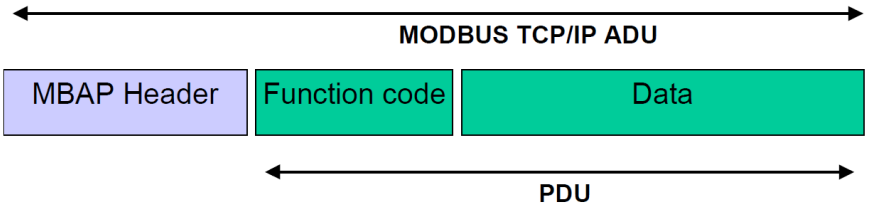


Figura 1.8. Solicitud/respuesta de Modbus sobre TCP [53].

En Modbus/TCP se utiliza un encabezado para identificar a la unidad de datos de aplicación ADU el cual se conoce como MBAP (Modbus *Application Protocol header*). El MBAP presenta algunas diferencias en comparación con la ADU utilizada en Modbus serial, a continuación se describen sus campos:

- El campo '*slave address*' usado en Modbus serial es reemplazado por un único byte '*Unit Identifier*' dentro de la cabecera MBAP. El '*Unit Identifier*' se utiliza para comunicarse a través de dispositivos como puentes, enrutadores y puertas de enlace que utilizan una sola dirección IP para soportar múltiples unidades Modbus independientes.

- Todas las solicitudes y respuestas están diseñadas de tal manera que el destinatario puede verificar que un mensaje ha finalizado. Para los códigos de función, donde el PDU tiene una longitud fija, el código de función es suficiente. Para los códigos de función que llevan una cantidad variable de datos en la solicitud o respuesta, el campo de datos incluye un conteo de bytes.
- Cuando Modbus se transporta sobre TCP, se adiciona información sobre la longitud en el encabezado MBAP, que permite al receptor reconocer los límites del mensaje, incluso si el mensaje se ha dividido en varios paquetes para su transmisión. La existencia de normas de longitud explícita e implícita, y el uso de un código de comprobación de errores CRC-32 (en Ethernet) resulta en una probabilidad infinitesimal de la no detección de errores de un mensaje de petición o respuesta.

En la figura 1.9 se ilustra la construcción de un paquete TCP para la transmisión de datos Modbus. En este caso, la capa de aplicación es Modbus y la Unidad de Datos de Aplicación (ADU) está integrada en la matriz de datos TCP. Cuando una aplicación envía sus datos a través de la red, estos se transmiten desde la capa superior hasta la capa más baja, en donde cada una de ellas tiene una función designada y adiciona su propio encabezado a cada paquete. La capa más baja es la realmente responsable de enviar los datos a través del medio físico, luego el paquete viaja a través de las diferentes capas del receptor, decodificando su porción del mensaje y eliminando el encabezado. Finalmente, el paquete llega a la aplicación de destino.

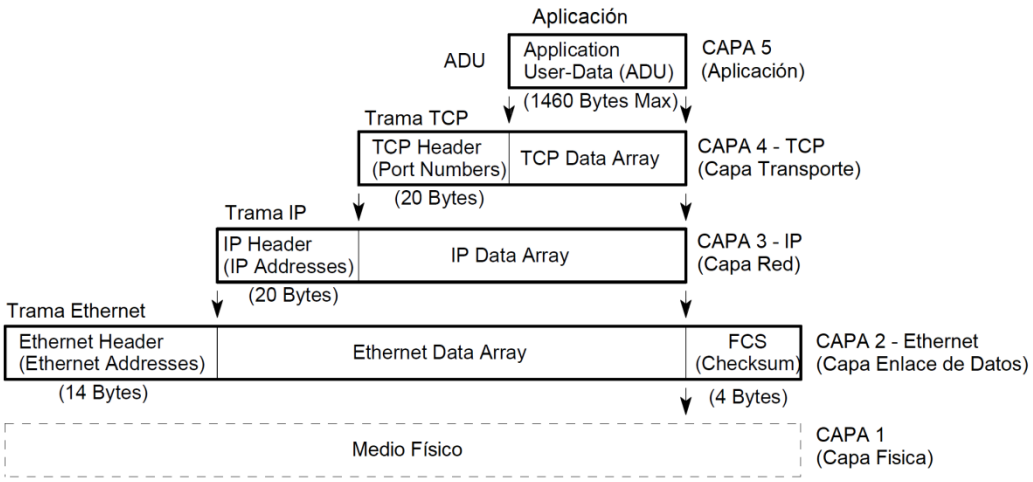


Figura 1.9. Encapsulación de Modbus sobre TCP [52].

1.2.10. DNP3

DNP3 es un protocolo para la transmisión de datos mediante comunicaciones serie y TCP. Fue diseñado para optimizar la transmisión de información en la adquisición de datos y comandos de control de un dispositivo a otro y está destinado a aplicaciones SCADA [55]. En DNP3 se utiliza el término *outstation* para denotar a los dispositivos remotos que se encuentran en campo y maestro para las computadoras en los centros de control. DNP3 es un protocolo no propietario, esto significa que la estación maestra puede comunicarse con equipos *outstation* de cualquier fabricante de manera confiable, permitiendo el envío de datos y comandos de control.

En DNP3, el maestro se encarga de revisar periódicamente a los o*utstation* utilizando solicitudes, los mensajes de solicitud se ejecutan en el dispositivo o*utstation* y envía el mensaje de respuesta al maestro. Además, los o*utstation* pueden enviar mensajes no solicitados que contienen información relevante.

Descripción del protocolo. El protocolo fue desarrollado siguiendo el modelo de Arquitectura de Rendimiento Mejorado (EPA, *Enhanced Performance Architecture*), que tiene un número reducido de capas. En la figura 1.10 se presenta una comparación del modelo OSI y el modelo EPA que opera con sólo tres de las siete capas del modelo OSI, siendo ellas la capa física, enlace y aplicación [22].

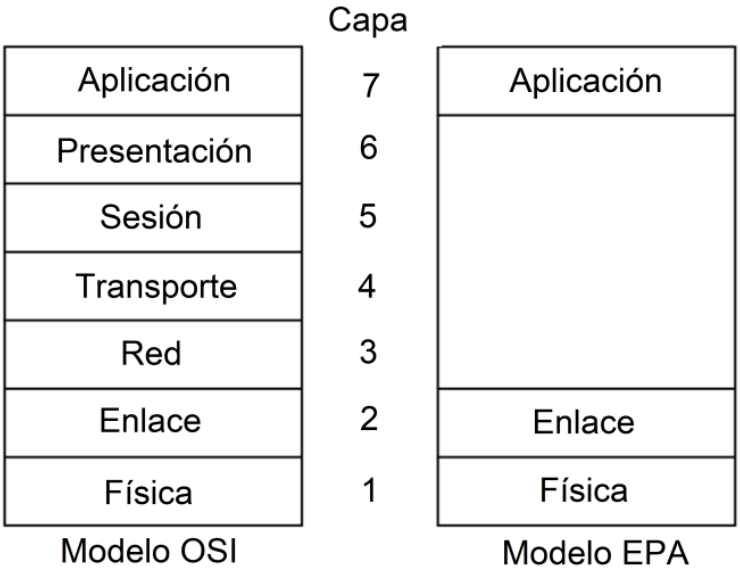


Figura 1.10. Comparación entre el modelo de referencia OSI y el modelo EPA [55].

Además, DNP3 agrega una funcionalidad de transporte, que se denomina capa de pseudo-transporte, en la Figura 1.11 se ilustra la pila del protocolo DNP3 y la comunicación entre el maestro y el *outstation*.

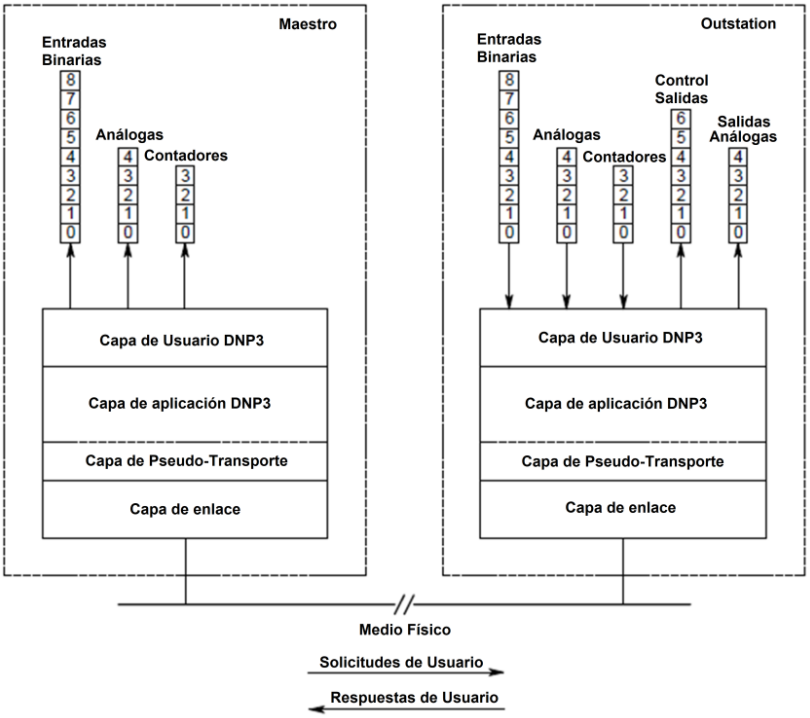


Figura 1.11. Pila del protocolo DNP3 [22].

A continuación se describe cada una de las capas [22]:

- **Capa de usuario:** la estación maestro interactúa con la base de datos y solicita la información a la *outstation*. En la *outstation*, el software extrae la información y la envía a la estación maestro.
- **Capa de aplicación:** la estación maestro organiza y envía mensajes a la *outstation* para solicitar información, realizar una función especial o ejecutar un comando. La *outstation* genera el mensaje apropiado según el requerimiento y envía el mensaje a la estación maestro. El tamaño del fragmento (paquete) depende del tamaño del búfer del dispositivo y puede estar en un intervalo entre 2048 y 4096 bytes.
- **Capa de pseudo-transporte:** no es una capa, es una función incorporada en la capa de aplicación que se limita a dividir un mensaje de la capa de aplicación en paquetes más pequeños para ser enviados a la capa de enlace.

En la recepción, la función de esta capa es montar varios segmentos en un fragmento y notificar a la capa de aplicación que hay un fragmento recibido y disponible.

- **Capa de enlace:** es la capa responsable de asegurar que la transmisión de datos por la capa física sea confiable realizando detección de errores CRC. El CRC es un esquema ampliamente utilizado en la verificación de la integridad de los datos. En el protocolo DNP3 ocupa un espacio de dos octetos (2 bytes = 16 bits) que se adjuntan después de cada trama (16 octetos) de datos enviada, incluidos los campos de encabezado. La comprobación de redundancia cíclica se genera a partir de la expresión 2.1.

$$x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1 \quad (2.1)$$

Considerando que el polinomio forma un número binario de 17 bits y que cada término presente en el polinomio determina la existencia de un 1 y que cada ausencia corresponde a cero, ese número es invertido bit a bit y anexado a cada trama de hasta 16 octetos, inclusive a los 8 octetos del encabezado. La ejecución de la transmisión de datos se realiza según el siguiente procedimiento:

- Selecciona el bloque B que contiene n bits.
- Multiplica B por 2^{16} para que se obtenga $(2^{16} * B)$ de 17 bits.
- Divide $(2^{16} * B)$ por el polinomio P, que también es de 17 bits, en complemento a dos para obtener el resto R que es de 16 bits.
- Invertir R bit-a-bit obteniendo R' y adiciona $(2^{16} * B)$.
- Transmite $T' = (2^{16} * B) + R'$.

En la recepción de datos el procedimiento es el siguiente:

- Recibe el cuadro $T' = (2^{16} * B) + R' \rightarrow$ (N + 16 bits).
- Invierte R '(16 bits) en T' (N + 16 bits) para obtener T (N + 16 bits).
- Divide T (complemento a dos) por P (17 bits) para obtener el resto.
- Si el resto es cero entonces la trama se transmitió con éxito, de lo contrario se produjo un error en la transmisión.

- **Capa física:** son los medios físicos a través de los cuales se realiza la transferencia de datos. Esta capa abarca funciones para controlar los medios de comunicación, estableciendo y manteniendo una conexión física y controlando el flujo de datos [29].

Una de las funcionalidades que diferencian a DNP3 de Modbus es la capacidad de enviar mensajes no solicitados desde las *outstation*. Esta funcionalidad permite enviar datos críticos de operación, cambio de estado del dispositivo y el sobrepaso de valores límites establecidos, sin la necesidad de esperar una solicitud de lectura por parte de la estación maestro. La estación maestro responde confirmando la recepción de este tipo de mensaje, la estación remota espera por esa confirmación para evitar una retransmisión o para eliminar de la memoria la información ya enviada a la estación maestro; sólo después de la confirmación por parte de la estación maestro, la *outstation* elimina dicha información. En la figura 1.12 se presenta una transacción DNP3 para una función de lectura y para el envío de un mensaje no solicitado.

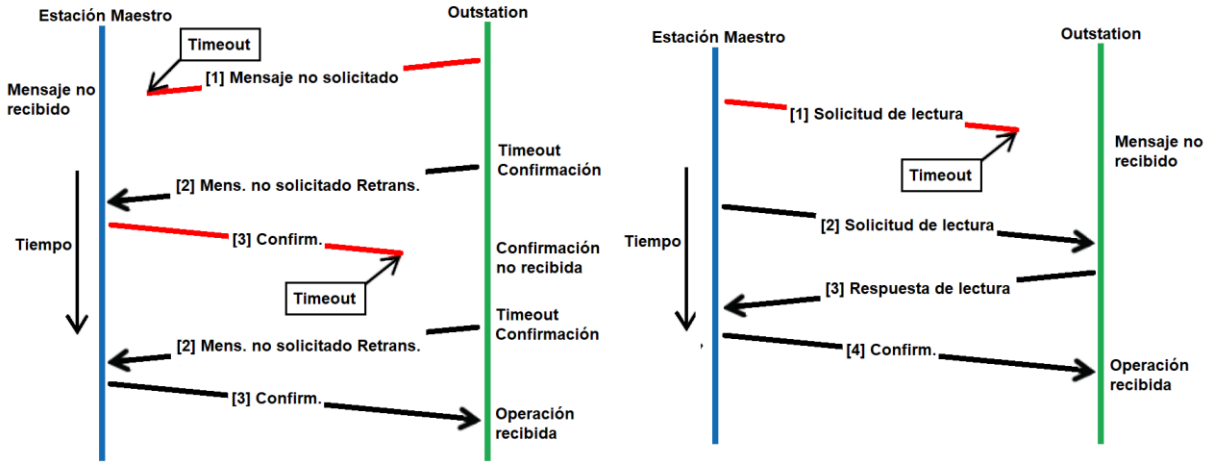


Figura 1.12. Transacción DNP3 para mensaje no solicitado y función de lectura [38].

Por otra parte, una trama DNP3 se compone de un encabezado de longitud fija seguido de bloques de datos opcionales. En la figura 1.13 se muestra una trama DNP3.

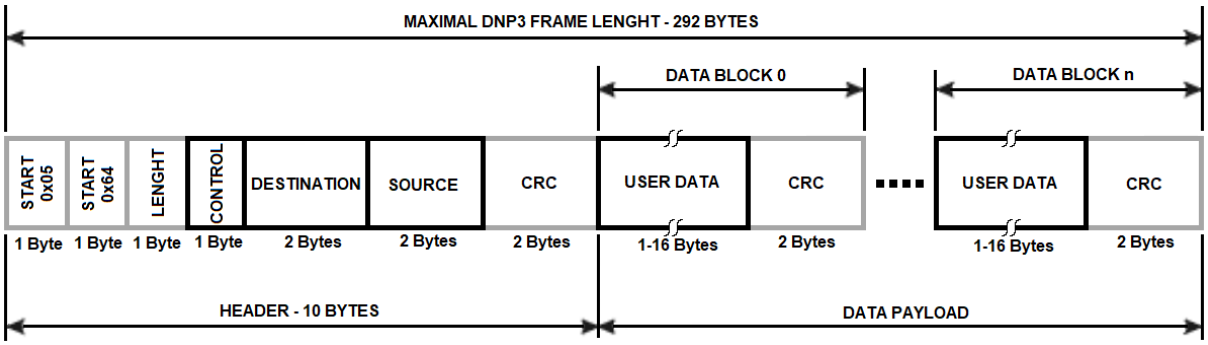


Figura 1.13. Trama DNP3 [56].

A continuación se describen cada uno de los bloques [56]:

Encabezado DNP3:

Inicio: el campo de inicio tiene 2 bytes de longitud que corresponden al 05H y 64H respectivamente.

Longitud: este campo consta de 1 byte y especifica la cantidad de bytes enviados en la trama.

Control: este campo contiene la dirección de la trama, tipo de trama e información de control de flujo.

Destino: este campo tiene un tamaño de 2 bytes y especifica la dirección de la estación a la que se dirige la trama.

Fuente: este campo tiene un tamaño de 2 bytes y especifica la dirección de la estación desde la cual se originó la trama.

CRC: se agrega una verificación de redundancia cíclica de dos bytes a cada bloque de la trama.

Datos de Usuario: estos bloques pueden contener de 1 a 16 bytes de datos. Si se desea transmitir más de 16 bytes, cada bloque debe contener 16 bytes excepto el último bloque, que contendrá los bytes restantes. Cada bloque de datos tiene un CRC anexado a él.

DNP3/TCP. El protocolo DNP3 fue desarrollado originalmente para establecer una conexión serial punto a punto, pero con el paso de los años se expandió a redes de mayor tamaño y capacidad, es así como surgió la implementación del protocolo sobre TCP [21]. En la figura 1.14 se presenta la implementación de DNP3 sobre TCP. En este modo las capas anteriores no cambian y el envío de mensajes es transparente e independiente del protocolo TCP. Las modificaciones que existen son, el método de sincronización de tiempo y que no se utilizan confirmaciones en la capa de enlace, sólo la confirmación en la capa de aplicación de forma similar a lo que ocurre en la transmisión serial [22]. Por su parte, la comunicación con un dispositivo vía DNP3/TCP se realiza generalmente por el puerto 20000.

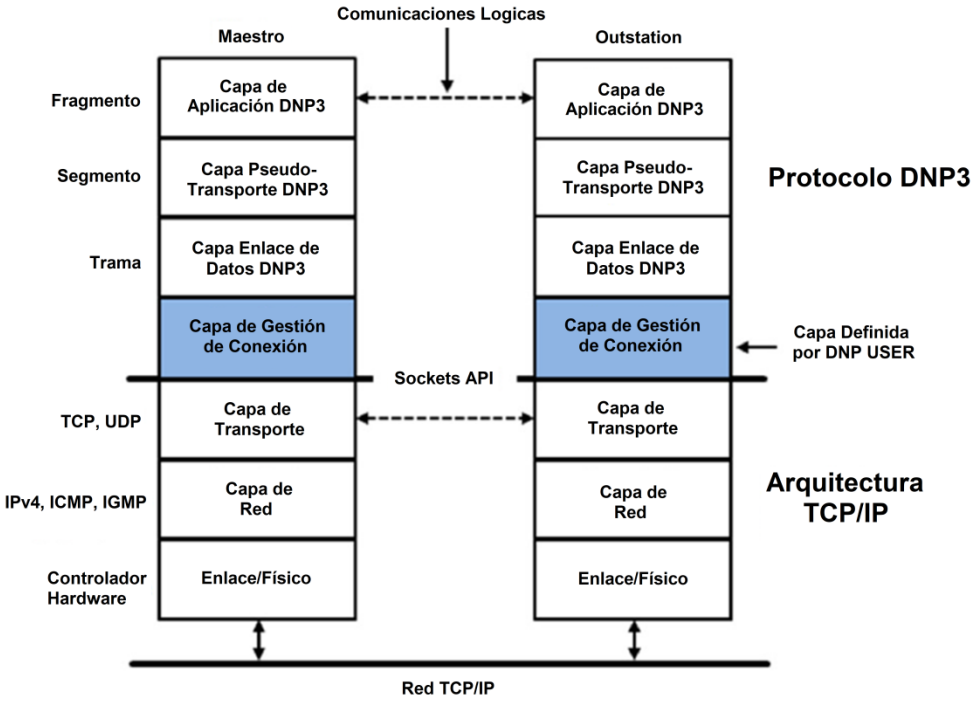


Figura 1.14. DNP3 sobre TCP [22].

En la capa de gestión de conexión se encuentra la interfaz entre las capas de los protocolos DNP3 y TCP/IP, esta interfaz se encarga de establecer y cerrar conexiones TCP, transmitir, aceptar datagramas y enviar las tramas de la capa de enlace del protocolo DNP3. Por otro lado, existen tres niveles de implementación de DNP3, el nivel 1 se compone de funciones básicas del protocolo, los demás niveles son opcionales y están orientados a establecer comunicación con los IED. El nivel 2

permite más funciones, grupos y variaciones y los IED y RTU son más sofisticados. El nivel 3 soporta todas las funcionalidades del protocolo.

Por otro lado, el grupo DNP USER encargado de promover el uso de DNP3, ha realizado algunas recomendaciones para la implementación del protocolo sobre TCP, las más importantes se presentan a continuación [22]:

- Las confirmaciones de la capa de enlace de datos DNP3 deben ser deshabilitadas, porque TCP se encarga de garantizar una conexión confiable.
- La capa física recomendada es Ethernet.
- Todos los equipos deben soportar TCP y UDP.

Por su parte, en la figura 1.15 se presenta la encapsulación de DNP3 sobre TCP/IP. En este caso el modelo de capas EPA no cambia, por lo tanto las capas de aplicación, pseudo-transporte y enlace de datos del protocolo DNP3 se mantienen y actúa sobre la capa de transporte, capa de red y capa física de la arquitectura TCP/IP. La capa de pseudo-transporte y de enlace de datos son los elementos y servicios esenciales que permiten el direccionamiento y detección de errores necesarios para trabajar junto con la arquitectura TCP/IP.

En la capa de enlace DNP3 las confirmaciones no son necesarias, ya que TCP es un protocolo de transporte orientado a la conexión que establece una conexión a través de confirmaciones ACK (*Acknowledgment*) con lo que garantiza una comunicación confiable de principio a fin.

La construcción de un mensaje DNP3 inicia en la capa de aplicación y termina en la capa de enlace de datos DNP3. La trama de la capa de enlace tiene un tamaño máximo de 292 bytes y es encapsulada en un segmento TCP en la capa de transporte TCP/IP, en la misma se añade un encabezado con 20 bytes y su tamaño máximo es de 556.

A continuación, el mensaje se encapsula en un datagrama IP en la capa de Internet, en esta capa se añaden otros 20 bytes de encabezado, formando un mensaje de tamaño máximo de 576 bytes. En la capa de interfaz de red TCP/IP, el mensaje se vuelve a encapsular en una trama Ethernet, que contiene de 46 a 1.500 bytes de

datos y se agrega un encabezado de 18 bytes, para un tamaño máximo en una trama Ethernet de 1.518 bytes.

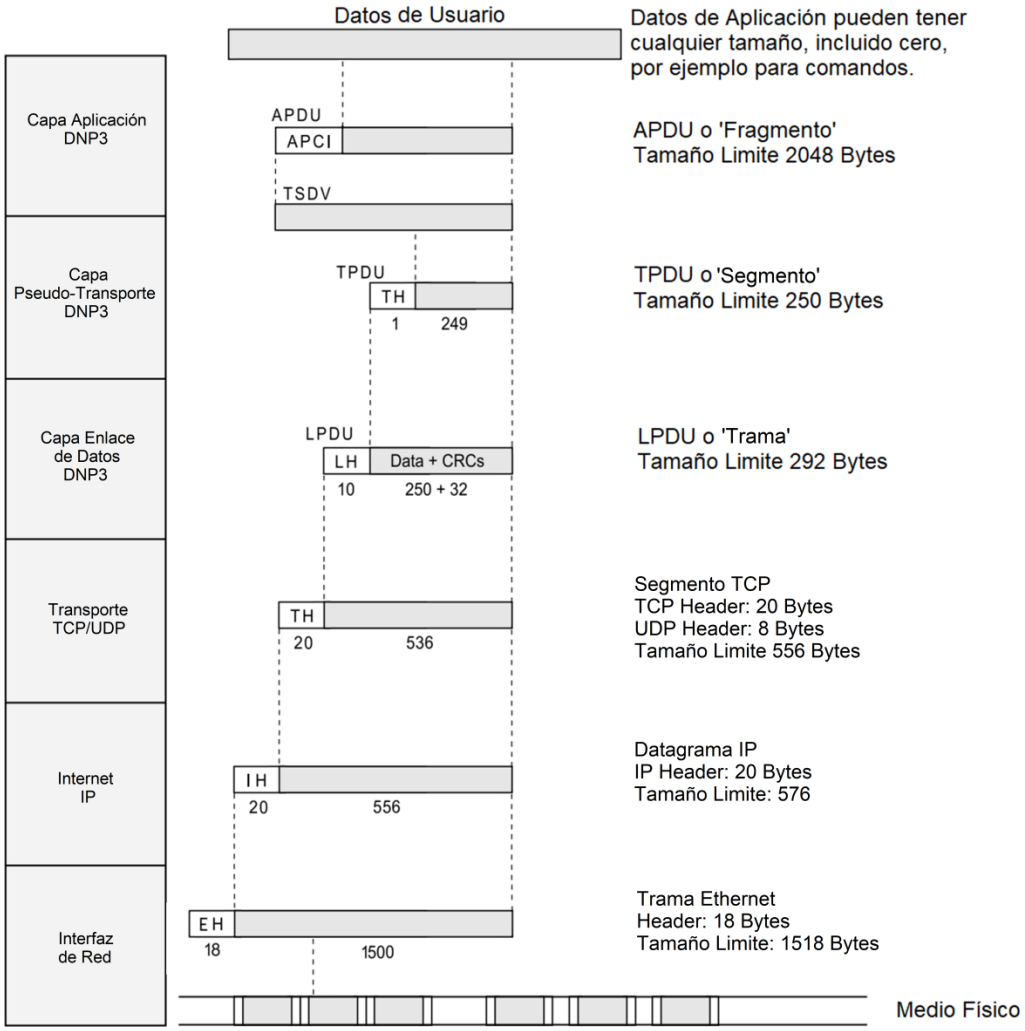


Figura 1.15. Encapsulación de DNP3 sobre TCP/IP [57].

1.2.11. IEEE 802.3/Ethernet

IEEE 802.3 es actualmente la tecnología LAN dominante en el mundo, soporta topologías en bus o estrella y soporta velocidades de transferencia de datos de hasta 1 Gbps. Su estructura se puede entender mejor utilizando el modelo de referencia OSI [58]. En términos del modelo OSI, IEEE 802.3 es mapeado en las capas inferiores, específicamente en la capa 1 (capa física) y capa 2 (capa de enlace de datos). Los protocolos de capas superiores son independientes de la arquitectura de

red y pueden implementarse en cualquier tipo de red. El mapeo del protocolo IEEE 802.3 se muestra en la figura 1.16 y muestra la relación con la arquitectura del modelo OSI [59].

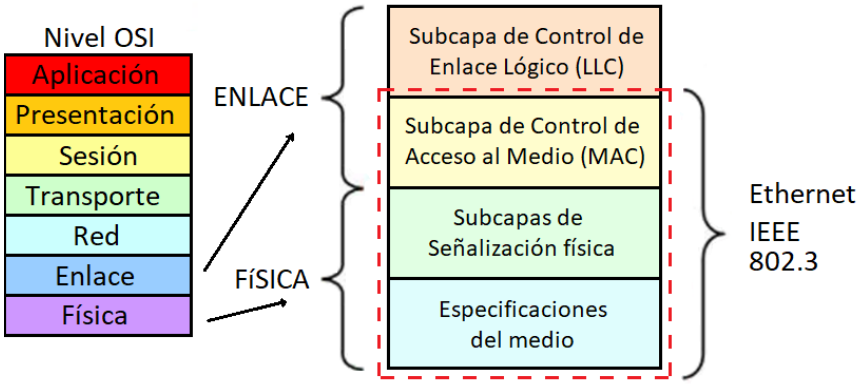


Figura 1.16. 802.3 ubicado en la capa 1 y 2 del modelo OSI [59].

Los datos generados en la capa de aplicación pasan a la capa de transporte, en donde se dividen en segmentos, porciones de datos aptas para su transporte en la red. Luego, los datos van bajando por la pila de cada capa hasta llegar a la capa física. Conforme los datos van bajando, cada protocolo va añadiendo una serie de cabeceras y datos adicionales, necesarios para poder ser enviados correctamente a su destino. El resultado final son unidades de información denominadas tramas que son las que viajan de un host a otro. En la figura 1.17 se presenta la estructura de una trama Ethernet.

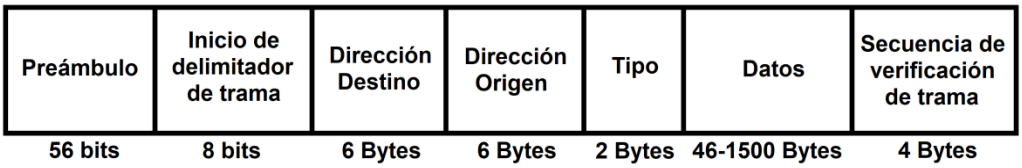


Figura 1.17. Trama 802.3/Ethernet [59].

Cada trama está dividida en 7 campos que se describen a continuación:

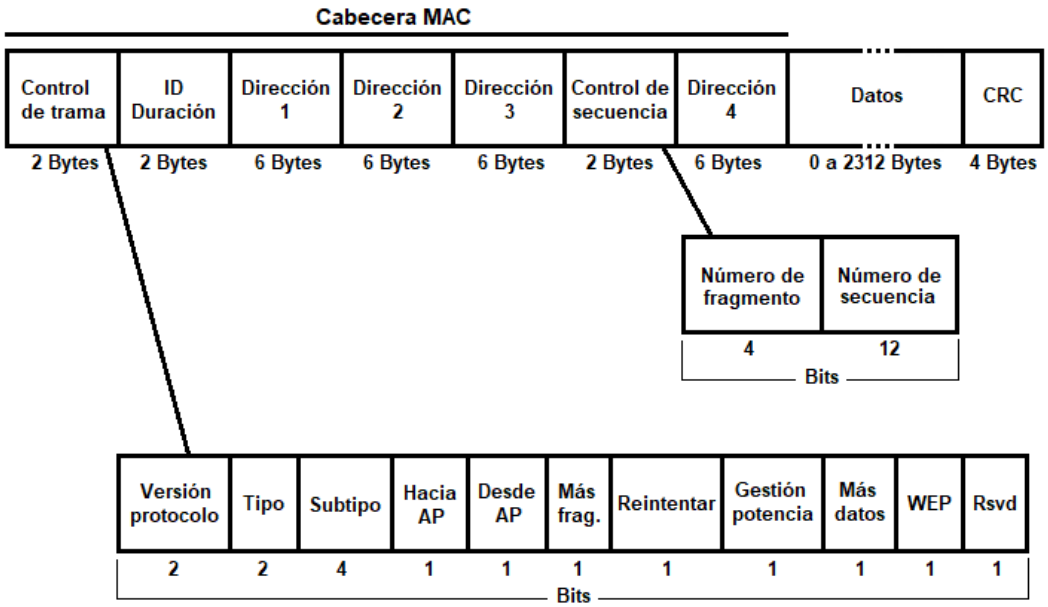
- Preámbulo: campo de 7 bytes de longitud con una secuencia de bits utilizada para sincronizar y estabilizar el medio físico antes de iniciar la transmisión.

- SFD (*Start Frame Delimiter*): indicador de inicio de trama. Campo de 1 byte de longitud que contiene la secuencia 10101011, indica el inicio de una trama de datos.
- Dirección de destino: campo de 6 bytes de longitud que contiene la dirección MAC a la que se envía la trama.
- Dirección de origen: campo de 6 bytes de longitud que contiene la dirección MAC del dispositivo que envía la trama.
- Tipo (Ethernet) o Longitud (IEEE 802.3): campo de 2 bytes de longitud. Este campo es el que distingue a las tramas 802.3 de las tramas Ethernet. Valores iguales o mayores de x0600 indican que es una trama Ethernet y el valor especifica el protocolo de capa superior que recibe los datos, por ejemplo x0800 representa el protocolo IP. Valores iguales o menores de x05DC (1500 en decimal) indican que es una trama 802.3 y el valor representa la cantidad de bytes de datos que siguen a este campo.
- Datos: campo de 46 a 1500 bytes de longitud. Contiene los datos a transferir entre origen y destino. Si este campo fuera menor de 46 bytes se añade un campo de 'relleno' para mantener el tamaño mínimo de paquete.
- FCS (*Frame Check Sequence*): secuencia de verificación de trama, campo de 4 bytes de longitud que contiene un valor para control de errores, CRC. La Verificación de Redundancia Cíclica (CRC, *Cyclic Redundancy Check*), consiste en un valor calculado por el emisor que resume todos los datos de la trama. El receptor calcula nuevamente el valor y, si coincide con el de la trama, entiende que la trama se ha transmitido sin errores. El campo FCS es generado sobre los campos dirección de destino, la dirección de origen, el tipo/longitud y datos.

1.2.12. IEEE 802.11/Wi-Fi

IEEE 802.11 es un estándar que define las características de una Red de Área Local Inalámbrica (WLAN). La palabra Wi-Fi (*Wireless Fidelity*, Fidelidad inalámbrica) es el nombre de la certificación otorgada por la Wi-Fi Alliance, grupo que garantiza la compatibilidad entre dispositivos que utilizan el estándar 802.11. Por el uso indebido de los términos el nombre del estándar se confunde con el nombre de la certificación. Una red Wi-Fi es en realidad una red que cumple con el estándar 802.11. La norma IEEE 802.11 fue diseñada para sustituir las capas físicas y MAC de la norma 802.3

(Ethernet), por lo tanto la única diferencia entre estas dos normas es en cómo se transmiten las tramas de datos, en la figura 1.18 se presenta una trama 802.11 [60].



. Figura 1.18. Trama IEEE 802.11 [60].

Cada trama está dividida en 9 campos que se describen a continuación:

- Control de trama: este campo se compone de 16 bits ordenados de la siguiente manera:
    - Versión de protocolo: la versión de la trama 802.11 en uso.
    - Tipo y Subtipo: identifican una de las tres funciones y subfunciones de la trama (control, datos y administración).
    - Hacia AP: se establece en 1 para las tramas de datos destinadas al sistema de distribución.
    - Desde AP: se establece en 1 para las tramas de datos que salen del sistema de distribución.
    - Más fragmentos: se establece en 1 para las tramas que tienen otro fragmento.
    - Reintentar: se establece en 1 si la trama es una retransmisión de una trama anterior.
    - Gestión de potencia: se establece en 1 para indicar que un nodo estará en el modo de ahorro de energía.

  - Más datos: se establece en 1 para indicarle a un nodo en el modo de ahorro de energía que se almacenan más tramas en *búfer* para ese nodo.
  - WEP: se establece en 1 si la trama contiene información encriptada mediante WEP para propósitos de seguridad
  - Rsvd: se establece en 1 en una trama de tipo de datos que utiliza la clase de servicio estrictamente ordenada (no requiere reordenamiento).
- ID Duración: según el tipo de trama, representa el tiempo que se requiere en microsegundos para transmitir la trama o una identidad de asociación para la estación que transmitió la trama.
- Dirección 1: contiene la dirección MAC del nodo de destino final en la red.
- Dirección 2: contiene la dirección MAC del nodo que inició la trama.
- Dirección 3: contiene la dirección MAC que identifica al dispositivo inalámbrico que es el destinatario inmediato de la trama.
- Control de secuencia: consta de 16 bits, divididos en los siguientes campos:
  - Número de fragmento: indica el número de cada fragmento de la trama.
  - Número de secuencia: indica el número de secuencia asignado a la trama.
- Dirección 4: contiene la dirección MAC que identifica al dispositivo inalámbrico que transmitió la trama.
- Datos: contiene la información que se transporta.
- CRC: contiene una comprobación de redundancia cíclica (CRC) de 32 bits de la trama.

En 1997 se creó el primer estándar 802.11 con un ancho de banda máximo de 2 Mbps en la banda de frecuencia de 2.4 GHz. Con el transcurso de los años las aplicaciones requirieron mayor ancho de banda, por lo que surgió una familia entera a partir de esta norma inicial. A continuación se presentan las normas más importantes y sus características [61].

802.11b. El estándar 802.11 original se amplió en julio de 1999, creando la especificación 802.11b que soporta una velocidad teórica de hasta 11 Mbps a la misma frecuencia (2.4 GHz) que el estándar 802.11 original.

802.11a. Mientras 802.11b estaba en desarrollo, se creó una segunda extensión del estándar 802.11 original llamado 802.11a. Por su mayor costo 802.11a se encuentra generalmente en redes empresariales, admite un ancho de banda de hasta 54 Mbps y trabaja en la frecuencia de 5 GHz, que en comparación con 802.11b acorta el rango de alcance, además, tiene mayor dificultad para penetrar paredes y otras obstrucciones. Debido a que 802.11a y 802.11b utilizan frecuencias diferentes, las dos tecnologías son incompatibles entre sí.

802.11g. En 2002 y 2003 surgió el estándar 802.11g, que combina lo mejor de 802.11a y 802.11b. Esta norma admite un ancho de banda de hasta 54 Mbps en la frecuencia de 2.4 GHz por lo que tiene un mayor rango de cobertura. 802.11g es compatible con 802.11b, lo que significa que los puntos de acceso 802.11g funcionan con los adaptadores de red inalámbricos 802.11b y viceversa.

802.11n. En 2009 se presentó 802.11n que fue diseñado para mejorar el ancho de banda de 802.11g, admitiendo hasta 600 Mbps. Lo anterior fue posible mediante el uso de la tecnología MIMO (*Multiple-Input & Multiple-Output*) que permite utilizar diferentes canales a la vez gracias a la incorporación de varias antenas. Este estándar puede trabajar en las bandas de 2.4 GHz y 5 GHz por lo que es compatible con dispositivos basados en todas las ediciones anteriores.

802.11ac. Este estándar fue presentado en el año 2013. Aunque fue diseñado para trabajar en la banda de 5 GHz utiliza tecnología inalámbrica de banda dual que admite conexiones simultáneas en las bandas de 2.4 GHz y 5 GHz ofreciendo compatibilidad con versiones anteriores. El ancho de banda nominal es de 450 Mbps en 2.4 GHz y de hasta 1300 Mbps en la banda de 5 GHz. En este estándar se implementa la tecnología MIMO Multiusuario, con la que el *router* puede enviar información a hasta cuatro usuarios que estén conectados a la red al mismo tiempo, por lo que se reduce la velocidad y las conexiones son más estables.

802.11ad. En 2012 se propone el estándar 802.11ad que opera en la banda de 60 GHz y puede suministrar una velocidad de transmisión de hasta 7 Gbps. También funciona en las bandas de 2,4 GHz y 5 GHz brindando soporte a los estándares anteriores. Su rango de trabajo en la banda de 60 GHz es de aproximadamente 10

metros y su consumo de energía en menor a una misma tasa de datos que en 802.11n u 802.11ac, siendo más eficiente para móviles y portátiles.

802.11ah. En 2017 fue aprobado este estándar que trabaja en la banda de 900 MHz permitiendo una mayor cobertura que una red de 2.4 GHz o 5 GHz. El ancho de banda que proporciona es de 150 Kbps, por lo que está orientado específicamente a aplicaciones de Internet de las Cosas donde se requiere amplia cobertura para cubrir un hogar y baja velocidad ya que se envía poca información.

Actualmente algunas de las mejoras en las que se está trabajando son las siguientes:

802.11ax. El objetivo de esta próxima generación que se espera se finalice en 2019, es permitir velocidades de hasta 10 Gbps utilizando la banda de 5 GHz. El nuevo estándar en desarrollo está considerando el uso del Acceso Múltiple por División de Frecuencia Ortogonal (OFDMA, *Orthogonal Frequency-Division Multiple Access*), MIMO multiusuario y otras mejoras tecnológicas que permitan utilizar al máximo el ancho de banda disponible.

802.11ay. Es el sucesor de 802.11ad, por lo que utiliza la banda de 60 GHz, está diseñado para proporcionar grandes cantidades de ancho de banda a distancias cortas. Este nuevo estándar promete mejorar el ancho de banda hasta 20 Gbps, aproximadamente una mejora tres veces mayor que 802.11ad. Las mejoras principales sobre 802.11ad son la adición de antenas MIMO con hasta cuatro flujos de datos simultáneos.

802.11az. Los dos estándares mencionados anteriormente están dirigidos a mejorar el ancho de banda para los usuarios. 802.11az, en cambio, busca mejorar la ubicación y el posicionamiento de los usuarios en una red. Dado que muchas de las mejoras utilizan antenas MIMO, el rendimiento del sistema mejora significativamente si el enrutador puede realizar un seguimiento preciso y rápido de la ubicación de cada usuario en la red. Esa es una tarea relativamente simple en un entorno estático con dispositivos de ubicación fija como televisores, pero mucho más difícil de hacer con dispositivos móviles como teléfonos inteligentes o tabletas. Las mejoras en la tecnología de localización permiten que una red Wi-Fi rastree y se conecte más

rápidamente a un dispositivo sin tener que desperdiciar recursos para encontrar al dispositivo antes de cada transmisión.

1.2.13. Simulador NS-2

Network Simulator 2 es un simulador de código abierto de eventos discretos creado por la Universidad de Berkeley orientado a la investigación en redes, teniendo en cuenta la estructura (topología) de la red y el tráfico de paquetes que posee la misma, con el fin de crear un diagnostico que muestre el comportamiento que se obtiene en la red bajo ciertas condiciones. A través de una simulación realizada en NS-2 es posible evaluar los resultados generados y garantizar la calidad del funcionamiento de un determinado protocolo, simular redes que poseen una implementación física compleja y validar un nuevo protocolo antes de su implementación física [26]. Además, soporta redes cableadas e inalámbricas y provee soporte para protocolos como TCP y UDP, en los que es posible estudiar el comportamiento de tráfico FTP, Telnet, Web, CBR y VBR. NS-2 permite modelar los sistemas a medida que éstos progresan a través del tiempo, modelar eventos aleatorios y predecir los efectos de las complejas interacciones entre estos eventos [30].

NS-2 basa su funcionamiento en dos lenguajes de programación, C++ y OTcl *(Object-oriented Tool Command Language*). La estructura de NS-2 está orientada a objetos, por lo tanto su núcleo se define por medio de clases que siguen una jerarquía, predominando la presencia de estructuras que ahorran código como es el caso de la herencia [62]. El núcleo de NS-2 está escrito en C++, un lenguaje robusto, compilado, usado para la manipulación de bytes, paquetes, definición de patrones y para la implementación de algoritmos.

Por su parte, OTcl es una extensión del lenguaje orientado a objetos Tcl desarrollado por el MIT (*Massachusetts Institute of Technology*). Este lenguaje es utilizado para la creación de los *scripts* que generan el escenario de simulación, en él se carga todo tipo de información, como cantidad de nodos, tamaño de los paquetes y ancho de banda [28].

Una simulación se inicia a través del *script* OTcl, que se somete al núcleo de NS-2 para su ejecución y finalmente, genera los resultados que se muestran en forma de tablas en el archivo *Trace* generado por NS-2. En la figura 1.19 se muestra el esquema general de funcionamiento de NS-2 [63].

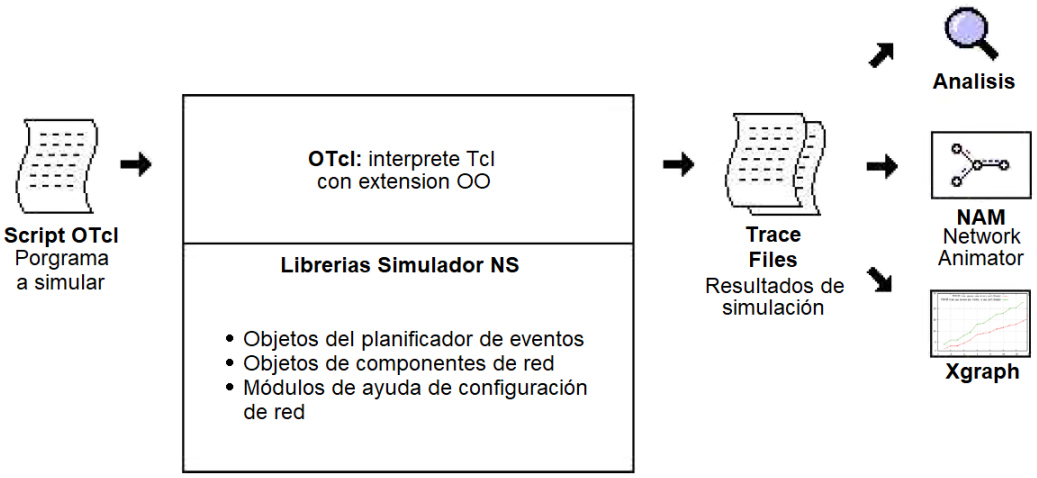


Figura 1.19. Esquema general de NS-2 [63].

Cada paquete enviado durante la simulación genera una línea en el archivo *trace* donde se registran todos los eventos ocurridos durante el proceso de simulación. En la figura 1.20 se presenta un ejemplo de un archivo *trace p*ara una simulación con tecnología cableada y en la figura 1.21 con formato inalámbrico.

```
A    B     C D  E    F     G       H  I   J   K  L
r 0.494 2 3 cbr 1000 ------- 2 1.0 3.1 44 44
r 0.498 1 2 cbr 1000 ------- 2 1.0 3.1 48 48
+ 0.498 2 3 cbr 1000 ------- 2 1.0 3.1 48 48
- 0.498 2 3 cbr 1000 ------- 2 1.0 3.1 48 48
+ 0.5   0 2 tcp   40 ------- 1 0.0 3.0  0 50
- 0.5   0 2 tcp   40 ------- 1 0.0 3.0  0 50
+ 0.5   1 2 cbr 1000 ------- 2 1.0 3.1 50 51
- 0.5   1 2 cbr 1000 ------- 2 1.0 3.1 50 51
r 0.502 2 3 cbr 1000 ------- 2 1.0 3.1 45 45
r 0.506 1 2 cbr 1000 ------- 2 1.0 3.1 49 49
```

Figura 1.20. Ejemplo archivo *Trace* - Tecnología Cableada [63].

Los campos del archivo *trace* se describen a continuación:

- A. Evento ocurrido, entrada (+) o salida (-) de la cola, descarte (d) o recepción (r) de paquete.
- B. Instante en segundos en el que ocurrió el evento.
- C. Nodo de origen del evento.
- D. Nodo de destino del evento.
- E. Tipo de paquete.
- F. Tamaño del paquete en bytes.
- G. Banderas utilizados para la notificación de congestión.
- H. Identificación del flujo de paquetes.
- I. Dirección de la fuente del evento.
- J. Dirección del destino del evento.
- K. Número de secuencia del paquete.
- L. Número que identifica el paquete en la red.

```
A     B         C   D   E  F    G    H      I        J              K
s 0.032821055 _1_ RTR --- 0 mensaje 32 [0 0 0 0] ------- [1: 255 -1: 255 32 0]
```

Figura 1.21. Ejemplo archivo *Trace* - Tecnología Inalámbrica [63].

Los campos del archivo *trace* Inalámbrico se describen a continuación:

- A. ACCIÓN: [s | r | D]: s - enviado, r - recibido, D – eliminado.
- B. CUANDO: el momento en que ocurrió la acción.
- C. DÓNDE: el nodo donde ocurrió la acción.
- D. CAPA: AGT - aplicación, RTR - enrutamiento, LL - capa de enlace, IFQ: cola de paquetes salientes (entre el enlace y la capa mac), MAC - mac, PHY – físico.
- E. Banderas.
- F. SEQNO: número de secuencia del paquete.
- G. TIPO: el tipo de paquete. CBR, DNP3, MODB.
- H. TAMAÑO: el tamaño del paquete en la capa actual.

- I. [abcd]: a - duración del paquete en el encabezado de la capa MAC, b - dirección MAC de destino, c - dirección MAC de la fuente, d - tipo de MAC del cuerpo del paquete.
- J. Banderas.
- K. [.....]: [nodo de origen IP: número de puerto, nodo de destino IP (-1 significa difusión): número de puerto, IP *header* ttl, IP del siguiente salto (0 significa nodo 0 o transmisión)]

NAM – Network Animator. Es una herramienta de animación que permite observar de forma gráfica los resultados del archivo *trace*. NAM soporta topología de capas, animación a nivel de paquetes, comportamiento de nodos, flujo, caída de enlace, colas y varios tipos de herramientas para la inspección de datos [38]. En la figura 1.22 se presenta un ejemplo de una red con cuatro nodos, donde se observa el flujo de datos entre ellos.

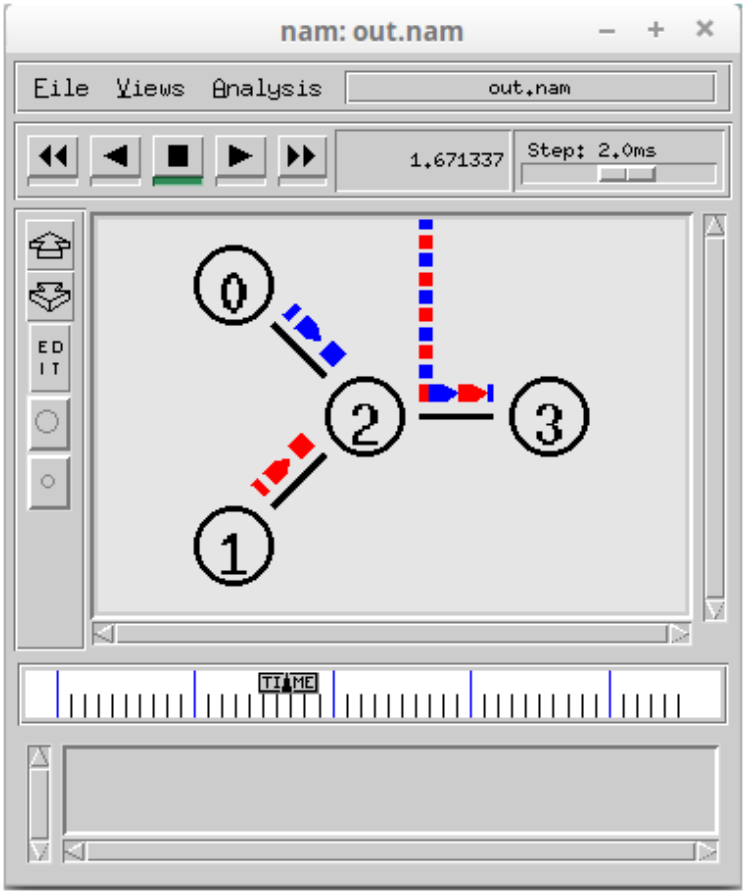


Figura 1.22. Ejemplo de topología en NAM [64].

1.2.14. Wireshark

Wireshark es un analizador de protocolos de red de código abierto, también conocido como *sniffer*, utilizado para realizar el análisis y solucionar problemas en redes de comunicaciones. Permite capturar y mostrar en tiempo real los paquetes transmitidos y recibidos en la red mediante la interfaz que se muestra en la figura 1.23. Los paquetes de datos capturados se pueden leer desde Ethernet, IEEE 802.11, PPP / HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, entre otros. Para el análisis se pueden seleccionar tramas individuales y en los diferentes bytes que la componen visualizar su significado, ya que Wireshark puede decodificar tramas para una amplia gama de protocolos, entre ellos Modbus y DNP3 [65].

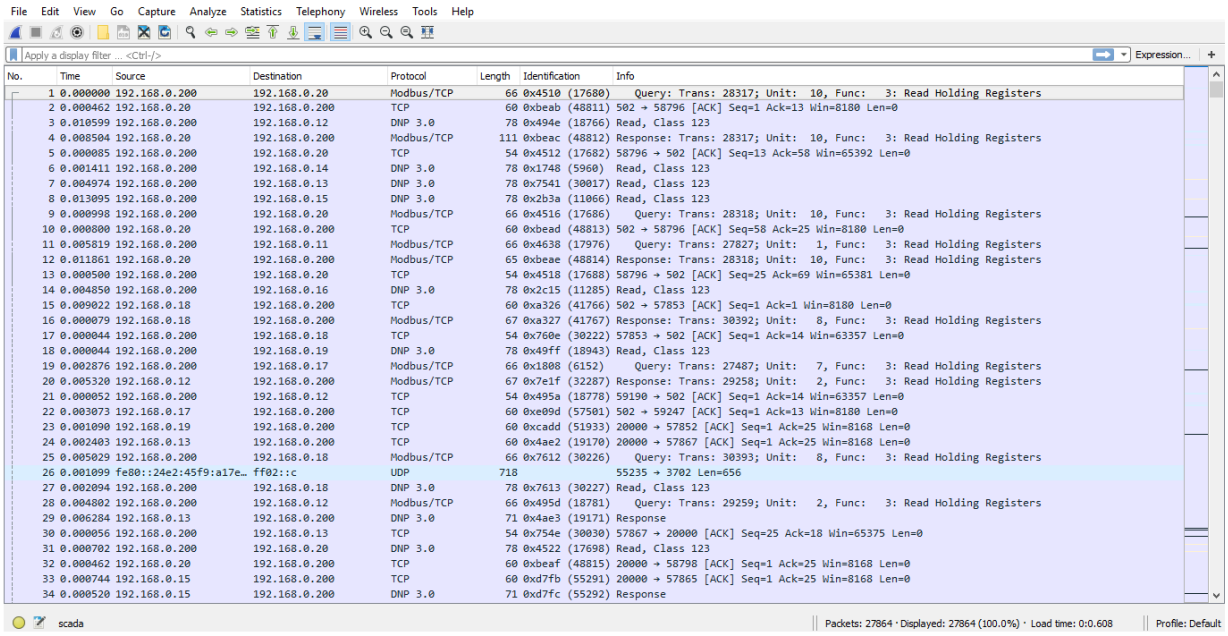


Figura 1.23. Interfaz de Wireshark.

Este *software* cuenta con diversas opciones de organización y filtrado de información, que permiten analizar detalladamente cada paquete o mediante recursos estadísticos, todos los paquetes capturados. Además, permite generar gráficos de algunos parámetros como retardos y *throughput*. La herramienta de gráficos se presenta en la figura 1.24.

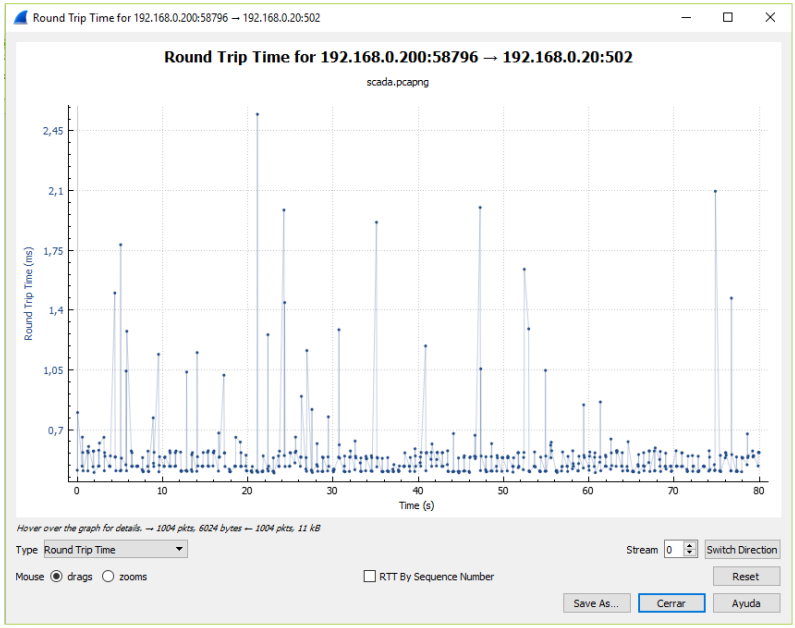


Figura 1.24. Gráfica de retardos en Wireshark.

1.3. Aporte investigativo.

Como resultado de este trabajo de grado se destacan los siguientes aportes:

1. Diseño de la infraestructura AMI para la *microgrid* de la Universidad de Nariño, considerando sus tres componentes básicos como: medidores inteligentes, SCADA y red de comunicaciones cumpliendo con los requisitos de QoS.
2. Modelo de los protocolos Modbus y DNP3. Se desarrolló un parche para el simulador NS-2 que permite implementar escenarios con Modbus y DNP3 sobre redes cableadas e inalámbricas
3. Análisis comparativo del rendimiento de los protocolos bajo diferentes escenarios. Se evaluó el desempeño de Modbus y DNP3 sobre Ethernet y Wi-Fi, por medio de pruebas de campo y simulación en NS-2, considerando los parámetros QoS de *delay* y *throughput.*
4. Se publicó un artículo en revista indexada con resultados parciales del trabajo, a continuación se detalla este documento.

1.4. Publicaciones

Como resultado parcial de este trabajo, se ha publicado el siguiente artículo:

- A. J. Cervelion-Bastidas, G. L. Agredo-Méndez y J. Revelo-Fuelagán, “Diseño y evaluación de desempeño de la infraestructura AMI para la microrred de la Universidad de Nariño”, Revista Ingeniería Solidaria, vol. 14, no. 26, 2018. doi: https://doi.org/10.16925/in.v14i26.2418.

En este documento se presentan las consideraciones para el diseño de la infraestructura AMI implementada en la Universidad de Nariño y se evalúa el desempeño de la red de comunicaciones.

1.5. Objetivos

1.5.1. Objetivo General.

- Analizar y evaluar el desempeño de los Protocolos Modbus y DNP3 en la red de comunicaciones de la Microrred eléctrica de la Universidad de Nariño.

1.5.2. Objetivos Específicos.

- Diseñar una red de comunicaciones para la microrred de la Universidad de Nariño contemplando la topología y tipos de tecnologías.

- Evaluar el desempeño de los protocolos Modbus/TCP y DNP3/TCP haciendo uso de una herramienta de simulación.

- Evaluar el desempeño de los protocolos Modbus/TCP y DNP3/TCP en la red de comunicaciones implementada en el Campus de la Universidad de Nariño.

- Establecer el protocolo que ofrezca el mejor desempeño a partir del análisis comparativo de los resultados obtenidos.

1.5. Metodología

El presente trabajo está dividido en dos fases, la primera se orienta al diseño de una red de comunicaciones para AMI y la segunda a la evaluación de desempeño de los protocolos Modbus y DNP3 mediante medidas de campo y simulación. Por lo anterior se han adoptado las metodologías propuestas en [66] y [67]. En [66], se presenta la metodología de diseño de red Top-Down, ampliamente utilizada en el diseño de redes de comunicaciones, por su parte en [67] se presenta una metodología para la implementación, verificación y validación de modelos de simulación. En la figura 1.25 se presentan las etapas de la metodología y a continuación se realiza una descripción de cada una de ellas:

Etapa 1. Identificación de necesidades: inicialmente se define el escenario para el diseño y se plantean las necesidades de la AMI, con el fin de determinar los requerimientos de la red de comunicaciones que permitan seleccionar la topología y tecnología de comunicaciones.
Etapa 2. Diseño físico de la red: se seleccionan las tecnologías de comunicación, se define la topología física de red y se realiza la elección de los equipos de medida y comunicación
Etapa 3. Diseño lógico de la red: se define la topología lógica de red y se determina el direccionamiento de los equipos.
Etapa 4. Pruebas de funcionamiento: se definen las métricas QoS y las herramientas, posteriormente se ejecutan las pruebas de funcionamiento de la red de comunicaciones y de la AMI en general.
Etapa 5. Definición del sistema: comprende la definición de los objetivos del modelamiento y descripción del sistema a modelar.
Etapa 6. Formulación del modelo: se seleccionan las herramientas necesarias para llevar a cabo el estudio, se definen los modelos requeridos y se construyen los diagramas de flujo que describan el modelo.

Etapa 7. Implementación del modelo: se construye el modelo mediante códigos computacionales.

Etapa 8. Verificación del modelo: consiste en comprobar que el modelo simulado cumple con los requisitos de diseño para los que se elaboró, es decir, se trata de evaluar que el modelo se comporta de acuerdo a su diseño y de ser necesario se regresa a las etapas 6 y 7.

Etapa 9. Validación del modelo: en esta etapa se valoran las diferencias entre el funcionamiento del simulador y el sistema real.

Etapa 10. Experimentación: se realizan pruebas sobre los escenarios seleccionados recolectando información pertinente para el análisis.

Etapa 11. Análisis de resultados: se estudian los datos recolectados en las pruebas con el fin de determinar el desempeño de los protocolos.

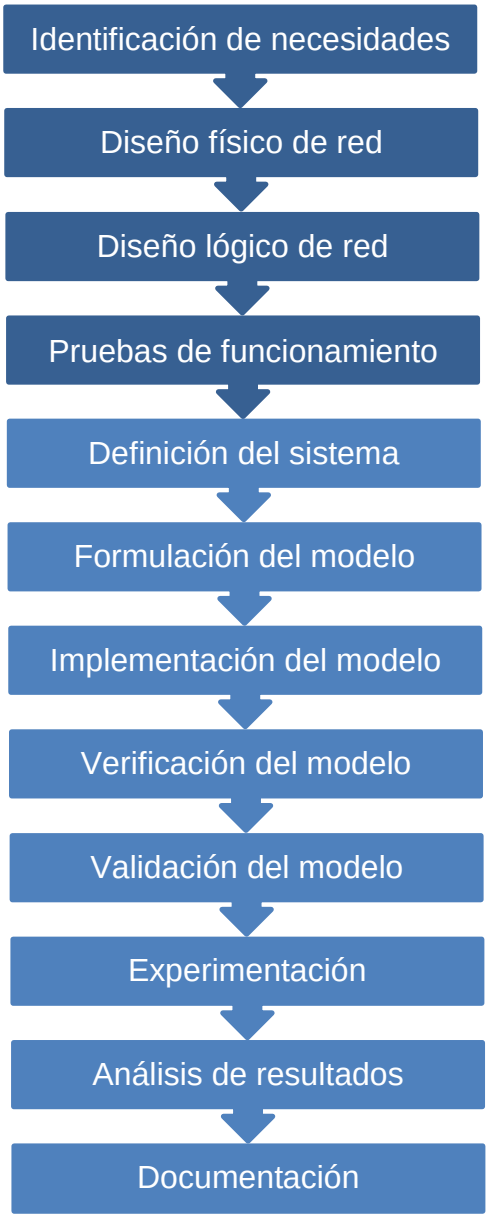


Figura 1.25. Etapas de la metodología utilizada.

1.6. Organización del documento

El documento se encuentra organizado por capítulos, a continuación se describe el contenido de cada uno de ellos:

En el capítulo 2, se presenta el diseño de la infraestructura AMI para la *microgrid* de la Universidad de Nariño. Inicialmente, se contextualiza el escenario, se describe la ubicación del centro de gestión y se define la cantidad y ubicación de los medidores inteligentes con el fin de determinar la topología y tecnologías a utilizar. Posteriormente, se presenta el diseño lógico y físico de la red de comunicaciones, donde se define la conexión lógica y física de los dispositivos y se selecciona el *software* SCADA, los equipos de comunicación y de medida. Finalmente, se presentan los resultados de las pruebas de funcionamiento de la AMI.

En el capítulo 3, se describe el procedimiento para el modelamiento de los protocolos Modbus y DNP3. Inicialmente, se formulan los modelos por medio de diagramas de estado que permiten determinar las rutinas que se deben programar. Luego, se revisan los modelos de canal disponibles en NS-2 y los parámetros que se pueden modificar. En el siguiente paso se presentan las métricas seleccionadas para la evaluación de desempeño, en este punto se eligió *delay* y *throughput,* por ser métricas ampliamente utilizadas en la evaluación de desempeño, además los valores de *delay* definen la funcionalidad de la AMI. Posteriormente, se describe la implementación de los modelos, definiendo los archivos que se deben modificar en el *core* de NS-2. Paso seguido se verifica el funcionamiento de los modelos y finalmente se presenta la validación de los modelos por medio de comparaciones gráficas y estadísticas con el sistema real.

En el capítulo 4, se presentan los resultados de la evaluación de desempeño de los protocolos Modbus y DNP3, para ello se plantean tres escenarios, en el primero se propone la comunicación del SCADA con un medidor inteligente bajo tecnología Wi-Fi. En el segundo, también se considera la comunicación del SCADA con un medidor, pero en este caso bajo tecnología Ethernet. Finalmente, en el tercer escenario se considera la comunicación del SCADA con los diez medidores que componen la AMI.

Capítulo 5. En este capítulo se presentan las conclusiones y recomendaciones del trabajo.

Capítulo 2

Diseño de la infraestructura AMI

Las funcionalidades y ventajas de una *microgrid* no serían viables si no se cuenta con una infraestructura AMI capaz de soportar el flujo de información que la *microgrid* necesita para su correcto funcionamiento, debe permitir el monitoreo en tiempo real, proporcionar información para control de carga y respuesta a la demanda para que el consumo de energía sea más eficiente [68].

Por lo anterior, es de suma importancia diseñar correctamente la AMI y en especial la red de comunicaciones, eje principal de la infraestructura. En el diseño se consideran aspectos como topografía del terreno, ubicación y distancia de los dispositivos a interconectar, con el fin de seleccionar la topología y tecnologías de comunicación que más se ajusten a las necesidades de la *microgrid*. A continuación se desarrolla cada una de las fases de la metodología propuesta.

2.1. Identificación de necesidades

Para identificar las necesidades de diseño inicialmente se describe el lugar donde se desarrolla el trabajo, siendo este el Campus de la Universidad de Nariño que cuenta con 17 edificios en un área plana de aproximadamente 130.000 mt², que sugiere la implementación de una red NAN. Posteriormente, se define la cantidad y ubicación de puntos de medida, que se determinaron a partir de la observabilidad de un sistema. Este concepto expresa que una red es observable si se puede obtener de

ella un número suficiente de medidas que permitan determinar los estados del sistema [69]. Para lograr la observabilidad de la red eléctrica lo ideal es la instalación de equipos de medida en cada uno de los nodos del sistema, este procedimiento, aunque válido, es probablemente inviable por los excesivos costos de infraestructura y equipos [70]. Por lo tanto una buena opción y considerando que el número de subestaciones del Campus es pequeño, se optó por la instalación de un medidor inteligente en cada uno de los nueve transformadores de potencia y uno en el generador diésel de respaldo dentro del Campus.

Finalmente, se determinó la ubicación del centro de gestión, el propósito es que el SCADA se encuentre aproximadamente a la misma distancia de todos los puntos de medida, con el fin de evaluar la viabilidad de cada tecnología. Por este motivo, el centro de gestión se encuentra en el laboratorio de microrredes ubicado en el edificio de laboratorios de docencia. En la figura 2.1 se presenta en verde la ubicación del centro de gestión y en rojo los medidores inteligentes con su distancia en línea recta hasta el centro de gestión.

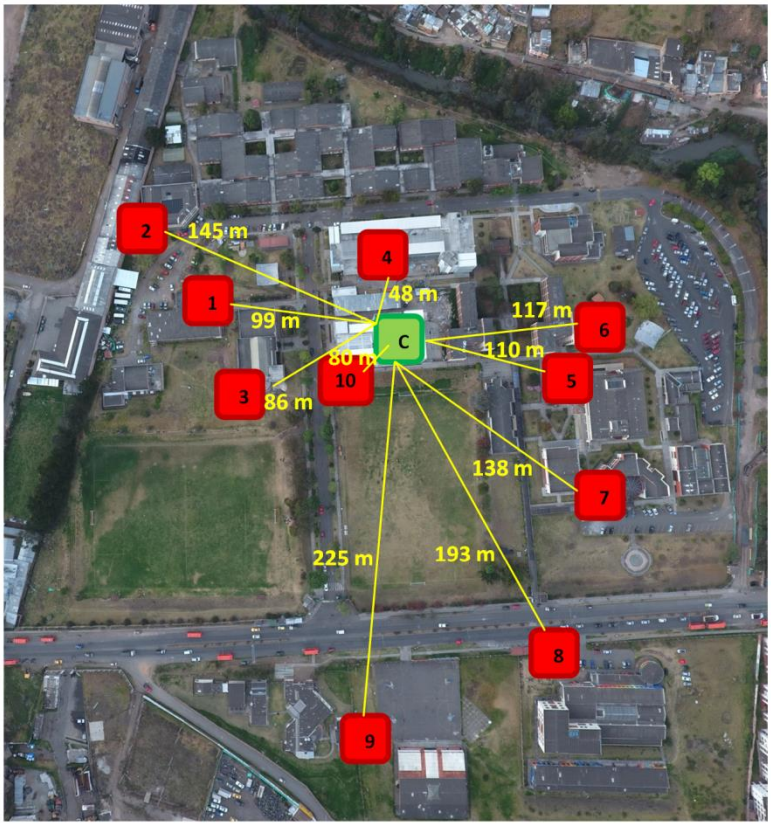


Figura 2.1. Ubicación de medidores inteligentes y centro de gestión.

2.2. Diseño físico de la red

En esta sección se presenta el proceso de selección de la tecnología y topología de comunicación, además de los equipos necesarios para la implementación de la AMI.

2.2.1. Tecnología y topología de comunicación

En esta subsección se presenta un análisis comparativo de las tecnologías descritas en la sección 1.2.7 con el fin de determinar la más apropiada para la implementación de la AMI, un resumen de la comparación se presenta en la tabla 2.1. La evaluación de los criterios presentados en esta tabla se realizó considerando los siguientes aspectos:

- Ethernet (sin repetidores), PLC y ZigBee tienen corto alcance, que no les permite cubrir las distancias presentadas en la figura 2.1.
- PLC, RS-485, y ZigBee ofrecen un bajo ancho de banda, que limitaría futuras ampliaciones en la red de comunicaciones y a ofrecer el valor mínimo recomendado (2-5 Mbps) para aplicaciones AMI [71].
- La instalación de tecnologías cableadas está en desventaja pues requieren de mayor infraestructura y de instalaciones fijas, por el contrario en las inalámbricas la instalación es más sencilla y el sistema se puede reconfigurar mayor facilidad.
- La fibra óptica requiere de inversiones considerables en su instalación. Por su parte, WiMax y la red celular requieren de costos adicionales para su operación por trabajar en bandas licenciadas.

Tabla 2.1. Resumen comparativo entre tecnologías utilizadas en *microgrids*.

Tecnología	Alcance	Ancho de Banda	Instalación	Costo
Ethernet	Bajo	Alto	Compleja	Medio
PLC	Bajo	Bajo	Compleja	Bajo
RS-485	Medio	Bajo	Compleja	Medio
Fibra Óptica	Alto	Alto	Compleja	Alto
Wi-Fi	Medio	Alto	Sencilla	Medio

WiMax	Alto	Alto	Sencilla	Alto
ZigBee	Bajo	Bajo	Sencilla	Bajo
Celular 4G	Alto	Alto	Sencilla	Alto

Por otra parte, en la tabla 2.2 se presentan los resultados del análisis comparativo donde se observa que Wi-Fi se ajusta a los requerimientos de la implementación. Esta tecnología se utilizó para los medidores 1 a 9, en el caso del medidor 10 no es posible utilizar esta tecnología debido a que el trasformador de potencia se encuentra al interior del edificio de laboratorios de docencia, donde también se encuentra el centro de gestión, por tal motivo para este medidor se utilizó Ethernet, teniendo en cuenta que el trayecto para la conexión es de 80 metros, cumpliendo con las distancias permitidas por esta norma.

Tabla 2.2. Resultados análisis comparativo entre tecnologías utilizadas en *microgrids*.

Tecnología	Alcance	Ancho de Banda	Instalación	Costo
Ethernet	✗	✓	✗	✓
PLC	✗	✗	✗	✓
RS-485	✓	✗	✗	✓
Fibra Óptica	✓	✓	✗	✗
Wi-Fi	✓	✓	✓	✓
WiMax	✓	✓	✓	✗
ZigBee	✗	✗	✓	✓
Celular 4G	✓	✓	✓	✗

El siguiente paso es definir la topología física de la red, considerando que el objetivo principal es que los medidores se comuniquen mediante Wi-FI con el SCADA, la red debe configurarse en modo Infraestructura (BSS, *Basic Service Set*), en donde existe un elemento de “coordinación”, que gestiona la conexión con los clientes inalámbricos [72]. Para ello se requiere de un punto de acceso que garantice cobertura en la ubicación de los medidores inteligentes, en donde se debe instalar un radio Wi-Fi. En la figura 2.2 se presenta la configuración de la red, que consta de 9 conexiones inalámbricas Wi-Fi, una conexión Ethernet para el medidor 10 y otra para la conexión con el SCADA.

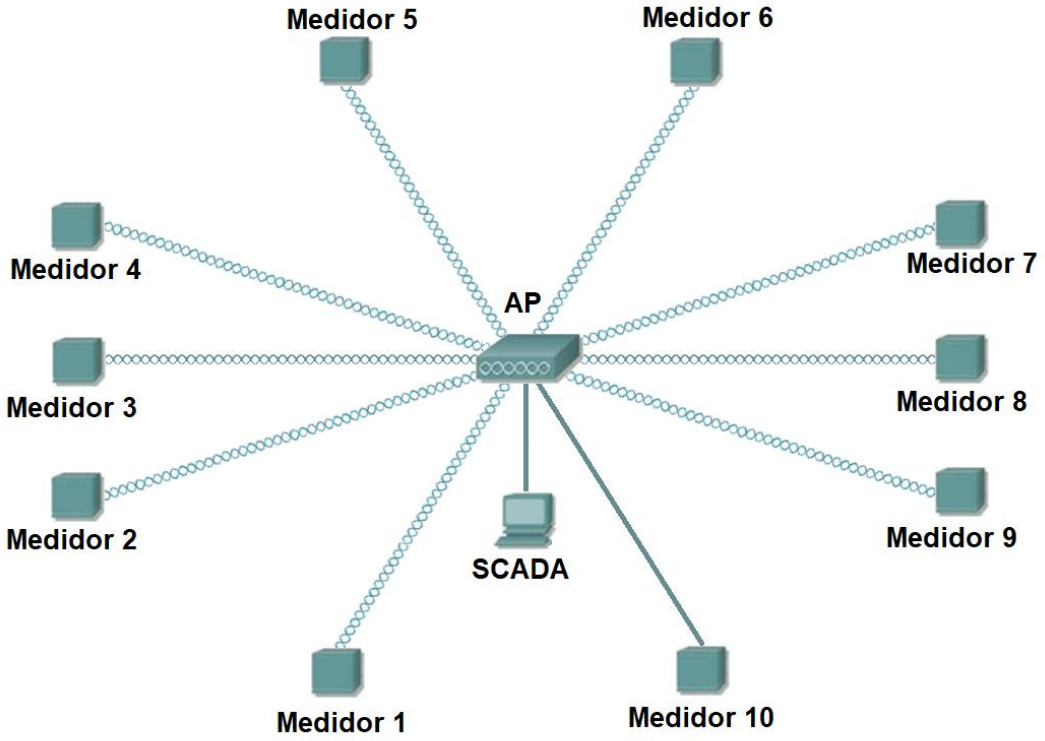


Figura 2.2. Topología de la red de comunicaciones.

Finalmente se deben seleccionar los equipos para la implementación de la red de comunicaciones y del sistema AMI en general. A continuación se presentan los criterios de selección:

2.2.2. Medidores inteligentes

En la sección 1.2.3 se describieron algunas de las características a considerar para la elección de los medidores inteligentes, en la tabla 2.3 se presenta una comparación entre cinco medidores utilizados en redes inteligentes.

Tabla 2.3. Comparación entre medidores inteligentes.

Características	A1884 ALPHA (elster)	PowerMonitor 1000 (Allen Bradley)	PowerLogic ION7550 (Schneider)	SENTRON PAC3200 (Siemens)	Expertmeter EM133 (Satec)
Voltaje max. Medida	528 V	347V AC	347V AC	400 V	400 v
Corriente max. Medida	10 A	5 A	5 A	5 A	5 A

Medición de kW, kVAR, KVA, PF.	✓	✓	✓	✓	✓
Medición de Armónicos	✗	✗	✓	✓	✓
Precisión	0.2S	0.5S	0.2S	0.2S	0.5S
Puertos de comunicación	- Ethernet - RS-485 (Opcional)	- RS-485 - Ethernet	- RS-485 - Ethernet (Opcional)	- Ethernet	- RS-485 - Ethernet (Opcional)
Salidas digitales	✗	✗ (opcional)	✗ (opcional)	✓	✓
Protocolos de comunicación	- DNP3 - ANSI C12.21	- Modbus	- Modbus - DNP3 - IEC61850	- SEAbus - Modbus	- Modbus - DNP3 - IEC 60870

A continuación se realiza un análisis de la tabla 2.3:

- Voltaje máximo de medida: todos los medidores tienen un rango de medida que cumple con las necesidades de la AMI, ya que se requiere medir voltajes de 120 v de Línea a Neutro.
- Corriente máxima de medida: según un estudio previo las corrientes que se medirán en el Campus Universitario se encuentran en el rango de 80 a 700 amperios, por lo tanto todos los medidores requieren de transformadores de corriente para su funcionamiento.
- Medición de kW, kVAR, KVA, PF y Medición de Armónicos: estas variables permiten determinar con mayor exactitud las condiciones de la red eléctrica, en este caso los medidores A1884 ALPHA y PowerMonitor 1000 no cumplen con estas medidas.
- Precisión: para este tipo de aplicaciones la precisión recomendada es 0.5S [73]. Como se observa en la tabla 2.3 los medidores A1884 ALPHA, PowerLogic ION7550 y SENTRON PAC3200 tienen mejor precisión de la recomendada lo que sugiere mayor exactitud en las medidas, pero así mismo su precio es en promedio 4 veces mayor que uno de precisión 0.5S.
- Puertos de comunicación: como se observa en la tabla ningún medidor está equipado con interfaz Wi-Fi y en algunos modelos es un módulo opcional que garantiza una cobertura de apenas 100 metros, con lo cual no se cubriría las necesidades de la AMI. Por este motivo se optó por equipos con puerto Ethernet que faciliten la integración con una interfaz Wi-Fi. En este sentido los

medidores A1884 ALPHA, PowerMonitor 1000 y SENTRON PAC3200 cumplen con este requerimiento, mientras que en el PowerLogic ION7550 y Expertmeter EM133 lo disponen como modulo opcional.

- Salidas digitales: esta característica que permite realizar control de carga solo está disponible en los medidores SENTRON PAC3200 y Expertmeter EM133.
- Protocolos de comunicación: esta característica es la más importante para la elección del medidor, considerando que el objetivo principal de este trabajo es evaluar los protocolos Modbus y DNP3. En la tabla 2.3 se observa que el medidor A1884 ALPHA está equipado con DNP3 y su elección obliga la adquisición del PowerMonitor 1000 o del SENTRON PAC3200 que soportan Modbus. Por otro lado, según la tabla 2.3 una buena opción son los medidores PowerLogic ION7550 y Expertmeter EM133 que soportan los dos protocolos, pero como se mencionó anteriormente el PowerLogic ION7550 tiene un costo bastante elevado. Por todo lo anterior, el medidor elegido para la implementación de la AMI es el SATEC Expertmeter EM133, a pesar de que requiere del módulo Ethernet adicional, en la figura 2.3 se presenta el medidor con el módulo de comunicaciones instalado.

Figura 2.3. Medidor Satec EM133 [74].

Finalmente, los requerimientos de instalación, alimentación y protección para el medidor se encuentran en el manual de instalación y operación “EM13x Series SMART MULTIFUNCTION METER” [75].

2.2.3. Sistema de Comunicaciones

Como se mencionó en la subsección 2.2.1 la red de comunicaciones necesita de un punto de acceso, nueve radios y un enlace Ethernet para la comunicación con los medidores. Entre las múltiples marcas como Cisco Aironet, TP-Link TL-WA, ALfa Network N2C que ofrecen dispositivos para este tipo de aplicación, la marca americana Ubiquiti Networks es una buena alternativa por su relación calidad/precio. Por lo anterior, los equipos seleccionados se describen a continuación:

- Nodo central: se utilizó un radio UBiQUiTi Rocket M2 que trabaja en la banda de 2.4 GHz, tiene una potencia de transmisión de 630mW y una velocidad de 150 Mbps. Esta acoplado a una antena omnidireccional AMO-2G10 de 10 dBi de ganancia, cubriendo todo el Campus Universitario.
- Nodos terminales: se instalaron radios UBiQUiTi NanoStation loco M2 que trabajan en la banda de 2.4 GHz. Se ha seleccionado este dispositivo por sus múltiples ventajas como fácil instalación con alimentación POE, antena integrada de 8 dBi de ganancia, Interfaz de red Ethernet (Cat. 5, RJ-45), potencia de transmisión de 23dBm y su bajo costo.

Por su parte, para las conexiones de los medidores a los NanoStation loco M2 y del medidor 10 al centro de gestión, se utilizó cable STP categoría 5e y conectores RJ45 apantallados.

2.2.4. Centro de gestión

Para la gestión de la red y de los datos, es necesario un software robusto y confiable que brinde las herramientas necesarias para la gestión de la *microgrid*. Aplicaciones como *PowerStudio Scada Delux* de *Circutor* o *Power Monitoring Expert* de *Schneider Electric*, cumplen con las expectativas pero su licencia tiene un costo considerable. En este punto, se recibió el apoyo de la compañía canadiense *Survalent* Technology, con amplia trayectoria en el sector eléctrico, quienes facilitaron una licencia Demo del *software* SurvalentONE SCADA, que se compone de los siguientes aplicativos:

- SCADA *Explorer*: se utiliza para la creación de la base de datos (SCADA *Server*), establecer las líneas de comunicación, definir protocolos de comunicación, configuración de los dispositivos e información a solicitar.
- SmartVU: en este aplicativo se realiza la Interfaz con el usuario.

Por su parte, SurvalentONE SCADA solicita a los medidores las siguientes variables eléctricas: voltaje, corriente, frecuencia, factor de potencia, potencia real, potencia reactiva, potencia aparente, Distorsión Armónica Total (THD, *Total Harmonic Distortion*) de voltaje, Distorsión Armónica Total (THD, *Total Harmonic Distortion*) de corriente. Además, se ha configurado para que almacene las variables en la base de datos cada 15 minutos.

Finalmente, considerando que la adquisición de datos debe realizarse continuamente, los aplicativos de SurvalentONE SCADA se instalaron en un servidor que permita el funcionamiento continuo del sistema las 24 horas del día. Para ello se utilizó el servidor HP Proliant ML110 Gen 9, que tiene las siguientes características:

- Procesador (1): Intel Xeon (6-Core) E5-2603v4 - 1.7GHz.
- Memoria RAM: Estándar 8GB (1x8GB)
- Discos Duros (1): 2TB HDD
- Controlador de Red: Tarjeta integrada de dos puertos de 1Gb - HP 330i

2.3. Diseño lógico de la red

Considerando la cantidad de equipos a conectar, se diseñó una red clase C, con direccionamiento estático, necesario debido a que en el SCADA se debe configurar la IP a la cual se solicita la información, además, permite detectar fallas más fácilmente. En la tabla 2.4 se presentan las asignaciones IP y en la figura 2.4 el diagrama lógico.

Tabla 2.4. Asignación de direcciones IP a los elementos de la AMI.

Dirección IP	Dispositivo
192.168.0.1/24	Servidor
192.168.0.2/24	AP Rocket M2
192.168.0.11/24	Medidor 1
192.168.0.12/24	Medidor 2
192.168.0.13/24	Medidor 3
192.168.0.14/24	Medidor 4
192.168.0.15/24	Medidor 5
192.168.0.16/24	Medidor 6
192.168.0.17/24	Medidor 7
192.168.0.18/24	Medidor 8
192.168.0.19/24	Medidor 9
192.168.0.20/24	Medidor 10
192.168.0.31/24	NanoStation loco M2 – Medidor 1
192.168.0.32/24	NanoStation loco M2 – Medidor 2
192.168.0.33/24	NanoStation loco M2 – Medidor 3
192.168.0.34/24	NanoStation loco M2 – Medidor 4
192.168.0.35/24	NanoStation loco M2 – Medidor 5
192.168.0.36/24	NanoStation loco M2 – Medidor 6
192.168.0.37/24	NanoStation loco M2 – Medidor 7
192.168.0.38/24	NanoStation loco M2 – Medidor 8
192.168.0.39/24	NanoStation loco M2 – Medidor 9

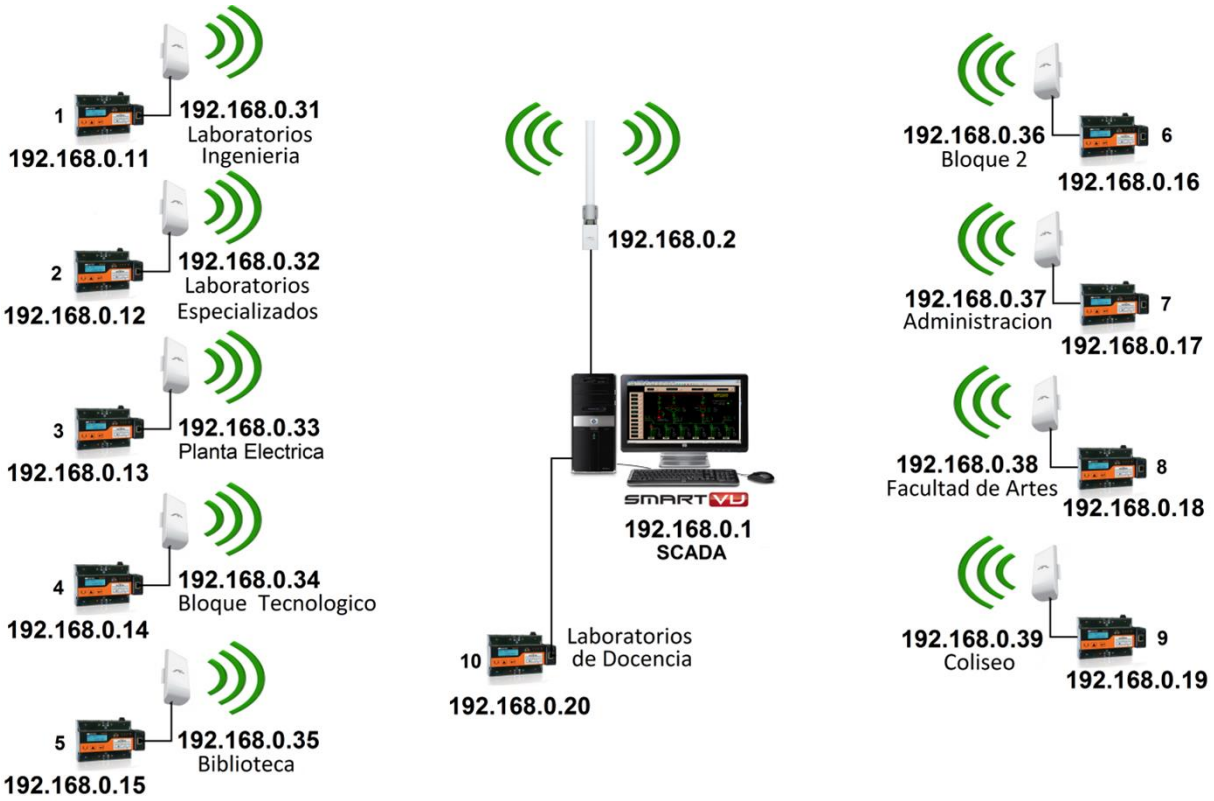


Figura 2.4. Diagrama lógico de red.

2.4. Pruebas de funcionamiento

Una vez realizado el diseño de la red de comunicaciones y seleccionado los equipos se realizó la implementación de la infraestructura AMI en el Campus universitario. Finalmente, la última fase de la metodología es realizar pruebas de funcionamiento de la red de comunicaciones, para conocer su capacidad máxima y comprobar que cumple los requerimientos para la implementación de la AMI. Para este fin, se consideran las siguientes métricas de rendimiento [15, 9]:

- ***Delay*****:** es el tiempo que tardan los datos para que viajen, ya sea en un solo sentido o de ida y vuelta, a través de la red de comunicaciones. Este retraso es producido por la demora en la transmisión y propagación de paquetes dentro de la red. Una característica importante de la AMI es que muchas de las interacciones se realizan en tiempo real, por lo tanto los

mensajes deben transmitirse dentro de un marco de tiempo muy corto, es así como para señales de protección el tiempo permitido está en el rango de 3 - 16 ms, para monitoreo en tiempo real el rango es de 16 - 100 ms y para un sistema SCADA de 100ms - 2s [76].

- **Ancho de banda:** es la velocidad de datos medida en bits por segundo (bps). Para asegurar un óptimo funcionamiento de la AMI se debe ofrecer un ancho de banda grande sin que ello implique costos exagerados. En ese orden de ideas, se recomienda un ancho de banda de 2 - 5 Mbps para la transmisión de múltiples variables y datos de control [71].
- ***Throughput* (Rendimiento):** es la tasa de bits neta con la que los datos son transferidos a través de un enlace de comunicaciones en un determinado período de tiempo. Este dato puede ser entregado sobre un enlace físico o lógico, o a través de un cierto nodo de la red. Por regla general, el *throughput* es medido en bits por segundo (bps) o paquetes por segundo (pps) [16].
- **Pérdida de paquetes:** esta métrica permite identificar las condiciones de la red, cuando los paquetes se transmiten desde la fuente al destino se espera que sean entregados en los *buffers* del receptor, de lo contrario se obliga a una retransmisión lo que se traduce en mayores retardos y por lo tanto ineficiencia de la red. Se debe procurar que las perdidas sean mínimas debido a que podría tratarse de información crítica para la operación segura de la *microgrid*. Una pérdida aceptable es 10^{-3} (implica que máximo un (1) paquete de datos puede contener errores de cada 1000 paquetes) [77].

Por otro lado, las herramientas utilizadas en las pruebas de funcionamiento fueron las siguientes [15]:

- **Fping:** es una herramienta similar a ping, pero con un mejor rendimiento, además, permite realizar ping a múltiples host. Se utilizó para medir *delay*.
- **JPerf:** es un programa cliente-servidor que permite medir la velocidad máxima que alcanzan 2 host conectados a una red local, con esta herramienta se midió el ancho de banda.
- ***Throughput Test:*** esta utilidad usa un cliente y un servidor para medir *throughput* y pérdida de paquetes.

Es importante aclarar que el objetivo en esta fase era determinar el desempeño de los canales de comunicación instalados para los diez medidores, por lo tanto en este proceso no se incluyen los protocolos Modbus y DNP3, en la tabla 2.5 se presentan los resultados de las pruebas realizadas en un periodo de 3600 segundos.

Inicialmente, la columna de *delay* presenta los valores promedio para los diez enlaces, aquí se observa que el enlace para el medidor 10 tiene el *delay* más bajo, puesto que se implementó con tecnología cableada Ethernet. En general, el *delay* promedio de todos los enlaces cumple con los requisitos de una infraestructura AMI, garantizando la transmisión de información en los tiempos establecidos de 3 - 16 ms para protección, 16 - 100 m para monitoreo en tiempo real y 100 ms - 2 S para sistemas SCADA [76].

Para el ancho de banda disponible en cada enlace, se observa que este valor se encuentra alrededor de los 94 Mbps y claramente cumple sin inconveniente el valor mínimo recomendado (2 - 5 Mbps) para aplicaciones AMI [71].

Del mismo modo el *throughput* promedio medido en los enlaces inalámbricos se encuentra alrededor de los 86 Mbps y para el enlace Ethernet es de 90 Mbps, estos valores supera sin dificultad los valores propuestos para sistemas SCADA, que se encuentran cerca de los 100 Kbit/s.

La pérdida de paquetes en el enlace del medidor 10 por utilizar conexión cableada muestra el nivel más bajo con un porcentaje promedio de 0,0%. Mientras que los enlaces Wi-Fi son menos confiables; sin embargo, su valor se encuentra por debajo del 1%, valor aceptado para esta métrica.

En general se puede concluir que la tecnología Ethernet tiene un mejor desempeño, ya que si bien Ethernet sufre de atenuaciones e interferencias, estas son mayores en los enlaces Wi-Fi. El inconveniente de Ethernet es que no siempre se puede implementar, por limitaciones como distancia máxima de conexión sin repetidores, infraestructura necesaria y ubicación geográfica de los puntos a medir.

Tabla 2.5. Resultados pruebas de desempeño.

Enlace Medidor	*Delay* [ms]	Ancho de banda [Mbps]	*Throughput* [Mbps]	Pérdida de paquetes [%]
1	4.7	95.1	86.5	0.25
2	3.4	94.8	86.7	0.26
3	3.6	94.8	86.4	0.22
4	3.7	94.8	86.5	0.29
5	3.0	94.0	86.6	0.26
6	3.2	94.8	86.5	0.24
7	3.3	95.5	86.8	0.25
8	3.8	94.0	86.5	0.29
9	3.3	95.5	86.3	0.23
10	0.7	94.9	90.1	0.0

Finalmente, una vez verificado que la red de comunicaciones cumple con los requerimientos para AMI, se ejecutó el SCADA y se inició con el monitoreo de la red eléctrica del Campus Universitario. En la figura 2.5, se presentan dos imágenes de SmartVU donde se observa la interfaz gráfica que permite visualizar en tiempo real la información enviada por los medidores. En la parte superior de la imagen se presenta la interfaz principal que consta de un mapa del Campus en el que se representa la ubicación de los medidores por medio de un cuadro verde, siempre y cuando exista comunicación con dicha estación, de lo contrario el cuadro es rojo. Al presionar uno de los cuadros aparece la interfaz de la imagen inferior, en la que se presenta la información recibida. Además, es posible presentar la información almacenada como se muestra en la figura 2.6, en ella se observan las curvas de demanda de los bloques de administración y biblioteca, que permiten la implementación de la gestión energética, programas de ahorro de energía y acciones propias de las *microgrids*.

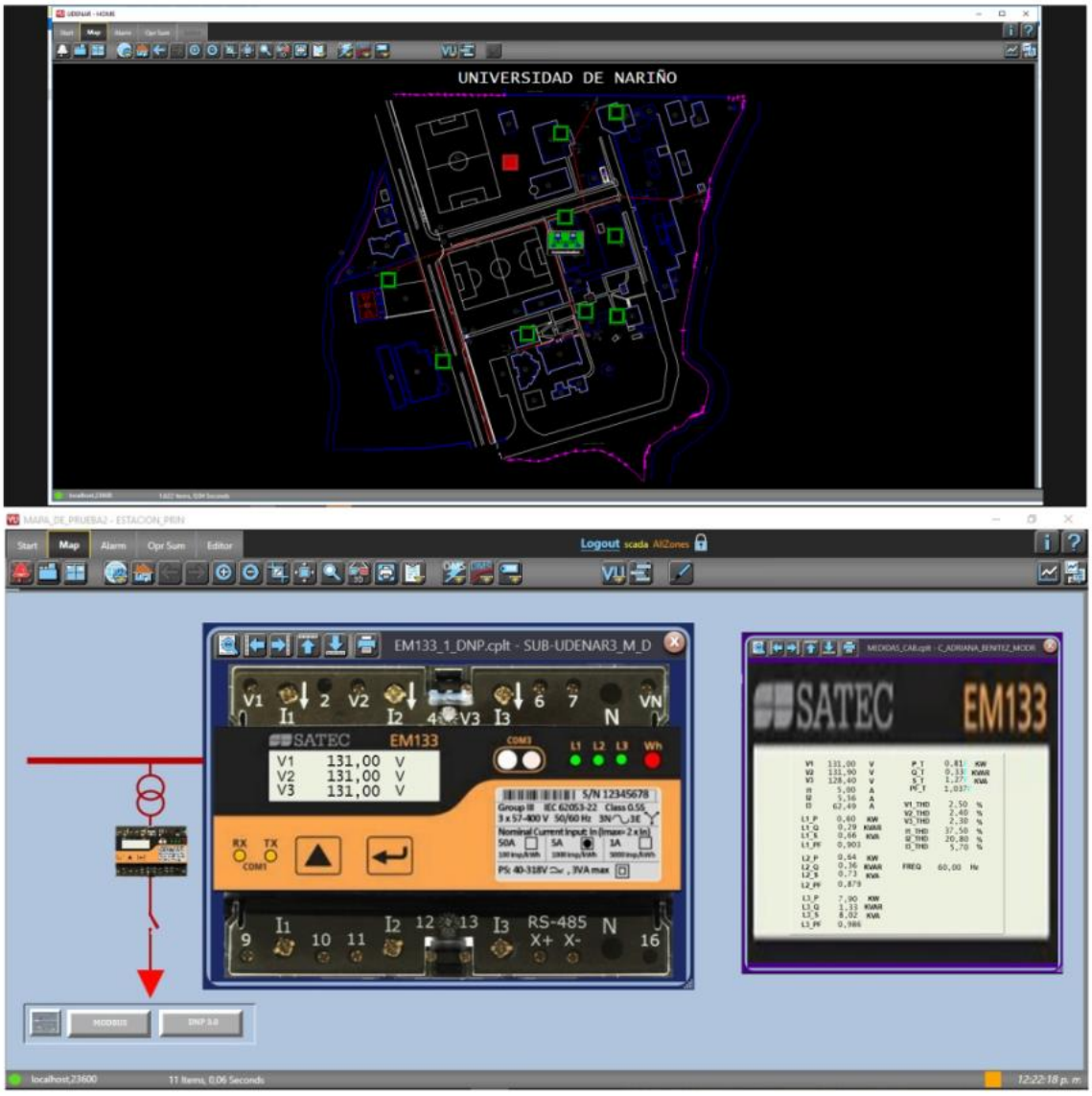


Figura 2.5. Interfaz de usuario diseñada en SmartVU.

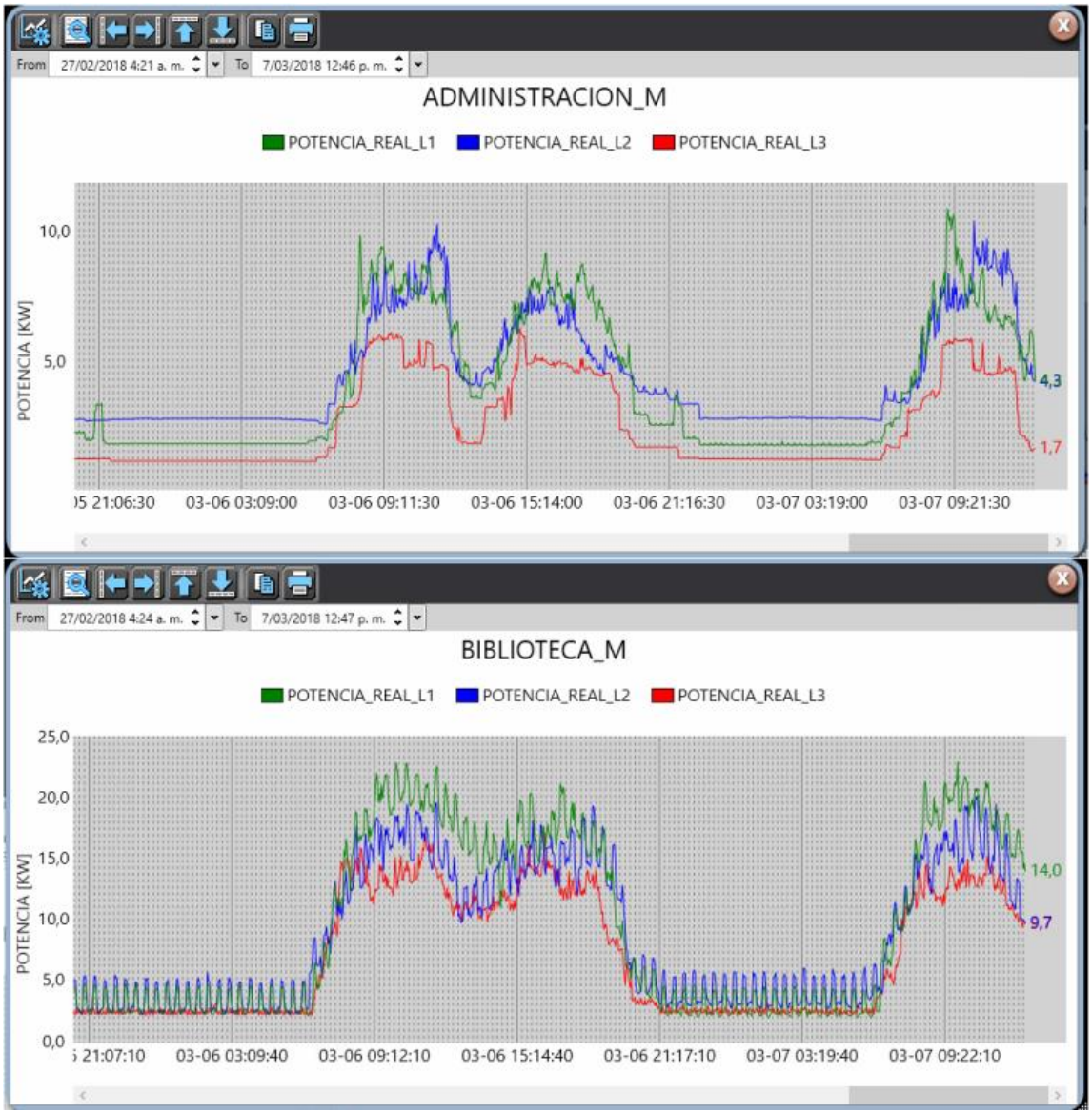


Figura 2.6. Gráficas de la información almacenada en el SCADA.

Capítulo 3

Diseño del modelo para Modbus y DNP3

En este capítulo se presenta el diseño de los modelos para los protocolos Modbus y DNP3 sobre NS-2, en este proceso se describe el funcionamiento de los protocolos y se definen las funciones de cada modelo por medio de diagramas de flujo. Posteriormente, se documenta la implementación de los modelos mediante códigos computacionales, destacando las modificaciones realizadas en el núcleo de NS-2. El siguiente paso es verificar los modelos para lo cual se simula un escenario con dos puntos, con el fin de visualizar el flujo de información entre ellos que permita comprobar el funcionamiento de los modelos. Finalmente, se realiza una validación de los modelos con las medidas reales.

3.1. Definición del sistema

Los protocolos de comunicación definen las reglas con las que se transmite la información y su elección es fundamental para garantizar la transferencia de datos dentro de la *microgrid*. El estudio a desarrollar tiene como objetivo el análisis de desempeño de los protocolos Modbus y DNP3 en la infraestructura AMI con el fin de determinar sus funcionalidades y como incide su estructura en el tiempo de transferencia de información. Para ello se requiere de una herramienta que permita el modelamiento de los protocolos Modbus y DNP3 y realizar simulaciones de la red de comunicaciones con tecnología Ethernet y Wi-Fi en diferentes escenarios.

3.2. Formulación del modelo

Esta fase comenzó con una revisión bibliográfica de los estudios presentados en la sección 1.1, con el fin de obtener la fundamentación teórica para el desarrollo del trabajo. En este punto, se analizaron las características y especificaciones técnicas de Modbus y DNP3. Además, se desarrolló un estudio de NS-2, realizando algunas simulaciones de redes con las clases disponibles en el programa, analizando los resultados presentados en los archivos de salida que permitan posteriormente el análisis de Modbus y DNP3, para ello se instaló en Ubuntu 14 la versión 2.35 del *software.*

A pesar de que NS-2 se encuentra descontinuado, siendo NS-3 el sustituto natural, su elección se dio en virtud de que NS-2 dispone de amplia documentación, además de utilizar el trabajo con DNP3 ya implementado en [78], el que fue adaptado a las necesidades de este estudio y se tomó como base para la implementación de Modbus.

Por otro lado, la definición de la estructura de datos del protocolo incluye las reglas que el mensaje debe seguir, como se explica en las secciones 1.2.9 y 1.2.10, Modbus y DNP3 tienen una estructura de tipo maestro - esclavo, pudiendo representarse cada estación por medio de un diagrama de estado. A continuación se presentan los modelos para los dos protocolos.

3.2.1. Modelo Modbus en NS-2

Para el modelamiento de Modbus en NS-2 es fundamental analizar el funcionamiento del protocolo. Modbus opera de acuerdo con el modelo de cliente/servidor (maestro / esclavo). Es decir, el cliente (maestro) envía un mensaje de solicitud (solicitud de servicio) al servidor (esclavo), y el servidor responde con un mensaje de respuesta. Si el servidor no puede procesar una solicitud, en su lugar devolverá un código de función de error (respuesta de excepción) que es el código de función original más 80H (es decir, con el bit más significativo establecido en 1) [52].

A continuación, se describe la forma en que se entabla la comunicación desde el cliente y el servidor. El servidor Modbus espera la llegada de la solicitud del cliente Modbus, después de procesar la solicitud, produce la respuesta adecuada. Si la solicitud no es válida, el servidor Modbus genera una excepción con un código apropiado. En la Figura 3.1 se muestra un diagrama de estado simplificado del servidor Modbus.

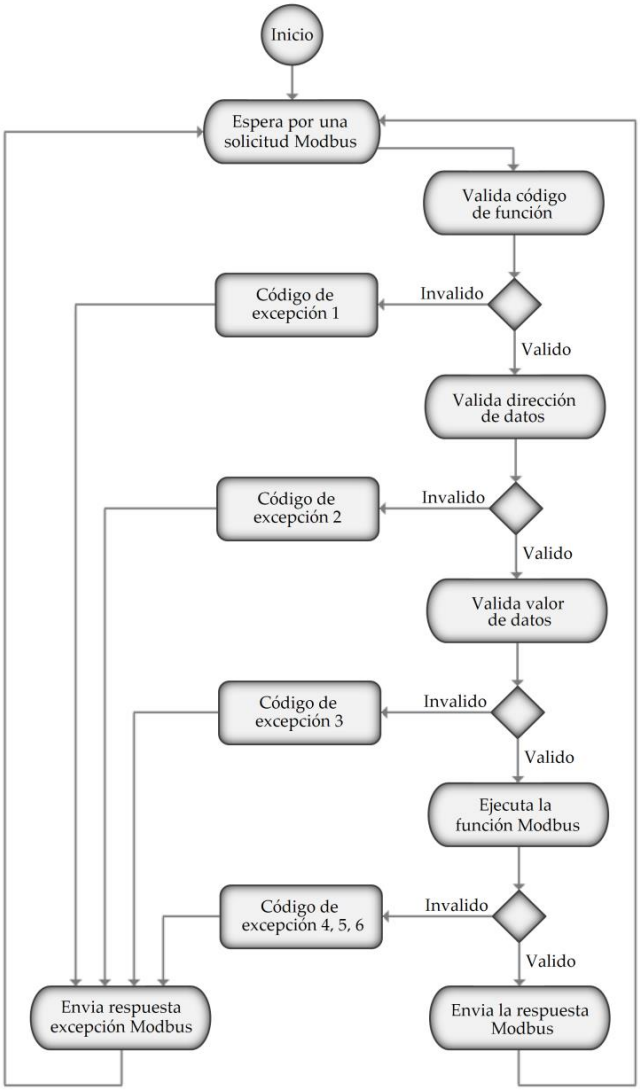


Figura 3.1. Diagrama de estado de transacción Modbus del lado servidor [33].

En la figura 3.2 se presenta el diagrama de estado del cliente y a continuación se describen los procesos realizados en esta estación. El cliente Modbus espera recibir una respuesta válida a la solicitud a menos que ocurra una excepción o un tiempo de espera excedido. Si el servidor no recibe una solicitud debido a una falla de

comunicación o si la solicitud tiene un error de suma de comprobación, entonces no se genera ninguna respuesta y posteriormente el cliente agota el tiempo de espera. Si la solicitud recibida no es válida, el servidor genera y responde con una excepción con un código apropiado. El código de excepción indica si la solicitud tenía una función ilegal (código 01), una dirección de datos ilegal (código 02) o un valor de datos ilegal (código 03). Se envía una excepción (código 05) como acuse de recibo cuando se acepta la solicitud y se requiere un período de tiempo relativamente grande para procesar la solicitud. Además, las excepciones muestran si hay una falla en la comprobación de coherencia de archivos (código 08) o un problema de puerta de enlace encontrado (0A y 0B). De lo contrario, el servidor Modbus responde normalmente [33].

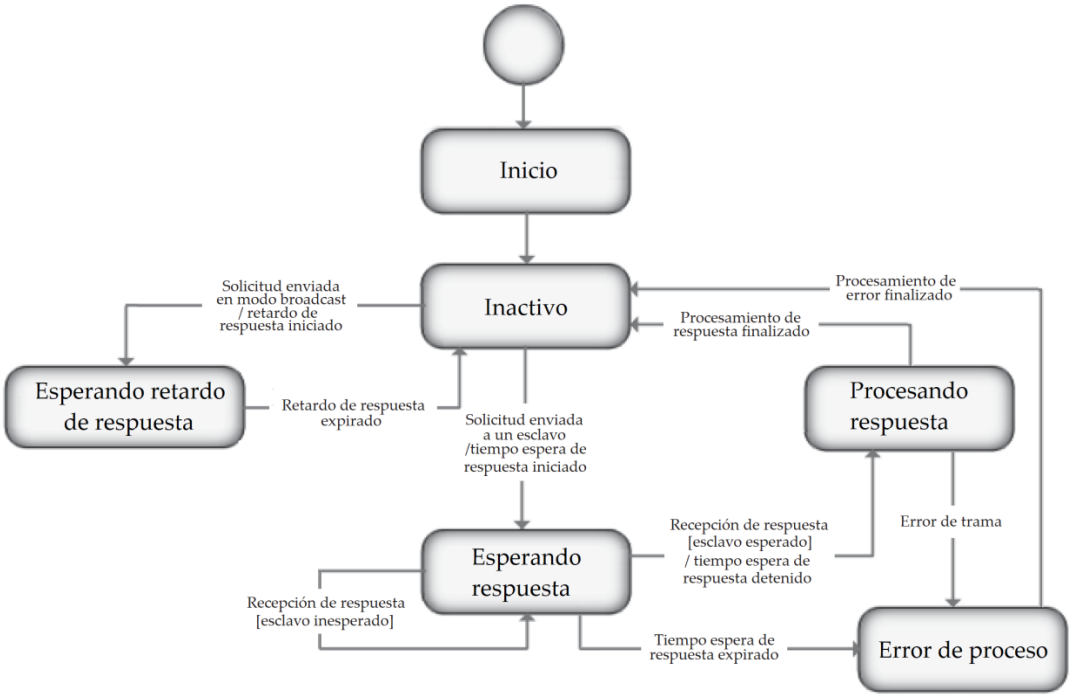


Figura 3.2. Diagrama de estado de transacción Modbus del lado cliente [33].

3.2.2. Modelo DNP3 en NS-2

Para la implementación de DNP3 sobre NS-2 se debe modelar el protocolo y sus interacciones, para este fin se crearon los agentes *Master* y *outstation*. En la figura 3.3 se presenta el diagrama de estados de la *outstation* y a continuación se describe el proceso de comunicación:

Después del establecimiento de la conexión, la *outstation* queda en estado inactivo (Idle) hasta que ocurra un evento que exija el envío de un mensaje al dispositivo maestro mediante el estado *(Send Msg Otcl*). Luego, pasa a esperar una respuesta de la estación maestra (*Waiting Resp from Master*), si el mensaje se recibe, se devuelve al estado inactivo, pero si el tiempo de espera es alcanzado, se devuelve al estado de envío del mensaje Otcl y el bucle se repite.

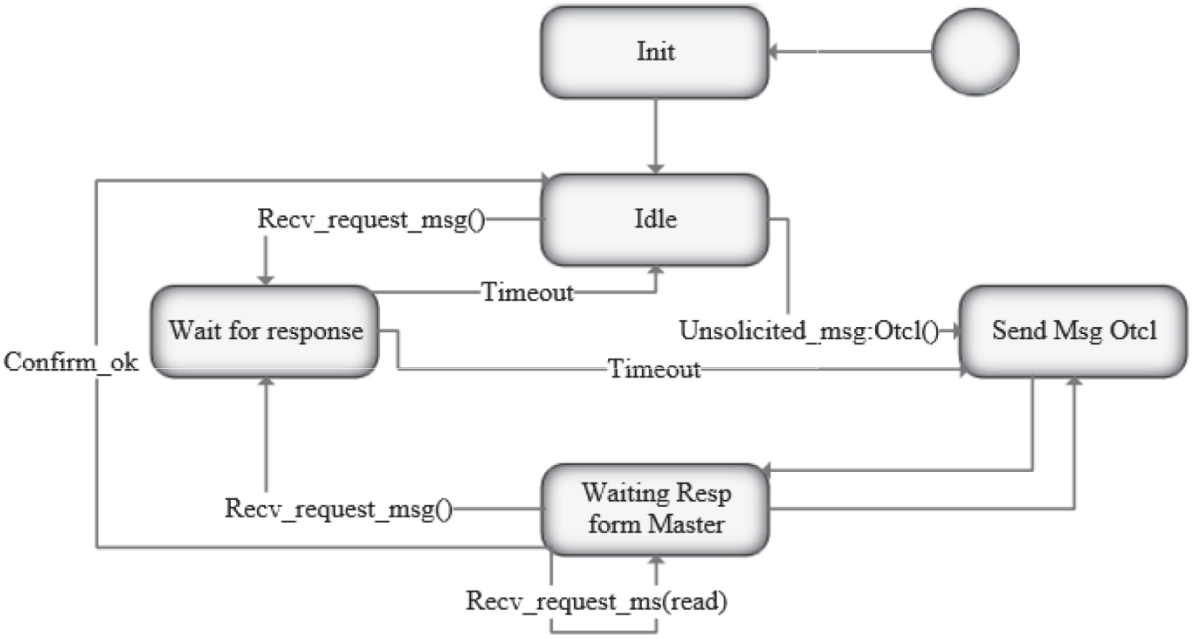


Figura 3.3. Diagrama de estado del dispositivo o*utstation* [79].

En la figura 3.4 se presenta el diagrama de estado del dispositivo maestro, en este el proceso de envío y recepción de mensajes es similar al proceso de los dispositivos o*utstation*. El dispositivo maestro envía una respuesta a la o*utstation* cuando es solicitada o envía una solicitud de mensaje que puede ser de lectura de variables, activación de salidas, quedando a la espera de la respuesta (*Wait for response*) y se reenvía la solicitud cuando se alcanza el tiempo de espera (*timeout*).

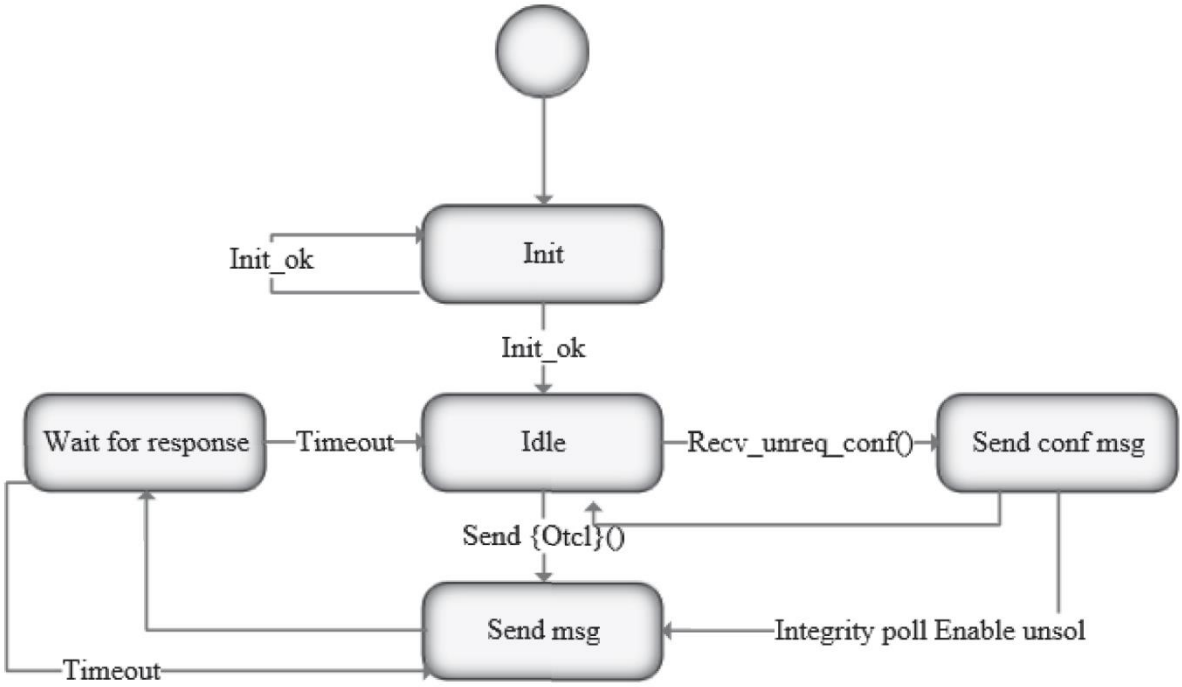


Figura 3.4. Diagrama de estado del dispositivo maestro [79].

3.2.3. Modelos de Canal

Un modelo de canal se utiliza para predecir la potencia de la señal recibida en cada paquete. En la capa física de cada nodo inalámbrico hay un umbral de recepción, cuando se recibe un paquete y la potencia de la señal está por debajo del umbral de recepción, la capa MAC lo marca como error y lo elimina. A continuación se presentan los tres modelos de canal con los que cuenta el simulador ns-2.

Modelo de espacio libre (*Free Space*): este modelo asume una propagación ideal, es decir, las ondas se transmiten con línea de vista directa entre el transmisor y el receptor. En este modelo la potencia recibida depende únicamente de la potencia transmitida, las ganancias de las antenas y la distancia entre el emisor y el receptor. Esto se debe principalmente al hecho de que una onda de radio que se aleja del emisor tiene que cubrir un área mayor. Por lo tanto, la potencia recibida disminuye con el cuadrado de la distancia. El modelo de propagación en espacio libre asume una condición de propagación ideal en la que solo hay una ruta clara de línea de vista entre el transmisor y el receptor. H. T. Friss presentó la ecuación 3.1 para

calcular la potencia de la señal recibida por un nodo en espacio abierto y a cierta distancia del emisor [80].

$$P_r(d) = \frac{(P_t G_t G_r \lambda^2)}{((4\pi)^2 d^2 L)} \quad (3.1)$$

Dónde:

P_t: potencia de la señal transmitida.

G_t y G_r: ganancias de antena del transmisor y el receptor, respectivamente.

d: distancia entre emisor y receptor.

L: constante de pérdidas del sistema.

λ: longitud de onda.

Modelo de dos rayos con reflexión en la tierra (*TwoRay ground reflection*): considerar que la propagación de las ondas se realiza solo por línea de vista directa es una aproximación generalmente muy pobre, debido a que en la mayoría de los casos existen reflexiones con distintos objetos que se encuentran entre el emisor al receptor. En este modelo se asume que la energía recibida es la suma de las energías en línea de vista directa y de una ruta de reflexión en el suelo entre el emisor y el receptor. Este modelo proporciona una predicción más precisa que el modelo de espacio libre [81]. La potencia recibida se calcula con la ecuación 3.2.

$$P_r(d) = \frac{(P_t G_t G_r h_t^2 h_r^2)}{(d^4 L)} \quad (3.2)$$

Dónde:

P_t: potencia de la señal transmitida.

G_t y G_r: ganancias de antena del transmisor y el receptor, respectivamente.

h_t y h_r: altura de las antenas del transmisor y el receptor, respectivamente.

d: distancia entre emisor y receptor.

L: constante de pérdidas del sistema.

λ: longitud de onda.

En la ecuación 3.2 se observa una pérdida de potencia más rápida en relación a la ecuación 3.1 a medida que aumenta la distancia. Sin embargo, este modelo no da buenos resultados en distancias cortas debido a la oscilación causada por la combinación constructiva y destructiva de los dos rayos.

Modelo de sombras o Multitrayecto: los modelos anteriores predicen la potencia recibida como una función determinista con respecto a la distancia. Ambos representan el área cubierta por el emisor como una esfera perfecta. Pero en realidad la potencia recibida es una variable aleatoria a causa de la propagación de las señales multitrayecto, lo que produce un desvanecimiento en la señal. De hecho, los modelos anteriores dan como resultado la potencia media recibida a cierta distancia del emisor. Un modelo más general y ampliamente usado es el denominado modelo de sombras o "*shadowing*" [80]. Este modelo consta de dos partes, la primera se conoce como modelo de pérdida de ruta, que predice la potencia media recibida a una distancia d, indicada por $P_r(d)$. Además, utiliza una distancia de acercamiento d_0 como referencia. La ecuación 3.3 indica como calcular $P_r(d)$ con relación a $P_r(d_0)$.

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) \quad (3.3)$$

Dónde:

β: se denomina exponente de pérdida de trayectoria y generalmente se determina empíricamente mediante mediciones en campo. En la tabla 3.1 se presentan algunos valores típicos para β. Los valores más grandes corresponden a más obstrucciones y, por lo tanto, a una disminución más rápida de la potencia media recibida a medida que aumenta la distancia.

Tabla 3.1. Valores típicos del exponente de pérdida de trayectoria β [80].

Ambiente		$\boldsymbol{\beta}$
Aire Libre	Espacio Libre	2
	Sombreado área urbana	2.7 - 5
En construcciones	Línea de vista	1.6 - 1.8

	Obstruido	4 - 6

La segunda parte del modelo refleja la variación de la potencia recibida a cierta distancia. El modelo general está representado por la ecuación 3.4.

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{dB} \quad (3.4)$$

Donde X_{dB} es una variable aleatoria gaussiana con media cero y desviación estándar σ_{dB}. El término σ_{dB} se denomina desviación de sombreado y también se obtiene por medición. La Tabla 3.2 muestra algunos valores típicos de σ_{dB}.

Tabla 3.2. Valores típicos para la desviación de sombreado σ_{dB}[80].

Ambiente	σ_{dB}
Aire Libre	4 - 12
Oficina, divisiones duras	7
Oficina, divisiones suaves	9.6
Industrial, línea de vista	3 - 6
Industrial, obstruido	6.8

3.2.4. Métricas de rendimiento

Con el objetivo de producir resultados relevantes en el estudio de rendimiento de los protocolos, se escogió el *delay* y el *throughput,* dos métricas bastante utilizadas para analizar el desempeño en redes de comunicaciones y además porque son las demandadas para este tipo de aplicación [82]. A continuación se presenta una descripción de estas métricas de rendimiento.

***Delay*.** Para efectos de este estudio el *delay* se mide como el Tiempo de Ida y Vuelta (RTT, *Round Trip Time*). El RTT es el tiempo promedio transcurrido desde el momento en que se envía una solicitud desde un cliente al servidor hasta que el cliente recibe la respuesta relacionada [78]. El *delay* se mide en milisegundos y se calcula como se expresa en la ecuación 3.5.

$$delay(ms) = tiempo\ de\ llegada - tiempo\ de\ envío \quad (3.5)$$

Throughput. Es la tasa efectiva de bits que se transmiten a un terminal por unidad de tiempo. Se define como la razón entre la cantidad de datos transferida de un nodo origen a un nodo de destino por unidad de tiempo. Para el análisis de los protocolos, el *throughput* se mide en bis por segundo (bps) pudiendo también medirse en paquetes por segundo (pps) [82]. En la ecuación 3.5 se expresa matemáticamente el cálculo de esta métrica.

$$Throughput\ (bps) = \frac{Número\ de\ bits\ entregados}{Tiempo\ (s)} \quad (3.6)$$

3.3. Implementación del modelo

Para el desarrollo del trabajo fue necesario crear dos parches generadores de tráfico, uno para Modbus y otro para DNP3. Para ello, el modelo del protocolo DNP3 desarrollado en [78] se adaptó para ser utilizado en el presente trabajo, así como se realizó en [38, 62, 79]. Para ejecutar nuevos parches es necesario realizar algunos cambios en el núcleo de NS-2, con la inclusión de parámetros en algunos de los archivos existentes, así como con la inclusión de nuevos archivos. A través de los generadores de tráfico implementados se pueden construir escenarios de simulación que permitan evaluar el desempeño de Modbus y DNP3 en aplicaciones de *microgrid*.

Por otro lado, con base en el diagrama de estados para el cliente y servidor Modbus mostrados en las figuras 3.1 y 3.2, y para el maestro y *outstation* DNP3 mostrados en las figuras 3.3 y 3.4, respectivamente, se elaboraron las clases necesarias para la implementación de los protocolos. En [78], se presenta un diagrama de clases necesarias para la implementación de DNP3. Según estas clases se crearon las necesarias para la implementación de Modbus y DNP3. En la Figura 3.5 se presentan las clases que integran DNP3, las cuales se detallan a continuación:

- TcpAppMod: elaborada en [78] a partir de una modificación de la clase TcpApp existente en el NS-2. Se creó una nueva clase en lugar de modificar la existente para no causar cambios en la distribución oficial del programa. Esta clase simula el envío de datos en la capa de aplicación, a través de una secuencia de
- tipo cola de datos, cuya información del paquete se almacena para el envío al nodo de destino.
- TimerHandler: se utiliza para implementar el temporizador de estado de tiempo de espera.
- Dnp3App: se implementó desde una clase de capa de aplicación también existente en el programa, la clase HTTP, que permite trabajar con varias conexiones por parte de un agente a nivel de aplicación. De esta clase, otras dos heredan sus atributos, las clases dnp3Appmaster y dnp3Appclient, la primera para la estación maestra y la segunda para estación *outstation*. El objetivo es establecer las funciones constructoras de las estaciones maestras con sus temporizadores y su tabla hash de conexiones y la inicialización de la estación o*utstation*.
- dnp3Appmaster: utiliza tres clases, stackid que almacena una cola de datos que mantiene el valor de identificación de cada estación remota, stackdata que almacena la estructura de datos de cada mensaje, en caso de que sea necesaria una retransmisión, y dnp3AppTimermaster, que hereda los atributos de TimerHandler, para implementar el temporizador de estado de espera.
- dnp3Appclient: utiliza solo la clase dnp3AppTimerclient, cuyo objetivo es el mismo de la temporización de la clase maestra. Su activación se realiza en el momento en que el mensaje es enviado, si el tiempo se alcanza hace que se haga el reenvío de mensajes a la estación maestra.

En las clases dnp3Appmaster y dnp3Appclient, se heredan algunos métodos de la clase Dnp3App, tales como command y process_data. Además de estos, otros métodos se implementan en sus clases correspondientes, como el setmsg y resend, cuyo detalle se realiza a continuación:

- Función command: responsable de generar los mensajes.
- Función process_data: recoge los datos en un evento de recepción, los analiza y determina la próxima operación del agente.

- Función setmsg: ejecuta los procesos definidos en las capas del protocolo como fragmentación, cálculo del tamaño de paquetes y activación del temporizador para retransmisión.
- Función resend: realiza el proceso de transmisión de mensajes cuando es necesario.

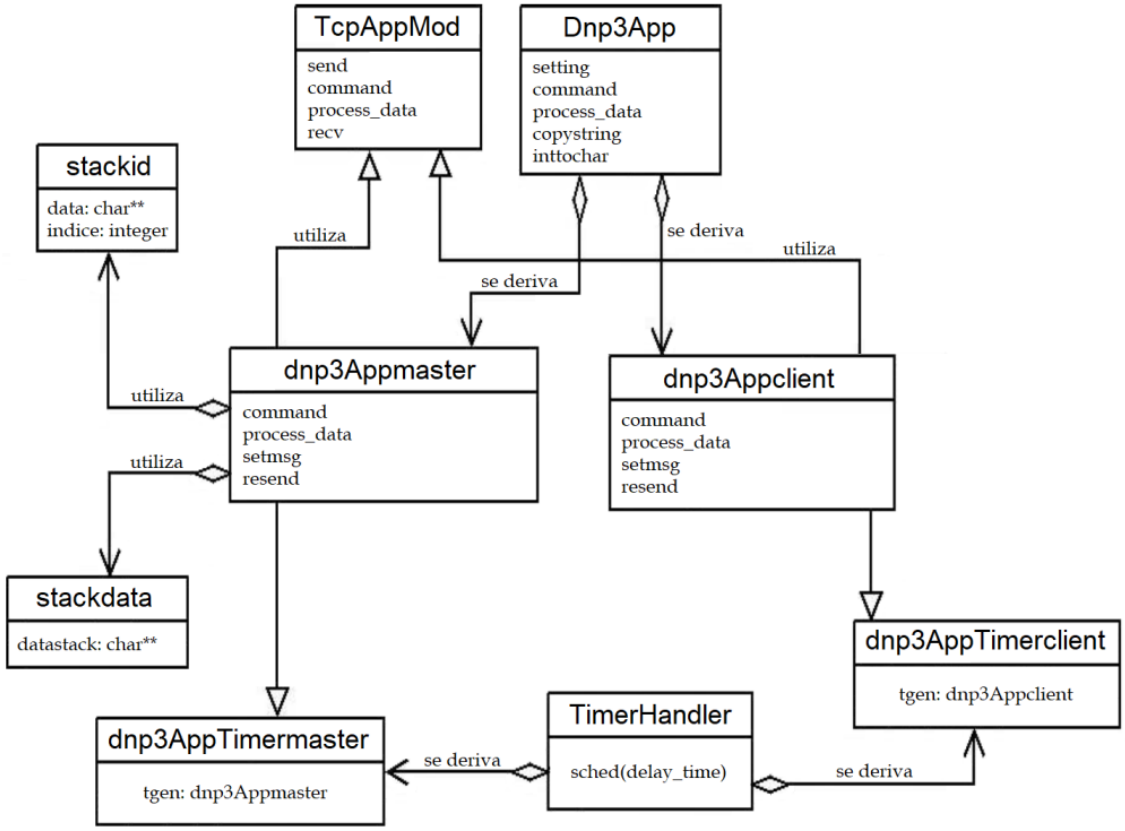


Figura 3.5. Diagrama de clases DNP3 [78].

Después de elaborar los diagramas de estados y de la descripción del diagrama de clases, se crearon en el núcleo de NS-2 dos carpetas, “Modbus” y “DNP3”. Cada una contiene cuatro archivos principales, necesarios para la implementación de los protocolos. Los archivos fueron creados en lenguaje C++ y se indican a continuación:

- tcpappmodb/tcpappmod: tienen por función generar el tráfico de datos y hacer la simulación de las operaciones básicas, haciendo el encapsulado del agente Modbus y DNP3 respectivamente.
- modbapp/dnp3app: definen las clases modbApp y dnp3App, que proporcionan los atributos para las siguientes.

- modbAppmaster/dnp3Appmaster: definen las funciones para las estaciones cliente/maestro, respectivamente.
- modbAppclient/dnp3Appclient: definen las funciones para las estaciones servidor/outstation, respectivamente.

3.3.1. Cambios en los archivos del núcleo de NS-2

Algunos archivos del núcleo de NS-2 necesitaron ser modificados, con el objetivo de informar a NS-2 que existen dos nuevos protocolos. Los archivos modificados se describen a continuación:

- packet.h: en este archivo ubicado en /ns-allinone/ns-2.35/common/, se informa de la existencia de dos nuevos tipos de paquetes, como se muestra en la figura 3.6.

```
     // M-DART packets
static const packet_t PT_MDART = 72;
     // MODB packets
static const packet_t PT_MODB = 73;
     // DNP3 packets
static const packet_t PT_DNP3 = 74;
     // insert new packet types here
static packet_t PT_NTYPE = 75;// This MUST be the LAST one
```

Figura 3.6. Asignación de los paquetes Modbus y DNP3 en packet.h.

- ns-process.h: en este archivo ubicado en /ns-allinone/ns-2.35/common/, se informa de la existencia de las ADU de tipo Modbus y DNP3, como se muestra en la figura 3.7.

```
     // Modbus ADU
     MODB,
     // DNP3 ADU
     DNP3,
     // Last ADU
     ADU_LAST
```

Figura 3.7. Asignación de las ADU Modbus y DNP3 en ns-process.h.

- ns-default.tcl: este archivo ubicado en /ns-allinone/ns-2.35/tcl/lib/, es utilizado para especificar valores predeterminados. Para Modbus y DNP3 se definió el tiempo de retransmisión retrytimer a 1 segundo y la probabilidad de errores proberror a cero, como se observa en la figura 3.8.

```
MODBmaster set retrytimer_ 1
MODBmaster set proberror_ 0
MODBclient set retrytimer_ 1
MODBclient set proberror_ 0

DNP3master set retrytimer_ 1
DNP3master set proberror_ 0
DNP3client set retrytimer_ 1
DNP3client set proberror_ 0
```

Figura 3.8. Modificaciones del archivo ns-default.tcl para Modbus y DNP3.

- ns-packet.tcl: en este archivo disponible en /ns-allinone/ns-2.35/tcl/lib/, se adiciona a Modbus y DNP3 a la lista de protocolos de capa de aplicación, como se muestra en la figura 3.9.

```
# Application-Layer Protocols:
     Message # a protocol to carry text messages
     Ping    # Ping
     PBC     # PBC
     MODB    # MODB
     DNP3    # DNP3
```

Figura 3.9. Adición de Modbus y DNP3 a la lista de protocolos de capa de aplicación.

- Makefile: cuando se realizan cambios en el núcleo del programa, este se debe compilar nuevamente. Además, las nuevas clases implementadas para Modbus y DNP3 también necesitan ser compiladas, para que sean generados los archivos objeto .o, cuya existencia debe ser indicada en el archivo Makefile, como se muestra en la figura 3.10.

```
apps/pbc.o \
dnp3/dnp3App.o dnp3/dnp3Appclient.o dnp3/dnp3Appmaster.o
dnp3/tcpappmod.o \
modbus/modbApp.o modbus/modbAppclient.o
modbus/modbAppmaster.o modbus/tcpappmodb.o \
$(OBJ_STL)
```

Figura 3.10. Ubicación de los archivos objeto para Modbus y DNP3.

Para modelar y simular una red en NS-2 es necesario escribir un *script* en lenguaje OTcl, que será ejecutado, generando las salidas especificadas. Para la creación de un *script* en NS-2 se recomiendan las etapas que se presentan en la figura 3.11 [38]:

```
#Definición del objeto Simulador
set ns [new Simulator]

#Definición de los archivos de salida (Nam, trace file)
set nf [open out.nam w]
$ns namtrace-all $nf
set f [open out.tr w]
$ns trace-all $f

#Definición del procedimiento de finalización de la
simulación
proc fim {} {
global ns nf f
$ns flush-trace
close $f
close $nf
exec nam out.nam &
exit 0
}

#Cantidad de nodos del sistema
set n1 [$ns node]
set n2 [$ns node]
```

```
#Enlaces de nodos del sistema
$ns duplex-link $n1 $n2 10Mb 1ms DropTail

#Protocolos a nivel de transporte
set tcp1 [new Agent/TCP/SimpleTcp]
set tcp2 [new Agent/TCP/SimpleTcp]
$ns attach-agent $n1 $tcp1
$ns attach-agent $n2 $tcp2
$ns connect $tcp1 $tcp2

#Aplicaciones, creación de generadores de tráfico y
#conexión con los agentes de la capa de transporte
set encap [new Application/Tcpmod $tcp1]
set encap2 [new Application/Tcpmod $tcp2]
$encap connect $encap2
set master [new DNP3master]
set client [new DNP3client 1]
$master connect $encap
$client connect $encap2

#Programación de los eventos de la simulación.
$ns at 1.0 "$master read client"
$ns at 2.0 "stop"

#Ejecutar la simulación.
$ns run
```

Figura 3.11. Etapas básicas de archivo .tcl en NS-2.

3.4. Verificación del modelo

Para la verificación del modelo, inicialmente se crea el *script* que contiene el código donde se configura el canal de comunicaciones, se definen los nodos, sus interacciones, entre otros. Luego, se ejecuta el programa generando los archivos

.nam y .tr. El primero permite apreciar visualmente la interacción entre nodos y el archivo de análisis *trace* (.tr) contiene el registro de los eventos que pueden procesarse para conocer el funcionamiento y rendimiento de la red. En la figura 3.12 se presenta el proceso general para realizar una simulación en NS-2.

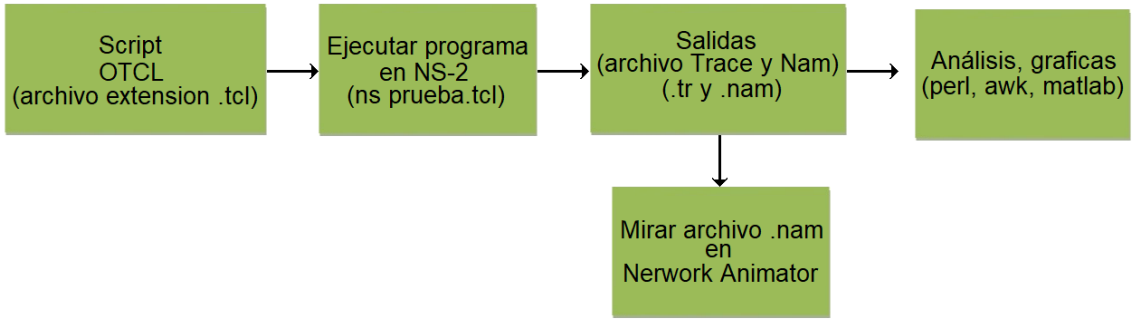


Figura 3.12. Modelo de simulación en NS-2 [83].

Por otra parte, el escenario para la verificación consta de dos nodos conectados vía Wi-Fi, uno representa al SCADA y el otro a un medidor inteligente. Con esta simple configuración es posible verificar que el intercambio de información se realice según lo esperado. Siguiendo el diagrama que se presenta en la figura 3.12, inicialmente se creó el *script.* En la cabecera se definen parámetros del canal de comunicación como: tipo de canal, modelo de propagación, frecuencia de operación y dimensiones del área de simulación. En la figura 3.13 se presenta el código Otcl y en la tabla 3.3 los parámetros de configuración del canal.

```
Phy/WirelessPhy set freq_ 2.4e+9   ;# frecuencia de operación
set val(chan)  Channel/WirelessChannel ;# channel type
set val(prop)  Propagation/Shadowing   ;# radio-propagation
set val(mac)   Mac/802_11              ;# MAC type
set val(ant)   Antenna/OmniAntenna     ;# antenna model
set val(xx)    350                ;# X dimension of topography
set val(yy)    300                ;# Y dimension of topography
```

Figura 3.13. Configuración del canal de comunicación en NS-2.

Tabla 3.3. Parámetros de la red modelada.

Parámetro	Valor
Canal	WirelessChannel
Propagación	Shadowing
Interfaz de red	Wireless Phy

Capa MAC	802_11
Frecuencia	2.4 GHz
Tipo de cola	DropTail
Capa de enlace	LL
Modelo de antena	OmniAntenna
Número de nodos	2
Protocolo de enrutamiento	DumbAgent
Área de cobertura (XxY)	350 m x 300 m
Tasa de transmisión	11 Mb

El siguiente paso es definir el objeto de simulación y crear los archivos de salida .nam y .tr como se muestra en la figura 3.11. Además, en la figura 3.14 se presenta la configuración de la posición de los nodos representada en un plano XY y se establecen algunos atributos de los nodos como: color, etiqueta y tamaño.

```
# Coordenadas iniciales (X, Y, Z = 0) para nodos móviles
$node_(0) set X_ 180.0
$node_(0) set Y_ 50.0
$node_(0) set Z_ 0.0
$node_(0) color black

$node_(1) set X_ 180.0
$node_(1) set Y_ 200.0
$node_(1) set Z_ 0.0
$node_(1) color black

$ns initial_node_pos $node_(0) 50 ;#tamaño de nodos
$ns initial_node_pos $node_(1) 50

$ns at 0.0 "$node_(0) color yellow" ;#color de nodos
$ns at 0.0 "$node_(0) label Maestro" ;#texto de nodos
$ns at 0.0 "$node_(1) color blue"
$ns at 0.0 "$node_(1) label Outstation"
```

Figura 3.14. Configuración de posición y atributos en nodos.

Luego, se definen los agentes TCP, se crean la estación maestro y cliente y sus conexiones y finalmente se establecen los eventos a simular, en este caso, el maestro lee información del cliente en el segundo cero, uno y dos; en el segundo tres

finaliza la simulación. En la figura 3.15 se presentan los pasos descritos anteriormente.

```
#Configurar flujo de tráfico entre nodos
set tcp1 [new Agent/TCP/SimpleTcp]
set tcp2 [new Agent/TCP/SimpleTcp]

$ns attach-agent $node_(0) $tcp1
$ns attach-agent $node_(1) $tcp2
$ns connect $tcp1 $tcp2

set encap [new Application/TcpAppmod $tcp1]
set encap2 [new Application/TcpAppmod $tcp2]
$encap connect $encap2

# Crea estacion maestro
set master [new DNP3master]
# Crea estacion Esclavo
set client [new DNP3client 1]
$master connect $client $encap
$client connect $master $encap2

$ns at 0.0 "$master read $client"
$ns at 1.0 "$master read $client"
$ns at 2.0 "$master read $client"

$ns at 3.0 "stop"

$ns run
```

Figura 3.15. Creación de las estaciones maestro y cliente DNP3.

Una vez se ejecuta el programa se crean los archivos .nam y .tr, en la figura 3.16 se muestra la simulación en NAM para DNP3, en esta ilustración es posible visualizar que los nodos maestro y *outstation* están dentro del alcance y los círculos representan el tráfico generado así: el más grande representa la trama inicial en la comunicación e indica la solicitud por parte del maestro, el circulo mediano corresponde a la respuesta del *outstation* y el circulo pequeño a la confirmación del maestro.

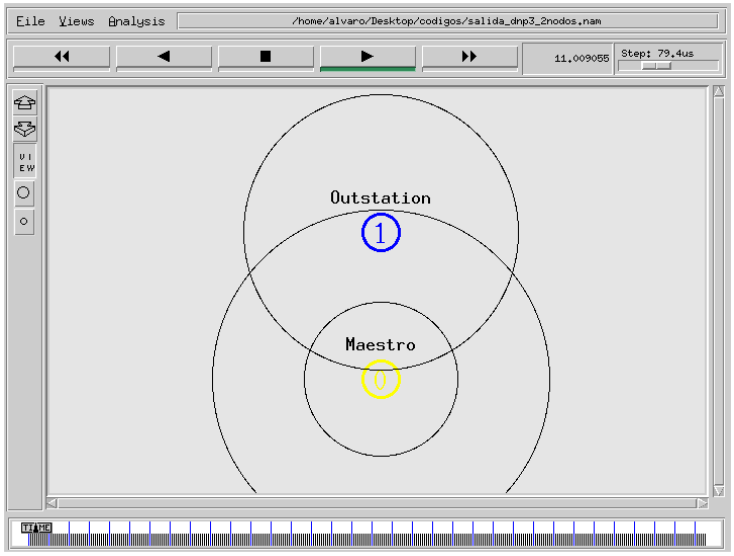


Figura 3.16. Estación *outstation* enviando paquetes a la estación maestro.

Por otro lado, cada fila del fichero *trace* corresponde a un evento diferente como un paquete enviado, un paquete recibido, un paquete perdido, entre otros, en la sección 1.2.13 se presentó una descripción de cada uno de los campos de este archivo. En la figura 3.17 se presenta el archivo *trace* de la simulación de DNP3, en la prueba se envió una solicitud de lectura cada segundo iniciando en el instante 0.0s con un tamaño de trama aleatorio de 62 bytes, la *outstation* recibió el mensaje en el instante 0.004891667s y de inmediato respondió con una trama de 85 bytes de longitud, por su parte el maestro recibió la respuesta en el instante 0.007317667s que finalmente transmite una trama de confirmación con una longitud fija de 50 bytes recibida en el instante 0.009343667s, este análisis se puede realizar para las otras dos solicitudes de lectura.

```
s 0.000000000 _0_ AGT  --- 0 DNP3 62 [0 0 0 0] ------- [0:0 1:0 32 0]
r 0.000000000 _0_ RTR  --- 0 DNP3 62 [0 0 0 0] ------- [0:0 1:0 32 0]
s 0.000000000 _0_ RTR  --- 0 DNP3 62 [0 0 0 0] ------- [0:0 1:0 32 0]
r 0.004891667 _1_ AGT  --- 0 DNP3 62 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 0.004891667 _1_ AGT  --- 1 DNP3 85 [0 0 0 0] ------- [1:0 0:0 32 0]
r 0.004891667 _1_ RTR  --- 1 DNP3 85 [0 0 0 0] ------- [1:0 0:0 32 0]
s 0.004891667 _1_ RTR  --- 1 DNP3 85 [0 0 0 0] ------- [1:0 0:0 32 0]
r 0.007317667 _0_ AGT  --- 1 DNP3 85 [13a 0 1 800] ------- [1:0 0:0 32 0]
s 0.007317667 _0_ AGT  --- 2 DNP3 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 0.007317667 _0_ RTR  --- 2 DNP3 50 [0 0 0 0] ------- [0:0 1:0 32 0]
s 0.007317667 _0_ RTR  --- 2 DNP3 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 0.009343667 _1_ AGT  --- 2 DNP3 50 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 1.000000000 _0_ AGT  --- 3 DNP3 71 [0 0 0 0] ------- [0:0 1:0 32 0]
r 1.000000000 _0_ RTR  --- 3 DNP3 71 [0 0 0 0] ------- [0:0 1:0 32 0]
s 1.000000000 _0_ RTR  --- 3 DNP3 71 [0 0 0 0] ------- [0:0 1:0 32 0]
r 1.002358000 _1_ AGT  --- 3 DNP3 71 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 1.002358000 _1_ AGT  --- 4 DNP3 83 [0 0 0 0] ------- [1:0 0:0 32 0]
r 1.002358000 _1_ RTR  --- 4 DNP3 83 [0 0 0 0] ------- [1:0 0:0 32 0]
s 1.002358000 _1_ RTR  --- 4 DNP3 83 [0 0 0 0] ------- [1:0 0:0 32 0]
r 1.004888000 _0_ AGT  --- 4 DNP3 83 [13a 0 1 800] ------- [1:0 0:0 32 0]
s 1.004888000 _0_ AGT  --- 5 DNP3 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 1.004888000 _0_ RTR  --- 5 DNP3 50 [0 0 0 0] ------- [0:0 1:0 32 0]
s 1.004888000 _0_ RTR  --- 5 DNP3 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 1.006954000 _1_ AGT  --- 5 DNP3 50 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 2.000000000 _0_ AGT  --- 6 DNP3 67 [0 0 0 0] ------- [0:0 1:0 32 0]
r 2.000000000 _0_ RTR  --- 6 DNP3 67 [0 0 0 0] ------- [0:0 1:0 32 0]
s 2.000000000 _0_ RTR  --- 6 DNP3 67 [0 0 0 0] ------- [0:0 1:0 32 0]
r 2.002018000 _1_ AGT  --- 6 DNP3 67 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 2.002018000 _1_ AGT  --- 7 DNP3 78 [0 0 0 0] ------- [1:0 0:0 32 0]
r 2.002018000 _1_ RTR  --- 7 DNP3 78 [0 0 0 0] ------- [1:0 0:0 32 0]
s 2.002018000 _1_ RTR  --- 7 DNP3 78 [0 0 0 0] ------- [1:0 0:0 32 0]
r 2.004168000 _0_ AGT  --- 7 DNP3 78 [13a 0 1 800] ------- [1:0 0:0 32 0]
s 2.004168000 _0_ AGT  --- 8 DNP3 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 2.004168000 _0_ RTR  --- 8 DNP3 50 [0 0 0 0] ------- [0:0 1:0 32 0]
s 2.004168000 _0_ RTR  --- 8 DNP3 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 2.006294000 _1_ AGT  --- 8 DNP3 50 [13a 1 0 800] ------- [0:0 1:0 32 0]
```

Figura 3.17. Resultados en el archivo *trace* para DNP3.

Por su parte, en las figuras 3.18 y 3.19 se presenta la simulación en NAM y el archivo *trace* para Modbus, respectivamente. Para este caso, el código es similar al creado para DNP3, el único cambio es que en este caso se crea un maestro y un cliente Modbus. Además, el análisis es similar, donde se observa que la transmisión de datos se ejecuta según lo diseñado. En comparación con los datos DNP3, se observa en los archivos *trace* que los datos Modbus tardan menos tiempo en llegar al cliente, esta característica se estudiará en el siguiente capítulo.

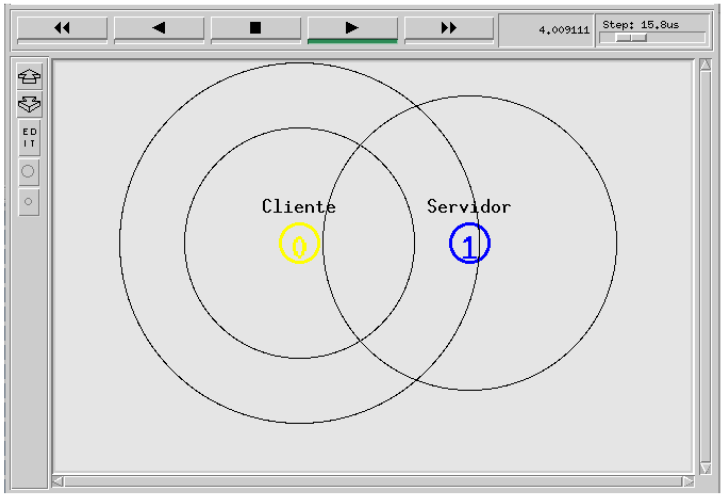


Figura 3.18. Estación servidor enviando paquetes a la estación cliente.

```
s 0.000000000 _0_ AGT  --- 0 MODdB 68 [0 0 0 0] ------- [0:0 1:0 32 0]
r 0.000000000 _0_ RTR  --- 0 MODdB 68 [0 0 0 0] ------- [0:0 1:0 32 0]
s 0.000000000 _0_ RTR  --- 0 MODdB 68 [0 0 0 0] ------- [0:0 1:0 32 0]
r 0.004844133 _1_ AGT  --- 0 MODdB 68 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 0.004844133 _1_ AGT  --- 1 MODdB 67 [0 0 0 0] ------- [1:0 0:0 32 0]
r 0.004844133 _1_ RTR  --- 1 MODdB 67 [0 0 0 0] ------- [1:0 0:0 32 0]
s 0.004844133 _1_ RTR  --- 1 MODdB 67 [0 0 0 0] ------- [1:0 0:0 32 0]
r 0.007126333 _0_ AGT  --- 1 MODdB 67 [13a 0 1 800] ------- [1:0 0:0 32 0]
s 0.007126333 _0_ AGT  --- 2 MODdB 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 0.007126333 _0_ RTR  --- 2 MODdB 50 [0 0 0 0] ------- [0:0 1:0 32 0]
s 0.007126333 _0_ RTR  --- 2 MODdB 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 0.009152533 _1_ AGT  --- 2 MODdB 50 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 1.000000000 _0_ AGT  --- 3 MODdB 76 [0 0 0 0] ------- [0:0 1:0 32 0]
r 1.000000000 _0_ RTR  --- 3 MODdB 76 [0 0 0 0] ------- [0:0 1:0 32 0]
s 1.000000000 _0_ RTR  --- 3 MODdB 76 [0 0 0 0] ------- [0:0 1:0 32 0]
r 1.002310200 _1_ AGT  --- 3 MODdB 76 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 1.002310200 _1_ AGT  --- 4 MODdB 73 [0 0 0 0] ------- [1:0 0:0 32 0]
r 1.002310200 _1_ RTR  --- 4 MODdB 73 [0 0 0 0] ------- [1:0 0:0 32 0]
s 1.002310200 _1_ RTR  --- 4 MODdB 73 [0 0 0 0] ------- [1:0 0:0 32 0]
r 1.004712400 _0_ AGT  --- 4 MODdB 73 [13a 0 1 800] ------- [1:0 0:0 32 0]
s 1.004712400 _0_ AGT  --- 5 MODdB 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 1.004712400 _0_ RTR  --- 5 MODdB 50 [0 0 0 0] ------- [0:0 1:0 32 0]
s 1.004712400 _0_ RTR  --- 5 MODdB 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 1.006778600 _1_ AGT  --- 5 MODdB 50 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 2.000000000 _0_ AGT  --- 6 MODdB 66 [0 0 0 0] ------- [0:0 1:0 32 0]
r 2.000000000 _0_ RTR  --- 6 MODdB 66 [0 0 0 0] ------- [0:0 1:0 32 0]
s 2.000000000 _0_ RTR  --- 6 MODdB 66 [0 0 0 0] ------- [0:0 1:0 32 0]
r 2.001970200 _1_ AGT  --- 6 MODdB 66 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 2.001970200 _1_ AGT  --- 7 MODdB 111 [0 0 0 0] ------- [1:0 0:0 32 0]
r 2.001970200 _1_ RTR  --- 7 MODdB 111 [0 0 0 0] ------- [1:0 0:0 32 0]
s 2.001970200 _1_ RTR  --- 7 MODdB 111 [0 0 0 0] ------- [1:0 0:0 32 0]
r 2.004384400 _0_ AGT  --- 7 MODdB 111 [13a 0 1 800] ------- [1:0 0:0 32 0]
s 2.004384400 _0_ AGT  --- 8 MODdB 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 2.004384400 _0_ RTR  --- 8 MODdB 50 [0 0 0 0] ------- [0:0 1:0 32 0]
s 2.004384400 _0_ RTR  --- 8 MODdB 50 [0 0 0 0] ------- [0:0 1:0 32 0]
r 2.006510600 _1_ AGT  --- 8 MODdB 50 [13a 1 0 800] ------- [0:0 1:0 32 0]
```

Figura 3.19. Resultados en el archivo *trace* para Modbus.

Como se observa los resultados fueron positivos en la medida que la interacción entre los dos nodos se realizó según lo diseñado. En el Apéndice A y B se presentan los códigos en NS-2 para las pruebas realizadas para DNP3 y Modbus, respectivamente.

3.5. Validación del modelo

Existen diferentes técnicas para comparar datos simulados y reales, algunas técnicas subjetivas como la comparación gráfica es adecuada en primera instancia para realizar un primer juicio sobre la precisión de la simulación. Esta comparación es una técnica de validez aparente, pero que no requiere de condiciones como datos independientes o satisfacer ningún requisito estadístico. Por su parte, las pruebas objetivas comparan los modelos usando pruebas estadísticas, por ejemplo, pruebas de hipótesis para comparación de medias, varianzas o distribuciones de las salidas de un modelo y un sistema, con el fin de determinar si el comportamiento del modelo de simulación tiene un rango aceptable de precisión [67].

Por otro lado, es importante calcular el tamaño de la muestra que permita determinar que los datos obtenidos sean representativos, para ello se consideró un intervalo de confianza del 95% [84]. Para asegurar este valor, inicialmente se realizaron 10 pruebas de campo piloto con el fin de encontrar la media μ y la desviación estándar σ del *delay*, cuyos resultados se presentan en la tabla 3.4.

Tabla 3.4. Parámetros iniciales para el análisis estadístico.

Parámetro	Valor
Número de simulaciones	10
Delay medio (Modbus)	20.738 ms
Delay medio (DNP3)	27.133 ms
Desviación estándar (Modbus)	3.684 ms
Desviación estándar (DNP3)	3.765 ms

A partir de los resultados de la tabla 3.4 y considerando una distribución t de Student se determina el número de muestras necesarias utilizando la ecuación 3.7.

$$n = \left(\frac{Z_{\alpha/2}\sigma}{e}\right)^2 \qquad (3.7)$$

Dónde:

n: número de elementos de la muestra.
$Z_{\alpha/2}$: coeficiente de confianza.
σ: desviación estándar de la muestra.
e: representa el error.

Finalmente, los resultados se presentan en la tabla 3.5 indicando que para Modbus y DNP3 se requiere de un mínimo de 28 muestras, para el estudio se tomaron 30 muestras.

Tabla 3.5. Parámetros calculados para el análisis estadístico.

Parámetro	Valor
Error (Modbus)	1.373
Error (DNP3)	1.403
Número de muestras (Modbus)	27.64
Número de muestras (DNP3)	27.66

Por su parte, para analizar la información de la red de comunicaciones real se utilizó Wireshark, este software es un analizador de paquetes que captura todo el tráfico que viaja por la red [85]. Además, soporta los protocolos Modbus y DNP3, por lo que entrega información detallada de dichas tramas. En la figura 3.20 se muestra una imagen de Wireshark presentando información de las tramas Modbus capturadas. La información se detalla a continuación:

1. **No:** número de trama capturada.
2. ***Time:*** tiempo en el que ocurrió el evento.
3. ***Source:*** dirección IP del dispositivo que generó la trama.
4. ***Destination:*** dirección IP del dispositivo destino.
5. ***Protocol:*** describe el protocolo utilizado en la trama.
6. ***Length:*** número de Bytes que componen la trama.

7. ***Info:*** describe la naturaleza de la trama.
8. Descripción de la trama.

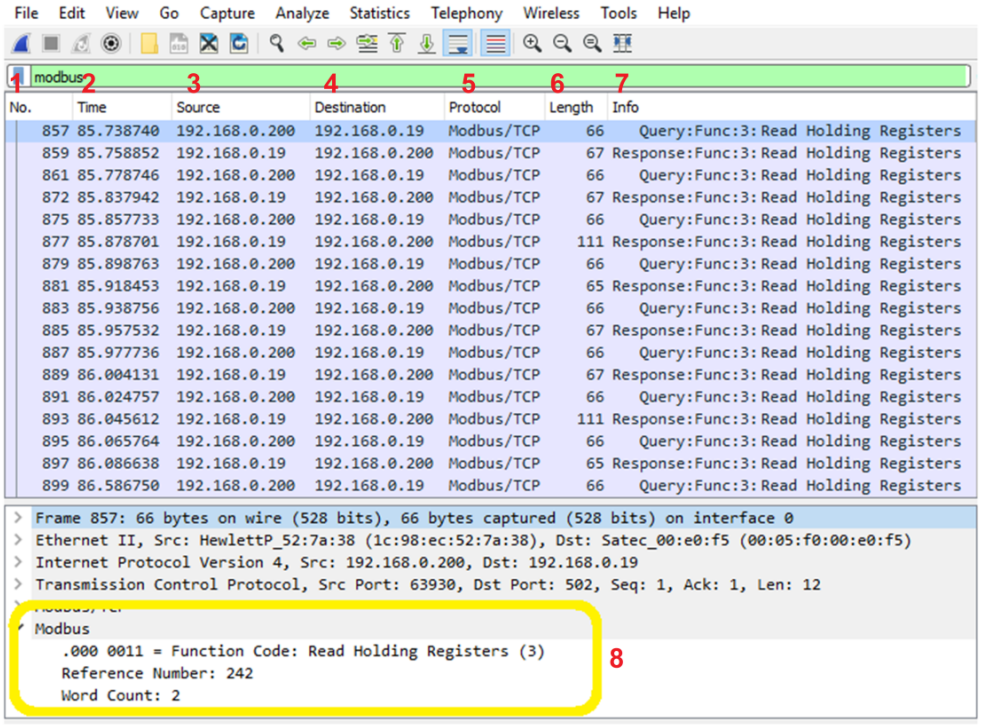


Figura 3.20. Detalles del protocolo Modbus en la interfaz de Wireshark.

Para la validación de los modelos se planteó el escenario de estudio que se presenta en la figura 3.21, en este se conectan vía Wi-Fi el SCADA y el medidor 9 situado aproximadamente a 225 metros del centro de gestión, a su vez se implementó el mismo escenario en NS-2.

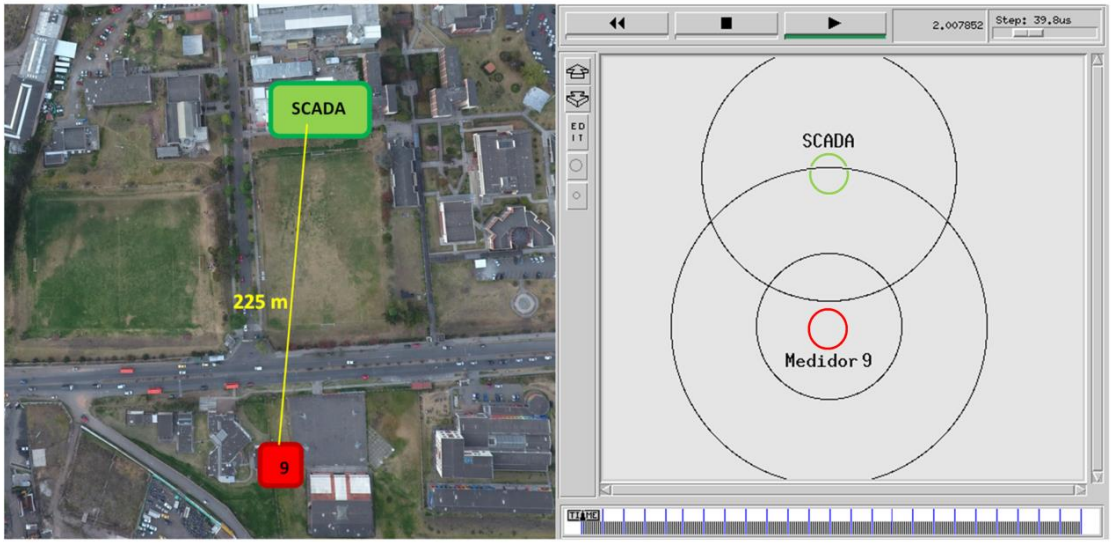


Figura 3.21. Escenario para la validación de los modelos.

Para las pruebas se configuró el SCADA para que envíe solicitudes de lectura al medidor cada segundo, con ello se busca determinar el comportamiento en el flujo de datos, *delay* y *throughput.* En la figura 3.22 se presenta un fragmento de la captura en Wireshark, en ella se observa que el flujo de datos es periódico, es decir, la misma secuencia de consultas y respuestas se repiten una y otra vez. En la figura 3.23a y 3.23b se presenta el patrón de tráfico, en azul se muestra la información que transmite el SCADA y en rojo la respuesta del medidor. Por lo tanto, a menos de que ocurra un evento atípico, se conoce exactamente la cantidad de bits que deben transmitir el SCADA y el medidor, de modo que este parámetro es simple de configurar en la simulación.

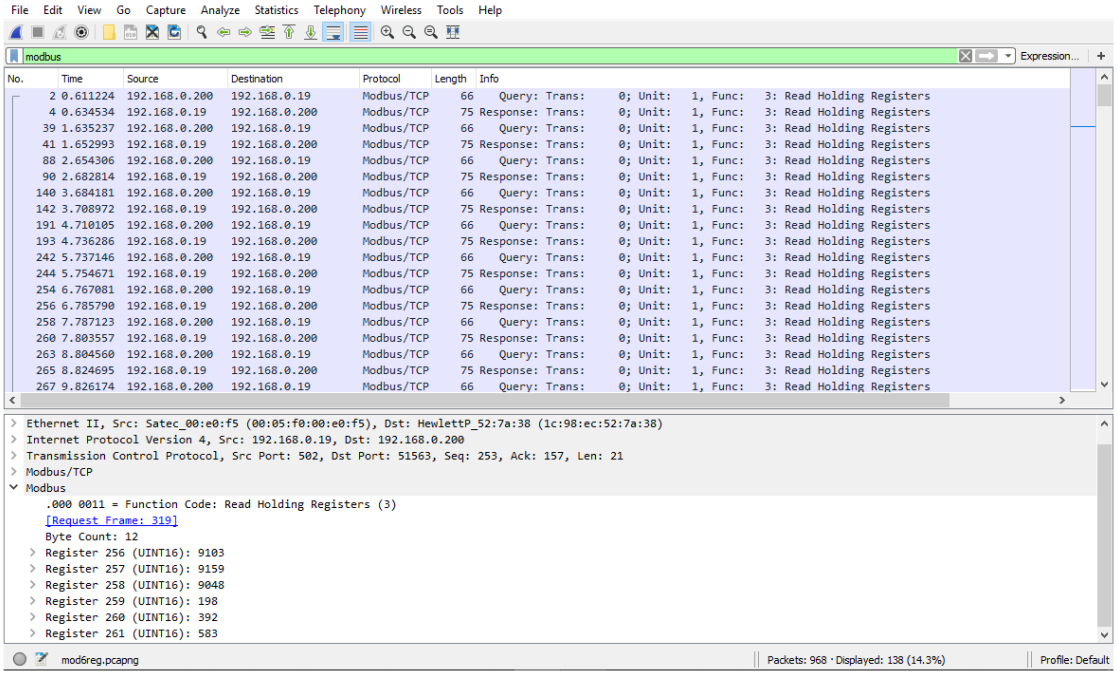
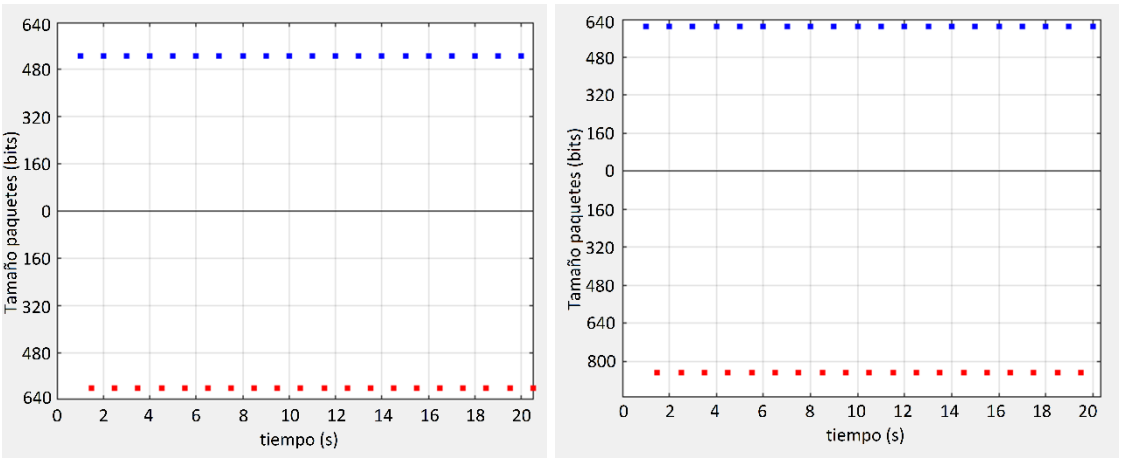


Figura 3.22. Captura de Tráfico Modbus para validación.



a. b.

Figura 3.23. a. Tráfico Modbus b. Tráfico DNP3 para validación.

Por otra parte, los resultados de *delay* medio y *throughput* se presentan en la tabla 3.6, se observa que bajo Modbus el *delay* promedio es de 20.61 ms, mientras que con DNP3 es de 23.37 ms. Para el *throughput* los resultados indican que Modbus utiliza 528 bps para la solicitud de lectura y de 600 bps para la respuesta, frente a 608 bps requeridos en DNP3 para la solicitud de lectura y 848 bps para la respuesta.

Tabla 3.6. Resultados de *delay* y *throughput* en el escenario de validación.

Modbus			**DNP3**		
Delay **[ms]**	***Throughput*** **[bps]**		***Delay*** **[ms]**	***Throughput*** **[bps]**	
	SCADA	**Medidor**		**SCADA**	**Medidor**
20.61	528	600	23.37	608	848

El siguiente paso fue realizar la simulación con diferentes modelos y parámetros de canal, en este caso se realizaron seis simulaciones denominadas como M1 a M6, en la tabla 3.7 se presentan los valores utilizados para simular el modelo Modbus. A demás, en la figura 3.24 se presenta una gráfica de cajas donde se representa los valores para las seis simulaciones y los resultados reales denominados como Modbus.

Tabla 3.7. Parámetros utilizados en el modelo de canal.

Simulación	**Modelo**	**Parámetros**
M1	*freespace*	N/A
M2	*TwoRay*	N/A
M3	*Shadowing*	$\beta = 2 \quad \sigma_{dB} = 4$
M4	*Shadowing*	$\beta = 3.4 \quad \sigma_{dB} = 5.4$
M5	*Shadowing*	$\beta = 3.5 \quad \sigma_{dB} = 5.4$
M6	*Shadowing*	$\beta = 3.5 \quad \sigma_{dB} = 6.2$

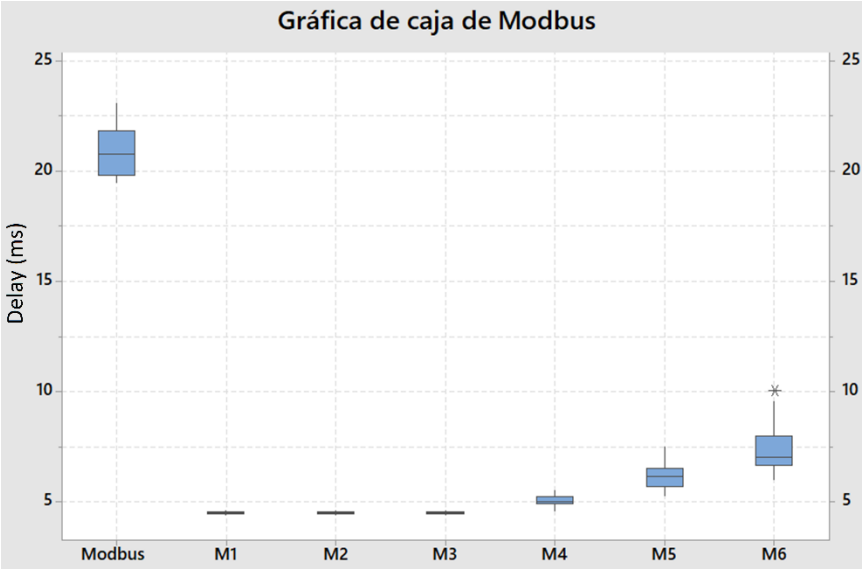


Figura 3.24. Gráfica de cajas para los valores reales y simulados bajo Modbus.

La gráfica de cajas de la figura 3.24 indica que para los modelos *freespace, TwoRay y Shadowing* con parámetros $\beta = 2$ *y* $\sigma_{dB} = 4$, los resultados de las simulaciones M1, M2 y M3 son los mismos y la distribución de valores que toma el *delay* es pequeña. Por otra parte, a medida que se aumentan los parámetros de pérdida de trayectoria y desviación de sombreado, el modelo se asemeja más a la distribución de las medidas reales en cuanto al ancho del intervalo de valores, pero los datos se encuentran demasiado distantes, ya que la media para los valores reales es de 20.61 ms y para M6 es de 7.414 ms.

En este punto se realizó un análisis exhaustivo del archivo *trace* para las simulaciones con el fin de determinar el porqué de dicha diferencia. En la figura 3.25 se presenta un fragmento del archivo *trace* para la simulación M6, en ella se observa que el SCADA envía la solicitud de lectura en el tiempo 0.000s, esta es recibida por el medidor en el tiempo 0.004844133s y en ese mismo instante genera la respuesta, que es recibida por el SCADA en el tiempo 0.007126333s. Como puede observarse la recepción de la solicitud en el medidor y el envío de la respuesta se realizan en el mismo instante (0.004844133s), lo que se aleja de la realidad, debido a que el medidor necesita de un intervalo mínimo de tiempo para procesar la solicitud y preparar la respuesta. El manual del medidor EM133 indica que el tiempo típico de respuesta a las solicitudes del maestro es de 13ms, por lo tanto para obtener un modelo semejante al real se debería considerar los 13ms de procesamiento de la

solicitud, aunque cabe aclarar que de este modo la simulación se ajusta solo al comportamiento del medidor utilizado, debido a que otros dispositivos podrían procesar las solicitudes en mayor o menor tiempo.

```
s 0.000000000 _0_ AGT  --- 0 MODdB 68
r 0.000000000 _0_ RTR  --- 0 MODdB 68
s 0.000000000 _0_ RTR  --- 0 MODdB 68
r 0.004844133 _1_ AGT  --- 0 MODdB 68
s 0.004844133 _1_ AGT  --- 1 MODdB 67
r 0.004844133 _1_ RTR  --- 1 MODdB 67
s 0.004844133 _1_ RTR  --- 1 MODdB 67
r 0.007126333 _0_ AGT  --- 1 MODdB 67
```

Figura 3.25. Fragmento del archivo *trace* para la simulación de Modbus.

En la figura 3.26 se presenta una gráfica de cajas para los valores reales y simulados con la adición de los 13ms de procesamiento, en este caso se observa que el modelo de simulación de M6a es el que más se asemeja a los valores reales. Por lo tanto se realiza una prueba de hipótesis para la diferencia entre las medias con el fin de contrastar los valores reales y de la simulación M6a. Este procedimiento se realizó en la herramienta estadística Minitab 18, en la tabla 3.8 se presentan detalles de la prueba.

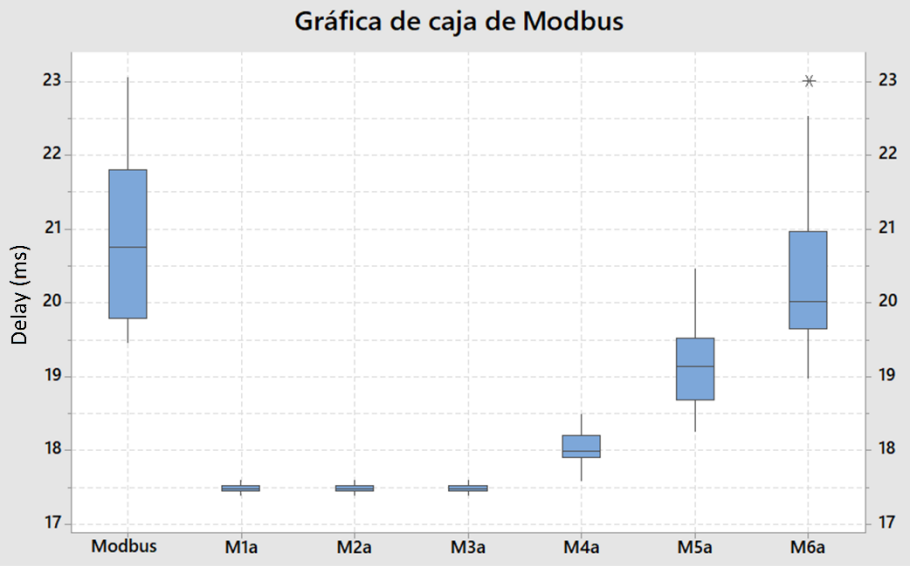


Figura 3.26. Gráfica de cajas para los valores reales y simulados modificados bajo Modbus.

Tabla 3.8. Parámetros prueba de hipótesis para Modbus.

Parámetro	Valor
Nivel de confianza	95%
Hipótesis nula	H_0: $\mu_1 - \mu_2 = 0$
Hipótesis alterna	H_1: $\mu_1 - \mu_2 \neq 0$
Valor p	0.116

En la tabla 3.8 se observa que el valor p es mayor a 0.05, por lo tanto no se puede rechazar la hipótesis nula y no existe evidencia suficiente para afirmar que la media de los datos reales y la media de los datos simulados difieran.

Para el caso de DNP3 se realizó un procedimiento similar, en la figura 3.27 se presenta una gráfica de cajas donde se representa los valores para las seis simulaciones y los resultados reales denominados como DNP3.

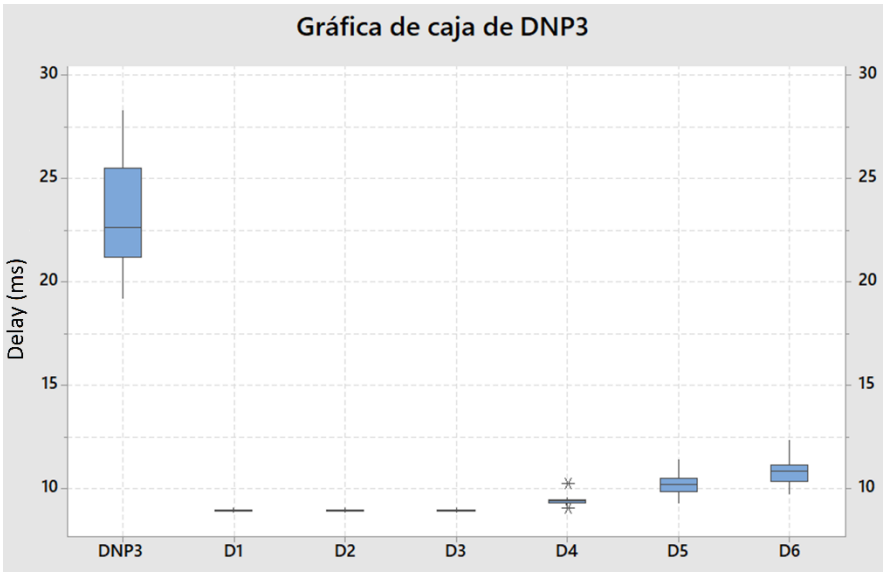


Figura 3.27. Gráfica de cajas para los valores reales y simulados bajo DNP3.

Posteriormente, se realiza el ajuste de los valores de simulación adicionando el tiempo de procesamiento de la solicitud, en la figura 3.28 se presentan los resultados.

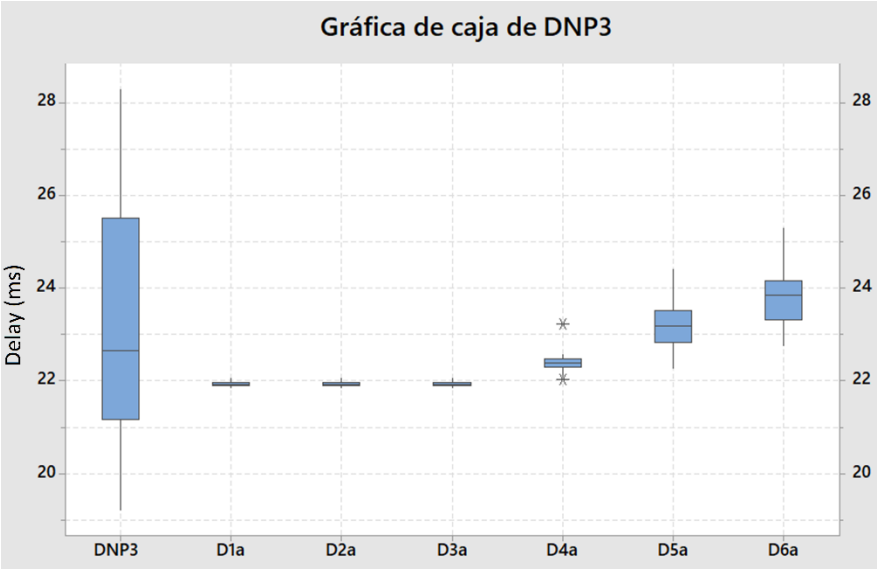


Figura 3.28. Gráfica de cajas para los valores reales y simulados modificados bajo DNP3.

A continuación, se realiza la prueba de hipótesis para la diferencia entre las medias con el fin de contrastar los valores reales y de la simulación D6a. En la tabla 3.9 se presentan detalles de la prueba.

Tabla 3.9. Parámetros prueba de hipótesis para DNP3.

Parámetro	Valor
Nivel de confianza	95%
Hipótesis nula	H_0: μ_1 - μ_2 = 0
Hipótesis alterna	H_1: μ_1 - μ_2 ≠ 0
Valor p	0.286

En la tabla 3.9 se observa que el valor p es mayor a 0.05, por lo tanto no se puede rechazar la hipótesis nula y no existe evidencia suficiente para afirmar que la media de los datos reales y la media de los datos simulados difieran.

Finalmente, se puede concluir que se validaron satisfactoriamente los modelos simulados y que estos se aproximan al sistema real cuando se utiliza un modelo de canal *Shadowing* con parámetros pérdida de trayectoria y desviación de sombreado iguales a $\beta = 3.5$ y $\sigma_{dB} = 6.2$, respectivamente.

Capítulo 4

Experimentación y análisis de resultados.

En este capítulo se presentan los resultados de la evaluación de desempeño de los protocolos Modbus y DNP3 bajo tres escenarios reales y de simulación, se calculó el *throughput* y el *delay* para determinar las funcionalidades que soporta la AMI implementada, considerando los valores de la tabla 4.1.

Tabla 4.1. Requisitos de *delay* para transmisión de mensajes [78].

Tipo	*Delay*	Aplicación
Protección	3 – 16 ms	Activación de alarmas, apertura, cierre
Monitoreo en tiempo real	16 – 100 ms	Reporte de estado
Operación SCADA	100 ms – 2 s	Lectura RTU

A continuación se describen los escenarios propuestos.

4.1. Evaluación de desempeño de Modbus y DNP3 mediante pruebas de campo

4.1.1. Escenario 1. Topología inalámbrica punto a punto – Modbus/DNP3

En este escenario, se configuró el SCADA para que envíe cada segundo solicitudes de lectura vía Wi-FI al medidor 9. El objetivo de esta configuración es estudiar el

comportamiento del flujo de datos entre el SCADA y el medidor para determinar el *delay* y el *throughput*. Para ello se realizaron dos pruebas, en la primera el SCADA solicita información de 6 variables (3 voltajes y 3 corrientes) y en la segunda de 27 variables. En la figura 4.1 se presenta el escenario 1 y en la tabla 4.2 los resultados de *delay* medio y *throughput* para la primera prueba.

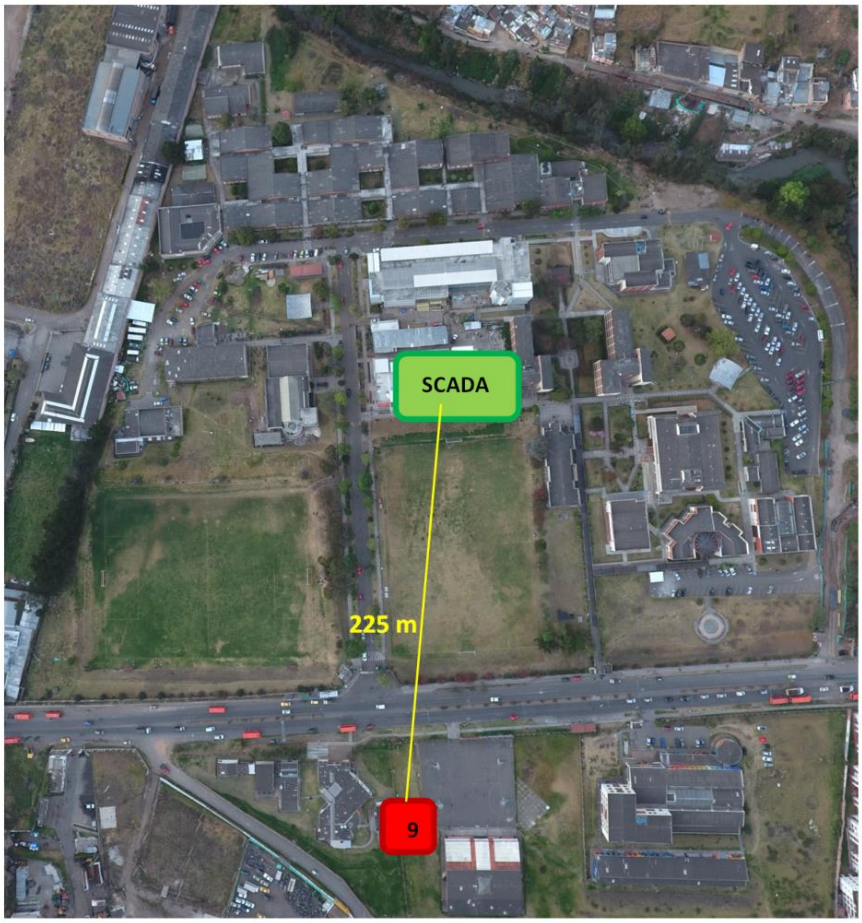


Figura 4.1. Escenario 1, topología inalámbrica punto a punto.

Tabla 4.2. Resultados Topología inalámbrica punto a punto para 6 variables.

Modbus			**DNP3**		
Delay **[ms]**	***Throughput*** **[bps]**		***Delay*** **[ms]**	***Throughput*** **[bps]**	
	SCADA	**Medidor**		**SCADA**	**Medidor**
20.61	528	600	23.37	576	848

Para esta primera prueba es importante resaltar que las variables de voltaje y corriente se encuentran almacenadas en posiciones de memoria consecutivas dentro del medidor, por lo tanto es necesario realizar una sola solicitud de lectura para los dos protocolos.

En la tabla 4.2 se observa que en la lectura de 6 variables mediante Modbus el *delay* promedio es de 20.61 ms, mientras que con DNP3 es más alta con un valor promedio de 23.37 ms. Para el *throughput* se indica la cantidad de bps necesarios para realizar la solicitud de lectura desde el SCADA y la cantidad de bps de respuesta desde el medidor. Por lo tanto, este parámetro indica que a menor cantidad de bps requeridos para obtener la misma información, mejor eficiencia del protocolo. Los resultados indican que bajo Modbus se obtienen mejores resultados con un *throughput* de 528 bps para la solicitud de lectura y de 600 bps para la respuesta, frente a 576 bps requeridos en DNP3 para la solicitud de lectura y 848 bps para la respuesta. Los resultados anteriores demuestran que para lecturas de posiciones de memoria consecutivas Modbus tiene mejor rendimiento, esto se debe a que DNP3 es un protocolo más robusto, lo que implica que sus tramas requieran mayor información y por ende mayor cantidad de bits son transmitidos. Además, los datos en Modbus se envían en un formato de 16 bits (2 Bytes), mientras que en DNP3 se envían en formato de 32 bits (4 Bytes), incrementando el número de Bytes transmitidos.

Por otra parte, en las figuras 4.2a y 4.2b se presentan los resultados de *delay* para Modbus y DNP3, respectivamente. Los resultados indican que los protocolos Modbus y DNP3 no son adecuados para aplicaciones de protección, al menos bajo la tecnología inalámbrica utilizada, ya que sus valores promedio se encuentran por encima de los 16 ms requeridos como máximo, como se indicó en la tabla 4.1. Sin embargo, los dos protocolos son ideales para aplicaciones de monitoreo en tiempo real y para sistemas SCADA.

En las figuras 4.2c y 4.2d se presenta el patrón de tráfico entre el SCADA y el medidor, para Modbus y DNP3, respectivamente. En azul se representa las solicitudes realizadas por el SCADA y en rojo las respuestas del medidor. Las gráficas indican que el flujo de información es periódico, es decir, la misma secuencia de consultas y respuestas se repiten una y otra vez. Este tipo de comportamiento es

deseado en la detección de intrusiones e implementación de sistemas contra ciberataques en sistemas SCADA [86. 87], por lo tanto, se conoce exactamente la cantidad de bits que deben transmitirse por la red de comunicaciones, según el requerimiento.

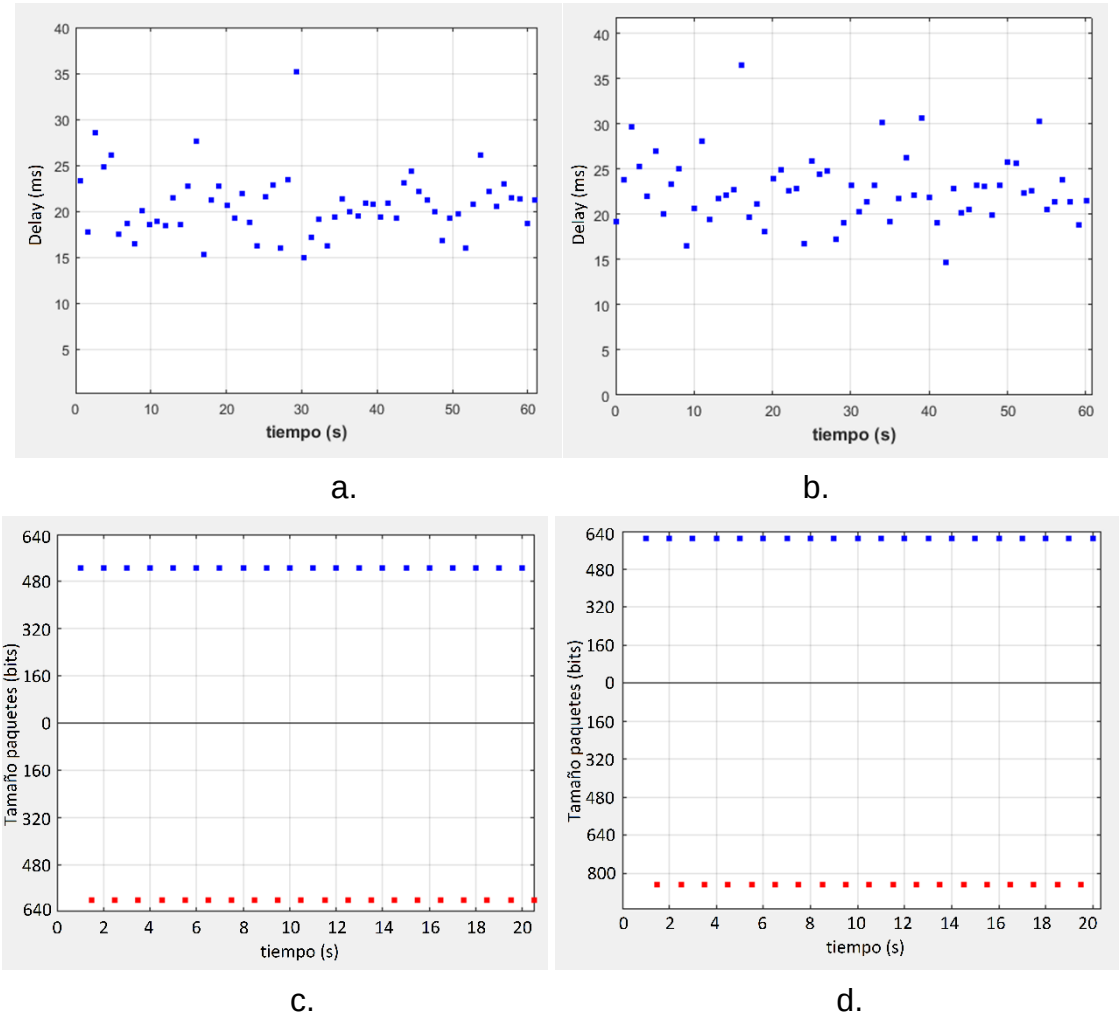


Figura 4.2. a. Retardos en mensajes Modbus. b. Retardos en mensajes DNP3. c. Tráfico Modbus. d. Tráfico DNP3.

La segunda prueba realizada en este escenario es la solicitud de lectura de varios parámetros de la red eléctrica, que comúnmente se recolectan en un sistema SCADA. Los parámetros solicitados por cada fase son los siguientes: voltaje, corriente, frecuencia, factor de potencia, potencia real, potencia reactiva, potencia aparente, Distorsión Armónica Total (THD, *Total Harmonic Distortion*) de voltaje, Distorsión Armónica Total (THD, *Total Harmonic Distortion*) de corriente, para un total

de 27 parámetros. Estas variables no se encuentran almacenadas de forma secuencial en la memoria del dispositivo, lo que permite conocer el comportamiento de los protocolos frente a este tipo de solicitudes. En la tabla 4.3 se presentan los resultados del estudio.

Tabla 4.3. Resultados topología inalámbrica punto a punto para 27 variables.

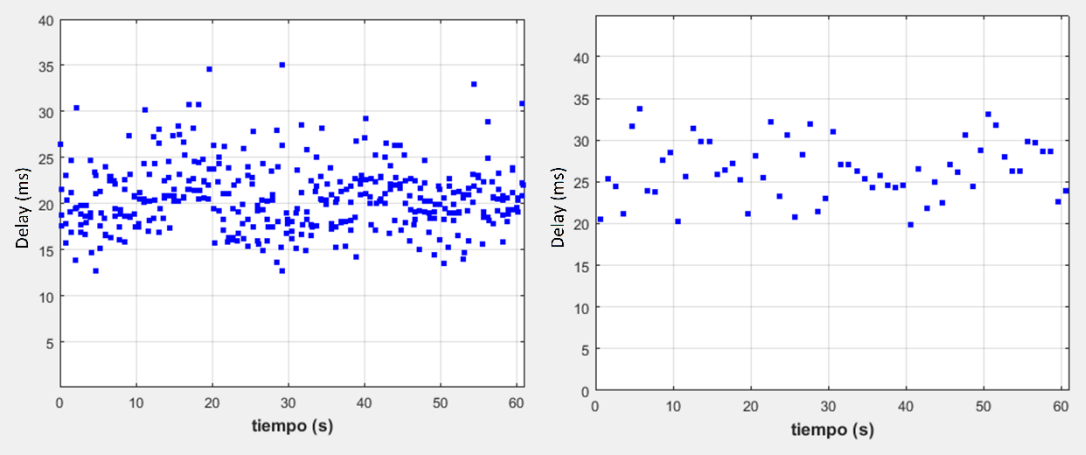
Modbus			**DNP3**		
Delay **[ms]**	***Throughput*** **[bps]**		***Delay*** **[ms]**	***Throughput*** **[bps]**	
	SCADA	**Medidor**		**SCADA**	**Medidor**
84.20	2112	2480	26.98	576	2024

En la tabla 4.3 se observa que para la lectura de 27 variables el *delay* promedio para Modbus es de 84.20 ms, muy por encima de los 26.98 ms que requiere DNP3. Del mismo modo, los cálculos de *throughput* indican mejores resultados bajo DNP3 con 576 bps para solicitud de lectura y 2024 bps para respuesta, frente a los 2112 bps que requiere Modbus para solicitudes de lectura y 2480 bps para respuesta. Los resultados demuestran que el rendimiento de Modbus en este tipo de solicitudes está muy por debajo de DNP3, para comprender este cambio en el rendimiento de Modbus se debe considerar que las 27 variables de interés se encuentran almacenadas en cuatro bloques de memoria diferentes. Los registros de wireshark muestran que bajo Modbus se realizan cuatro lecturas para recuperar la información requerida por el SCADA, por lo tanto necesita realizar una solicitud de lectura por cada bloque de datos que se desee leer. Por otra parte, DNP3 realizó una sola solicitud de lectura para recibir información de las 27 variables, esto se debe a que DNP3 permite configurar la respuesta asignando hasta 32 rangos de puntos a leer. Esta característica le permite a DNP3 tener un mejor rendimiento en este tipo de escenarios. En la tabla 4.4 se presenta el *delay* y *throughput* para cada una de las lecturas Modbus, frente a la solicitud de lectura DNP3.

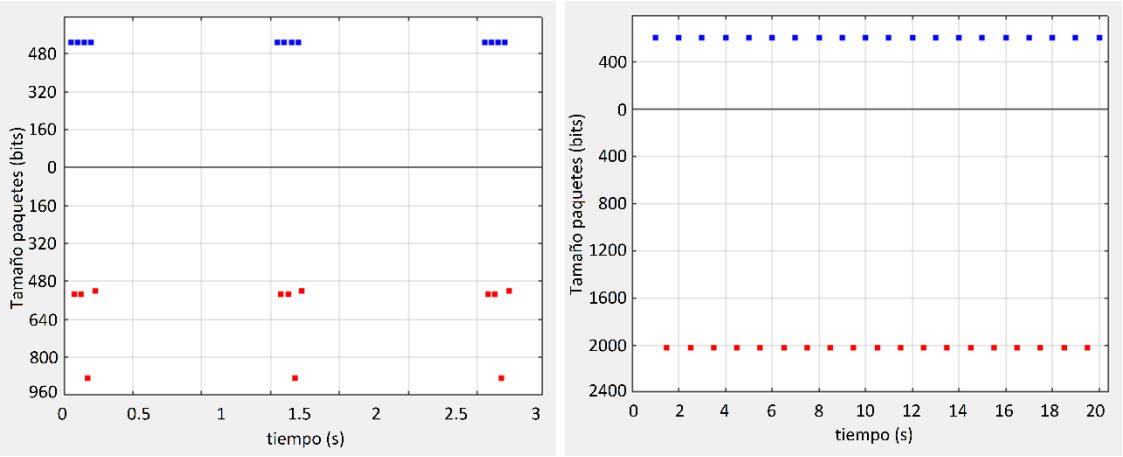
Tabla 4.4. Resultados de *delay* y *throughput* topología inalámbrica para 27 variables.

	Modbus				**DNP3**	
	***Delay* [ms]**	**SCADA [bps]**	**Medidor [bps]**	***Delay* [ms]**	**SCADA [bps]**	**Medidor [bps]**
Lectura 1	20.59	528	536	26.98	576	2024
Lectura 2	20.75	528	536	-	-	-
Lectura 3	22.28	528	888	-	-	-
Lectura 4	20.58	528	520	-	-	-
Total	84.20	2112	2480	26.98	576	2024

En las figuras 4.3a y 4.3b se presentan las gráficas de retardos para Modbus y DNP3, respectivamente y en las figuras 4.3c y 4.3d se presentan los patrones de tráfico para Modbus y DNP3, respectivamente. Al igual que en el estudio anterior el patrón de tráfico es constante, con solicitudes y lecturas de tamaños fijos.



a. b.



c. d.

Figura 4.3. a. Retardos en mensajes Modbus. b. Retardos en mensajes DNP3. c. Tráfico Modbus. d. Tráfico DNP3 para 27 variables.

4.1.2. Escenario 2. Topología cableada punto a punto – Modbus/DNP3

En este escenario se conectó el medidor 10 vía Ethernet con el SCADA, que a su vez envía solicitudes de lectura cada segundo. El objetivo de este escenario fue analizar el flujo de información para determinar el *throughput* y el *delay* de los protocolos bajo tecnología Ethernet y comparar los resultados con los de Wi-Fi. Para esta prueba el SCADA solicitó información de las 27 variables, en la figura 4.4 se muestra el escenario de estudio y los resultados se presentan en la tabla 4.5.

Figura 4.4. Escenario 1, topología cableada punto a punto.

Tabla 4.5. Resultados Topología cableada punto a punto.

	Modbus			**DNP3**		
	***Delay* [ms]**	**SCADA [bps]**	**Medidor [bps]**	***Delay* [ms]**	**SCADA [bps]**	**Medidor [bps]**
Lectura 1	19.19	528	536	23.48	576	2024
Lectura 2	19.77	528	536	-	-	-
Lectura 3	21.12	528	888	-	-	-
Lectura 4	18.65	528	520	-	-	-
Total	78.73	2112	2480	23.48	576	2024

Los resultados de la tabla 4.5 demuestran que bajo tecnología Ethernet se mejoró el *delay*, en Modbus se redujo de 84.20 ms a 78.73 ms y en DNP3 de 26.98 ms a 23.48 ms, el *throughput* permaneció sin cambios. A pesar de que los valores de *delay*

disminuyeron, no se encuentran dentro del rango admisible para funciones de protección.

4.1.3. Escenario 3. Topología inalámbrica multipunto – Modbus/DNP3

En este escenario se configuró el SCADA para que realice solicitudes de lectura a los diez medidores inteligentes. Esto permite analizar si los parámetros de rendimiento se afectan por la conexión de más de un medidor al SCADA. En la figura 4.5 se presenta el escenario propuesto y en la tabla 4.6 los resultados de la prueba.

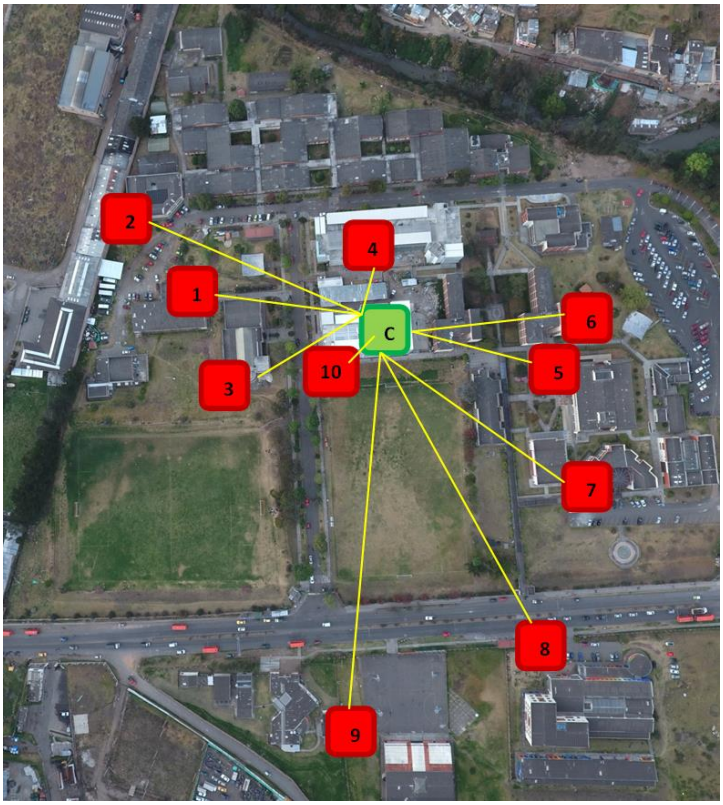


Figura 4.5. Escenario 3, topología inalámbrica multipunto.

Tabla 4.6. Resultados Topología inalámbrica multipunto.

Medidor	Modbus		DNP3	
	Delay **[ms]**	***Throughput*** **[bps]**	***Delay*** **[ms]**	***Throughput*** **[bps]**
1	86.51	2480	28.67	2024
2	87.15	2480	28.23	2024

3	82.94	2480	27.02	2024
4	82.88	2480	26.87	2024
5	83.88	2480	27.03	2024
6	84.46	2480	28.01	2024
7	84.76	2480	26.98	2024
8	84.20	2480	27.50	2024
9	89.89	2480	29.82	2024
10	77.78	2480	23.02	2024

En la tabla 4.6 se muestra que el *delay* no sufre cambios significativos por la conexión de los diez medidores, además se observa que el mejor rendimiento pertenece a la conexión Ethernet del medidor 10. Los valores de *throughput* permanecen constantes con 2480 bps para el envío de información desde el medidor bajo Modbus y de 2024 bps bajo DNP3. Por otra parte, se observó que el *throughput* del SCADA sufrió cambios significativos, pues es el encargado de enviar y recibir toda la información referente a las solicitudes de lectura.

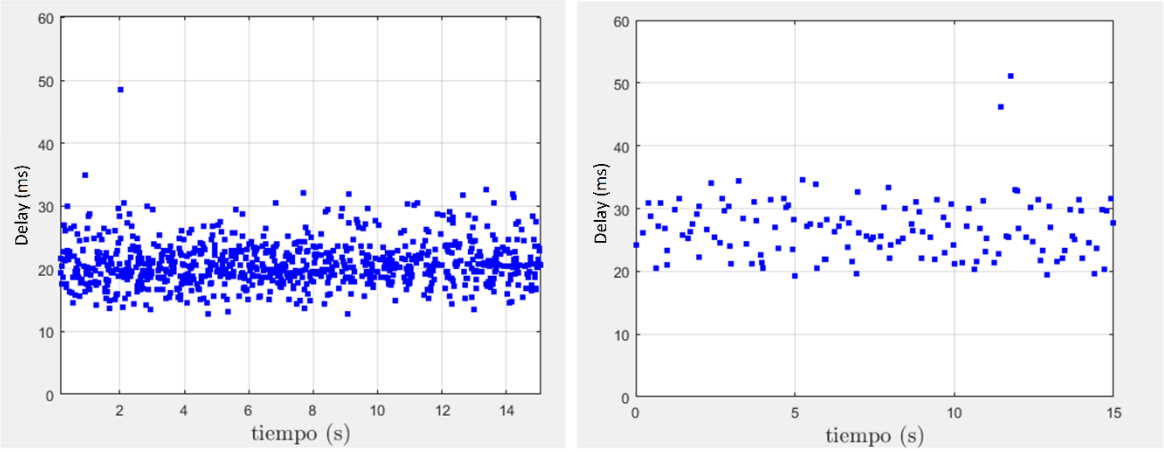
En la tabla 4.7 se presentan los resultados de *throughput* para el SCADA, cabe anotar que el *throughput* de envío se refiere a los bps necesarios para la solicitud de lectura y *throughput* recibe a los bps de la respuesta. De esta forma, bajo Modbus el *throughput* total para envío de solicitudes es de 21120 bps y de 24800 bps para las respuestas, frente a 5760 bps para envío de solicitudes y 20240 bps para respuestas bajo DNP3.

Tabla 4.7. Resultados de *throughput* en el SCADA para conexión de diez medidores.

Medidor	**SCADA – Modbus**		**SCADA - DNP3**	
	***Throughput* [bps] Envío**	***Throughput* [bps] Recibe**	***Throughput* [bps] Envío**	***Throughput* [bps] Recibe**
1	2112	2480	576	2024
2	2112	2480	576	2024
3	2112	2480	576	2024
4	2112	2480	576	2024
5	2112	2480	576	2024

6	2112	2480	576	2024
7	2112	2480	576	2024
8	2112	2480	576	2024
9	2112	2480	576	2024
10	2112	2480	576	2024
Total	21120	24800	5760	20240

Finalmente, en la figura 4.6 se presentan los retardos para las solicitudes de lectura en un intervalo de 15 segundos. Es evidente la gran cantidad de solicitudes que requiere Modbus para obtener la misma información que con DNP3.



a. b.

Figura 4.6. a. Retardos en mensajes Modbus. b. Retardos en mensajes DNP3.

4.2. Evaluación de desempeño de Modbus y DNP3 en NS-2

En esta sección se presentan los resultados de la evaluación de desempeño de los protocolos por medio de simulación, para ello se simularon escenarios similares a los reales.

4.2.1. Escenario 1. Topología inalámbrica punto a punto – Modbus/DNP3 en NS-2

En este escenario, se crearon dos nodos, uno como servidor y otro como cliente que se comunican vía Wi-Fi, en los apéndices A y B se presentan los códigos Otcl para este escenario bajo DN3 y Modbus, respectivamente, y que se detallaron en la sección 3.4. Por su parte, en la tabla 4.8 se presentan los principales parámetros de configuración y en la figura 4.7 se muestra el escenario en *Network Animator*. Para el flujo de datos se simuló solicitudes de información de 27 variables cada segundo.

Tabla 4.8. Parámetros de la red modelada.

Parámetro	Valor
Canal	WirelessChannel
Propagación	Shadowing
Interfaz de red	Wireless Phy
Capa MAC	802_11
Frecuencia	2.4 GHz
Tipo de cola	DropTail
Capa de enlace	LL
Modelo de antena	OmniAntenna
Número de nodos	2
Protocolo de enrutamiento	DumbAgent
Distancia entre nodos	220 m
Taza de transmisión	11 Mb

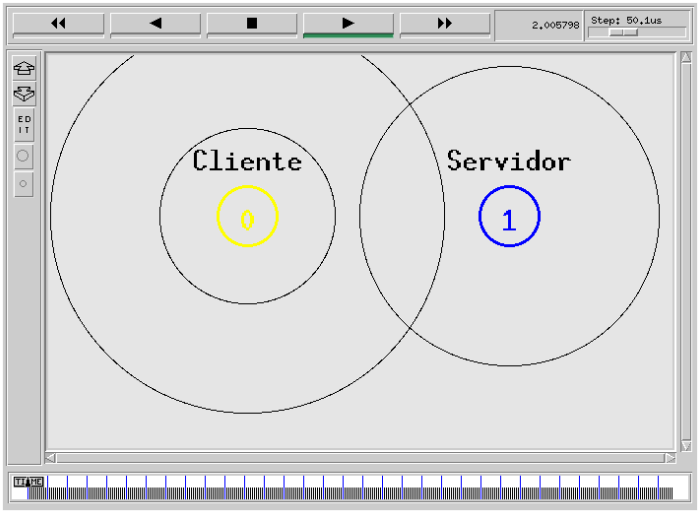


Figura 4.7. Topología inalámbrica punto a punto en NS-2.

En este escenario se midió el *delay* y el *throughput,* en la tabla 4.9 se presentan los resultados de la prueba y en la figura 4.8 se presenta un fragmento del archivo *trace* para la simulación con DNP3, en donde se observa que el flujo de datos se ajusta a las pruebas reales, es decir, 576 bits (72 Bytes) para la solicitud de lectura y 2024 bits (253 Bytes) para la respuesta.

```
s 0.000000000 _0_ AGT  --- 0 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
r 0.000000000 _0_ RTR  --- 0 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
s 0.000000000 _0_ RTR  --- 0 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
r 0.004891667 _1_ AGT  --- 0 DNP3 72 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 0.004891667 _1_ AGT  --- 1 DNP3 253 [0 0 0 0] ------- [1:0 0:0 32 0]
r 0.004891667 _1_ RTR  --- 1 DNP3 253 [0 0 0 0] ------- [1:0 0:0 32 0]
s 0.004891667 _1_ RTR  --- 1 DNP3 253 [0 0 0 0] ------- [1:0 0:0 32 0]
r 0.008661667 _0_ AGT  --- 1 DNP3 253 [13a 0 1 800] ------- [1:0 0:0 32 0]
s 1.000000000 _0_ AGT  --- 3 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
r 1.000000000 _0_ RTR  --- 3 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
s 1.000000000 _0_ RTR  --- 3 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
r 1.002358000 _1_ AGT  --- 3 DNP3 72 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 1.002358000 _1_ AGT  --- 4 DNP3 253 [0 0 0 0] ------- [1:0 0:0 32 0]
r 1.002358000 _1_ RTR  --- 4 DNP3 253 [0 0 0 0] ------- [1:0 0:0 32 0]
s 1.002358000 _1_ RTR  --- 4 DNP3 253 [0 0 0 0] ------- [1:0 0:0 32 0]
r 1.006248000 _0_ AGT  --- 4 DNP3 253 [13a 0 1 800] ------- [1:0 0:0 32 0]
s 2.000000000 _0_ AGT  --- 6 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
r 2.000000000 _0_ RTR  --- 6 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
s 2.000000000 _0_ RTR  --- 6 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
r 2.002018000 _1_ AGT  --- 6 DNP3 72 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 2.002018000 _1_ AGT  --- 7 DNP3 253 [0 0 0 0] ------- [1:0 0:0 32 0]
r 2.002018000 _1_ RTR  --- 7 DNP3 253 [0 0 0 0] ------- [1:0 0:0 32 0]
s 2.002018000 _1_ RTR  --- 7 DNP3 253 [0 0 0 0] ------- [1:0 0:0 32 0]
r 2.005568000 _0_ AGT  --- 7 DNP3 253 [13a 0 1 800] ------- [1:0 0:0 32 0]
s 3.000000000 _0_ AGT  --- 9 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
r 3.000000000 _0_ RTR  --- 9 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
s 3.000000000 _0_ RTR  --- 9 DNP3 72 [0 0 0 0] ------- [0:0 1:0 32 0]
r 3.002218000 _1_ AGT  --- 9 DNP3 72 [13a 1 0 800] ------- [0:0 1:0 32 0]
s 3.002218000 _1_ AGT  --- 10 DNP3 253 [0 0 0 0] ------- [1:0 0:0 32 0]
r 3.002218000 _1_ RTR  --- 10 DNP3 253 [0 0 0 0] ------- [1:0 0:0 32 0]
```

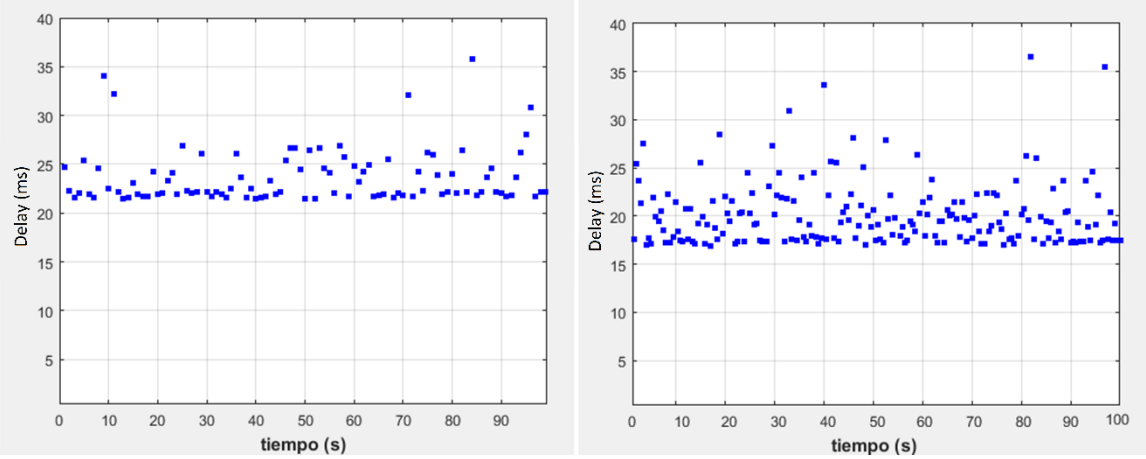
Figura 4.8. Fragmento archivo *trace* para escenario inalámbrico punto a punto.

Tabla 4.9. Resultados Topología inalámbrica punto a punto en NS-2.

Modbus			**DNP3**		
Delay **[ms]**	***Throughput*** **[bps]**		***Delay*** **[ms]**	***Throughput*** **[bps]**	
	SCADA	**Medidor**		**SCADA**	**Medidor**
81.65	2112	2480	23.81	576	2024

En la tabla 4.9 se observa que en la lectura de 27 variables mediante Modbus el *delay* promedio es de 81.65 ms, mientras que con DNP3 es mucho menor con un valor promedio de 23.81 ms, en general estos valores son menores a los obtenidos en las pruebas de campo, pero se encuentran en un intervalo aceptable para concluir que los dos protocolos no soportan funciones de protección. Para el *throughput* los

resultados indican que este parámetro se puede ajustar perfectamente a los resultados reales, bajo Modbus se requiere de 2112 bps para la solicitud de lectura y de 2480 bps para la respuesta, para DNP3 se tiene que 576 bps son requeridos para la solicitud de lectura y 2024 bps para la respuesta. Finalmente, en la figura 4.9a y 4.9b se presentan las gráficas de retardos para DNP3 y Modbus, respectivamente, en donde se observa un mayor número de transmisiones para Modbus.



a. b.

Figura 4.9. a. Retardos en mensajes DNP3. b. Retardos en mensajes Modbus. En escenario inalámbrico en NS-2.

4.2.2. Escenario 2. Topología cableada punto a punto – Modbus/DNP3 en NS-2

En este escenario se configuró la conexión de dos nodos vía Ethernet, que representan al SCADA y a un medidor inteligente, en este sistema se enviaron solicitudes de lectura cada segundo, en los apéndices C y D se presentan los códigos de NS-2 correspondientes a la comunicación bajo DNP3 y Modbus, respectivamente, los detalles del código se presentaron en la sección 3.3.1. Análogamente al escenario anterior se analizó el flujo de información para determinar el *throughput* y el *delay*. En la figura 4.10 se muestra el escenario de estudio y los resultados se presentan en la tabla 4.10.

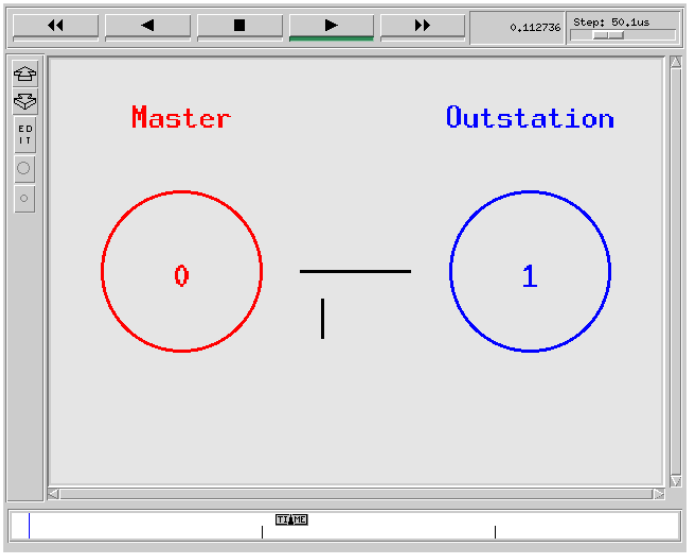


Figura 4.10. Topología cableada punto a punto en NS-2.

Tabla 4.10. Resultados Topología cableada punto a punto en NS-2.

Modbus			**DNP3**		
Delay **[ms]**	***Throughput*** **[bps]**		***Delay*** **[ms]**	***Throughput*** **[bps]**	
	SCADA	**Medidor**		**SCADA**	**Medidor**
72.45	2112	2480	20.26	576	2024

Los resultados de la tabla 4.10 demuestran que en la simulación de un escenario con tecnología Ethernet se obtienen mejores resultados que con comunicación Wi-Fi, como ocurrió en las pruebas de campo. En este caso, el *delay*, en Modbus se redujo de 81.65 ms a 72.45ms y en DNP3 de 23.81 ms a 20.26 ms, el *throughput* permaneció sin cambios. En la simulación también se concluye que a pesar de que los valores de *delay* disminuyeron, no se encuentran dentro del rango admisible para funciones de protección.

4.2.3. Escenario 3. Topología inalámbrica multipunto – Modbus/DNP3 en NS-2

En este escenario, se crearon 11 nodos que representan a diez medidores y al SCADA, que envía solicitudes de lectura a cada medidor cada segundo. En la figura 4.11 se muestra el escenario en *Network Animator,* en este punto se configuró el área de trabajo con dimensiones similares a las del Campus universitario y los nodos

se ubicaron aproximadamente a la misma distancia que se encuentran del SCADA en la instalación real. En la tabla 4.11 se exponen los resultados de la prueba y en los apéndices E y F se presentan los códigos computacionales para la prueba bajo Modbus y DNP3, respectivamente. Estos códigos son similares a los presentados en los apéndices A y B, solo que en este caso se crearon más nodos.

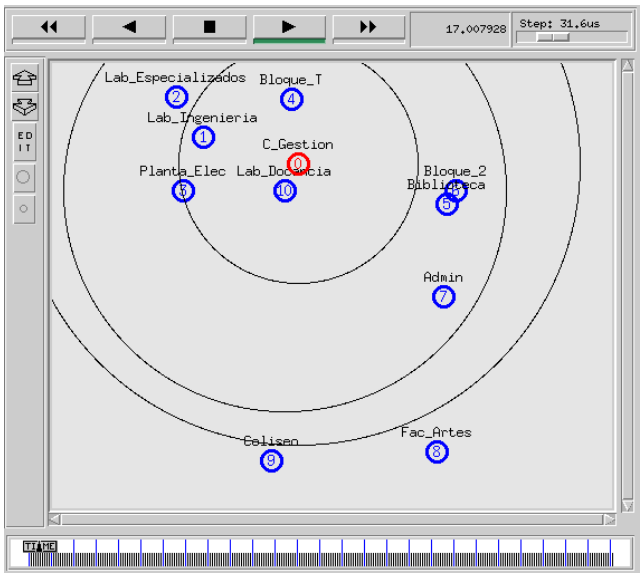


Figura 4.11. Topología inalámbrica multipunto en NS-2.

Tabla 4.11. Resultados topología multipunto en NS-2.

Medidor	Modbus		DNP3	
	Delay **[ms]**	***Throughput*** **[bps]**	***Delay*** **[ms]**	***Throughput*** **[bps]**
1	82.23	2480	21.93	2024
2	81.75	2480	21.95	2024
3	81.95	2480	21.93	2024
4	81.61	2480	21.93	2024
5	81.75	2480	21.98	2024
6	81.27	2480	21.98	2024
7	81.75	2480	22.21	2024
8	87.54	2480	25.35	2024
9	84.74	2480	24.54	2024
10	82.14	2480	21.93	2024

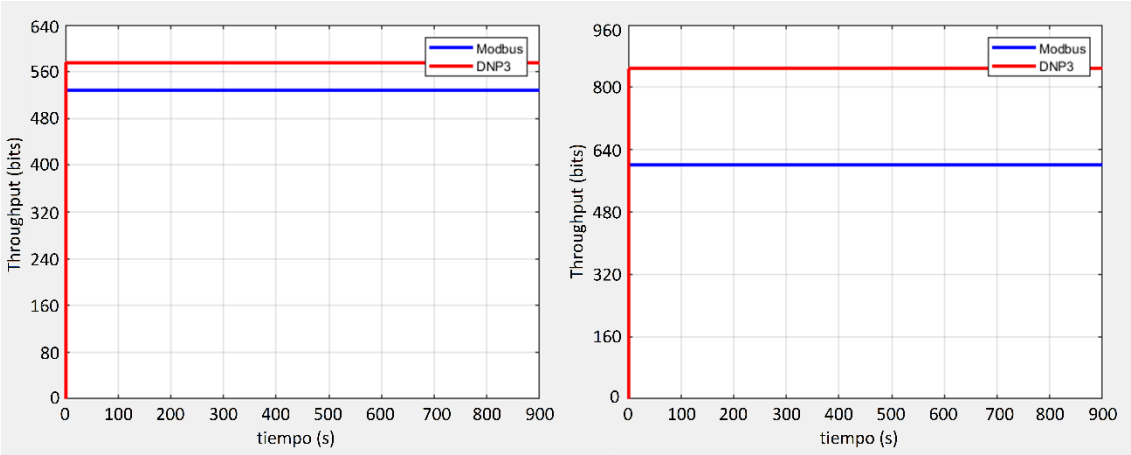
En la tabla 4.11 se muestra que el *delay* no sufre cambios significativos por la conexión de los diez medidores, pero se observa una leve variación en función de la distancia a la que se encuentra del SCADA. Los valores de *throughput* permanecen constantes con 2480 bps para el envío de información desde el medidor bajo Modbus y de 2024 bps bajo DNP3.

4.3. Análisis comparativo de resultados

En esta sección se realiza un análisis comparativo de los resultados de las pruebas de campo con el fin de establecer el protocolo que ofrezca mejor desempeño y un análisis comparativo de los resultados reales y teóricos para establecer la diferencia entre las dos pruebas.

4.3.1. Análisis comparativo de resultados en pruebas de campo

Inicialmente, se analiza el tráfico generado en la comunicación de dos nodos (SCADA y medidor inteligente) conectados vía Wi-Fi. Aquí el SCADA envía cada segundo solicitudes de lectura de posiciones de memoria consecutivas, es decir, que se realiza una sola lectura. En la figura 4.12a se presenta el resultado de *throughput*, se observa que Modbus requiere de 528 bps para la solicitud de lectura frente a 576 bps bajo DNP3. En la figura 4.12b se presentan las respuestas, en Modbus son necesarios 600 bps frente a 848 bps en DNP3.



a. b.

Figura 4.12. Comparativa de *throughput* para lectura de posiciones de memoria consecutiva.

Por su parte, en la figura 4.13, se muestra el *delay* en la entrega de información, los resultados obtenidos mediante la función de distribución acumulativa en una muestra de 900 solicitudes, demuestran a través de las curvas que el retardo bajo Modbus es menor que con DNP3, en general el 90% de los paquetes enviados bajo Modbus tienen un retraso menor o igual a 23 ms, mientras que en DNP3 es de 27 ms.

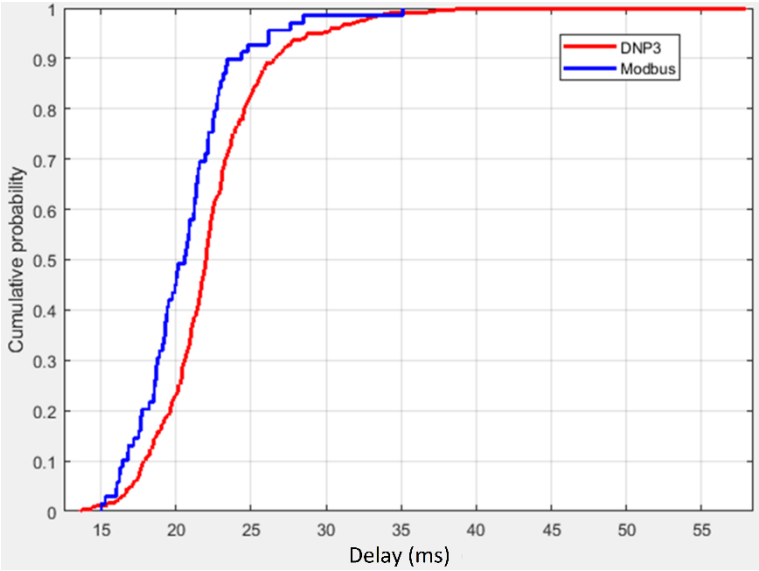
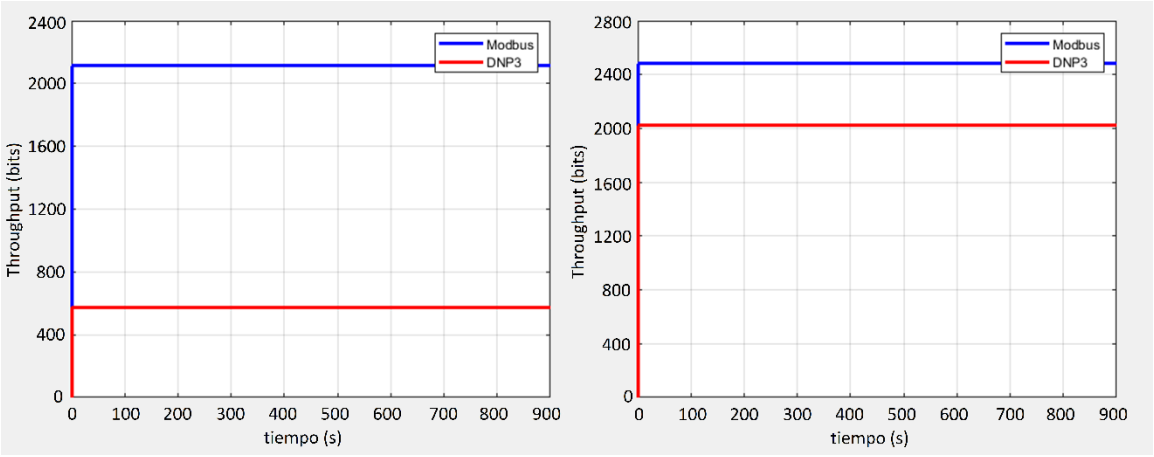


Figura 4.13. Comparativa de *delay* para lectura de posiciones de memoria consecutiva.

En segundo lugar se comparan los resultados cuando el SCADA solicita información que se encuentra en diferentes bloques de memoria en el medidor. En este caso, se leen cada segundo 27 variables distribuidas en 4 bloques, es decir, es necesario realizar 4 lecturas bajo Modbus frente a una bajo DNP3. En la figura 4.14 se presentan los resultados de *throughput*, en la figura 4.14a se observa en azul que son necesarios transmitir 2112 bps para la solicitud de lectura bajo Modbus frente a 576 bps para DNP3. En la figura 4.14b se presentan las respuestas, en Modbus se requieren 2480 bps frente a 2024 bps en DNP3.



a. b.

Figura 4.14. Comparativa de *throughput* para lectura de cuatro bloques de memoria.

Por otro lado, en la figura 4.15, se muestra el *delay* en la entrega de información, los resultados obtenidos mediante la función de distribución acumulativa en una muestra de 900 solicitudes, demuestran a través de las curvas que el retardo bajo Modbus es mucho mayor que con DNP3, en general el 90% de los paquetes enviados bajo Modbus tienen un retraso menor o igual a 95 ms, mientras que en DNP3 es de 32 ms.

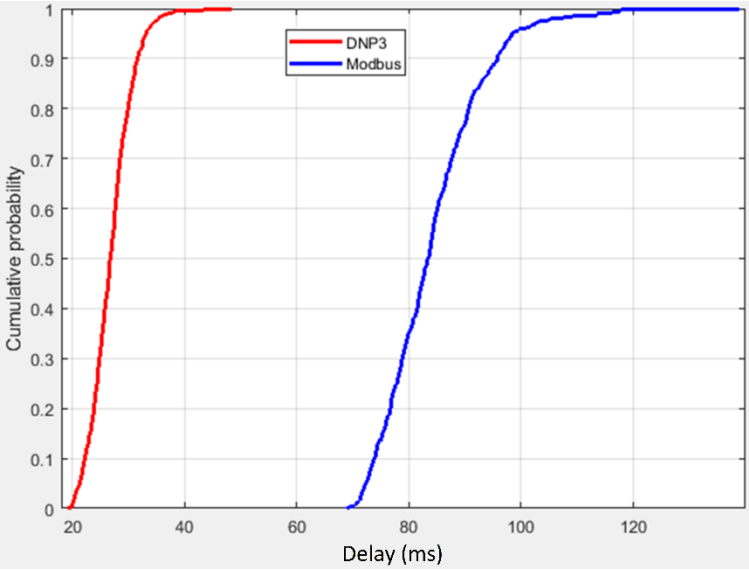
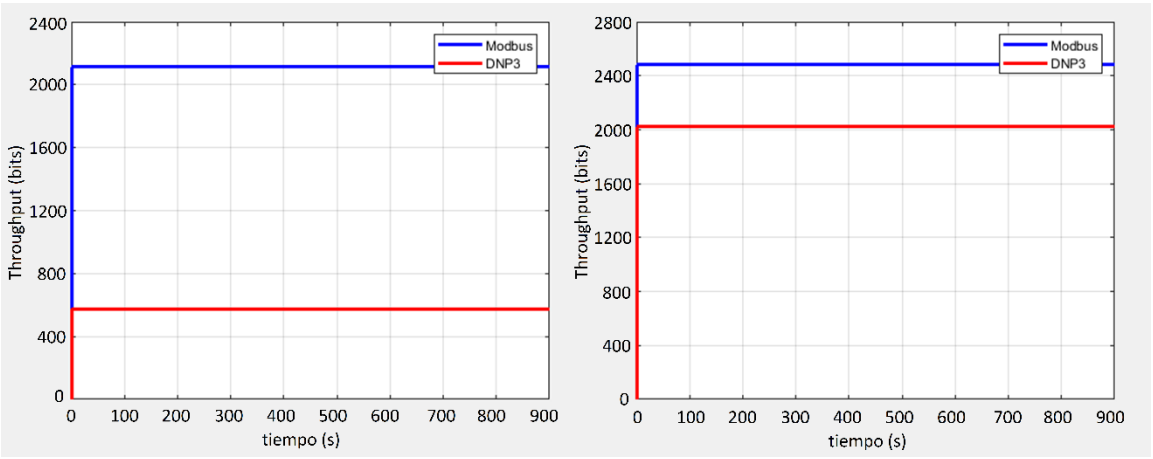


Figura 4.15. Comparativa de *delay* para lectura de cuatro bloques de memoria.

En tercer lugar, se comparan los resultados de la comunicación entre el SCADA y un medidor conectados bajo Ethernet. En este caso, cada segundo se leen cuatro bloques de memoria y la distancia entre los nodos es de aproximadamente 80 metros. En la figura 4.16 se observa que el *throughput* no sufrió cambios, en la figura 4.16a se detalla en azul que son necesarios 2112 bps para la solicitud de lectura bajo Modbus frente a 576 bps bajo DNP3. En la figura 4.16b se presentan las respuestas, en Modbus se requieren 2480 bps frente a 2024 bps en DNP3.



a. b.

Figura 4.16. Comparativa de *throughput* para lectura de cuatro bloques de memoria bajo Ethernet.

Por su parte, la función de distribución acumulativa demuestra una disminución en el *delay* frente a Wi-Fi. En la figura 4.17 se observa que el 90% de los paquetes enviados bajo Modbus experimentan un retardo menor de 85 ms, mientras que en DNP3 es de 27 ms.

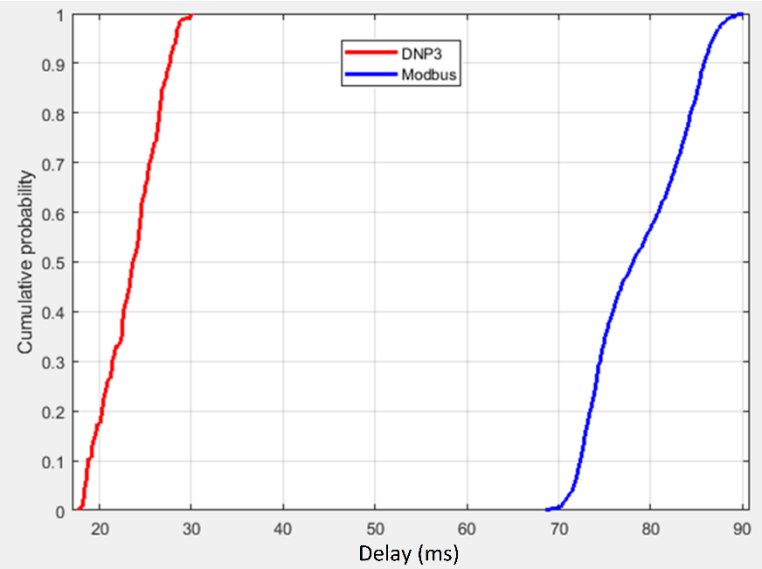


Figura 4.17. Comparativa de *delay* para lectura de cuatro bloques de memoria bajo Ethernet.

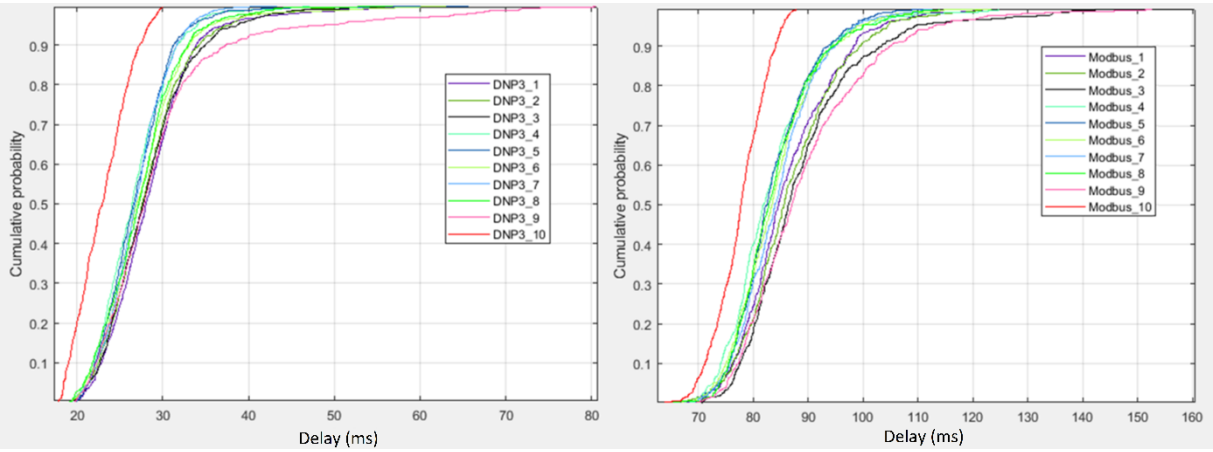
Finalmente, se presenta una comparativa del *delay* en la comunicación de los diez medidores, nueve vía Wi-Fi y uno bajo Ethernet. De igual forma en este escenario, se leen cada segundo cuatro bloques de memoria para un total de 27 variables. En la tabla 4.12 se presentan los resultados de *delay* y las distancias entre los medidores y el SCADA, los resultados indican que a medida que la distancia aumenta también el *delay*.

Tabla 4.12. Distancia y *delay* de medidores a SCADA.

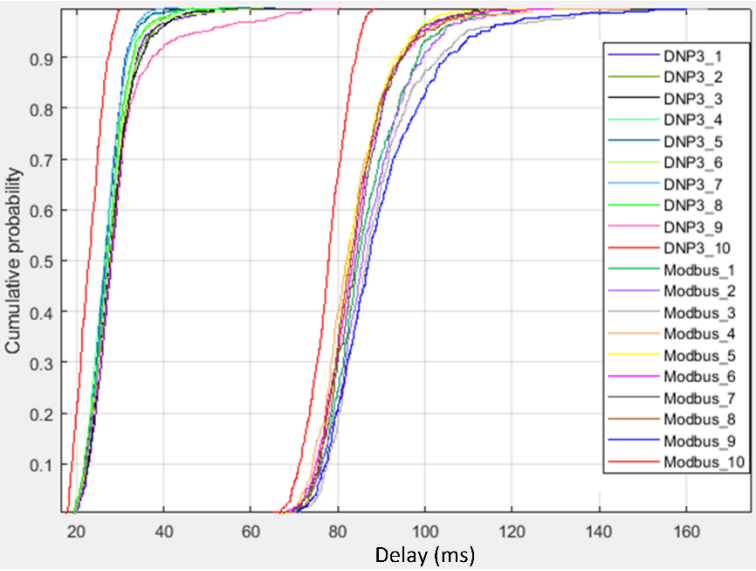
Medidor	**Distancia al SCADA [m]**	***Delay* Modbus [ms]**	***Delay* DNP3 [ms]**
1	99	86.51	28.67
2	145	87.15	28.23
3	86	82.94	27.02
4	48	82.88	26.87
5	110	83.88	27.03

6	117	84.46	28.01
7	138	84.76	26.98
8	193	84.20	27.50
9	225	89.89	29.82
10	80	77.78	23.02

Por otro lado, en la figura 4.18 se presentan las funciones de distribución acumulativa para las muestras bajo Modbus y DNP3. En la figura 4.18a se observan los resultados para DNP3, en donde se aprecia que el *delay* se incrementa con el aumento de distancia, en general el 90% de los paquetes enviados tienen un retraso que oscila entre 31 y 39 ms a excepción de la curva roja que representa el *delay* del medidor 10 conectado vía Ethernet con un retraso de 27 ms. Para el caso de Modbus ocurre algo similar, en la figura 4.18b se observa que a medida que la distancia aumenta también se incrementa el *delay*. En general el 90% de los paquetes enviados tienen un retraso que oscila entre 93 y 105 ms a excepción de la curva roja que representa el *delay* del medidor 10 conectado vía Ethernet con un retraso de 85 ms. Finalmente, en la figura 4.18c se presenta una comparativa del *delay* bajo Modbus y DNP3 en donde se evidencia que los retardos en Modbus son más altos que con DNP3. Para el *throughput* no se realizó una comparativa, ya que este parámetro permanece constante, es decir, en Modbus se requieren 2112 bps para la solicitud y 2480 bps para la respuesta y en DNP3 576 bps para solicitud y 2024 bps respuesta.



a. b.



c.

Figura 4.18. Comparativa de *delay* para lectura de cuatro bloques de memoria en los 10 medidores.

Según los resultados anteriores se puede concluir que en el rendimiento de los protocolos inciden diversos factores como tecnología de comunicación, distancia entre nodos y la cantidad de solicitudes necesarias para obtener la información.

En el caso de la tecnología de comunicación se demuestra que con Ethernet hay un aumento de rendimiento frente a Wi-Fi del 11% bajo Modbus y del 14% con DNP3. En cuanto a la distancia, comparando los datos del medidor más cercano al SCADA (48 m) frente al más lejano (225 m) se observa un incremento del 8% en el *delay*.

Por otra parte, los cambios más significativos en el rendimiento se obtienen según la cantidad de solicitudes necesarias para obtener la información. Cuando es suficiente con una lectura Modbus es 15% más efectivo que DNP3, pero cuando es necesario leer más parámetros y que se encuentren en diferentes ubicaciones de memoria DNP3 aumenta su rendimiento. En las pruebas realizadas se leyeron 27 variables de 4 ubicaciones diferentes con lo cual DNP3 mejoró su rendimiento en un 296%.

La anterior información es clave en la elección de los medidores inteligentes, pues generalmente un medidor que soporta DNP3 tiene un mayor costo que uno que soporte Modbus y cuando los puntos a medir aumenta las diferencias son aún más

evidentes. Se puede concluir que cuando la AMI o el sistema de gestión no requieran de la lectura de un número elevado de variables, Modbus cumple con los requerimientos, pero si el sistema es más robusto, con distancias más grandes y mayor cantidad de nodos la mejor opción es DNP3.

4.3.2. Análisis comparativo de pruebas de campo y resultados de simulación

En esta sección se presenta una comparación entre los datos obtenidos en las pruebas de campo y simulación. Inicialmente, en la figura 4.19 se presentan las curvas para el escenario en donde el SCADA solicita lectura de 27 variables vía Wi-Fi, las azules corresponden a los datos reales y las verdes a los datos de simulación. Los resultados obtenidos mediante la función de distribución acumulativa demuestran que el 90% de los paquetes enviados bajo DNP3 en la red real tienen un retraso menor o igual a 32 ms mientras que en la simulación es de 27 ms. Por otro lado, bajo Modbus el *delay* en los datos reales es de 95 ms frente a 90 ms en la simulación.

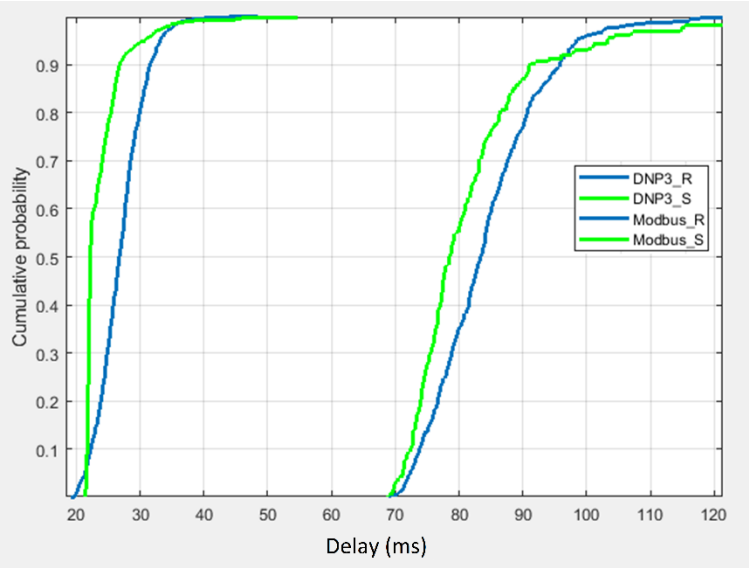


Figura 4.19. Comparativa de *delay* en datos reales y de simulación mediante tecnología Wi-Fi.

Por otra parte, cuando la comunicación se realizó mediante Ethernet se obtuvieron los resultados que se presentan en la figura 4.20, en este caso se observa que la distribución de los datos de simulación se restringe solo a algunos valores. Mediante las curvas de función de distribución acumulativa se observa que el 90% de los

paquetes enviados bajo DNP3 en la red real tienen un retraso menor o igual a 27 ms mientras que en la simulación es de 22 ms. Por otro lado, bajo Modbus el *delay* en los datos reales es de 85 ms frente a 74 ms en la simulación.

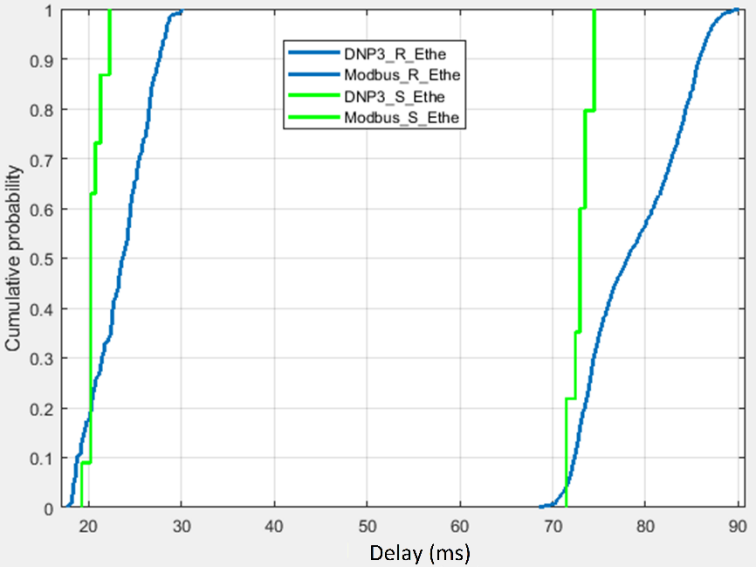


Figura 4.20. Comparativa de *delay* en datos reales y de simulación mediante tecnología Ethernet.

Finalmente, en la figura 4.21 se presenta una comparativa del *delay* en la comunicación de los diez medidores bajo DNP3 y Modbus. Las curvas en azul representan las medidas reales, las verdes los datos de simulación y las rojas la comunicación mediante Ethernet. Como se observa, los datos obtenidos en la simulación son coherentes y se ajustan a los reales, se aprecia mediante las curvas rojas que la comunicación vía Ethernet tiene el menor *delay*, por otra parte, en las comunicaciones vía Wi-Fi se demuestra que a medida que la distancia aumenta también el *delay*, la curva azul situada más a la derecha representa en ambos casos el *delay* del medidor 9 que se encuentra a mayor distancia del SCADA, de igual manera la curva verde situada más a la derecha representa los datos de simulación para el medidor 9.

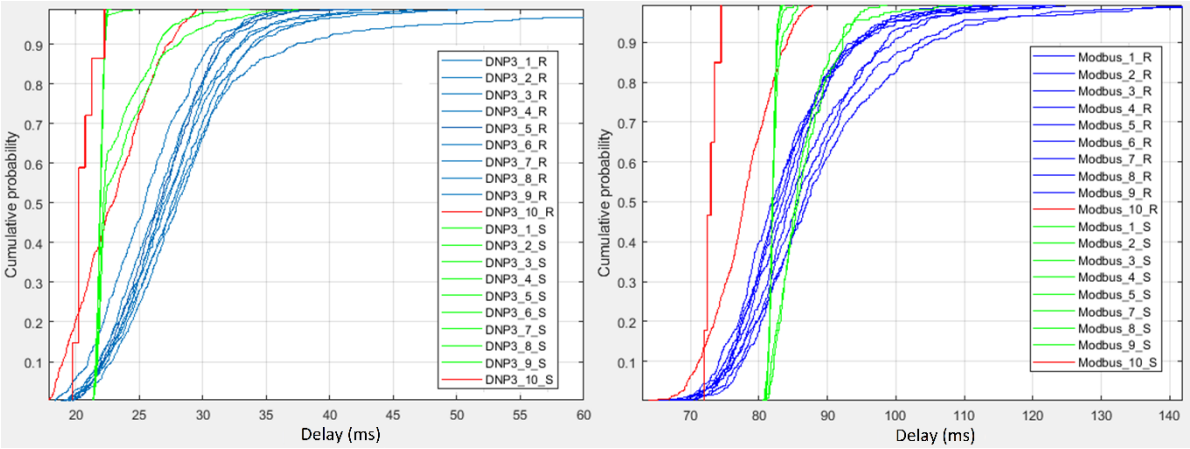


Figura 4.21. Comparativa de *delay* en datos reales y de simulación para los 10 medidores.

4.3.3. *Delay* en función de la distancia

Como se observó en la sección anterior el *delay* aumenta a medida que se incrementa la distancia entre el medidor inteligente y el SCADA. Para determinar el comportamiento del *delay* se simuló el escenario de la figura 4.22, este consta de dos nodos conectados vía Wi-Fi, uno de ellos actúa como medidor inteligente y el otro como SCADA, este último envía una solicitud de lectura cada segundo. Se realizaron simulaciones con diferentes distancias entre los nodos, los resultados se presentan en la tabla 4.13.

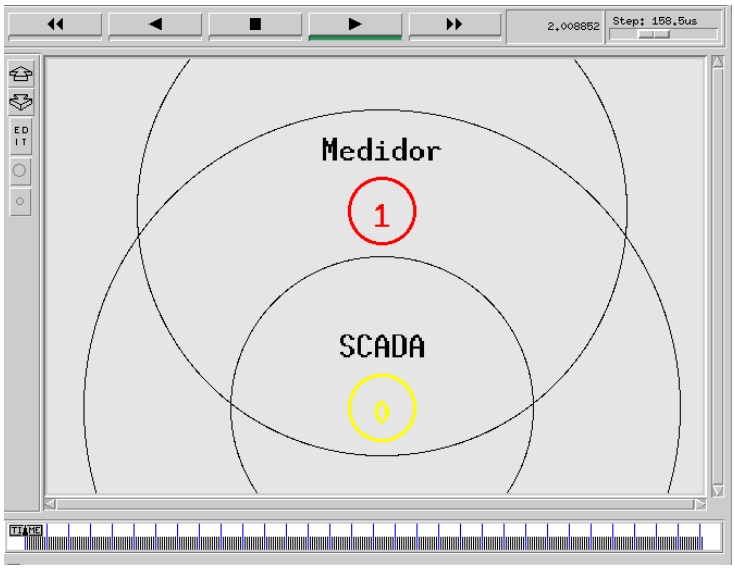


Figura 4.22. Topología propuesta para el estudio de efectos de distancia.

Tabla 4.13. Valores de *delay* en función de la distancia al SCADA.

Distancia [m]	***Delay* [ms]**
0	18.93452
10	18.93472
20	18.93492
30	18.93512
40	18.9353
50	18.93552
60	18.93572
70	18.93592
80	18.93959
90	18.94151
100	18.95426
110	18.96922
120	19.01002
130	19.08127
140	19.20632
150	19.32639
160	19.46787
170	19.67048
180	19.88119
190	20.28466
200	20.81463
208	21.23
209	21.39

Inicialmente, el medidor se encuentra en la misma ubicación que el SCADA, obteniendo el *delay* más bajo, las siguientes medidas se tomaron incrementando la distancia en intervalos de 10 metros. En la figura 4.23 se presentan los resultados, en donde se observa que el *delay* se mantiene prácticamente constante hasta una distancia de 120 metros, a partir de ahí el *delay* sufre incrementos de forma exponencial que degradan rápidamente la comunicación y que afectan el desempeño de los protocolos y de la AMI en general.

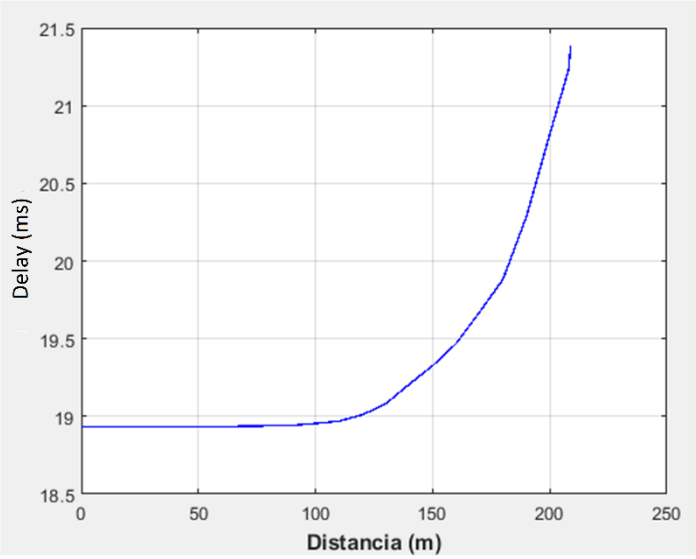


Figura 4.23. Gráfica de distancia vs *delay*.

4.3.4. *Delay* en función de la cantidad de Bytes transmitidos

Otro factor que se observó que afecta al *delay* es la cantidad de Bytes que se transmite, para determinar sus efectos se utilizó el mismo escenario del estudio anterior, es decir, dos nodos conectados vía Wi-Fi mediante DNP3. Se tomaron medidas de *delay* según la cantidad de Bytes que se presenta en la tabla 4.14, aquí se observa que a medida que se incrementa la cantidad de Bytes transmitidos el *delay* también aumenta.

Tabla 4.14. Valores de *delay* en función de los Bytes transmitidos.

Bytes enviados	***Delay* [ms]**
53	20.35248
62	20.42456
74	20.52067
84	20.60076
96	20.69687
108	20.79297
118	20.87306
130	20.96917
142	21.06527

152	21.14536
164	21.24147
174	21.32156
186	21.41767
198	21.51377
208	21.59386
220	21.68997
232	21.78607
242	21.86616
254	21.96227
264	22.04236
276	22.04236
288	22.23457
298	22.31466
310	22.41077
322	22.50687
332	22.58696
344	31.4718
356	31.5678
368	31.6478
380	31.7438
390	31.8398

En la figura 4.24 se presenta una gráfica de los resultados, en ella se observa que a medida que aumenta el número de Bytes transmitidos el *delay* se incrementa linealmente. En el momento en que los datos alcanzan la cantidad máxima de Bytes que se puede transmitir (292 Bytes más 40 Bytes de cabecera de TCP/IP), el mensaje se fracciona y es necesario realizar dos transmisiones para el envío de la información, este proceso se detalla en la gráfica mediante el escalón que indica un aumento abrupto en el *delay* a consecuencia del aumento de Bytes transmitidos y por lo tanto una disminución en el rendimiento del protocolo.

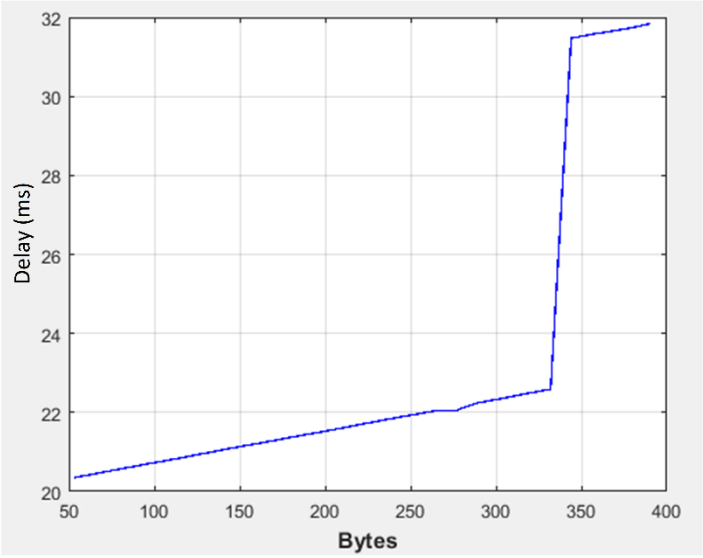


Figura 4.24. Gráfica de número de Bytes vs *delay* en NS-2.

Capítulo 5

Conclusiones y Recomendaciones

5.1. Conclusiones.

En este estudio, se evaluaron dos protocolos de comunicación con el fin de conocer su desempeño en aplicaciones de redes inteligentes, dicha evaluación se desarrolló considerando el *delay* y *throughput,* métricas que permiten conocer el comportamiento de los protocolos.

El estudio se realizó en dos fases, la primera mediante pruebas de campo y la segunda por medio de simulación. Para la evaluación por pruebas de campo se diseñó e implementó una infraestructura de medición avanzada AMI, que facilita la recolección de información de la red eléctrica dentro del Campus de la Universidad de Nariño.

Una AMI se compone de una red de comunicaciones, un sistema de medición (medidores inteligentes) y un centro de gestión donde se almacena la información y se toman decisiones de control. El diseño de la AMI inicia con el estudio del entorno donde se realizará la implementación, considerando aspectos como topografía del terreno, cantidad de puntos de medida y su distancia con el centro de recolección de información que generalmente es un sistema SCADA.

El diseño de la red de comunicaciones se basa en la elección de las tecnologías y la topología de implementación. En este punto se puede concluir que la topología en

estrella es una buena alternativa, debido a que su implementación es más sencilla y necesita menos hardware.

Otro aspecto importante a considerar en el diseño de la red de comunicaciones son las tecnologías de comunicación. En este punto se concluye que las tecnologías cableadas como Ethernet ofrecen mayor eficiencia pero requieren de mayor infraestructura para su implementación y tiene una limitante de distancia sin utilizar repetidores, por su parte las tecnologías inalámbricas puede reducir considerablemente el costo y tiempo de implementación, ya que generalmente requieren de menor infraestructura para su instalación.

Por otra parte, los medidores inteligentes ofrecen diversas características en cuanto a los parámetros de medición y se deben seleccionar según los requerimientos del sistema que se esté implementando. Un aspecto importante para su elección es el protocolo de comunicaciones que soporte, pues existen los propietarios que permiten que el dispositivo se integre solo con equipos y *software* del mismo fabricante. Por su parte, los que soportan protocolos abiertos como Modbus y DNP3 son fácilmente integrables a cualquier sistema de gestión.

La red de comunicaciones implementada se verificó bajo ciertos parámetros QoS, con el fin de comprobar si cumple con los requerimientos que exige una AMI, el test se desarrolló bajo las métricas de *delay*, ancho de banda, *throughput* y pérdida de paquetes. Los resultados fueron satisfactorios demostrando que la red de comunicaciones puede soportar aplicaciones de control y protección, pues el *delay* de los enlaces se encuentra alrededor de los 4.5 ms.

Los simuladores de redes permiten realizar estudios de comportamiento, detección de errores y posibles ampliaciones de la red. Para realizar un modelamiento se deben seguir algunos pasos generales como diseño del modelo, verificación de funcionamiento y validación de la implementación.

Este trabajo contribuyó con el modelado de los protocolos Modbus y DNP3 en el simulador de redes NS-2, permitiendo la simulación de escenarios de comunicación bajo tecnologías Ethernet y Wi-Fi. El modelo presentado es una buena aproximación

que refleja con un buen grado de confianza el comportamiento dentro de la red de comunicaciones real.

La evaluación de desempeño con pruebas de campo se desarrolló utilizando la herramienta Wireshark, que permite obtener información detallada del flujo de datos dentro de una red. Con los archivos generados es posible obtener características de las comunicaciones y de los protocolos involucrados.

Con los resultados obtenidos se puede concluir que los protocolos Modbus y DNP3, bajo tecnología Ethernet y Wi-Fi no son adecuados para sistemas de protección, debido a que el *delay* en la entrega de información sobrepasa los valores aceptados, pero su desempeño es óptimo para funciones de monitoreo en tiempo real y para sistemas SCADA.

En conclusión, la comparativa de los protocolos Modbus y DNP3 demuestra que bajo ciertas condiciones Modbus presenta mejor desempeño que DNP3, en especial cuando el requerimiento de información no es alto. Por su parte DNP3 por ser más robusto es más eficiente cuando se transmite mayor cantidad de información. Esta información es importante para la elección de los medidores inteligentes, pues generalmente tiene mayor costo un medidor que incorpore DNP3 a otro que soporte Modbus.

5.2. Recomendaciones.

Como continuidad de este trabajo se sugiere estudiar el comportamiento de Modbus y DNP3 bajo otras tecnologías que se utilizan en las *microgrids* como PLC, Zigbee y WiMax.

Para el modelamiento de los protocolos fue necesario un ajuste en los valores de salida, pues en el simulador no se considera el tiempo de procesamiento que necesita un medidor inteligente antes de enviar una respuesta. En este caso el ajuste se realizó según las especificaciones del medidor Satec EM133 y podrían no ajustarse a los valores de otros dispositivos.

Los modelos desarrollados en este trabajo de los protocolos Modbus y DNP3 para el simulador NS-2, son prometedores. Los resultados de las evaluaciones para los escenarios punto a punto y multipunto, demostraron que el mecanismo propuesto puede ser utilizado para simular otros escenarios de aplicaciones en redes inteligentes.

Otro trabajo a futuro sería evaluar la cantidad máxima de medidores en el escenario multipunto, así como el nivel de tráfico Modbus y DNP3 soportable. Es evidente que cuanto mayores sean estos dos factores, mayor será el *delay* experimentado por los mensajes y por lo tanto es esencial determinar hasta qué punto el modelo propuesto responde adecuadamente a determinado tipo de aplicación.

Un inconveniente de las simulaciones de redes inalámbricas en NS-2 es que solo está disponible la antena omnidireccional, lo que no siempre se ajusta a las necesidades de diseño. Sería interesante modelar otros tipos de antena que permitan un modelamiento más exacto.

Referencias

[1] C. Lo and N. Ansari, “The Progressive Smart Grid System from Both Power and Communications Aspects,” in IEEE Communications Surveys & Tutorials, 2012, vol. 14, no. 3, pp. 799-821.

[2] S. Safdar et al., “*A Survey on Communication Infrastructure for Micro-grids*,” School of EECS, Oregon State University y Qatar University.

[3] A. Remanidevi, M. V. Ramesh y V. P. Rangan, “*High performance communication architecture for Smart distribution power grid in developing nations*,” Springer Science+Business Media, New York, 2016.

[4] G. Karagiannis et al., “*Performance of LTE for Smart Grid Communications*,” Springer-Verlag Berlin Heidelberg, 2011.

[5] U.S. Department of Energy. [En línea]. Available: https://www.energy.gov/

[6] J. García, “Recent progress in the implementation of AMI projects: Standards and communications technologies,” in International Conference on Mechatronics, Electronics and Automotive Engineering, Cuernavaca, México, 2015, pp. 251.256.

[7] W. Wand, Y. Xu y M. Khanna, “*A survey on the communication architectures in smart grid*,” Department of Electrical and Computer Engineering, North Carolina State University, United States, 2011.

[8] *IEEE 2030 IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads, IEEE, 2011.*

[9] N. Q. Do, H. S. Ong, L. C. Lai, Y. X. Che and X. J. Ong, “Open-Source Testing Tools for Smart Grid Communication Network,” in Proc. IEEE Conference on Open Systems (ICOS), Sarawak, Malaysia, Dec. 2013, pp. 156-161.

[10] M. C. Jeruchim, P. Balaban y K. S. Shanmugan, *Simulation of Communication Systems*, 2nd Edition, New York, KLUWER ACADEMIC PUBLISHERS, 2002.

[11] R. de Miranda, "Análise do Escalonamento de Redes Ad Hoc IEEE 802.11 através de medidas de Vazão e Atraso usando o NS-2," Tesis de pregrado, Departamento de Sistemas e Computação, Escola Politécnica de Pernambuco, Recife, Brasil, 2008.

[12] K. Tan, D. Wu and A. Chan,"Comparing Simulation Tools and Experimental Testbeds for Wireless Mesh Networks," Department of Computer Science, University of California,USA, 2010.

[13] A. Echeverry, "Wireless Network Simulation with NS-2," Scientia et Technica, Universidad Tecnológica de Pereira, 2010.

[14] L. R. Prete, “Análise e desempenho de redes de acesso sem fio,” Tesis de Maestria, Facultad de Ingenieria, Universidad Estatal Paulista- UNESP, Ilha Solteira, Brasil, 2011.

[15] D. N. Quang et al., “*Performance Testing Framework in a Heterogeneous and Hybrid Smart Grid Communication Network*,” Research Journal of Applied Sciences, Engineering and Technology 6(23): 4506-4518, 2013.

[16] O. Vondrous, P. Macejko, T. Hegr, and Z. Kocur, “Testing Methodology for Performance Evaluation of Communication Systems for Smart Grid,” Department of Telecommunication Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic, 2016.

[17] X. Lu et al., “*On Network Performance Evaluation toward the Smart Grid: A Case Study of DNP3 over TCP/IP*,” IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings, 2011.

[18] X. Lu, W. Wang and J. Ma, *"*An Empirical Study of Communication Infrastructures Towards the Smart Grid: Design, Implementation, and Evaluation,*" in* IEEE transactions on smart grid, 2013, vol. 4, no. 1, *pp. 170-183.*

[19] D. Ramirez, S. Céspedes, C. Becerra and C. Lazoz, "Performance Evaluation of Future AMI Applications in Smart Grid Neighborhood Area Networks," in IEEE COLCOM 2015.

[20] A. Ortega et al., "*Performance Evaluation of the DNP3 Protocol for Smart Grid Applications over IEEE 802.3/802.11 Networks and Heterogeneous Traffic*," Recent Advances in Communications, Universidade Estadual de Mato Grosso do Sul (UEMS), Brasil.

[21] A. Ortega y A. Akira Shinoda, "*Simulation in NS-2 of DNP3 Protocol Encapsulated over TCP/IP in Smart Grid Applications*," Shinoda estao com UNESP, Campus Ilha Solteira, SP Brasil, 2013.

[22] A. Ortega y C. M. Schweitzer, "*Performance Analysis of Smart Grid Communication Protocol DNP3 over TCP/IP in a Heterogeneous Traffic Environment*," Department of electrical engineering Ilha Solteira, Brasil, 2013.

[23] A. Ortega, A. Shinoda, M. Schweitzer, A. Ortega and L. Prete, "Análise de Desempenho de Rede *Smart Grid* Protocolo de Comunicação DNP3 Sobre IEEE 802.11," in The 9th latin-american congress on electricity generation and transmission - CLAGTEE 2013.

[24] A. Ortega, A. Shinoda, C. Schweitzer and A. Ortega, "DNP3 integration performance with IEEE 802.11 in Smart Grid Applications Through the CDF of delay and Jitter," This work had the financial support of the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Padtec S/A, *2013.*

[25] A. Ortega, A. Shinoda and A. Ortega, "Simulação Computacional do Protocolo de Comunicação em Aplicações de *Smart Grid," in Anais do Congresso de Matemática Aplicada e Computacional, CMAC Sudeste, 2013, pp. 370-375*

[26] A. Ortega et al., "*Proposal DNP3 Protocol Simulation on NS-2 in IEEE 802.11g Wireless Network Ad Hoc Over TCP/IP in Smart Grid Applications*," 2015 IEEE PES Innovative Smart Grid Technologies Latin America (ISGT LATAM), 2015.

[27] A. Ortega y C. M. Schweitzer, "*Simulation of the DNP3 Protocol Over TCP/IP on a Network IEEE 802.11g Ad-hoc With Smart Meter*," Universidade Estadual de Mato Grosso do Sul (UEMS), Brasil, 2016.

[28] A. Ortega, A. Shinoda and C. Schweitzer, "Análise e desempenho do protocolo de comunicação dnp3 sobre redes wireless em aplicações smart grid," in XIII International Conference on Engineering and Technology Education, 2014, Guimarães, PORTUGAL, pp. 153-157.

[29] C. C. da Silva Pereira et al., "*A NS-2 simulation model for DNP3 protocol over IEEE 802.15.4 wireless protocol toward low cost simulation of Smart Grid applications*," Universidade Estadual Paulista "Julio de Mesquita Filho", São Paulo, Brasil, 2014.

[30] A. Ortega, A. Shinoda, M. Schweitzer, F. Granelli and A. Ortega, "Performance Analysis of the DNP3 Protocol Over IEEE 802.11g Wireless Network in Smart Grid Application through of the Simulation in the NS-2," in the XI latin-american congress electricity generation and transmission - CLAGTEE 2015.

[31] A. Ortega, A. Shinoda and C. Schweitzer, "Simulação Aplicada à Infraestrutura da Rede Elétrica com *Smart Grid* Empregando Comunicação Sem Fio," in ITA, 2013, pp. 183-187.

[32] A. Ortega, A. Shinoda, C. Schweitzer and F. Granelli, "Simulation of the DNP3 communication protocol over tcpip protocol on a 802.11g wireless network in smart grid applications," in WCSEIT - world congres on systems engineering and information technology – Challenges, Practices and Technologies in the Era of Information, Vigo, España, 2015.

[33] M. R. Sahraei, "*Ns-Modbus: Integration of Modbus with ns-3 Network Simulator*," tesis de maestría, SIMON FRASER UNIVERSITY, British Columbia, Canada, 2013.

[34] B. Kim, D. Lee and T. Choi, “Performance Evaluation for Modbus/TCP Using Network Simulator NS3,” in IEEE, 2015.

[35] G. A. W. Gómez, “Caracterización tecnológica de la topología de una sistema de gestión energética residencial”, Universidad Industrial de Santander, Bucaramanga, 2012.

[36] Y. Z. Luhua, “Effects of advanced metering infrastructure (AMI) on relations of Power Supply and Application on Smart Grid”, Beijing, North China: Grid Company Limited Metering Centre, 2011. [En línea]. Available: http://ieeexplore.ieee.org/document/5736035/

[37] H. Sui, H. Wang, M. Lu, and W. Lee, “An AMI System for the Deregulated Electricity Markets,” IEEE Trans.on Industry Applications, Dec. 2009, vol. 45, no. 6, pp. 2104–2108.

[38] A. Ortega, “Análise de desempenho de redes de comunicação wireless em aplicações de smart grid,” Universidade Estadual Paulista “Júlio de Mesquita filho”, Faculdade de engenharia, Ilha Solteira, Brasil, 2015.

[39] M. Islam y H. Hee Lee, “*Microgrid Communication Network with Combined Technology*,” en 5th International Conference on Informatics, Electronics and Vision, 2016.

[40] F. J. Hartmann et al., “Generating realistic smart grid communication topologies based on real-data,” in IEEE International Conference on Smart Grid Communications, Venice, Italy, 2015, pp. 428 – 433.

[41] H. Farooq, L. T. Jung, “Choices available for implementing smart grid communication network,” in International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 2014, pp. 1-5.

[42] J. Zheng y D. W. Gao, “Smart meters in smart grid: an overview,” Proc. IEEE Conference Green Technologies, Denver, USA, 2013, pp. 57-64.

[43] M.A. Hammoudeh, "*Comparative analysis of communication architectures and technologies for smart grid distribution network*," Tesis de Maestría, Universidad de Denver, Colorado, 2012.

[44] NCR, Nuclear Regulatory Comission, "U.S. NRC - Nuclear Regulatory Commission," [En línea]. Available: http://www.nrc.gov/.

[45] S. K. Aggarwal, "A proposed communications infrastructure or the smart grid," Proc. Innovative Smart Grid Technologies, Gaithersburg, USA, 2010.

[46] "Modbus parte II: comunicación a través de una red rs-485," 2012. [En línea]. Available: http://tecdigitaldelbajio.com/blog/25-modbus-parte-ii-omunicacion-a-traves-de-una-red-rs-485.html.

[47] Z. Qin, "*A survey of networking issues in smart grid*," [En línea]. Available: http://www.cse.wustl.edu/~jain/cse570-13/ftp/smrtgrid.pdf

[48] M. A. Gao, "A review of voltage control in smart grid and smart metering technologies on distribution networks," Universities' Power Engineering Conference (UPEC), Soest, Germany, 2011, pp. 1-5. [En línea]. Available: http://ieeexplore.ieee.org/document/6125583/

[49] C. M. Poveda, C. Medina y M. Zambrano, "Tecnologías de comunicación para redes de potencia inteligentes de media y alta tensión", Prisma tecnológico, vol. 5, n° 1, pp. 29-32, 2014.

[50] J. Pérez y A. Gardey, "Definición de protocolo de comunicación," [En línea]. Available: http://definicion.de/protocolo-de-comunicacion/

[51] Modbus application protocol specification v1.1b3, 2012.

[52] ACROMAG INCORPORATED, "Introduction to Modbus TCP/IP," USA, 2005.

[53] Modbus messaging on TCP/IP implementation guide v1.0b, 2006.

[54] M. Cerezuela, "Protocolos de comunicación en automatización industrial," tesis de pregrado, Escuela técnica superior de ingeniería de telecomunicación, Universidad Politécnica de Cartagena, 2014.

[55] DNP3 Primer, DNP Users Group, 2005.

[56] Protocol Translator DNP3 User Manual, MULTITRODE, 2007.

[57] G. Clarke and D. Reynders, "Pratical Modern SCADA Protocol,"First, Ed. Vivek Mehra:Mumbai, India: Elsevier, 2004, pp. 66-164.

[58] Keiser, G. (2002), Local Area Network, Second Edition, McGraw Hill, New York

[59]802.3-2015 - IEEE Standard for Ethernet, IEEE, 2016.

[60] Trama inalámbrica 802.11, “Introducción a redes ,” Cisco Networking Academy. [En línea]. Available: http://www.itesa.edu.mx/netacad/introduccion/course/module4/#4.4.4.8

[61] B. Mitchell, 802.11a, "The 802.11 family explained, from 802.11a through 802.11az," Octubre 2018. [En línea]. Available: https://www.lifewire.com/bradley-mitchell-816228

[62] M. Andrade, "Análise de desempenho do protocolo DNP3 encapsulado sobre PLC para aplicações em Redes Inteligentes," Universidad Federal de Sergipe, Programa de Pós-Graduação em Engenharia Elétrica, São Cristóvão, Brasil, 2017.

[63] J. M. Herrera, “NS2 - Network Simulator,” Estudiante de Ingeniería Civil Informática, Valparaíso, 2004.

[64] Manual NAM Network Animator. [En línea]. Available: http://www.isi.edu/nsnam/nam/nam2.html#TUTORIAL

[65] Calypso Barnes. Verification and validation of wireless sensor network protocol properties through the system's emulation. Networking and Internet Architecture [cs.NI]. Université Côte d'Azur, 2017. English. .

[66] P. Oppenheimer, "Top-Down Network Design, a systems analysis approach to enterprise network desing," CISCO, Third Edition, 2010.

[67] R.G. Sargent, "VERIFICATION AND VALIDATION OF SIMULATION MODELS," Department of Electrical Engineering and Computer Science, Syracuse University, U.S.A., 2011.

[68] J. Silva, "Understanding wireless topologies for smart grid applications", Grid-Interop Forum, Phoenix, USA, 2011. [En línea]. Available: https://www.gridwiseac.org/pdfs/forum_papers11/silva_paper_gi11.pdf

[69] B. Gou, "Generalized Integer Linear Programming Formulation for Optimal PMU Placement," in IEEE Transactions on Power Systems, 2008, pp. 1099-1104.

[70] D. L. Alvarez, J. Reyes, W. Montaño and E. Parra, "Sistema de Gestión de Energía en tiempo real del campus de la Universidad Nacional de Colombia", Mundo Eléctrico vol. 101, Bogotá 2015, pp. 48-57. [En línea]. Available: https://www.neplan.ch/wp-content/uploads/2015/10/48-57-WEB.pdf

[71] V. K. Sood, D. Fischer, J. M. Eklund, and T. Brown, "Developing a communication infrastructure for the smart grid," in IEEE Electrical Power Energy Conference, Montreal, Canada, 2009, pp. 1–7.

[72] S. Buettrich y A. Escudero, "Topología e Infraestructura Básica de Redes Inalámbricas," TRICALCAR, Octubre, 2007.

[73] R.Nice, "ANSI and IEEE Standards for Metering," Metering Task Force, PJM, 2015.

[74] SATEC EM133 sub-meter, Electrical Comms Data ECD. [En línea]. Available: https://www.ecdonline.com.au/content/electrical-distribution/hot-product/satec-em133-sub-meter-621961450#axzz5eTJqnGaV

[75] SATEC, “EM13x Series SMART MULTIFUNCTION METER Installation and Operation Manual,” [En línea]. Available: SATEC https://satec-global.com.au/manuals/em133/EM132-133.pdf

[76] Tecnología Digital del Bajío, “Modbus”, 2012. [En línea]. Available: http://tecdigitaldelbajio.com/blog/25-modbus-parte-ii-comunicacion-a-traves-de-una-red-rs-485.html.

[77] A. Cataliotti et al., “Experimental Evaluation of an Hybrid Communication System Architecture for Smart Grid Applications,” Department of Energy, Information engineering and Mathematic Models, Università di Palermo, Palermo, Italy, 2015, pp. 96-101.

[78] O. Ramírez, “Estudios de desempeño de escenarios SCADA que utilizan el Protocolo DNP3,” Tesis de Maestría, Universidad de los Andes, Colombia, 2012.

[79] C. C. Da Silva “Modelo de simulação NS-2 para o protocolo DNP3 sobre o protocolo de rede sem fio IEEE 802.15.4 para simulação de baixo custo de aplicação Smart Grid,” Universidade Estadual Paulista “julio de mesquita filho”, Faculdade de Engenharia, Ilha Solteira, Brasi, 2015.

[80] A. Mehta, R. Jain and V. Somani, "Comparison of different Radio Propagation Models with and without Black Hole Attack on AODV Routing Protocol in MANET," in Proc. International Journal of Computer Applications, Volume 61– No.1, Jan. 2013, pp. 20-24.

[81] Radio Propagation Models Implemented in Ns2.

[82] M. Ullah, W. Ahmad, “Evaluation of Routing Protocols in Wireless Sensor Networks,” Tesis de Maestría, Blekinge Institute of Technology Soft Center, SWEDEN, 2009.

[83] M. Skariah et al., "An Analysis on the Performance Evaluation of Routing Protocols in Wi-Fi/802.11b Network," International Journal of Computer Science and Mobile Computing, Vol.3 Issue.3, March- 2014, pg. 409-416

[84] M. D. Byrne, "How Many Times Should a Stochastic Model Be Run? An Approach Based on Confidence Intervals," Departments of Psychology and Computer Science, Rice University, Houston, USA.

[85] P. Zand et al., "Implementation of WirelessHART in the NS-2 Simulator and Validation of Its Correctness," OPEN ACCESS sensors, 2014, pp. 8633-8668.

[86] H. Yang, L. Chengy and M. Chuahz, “Modeling DNP3 Traffic Characteristics of Field Devices in SCADA Systems of the Smart Grid,” Department of Computer Science and Engineering, Lehigh University, Bethlehem, USA.

[87] N. Goldenberg and A. Wool, “Accurate modeling of Modbus/TCP for intrusión detection in SCADA systems,” School of Electrical Engineering, Tel Aviv University, Israel, 2013.

Apéndice

Códigos Computacionales

Apéndice A. Archivo de prueba dnp3_2nodos_inal.tcl

```
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 5.5
Antenna/OmniAntenna set Gt_ 8.0
Antenna/OmniAntenna set Gr_ 8.0

Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-14
Phy/WirelessPhy set RXThresh_ 1.995262315e-13
Phy/WirelessPhy set Pt_ 0.63
Phy/WirelessPhy set freq_ 2.4e+9

Propagation/Shadowing set pathlossExp_ 2.0  ;# path loss exponent B
Propagation/Shadowing set std_db_ 4.0       ;# shadowing deviation (odB)
Propagation/Shadowing set dist0_ 1.0        ;# reference distance (m)
Propagation/Shadowing set seed_ 0           ;# seed for RNG

set val(chan)           Channel/WirelessChannel    ;# channel type
set val(prop)           Propagation/Shadowing      ;# radio-propagation
model FreeSpace TwoRayGround Shadowing
set val(netif)          Phy/WirelessPhy            ;# network interface type
set val(mac)            Mac/802_11                 ;# MAC type
set val(ifq)            Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)             LL                         ;# link layer type
set val(ant)            Antenna/OmniAntenna        ;# antenna model
set val(ifqlen)         1000                       ;# max packet in ifq
set val(nn)             2                          ;# number of mobilenodes
set val(rp)             DumbAgent                  ;# routing protocol
set val(xx)       350                   ;# X dimension of topography
set val(yy)             300                    ;# Y dimension of topography
set val(vv)       3                     ;# veces que lee a las outstation

# Inicializa variables Globales
set ns            [new Simulator]
```

```
#Creacion archivo Nam
set namfile [open salida_dnp3_2nodos.nam w]
$ns namtrace-all-wireless $namfile $val(xx) $val(yy)
#Creacion archivo Trace
set tracefile [open salida_dnp3_2nodos.tr w]
$ns trace-all $tracefile

proc stop {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam salida_dnp3_2nodos.nam &
    exec gawk -f delay.awk salida_dnp3_2nodos.tr &
    #exit 0
}

# Configurar objetos de topografía
set topo       [new Topography]
$topo load_flatgrid $val(xx)  $val(yy)

# Crea God (General Operations Director)
create-god $val(nn)

#  Configuraciones de Escenario
$ns node-config -adhocRouting $val(rp) \
 -llType $val(ll) \
 -macType $val(mac) \
 -ifqType $val(ifq) \
 -ifqLen $val(ifqlen) \
 -antType $val(ant) \
 -propType $val(prop) \
 -phyType $val(netif) \
 -channelType $val(chan) \
 -topoInstance $topo \
 -agentTrace ON \
 -routerTrace ON \
 -macTrace OFF \
 -movementTrace ON

# Creear nodos móviles [$ val (nn)]
for {set i 0} {$i < $val(nn) } {incr i} {
  set node_($i) [$ns node]
  $node_($i) random-motion 0 ;# desactivar el movimiento aleatorio
}

# Coordenadas iniciales (X, Y, Z = 0) para nodos móviles
$node_(0) set X_ 180.0
$node_(0) set Y_ 80.0
$node_(0) set Z_ 0.0
$node_(0) color black

$node_(1) set X_ 180.0
$node_(1) set Y_ 280.0
$node_(1) set Z_ 0.0
$node_(1) color black
```

```
$ns initial_node_pos $node_(0) 50 ;#tamaño de nodos
$ns initial_node_pos $node_(1) 50

$ns at 0.0 "$node_(0) color yellow" ;#color de nodos
$ns at 0.0 "$node_(0) label Nodo_1" ;#texto de nodos
$ns at 0.0 "$node_(1) color blue"
$ns at 0.0 "$node_(1) label Nodo_2"

#Configurar flujo de tráfico entre nodos
set tcp1 [new Agent/TCP/SimpleTcp]
set tcp2 [new Agent/TCP/SimpleTcp]

$ns attach-agent $node_(0) $tcp1
$ns attach-agent $node_(1) $tcp2
$ns connect $tcp1 $tcp2

set encap [new Application/TcpAppmod $tcp1]
set encap2 [new Application/TcpAppmod $tcp2]
$encap connect $encap2

# Crea estacion maestro
set master [new DNP3master]
# Define tiempo de espera para retransmision
$master set retrytimer_ 1
# Define prob. de perdida de paquetes valor aleatorio en intervalo de 0 a
10
$master set proberror_ 0
# Crea estacion Esclavo
set client [new DNP3client 1]
# Define tiempo de espera para retransmision
$client set retrytimer_ 1
# Define prob. de perdida de paquetes valor aleatorio en intervalo de 0 a
10
$client set proberror_ 1
$master connect $client $encap
$client connect $master $encap2

for {set j 0} {$j < $val(vv) } {incr j} {
      $ns at $j.0 "$master read $client"
}

$ns at $val(vv) "stop"

$ns run
```

Apéndice B. Archivo de prueba para comunicación inalámbrica de dos nodos bajo Modbus.

```
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 5.5
Antenna/OmniAntenna set Gt_ 8.0
Antenna/OmniAntenna set Gr_ 8.0

Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-14
Phy/WirelessPhy set RXThresh_ 5.01187233e-12
Phy/WirelessPhy set Pt_ 0.63
Phy/WirelessPhy set freq_ 2.4e+9

Propagation/Shadowing set pathlossExp_ 2.0  ;# path loss exponent B
Propagation/Shadowing set std_db_ 4.0       ;# shadowing deviation (odB)
Propagation/Shadowing set dist0_ 1.0        ;# reference distance (m)
Propagation/Shadowing set seed_ 0           ;# seed for RNG


set val(chan)           Channel/WirelessChannel    ;# channel type
set val(prop)           Propagation/Shadowing      ;# radio-propagation
      model FreeSpace TwoRayGround Shadowing
set val(netif)          Phy/WirelessPhy            ;# network interface type
set val(mac)            Mac/802_11                 ;# MAC type
set val(ifq)            Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)             LL                         ;# link layer type
set val(ant)            Antenna/OmniAntenna        ;# antenna model
set val(ifqlen)         1000                       ;# max packet in ifq
set val(nn)             2                          ;# number of mobilenodes
set val(rp)             DumbAgent                  ;# routing protocol
set val(xx)       350                  ;# X dimension of topography
set val(yy)             300                  ;# Y dimension of topography
set val(vv)       9                    ;# veces que lee a las outstation

# Inicializa variables Globales
set ns            [new Simulator]

#Creacion archivo Nam
set namfile [open salida_mod_2nodos_inal.nam w]
$ns namtrace-all-wireless $namfile $val(xx) $val(yy)
#Creacion archivo Trace
set tracefile [open salida_mod_2nodos_inal.tr w]
$ns trace-all $tracefile

proc stop {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam salida_mod_2nodos_inal.nam &
     exec gawk -f delay.awk salida_mod_2nodos_inal.tr &
    #exit 0
```

```
}
# Configurar objetos de topografía
set topo       [new Topography]
$topo load_flatgrid $val(xx)  $val(yy)

# Crea God (General Operations Director)
create-god $val(nn)

#  Configuraciones de Escenario
$ns node-config -adhocRouting $val(rp) \
 -llType $val(ll) \
 -macType $val(mac) \
 -ifqType $val(ifq) \
 -ifqLen $val(ifqlen) \
 -antType $val(ant) \
 -propType $val(prop) \
 -phyType $val(netif) \
 -channelType $val(chan) \
 -topoInstance $topo \
 -agentTrace ON \
 -routerTrace ON \
 -macTrace OFF \
 -movementTrace ON


# Creear nodos móviles [$ val (nn)]
for {set i 0} {$i < $val(nn) } {incr i} {
  set node_($i) [$ns node]
  $node_($i) random-motion 0 ;# desactivar el movimiento aleatorio
}

# Coordenadas iniciales (X, Y, Z = 0) para nodos móviles
$node_(0) set X_ 80.0
$node_(0) set Y_ 180.0
$node_(0) set Z_ 0.0
$node_(0) color black

$node_(1) set X_ 300.0
$node_(1) set Y_ 180.0
$node_(1) set Z_ 0.0
$node_(1) color black

$ns initial_node_pos $node_(0) 50 ;#tamaño de nodos
$ns initial_node_pos $node_(1) 50

$ns at 0.0 "$node_(0) color yellow" ;#color de nodos
$ns at 0.0 "$node_(0) label Cliente" ;#texto de nodos
$ns at 0.0 "$node_(1) color blue"
$ns at 0.0 "$node_(1) label Servidor"

#Configurar flujo de tráfico entre nodos
set tcp1 [new Agent/TCP/SimpleTcp]
set tcp2 [new Agent/TCP/SimpleTcp]

$ns attach-agent $node_(0) $tcp1
$ns attach-agent $node_(1) $tcp2
```

```
$ns connect $tcp1 $tcp2
set encap [new Application/TcpAppmodb $tcp1]
set encap2 [new Application/TcpAppmodb $tcp2]
$encap connect $encap2

# Crea estacion maestro
set master [new MODBmaster]
# Define tiempo de espera para retransmision
$master set retrytimer_ 1
# Define prob. de perdida de paquetes valor aleatorio en intervalo de 0 a
      10
$master set proberror_ 0
# Crea estacion Esclavo
set client [new MODBclient 1]
# Define tiempo de espera para retransmision
$client set retrytimer_ 1
# Define prob. de perdida de paquetes valor aleatorio en intervalo de 0 a
      10
$client set proberror_ 1
$master connect $client $encap
$client connect $master $encap2


for {set j 0} {$j < $val(vv) } {incr j} {
      $ns at $j.0 "$master read $client"
}

$ns at $val(vv) "stop"


$ns run
```

Apéndice C. Archivo para escenario punto a punto Ethernet bajo DNP3

```
set val(vv)       900                  ;# veces que lee a las outstation
#Crear objeto simulador
set ns [new Simulator]

#define color flujo datos
$ns color 1 Blue
$ns color 2 Red

#Creacion archivo Nam
set namfile [open salida_dnp3_2nodos_ether.nam w]
$ns namtrace-all $namfile

# crear archivo trace
set f [open salida_dnp3_2nodos_ether.tr w]
$ns trace-all $f

# Define procedimiento para finalizar a simulacion
proc stop {} {
```

```
global ns f namfile
$ns flush-trace
close $f
close $namfile
#exec nam salida_dnp3_2nodos_ether.nam &
exec gawk -f delayDNPethe.awk salida_dnp3_2nodos_ether.tr &
exit 0
}
#asignación de semilla
$defaultRNG seed 23

# Crear nodos
set n1 [$ns node]
set n2 [$ns node]


# Enlace de los nodos del sistema
# Creación del enlace que conecta n1 y n2
# Canal dúplex, 10Mbps de ancho de banda, 7ms de retardo, cola "droptail"

$ns duplex-link $n1 $n2 10Mb 10ms DropTail

$n1 color red
$n2 color blue

$ns at 0.0 "$n1 label Master" ;#texto de nodos
$ns at 0.0 "$n2 label Outstation" ;#texto de nodos

#posición
$ns simplex-link-op $n1 $n2 orient right
$ns simplex-link-op $n2 $n1 orient left

# Protocolo a Nível de Transporte
# Crea un agente TCP Simple
set tcp1 [new Agent/TCP/SimpleTcp]
set tcp2 [new Agent/TCP/SimpleTcp]

# Asigna el agente al nodo n1 y n2
$ns attach-agent $n1 $tcp1
$ns attach-agent $n2 $tcp2
$ns connect $tcp1 $tcp2

#Crea generador de tráfico TCP y hace el encapsulado en el agente
set encap [new Application/TcpAppmod $tcp1]
set encap2 [new Application/TcpAppmod $tcp2]
$encap connect $encap2

# Crea la estacion Maestra
set master [new DNP3master]
# Define el tiempo de espera retransmision
$master set retrytimer_ 1
# Define prob. de pérdida de paquetes
#$master set proberror_ 20

# Crea la estacion Esclava
set client [new DNP3client 1]
```

```
# Define el tiempo de espera retransmision
$client set retrytimer_ 2
# Define prob. de pérdida de paquetes
#$client set proberror_ 1

$master connect $client $encap
$client connect $master $encap2

# Configura envío de paquetes DNP3
for {set j 0} {$j < $val(vv) } {incr j} {
      $ns at $j.0 "$master read $client"
}

# Configura finalización de simulacion
$ns at $val(vv) "stop"
# Inicia Simulacion
$ns run
```

Apéndice D. Archivo para escenario punto a punto Ethernet bajo Modbus

```
set val(vv)       900                  ;# veces que lee a las outstation
#Crear objeto simulador
set ns [new Simulator]

#define color flujo datos
$ns color 1 Blue
$ns color 2 Red

#Creacion archivo Nam
set namfile [open salida_mod_2nodos_ether.nam w]
$ns namtrace-all $namfile

# crear archivo trace
set f [open salida_mod_2nodos_ether.tr w]
$ns trace-all $f

# Define procedimiento para finalizar a simulacion
proc stop {} {
global ns f namfile
$ns flush-trace
close $f
close $namfile
exec nam salida_mod_2nodos_ether.nam &
exec gawk -f delayMODethe.awk salida_mod_2nodos_ether.tr &
exit 0
}
#asignación de semilla
$defaultRNG seed 23

# Crear nodos
set n1 [$ns node]
set n2 [$ns node]
```

```
# Enlace de los nodos del sistema
# Creación del enlace que conecta n1 y n2
# Canal dúplex, 10Mbps de ancho de banda, 5ms de retardo, cola "droptail"

$ns duplex-link $n1 $n2 10Mb 5ms DropTail

$n1 color yellow
$n2 color blue

$ns at 0.0 "$n1 label Cliente" ;#texto de nodos
$ns at 0.0 "$n2 label Servidor" ;#texto de nodos

#posición
$ns simplex-link-op $n1 $n2 orient right
$ns simplex-link-op $n2 $n1 orient left

# Protocolo a Nível de Transporte
# Crea un agente TCP Simple
set tcp1 [new Agent/TCP/SimpleTcp]
set tcp2 [new Agent/TCP/SimpleTcp]

# Asigna el agente al nodo n1 y n2
$ns attach-agent $n1 $tcp1
$ns attach-agent $n2 $tcp2
$ns connect $tcp1 $tcp2

#Crea generador de tráfico TCP y hace el encapsulado en el agente
set encap [new Application/TcpAppmod $tcp1]
set encap2 [new Application/TcpAppmod $tcp2]
$encap connect $encap2

# Crea la estacion Maestra
set master [new MODBmaster]
# Define el tiempo de espera retransmision
$master set retrytimer_ 1
# Define prob. de pérdida de paquetes
#$master set proberror_ 20

# Crea la estacion Esclava
set client [new MODBclient 1]
# Define el tiempo de espera retransmision
$client set retrytimer_ 2
# Define prob. de pérdida de paquetes
#$client set proberror_ 1

$master connect $client $encap
$client connect $master $encap2

# Configura envío de paquetes Modbus
for {set j 0} {$j < $val(vv) } {incr j} {
      $ns at $j.0 "$master read $client"
}

# Configura finalización de simulacion
$ns at $val(vv) "stop"
# Inicia Simulacion
```

```
$ns run
```

Apéndice E. Archivo para escenario multipunto bajo Modbus.

```
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 5.5
Antenna/OmniAntenna set Gt_ 8.0
Antenna/OmniAntenna set Gr_ 8.0

Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-14
Phy/WirelessPhy set RXThresh_ 5.01187233e-12
Phy/WirelessPhy set Pt_ 0.63
Phy/WirelessPhy set freq_ 2.4e+9

Propagation/Shadowing set pathlossExp_ 3.5  ;# path loss exponent B
Propagation/Shadowing set std_db_ 6.2       ;# shadowing deviation (odB)
Propagation/Shadowing set dist0_ 1.0        ;# reference distance (m)
Propagation/Shadowing set seed_ 0           ;# seed for RNG


set val(chan)           Channel/WirelessChannel    ;# channel type
set val(prop)           Propagation/Shadowing      ;# radio-propagation
      model FreeSpace TwoRayGround Shadowing
set val(netif)          Phy/WirelessPhy            ;# network interface type
set val(mac)            Mac/802_11                 ;# MAC type
set val(ifq)            Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)             LL                         ;# link layer type
set val(ant)            Antenna/OmniAntenna        ;# antenna model
set val(ifqlen)         1000                       ;# max packet in ifq
set val(nn)             11                          ;# number of mobilenodes
set val(rp)             DumbAgent                  ;# routing protocol
set val(xx)       350                   ;# X dimension of topography
set val(yy)             300                    ;# Y dimension of topography
set val(vv)       900                   ;# veces que lee a las outstation

# Inicializa variables Globales
set ns            [new Simulator]

#Creacion archivo Nam
set namfile [open salida_mod_10nodos.nam w]
$ns namtrace-all-wireless $namfile $val(xx) $val(yy)
#Creacion archivo Trace
set tracefile [open salida_mod_10nodos.tr w]
$ns trace-all $tracefile

proc stop {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
  #  exec nam salida_mod_10nodos.nam &
    exec gawk -f delayMOD.awk salida_mod_10nodos.tr &
```

```
    #exit 0
}
# Configurar objetos de topografía
set topo       [new Topography]
$topo load_flatgrid $val(xx)  $val(yy)

# Crea God (General Operations Director)
create-god $val(nn)

#  Configuraciones de Escenario
$ns node-config -adhocRouting $val(rp) \
 -llType $val(ll) \
 -macType $val(mac) \
 -ifqType $val(ifq) \
 -ifqLen $val(ifqlen) \
 -antType $val(ant) \
 -propType $val(prop) \
 -phyType $val(netif) \
 -channelType $val(chan) \
 -topoInstance $topo \
 -agentTrace ON \
 -routerTrace ON \
 -macTrace OFF \
 -movementTrace ON

# Creear nodos móviles [$ val (nn)]
for {set i 0} {$i < $val(nn) } {incr i} {
  set node_($i) [$ns node]
  $node_($i) random-motion 0 ;# desactivar el movimiento aleatorio
}

# Coordenadas iniciales (X, Y, Z = 0) para nodos móviles
$node_(0) set X_ 150.0
$node_(0) set Y_ 240.0
$node_(0) set Z_ 0.0
$node_(0) color black

$node_(1) set X_ 80.0
$node_(1) set Y_ 260.0
$node_(1) set Z_ 0.0
$node_(1) color black

$node_(2) set X_ 60.0
$node_(2) set Y_ 290.0
$node_(2) set Z_ 0.0
$node_(2) color black

$node_(3) set X_ 65.0
$node_(3) set Y_ 220.0
$node_(3) set Z_ 0.0
$node_(3) color black

$node_(4) set X_ 145.0
$node_(4) set Y_ 288.0
$node_(4) set Z_ 0.0
$node_(4) color black
```

```
$node_(5) set X_ 260.0
$node_(5) set Y_ 210.0
$node_(5) set Z_ 0.0
$node_(5) color black

$node_(6) set X_ 267.0
$node_(6) set Y_ 220.0
$node_(6) set Z_ 0.0
$node_(6) color black

$node_(7) set X_ 258.0
$node_(7) set Y_ 142.0
$node_(7) set Z_ 0.0
$node_(7) color black

$node_(8) set X_ 253.0
$node_(8) set Y_ 57.0
$node_(8) set Z_ 0.0
$node_(8) color black

$node_(9) set X_ 130.0
$node_(9) set Y_ 30.0
$node_(9) set Z_ 0.0
$node_(9) color black

$node_(10) set X_ 140.0
$node_(10) set Y_ 220.0
$node_(10) set Z_ 0.0
$node_(10) color black

$ns initial_node_pos $node_(0) 15 ;#tamaño de nodos
$ns initial_node_pos $node_(1) 15
$ns initial_node_pos $node_(2) 15
$ns initial_node_pos $node_(3) 15
$ns initial_node_pos $node_(4) 15
$ns initial_node_pos $node_(5) 15
$ns initial_node_pos $node_(6) 15
$ns initial_node_pos $node_(7) 15
$ns initial_node_pos $node_(8) 15
$ns initial_node_pos $node_(9) 15
$ns initial_node_pos $node_(10) 15

$ns at 0.0 "$node_(0) color red" ;#color de nodos
$ns at 0.0 "$node_(0) label C_Gestion" ;#texto de nodos
$ns at 0.0 "$node_(1) color blue"
$ns at 0.0 "$node_(1) label Lab_Ingenieria"
$ns at 0.0 "$node_(2) color blue"
$ns at 0.0 "$node_(2) label Lab_Especializados"
$ns at 0.0 "$node_(3) color blue"
$ns at 0.0 "$node_(3) label Planta_Elec"
$ns at 0.0 "$node_(4) color blue"
$ns at 0.0 "$node_(4) label Bloque_T"
$ns at 0.0 "$node_(5) color blue"
$ns at 0.0 "$node_(5) label Biblioteca"
$ns at 0.0 "$node_(6) color blue"
$ns at 0.0 "$node_(6) label Bloque_2"
```

```
$ns at 0.0 "$node_(7) color blue"
$ns at 0.0 "$node_(7) label Admin"
$ns at 0.0 "$node_(8) color blue"
$ns at 0.0 "$node_(8) label Fac_Artes"
$ns at 0.0 "$node_(9) color blue"
$ns at 0.0 "$node_(9) label Coliseo"
$ns at 0.0 "$node_(10) color blue"
$ns at 0.0 "$node_(10) label Lab_Docencia"

#Configurar flujo de tráfico entre nodos
set tcp1 [new Agent/TCP/SimpleTcp]
set tcp2 [new Agent/TCP/SimpleTcp]
set tcp3 [new Agent/TCP/SimpleTcp]
set tcp4 [new Agent/TCP/SimpleTcp]
set tcp5 [new Agent/TCP/SimpleTcp]
set tcp6 [new Agent/TCP/SimpleTcp]
set tcp7 [new Agent/TCP/SimpleTcp]
set tcp8 [new Agent/TCP/SimpleTcp]
set tcp9 [new Agent/TCP/SimpleTcp]
set tcp10 [new Agent/TCP/SimpleTcp]
set tcp11 [new Agent/TCP/SimpleTcp]
set tcp12 [new Agent/TCP/SimpleTcp]
set tcp13 [new Agent/TCP/SimpleTcp]
set tcp14 [new Agent/TCP/SimpleTcp]
set tcp15 [new Agent/TCP/SimpleTcp]
set tcp16 [new Agent/TCP/SimpleTcp]
set tcp17 [new Agent/TCP/SimpleTcp]
set tcp18 [new Agent/TCP/SimpleTcp]
set tcp19 [new Agent/TCP/SimpleTcp]
set tcp20 [new Agent/TCP/SimpleTcp]

$ns attach-agent $node_(0) $tcp1
$ns attach-agent $node_(1) $tcp2
$ns attach-agent $node_(0) $tcp3
$ns attach-agent $node_(2) $tcp4
$ns attach-agent $node_(0) $tcp5
$ns attach-agent $node_(3) $tcp6
$ns attach-agent $node_(0) $tcp7
$ns attach-agent $node_(4) $tcp8
$ns attach-agent $node_(0) $tcp9
$ns attach-agent $node_(5) $tcp10
$ns attach-agent $node_(0) $tcp11
$ns attach-agent $node_(6) $tcp12
$ns attach-agent $node_(0) $tcp13
$ns attach-agent $node_(7) $tcp14
$ns attach-agent $node_(0) $tcp15
$ns attach-agent $node_(8) $tcp16
$ns attach-agent $node_(0) $tcp17
$ns attach-agent $node_(9) $tcp18
$ns attach-agent $node_(0) $tcp19
$ns attach-agent $node_(10) $tcp20

$ns connect $tcp1 $tcp2
$ns connect $tcp3 $tcp4
$ns connect $tcp5 $tcp6
$ns connect $tcp7 $tcp8
```

```
$ns connect $tcp9 $tcp10
$ns connect $tcp11 $tcp12
$ns connect $tcp13 $tcp14
$ns connect $tcp15 $tcp16
$ns connect $tcp17 $tcp18
$ns connect $tcp19 $tcp20

set encap1 [new Application/TcpAppmod $tcp1]
set encap2 [new Application/TcpAppmod $tcp2]
set encap3 [new Application/TcpAppmod $tcp3]
set encap4 [new Application/TcpAppmod $tcp4]
set encap5 [new Application/TcpAppmod $tcp5]
set encap6 [new Application/TcpAppmod $tcp6]
set encap7 [new Application/TcpAppmod $tcp7]
set encap8 [new Application/TcpAppmod $tcp8]
set encap9 [new Application/TcpAppmod $tcp9]
set encap10 [new Application/TcpAppmod $tcp10]
set encap11 [new Application/TcpAppmod $tcp11]
set encap12 [new Application/TcpAppmod $tcp12]
set encap13 [new Application/TcpAppmod $tcp13]
set encap14 [new Application/TcpAppmod $tcp14]
set encap15 [new Application/TcpAppmod $tcp15]
set encap16 [new Application/TcpAppmod $tcp16]
set encap17 [new Application/TcpAppmod $tcp17]
set encap18 [new Application/TcpAppmod $tcp18]
set encap19 [new Application/TcpAppmod $tcp19]
set encap20 [new Application/TcpAppmod $tcp20]

$encap1 connect $encap2
$encap3 connect $encap4
$encap5 connect $encap6
$encap7 connect $encap8
$encap9 connect $encap10
$encap11 connect $encap12
$encap13 connect $encap14
$encap15 connect $encap16
$encap17 connect $encap18
$encap19 connect $encap20

# Crea estacion maestro
set master [new MODBmaster]
$master set retrytimer_ 1
$master set proberror_ 0

# Crea estacion Esclavo
set client1 [new MODBclient 1]
$client1 set retrytimer_ 1
$client1 set proberror_ 1
set client2 [new MODBclient 2]
$client2 set retrytimer_ 1
$client2 set proberror_ 1
set client3 [new MODBclient 3]
$client3 set retrytimer_ 1
$client3 set proberror_ 1
set client4 [new MODBclient 4]
$client4 set retrytimer_ 1
```

```
$client4 set proberror_ 1
set client5 [new MODBclient 5]
$client5 set retrytimer_ 1
$client5 set proberror_ 1
set client6 [new MODBclient 6]
$client6 set retrytimer_ 1
$client6 set proberror_ 1
set client7 [new MODBclient 7]
$client7 set retrytimer_ 1
$client7 set proberror_ 1
set client8 [new MODBclient 8]
$client8 set retrytimer_ 1
$client8 set proberror_ 1
set client9 [new MODBclient 9]
$client9 set retrytimer_ 1
$client9 set proberror_ 1
set client10 [new MODBclient 10]
$client10 set retrytimer_ 1
$client10 set proberror_ 1

$master connect $client1 $encap1
$client1 connect $master $encap2
$master connect $client2 $encap3
$client2 connect $master $encap4
$master connect $client3 $encap5
$client3 connect $master $encap6
$master connect $client4 $encap7
$client4 connect $master $encap8
$master connect $client5 $encap9
$client5 connect $master $encap10
$master connect $client6 $encap11
$client6 connect $master $encap12
$master connect $client7 $encap13
$client7 connect $master $encap14
$master connect $client8 $encap15
$client8 connect $master $encap16
$master connect $client9 $encap17
$client9 connect $master $encap18
$master connect $client10 $encap19
$client10 connect $master $encap20

for {set j 0} {$j < $val(vv) } {incr j} {
      $ns at $j.0 "$master read $client1"
      $ns at $j.0 "$master read $client2"
      $ns at $j.0 "$master read $client3"
      $ns at $j.0 "$master read $client4"
      $ns at $j.0 "$master read $client5"
      $ns at $j.0 "$master read $client6"
      $ns at $j.0 "$master read $client7"
      $ns at $j.0 "$master read $client8"
      $ns at $j.0 "$master read $client9"
      $ns at $j.0 "$master read $client10"

}

$ns at $val(vv) "stop"
```

```
$ns run
```

Apéndice F. Archivo para escenario multipunto bajo DNP3.

```
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 5.5
Antenna/OmniAntenna set Gt_ 8.0
Antenna/OmniAntenna set Gr_ 8.0

Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-14
Phy/WirelessPhy set RXThresh_ 5.01187233e-12
Phy/WirelessPhy set Pt_ 0.63
Phy/WirelessPhy set freq_ 2.4e+9

Propagation/Shadowing set pathlossExp_ 3.5  ;# path loss exponent B
Propagation/Shadowing set std_db_ 6.2       ;# shadowing deviation (odB)
Propagation/Shadowing set dist0_ 1.0        ;# reference distance (m)
Propagation/Shadowing set seed_ 0           ;# seed for RNG


set val(chan)           Channel/WirelessChannel    ;# channel type
set val(prop)           Propagation/Shadowing      ;# radio-propagation
      model FreeSpace TwoRayGround Shadowing
set val(netif)          Phy/WirelessPhy            ;# network interface type
set val(mac)            Mac/802_11                 ;# MAC type
set val(ifq)            Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)             LL                         ;# link layer type
set val(ant)            Antenna/OmniAntenna        ;# antenna model
set val(ifqlen)         1000                       ;# max packet in ifq
set val(nn)             11                          ;# number of mobilenodes
set val(rp)             DumbAgent                  ;# routing protocol
set val(xx)       350                   ;# X dimension of topography
set val(yy)             300                    ;# Y dimension of topography
set val(vv)       900                   ;# veces que lee a las outstation

# Inicializa variables Globales
set ns            [new Simulator]

#Creacion archivo Nam
set namfile [open salida_dnp3_10nodos.nam w]
$ns namtrace-all-wireless $namfile $val(xx) $val(yy)
#Creacion archivo Trace
set tracefile [open salida_dnp3_10nodos.tr w]
$ns trace-all $tracefile

proc stop {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
 #   exec nam salida_dnp3_10nodos.nam &
    exec gawk -f delayDNP.awk salida_dnp3_10nodos.tr &
```

```
    #exit 0
}
# Configurar objetos de topografía
set topo       [new Topography]
$topo load_flatgrid $val(xx)  $val(yy)

# Crea God (General Operations Director)
create-god $val(nn)

#  Configuraciones de Escenario
$ns node-config -adhocRouting $val(rp) \
 -llType $val(ll) \
 -macType $val(mac) \
 -ifqType $val(ifq) \
 -ifqLen $val(ifqlen) \
 -antType $val(ant) \
 -propType $val(prop) \
 -phyType $val(netif) \
 -channelType $val(chan) \
 -topoInstance $topo \
 -agentTrace ON \
 -routerTrace ON \
 -macTrace OFF \
 -movementTrace ON

# Creear nodos móviles [$ val (nn)]
for {set i 0} {$i < $val(nn) } {incr i} {
  set node_($i) [$ns node]
  $node_($i) random-motion 0 ;# desactivar el movimiento aleatorio
}

# Coordenadas iniciales (X, Y, Z = 0) para nodos móviles
$node_(0) set X_ 150.0
$node_(0) set Y_ 240.0
$node_(0) set Z_ 0.0
$node_(0) color black

$node_(1) set X_ 80.0
$node_(1) set Y_ 260.0
$node_(1) set Z_ 0.0
$node_(1) color black

$node_(2) set X_ 60.0
$node_(2) set Y_ 290.0
$node_(2) set Z_ 0.0
$node_(2) color black

$node_(3) set X_ 65.0
$node_(3) set Y_ 220.0
$node_(3) set Z_ 0.0
$node_(3) color black

$node_(4) set X_ 145.0
$node_(4) set Y_ 288.0
$node_(4) set Z_ 0.0
$node_(4) color black
```

```
$node_(5) set X_ 260.0
$node_(5) set Y_ 210.0
$node_(5) set Z_ 0.0
$node_(5) color black

$node_(6) set X_ 267.0
$node_(6) set Y_ 220.0
$node_(6) set Z_ 0.0
$node_(6) color black

$node_(7) set X_ 258.0
$node_(7) set Y_ 142.0
$node_(7) set Z_ 0.0
$node_(7) color black

$node_(8) set X_ 253.0
$node_(8) set Y_ 57.0
$node_(8) set Z_ 0.0
$node_(8) color black

$node_(9) set X_ 130.0
$node_(9) set Y_ 20.0
$node_(9) set Z_ 0.0
$node_(9) color black

$node_(10) set X_ 140.0
$node_(10) set Y_ 220.0
$node_(10) set Z_ 0.0
$node_(10) color black

$ns initial_node_pos $node_(0) 15 ;#tamaño de nodos
$ns initial_node_pos $node_(1) 15
$ns initial_node_pos $node_(2) 15
$ns initial_node_pos $node_(3) 15
$ns initial_node_pos $node_(4) 15
$ns initial_node_pos $node_(5) 15
$ns initial_node_pos $node_(6) 15
$ns initial_node_pos $node_(7) 15
$ns initial_node_pos $node_(8) 15
$ns initial_node_pos $node_(9) 15
$ns initial_node_pos $node_(10) 15

$ns at 0.0 "$node_(0) color red" ;#color de nodos
$ns at 0.0 "$node_(0) label C_Gestion" ;#texto de nodos
$ns at 0.0 "$node_(1) color blue"
$ns at 0.0 "$node_(1) label Lab_Ingenieria"
$ns at 0.0 "$node_(2) color blue"
$ns at 0.0 "$node_(2) label Lab_Especializados"
$ns at 0.0 "$node_(3) color blue"
$ns at 0.0 "$node_(3) label Planta_Elec"
$ns at 0.0 "$node_(4) color blue"
$ns at 0.0 "$node_(4) label Bloque_T"
$ns at 0.0 "$node_(5) color blue"
$ns at 0.0 "$node_(5) label Biblioteca"
$ns at 0.0 "$node_(6) color blue"
```

```
$ns at 0.0 "$node_(6) label Bloque_2"
$ns at 0.0 "$node_(7) color blue"
$ns at 0.0 "$node_(7) label Admin"
$ns at 0.0 "$node_(8) color blue"
$ns at 0.0 "$node_(8) label Fac_Artes"
$ns at 0.0 "$node_(9) color blue"
$ns at 0.0 "$node_(9) label Coliseo"
$ns at 0.0 "$node_(10) color blue"
$ns at 0.0 "$node_(10) label Lab_Docencia"

#Configurar flujo de tráfico entre nodos
set tcp1 [new Agent/TCP/SimpleTcp]
set tcp2 [new Agent/TCP/SimpleTcp]
set tcp3 [new Agent/TCP/SimpleTcp]
set tcp4 [new Agent/TCP/SimpleTcp]
set tcp5 [new Agent/TCP/SimpleTcp]
set tcp6 [new Agent/TCP/SimpleTcp]
set tcp7 [new Agent/TCP/SimpleTcp]
set tcp8 [new Agent/TCP/SimpleTcp]
set tcp9 [new Agent/TCP/SimpleTcp]
set tcp10 [new Agent/TCP/SimpleTcp]
set tcp11 [new Agent/TCP/SimpleTcp]
set tcp12 [new Agent/TCP/SimpleTcp]
set tcp13 [new Agent/TCP/SimpleTcp]
set tcp14 [new Agent/TCP/SimpleTcp]
set tcp15 [new Agent/TCP/SimpleTcp]
set tcp16 [new Agent/TCP/SimpleTcp]
set tcp17 [new Agent/TCP/SimpleTcp]
set tcp18 [new Agent/TCP/SimpleTcp]
set tcp19 [new Agent/TCP/SimpleTcp]
set tcp20 [new Agent/TCP/SimpleTcp]

$ns attach-agent $node_(0) $tcp1
$ns attach-agent $node_(1) $tcp2
$ns attach-agent $node_(0) $tcp3
$ns attach-agent $node_(2) $tcp4
$ns attach-agent $node_(0) $tcp5
$ns attach-agent $node_(3) $tcp6
$ns attach-agent $node_(0) $tcp7
$ns attach-agent $node_(4) $tcp8
$ns attach-agent $node_(0) $tcp9
$ns attach-agent $node_(5) $tcp10
$ns attach-agent $node_(0) $tcp11
$ns attach-agent $node_(6) $tcp12
$ns attach-agent $node_(0) $tcp13
$ns attach-agent $node_(7) $tcp14
$ns attach-agent $node_(0) $tcp15
$ns attach-agent $node_(8) $tcp16
$ns attach-agent $node_(0) $tcp17
$ns attach-agent $node_(9) $tcp18
$ns attach-agent $node_(0) $tcp19
$ns attach-agent $node_(10) $tcp20

$ns connect $tcp1 $tcp2
$ns connect $tcp3 $tcp4
$ns connect $tcp5 $tcp6
```

```
$ns connect $tcp7 $tcp8
$ns connect $tcp9 $tcp10
$ns connect $tcp11 $tcp12
$ns connect $tcp13 $tcp14
$ns connect $tcp15 $tcp16
$ns connect $tcp17 $tcp18
$ns connect $tcp19 $tcp20

set encap1 [new Application/TcpAppmod $tcp1]
set encap2 [new Application/TcpAppmod $tcp2]
set encap3 [new Application/TcpAppmod $tcp3]
set encap4 [new Application/TcpAppmod $tcp4]
set encap5 [new Application/TcpAppmod $tcp5]
set encap6 [new Application/TcpAppmod $tcp6]
set encap7 [new Application/TcpAppmod $tcp7]
set encap8 [new Application/TcpAppmod $tcp8]
set encap9 [new Application/TcpAppmod $tcp9]
set encap10 [new Application/TcpAppmod $tcp10]
set encap11 [new Application/TcpAppmod $tcp11]
set encap12 [new Application/TcpAppmod $tcp12]
set encap13 [new Application/TcpAppmod $tcp13]
set encap14 [new Application/TcpAppmod $tcp14]
set encap15 [new Application/TcpAppmod $tcp15]
set encap16 [new Application/TcpAppmod $tcp16]
set encap17 [new Application/TcpAppmod $tcp17]
set encap18 [new Application/TcpAppmod $tcp18]
set encap19 [new Application/TcpAppmod $tcp19]
set encap20 [new Application/TcpAppmod $tcp20]

$encap1 connect $encap2
$encap3 connect $encap4
$encap5 connect $encap6
$encap7 connect $encap8
$encap9 connect $encap10
$encap11 connect $encap12
$encap13 connect $encap14
$encap15 connect $encap16
$encap17 connect $encap18
$encap19 connect $encap20

# Crea estacion maestro
set master [new DNP3master]
$master set retrytimer_ 1
$master set proberror_ 0

# Crea estacion Esclavo
set client1 [new DNP3client 1]
$client1 set retrytimer_ 1
$client1 set proberror_ 1
set client2 [new DNP3client 2]
$client2 set retrytimer_ 1
$client2 set proberror_ 1
set client3 [new DNP3client 3]
$client3 set retrytimer_ 1
$client3 set proberror_ 1
set client4 [new DNP3client 4]
```

```
$client4 set retrytimer_ 1
$client4 set proberror_ 1
set client5 [new DNP3client 5]
$client5 set retrytimer_ 1
$client5 set proberror_ 1
set client6 [new DNP3client 6]
$client6 set retrytimer_ 1
$client6 set proberror_ 1
set client7 [new DNP3client 7]
$client7 set retrytimer_ 1
$client7 set proberror_ 1
set client8 [new DNP3client 8]
$client8 set retrytimer_ 1
$client8 set proberror_ 1
set client9 [new DNP3client 9]
$client9 set retrytimer_ 1
$client9 set proberror_ 1
set client10 [new DNP3client 10]
$client10 set retrytimer_ 1
$client10 set proberror_ 1
$master connect $client1 $encap1
$client1 connect $master $encap2
$master connect $client2 $encap3
$client2 connect $master $encap4
$master connect $client3 $encap5
$client3 connect $master $encap6
$master connect $client4 $encap7
$client4 connect $master $encap8
$master connect $client5 $encap9
$client5 connect $master $encap10
$master connect $client6 $encap11
$client6 connect $master $encap12
$master connect $client7 $encap13
$client7 connect $master $encap14
$master connect $client8 $encap15
$client8 connect $master $encap16
$master connect $client9 $encap17
$client9 connect $master $encap18
$master connect $client10 $encap19
$client10 connect $master $encap20

for {set j 0} {$j < $val(vv) } {incr j} {
      $ns at $j.0 "$master read $client1"
      $ns at $j.0 "$master read $client2"
      $ns at $j.0 "$master read $client3"
      $ns at $j.0 "$master read $client4"
      $ns at $j.0 "$master read $client5"
      $ns at $j.0 "$master read $client6"
      $ns at $j.0 "$master read $client7"
      $ns at $j.0 "$master read $client8"
      $ns at $j.0 "$master read $client9"
      $ns at $j.0 "$master read $client10"
}

$ns at $val(vv) "stop"
$ns run
```