



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**PROYECTO DE GRADUACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES**

AUTOR: VERÓNICA ISABEL POAQUIZA YUMBULEMA

**TEMA: OPTIMIZACIÓN DEL NÚMERO DE NODOS EN REDES DE
SENSORES INALÁMBRICOS APLICADAS A MONITORIZACIÓN
VOLCÁNICA**

**DIRECTOR: ING. ROMÁN LARA
CODIRECTOR: ING. RODOLFO GORDILLO**

**SANGOLQUÍ – ECUADOR
2015**

Certificado de tutoría

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

CERTIFICADO

Ing. Román Lara

Ing. Rodolfo Gordillo

Certificamos que el presente proyecto de graduación titulado “Optimización del Número de Nodos En Redes de Sensores Inalámbricos Aplicadas a Monitorización Volcánica”, fue realizado en su totalidad por la Srta. Verónica Isabel Poaquiza Yumbulema, y ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

Sangolquí, Mayo del 2015.

Ing. Román Lara
DIRECTOR

Ing. Rodolfo Gordillo
CODIRECTOR

Declaración de Responsabilidad

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Verónica Isabel Poaquiza Yumbulema

DECLARO QUE:

El proyecto de graduación denominado “Optimización del Número de Nodos En Redes de Sensores Inalámbricos Aplicadas a Monitorización Volcánica”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie, de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de graduación en mención.

Sangolquí, Mayo del 2015



Verónica Isabel Poaquiza Yumbulema

ID: L00248642

CI:0502864572

Autorización de Publicación

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, Verónica Isabel Poaquiza Yumbulema

Autorizo a la Universidad de las Fuerzas Armadas-ESPE la publicación del trabajo de graduación titulado “Optimización del Número de Nodos En Redes de Sensores Inalámbricos Aplicadas a Monitorización Volcánica”, en la biblioteca virtual de la Institución, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, Mayo del 2015

Verónica Isabel Poaquiza Yumbulema

ID: L00248642

CI:0502864572

DEDICATORIA

A mis padres, Angel y María,

por enseñarme el significado de esfuerzo, perseverancia y amor,

A mis hermanos,

por regalarme alegrías y enseñarme a no rendirme en momentos difíciles,

A mis amigos,

por estar a mi lado y compartir momentos increíbles.

Verónica

AGRADECIMIENTOS

A Dios por ser bondadoso y regalarme cada día de vida junto a mi familia, por darme salud, fortaleza, un corazón bueno para amar, y sobre todo mantenerme rodeada de los seres que más amo.

A mis padres en especial a mi madre quien con su ejemplo y amor a lo largo de toda mi vida ha velado por mi bienestar y educación, quien ha sido mi apoyo incondicional en todo momento, gracias mami por ser una mujer ejemplar.

A mis hermanas Rocy, Mayra, Silvia, Katty y Anita por su cariño y comprensión, gracias por ser mi compañía y ser parte de mis más grandes alegrías.

A mis hermanos Luis, Paul, Jhonatan y Joel quienes siempre han confiado en mí y han sido parte muy importante en cada etapa de mi vida.

A ti Cesar por todas las veces que apoyaste mis sueños, mis proyectos y mis aventuras, gracias por tu ayuda y por cada momento que compartimos.

A ti Anita por ser una buena amiga, por tu apoyo constante te convertiste en una hermana y compañera en toda esta fase universitaria, gracias por haberme brindado mucho cariño y apoyo cuando más necesitaba.

A mis amigos, gracias por brindarme una amistad sincera.

Y Agradezco al director y codirector de mi Proyecto de Grado el Ingeniero Román Lara y el Ingeniero Rodolfo Gordillo, por su paciencia, su ayuda y especialmente por darme la oportunidad de aprender y crecer profesionalmente con la realización de éste proyecto.

ÍNDICE GENERAL

DEDICATORIA	iv
AGRADECIMIENTOS	v
ÍNDICE GENERAL	vi
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	xi
GLOSARIO	xii
RESUMEN	xviii
ABSTRAC	xix
PRÓLOGO	xx

CAPÍTULO 1

PRESENTACIÓN DEL PROYECTO

1.1 INTRODUCCIÓN	1
1.2 ANTECEDENTES	2
1.3 JUSTIFICACIÓN E IMPORTANCIA	5
1.4 ALCANCE DEL PROYECTO	6
1.5 OBJETIVOS	7
1.5.1 GENERAL	7
1.5.2 ESPECÍFICOS	7

CAPÍTULO 2

FUNDAMENTO TEÓRICO

2.1 REDES DE SENSORES INALÁMBRICAS	8
2.1.1 INTRODUCCIÓN	8

2.1.2	CARACTERÍSTICAS DE LAS WSN	9
2.1.3	ARQUITECTURA DE COMUNICACIÓN DE LAS WSN	10
2.1.4	COMPONENTES DE UNA WSN	13
2.1.5	FACTORES QUE INFLUYEN EN EL DISEÑO DE LAS WSNs	14
2.2	ZIGBEE Y EL ESTÁNDAR 802.15.4	18
2.2.1	CARACTERÍSTICAS DE ZIGBEE.	23
2.2.2	CARACTERÍSTICAS DE IEEE 802.15.4	24
2.2.3	TOPOLOGÍAS DE RED	24
2.2.4	TIPO DE DISPOSITIVOS ZIGBEE	27
2.3	MÉTRICAS DE QoS EN WSN	29
2.3.1	THROUGHPUT	30
2.3.2	RETARDO (<i>END-TO-END DELAY</i>)	30
2.3.3	JITTER	31
2.3.4	PAQUETES PERDIDOS	31
2.4	OPTIMIZACIÓN DE FUNCIONES	31
2.4.1	OPTIMIZACIÓN LINEAL	34
2.4.2	OPTIMIZACIÓN NO LINEAL	35
2.5	TÉCNICAS DE OPTIMIZACIÓN	37
2.5.1	OPTIMIZACIÓN ESCALAR	37
2.5.2	OPTIMIZACIÓN MULTIVARIABLE SIN RESTRICCIONES	44
2.5.3	OPTIMIZACIÓN MULTIVARIABLE CON RESTRICCIONES	50

CAPÍTULO 3

SIMULACIÓN WSN EN SOFTWARE NS-2

3.1	ESCENARIOS WSN DE SIMULACIÓN	54
3.2	SIMULACIÓN DE LAS TOPOLOGÍAS DE RED WSN	63
3.3	TABULACIÓN DE RESULTADOS	71

3.4	ANÁLISIS DE RESULTADOS	75
------------	-------------------------------------	-----------

CAPÍTULO 4

OPTIMIZACIÓN DEL NÚMERO DE SENSORES INALÁMBRICOS

4.1	APLICACIÓN DE TÉCNICAS DE OPTIMIZACIÓN	83
4.2	TABULACIÓN DE RESULTADOS	107
4.2.1	RESULTADOS DE OPTIMIZACIÓN NO LINEAL DE FUNCION UNIVARIABLE	107
4.2.2	RESULTADOS DE OPTIMIZACIÓN NO LINEAL DE FUNCION MULTIVARIABLE	109
4.3	RESULTADOS	112
4.3.1	REPRESENTACIÓN GRÁFICA Y ANÁLISIS	112

CAPÍTULO 5

DISCUSIÓN

5.1	PRINCIPALES CONTRIBUCIONES	117
5.2	CONCLUSIONES	118
5.3	TRABAJOS FUTUROS	119
REFERENCIAS BIBLIOGRÁFICAS		121

ÍNDICE DE FIGURAS

Figura. 2.1. Arquitectura WSN centralizada	11
Figura. 2.2. Arquitectura WSN distribuida.....	11
Figura. 2.3. Arquitectura general de una WSN	12
Figura. 2.4. Pila de protocolos en una WSN	13
Figura. 2.5. Arquitectura básica de un nodo sensor [17].....	16
Figura. 2.6. Arquitectura en capas ZigBee/802.15.4 [20].	19
Figura. 2.7. Canales de Frecuencias usadas por IEEE 802.15.4.....	21
Figura. 2.8. Topologías de la red ZigBee/802.15.4	25
Figura. 2.9. Máximos y Mínimos de una función	34
Figura. 2.10. Optimización escalar	38
Figura. 2.11. Selección de la figura simplex inicial	48
Figura. 2.12. Puntos óptimos de una función: a) máximos, b) mínimos,	51
Figura. 3.1. Distribución aleatoria, 15 nodos	56
Figura. 3.2. Escenarios de Simulación, 3, 7, 11, 19, 23, 28, 33, y 39 nodos.....	63
Figura. 3.3. Tiempo de encendido de dispositivos de red (Escenario de 8 Nodos).....	67
Figura. 3.4. Dispositivos de la red (Escenario de 10 Nodos)	67
Figura. 3.5. Propagación de evento sísmico hacia la red WSN.....	68
Figura. 3.6. Transmisión de paquetes Nodo 8 – Coordinador, Escenario 10 nodos, ns-2	69
Figura. 3.7. Transmisión de paquetes, Simulación ns-2, Escenarios 7 nodos	71
Figura. 3.8. Herramienta TraceGraph, Ventana Network information	71
Figura. 3.9. Throughput de bits generados en unidades de bits/TIL, Escenarios: a) 8 nodos, b) 21 nodos, c) 35 nodos, y d) 39 nodos	72
Figura. 3.10. Ventana principal Curve Fitting.....	76
Figura. 3.11. Rendimiento de la red: a) Throughput de la red, b) Trazado de Residuos...	79
Figura. 3.12. Métricas de calidad de la red a) Retardo, b) Paquetes Perdidos	80
Figura. 3.13. Superficie Nodos vs Throughput vs Retardo de la red	81
Figura. 4.1. Ventana principal, Optimization Toolbox™ Matlab.	92
Figura. 4.2. Solución usando la función fminbnd sobre interfaz gráfica.	93

Figura. 4.3. Restricción gráfica en $P(x,y)$	97
Figura. 4.4. Solución gráfica usando la función <code>fminunc</code>	99
Figura. 4.5. Solución gráfica con la función <code>fmincon</code>	101
Figura. 4.6. Solución gráfica sin intervención del Gradiente.	102
Figura. 4.7. Solución gráfica sujeta a restricciones.	103
Figura. 4.8. Resultado gráfico Optimización Univariable, Throughput vs Nodos.....	113
Figura. 4.9. Resultado gráfico Optimización Univariable, Retardo vs Nodos	113
Figura. 4.10. Resultado gráfico Optimización Univariable, Paquetes Perdidos vs Nodos	114
Figura. 4.11. Resultado gráfico de Optimización Multivariable	116

ÍNDICE DE TABLAS

Tabla. 2.1. Características de las Bandas de frecuencia del estándar IEEE 802.15.4	21
Tabla. 2.2. Comparación de topologías de WSN	26
Tabla. 2.3. Comparación de optimización Lineal y no lineal.....	36
Tabla. 3.1. Modelo “ASW” de velocidades para las ondas P y S, en el Ecuador [47].....	58
Tabla. 3.2. Distancia y Tiempo de arribo de las ondas P y S (Calculados).....	59
Tabla. 3.3. Modelado y Escenarios de Simulación.....	61
Tabla. 3.4. Parámetros de Simulación	64
Tabla. 3.5. Parámetros de simulación ns-2 en línea de comandos, Escenario 16 Nodos .	65
Tabla. 3.6. Resultados de Simulaciones ns-2 (TraceGraph)	73
Tabla. 3.7. Resultados de Simulaciones (Calculados).....	74
Tabla. 3.8. Estadísticas de bondad de ajuste.....	77
Tabla. 3.9. Funciones de métricas de la red en función del número de nodos	78
Tabla. 4.1. Funciones de minimización [54]	84
Tabla. 4.4. Función fminbnd, MATLAB® [55].....	85
Tabla. 4.5. Función fminunc, MATLAB® [55]	86
Tabla. 4.6. Función fmincon, MATLAB® [55]	87
Tabla. 4.7. Parámetros de Options [55].....	88
Tabla. 4.8. Restricciones de desigualdad no lineal.....	97
Tabla. 4.9. Configuración y programación de parámetros en fminbnd.....	108
Tabla. 4.10. Resultados de Optimización de funciones de una variable	109
Tabla. 4.11. Datos del proceso de Optimización	109
Tabla. 4.12. Configuración y programación de parámetros en fminunc y fmincon.....	110
Tabla. 4.13. Resultados de Optimización de funciones multivariantes sujetas a Desigualdades no lineales (DNL), mediante línea de Comandos (LC) e Interfaz Gráfica (IG)	111
Tabla. 4.14. Datos del proceso de Optimización	111
Tabla. 4.15. Comparación de Resultados del proceso de optimización.	115

GLOSARIO

A

- ACK** Acknowledgment, acuse de recibo.
- AD-HOC** Red Ad-hoc. Red inalámbrica donde la comunicación entre los dispositivos es directa, sin la intervención de ningún equipo central.
- AODV** Ad-Hoc On Demand Distance Vector, vector de distancia en base a demanda en Ad-Hoc. Protocolo de enrutamiento diseñado para redes móviles ad hoc capaz tanto de enrutamiento unicast y multicast.
- APL** Application Layer, Capa de Aplicación. Permite acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos.
- APOs** Application Objects, Objetos de Aplicación. Pieza de software que controla una unidad hardware, cada APO tiene un identificador único.
- ASW** Absorption of seismic waves, Ondas Sísmicas de Absorción.

B

- Beacon** Trama que ayuda a que los nodos se sincronicen al ritmo del coordinador, además evita interferencias en las comunicaciones ya que cada dispositivos de la red escucha antes de transmitir.
- BI** Beacon Interval, duración entre balizas.

Binding	Ligar. Se basa en unir dos dispositivos basados en sus servicios y necesidades.
BO	Beacon Order. Toma valores entre 0 y 14 para funcionamiento balizado o Beacon habilitado.
Broadcast	Difusión. Transmisión de información, nodo emisor envía información a una multitud de nodos receptores de manera simultánea.
C	
CBR	Constant Bit Rate, tasa de velocidad constante. Divide la capacidad total del ancho de banda y se encarga de la parte con un flujo constante de tráfico.
CIRAD	Centro de Investigaciones de Redes Ad Hoc del Departamento de Eléctrica y Electrónica de la Universidad de las Fuerzas Armadas - ESPE.
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance. Protocolo de control de acceso a redes de bajo nivel que permite que múltiples estaciones utilicen un mismo medio de transmisión.
CZ	Coordinador ZigBee. Este dispositivo requiere memoria y capacidad de computación ya que tiene como tarea iniciar la formación de la red inalámbrica.
D	
DARPA	Defense Advanced Research Projects Agency. Agencia del Departamento de Defensa de Estados Unidos responsable del desarrollo de nuevas tecnologías para uso militar.

Delay	Retardo. Tiempo promedio que se demora un paquete durante su transmisión que se envía a través de la red desde el emisor al receptor y viceversa.
Delivery Ratio	Relación de distribución de paquetes. Esto ilustra el nivel de los datos entregados al destino.
E	
EMAPQ	Empresa metropolitana de alcantarillado y agua potable de quito.
F	
FFD	Full Funtion Device, dispositivos con funciones completas.
Fminbnd	Función para minimización escalar. Encuentra el mínimo de una función de una variable.
Fminunc	Función para minimización sin restricciones. Encuentra el mínimo de una función de varias variables sin restricciones.
Fmincon	Función para minimización con restricciones. Encuentra el mínimo de una función no lineal de varias variables con restricciones.
FTP	File Transfer Protocol, protocolo de transferencia de archivos. Protocolo de red para transferencia de archivos entre sistemas conector a una red TCP.
G	
Gateway	Pasarela. Son los elementos destinados a la interconexión entre la red de sensores y una red de datos.
GPS	Global Positioning System, Sistema de Posicionamiento Global.

GTS	Granted Time Slot, Ranura deTiempo Concedido.
H	
Hessiana	Matriz cuadrada de $n \times n$, de las segundas derivadas parciales.
HTTP	Hipertext Transfer Protocol, Protocolo de Transferencia de Hipertexto. Usado en cada transacción de la World Wide Web.
I	
IEEE	Institute of Electrical and Electronics Engineers, Instituto de Ingenieros Eléctricos y Electrónicos
ISM	The industrial scientific and medical radio bands, Bandas de Radio para uso industrial, científico y médico.
IG-EPN	Instituto Geofísico de la Escuela Politécnica Nacional, Ecuador.
M	
MAC	Control de acceso al medio. Se encarga, en general, de gestionar y mantener las comunicaciones entre estaciones.
Multicast	Multidifusión. Envío de la información en múltiples redes a múltiples destinos simultáneamente.
O	
ONL	Optimización No Lineal. La función objetivo y las restricciones a las que está sujeta son no lineales.
OL	Optimización Lineal. La función objetivo y las restricciones a las que está sujeta son lineales.

P

PAN Personal Area Network. Red integrada por todos los dispositivos en el entorno local y cercano de su usuario.

PCG Método de gradientes conjugados pre-condicionados.

R

RMSE Error cuadrático medio. Medida de uso frecuente de la diferencia entre los valores pronosticados y los datos realmente obtenidos.

S

Solver Forma parte de una serie de comandos a veces denominados herramientas. Puede encontrar un valor óptimo de una función objetivo, sujeta a restricciones o limitaciones.

SQP Sequential Quadratic Programming. Método usado en las funciones de minimización con restricciones.

SSE Suma de errores cuadráticos. Mide la desviación total de los valores de respuesta a partir del ajuste a los valores resultantes.

T

Trama Formatos de Bytes enviados por cada paquete transmitido.

TIL Longitud de intervalo de Tiempo.

Throughput Rendimiento. Medida de la tasa de transferencia de datos a través de un sistema de comunicaciones en cierto período.

U

- UDP** User Datagram Protocol. Protocolo del nivel de transporte basado en el intercambio de datagramas (Encapsulado de capa 4 Modelo OSI).
- Z**
- ZDO** ZigBee Device Objects, Objetos para Dispositivos ZigBee.
- Zigbee** Es un conjunto estandarizado de soluciones basado en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (wireless personal área Newark, WPAN), tiene como objetivo las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

RESUMEN

Los sistemas de Monitorización Volcánica basados en Redes de Sensores Inalámbricos (WSN) deben ser implementados con eficacia y asegurando un buen desempeño, pues uno de los factores principales del cual depende es la cantidad de nodos que la componen. En trabajos previos relacionados se emplea un análisis de métricas de Calidad de Servicio, donde se establece que las pérdidas de paquetes son menor al 20%, y prueban el funcionamiento de la red con 12 nodos, sin embargo, no se garantiza el valor óptimo, es por esto que el objetivo primordial de este trabajo de graduación se enfocó en la búsqueda de un número óptimo de nodos, que en conjunto son capaces de recolectar información de un evento volcánico. Mediante la implementación de métodos de optimización de funciones y análisis de métricas de QoS para encontrar el número óptimo de sensores se garantiza que la red es competente y cumple con los requisitos de los sistemas de monitorización. El proceso de optimización se organizó en tres fases: Sobre el simulador *ns-2* fue simulada la WSN con distribución aleatoria. La transmisión de información se recogió durante 2 minutos de la simulación, que muestra resultados tolerables de retardo promedio en unidades de milisegundos. La segunda etapa corresponde a la deducción de la función objetivo, y la última incluye la aplicación de métodos de optimización no lineal. Los resultados muestran que el número óptimo de nodos es 10, se garantiza una pérdida de paquetes menor al 25%. Por lo tanto, una WSN aplicada a monitorización volcánica, se puede desplegar con 10 nodos sensores en futuros escenarios de pruebas reales sin inconvenientes en relación al rendimiento de la red.

PALABRAS CLAVES:

WSN

ZIGBEE/IEEE 802.15.4

METRICAS QOS

MÉTODOS DE OPTIMIZACIÓN

FUNCIÓN OBJETIVO.

ABSTRAC

Volcanic monitoring systems that use Wireless Sensor Networks (WSN), should be implemented effectively and ensuring a good performance, since one of the main factors on which it depends is the number of nodes that compose them. In previous work through the analysis of main metrics to guarantee Quality of Service (QoS), was established that packet losses are less than 20%, when the performance of the network was tested with 12 sensors. However, the optimal value is not guaranteed, that is why the objective of this research focused on finding an optimal number of nodes, which collectively are able to collect information of a volcanic event. By implementing function optimization methods and QoS metric analysis to find the optimum number of sensors, it is guaranteed that the network is competent and meets the requirements of monitoring systems. The optimization process was arranged in three phases: Network WSN on the ns-2 simulator was implemented with randomized distribution. The transmission of information was collected for 2 minutes of simulation, showing tolerable results of average delay in units of milliseconds. The second stage corresponds to the deduction of the algebraic objective function, and the last includes the application of nonlinear optimization methods on objective function. The results show that the optimal number of nodes is 10, it ensures packet loss less than 25%. Therefore, it is determined that WSN applied to Volcanic Monitoring, can be started with 10 sensor nodes in future scenarios of real tests without problem in relation the network performance.

KEYWORDS:

WSN

ZIGBEE/IEEE 802.15.4

QOS METRICS

OPTIMIZATION METHODS

OBJECTIVE FUNCTION.

PRÓLOGO

Las investigaciones realizadas hasta ahora se han dedicado principalmente a sentar las bases sobre el desarrollo de aplicaciones con Redes de Sensores Inalámbricas relacionadas a la Monitorización de variables o señales sísmicas correspondiente a una eventual erupción Volcánica, que es una de las aplicaciones que la Universidad de las Fuerzas Armadas – ESPE, ha venido desarrollando durante los últimos años, y es al cual aporta el presente trabajo, es así que existe un gran interés por determinar el mínimo de nodos que pueden ser utilizados para monitorización de fenómenos y en sí mantener la funcionalidad de la red. La determinación del número de nodos óptimo (mínimos ó máximo) utilizados en la topología de la WSN es un factor clave para la no redundancia de la información de interés [1] .

Este presente trabajo abarca el análisis de éste factor para eso es necesario el manejo de métricas de QoS de la WSN, tales como Rendimiento, Retardo extremo a extremo y Paquetes perdidos, y principalmente mediante un proceso de optimización de funciones de la relación de métricas con uso de algoritmos de optimización se determina el número óptimo de nodos. Inicialmente en el Capítulo uno se realiza la presentación del proyecto donde sobresalen los objetivos que se esperan e interesa alcanzar. Además se describen: los antecedentes, y justificación e importancia que impulsan a la realización del mismo. En el Capítulo dos se realiza un estudio del marco teórico respecto a las redes de sensores inalámbricos detallando sus características, arquitectura, componentes y factores que influyen en el diseño de las WSN, de igual forma se realiza una descripción de la tecnología Zigbee basado en el estándar IEEE 802.15.4 puntualizando sus características, topologías y dispositivos que engloba dicha tecnología. Además, se presentan las métricas de QoS de la red, la optimización de funciones y técnicas de optimización que se emplean en los capítulos posteriores. En el Capítulo tres se describen los escenarios de simulación mediante un análisis de la distribución randómica de elementos y parámetros de diseño de la red, seguido se simula la red en un intervalo de tiempo de 140 segundos sobre la plataforma *ns* versión 2.34 en base a topología estrella, inyección de tráfico CBR, protocolo de enrutamiento AODV y modelo de propagación dos rayos (*Two-Ray Ground*). Posterior

con ayuda de la herramienta *Curve Fitting* de MATLAB® se determinan las funciones algebraicas de interés que representen un nivel de ajuste de curva tolerable dentro de las estadísticas de bondad de ajuste. Adicionalmente se presentan todos los resultados hallados en las pruebas ejecutadas en las simulaciones, se tabula los datos extraídos y se presentan las gráficas resultantes. En el Capítulo cuatro se realiza un estudio minucioso de varios algoritmos de optimización de funciones tanto lineales como no lineales, se define dos técnicas de optimización apropiadas y flexibles para este tipo de aplicación. Y se muestran los datos y gráficos obtenidos tras la aplicación de los dos métodos de optimización sobre las funciones algebraicas que relacionan las variables de interés, además de una comparación entre los dos resultados obtenidos se define finalmente el número de nodos óptimo con el que se puede dar despliegue y funcionamiento a la red de sensores para Monitorización Volcánica. Finalmente en el Capítulo cinco se contempla las principales contribuciones y conclusiones que se determinan con el desarrollo del proyecto.

CAPÍTULO 1

PRESENTACIÓN DEL PROYECTO

1.1 INTRODUCCIÓN

El presente proyecto hace referencia a la optimización del número de nodos de una red de sensores inalámbricos que por el tipo de sensores y tecnología son aplicables a Monitorización Volcánica.

Zigbee sobre las redes de sensores inalámbricas como tecnología ha sido adoptada e implementada en diversas disciplinas y proyectos, los mismos que contribuyen principalmente con la sociedad, así como en el campo de la meteorología, una red de sensores con las características de WSN permite la fácil obtención, recolección y transmisión de datos de distinta naturaleza, información importante para ciertos sectores como en sistemas de alertas tempranas activados por algún fenómeno ambiental que está siendo monitorizado y que permitirá precautelar la integridad física de la población.

En el presente Capítulo se expone la situación actual de las redes de sensores inalámbricas en el mundo así como la tecnología utilizada. Además se presenta la recopilación de antecedentes, objetivos que justifican el proyecto y un breve detalle los programas y herramientas que se utilizaron para su elaboración. Nuestro trabajo investigativo responde a la pregunta ¿Cuál es el número nodos óptimo con los que se puede desplegar una WSN para Monitorización Volcánica?, en consecuencia se mantuvo un nivel estable de *Throughput* con la definición de un umbral respecto a la pérdida de paquetes y con esto se evitó en gran parte el incremento del retardo extremo a extremo.

1.2 ANTECEDENTES

Las redes de sensores inalámbricas en los últimos años han tenido un gran avance debido a la respuesta que dan a las exigencias actuales referentes al establecimiento de redes que cubran necesidades de comunicación de forma flexible en tiempo, espacio, autoconfiguración e independencia de estructura. Este tipo de redes permiten el despliegue de un elevado número de nodos, no sujetos a restricciones impuestas por el cableado. Los dispositivos son de bajo costo y elevada duración capaces de obtener información del entorno y reenviarla de forma inalámbrica a un centro de coordinación ofreciendo así una multitud de aplicaciones [2], tales como: la Monitorización del ambiente, la atención de la salud, la gestión de la energía, la construcción y la automatización [3]. Además este tipo de tecnología ofrece posibilidades para la prevención de desastres naturales lo cual es de gran importancia para la comunidad científica y la sociedad en general.

Las WSN son clasificadas como redes ad-hoc, se distinguen por estar compuestas de un conjunto autónomo y disperso de nodos conectados por enlaces inalámbricos que no precisan de una infraestructura fija. Son fabricados para operar en ambientes hostiles e irregulares, siendo capaces de modificar su topología para adoptar nuevas formas y mantener la red funcional. Estos pequeños dispositivos electrónicos tienen acceso al mundo exterior y a variables de interés por medio de nodos sensores o motes, que constituyen la pieza central de las redes de sensores. Un mote, es un elemento que combina capacidades de recolección, procesado y transmisión de datos en un mismo dispositivo, logrando un reducido costo económico, tamaño y consumo de potencia, controlan diversas condiciones en distintos puntos, entre ellas la temperatura, el sonido, la vibración, la presión, el movimiento [4], etc.

Las ventajas principales que presentan estos Dispositivos es que funcionan con baterías (similares a las de un teléfono móvil), incluso en ocasiones pueden ser recargadas mediante placas solares si la aplicación lo requiere, Además las comunicaciones que realizan se basan en tecnologías inalámbricas de bajo consumo, lo que les permite pasar del estado de latencia a realizar la transmisión necesaria y rápidamente volver a ese estado de mínimo consumo energético [3].

Las redes de sensores tienen origen en el ámbito tecnológico, de aplicaciones de carácter militar. La primera de estas redes fue desarrollada por Estados Unidos durante la guerra fría y se trataba de una red de sensores acústicos desplegados en el fondo del mar cuya misión era descubrir la posición de silenciosos submarinos soviéticos, esta red se denominó Sistema de Vigilancia de Sonido (*Sound Surveillance System - SOSUS*). Y a partir de 1980, la agencia del Departamento de Defensa de los Estados Unidos responsable del desarrollo de nuevas tecnologías para uso militar DARPA (*Defense Advanced Research Projects Agency*) comienza un programa focalizado en sensores denominado Redes de Sensores Distribuidos (*Distributed Sensor Networks - DSN*), gracias a él se crearon sistemas operativos (*Accent*) y lenguajes de programación (*SPLICE*) orientados de forma específica a las redes de sensores. Es así que en la década de los 90, DARPA lanza un nuevo programa enfocado hacia redes de sensores llamado SensIt, su objetivo viene a mejorar aspectos relacionados con la velocidad de adaptación de los sensores en ambientes cambiantes y asegurar que la información que recogen los sensores sea fiable [5]. Y a finales de la década fue cuando los sensores empezaron a decrecer en tamaño y a incrementar en su autonomía, es por esto que compañías como Crossbow y Dust Inc han desarrollado nodos sensores del tamaño de una moneda con baterías que les hacen tener una autonomía razonable y una independencia inédita [6].

En la última década muchas entidades dentro y fuera del país han realizado investigaciones acerca de este tipo de redes enfocándose en la prevención de desastres naturales, entre los proyectos más destacados con estrecha relación a la Monitorización Volcánica se tiene a las Universidades de Harvard, New Hampshire y la Universidad de Carolina del Norte que en conjunto y la asistencia del Instituto Geofísico de la Escuela Politécnica Nacional (IG-EPN), Ecuador han conducido en el año 2004 la implementación de una WSN para monitorización de erupciones en el Volcán Tungurahua, basándose en la plataforma Mica2, se desplegó 3 nodos acústicos de baja frecuencia y se transmitió los datos a través de un enlace inalámbrico de 9 km hacia una estación remota donde se recolectaron datos continuamente durante 54 horas, el sincronismo en los nodos era mediante el uso de receptores GPS (*Global Positioning System*) [7]. En el siguiente año desplegaron una red de 16 sensores sobre el Volcán activo El Reventador en el Ecuador. La transmisión de información se dio mediante un radioenlace de 10 km entre la estación maestra (Encargaba de la recolección de información sismo-acústica) y la estación base.

Cada estación consistía en un nodo Moteiv Tmote Sky [8], un sismómetro, un micrófono, y un tablero de interfaz de hardware personalizado [9]. Finalmente con el despliegue de las dos redes demostraron la viabilidad del uso de sensores inalámbricos para estudios y aplicaciones volcánicas. Ya que permitiría desarrollar más herramientas sofisticadas para la instrumentación Volcánica [9].

En el año 2011 en la Universidad de las Fuerzas Armadas – ESPE se desarrolló un proyecto de fin de carrera denominado Implementación de un Sistema de Monitorización de Señales Sísmicas del Volcán Cotopaxi Empleando una Red de Sensores Inalámbricos, que constan de dos despliegues realizados con 9 y 5 nodos sensores manufacturados por Crossbow Technology los cuales se configuraron utilizando el *software* MoteConfig 2.0 [1]. Con el banco de pruebas reales ejecutadas se determinó que el despliegue con 5 nodos disminuye costos y reduce el tráfico de la red e incluso reduce la redundancia de datos además se obtiene mejor información sísmica que al desplegar 9 nodos sensores, sin embargo surge un inconveniente en cuanto se refiere al número de enlaces posibles donde circulan los paquetes de información, al poseer nueve sensores disminuyen las rutas alternativas de comunicación lo que significa que se tendría mayor inconveniente al re-direccionar si un enlace falla.

Mientras que en el año 2013, la Escuela Superior Politécnica de Chimborazo como aporte a las investigaciones de Monitorización de eventos Volcánicos desarrolló el proyecto de fin de carrera titulado Evaluación de las Topologías Físicas de WSN Mediante la Implementación de un Prototipo de Medición de Variables Ambientales, donde se enfocaron en la plataforma libre Arduino y los módulos de comunicación XBee. La red fue formada por 10 nodos y 1 coordinador con esto se determinó que la topología física más adecuada para implementar una red WSN para mediciones ambientales es la topología estrella, que cuenta con un *Packet Delivery Ratio* del 99%, *Throughput* de 0,7542, un retardo de 0,01621453 ms, un consumo de energía bajo y mayor área de cobertura [10].

El estudio de los proyectos relacionados aporta con muchos puntos de interés para el desarrollo de esta investigación ya que cada uno ofrece de forma particular un escenario compuesto por diferente cantidad de nodos sobre la WSN, es así que, en el desarrollo de

éstos trabajos ninguno llevó a cabo un estudio minucioso para fundamentar el número de sensores con el cual se despliega la red y se realiza las pruebas simuladas y reales.

1.3 JUSTIFICACIÓN E IMPORTANCIA

En los últimos 10 años hemos asistido a un explosivo crecimiento de las redes en general y en concreto de las comunicaciones inalámbricas propiciado por los continuos avances tecnológicos, por tal motivo actualmente el diseño de una red de sensores está motivado o influido por un requisito tal como el despliegue masivo y aleatorio, es decir el escenario de la mayoría de las WSN contienen un número grande de sensores (del orden de cientos o millares, como pocos) [2], que pueden diseminarse de forma aleatoria o fija en las áreas de monitorización definidas por el usuario. En [1] mediante la puesta en marcha de dos escenarios diferentes bajo el estándar Zigbee/ IEEE 802.15.4, comprueba que con un número mayor de nodos sensores los datos obtenidos en uno de ellos es redundante con respecto a otros nodos. Por el contrario con un número menor se disminuye costos y tráfico de la red. Es así que el despliegue de una red con un alto número de nodos que pueden estar a pocos metros unos de otros hace necesario un manejo cuidadoso en el mantenimiento de la topología [11]. Por estos motivos, encontrar el rango o número de sensores óptimo para un sistema de monitorización es la propuesta que se desarrolla en este proyecto de graduación, que aporta con: El análisis y manejo de métricas de QoS que establecen el desempeño de la red. Así como la determinación de las funciones algebraicas correspondientes a la relación de éstas. Finalmente con la ejecución del proceso de optimización de dichas funciones en base algoritmos de optimización numérica, hallar como respuesta un valor óptimo para la función objetivo equivale al número de nodos óptimo de la red.

En general este trabajo hace uso de; una topología definida, características de módulos (XBee serie 2), tipo de tráfico específico y datos de un evento sísmico volcánico cercano a la realidad, con el objetivo de encontrar con más claridad y precisión el número óptimo de nodos.

1.4 ALCANCE DEL PROYECTO

El proyecto de forma específica comprende el análisis y aplicación de técnicas de optimización sobre variables correspondientes a la información de métricas QoS extraídas de una red de sensores inalámbricos, se pretende encontrar un rango o número de nodos óptimo donde el retardo extremo a extremo sean mínimo y el rendimiento de la red no tenga gran disminución y sea suficiente para establecer el evento que acontece en el área monitorizada.

El desarrollo del proyecto se inicia con un análisis de formas de distribución de elementos sobre la red como fija o randómica (Distribución de nodos aleatoriamente sobre una superficie). Seguidamente bajo la topología seleccionada se simula el funcionamiento de la red en la plataforma de simulación *ns-2.34* WSN, donde se contempla características y parámetros del medio, tipo de módulos de comunicación, tipo de tráfico que circula por la red, evento volcánico que se quiere simular, protocolo de enrutamiento, etc.

Efectuando un análisis de los resultados obtenidos de las simulaciones se determina un rango de nodos inicial entorno a la topología elegida, posteriormente se utilizan como valores iniciales para la búsqueda y determinación de las funciones o polinomios algebraicos que representan las relaciones existentes entre dichas variables, las funciones pueden ser de univariables o multivariables dependiendo del caso que se analice.

Se realiza un estudio minucioso de varios algoritmos, funciones y métodos de optimización algebraicos y numéricos que son adaptables para funciones y ecuaciones tanto lineales como no lineales que se encuentran inmersos en la *Optimization Toolbox*TM de MATLAB®, seguido se aplican los métodos de optimización que resulten de la investigación sobre los datos de la red, teniendo como variables el número de nodos y las métricas de QoS, Se determina cuál de los métodos son apropiados y flexibles respecto a los datos ingresados y mediante un análisis comparativo se determina el nivel de ajuste de cada uno respecto al cumplimiento de requisitos para el despliegue de una WSN como sistema de Monitorización Volcánica.

Finalmente se realiza un análisis de los resultados obtenidos de aplicar las técnicas de optimización, definiendo así el número óptimo de nodos sensores con los cuales se pueda

implementar en un futuro una red optimizada para Monitorización de Eventos Sísmicos Volcánicos.

1.5 OBJETIVOS

1.5.1 GENERAL

Optimizar el número de nodos en una red de sensores inalámbricos para la Monitorización de señales sísmicas del Volcán Cotopaxi que maximice el *Throughput* y minimice la pérdida de paquetes y el retardo.

1.5.2 ESPECÍFICOS

- Establecer el marco conceptual y teórico en el que se basa el funcionamiento y el estándar IEEE 802.15.4 de las redes de sensores inalámbricas que serán base fundamental para establecer los escenarios de simulación.
- Simular una red de sensores inalámbricos basada en distribución aleatoria de elementos, que permita determinar el rango de nodos con el cual el desempeño teórico de la red sea suficiente para poder interpretar los datos.
- Proveer una introducción de la teoría que es la base del desarrollo de varios algoritmos de optimización de funciones, con el fin de seleccionar dos de éstas que cubran los requerimientos ineludibles.
- Aplicar al menos dos métodos de optimización sobre el rango de nodos inicial resultantes de las simulaciones, haciendo uso del *Software* MATLAB y sus herramientas necesarias para su respectivo análisis.
- Determinar el número de sensores óptimo sujeto aplicando métodos de optimización, donde el valor del *Throughput* sea suficiente para detectar cualquier evento sísmico.
- Realizar un análisis comparativo de los resultados obtenidos de aplicar las dos técnicas de optimización basados en valores simulados.

CAPÍTULO 2

FUNDAMENTO TEÓRICO

2.1 REDES DE SENSORES INALÁMBRICAS

2.1.1 INTRODUCCIÓN

El desarrollo de las WSN se ha dado debido a su aplicación en distintos sectores (seguridad, medio ambiente, industria, agricultura etc.) tanto comerciales como militares, esta tecnologías han tenido un rápido progreso en los últimos años respaldado por los continuos avances tecnológicos. Así, se han podido desarrollar dispositivos sensores electrónicos de menor tamaño, bajo costo y bajo consumo, capaces de procesar información de manera local y comunicarse de forma inalámbrica.

En la actualidad pueden encontrarse diferentes tipos de sensores en un gran número de sistemas y dispositivos electrónicos, los principales analistas tecnológicos, dentro de las tecnologías inalámbricas, valoran las redes inalámbricas de sensores como una de las opciones de futuro más prometedora, fabricantes como Microsoft, Intel, IBM, Motorola y Texas Instruments, han lanzado líneas de investigación en base a esta tecnología.

En la mayoría de las aplicaciones estos sensores actúan únicamente como transductores, realizando la medición de una o más variables del entorno y enviando esta información a un nodo central que se encarga de su procesamiento. Se ha avanzado tremendamente en una nueva generación de sensores dotados de inteligencia propia, capaces de organizarse e interconectarse de forma autónoma con otros sensores de semejantes características [11]. Las comunicaciones inalámbricas se realizan usando un

nuevo protocolo inalámbrico o tecnología denominado ZigBee el cual permite enviar información a largas distancias con un consumo energético mínimo.

En este capítulo se presenta una investigación acerca de las redes de sensores inalámbricos, así como a sus conceptos y fundamentos teóricos. Además se tendrá un enfoque general de las tecnologías asociadas y elementos a las WSN.

2.1.2 CARACTERÍSTICAS DE LAS WSN

Las Redes de sensores inalámbricas comparten muchas características respecto las redes AD-HOC, y tienen algunas propias resultantes de su estructura y comportamiento, a continuación se puntualizan las características más generales:

- **Topología variable:** La posición de cada nodo sobre la red puede ser arbitraria y generalmente puede ser desconocida por los otros nodos. Por otro lado, los algoritmos y protocolos de red deberán de poder organizarse automáticamente.
- **Energía (alimentación):** El consumo de energía de las redes WSN es bajo y eficiente al igual que la memoria de cada nodo [12]. Estas redes funcionan con baterías y tienen una larga autonomía de funcionamiento por esta razón pueden operar sin mantenimiento durante varios meses o incluso años y actualmente los sensores pueden captar energía del ambiente y utilizarla para alimentarse.
- **Infraestructura de red:** Una red de sensores no tiene necesidad alguna de infraestructura para poder operar, ya que sus nodos pueden actuar de emisores, receptores o *router* [6].
- **Desplegado AD-HOC:** Es decir las WSNs tiene un despliegue a gran escala sobre una superficie donde el sistema debe adaptarse a los cambios en la conectividad de la red como resultado de las fallas de los nodos.
- **Gran Escala:** La cantidad de nodos desplegados en una red puede llegar hasta los miles, pudiendo crecer su número a lo largo de la vida de la red [13].
- **Tolerancia a fallas:** La falla de un sensor no debe de afectar el funcionamiento de la red; si un nodo falla, los protocolos MAC y de ruteo deben re-direccionar la información. Esto puede requerir ajustes en la potencia de transmisión y las tasas de

señalización en los enlaces existentes para reducir el consumo de energía o redireccionar los paquetes hacia regiones de la red con mayor disponibilidad de energía. Es decir, se requieren niveles de redundancia para la tolerancia a fallas [11].

- **Comunicación:** Los nodos sensores usan comunicación por difusión y debido a que están densamente desplegados, los vecinos están muy cerca unos de otros y la comunicación multihop (salto múltiple de uno a otro) consigue un menor consumo de potencia que la comunicación single hop (salto simple). Además, los niveles de transmisión de potencia se mantienen muy bajos y existen menos problemas de larga distancia.
- **Recursos limitados:** Los sensores, a cambio de su bajo consumo de potencia, costo y pequeño tamaño disponen de recursos limitados. Los dispositivos actuales más usados, los mica2, cuentan con procesadores a 4 MHz, 4 kbytes de RAM, 128kbytes de memoria de programa y 512 kbytes de memoria flash para datos. Su radio permite transmitir a una tasa de 38.4 kBaudios [13].

2.1.3 ARQUITECTURA DE COMUNICACIÓN DE LAS WSN

Las redes inalámbricas han tomado distintas distribuciones geográficas con respecto a su topología. Cuando un nodo sensor toma medidas o sucede cualquier actividad programada con el fenómeno en cuestión, la información recogida se convierte en digital en el propio nodo, y es transmitida fuera de la red, vía pasarelas (*Gateway*), a una estación base, donde la información es almacenada y tratada para acabar finalmente en un servidor con mayor capacidad que permita realizar un análisis de los datos.

Tomando como elementos principales de la red a los nodos sensores, la *Gateway* y las estaciones base, podemos distinguir dos tipos principales de arquitecturas.

2.1.3.1 Arquitectura centralizada

En las redes inalámbricas centralizadas (*Centralized Wireless Networks*) existe un nodo central que realiza la función de enlace entre los distintos equipos que constituyen la

red. Este nodo puede encaminar la información que circula por la red bien a una red cableada convencional, bien hacia otras redes o equipos distintos. Para poder realizar la comunicación con este nodo central o punto de acceso (*Access Point*, AP), los dispositivos de la red deben estar en el radio de cobertura del AP [14].



Figura. 2.1. Arquitectura WSN centralizada

2.1.3.2 Arquitectura distribuida

Las redes inalámbricas distribuidas (*Distributed Wireless Networks*) también denominadas redes ad-hoc, se caracterizan porque los dispositivos que constituyen la red pueden comunicarse entre sí siempre y cuando ambos se encuentren dentro de su radio de cobertura. La comunicación entre los dispositivos es directa, sin la intervención de ningún equipo central.



Figura. 2.2. Arquitectura WSN distribuida

Los nodos sensores se van a comunicar entre sus nodos vecinos y ejecutan algoritmos distribuidos para obtener una única respuesta global que un nodo (*cluster head*) se encargará de comunicar a la estación base a través de las pasarelas pertinentes.

2.1.3.3 Arquitectura de las WSNs

Las WSNs son comúnmente desplegadas dentro de un área trabajo de sensores como se presenta en la Figura 2.3.

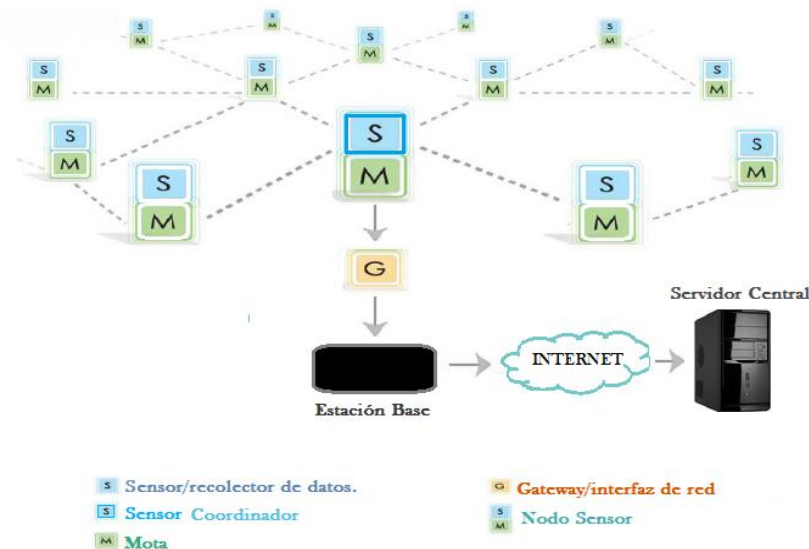


Figura. 2.3. Arquitectura general de una WSN

Cada uno de los nodos sensores de la red posee diversas funcionalidades que le permiten la recolección de datos y su enrutamiento hasta llegar a su destino. Hasta que la información recopilada llega al destino o usuario ésta debe pasar a través de una estructura distribuida de nodos ad-hoc y alcanzar al elemento *Gateway* o interfaz de red, el cual se encargará de centralizar toda esta información, para seguido pasar a la estación Base el cual almacenará y enviará los resultados al servidor central. Desde este punto hasta el nodo-central (Servidor), los datos pueden llegar vía Internet o conexión por satélite, el servidor tiene mayor capacidad y permite componer una secuencia histórica, realizar un análisis o procesamiento de los datos.

La manera en que se comunican y trabajan los distintos elementos que constituyen los nodos es basándose en la tecnología ZigBee el cual se define como una pila de protocolos que permite la comunicación de forma sencilla entre múltiples dispositivos. La Figura 2.4 especifica las diversas capas, adecuándose al modelo OSI, además en ella vemos como internamente los sensores están constituidos por ésta pila de protocolos.

Dicha pila combina las características propias de las distintas capas (heredadas de las redes de datos tradicionales) con las nuevas propiedades de las redes de sensores (energía, movilidad, entorno, etc.). En términos generales la pila de protocolos está compuesta por cinco capas: aplicación, transporte, red, enlace de datos y físico. En cada una de ellas se han desarrollado múltiples protocolos [15].

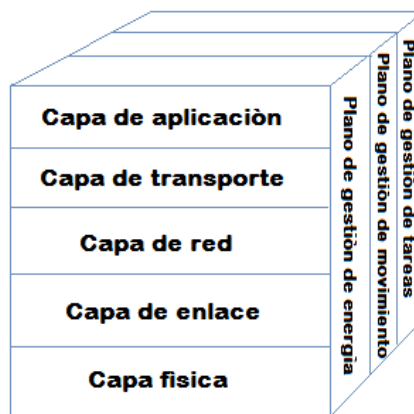


Figura. 2.4. Pila de protocolos en una WSN

2.1.4 COMPONENTES DE UNA WSN

Las WSN se componen de miles de dispositivos pequeños, autónomos, distribuidos geográficamente, llamados nodos sensores, al igual que estos elementos la WSN también está compuesta por los sensores, pasarelas, la estación base y la red inalámbrica.

2.1.4.1. Sensor

Estos gozan de capacidad de cómputo, almacenamiento y comunicación en una red conectada sin cables, e instalados alrededor de un fenómeno objeto para monitorizarlo, pueden ser de distinta naturaleza y tecnología, toman del medio la información y la convierten en señales eléctricas.

2.1.4.2. Nodo Sensor

Son los procesadores de radio, que toman los datos del sensor a través de sus puertas de datos, y envían la información a la estación base. Se compone de una mota y una placa

de sensores. Mota es la entidad compuesta de un procesador y los dispositivos de radio. La placa del sensor es una tarjeta de adquisición de datos conectado a la mota a través de un conector de expansión, que incluye un conjunto de sensores. Algunos modelos de nodo de sensores incorporan los sensores en la propia mota (por ejemplo, las motas de la familia *Telos*).

2.1.4.3. Pasarelas o Gateways.

Son los elementos destinados a la interconexión entre la red de sensores y una red de datos (por ejemplo TCP/IP).

2.1.4.4. Estación base

Actúa como recolector de datos basado en un ordenador común o un sistema integrado.

2.1.4.5. Red inalámbrica.

Típicamente basada en el estándar 802.15.4 - ZigBee.

2.1.5 FACTORES QUE INFLUYEN EN EL DISEÑO DE LAS WSNs

El diseño de una red de sensores inalámbrica está influenciado por muchos factores, los cuales incluyen tolerancia a fallos, escalabilidad, costos de producción, restricciones de hardware, topología de la red, ambientes de operación, medios de transmisión y consumo de energía. Estos factores son importantes porque sirven de guía para diseñar un protocolo o un algoritmo para WSNs.

2.1.5.1. Tolerancias a fallos

Algunos nodos sensores pueden fallar o ser bloqueados debido a la carencia o falla de energía, tener daños físicos o interferencia por causa del ambiente que lo rodea. Sin

embargo, estos aspectos no deberán afectar a las tareas de la WSN. La tolerancia a fallos es la destreza para mantener la funcionalidad de las WSNs sin interrupciones a pesar de fallos de los nodos y lograr retransmitir paquetes perdidos.

2.1.5.2. Escalabilidad

El número de nodos sensores desplegados dentro del área para la monitorización de un evento puede ser del orden de cientos o miles. Dependiendo de la aplicación, el número puede alcanzar hasta un valor de millones tornándose un reto importante. En lo que respecta a la densidad, ésta puede ir desde unos pocos sensores hasta unos pocos cientos de sensores en una región, los cuales pueden estar a menos de diez metros de distancia [16].

2.1.5.3. Costes de producción

Debido a que las WSNs se componen de un gran número de nodos sensores, el coste de un nodo es muy importante para justificar el coste total de las redes del sistema. Si el coste de la red de sensores inalámbricos es más elevado que el despliegue de sensores tradicionales, entonces el coste de la WSN no se justifica. Como resultado, el coste de cada nodo sensor tiene que mantenerse bajo.

2.1.5.4. Restricciones del hardware

El hardware de un sensor que se utilice puede imponer ciertos límites, como pueden ser: Capacidad de procesamiento de datos y almacenamiento limitada, tamaño, energía disponible, arquitectura, modularidad, capacidad de comunicación, etc.

Para lograr determinar las restricciones que puedan presentarse en una WSN debido al hardware, es necesario conocer detenidamente la estructura del sensor para la monitorización, éste variará dependiendo del fabricante.

En general un nodo sensor está compuesto de cuatro componentes básicos: una unidad de captación o monitorización con sensores, una unidad de procesamiento, una unidad de transmisión recepción y una unidad de energía como se observa en la Figura 2.5. Estos componentes también pueden tener aplicaciones que dependan de componentes adicionales, tales como sistemas de localización, un generador de energía y un generador de movimiento.

Las unidades de monitorización están usualmente compuestas de dos sub-unidades: el sensor y un convertidor analógico-digital (ADC). Las de señales analógicas producidas por el sensor están basadas en la obtención del evento, y son convertidas a señales digitales por el ADC, y después procesadas dentro de la unidad de procesamiento [16].

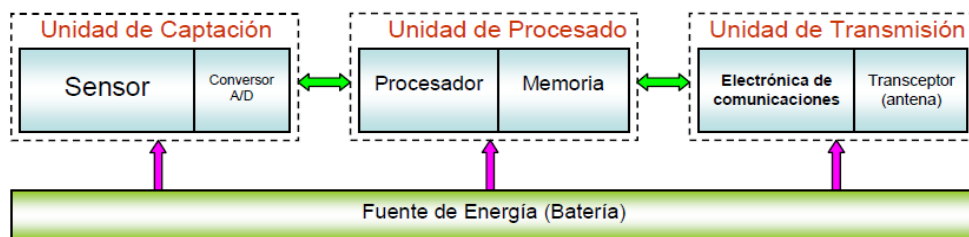


Figura. 2.5. Arquitectura básica de un nodo sensor [17]

2.1.5.5. Topología de la red de sensores

El despliegue de un alto número de nodos que pueden estar a pocos metros unos de otros, hace necesario un manejo cuidadoso de la topología. Se puede identificar tres fases relacionadas en las cuestiones de mantenimiento y cambio en una red de sensores:

- **Fase de despliegue y preparación previa:** Los sensores pueden ser desplegados de múltiples formas en la zona de interés: De forma masiva, uno a uno. Pueden ser lanzados desde un aeroplano o depositados uno a uno por un ser humano.
- **Fase posterior al despliegue:** Posteriormente al despliegue, hay muchos factores que pueden provocar cambios en la topología: Interferencias, ruido,

obstáculos móviles, terreno, etc. Una red de sensores puede estar sometida a frecuentes cambios después de la fase anterior.

- **Nuevas fases de despliegue:** Podría llegar a ser necesario una nueva fase de despliegue por varios motivos como reemplazar los sensores que han fallado en la fase anterior o por necesidades de cambio de tareas dinámicas.

2.1.5.6. Ambiente de operación

Las Redes de Sensores Inalámbricos generalmente se encuentran desplegadas en áreas remotas y desatendidas, donde pueden estar sometidas a un gran número de condiciones adversas: Contaminación química, en un edificio, una fábrica, sometidos a una gran presión atmosférica o temperaturas extremas, entornos con una gran cantidad de ruido electromagnético, etc. Esta gran diversidad de escenarios y condiciones extremas donde pueden estar obligadas a trabajar, influyen en aspectos como, la comunicación entre nodos y tasa de fallos.

2.1.5.7. Medios de transmisión

En una red de sensores que utiliza una comunicación *wireless* se pueden establecer enlaces entre los nodos mediante radio, sistemas ópticos o infrarrojos.

En el caso de las restricciones de hardware aplicables a redes de sensores de bajo consumo y bajo costo, existe una gran número de componentes basados en la banda ISM de 433 y 915 MHz o circuitos RF operando en el rango de 2.4 GHz, así como es posible el uso combinado para llevar a cabo ciertas tareas de localización. El uso de una banda u otra tiene ciertas discrepancias en algunos fenómenos como la atenuación, absorción, relación señal-ruido, entre otras.

2.1.5.8. Consumo de energía

Uno de los componentes más importantes de un nodo sensor es la unidad de energía, ésta puede ser proporcionada por una unidad generadora de energía, tal como celdas solares.

Un nodo sensor *wireless* está generalmente equipado con una fuente limitada de energía. En muchos escenarios, la utilización de una fuente de energía alternativa es imposible, limitando la fuente de energía del nodo al uso de baterías o acumuladores. La vida útil del sensor queda limitada entonces al tiempo de vida útil de la batería.

Debido a la naturaleza de este tipo de redes, cada uno de sus elementos juega dos papeles principales dentro de ella: Generador (datos locales) y enrutador (datos externos). El fallo de alguno de ellos implica un cambio en la topología y obliga a la reorganización de la red, modificando las tablas de *routing* de los nodos.

La conservación y gestión de energía adquiere una importancia adicional; motiva el empleo y desarrollo de nuevos protocolos y algoritmos. Las principales tareas de un nodo sensor son: Detectar eventos, realizar un mínimo tratamiento de datos local y finalmente transmisión y retransmisión de datos. El consumo de energía se distribuye entonces en tres dominios principales: Detección, comunicación y procesamiento de datos.

2.2 ZIGBEE Y EL ESTÁNDAR 802.15.4

ZigBee es un protocolo de comunicaciones inalámbrico basado en el estándar para Redes Inalámbricas de Área Personal (WPANs, *Wireless Personal Area Networks*) IEEE 802.15.4, que ofrece confiabilidad, robustez, escalabilidad, seguridad y bajo consumo de potencia. Este estándar inalámbrico surge del fruto de una alianza, sin ánimo de lucro, de más de 300 empresas, tales como Invensys, Mitsubishi, Honeywell, Philips y Motorola, etc, que trabajan para crear un sistema estándar de comunicaciones, vía radio y bidireccional. El objetivo fue conseguir el desarrollo e implantación de una tecnología inalámbrica de bajo costo, sin perder potencia ni posibilidades.

La especificación ZigBee se basa en el estándar IEEE 802.15.4. De hecho se complementan proporcionando una pila completa de protocolos que permiten las comunicaciones entre una multitud de dispositivos de una forma eficiente y sencilla. [18]

ZigBee amplía el estándar IEEE 802.15.4 aportando una capa de red (NWK, Network Layer) que gestiona las tareas de enrutado en una red de múltiples saltos y de mantenimiento de los nodos en la red; y un entorno de aplicación que proporciona una subcapa de aplicación (APS, *Application Sublayer*) que establece una interfaz para la capa de red, y los objetos de los dispositivos tanto de ZigBee como del diseñador [18], se puede apreciar en la Figura 2.6.

La capa de red (NWK) tiene tres objetivos principales; asociación o disociación de los dispositivos usando el coordinador de red, implementación de seguridad y encaminamiento de tramas a su destino. Además la capa de red del coordinador de red es responsable de iniciar una nueva red y asignar una dirección a los dispositivos nuevos asociados.

A nivel de red, existen dos tipos de direcciones: direcciones cortas (16 bits) y direcciones largas o direcciones IEEE (64 bits). Cada dispositivo debe tener asignada una dirección IEEE única. No puede haber dos dispositivos que cumplan con la especificación ZigBee y que posean la misma dirección IEEE. Así, esta dirección es asignada en el momento de la fabricación del dispositivo. Por contra, la dirección corta es asignada por la capa de red de forma dinámica. Dentro de una red ZigBee no puede haber más de un dispositivo con igual dirección corta [19].

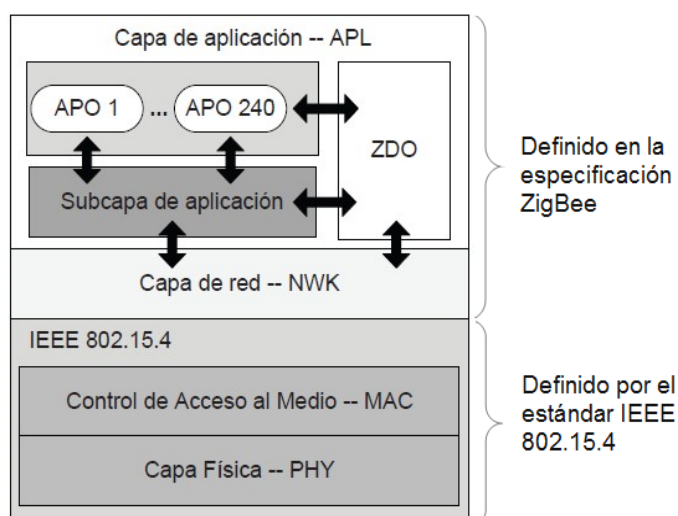


Figura. 2.6. Arquitectura en capas ZigBee/802.15.4 [20].

La capa de aplicación (*Application Layer – APL*) consiste en: la subcapa de aplicación (*Application Sub Layer – APS*), los objetos para dispositivos ZigBee (*ZigBee Device Objects – ZDO*) y los objetos de aplicación (*Application Objects – APOs*), definidos por el fabricante.

Un APO es una pieza de *software* que controla una unidad hardware (transductor, conmutador, lámpara) disponible en el dispositivo. Cada APO tiene un identificador único.

Un ZDO es un objeto especial que ofrece servicios a los APOs (descubrir dispositivos en la red y que servicios implementa, servicios de administración de seguridad, trabajo en red y comunicación) [20]. Fueron creados para simplificar el manejo de la red por las aplicaciones de los usuarios, es también responsable de iniciar y responder peticiones de unión a una red.

El APS es responsable de mantener tablas para ligar y enviar mensajes entre dispositivos es decir a los para los APOs y los ZDO. Es encargado del proceso *Binding* (Ligar), que es la habilidad de unir dos dispositivos basados en sus servicios y necesidades. También responsable de descubrir otros dispositivos que operan en el área local.

Los objetos ZigBee contienen perfiles de dispositivos ZigBee (*ZDP: ZigBee Device Profile*) que solo se ocupan del manejo de red y no del intercambio de datos específico de la aplicación. ZDP provee de un conjunto de comandos y respuestas para [21],

- Realizar una exploración del canal.
- Descubrir dispositivos.
- Manejo de la potencia de transmisión.

La capa física gestiona la transmisión y recepción de datos usando determinados canales radio. En 802.15.4 tenemos tres posibles bandas de frecuencia en las que poder trabajar: 868 MHz, 915 MHz y 2,4 GHz. En la primera banda únicamente tenemos un canal entre 868 y 868,6 MHz. En la siguiente banda tenemos 10 canales entre 902 y 928 MHz. Por último, 16 canales más están localizados entre 2,405 y 2,48 GHz. El protocolo permite una selección dinámica del canal, de forma que se elija el menos ruidoso de entre

los posibles. Lógicamente, la tasa de transferencia es distinta para cada una de las bandas de frecuencia. Para 868 MHz la tasa de transferencia es de 20 kbps, para 915 MHz es de 40 kbps y para 2,4 GHz es de 250 kbps. Por supuesto, las bandas de 868 y 915 MHz tienen un rango de alcance mayor que la de 2,4 GHz a cambio de una menor tasa de transferencia. Otra diferencia es que la banda de 868 MHz está disponible sólo en Europa mientras que la de 915 MHz sólo en Estados Unidos y Australia. Sin embargo la banda a 2,4 GHz es universal, por lo que su uso está permitido en la mayoría de países del mundo. En la Figura 2.7 podemos ver una representación de los canales radio 802.15.4 [22].

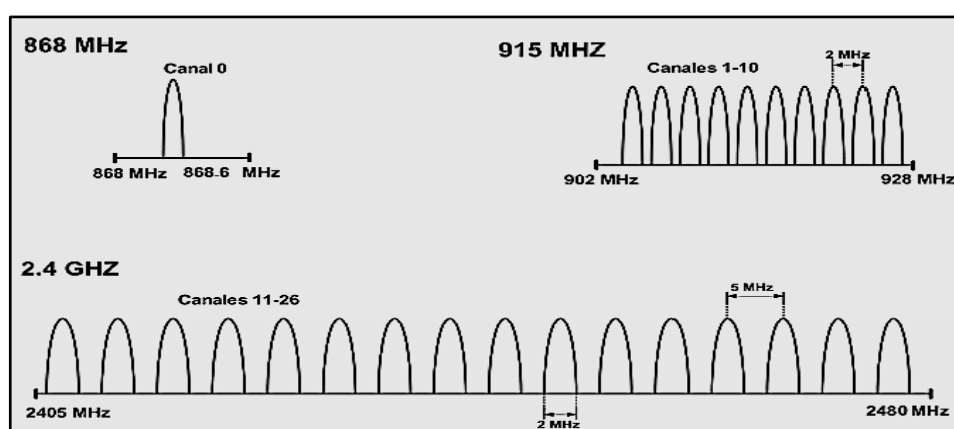


Figura. 2.7. Canales de Frecuencias usadas por IEEE 802.15.4

En 802.15.4 se utiliza DSSS (*Direct Sequence Spread Spectrum*) como técnica de ensanchado de espectro. En la Tabla 2.1 se muestra la tasa de transferencia, número y espaciamiento de canales y el tipo de modulación usada para las distintas frecuencias de transmisión.

Tabla. 2.1.

Características de las Bandas de frecuencia del estándar IEEE 802.15.4

Banda	Canales	Tasa de transferencia	Espaciamiento entre canales	Modulación
868 - 868.6 MHz	1	20 kbps	-	BPSK (<i>Binary phase shift keying</i>)
902 - 928 MHz	10	40 kbps	2 MHz	BPSK
2.4 - 2.4835	16	250 kbps	5 MHz	O-QPSK

Las especificaciones de sensibilidad de IEEE 802.15.4 define -85 dBm para la frecuencia de 2.4 GHz y de -92 dBm para las frecuencias de 868 y 915 MHz. Dichos valores encierran un margen suficiente para embarcar tolerancias regulares que se requieren debido a la no perfección de su fabricación.

El estándar IEEE 802.15.4 especifica que cada dispositivo debe ser capaz de transmitir al menos a 1mW, pero dependiendo de las necesidades de la aplicación, la potencia de transmisión puede ser mayor o menor pero dentro de los límites de regulación establecidos. Por lo que se espera que dispositivos típicos de 1mW de potencia cubran un rango de entre 10 y 75 m; sin embargo, con una buena sensibilidad y un incremento moderado en la potencia de transmisión, se logra alcanzar coberturas mayores.

La Subcapa MAC, o de acceso al medio del protocolo IEEE 802.15.4 es la responsable de asegurar la comunicación entre un nodo y todos los nodos conectados directamente a él, evitando colisiones y mejorando la eficiencia. Provee una interface entre la capa física y la próxima capa sobre la de MAC que en el caso de ZigBee es la de red, dado que el protocolo IEEE 802.15.4 se compone de las especificaciones para PHY y MAC.

En el caso de ser coordinador de red genera Guías (*Beacons*) con el objetivo de que el resto de nodos se sincronicen al ritmo de éstos. Evita interferencias en las comunicaciones ya que cada uno de los dispositivos de la red escucha antes de transmitir es decir se realiza un sondeo del canal, mediante el mecanismo de acceso al medio CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*).

Las tareas que la capa MAC tiene que realizar son:

- Validación de tramas.
- Asociación y Disociación.
- Funciones de seguridad (Encriptación AES de 128 bits).
- QoS mediante la gestión de GTS (*Granted Time Slot*).
- Entrega de tramas de configuración *Acknowledgment Frame* (ACK's).

Las tramas MAC en general tienen un formato de diseño flexible, el cual se ajusta a las necesidades de las diferentes aplicaciones con diversas topologías de red y al mismo tiempo sigue manteniendo un protocolo simple.

Existen cuatro tipos de tramas basadas en el estándar IEEE 802.15.4. Estas son:

- Tramas de datos (Data Frame),
- Tramas de confirmación (Acknowledgment Frame)
- Tramas de comandos MAC (MAC Command Frame)
- Tramas de Beacon “Guía” (Beacon Frame),

Solo las tramas de datos y tramas beacon contienen información proveniente de capas superiores; las tramas de mensajes de confirmación y la de comandos MAC son originadas en la MAC y son usadas por la MAC para comunicaciones igual a igual [23].

2.2.1 CARACTERÍSTICAS DE ZIGBEE.

Las características más importantes del estándar son:

- Diversas bandas de trabajo: 2,4 GHz (16 canales), 915 MHz (10 canales) y 868 MHz (1 canal).
- Direccionamiento a nivel red de 16 bits.
- Soporte para el encaminamiento de paquetes .
- Gracias a la posibilidad de encaminamiento se permiten las topologías de red mallada.
- Dos tipos de dispositivos. FFD (coordinador, encaminador, dispositivo final) y RFD (dispositivo final).
- Métodos de acceso al canal: CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*).
- Soporta redes *slotted* (QoS) y *non-slotted* .
- Bajo consumo energético.
- Gran densidad de nodos por red. ZigBee permite que las redes manejen hasta 216 dispositivos. Este atributo es fundamental para la creación de series masivas de sensores y redes de mando.

- Radio de cobertura hasta 500 m según el entorno.

2.2.2 CARACTERÍSTICAS DE IEEE 802.15.4

Entre las características más importantes se pueden mencionar:

- Puede trabajar tanto en las bandas de 2.4GHz como en la de 868/915MHz.
- Tasa de transmisión de hasta 250 kbps en 2.4 GHz, 40 kbps en 915MHz y 20 kbps en 868 MHz.
- Optimizado para aplicaciones con ciclo efectivo menor a 0.1 %.
- Usa CSMA-CA (*Carrier Sense Multiple Access Collision Avoidance*) para acceso al canal.
- Produce alto rendimiento y baja latencia para dispositivos de bajo ciclo de trabajo, muy adecuado esto para sensores y controles.
- Baja potencia. Ideal para equipos a batería.
- 64 bits de direccionamiento determina una cantidad máxima de $1.8 \cdot 10^{19}$ dispositivos.
- 16 bits para identificar redes que determina un total de 65536 redes.
- Permite el uso de ranuras de tiempo (*time slots*) para posibilitar aplicaciones de baja latencia.
- Protocolo con diálogo (*handshake*) para mejorar la seguridad en las transferencias.
- Rango: hasta 75 m (valor típico, depende del ambiente).

2.2.3 TOPOLOGÍAS DE RED

El estándar ZigBee soporta 3 tipos de topologías de red: estrella, malla y árbol, como se ilustra en la Figura 2.8 y la capa responsable de construirlas es la capa de Red NWK, ésta también se encarga de los servicios de enlace que integra las tareas necesarias de direccionamiento y seguridad.

Por otro lado IEEE 802.15.4 soporta dos tipos de topologías para su conexión en red: estrella y *peer to peer*. La última es similar a arquitecturas de red malla (*mesh*) o Ad-Hoc, por lo tanto esta no se detalla en la siguiente Figura.

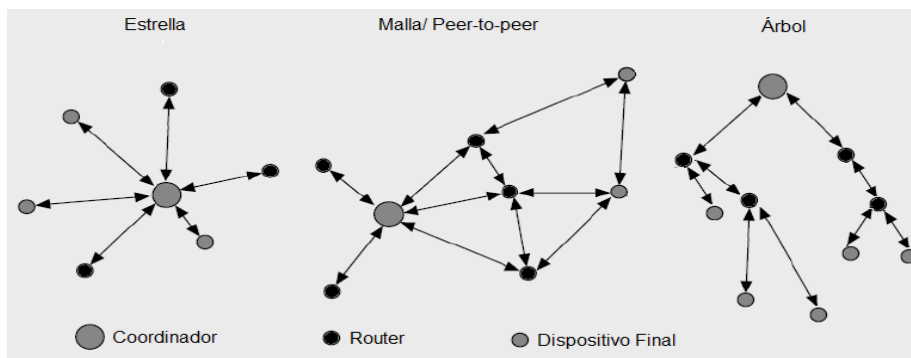


Figura. 2.8. Topologías de la red ZigBee/IEEE 802.15.4

2.2.3.1 Topología en estrella

Este tipo de redes basadas en topología estrella se componen por un nodo sensor funcionando como coordinador y varios sensores funcionando como dispositivos finales. Todos los dispositivos finales están directamente conectados al coordinador, es decir cualquier conexión que se quiera realizar entre los distintos nodos de la red debe pasar por éste nodo coordinador, que es el encargado de haber iniciado la red y de gestionarla. Un problema de esta configuración es que la escalabilidad de la red con esta topología está muy restringida ya que el tamaño de la red depende del rango de alcance del coordinador. Por otro lado el funcionamiento del nodo coordinador es esencial, dado que un fallo de este nodo central tiene como consecuencia la desconexión de toda la red.

2.2.3.2 Topología en malla

Este tipo de topología permite en la red que al menos un par de nodos pueda establecer comunicación directa sin permiso del coordinador. El nodo coordinador no realiza funciones relevantes comparadas a cualquier nodo (*Routers*) de la red, además, esta topología no requiere de un servidor o nodo central, con lo que se reduce el mantenimiento.

Con la topología malla la red puede ganar flexibilidad pero aumentar la complejidad. Esto se debe a que por cada comunicación de cualquier par de dispositivos hay más de un camino posible o caminos redundantes de modo que si falla un cable, otro se hará cargo del tráfico, con esto se logra incrementar la fiabilidad de la red. Para la elección de dicho camino la computación aumenta y debe realizarse a nivel de red, estas consideraciones no corresponde a la especificación del IEEE 802.15.4.

2.2.3.3 Topología en árbol

Un caso particular de la topología en malla es la topología en árbol. En ella los diferentes componentes de la red se organizan siguiendo una estructura jerárquica. El coordinador, se encargan de gestionar el encaminamiento de la información dentro de la red, pueden tener dispositivos de menos jerarquía o hijos. Estos dispositivos hijos pueden ser otros *routers* o dispositivos finales. La topología en árbol, es eficaz para expandir la red de forma dinámica.

Tabla. 2.2.

Comparación de topologías de WSN

Topología	Escalabilidad	Tráfico
Árbol	<ul style="list-style-type: none"> ▪ Por su topología permite que la red se expanda de forma dinámica. 	<ul style="list-style-type: none"> ▪ Los nodos del árbol están conectados a un concentrador central que controla el tráfico de la red. ▪ Los caminos de circulación de tráfico en la red no son redundantes.
Estrella	<ul style="list-style-type: none"> ▪ Tiene una escalabilidad muy restringida ya que el tamaño de la red depende del rango de alcance del coordinador. 	<ul style="list-style-type: none"> ▪ Fácil de controlar su tráfico, el cual es muy sencillo. ▪ Los caminos de circulación de tráfico en la red no son redundantes.
Malla	<ul style="list-style-type: none"> ▪ En este tipo de topología la escalabilidad es mucho mayor ya que no es necesario de un servidor o nodo central, además se logra reducir problemas de embotellamiento y en general de mantenimiento. 	<ul style="list-style-type: none"> ▪ En caso de existir una avería en el camino éste orienta el tráfico por otros alternativos. ▪ Los caminos de circulación de tráfico en la red son redundantes.

En la Tabla 2.2 se muestra una comparación de las topologías que junto a la tecnología ZigBee y estándar IEEE 802.15.4 se pueden aplicar para el desarrollo de las simulaciones, uno de los conceptos más importantes es la capacidad para aumentar el rendimiento del sistema en forma gradual a medida que crece la carga, es decir asegurar escalabilidad en la red.

Una vez realizada la comparación entre las tres posibles topologías a elección, se toma en cuenta que para la aplicación en el área de Monitorización Volcánica se necesita una red la cual brinde confiabilidad, escalabilidad, redundancia de caminos, fácil mantenimiento, ahorro de energía, etc. Por lo que se utilizará la topología estrella ya que ésta cumple con la mayoría de exigencias, además, ayudará hallar resultados con menor margen de error en el desempeño de la red.

2.2.4 TIPO DE DISPOSITIVOS ZIGBEE

Los diferentes tipos de nodos o dispositivos dentro de las publicaciones del estándar IEEE 802.15.4 y ZigBee desempeñan diferentes funciones dentro de la red inalámbrica. Sin embargo, cada dispositivo tiene su correspondiente en el otro estándar y lo único que varía es el nombre asignado.

En ZigBee existen tres tipos de dispositivos:

- Coordinador ZigBee (CZ)

Este dispositivo requiere memoria y capacidad de computación ya que tiene como tareas iniciar la formación de la red inalámbrica, escoge un canal libre y lo administra, selecciona un identificador de red y seguido a eso puede actuar como *router*. También actúa como centro de validación (*trust center*) que es el que permite que un nodo permanezca en la red y pueda intercambiar datos mediante una clave, o caso contrario se lo fuerce a salir de la red.

El coordinador ZigBee debe ser único en la red y es el encargado de controlar el proceso de incorporación de nuevos dispositivos, tomando en cuenta los parámetros

iniciales de configuración de la red y manejando las peticiones de los nuevos dispositivos, incluso puede eliminarlos o desasociarlos previa petición de los mismos, además, conserva una lista de los dispositivos incorporados y brinda soporte a dispositivos libres para que puedan asociarse o reincorporarse a la red.

– *Router ZigBee (RZ)*

Este dispositivo es el encargado de permitir que otros dispositivos se incorporen a dicha red una vez que el *router* ha sido validado y asociado a la red, mediante los parámetros de configuración que posee. Sin embargo, el dispositivo RZ es opcional ya que puede actuar como coordinador y cumplir con la función de enrutamiento de paquetes/mensajes de saltos múltiples, mediante una lista de los nodos asociados. De igual manera, también puede eliminar o quitar nodos de la porción de red que está bajo su cargo.

– Dispositivo Final – *End Device ZigBee (ZED)*

Son los dispositivos restantes que participan en la red pero que no son ni el coordinador ni los *routers*, por lo tanto no pueden enrutar paquetes, pero posee la funcionalidad necesaria para comunicarse con su nodo padre (el coordinador o un *router*), por lo tanto, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías. Un ZED tiene requerimientos mínimos de memoria y es por tanto significativamente más barato.

Los nodos definidos por las Recomendaciones IEEE 802.15.4 dentro de una red inalámbrica son:

– Dispositivos con Funciones Completas (*FFD – Full Function Device*)

Conocido como nodo activo, éste funciona como coordinador, ruteador o terminal, ya que posee memoria extra y capacidad computacional, permitiéndole cumplir con la tarea de enrutamiento de paquetes. Puede comunicarse con otros FFD o con RFD, posibilitando así la configurar cualquier topología existente de red.

- Dispositivos con Funciones Reducidas (*RFD - Reduced Function Device*)

Es un dispositivo conocido como nodo pasivo y sirve para realizar aplicaciones sencillas que no necesitan enviar o recibir grandes cantidades de datos. Su instalación es clara y fácil, bajo costo y gracias a su poca complejidad su consumo energético es reducido. Sin embargo, solo pueden comunicarse con el dispositivo FFD que actúa como coordinador de red, que será el encargado de gestionar sus peticiones. Esto limita a los RFD a estar únicamente dentro de una topología de red en estrella.

2.3 MÉTRICAS DE QoS EN WSN

El desempeño de una red, puede referirse al nivel de calidad de servicio (QoS) de un producto de telecomunicaciones desde el punto de vista del cliente. Al igual que cualquier otro sistema basado en computador, las redes deben tener un alto desempeño, por lo que la calidad de la red se basa entonces en dos aspectos principales:

- Algunas aplicaciones en WSN tienen propiedades en tiempo real. Estas aplicaciones pueden tener requisitos de calidad de servicio tales como, *throughput*, *delay*, pérdida de paquetes y *Jitter* que están dirigidas a la determinación del consumo de energía, obtención de un rango óptimo de número de nodos sensores, donde el *Throughput* se mantendrá en su nivel más alto y con esto se evitará en gran parte la pérdida de paquetes, además se evitará que los datos recibidos en uno de los sensores (*Gateway*) sean redundantes con respecto a otros dispositivos de su entorno. Muchas de estas métricas de evaluación de QoS están interrelacionadas y puede ser necesario disminuir el rendimiento en una de ellas para incrementar el de la otra.
- Los protocolos deben disponer de esquemas sencillos y mínimos de QoS, de modo que garanticen un retardo acorde con las especificaciones de la aplicación y una tasa de entrega aceptable, no poniendo en peligro otros aspectos. El desarrollar protocolos eficientemente energéticos no debe ser a costa de obtener deficiencias en los atributos principales de la red.

2.3.1 THROUGHPUT

El rendimiento o *throughput* se define como la medida de la tasa de transferencia de datos a través de un sistema de comunicaciones en cierto período, las unidades del *throughput* son bits/s o bytes/s.

En un entorno WSN con, la eficiencia global del sistema en términos de *throughput* (η) dependerá de la relación entre los intervalos destinados a la transmisión de datos (slot de datos) y los destinados a tráfico de control (principalmente, períodos de contención y paquete de *downlink*). Por lo tanto, cuanto mayor sean los paquetes transmitidos en el slot de datos y menor sean los intervalos destinados a tráfico de control, mayor será η .

Para determinar la cantidad de información que se entrega en un intervalo de tiempo (t_{tx}) en unidades de bits, la cantidad total de bytes transmitidos (BT_{tx}) es multiplicado por su equivalente en bits que es 8, se calcula con la siguiente fórmula.

$$\eta = \frac{8 \times BT_{tx}}{t_{tx}} \left[\frac{\text{Bits}}{\text{Seg}} \right] \quad (2.1)$$

Asimismo, el *Throughput* normalizado (η') como medida del desempeño de la red en función de la tasa máxima de transmisión (*Raw Bit Rate* - RBR) de 250 kbps, es hallado con la siguiente expresión.

$$\eta' = \frac{\eta}{RBR} \quad (2.2)$$

2.3.2 RETARDO (*END-TO-END DELAY*)

Es una métrica que da información sobre el tiempo promedio que se demora un paquete durante su transmisión que se envía a través de la red desde el emisor al receptor y viceversa. Una vez terminada la simulación, el retardo se calcula como la sumatoria de todos los retardos de cada paquete sobre el número total de paquetes entregados (PT_{Rx}) [25].

$$Retardo = \frac{\sum_{i=0}^k R_i}{P_{Tx}}, \quad (2.3)$$

donde R_i es el retardo que percibe el paquete i para llegar a su destino (coordinador).

2.3.3 JITTER

La variación en el tiempo de retardo de los paquetes que llegan a su destino. Los paquetes se envían a intervalos regulares desde el emisor al receptor, pero debido a la latencia de la red el intervalo entre los paquetes pueden variar en el destino. Esta variación puede afectar a la calidad de los medios de comunicación [26], este tipo de retardo se calcula como la diferencia de todos los retardos de entre cada paquete entregado o recibido. El *jitter* suele considerarse como una señal de ruido no deseada. En general se denomina *jitter* a un cambio indeseado y abrupto de la propiedad de una señal.

2.3.4 PAQUETES PERDIDOS

Es la relación existente entre los paquetes perdidos y el número total de paquetes enviados. La pérdida de paquetes influye en el rendimiento de la transmisión. Esta métrica da cuenta de qué tan probable es que un paquete se pierda en la red y se calcula como:

$$P_{pp} = \frac{P_{tx} - P_{Rx}}{P_{tx}} \quad (2.4)$$

donde P_{pp} es la probabilidad de pérdida de paquetes, P_{tx} es la cantidad de paquetes transmitidos, y P_{Rx} es la cantidad de paquetes recibidos [25].

2.4 OPTIMIZACIÓN DE FUNCIONES

La optimización es uno de los objetivos de la Matemática, dirigida a encontrar métodos que permitan resolver problemas donde está presente algún criterio de optimalidad. Existe una amplia variedad de métodos para aplicar según las características del problema objeto de análisis. Como es lógico dichos métodos están inmersos en la

asignatura de Calculo Diferencial para funciones reales de una y varias variables reales. No obstante, es importante mencionar que la búsqueda de tales métodos continúa en nuestros días. Estos objetivos de la Matemática responden a necesidades reales que existen en nuestro país o en cualquier otro para su desarrollo.

La optimización es una herramienta eficaz de modelado, también denominada programación matemática, sirve para encontrar la respuesta que proporciona el mejor resultado, ganancia, producción o el menor costo, pérdida o desperdicio, es decir se trata de encontrar el mínimo o el máximo de una cierta función con un número determinado de variables, definida en cierto dominio ya que implican utilizar de la manera más eficiente los recursos.

Un método de optimización simple para una función de una sola variable podría ser fijar un rango de valores de x y calcular el valor de $f(x)$ para cada valor de x , el resultado óptimo sería que la expresión matemática o función tenga el mayor valor numérico posible (maximización) o el menor valor numérico posible (minimización). Si utilizamos este procedimiento como base de la optimización a continuación se detallan definiciones de crucial importancia dentro del proceso general de optimización de funciones.

Definición [Punto Crítico]. Sea $A \subseteq \mathbb{R}^n$ un conjunto abierto y $f: A \rightarrow \mathbb{R}$ una función con derivadas segundas continuas en A . El punto $P_o = (x_1^0, \dots, x_n^0)$ es un punto crítico de f si:

$$\begin{cases} \frac{\partial f(x_1^0, \dots, x_n^0)}{\partial x_1} = 0 \\ \dots \\ \frac{\partial f(x_1^0, \dots, x_n^0)}{\partial x_n} = 0 \end{cases} \quad (2.5)$$

Todas las derivadas parciales de primer orden de f se anulan en P_o [27].

Se definen puntos críticos como los puntos en los que el gradiente de la función objetivo es igual a cero.

Definición [Máximos y Mínimos]. Sea $A \subseteq \mathbb{R}^n$ un conjunto abierto y $f: A \rightarrow \mathbb{R}$ una función con derivadas de segundo orden continuas en A ; se dice que el punto $P_0 = (x_1^0, \dots, x_n^0)$ es, para la función f :

- **Máximo absoluto** si, para cada otro punto $P = (x_1, \dots, x_n) \in A$:

$$f(x_1^0, \dots, x_n^0) \geq f(x_1, \dots, x_n) \quad (2.6)$$

- **Mínimo absoluto** si, para cada otro punto $P = (x_1, \dots, x_n) \in A$:

$$f(x_1^0, \dots, x_n^0) \leq f(x_1, \dots, x_n) \quad (2.7)$$

- **Máximo relativo** si existe un entorno B de P_0 tal que, para cada otro punto

$$P = (x_1, \dots, x_n) \in B:$$

$$f(x_1^0, \dots, x_n^0) \geq f(x_1, \dots, x_n) \quad (2.8)$$

- **Mínimo relativo** si existe un entorno B de P_0 tal que, para cada otro punto

$$P = (x_1, \dots, x_n) \in B:$$

$$f(x_1^0, \dots, x_n^0) \leq f(x_1, \dots, x_n) \quad (2.9)$$

- **Silla** si es siempre posible encontrar dos puntos $P_1 = (x_1^1, \dots, x_n^1)$ y $P_2 = (x_1^2, \dots, x_n^2)$ en un entorno B de P_0 tal que:

$$f(x_1^1, \dots, x_n^1) \leq f(x_1^0, \dots, x_n^0) \leq f(x_1^2, \dots, x_n^2) \quad (2.10)$$

Se puede demostrar que los máximos y mínimos de una función son puntos críticos si se alcanzan en puntos interiores (también pueden ser máximos y mínimos puntos en la frontera, pero entonces no son necesariamente críticos).

Definición [Matriz Hessiana¹]. Sea $A \subseteq \mathbb{R}^n$ un conjunto abierto y $f: A \rightarrow \mathbb{R}$ una función con derivadas segundas continuas en A .

Se define la matriz hessiana de f , en cada punto, como la siguiente matriz $n \times n$ [27]:

$$H(x_1, \dots, x_n) = \begin{pmatrix} \frac{\partial^2 f(x_1, \dots, x_n)}{\partial x_1^2} & \dots & \frac{\partial^2 f(x_1, \dots, x_n)}{\partial x_n \partial x_1} \\ \dots & \dots & \dots \\ \frac{\partial^2 f(x_1, \dots, x_n)}{\partial x_1 \partial x_n} & \dots & \frac{\partial^2 f(x_1, \dots, x_n)}{\partial x_n^2} \end{pmatrix} \quad (2.11)$$

¹ Matriz cuadrada de $n \times n$, de las segundas derivadas parciales.

Una manera de decidir si los puntos críticos son máximos, mínimos o puntos silla para una función está basada en el estudio de las derivadas segundas y en particular de la matriz hessiana.

En la Figura 2.9 se ilustra los puntos máximos y mínimos que se pueden encontrar mientras se evalúa la función f en todo su dominio.

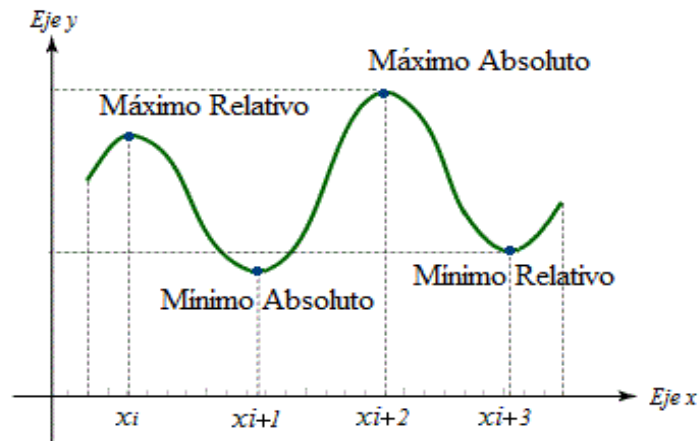


Figura. 2.9. Máximos y Mínimos de una función

Los problemas de optimización generalmente se clasifican en lineales y no lineales, según si las relaciones del problema sean lineales o no con respecto a sus variables. Existen muchos algoritmos matemáticos útiles para la optimización e incluso series de paquetes de *software* para resolver problemas donde se busca encontrar una respuesta óptima.

2.4.1 OPTIMIZACIÓN LINEAL

Este tipo de optimización es un caso especial de la optimización denominado también modelo de programación lineal, pues debe cumplir que, tanto la función objetivo F como todas las funciones de restricción, sean lineales ya que es una de las técnicas aplicable a problemas de asignación de recursos limitados, con actividades competitivas hacia un objetivo común, que puede ser de maximizar beneficios o minimizar pérdidas. Se utiliza un modelo matemático con representación válida de la problemática en estudio y como ya se dijo sus relaciones deben ser lineales, que significa utilizar, sólo variables de primer grado en cada término.

La Optimización Lineal (OL) se puede describir de la siguiente forma; Dada una función lineal de varias variables definida para todos los números reales, se requiere determinar valores reales para dichas variables que maximicen o minimicen el valor de la función lineal, sujetas o no a ciertas condiciones de la forma de un sistema de ecuaciones y/o inecuaciones lineales.

El objetivo de la OL es encontrar un valor denominado solución óptima o simplemente una solución al problema, tal como:

$$f(x) = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \quad (2.12)$$

Donde $c_1, c_2, c_3 \dots c_n$ son coeficientes de la función objetivo, $x_1, x_2, x_3 \dots x_n$ son las variables de decisión lineales y $f(x)$ es la función objetivo.

Para maximizar o minimizar una función como la mostrada en (2.12) se debe hallar el valor numérico más alto o bajo que alcance la función lineal según corresponda a lo requerido. También se debe tomar en cuenta los llamados ceros o raíces que no son más que los valores de x que tienen por imagen el cero, es decir cuando tocan al eje x , de igual manera al dominio que son los valores que puede tomar x .

2.4.2 OPTIMIZACIÓN NO LINEAL

Se define como optimización no lineal (ONL) al conjunto de métodos utilizados para optimizar una función objetivo, sujetas éstas en muchos casos a una serie de restricciones en los que una o más de las variables incluidas es no lineal.

El problema de programación no lineal es encontrar un punto factible $P = (x_1^0, \dots, x_n^0)$ tal que $f(x_1^0, \dots, x_n^0) \geq f(x_1, \dots, x_n)$ para cada punto de x . Si existe más de un punto óptimo, estos son referidos como soluciones alternativas óptimas o como se definió en el apartado anterior solución relativa (máximo o mínimo relativo), además todas estas soluciones forman una región denominada factible.

La forma o estructura de la conocida ecuación algebraica de una función no lineal puede describirse de la siguiente manera:

$$f(x) = c_1 x_1^{m_1} + c_2 x_2^{m_2} + \dots + c_n x_n^{m_n} \quad (2.13)$$

Como se observa la ecuación (2.13), ésta consta de los mismos parámetros que (2.12), con la excepción de m_1, m_2, \dots, m_n , lo cual representa la potencia de las variables x correspondientes a la función objetivo.

Un problema de optimización lineal y no lineal en la mayoría de ocasiones puede expresarse como un problema de maximización o minimización que esté sujeto a una o varias restricciones, se pueden escribir de la siguiente forma:

$f(x)$, sujeta a restricciones

- De desigualdad $g_i(x) > 0$, $g_i(x) < 0$, $g_i(x) \geq 0$, $g_i(x) \leq 0$, para $i = 1, 2, \dots, n$.
- De igualdad, $h_j(x) = 0$, $h_j(x) \geq 0$, $h_j(x) \leq 0$, para $j = 1, 2, \dots, m$

Donde $f(x)$, $g_i(x)$ y $h_j(x)$ son funciones definidas en el espacio Euclidiano² de n dimensiones [28].

Tabla. 2.3.

Comparación de optimización Lineal y no lineal

Optimización Lineal	Optimización No Lineal
La solución óptima se encuentra en un punto extremo de la región de factibilidad.	No siempre la solución óptima se encuentra en un punto extremo de la región de factibilidad.
El punto óptimo nunca está dentro de la región de factibilidad.	Hay casos donde el punto óptimo está en el interior de la región factible.
Sus métodos de optimización generan óptimos absolutos ó globales.	Generalmente se encuentra un óptimo local ó relativo, más no el óptimo global ó absoluto.
La región de factibilidad es un conjunto convexo.	Se pueden generar regiones de factibilidad que no son necesariamente convexas.
Sus funciones objetivo y restricciones son estrictamente lineales.	La función objetivo, las restricciones ó ambas pueden ser no lineales y/o lineales.
Los métodos de optimización son más simples.	Los métodos de optimización son más complejos.

² Espacio Cartesiano, bidimensional o tridimensional en el que se cumplen los axiomas de Euclides.

La optimización de funciones no lineales son mucho más difíciles de resolver que los de funciones lineal. A continuación se muestra en la Tabla 2.3 la comparación de las características más importantes de los problemas de Optimización lineales y no lineales.

2.5 TÉCNICAS DE OPTIMIZACIÓN

2.5.1 OPTIMIZACIÓN ESCALAR

La optimización escalar consiste en encontrar el mínimo o el máximo de una función de una sola variable. Por lo tanto una optimización escalar es aquella técnica en la cual la función está o no ligada a restricciones, se muestra aspectos elementales:

- En muchos problemas las restricciones se pueden incluir dentro de la función objetivo, por lo que la dimensionalidad del problema se reduce a una variable.
- Permiten introducir muchos conceptos y explorar ideas que se usarán en problemas ONL.
- La mayoría de los problemas sin restricciones, inherentemente incluyen una única variable.
- Las técnicas de optimización con y sin restricciones, generalmente incluyen pasos de búsqueda unidireccional en sus algoritmos.

La forma general de un problema de optimización escalar se puede escribir de la siguiente manera,

$$\max. f(x), \text{ s. a. } x_1 \leq x \leq x_2 \quad (2.14)$$

En la Figura 2.10 se presenta un ejemplo claro de la función $f(x)$ que depende exclusivamente de una única variable x , en el recorrido de la función sobre la variable encuentra un punto escalar mínimo absoluto que satisface a la restricción $x_1 \leq x \leq x_2$.

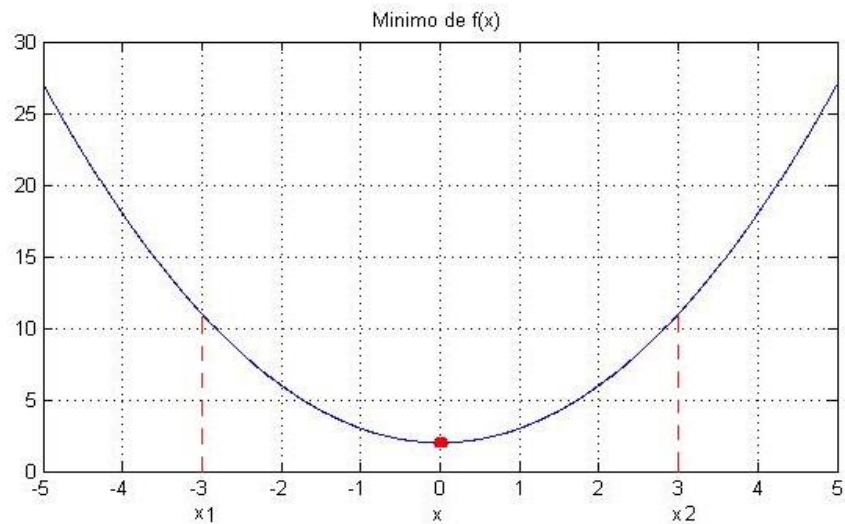


Figura. 2.10. Optimización escalar

En [29] se presenta un análisis de los métodos de optimización escalar y su respectiva clasificación que engloba diversos métodos, sin embargo de los métodos Indirectos, métodos de Eliminación de Región y métodos de Aproximación Polinómica que se describen ahí tomaremos como método de análisis a los métodos indirectos ya que presentan un alto índice aplicativo en diversas áreas.

2.5.1.1 Método Indirecto

Los métodos indirectos son una buena técnica de optimización de funciones, tienen la ventaja esencial de que la convergencia es generalmente más rápida incluso para calcular las derivadas. No obstante, en algunos problemas de optimización esta ventaja es muchas veces contrarrestada, ya que por la falta de interés de determinación precisa de la función objetivo en ocasiones existe falta de precisión de los coeficientes que muchas veces son utilizados.

En la mayoría de procedimientos de optimización unidimensional es necesario que el punto óptimo a encontrar esté acotado dentro de un intervalo conocido como primer punto de la estrategia de búsqueda. Existen varias estrategias que se pueden emplear para acotar el óptimo, la más sencilla consiste en fijar un punto y comenzar a movernos una distancia fija en una dirección.

Los métodos indirectos utilizan las condiciones necesarias, las derivadas (analíticas o numéricas) y la función objetivo, básicamente existen tres métodos para llevar a cabo la búsqueda indirecta unidireccional, basados en las condiciones de optimalidad, estos son:

- Método de Newton
- Aproximaciones finitas al método de Newton (Métodos cuasi-Newton)
- Métodos de secante.

2.5.1.1.1 Método de Newton

El método de Newton es uno de los menos complicados de manejar, este método no trabaja sobre un intervalo sino que basa su fórmula en un proceso iterativo, la primera condición que se tiene que cumplir para que una función tuviera un mínimo local es que la primera derivada escrita de la forma $f'(x)$ se igual a cero, una vez cumplida esta condición se puede aplicar el método de Newton a la derivada de la siguiente forma:

$$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)} \quad (2.15)$$

En cualquier etapa ó iteración k , se tiene que cumplir la siguiente condición $f(x^{k+1}) < f(x^k)$, para minimización de la función. En general los algoritmos de interpolación que actúan inmersos en el método Newton realizan en cada iteración una aproximación de la función $f(x)$, en el punto x^k considerado en dicha iteración, por un polinomio de segundo o de tercer grado que bien puede ser parábola $q(x^{k+1})$ y toma x^{k+1} como el vértice de dicha parábola. Para hacerlo necesita evaluar las primeras derivadas de la función en x^k .

$$q(x^{k+1}) = f(x^k) + f'(x^k)(x - x^k) + \frac{1}{2}f''(x^k)(x - x^k)^2 \quad (2.16)$$

y dado que $q'(x^{k+1}) = 0$ y $f'(x) = 0$, diferenciando la ecuación anterior tenemos,

$$f'(x^k) + 2\left(\frac{1}{2}\right) f''(x^k)(x - x^k) = 0 \quad (2.17)$$

El método de Newton es equivalente a usar un modelo cuadrático y aplicar las condiciones necesarias de optimalidad.

Las ventajas del método de Newton son:

- Para una función cuadrática el mínimo se obtiene en una única iteración.
- El procedimiento es cuadráticamente convergente ($p=2$), siempre que $f''(x) \neq 0$

Las desventajas son:

- Se debe calcular tanto $f'(x)$ como $f''(x)$.
- Si $f''(x) \rightarrow 0$ el método converge muy lentamente lo que le convierte en un método no eficiente.
- Si existe más de un extremo, el método podría no converger al extremo deseado, es decir la convergencia en este método no está asegurada.

2.5.1.1.2 Método de Newton en diferencias finitas

Este método básicamente consiste en que las derivadas son reemplazadas por aproximaciones en diferencias finitas, convirtiendo entonces un problema de ecuaciones diferenciales en un problema algebraico fácilmente resoluble por medios comunes que puede llevar un número largo de operaciones matemáticas.

En el caso particular de que las abscisas de los nodos de interpolación sean equidistantes la expresión del polinomio de interpolación de Newton en diferencias divididas adopta otras formas tales como la fórmula en diferencias progresivas y la fórmula en diferencias regresivas.

Dado un conjunto de puntos (x_i, y_i) , $0 \leq i \leq n$ donde $y_i = f(x_i)$ se define,

Diferencia progresiva de orden 1 en y_k y se denota por Δy_k a,

$$\Delta y_k = f(x_k + h) - f(x_k) = f(x_{k+1}) - f(x_k) = y_{k+1} - y_k = \Delta^1 y_k \quad (2.18)$$

Siendo h la diferencia constante entre dos abscisas consecutivas.

Diferencia regresiva de orden 1 en y_k y se denota por ∇y_k a

$$\nabla y_k = y_k - y_{k-1} = \nabla^1 y_k \quad (2.19)$$

Análogamente, se define diferencia progresiva de orden 2 en y_k a

$$\begin{aligned} \Delta^2 y_k &= \Delta(\Delta y_k) = \Delta(y_{k+1} - y_k) = \Delta y_{k+1} - \Delta y_k = y_{k+2} - y_{k+1} - (y_{k+1} - y_k) \\ &= y_{k+2} - 2y_{k+1} + y_k = \nabla^2 y_{k+2} \end{aligned} \quad (2.20)$$

En general se definen

$$\Delta^m y_k \stackrel{(\text{def})}{=} \Delta(\Delta^{m-1} y_k) \quad (2.21)$$

$$\nabla^m y_k \stackrel{(\text{def})}{=} \nabla(\nabla^{m-1} y_k) \quad (2.22)$$

y convenimos en que $\Delta^0 y_k = y_k$, $\nabla^0 y_k = y_k$ y ocurre además que $\Delta^m y_k = \nabla^m y_{k+m}$ y también,

$$\Delta^n y_k = \sum_{j=0}^n (-1)^j \binom{n}{j} y_{k+m-j} \quad (2.23)$$

Para poder reescribir la fórmula de Newton en términos de diferencias progresivas o regresivas es preciso establecer una relación entre diferencias divididas y diferencias finitas. Dicha relación nos viene dada por el siguiente lema [30],

Lema 2.1. $\forall i \geq 0$: $f[x_k, x_{k+1}, \dots, x_{k+i}] = \frac{\Delta^i y_k}{i! h^i}$

$$f[x_k, \dots, x_{k+n+1}] = \frac{f[x_k, \dots, x_{k+n}] - f[x_{k+1}, \dots, x_{k+n+1}]}{x_k - x_{k+n+1}} =$$

$$\frac{\Delta^n y_k / n! h^n - \Delta^n y_{k+1} / n! h^n}{-(n+1)h} = \frac{\Delta^n y_{k+1} - \Delta^n y_k}{(n+1)! h^{n+1}}$$

$$\frac{\Delta^n (y_{k+1} - y_k)}{(n+1)! h^{n+1}} = \frac{\Delta^n (y_k)}{(n+1)! h^{n+1}} = \frac{\Delta^{n+1} y_k}{(n+1)! h^{n+1}} \quad (2.24)$$

- Se demuestra para $i = 0 : f[x_k] = f(x_k) = y_k = \Delta^0 y_k$
- Se supone cierto para $i = n > 0$. Y entonces
- Se demuestra para $i = n + 1$. Tenemos

Utilizando éste lema podemos entonces obtener la fórmula de Newton en diferencias progresivas, que es la misma que en diferencias divididas pero expresada en diferencias finitas, que es posible si en los puntos de interpolación las abscisas son equidistantes o sea, si $x_{i+1} - x_i = h, \forall i = 0, 1, \dots, n - 1$.

Tenemos entonces los puntos de interpolación $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, donde $y_i = f(x_i)$ y además $x_{i+1} = x_i + h, \forall i = 0, 1, \dots, n - 1$ y podemos escribir $x_1 = x_0 + h \Rightarrow x_2 = x_1 + h = x_0 + 2h \Rightarrow x_3 = x_0 + 3h \Rightarrow \dots x_i = x_0 + ih, i = 0, 1, 2, \dots, n$.

La expresión del polinomio de interpolación en diferencias divididas es,

$$p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_{n-1}) \quad (2.25)$$

Aplicando el lema anterior a la fórmula 2.25 se obtiene:

$$p_n(x) = y_0 + \frac{\Delta^1 y_0}{1! h^1} (x - x_0) + \frac{\Delta^2 y_0}{2! h^2} (x - x_0)(x - x_1) + \dots + \frac{\Delta^n y_0}{n! h^n} (x - x_0)(x - x_1) \dots (x - x_{n-1}) \quad (2.26)$$

por otra parte, hacemos el siguiente cambio

$$\frac{(x - x_0)}{h} = s \Rightarrow \frac{(x - x_i)}{h} = \frac{x - (x_0 + ih)}{h} = \frac{x - x_0 - ih}{h} = \frac{(x - x_0)}{h} - \frac{ih}{h} = s - i \quad (2.27)$$

Reemplazando 2.27 en 2.26 tenemos la fórmula que se conoce con el nombre de Newton en diferencias finitas progresivas.

$$p_n(x) = \sum_{k=0}^n \binom{s}{k} \Delta^k y_0 \quad (2.28)$$

Mientras que siguiendo el mismo procedimiento encontramos la fórmula de Newton en diferencias regresivas, para esto se considera los puntos de interpolación en el orden $(x_n, y_n), (x_{n-1}, y_{n-1}), \dots, (x_0, y_0)$ y para este orden se tiene una expresión del polinomio de interpolación en diferencias divididas similar a 2.25.

Sabiendo el siguiente cambio:

$$\frac{(x - x_n)}{h} = u + n - i \quad (2.29)$$

Encontramos la formula de Newton en diferencias regresivas al reemplazar 2.29 en la expresión del polinomio de interpolación en diferencias divididas para orden $(x_n, y_n), (x_{n-1}, y_{n-1}), \dots, (x_0, y_0)$.

$$p_n(x) = \sum_{k=0}^n \binom{u + k - 1}{k} \nabla^k y_0 \quad (2.30)$$

2.5.1.1.3 Métodos de secante

En este método en principio se requiere de dos puntos iniciales x_0 y x_1 , los cuales pueden ser arbitrarios. Seguido realiza un trazado de las rectas secantes a la curva de la ecuación que se está analizando, y se verifica la intersección de dichas rectas con el eje de las x para conocer si es la raíz que se busca.

El método de secante resuelve una ecuación similar a la del método de Newton-Raphson ya que, converge con la raíz con una velocidad, aunque de igual forma corre el riesgo de no converger con esta nunca. Su principal diferencia con el método de Newton-Raphson es que no se requiere obtener la derivada de la función para realizar las aproximaciones, lo cual facilita las cosas al momento de crear un código para encontrar raíces por medio de este método.

El método de la secante realiza una aproxima de la segunda derivada por una línea recta, este valor se aproxima por,

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \quad (2.31)$$

Se reemplaza 2.31 en la fórmula de Newton resultando:

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}, \quad n \geq 1 \quad (2.32)$$

La recta tangente a la curva se reemplaza por una recta secante. El cero de f se aproxima por el cero de la recta secante a f , Si x_0 y x_1 son las aproximaciones iniciales, la aproximación x_2 es la intersección de la recta que une los puntos $(x_0, f(x_0))$ y $(x_1, f(x_1))$. La aproximación x_3 es la intersección de la recta que une los puntos $(x_1, f(x_1))$ y $(x_2, f(x_2))$ y así sucesivamente [31].

El método de la secante aparentemente es un método difícil de manejar pero en la práctica funciona bastante bien. El orden de convergencia para funciones de una sola variable es de $(1 + \sqrt{5}) / 2 \approx 1.6$, Por lo tanto su convergencia es menor que la del método de Newton de diferencias finitas, pero en muchas ocasiones funciona mejor que este en lo que a número de evaluaciones de la función objetivo se refiere.

2.5.2 OPTIMIZACIÓN NUMÉRICA MULTIVARIABLE SIN RESTRICCIONES.

La optimización numérica de funciones no lineales requiere la utilización de técnicas de optimización eficientes y robustas. La eficiencia es importante porque la solución de estos problemas se lleva a cabo por un procedimiento iterativo. La robustez (habilidad para encontrar la solución) es una propiedad deseable dado que el comportamiento de las funciones no lineales puede ser impredecible, incluso pueden presentar uno o varios máximos, mínimos y puntos de silla. En algunas regiones la convergencia hacia el óptimo puede ser muy lenta, necesitando mucho tiempo de cálculo etc.

La mayor parte de los métodos los cuales realizan procedimientos iterativos que son efectivos, alternan la optimización en dos fases [29]:

- a) Elección de una dirección s^k .
- b) Movimiento en la dirección s , (en alguna extensión, o hasta encontrar un mínimo) para encontrar un nuevo punto $x^{k+1} = x^k + \Delta x$ donde Δx^k se suele llamar el tamaño del paso.

Además de (a) y (b) un algoritmo debe especificar:

- c) Un vector de valores iniciales $x^0 = [x_1^0, x_2^0, \dots, x_n^0]^T$
- d) Un criterio de convergencia para la terminación del algoritmo.

La mayoría de algoritmos siguen una metodología similar. Se determina un punto inicial, se evalúa la función en ese punto y se elige una dirección de búsqueda. Se comienza entonces un movimiento en la dirección de búsqueda, hasta encontrar un óptimo en esa dirección, o bien hasta que se produzca una mejoría determinada.

2.5.2.1. Métodos Directos

En los años anteriores, los métodos de optimización estaban habitualmente restringidos a los métodos indirectos en los cuales el cálculo del extremo potencial estaba limitado al uso de derivadas y las condiciones necesarias de optimalidad. Los métodos directos consisten básicamente en la búsqueda de un óptimo por comparación sucesiva de los valores de la función $f(x)$ en una secuencia de puntos $x_1, x_2, x_3, \dots, x_n$ sin la necesidad de hacer intervenir derivadas analíticas.

Para llevar a cabo los métodos directos de minimización numérica solamente se usa el valor de la función objetivo. Se comienza con un valor inicial de x y se continúa seleccionando valores de x de acuerdo con una estrategia pre-seleccionada. El proceso termina cuando $f(x^{k+1}) - f(x^k) < \varepsilon$ donde el superíndice k designa el número de

iteración y ε es la tolerancia pre-especificada o criterio de tolerancia [29].

Los métodos directos no hacen uso de la información proporcionada por las derivadas. Bajo estas circunstancias, estos métodos se pueden usar con bastante efectividad, pero son muy ineficientes comparados a los métodos indirectos. Tienen la ventaja de que estos métodos son muy simples de entender y muy fáciles de ejecutar [29].

2.5.2.1.1. Métodos de búsqueda aleatoria

Los procedimientos de búsqueda aleatoria se basan en la utilización de generadores de números aleatorios en la búsqueda del punto mínimo, en principio selecciona un vector inicial x^0 , evalúa la función objetivo en ese punto y entonces aleatoriamente selecciona otro vector x^1 . Tanto la dirección de búsqueda como la longitud de búsqueda son elegidas simultáneamente. Después de una o más fases, el valor de $f(x^k)$ se compara con el mejor valor previo de $f(x)$ y se toma la decisión de continuar o terminar el procedimiento.

La búsqueda aleatoria casi siempre converge a la solución óptima y es aplicable a la casi totalidad de los problemas de optimización, debido a los pocos supuestos necesarios para su uso [32].

Existen diversas variaciones de este algoritmo, aunque estrictamente hablando sólo se alcanza la solución cuando $k \rightarrow \infty$ pero desde un punto de vista práctico, si el objetivo tiene una forma muy plana se pueden encontrar soluciones sub-óptimas bastante aceptables. Aunque el método es bastante ineficiente por sí mismo, puede dar valores aceptables de partida para otros métodos [29], En [33] se detalla su típica formulación matemática.

2.5.2.1.2. Método simplex flexible.

Método para optimización de una función multivariable sin Restricciones, es un método cuya gran virtud es su sencillez, método muy práctico, ya que solamente trabaja

con los coeficientes de la función objetivo, en general emplea un proceso iterativo que principia en un punto extremo factible, normalmente el origen, y se desplaza sistemáticamente de un punto extremo factible a un punto extremo adyacente, hasta que se llega por último al punto óptimo.

Este método en principio se basa en tomar una Figura regular (conocida como *simplex*) como base. Así en 2 dimensiones tal Figura debería ser un triángulo equilátero. Los experimentos se localizan de tal manera que la función objetivo se evalúa en cada uno de los vértices que forman la Figura geométrica.

Los valores de la función objetivo obtenida en cada uno de los vértices se comparan entre sí rechazando el peor valor, es decir el que da un valor más elevado formando una nueva figura geométrica por reflexión del peor de los puntos respecto a los que no han sido rechazados. La repetición del proceso, eliminando los vértices que dan un mayor valor para f (función objetivo), junto con ciertas reglas para reducir la reflexión, o bien para extenderla, permiten la obtención de los parámetros que minimizan dicha función simplemente por evaluación de la función con los distintos juegos de parámetros correspondientes a cada uno de los vértices del *Simplex* [34].

Para la selección de la Figura *Simplex* inicial se puede empezar por el caso de dos dimensiones. Así la distancia entre un punto $x_1 = (x_{11}, x_{12})$ y otro $x_2 = (x_{21}, x_{22})$ viene dada por la expresión:

$$(x_{1j} - x_{1k})^2 + (x_{2j} - x_{2k})^2 = \sum_{i=1}^2 (x_{ij} - x_{ik})^2 = a^2 \quad j \neq k \quad (2.33)$$

como tenemos tres puntos podemos hacer $C_2^3 = \frac{3!}{2!1!} = 3$ así en dos dimensiones (3 vértices) podemos definir tres distancias entre vértices iguales. Por la tanto, especificando un punto base x_1 , el punto x_2 estará localizado en un punto cualquiera de una circunferencia de radio a centrada en x_1 . Una vez elegido el punto x_2 , el punto x_3 puede tomar dos opciones en la intersección entre las circunferencias de radio a centradas en x_1 y en x_2 respectivamente como se ilustra en la Figura 2.11 [29].

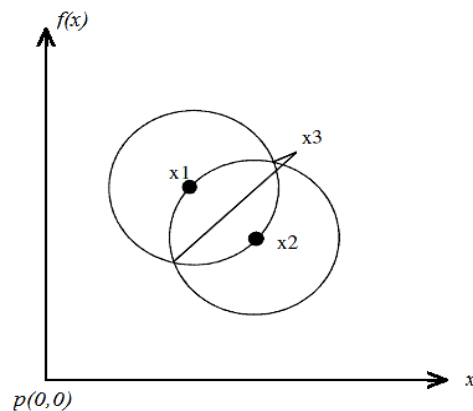


Figura. 2.11. Selección de la figura simplex inicial

De forma similar en n dimensiones tenemos $n + 1$ puntos, en [29] se detalla la tabla como una propuesta para comenzar un simplex de n variables cuando x_1 es el origen u otro punto cualquiera.

Una vez fijada la Figura *Simplex* inicial, se sigue una búsqueda secuencial, en la que en cada paso se eliminará un vértice y se incluirá otro nuevo.

2.5.2.1.3. Métodos de búsqueda en rejilla

Los métodos de búsqueda en rejilla se pueden aplicar seleccionando una serie de puntos alrededor de un punto base de referencia, después se pasa al punto que más mejora la función objetivo y se continúa la búsqueda. Sin embargo el sistema es muy ineficaz, por ejemplo con $n = 10$ y una búsqueda factorial a tres niveles deberíamos realizar $3^{10} - 1 = 59048$ evaluaciones de la función objetivo, lo cual es obviamente prohibitivo [29].

2.5.2.2. Métodos Indirectos

Los métodos indirectos para optimización multivariable sin restricciones hacen uso de las derivadas en la determinación de la dirección de búsqueda. Son métodos más complejos dado que trabajan con n variables de una función objetivo así como

restricciones y es así que uno de los métodos más utilizados es el método de Gradiente³ Conjugado que requiere de la combinación obtenida del vector gradiente con la información acerca del vector gradiente de iteraciones previas. Lo que hace el método es calcular la nueva dirección de búsqueda utilizando una combinación lineal del gradiente en la etapa considerada y el de la etapa anterior. La principal ventaja del método es que necesita almacenar muy poca cantidad de información con lo que puede ser programado fácilmente incluso en calculadoras.

Esta clasificación encierra diversos métodos a parte del Método de Gradiente Conjugado tales como: Método de Gradiente (Máximo Descenso), Método de Newton (Segundo Orden), Forzando a la Matriz Hessiana a ser Definida Positiva (Segundo Orden) y, métodos de Secante (Segundo Orden), No obstante es necesario recalcar que el método Newton y Secante siguen el mismo concepto que para una única variable.

El método Newton para segundo orden hace uso de la aproximación de segundo orden de la función utilizando las derivadas segundas con respecto a cada una de las variables independientes. De esta forma es posible tener en cuenta la curvatura de la función en el punto e identificar las mejores direcciones de búsqueda y el mínimo de $f(x)$ se obtiene diferenciando la aproximación cuadrática de $f(x)$ con respecto a cada una de las variables e igualando a cero.

El método de secante inicia con la misma ecuación usada en el método de Newton, pero a diferencia del método de Newton se tiene como prioridad conseguir una actualización de la matriz hessiana $H(x)$ usando solamente las derivadas parciales primeras de la función, a semejanza de lo que hacíamos en los métodos de secante unidimensionales.

Si queremos aproximar la matriz inversa, el método de la secante calcula un nuevo vector x a partir de otro de una etapa precedente a través de ecuaciones análogas a las del método de Newton [35].

³ Vector en un punto x que proporciona la dirección (local) de máxima variación de la función $f(x)$.

2.5.3 OPTIMIZACIÓN NUMÉRICA MULTIVARIABLE CON RESTRICCIONES

Un gran número de modelos de optimización imponen a las variables una serie de restricciones que se traducen en que el punto óptimo no se busca en todo el espacio sino en un subconjunto del espacio definido por las restricciones.

La principal dificultad de los problemas de minimización con restricciones reside en que no se tiene una caracterización de un punto mínimo que dependa únicamente de la función objetivo, también se requiere que se satisfagan las condiciones respecto a las restricciones.

En este método el objetivo consiste en determinar los valores de x_1 y x_2 que hacen máximo o mínimo el valor de una función objetivo f dependiente de dos variables. Respecto al caso de una función dependiente de dos variables el problema puede escribirse de la siguiente manera:

$$\text{Max } f = f(x_1, x_2) \quad (2.34)$$

Donde la condición de primer orden (Derivada primera) viene dada por el siguiente sistema de ecuaciones:

$$\begin{cases} f'_{x_1}(x_1, x_2) = 0 \\ f'_{x_2}(x_1, x_2) = 0 \end{cases} \Rightarrow (x_1^*, x_2^*) \quad (2.35)$$

Donde x_1^* y x_2^* son la solución óptima para la variable x .

La condición de primer orden geoméricamente establece que hay que buscar los valores de las variables o punto del dominio para el cual el plano tangente a la gráfica de f es horizontal. Dicho sistema por lo general es no lineal y puede tener varias soluciones, por lo tanto pueden existir varios puntos para los cuales el plano tangente a la gráfica de f es horizontal [36]. Dichos puntos son denominados puntos críticos o candidatos a óptimos,

éstos pueden ser máximos, mínimos o puntos de silla como se aprecian en las siguientes gráficas:

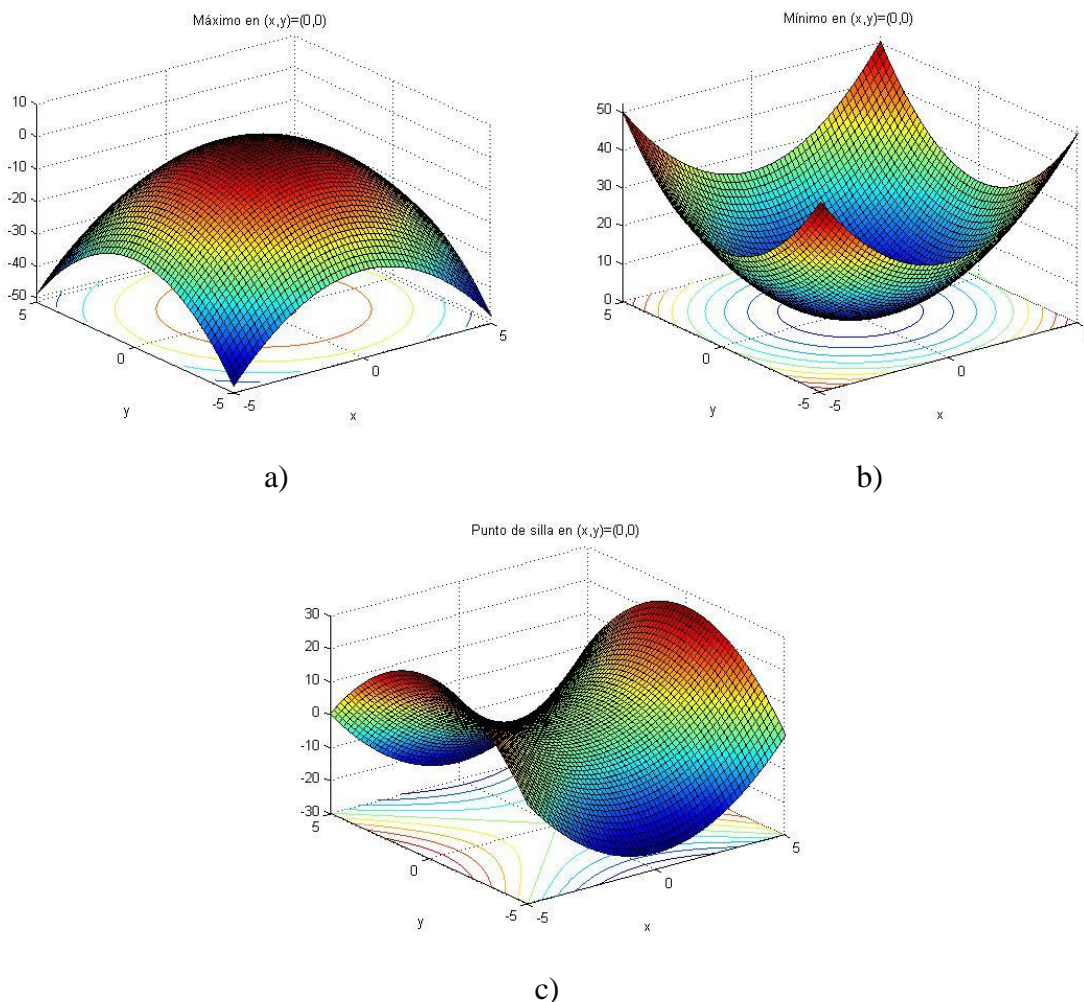


Figura. 2.12. Puntos óptimos de una función: a) máximos, b) mínimos, c) puntos de inflexión (silla)

Para la condición de segundo orden (condición suficiente) hace referencia a la matriz hessiana evaluada en cada punto crítico:

$$H_f(x_1^*, x_2^*) = \begin{pmatrix} f''_{x_1x_1}(x_1^*, x_2^*) & f''_{x_1x_2}(x_1^*, x_2^*) \\ f''_{x_2x_1}(x_1^*, x_2^*) & f''_{x_2x_2}(x_1^*, x_2^*) \end{pmatrix} \quad (2.36)$$

Dicha matriz provee toda la información necesaria de f en las inmediaciones de cada punto por medio de una aproximación de Taylor⁴ de segundo orden pues da a conocer si el entorno del punto es cóncavo o convexo. Algebraicamente, para discernir si cada punto crítico es efectivamente un punto que optimiza a f se hace referencia al concepto de menores principales⁵ de orden n asociado a una matriz de orden m que es el determinante que surge de considerar una submatriz de orden n conformada por las primeras n filas y las primeras n columnas. La condición de segundo orden se puede enunciar en términos de los menores principales de la matriz hessiana evaluada en cada punto crítico.

En esta situación, para el caso de mínimo, es suficiente que todos los menores principales sean positivos mientras que para el caso de máximo, se requiere que alternen en signo comenzando por signo negativo. Algebraicamente se tiene [36]:

$$\text{Mínimo} \begin{cases} |H_1| > 0 \\ |H_2| > 0 \\ \vdots \\ |H_n| > 0 \end{cases} \quad \text{Máximo} \begin{cases} |H_1| < 0 \\ |H_2| > 0 \\ \vdots \\ (-1)^n |H_n| > 0 \end{cases} \quad (2.37)$$

Ahora para el caso de un problema donde tengamos n variables y m restricciones ($m < n$) se debe hallar un punto n dimensional de la forma x_1, x_2, \dots, x_n tales que satisfaciendo un conjunto de restricciones confieran a f un valor máximo o mínimo. Se puede representar matemáticamente por la expresión:

$$\text{Max } f = f(x_1, x_2, \dots, x_n) \quad (2.38)$$

s.a :

$$g(x_1, x_2, \dots, x_n) = m_1$$

$$g(x_1, x_2, \dots, x_n) = m_2$$

⋮

$$g(x_1, x_2, \dots, x_n) = m_n \quad (2.39)$$

⁴ Aproximación de funciones mediante una serie de potencias o suma de potencias enteras de polinomios como $(x-a)^n$ llamados términos de la serie, la suma se calcula a partir de las derivadas de la función para un determinado punto a en un entorno sobre el cual converja la serie.

⁵ Donde H_n denota al menor principal de orden n de la matriz H .

Para su resolución se hace uso de la función Lagrangeana (L) y de los multiplicadores de Lagrange (λ) [37] [38] y se generaliza de la siguiente manera:

$$\begin{aligned} \underset{x_1, x_2, \dots, x_n}{Max} \quad L = & f(x_1, x_2, \dots, x_n) + \lambda_1[m_1 - g_1(x_1, x_2, \dots, x_n)] \\ & + \lambda_2[m_2 - g_2(x_1, x_2, \dots, x_n)] + \lambda_m[m_m - g_m(x_1, x_2, \dots, x_n)] \end{aligned} \quad (2.40)$$

Donde a λ , se le conoce con el nombre de multiplicador de Lagrange.

La condición de primer orden ahora se escribe en términos de la función Lagrangeana y arroja el siguiente sistema de $n + m$ ecuaciones en $n + m$ incógnitas:

$$\left\{ \begin{array}{l} L_{x_1}(x_1, x_2, \dots, x_n, \lambda_1, \dots, \lambda_m) = 0 \\ L_{x_2}(x_1, x_2, \dots, x_n, \lambda_1, \dots, \lambda_m) = 0 \\ \vdots \\ L_{x_n}(x_1, x_2, \dots, x_n, \lambda_1, \dots, \lambda_m) = 0 \\ L_{\lambda_1}(x_1, x_2, \dots, x_n, \lambda_1, \dots, \lambda_m) = 0 \\ L_{\lambda_2}(x_1, x_2, \dots, x_n, \lambda_1, \dots, \lambda_m) = 0 \\ \vdots \\ L_{\lambda_m}(x_1, x_2, \dots, x_n, \lambda_1, \dots, \lambda_m) = 0 \end{array} \right. \Rightarrow (x_1^*, x_2^*, \dots, x_n^*, \lambda_1^*, \lambda_2^*, \dots, \lambda_m^*) \quad (2.41)$$

Las soluciones para dicho sistema constituyen los puntos críticos candidatos a extremos relativos (máximos o mínimos) de la función sujeta a las restricciones. En [36] se detalla la deducción de la matriz hessiana y una vez obtenida dicha matriz se debe atender al signo de los menores principales a partir del orden $2m + 1$. Para el caso de mínimo se requieren que todos tengan el signo de $(-1)^m$ y para máximo se requiere que los menores alternen en signo comenzando por el signo de $(-1)^{m+1}$.

CAPÍTULO 3

SIMULACIÓN WSN EN SOFTWARE ns-2

3.1 ESCENARIOS WSN DE SIMULACIÓN

En este capítulo se evalúa los parámetros más importantes que se deben tener en consideración al momento de implementar una red WSN. Para definir el escenario de simulación se toma en cuenta un modelado en relación a la distribución de los nodos sensores sobre la red. Entre los modelados existentes se encuentran los de tipo Determinista y Aleatorio, La ubicación de un sensor de forma determinista puede ser conveniente en entornos amigables y accesibles, ya que la colocación estaría bien controlada a fin de que cada nodo pueda ser colocado en una posición específica con un patrones predefinidos de implementación uniforme. La distribución de sensores en forma aleatoria es generalmente considerada en aplicaciones militares y para áreas remotas [39]. La distribución aleatoria ofrece para nuestro estudio un escenario más cercano a la realidad, pues la superficie terrena que existe sobre un Volcán no es totalmente plana y permite implementar una topología física con patrones de implementación uniformes, por tal razón en este proyecto se utiliza el despliegue de nodos sobre la red de tipo Aleatorio ya que se pretende simular un escenario donde la distancia y posición de nodos no sea una condición fundamental para obtener un buen rendimiento de la red.

Los escenarios de simulación de una red WSN dependen principalmente de la topología física en base a la cual se posicionan los nodos sensores sobre el área definida, Zigbee soporta diversas topologías como las mencionadas en el Capítulo II, cada una de ellas poseen sus propias características de diseño y en base a éstas pueden ser elegidas para su utilización en diferentes aplicaciones, adicionalmente es importante definir qué tipo de módulos de comunicación se utilizarán, ya que así se conocerán la información y

características técnicas necesarias de dichos módulos a tomarse en cuenta como parámetros iniciales dentro de la simulación de red, y con ésta información definir que topología conviene para nuestro estudio.

En el mercado existen diversos fabricantes de módulos de comunicación para WSN, el *hardware* de la mayoría de éstos promocionan las ventajas y desventajas de su funcionamiento al ser implementados, sin embargo a partir de los últimos años éstos han lanzado al mercado módulos para WSN con nuevas tendencias, como los ZigBee RF fabricados por *Digi International*⁶, existen varios tipos de hardware y modelos de los que se puede seleccionar de acuerdo a nuestros requerimientos, tales como: XBEE2, XBEEPRO2, PRO S2B, etc. Sin embargo nuestro proyecto viene siendo un aporte significativo a las investigaciones que se han venido ejecutando durante años por el Centro de Investigaciones de Redes Ad Hoc (CIRAD) de la Universidad de las Fuerzas Armadas – ESPE, entorno a la monitorización de señales sísmicas del Volcán Cotopaxi, donde mediante un minucioso estudio y pruebas de consumo de energía, radiofrecuencia y propagación se determinó que los módulos XBee funcionan de acuerdo a las especificaciones definidas en el estándar, soportando las necesidades particulares de las redes de sensores de bajo costo y consumo, requiriendo alimentación mínima y a la vez permitiendo el transporte confiable de datos entre dispositivos remotos, opera dentro de la banda 2.4 GHz ISM [40]. Es por esto que en el presente proyecto se despliega una red en escenarios simulados en base a la información de las características físicas y lógicas de los módulos XBee-PRO (S2) [41], más adelante se describe más detalladamente la información relevante de dichos módulos que se utilizará para nuestro interés.

La WSN aplicada a monitorización que se implementa más adelante es diseñada bajo la topología estrella, ya que el hardware usado contempla esta topología dentro de sus opciones. Además en [42] y [10] se ejecutan proyectos relacionados con la tecnología Zigbee estándar IEEE 802.15.4 acerca de monitorización de variables, que después de un análisis comparativo determinan que la topología estrella tiene un mejor efecto de *Packet Delivery Ratio*, adicionalmente se necesita provocar que todos los nodos de la red tomen datos para un mejor análisis, pues esto se logra con ayuda de la topología estrella ya que

⁶ Industria de servicios y productos inalámbricos, *routers*, *gateways*, adaptadores de comunicación inalámbrica (ZigBee, Wi-Fi, RF patentada), etc.

solo necesita un dispositivo para clasificar, almacenar e interpretar la información, consiguiendo una mayor extensión de área para la toma de datos y menor consumo de energía de la red.

No se evaluó las características de la topología malla ya que esta se usa en áreas que requieren mayor seguridad, redundancia, tolerancia a fallas, y son aplicadas más comúnmente en campos médicos, mientras que el propósito de este trabajo de investigación está orientado a la medición de variables ambientales.

Los nodos distribuidos en cada escenario están configurados para formar una red inalámbrica descentralizada ya que cada nodo está preparado para reenviar datos a lo demás, los nodos sensores son únicamente fijos y como se mencionó la topología física a utilizar es de tipo estrella que junto al modelado aleatorio se podrá tener una comunicación entre nodos completa.

El tamaño del escenario de simulación se relaciona directamente con la cantidad de nodos que se posee en la red como se ilustra en la Figura 3.1 en el cual se presenta uno de los escenarios con quince nodos situados de forma aleatoria donde el nodo cero es el coordinador, éstos se despliegan en un área aproximada de cien por ciento veinte y cinco metros cuadrados, esta área es común para todos los escenarios.

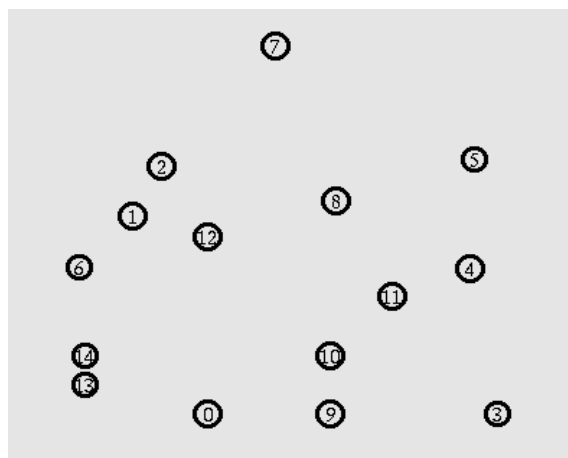


Figura. 3.1. Distribución aleatoria, 15 nodos

Para evaluar la red WSN se utiliza *ns-2* [43] que es el simulador de redes en código abierto más extendido en áreas de investigación, siendo aplicado para simular estructuras, y escenarios de diferente tamaño, *ns* proporciona soporte para la simulación de multitud de protocolos de las capas de aplicación (http, ftp, cbr, etc.), transporte (TCP, UDP, RTP, SRM), protocolos de enrutamiento *unicast*, *multicast*, *broadcast*, etc., tanto para redes cableadas como inalámbricas o vía satélite, además de simular topologías complejas con un gran número de generadores de tráfico, presenta una ventaja esencial pues permite extraer trazas (cada línea del archivo traza representa el comportamiento de los paquetes durante la simulación). Con información de la red, y ayuda de otras herramientas se puede generar gráficos representativos de los resultados para posteriores análisis.

Uno de los propósitos fundamentales de los escenarios próximos a simular es crearlos lo más cercano posible a la realidad, es decir se desea simular una red compuesta de nodos que censen señales de un evento volcánico, por lo tanto, es de prioridad conocer que existen una variedad de fenómenos y procesos físicos que ocurren antes, durante y después de una erupción volcánica, no obstante, debido a la velocidad a la que se propaguen estos eventos sobre la superficie hasta alcanzar la altura de uno de los sensores puede variar la distancia y tiempo de llegada de las señales a los diferentes nodos.

Se puede visualizar de manera más global la dinámica interna del Volcán en base a parámetros sísmicos, de deformación, de infrasonido o de gases, sin embargo, el análisis de las ondas sísmicas generadas en el interior del Volcán proveen la información más representativa sobre la dinámica del mismo. Ya que estos estudios pueden revelar información correspondiente a procesos profundos como superficiales, incluso existen estudios que relacionan la deformación del edificio volcánico con las señales sísmicas, con la emisión de gases y con la parte acústica [44]. Por tales motivos, se ha decidido plantear una metodología que relacione el dinamismo volcánico es decir, las señales sísmicas superficiales con el tiempo de inicio de recolección de información por parte de los nodos dispersos en la red.

Para definir la metodología planteada es necesario iniciar conociendo de forma general los tipos de ondas que pueden generarse en un sismo, pues se pueden clasificar en dos tipos de ondas de cuerpos, las ondas de compresión o primarias y las ondas de cizalla o secundarias [45]. Los términos “primarias” y “secundarias” son usados debido a la

velocidad a la que se propagan, las ondas primarias o P tienen una velocidad de propagación mayor a las ondas secundarias S ($\alpha > \beta$), y por lo tanto éstas arriban primero al detector, además tienen la propiedad de atravesar sólidos y fluidos. Por lo contrario las ondas S, son ondas en las cuales las partículas del medio se desplazan perpendicularmente a la dirección de propagación, por ello están asociadas con deformaciones del terreno [46], es por esto que son más lentas y provocan mayor daño [45]. Los valores de velocidades de propagación propuestos para las ondas P y S están relacionados con el modelo de velocidades *Absorption of seismic waves* (ASW), para el Ecuador, que nació a partir del “Proyecto Cooperativo Internacional - Nariño” en 1973 [47]. El modelo de velocidades se presenta en la Tabla 3.1 el cual es un modelo de 5 capas (modificado internamente por el Instituto Geofísico del EMAPQ, 1988) [45].

Tabla. 3.1.

Modelo “ASW” de velocidades para las ondas P y S, en el Ecuador [47].

Modelo de velocidades “ASW”		
Profundidad (km)	Velocidad de la onda P V_p (km/s)	Velocidad de la onda S V_s (km/s)
0.0	3.32	1.94
3.0	5.90	3.45
15.0	6.20	3.63
30.0	6.70	3.92
50.0	8.10	4.74

Para definir el tiempo de arribo de las ondas sísmicas a los detectores que no son más que los sensores, es necesario el estudio de la relación tiempo – trayectoria para sismos locales dado que cuando se produce un terremoto o sismo, este no se presenta simultáneamente en todos los sitios que es observado. Omori [48] establece una fórmula, que relaciona la distancia hipocentral (D) (en este caso D tiende a la distancia del epicentro cuando esta es mucho mayor que la profundidad), en esta relación se supone simplemente que todas las curvas de tiempo trayectoria, hasta casi 1000 km, son lineales [49].

$$T_{sp} = \frac{D(V_p - V_s)}{V_s \cdot V_p} \quad (3.1)$$

Donde T_{sp} es el tiempo S-P, D la distancia lineal desde el epicentro hacia un punto sobre la superficie, V_p la velocidad de la onda P y V_s la velocidad de la onda S.

Una vez especificados los valores de velocidad de propagación se establece como profundidad del hipocentro 30 km ya que en [50] se detalla un informe anual acerca de los eventos sísmicos presentados sobre el Volcán Cotopaxi desde el 2001 hasta el año 2009 donde la profundidad de los eventos sísmicos oscilan entre 0 y 30 km [44]. Las velocidades de propagación correspondientes a esta profundidad se encuentran en la Tabla 3.1.

En la mayoría de redes inalámbricas desplegadas con topología estrella la distancia entre los dispositivos RFD y FFD (Coordinador) sigue un patrón de implementación uniforme, sin embargo este trabajo contempla un patrón no constante como se observa en la Figura 3.1. En la Tabla 3.2 se muestra la distancia propuesta entre cada nodo hacia el nodo coordinador que conjuntamente con la distancia entre el epicentro y nodos actúan sobre la ecuación 3.1 y se calcula el tiempo de llegada de las ondas P y S a cada uno de los nodos de la red así como el tiempo de inicio de transmisión de datos.

Tabla. 3.2.

Distancia y Tiempo de arribo de las ondas P y S (Calculados)

Datos de propagación de Ondas P y S hacia la WSN				
Nodo	Distancia Nodo -Gateway (metros)	Distancia Epicentro - Nodo (metros)	Tsp (seg)	Tiempo de inicio de transmisión (seg)
1	35.74	86.11	0.00911	20.00911
2	42.76	85.57	0.00906	20.00906
3	47.07	20.00	0.00212	20.00212
4	49.37	34.65	0.00367	20.00367
5	61.22	49.39	0.00523	20.00523
6	32.56	91.47	0.00968	20.00968
7	63.41	83.92	0.00888	20.00888
8	41.73	58.66	0.00621	20.00621
9	20.00	47.07	0.00498	20.00498
10	22.36	48.12	0.00509	20.00509
11	36.06	42.12	0.00446	20.00446
12	30.00	73.48	0.00778	20.00778
13	20.62	87.22	0.00923	20.00923
14	22.36	87.65	0.00928	20.00928

Continúa →

Nodo	Distancia Nodo -Gateway (metros)	Distancia Epicentro - Nodo (metros)	Tsp (seg)	Tiempo de inicio de transmisión (seg)
15	29.15	57.76	0.00611	20.00611
16	43.86	94.01	0.00995	20.00995
17	46.10	103.19	0.01092	20.01092
18	35.44	101.57	0.01075	20.01075
19	15.62	79.70	0.00844	20.00844
20	40.00	107.07	0.01133	20.01133
21	59.55	118.67	0.01256	20.01256
22	25.06	46.64	0.00494	20.00494
23	10.00	67.82	0.00718	20.00718
24	10.63	75.40	0.00798	20.00798
25	48.37	111.68	0.01182	20.01182
26	50.25	87.72	0.00928	20.00928
27	47.54	65.91	0.00698	20.00698
28	48.17	44.61	0.00472	20.00472
29	63.01	115.90	0.01227	20.01227
30	57.57	74.38	0.00787	20.00787
31	62.36	103.29	0.01093	20.01093
33	8.00	75.07	0.00795	20.00795
35	54.74	79.30	0.00839	20.00839
37	54.64	57.67	0.00610	20.00610
39	67.98	114.65	0.01214	20.01214

Otro de los parámetros importantes en el escenario de simulación es el modelo de radio propagación que se emplea y una de las mejores opciones dentro de éstos modelos es el de dos rayos (*Two-Ray Ground*), pues se trata de una versión mejorada del modelo de propagación de espacio libre. Se basa en óptica geométrica considerando tanto la transmisión directa como la componente de propagación reflejada en la tierra entre el transmisor y el receptor. Por lo tanto, se utiliza éste modelo por ser más cercano a la realidad ya que se supone un área de monitorización con una superficie medianamente plana.

Con respecto al protocolo de encaminamiento el protocolo *Destination Sequenced Distance Vector* (DSDV) y el protocolo *Dynamic Source Routing* (DSR), utilizan paquetes de enrutamiento que superan los 127 bytes que se establece en el estándar IEEE 802.15.4 para realizar el envío de una trama, por lo tanto el protocolo utilizado es AODV (*Ad-Hoc*

On Demand Distance Vector). Mientras que el protocolo AODV no utiliza paquetes mayores, además tiene la ventaja de genera su tabla de ruteo solamente cuando tiene la necesidad de transmitir un mensaje a un destino en particular es decir es reactivo, de esta manera se logra disminuir el consumo de energía, el ancho de banda del enlace, los mensajes de control de tráfico y el *overhead* mejorando así la escalabilidad y la *performance* de la red, este protocolo también está diseñado para redes que contienen desde decenas hasta miles de nodos.

Se ha optado por agentes UDP (*User Datagram Protocol*). Bajo UDP se utiliza el tipo de tráfico CBR (*Constant Bit Rate*) ya que permite simular una categoría de servicio de datos donde el tráfico tiene una consistencia de velocidad, donde hay una confianza intrínseca entre el tiempo de sincronización entre la fuente y el destino.

En base a los parámetros definidos se crea un total de 34 escenarios de simulación: 30 conformados por 2, 3, 4, ..., 31 nodos y 4 escenarios formados por 33, 35, 37 y 39 nodos, se resume en la siguiente Tabla 3.3.

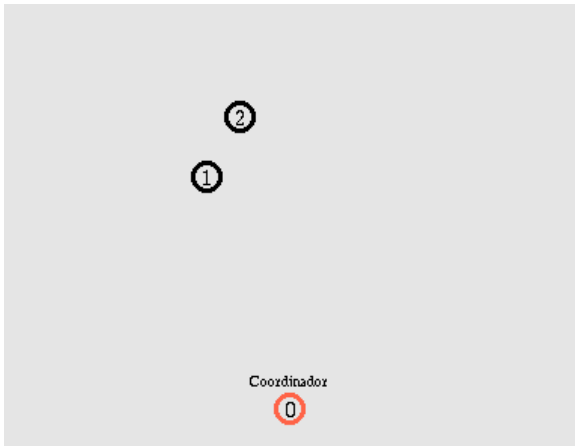
Tabla. 3.3.

Modelado y Escenarios de Simulación

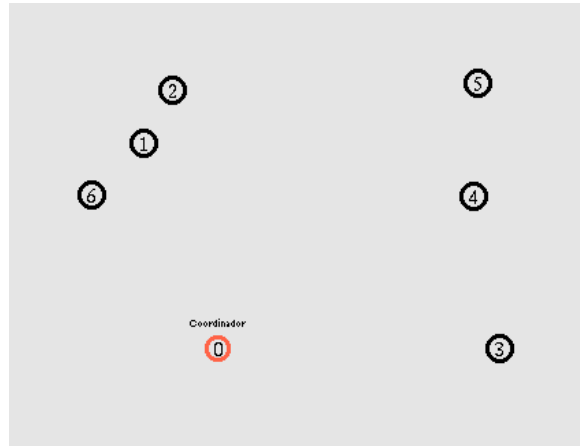
Modelado	Escenario
Aleatorio	2, 3, 4, 5, 6, ..., 30 y 31 (nodos)
Aleatorio	33, 35, 37 y 39 (nodos)

Siguiendo la tabla anterior se implementan los escenarios de simulación de la red empezando el primero con 1 nodo y el coordinador, el número de nodos se incrementa de uno en uno por cada simulación y a partir de la simulación 31 se aumenta de dos nodos hasta alcanzar los 39 nodos, haciendo un total de 34 escenarios.

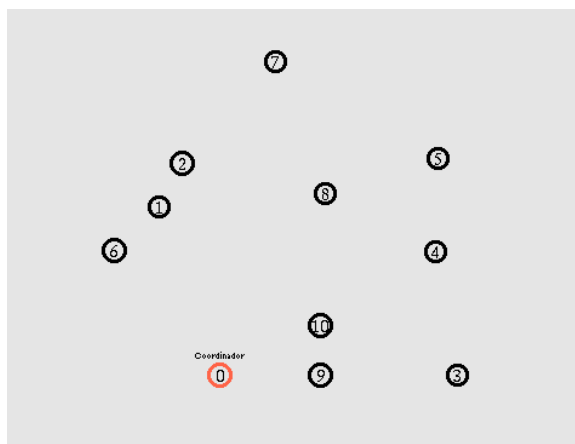
En la Figura 3.2 se muestran 8 de las 34 topologías implementadas, constan de un dispositivo FFD (Coordinador) separado de cada uno de los dispositivos RFD por una distancia variable definida en la Tabla 3.2.



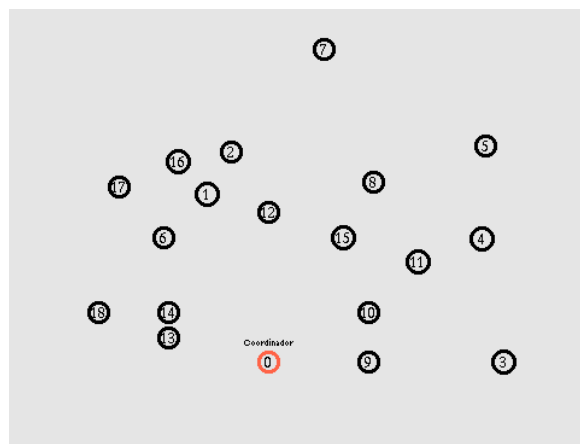
a) Escenario de 3 nodos



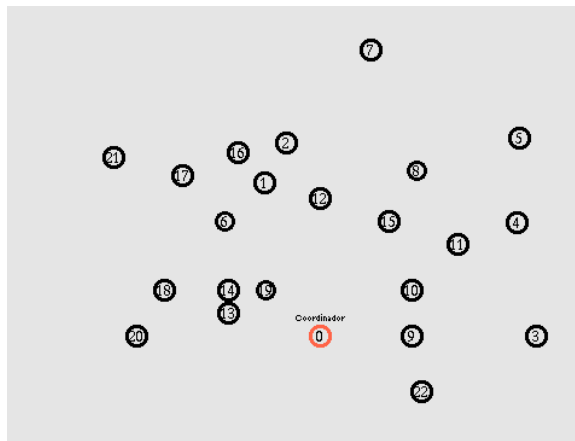
b) Escenario de 7 nodos



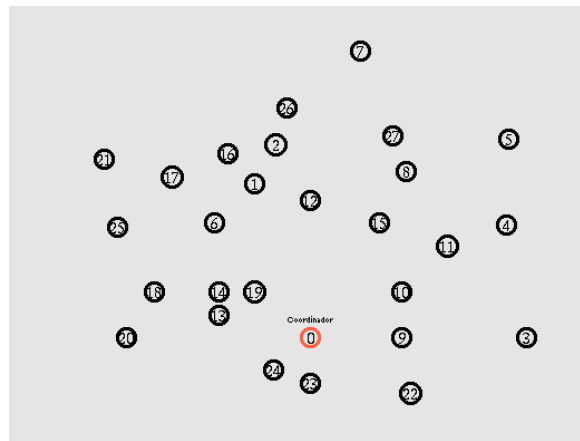
c) Escenario de 11 nodos



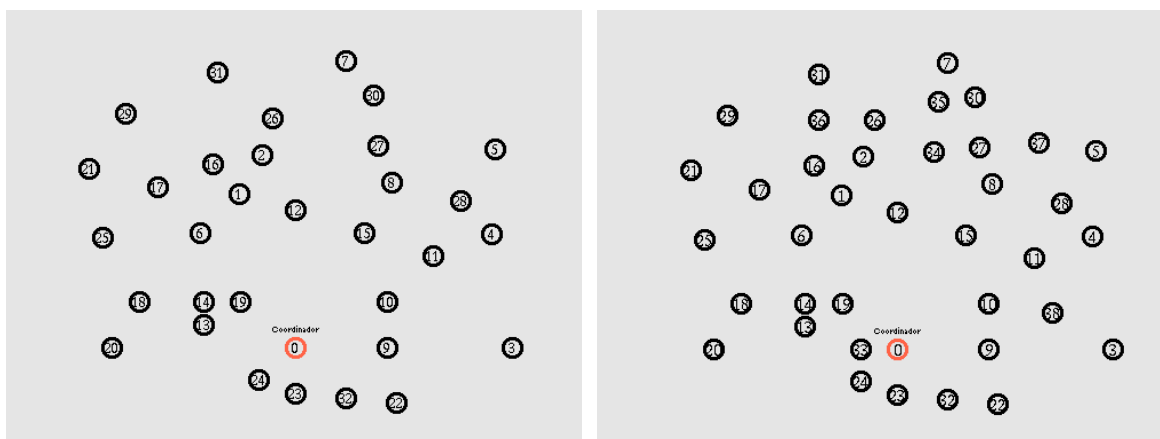
d) Escenario de 19 nodos



e) Escenario de 23 nodos



f) Escenario de 28 nodos



g) Escenario de 33 nodos

h) Escenario de 39 nodos

Figura. 3.2. Escenarios de Simulación, 3, 7, 11, 19, 23, 28, 33, y 39 nodos

3.2 SIMULACIÓN DE LAS TOPOLOGÍAS DE RED WSN

Las simulaciones desarrolladas en este Capítulo dependen de ciertos parámetros que intervienen en el comportamiento de los nodos así como de la transmisión de paquetes en la red, a continuación en Tabla 3.4 se resume la información y características que se requiere para la creación de cada escenario mediante el simulador de red *ns-2*.

Los parámetros de potencia tal como ganancia, potencia de transmisión, sensibilidad y tipo de antena corresponden a las características de funcionamiento de los equipos seleccionados para el presente proyecto. Por otro lado, el protocolo de enrutamiento, modelo de radio propagación, tasa de transmisión, tipo de tráfico y tamaño de paquetes son parámetros netamente concernientes al estándar y tecnología inalámbrica IEEE 802.15.4 (ZigBee, 2.4GHz).

Los parámetros necesarios de configuración dentro del entorno *ns-2* como se mencionó son respecto a las características de los módulos de comunicación así como del estándar IEEE 802.15.4 que se utilizan, éstos determinan el desempeño de la red dentro del tiempo de simulación (140 segundos), a continuación en la Tabla 3.5 se muestra la disposición de parámetros en líneas de comando.

Tabla. 3.4.
Parámetros de Simulación

Parámetros de transmisión	
Modelo de radio propagación	<i>Two-Ray Ground</i>
Protocolo de Enrutamiento	AODV
Tecnología Inalámbrica	IEEE 802.15.4 (<i>ZigBee, 2.4GHz</i>)
Tasa de transmisión	250 kbps
Tiempo simulado	140 seg.
Antena	Omnidireccional, 50mW (+17dBm) (XBee-PRO(S2))
Ganancia de antena	2.1dBi (XBee-PRO (S2))
Sensibilidad	-100 dBm (XBee-PRO (S2))
Área de cobertura	100×125 m ²
Parámetros de nodos	
Distribución de Sensores	Aleatorio
N° Coordinador	1
N° de nodos fijos	2, 3, 4.....30, 31, 33, 35, 37 y 39
Tipo de tráfico	CBR (<i>Constant Bit Rate</i>)
Tamaño de Paquete	70 bytes

Cada una de las simulaciones está programada para que la red tenga un tiempo igual a 20 segundos de encendido que son suficientes para que todos los dispositivos de la WSN se inicialicen, pues en 0.0 segundos es encendido el nodo cero que actúa como *Gateway* (Sumidero) y nodo coordinador de toda la información recolectada por los sensores del área de monitorización, luego cada uno de los nodos se enciende secuencialmente uno tras otro en intervalos de tiempo de 0.5 segundos como se aprecia en la Figura 3.3. Conjunto al encendido de los dispositivos se tiene la configuración para habilitar o no el modo *Beacon* de la red Zigbee/IEEE 802.15.4, en nuestro caso se habilita este modo ya que tiene la disponibilidad del *guaranteed time slot (GTS)* y de sincronizar los relojes de los nodos en la red. La implementación de la estructura de super trama que es definida por el coordinador PAN y configurada por la capa de red.

Tabla. 3.5.

Parámetros generales de simulación ns-2 en línea de comandos (Escenario de 16 Nodos)

Configuración de Parámetros de Transmisión		
Modelo de radio propagación	set val(prop)	Propagation/TwoRayGround
Protocolo de Enrutamiento	set val(rp)	AODV
Tecnología Inalámbrica	set val(netif) set val(mac)	Phy/WirelessPhy/802_15_4 Mac/802_15_4
Tipo de Canal	set val(chan)	Channel/WirelessChannel
Antena	set val(ant)	Antenna/OmniAntenna
Área de cobertura	set val(x) set val(y)	100 125
Tasa de transmisión	Mac/802_15_4	set dataRate_ 250kb
Frecuencia de Operación	Phy/WirelessPhy	set freq_ 2.4e+09
Potencia de Transmisión	Phy/WirelessPhy	set Pt_ 0.05
Sensibilidad	Phy/WirelessPhy Phy/WirelessPhy Phy/WirelessPhy	set CStresh_ 1.0e-13 set RXThresh_ 1.0e-13 set CPThresh_ 10
Pérdidas Por Trayectoria	Phy/WirelessPhy	set L_ 1.0
Ganancia de antena	Antenna/OmniAntenna Antenna/OmniAntenna	set Gt_ -0.05 set Gr_ -0.05
Tiempo simulado	set stopTime	140.0
Tiempo de envío de tráfico	set appTime3	20.00212
Configuración de Parámetros de Nodos		
Nº de nodos fijos	set val(nn)	16
Tipo de tráfico	set val(traffic)	cbr
Tamaño de Paquete	eval \\${cbr_}(\$src)	set packetSize_ 70

Cada una de las simulaciones está programada para que la red tenga un tiempo igual a 20 segundos de encendido que son suficientes para que todos los dispositivos de la WSN se inicialicen, pues en 0.0 segundos es encendido el nodo cero que actúa como *Gateway* (Sumidero) y nodo coordinador de toda la información recolectada por los sensores del área de monitorización, luego cada uno de los nodos se enciende secuencialmente uno tras otro en intervalos de tiempo de 0.5 segundos como se aprecia en la Figura 3.3. Conjunto al encendido de los dispositivos se tiene la configuración para habilitar o no el modo *Beacon*

de la red Zigbee/IEEE 802.15.4, en nuestro caso se habilita este modo ya que tiene la disponibilidad del *guaranteed time slot* (GTS) y de sincronizar los relojes de los nodos en la red. La implementación de la estructura de super trama que es definida por el coordinador PAN y configurada por la capa de red.

El coordinador PAN puede, opcionalmente limitar su tiempo de canal con la estructura de súper trama. La súper trama está limitada por la transmisión de una trama *beacon* y puede tener una parte activa y una parte inactiva. El coordinador debe interactuar con su PAN sólo durante la parte activa de la súper trama y, por tanto, puede entrar en un bajo consumo de energía (reposo) durante la parte inactiva.

La duración entre balizas, también conocida como *BI* (*Beacon Interval*) es determinada con el *BO* (*Beacon Order*) y la constante *aBaseSuperframeDuration*, relacionados mediante la siguiente ecuación [51]:

$$BI = aBaseSuperframeDuration \times 2^{BO} \quad (3.2)$$

donde *BO* toma valores entre 0 y 14 para funcionamiento balizado o *Beacon* habilitado, y 15 para funcionamiento no balizado. Análogamente, la longitud del período activo de la super trama, conocida como *SD* (*Superframe Duration*), es determinada con el *SO* (*Superframe Order*), relacionados como sigue:

$$SD = aBaseSuperframeDuration \times 2^{SO} \quad (3.3)$$

donde *SD* se mide en símbolos y su longitud no puede exceder de la del *BI*, por lo que el valor del *SO* siempre debe ser menor o igual que el de *BO* [51].

En *ns-2* el modo *Beacon* habilitado permite la utilización en su mayoría para aplicaciones con topología de malla o tipo árbol, con el objetivo de sincronizar sus operaciones y coordinar las comunicaciones de cada uno. También sirve para que los dispositivos se despierten de su modo de adormecimiento. Por lo tanto, en el desarrollo de las simulaciones se programó para que se designe por defecto los valores de *BO* y *SO* igual a 3.


```

# Encendido de Coordinador y nodos sensores

$ns_ at 0.0 "$node_(0) NodeLabel Coordinador"
$ns_ at 0.0 "$node_(0) sscs startPANCoord" ;# startPANCoord <txBeacon=1> <B0=3> <S0=3>
$ns_ at 0.5 "$node_(1) sscs startDevice 1 0" ;# startDevice <isFFD=1> <assoPermit=1>
$ns_ at 1.0 "$node_(2) sscs startDevice 1 0" <txBeacon=0> <B0=3> <S0=3>
$ns_ at 1.5 "$node_(3) sscs startDevice 1 0"
$ns_ at 2.0 "$node_(4) sscs startDevice 1 0"
$ns_ at 2.5 "$node_(5) sscs startDevice 1 0"
$ns_ at 3.0 "$node_(6) sscs startDevice 1 0"
$ns_ at 3.5 "$node_(7) sscs startDevice 1 0"

```

Figura. 3.3. Tiempo de encendido de dispositivos de red (Escenario de 8 Nodos)

A medida que se incrementa el número de nodos, estos van cumpliendo diferentes papeles dentro de la red, tal como dispositivos finales (RFD) o coordinador (FFD), es importante para su respectivo análisis identificar qué rol desempeña cada uno de estos elementos, por lo tanto *ns-2* establece un color específico para cada tipo de dispositivo dentro de la red ZigBee, en la Figura 3.4 se observa que cuando el nodo coordinador PAN se activa se pinta de color rojo y de verde los dispositivos finales.

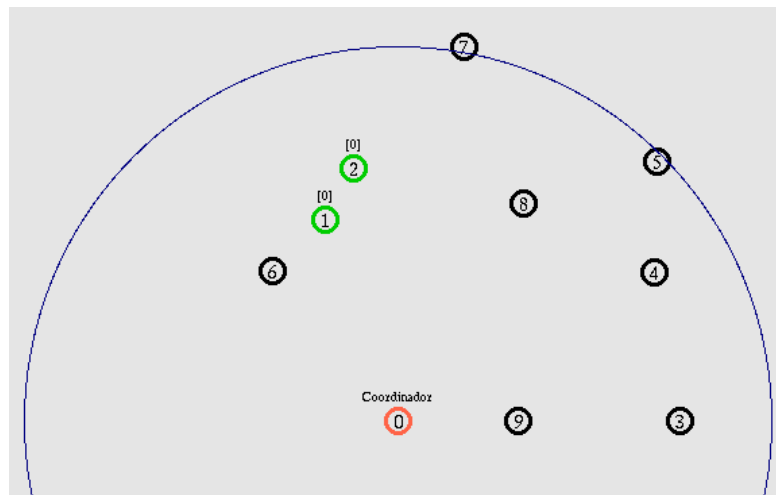


Figura. 3.4. Dispositivos de la red (Escenario de 10 Nodos)

Para establecer comunicación entre todos los elementos de la WSN, el nodo coordinador (nodo 0) realiza un *Broadcast* hacia todos los nodos esclavos, con este paso se logra despertar en caso de que se encuentren adormecidos y posterior establecer una comunicación inalámbrica entre dispositivos finales y coordinador, seguido a este proceso se plantea un evento sísmico que se simula dentro de un área definida de $100 \times 125 \text{ m}^2$,

donde las ondas sísmicas S y P se propagan a la velocidad de 6.70 y 3.92 km/seg hacia la WSN, en la Figura 3.5 se ilustra el evento sísmico propuesto para este proyecto, el cual se simula bajo los parámetros ya presentados en las Tabla 3.2 y 3.4.

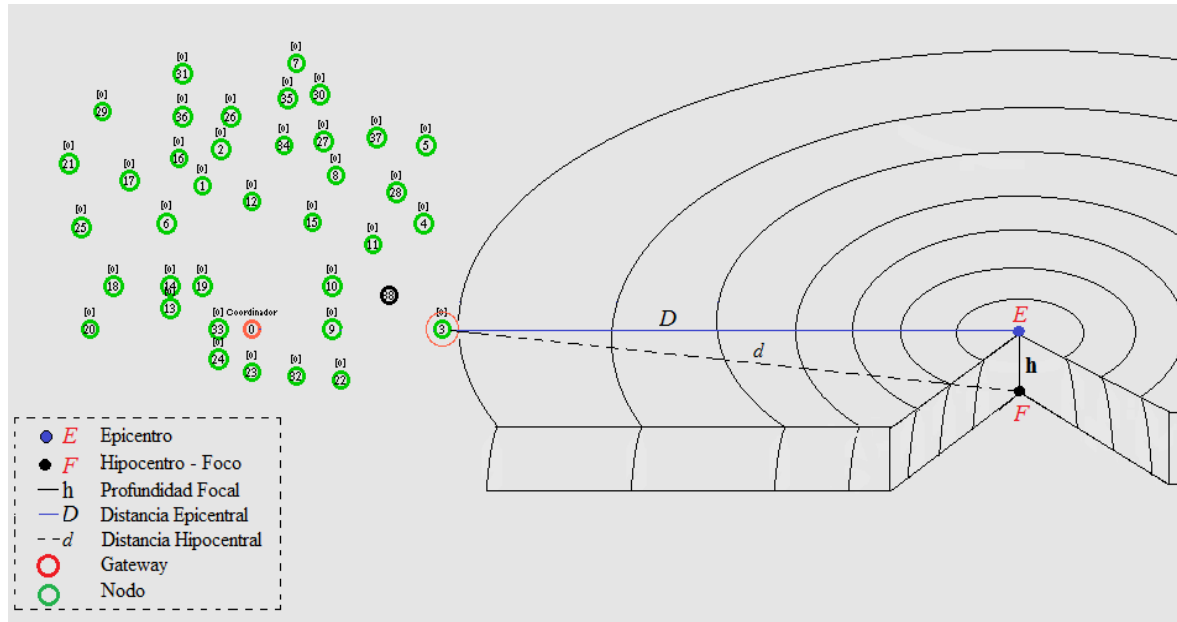


Figura. 3.5. Propagación de evento sísmico hacia la red WSN

Las ondas sísmicas se propagan a distintas direcciones hasta alcanzar uno o varios nodos, en la Figura 3.5 se observa que el primer nodo impactado por las ondas sísmicas es el tres, el cual inmediatamente solicita al coordinador o vecinos crear una comunicación hacia la *Gateway*, establecida la comunicación éste genera tráfico de tipo CBR sobre la red hacia el Coordinador, el tiempo a la que se transmite dicho tráfico ó información se programa a un tiempo igual a 20 segundos (tiempo de encendido de elementos de la WSN) más *Tsp* que es el tiempo de arribo de las ondas P y S al nodo, detallado en la Tabla 3.2.

Hay que tomar en cuenta que el tiempo de transmisión de información de nodo - coordinador es diferente para cada caso, ya que las ondas sísmicas impactan a diferentes tiempos a cada uno de los nodos, sin embargo, la diferencia no sobrepasa los milisegundos, es decir todos los dispositivos finales recolectan información de las vibraciones que producen las señales sísmicas sobre la superficie e inician la transmisión dentro de los 20 segundos de simulación, esto se puede observar en la interfaz gráfica de la herramienta NAM de *ns-2* en el panel inferior de envío de tráfico mostrada en la Figura 3.6.

La información transmitida por la red virtual simulada puede contener paquetes de un tamaño menor a 127 Bytes que es lo establecido por el estándar IEEE 802.15.4, no obstante, dentro de las características de funcionamiento de los Módulos XBee-PRO (S2) descritas en [41] cada transmisión *unicast* puede soportar hasta 84 bytes de carga útil de RF, por tal motivo se escoge un tamaño de paquetes igual a 70 bytes, pues con ayuda del protocolo de enrutamiento AODV éstos buscarán el mejor camino, es decir el más corto, se nota claramente la transferencia de paquetes a través de los nodos en la siguiente Figura.

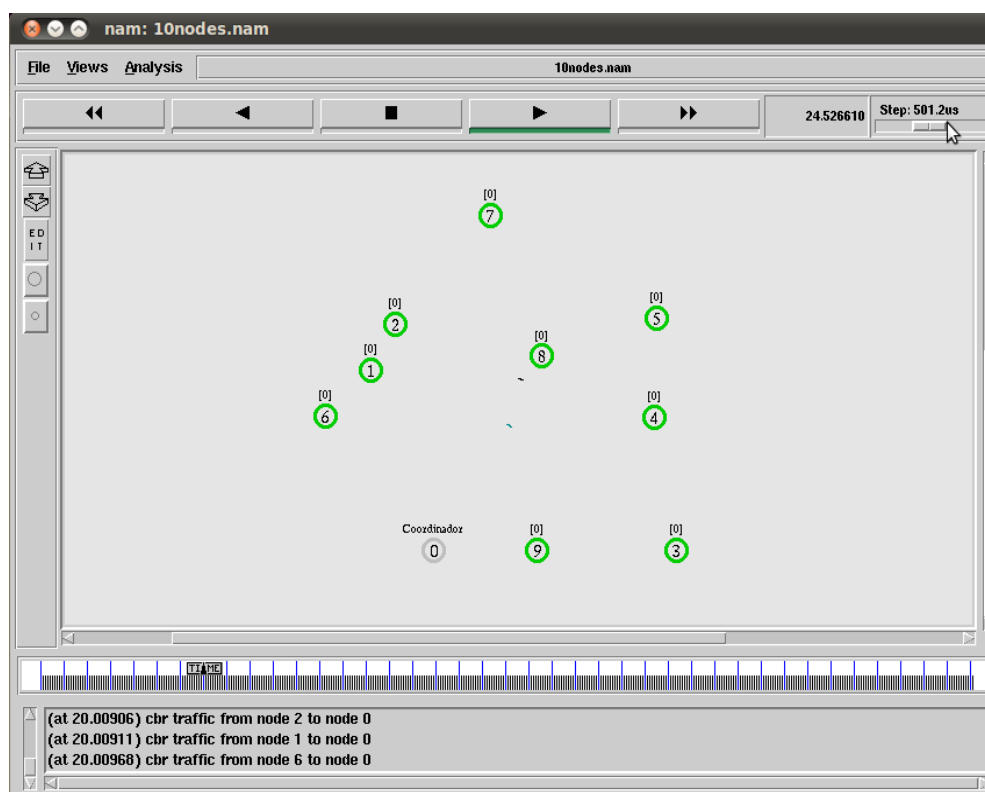
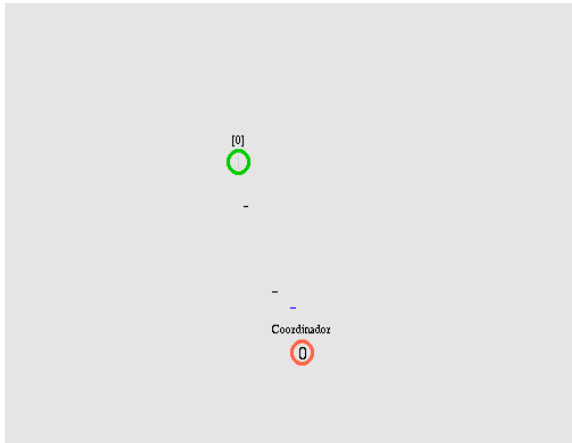
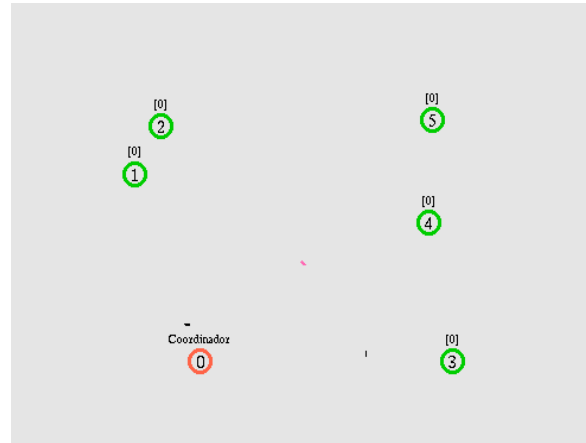


Figura. 3.6. Transmisión de paquetes Nodo 8 – Coordinador, Escenario 10 nodos, ns-2

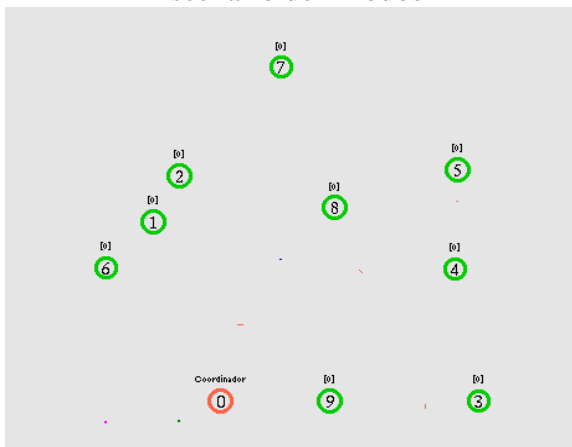
El tiempo de simulación total fue fijado a 140 segundos, pero si restamos los 20 segundos de encendido de los sensores tenemos 120 que equivale a 2 minutos de monitorización, que es un tiempo de muestreo suficiente para obtener la información necesaria para iniciar un posterior análisis, en la Figura 3.7 se muestra la transmisión de paquetes simulados en varios de los escenarios.



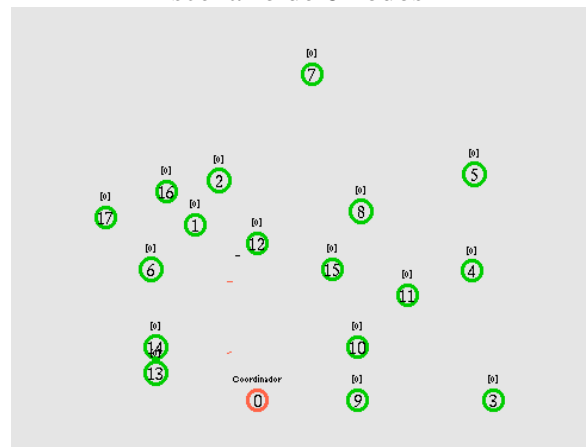
Escenario de 2 nodos



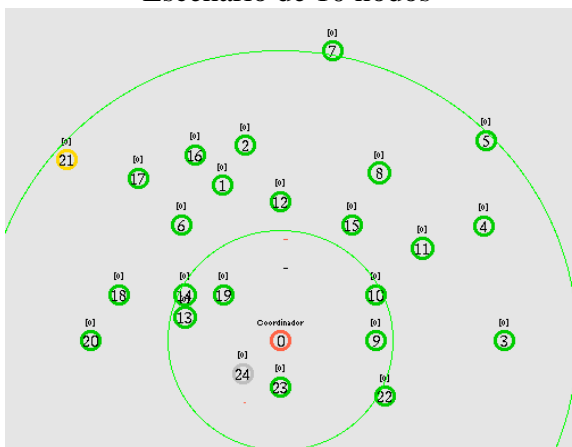
Escenario de 6 nodos



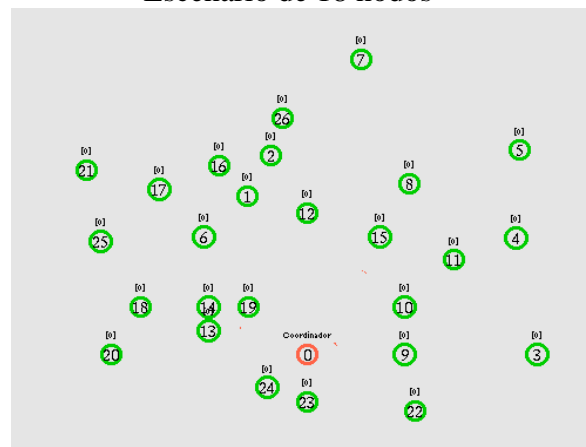
Escenario de 10 nodos



Escenario de 18 nodos



Escenario de 25 nodos



Escenario de 27 nodos

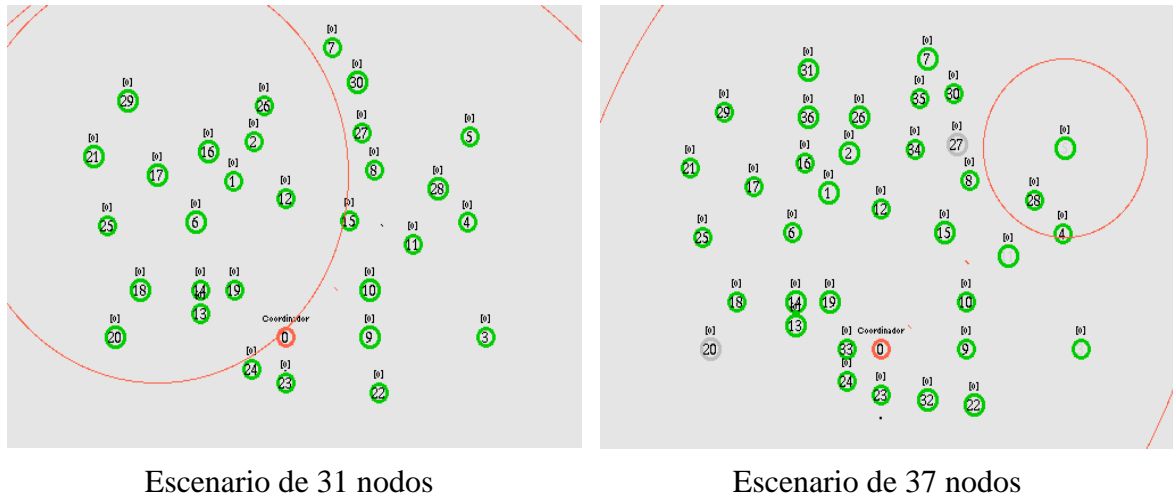


Figura. 3.7. Transmisión de paquetes, Simulación ns-2, Escenarios 2, 6, 10, 18, 25, 27, 31, y 37

3.3 TABULACIÓN DE RESULTADOS

Una vez realizadas las simulaciones de monitorización de señales sísmicas, el siguiente paso es analizar los resultados obtenidos, para lo cual se utiliza el *software* libre *Tracegraph205 Windows* que en conjunto a la herramienta MATLAB permiten graficar y especialmente analizar las trazas (*.tr) generadas por el simulador de redes ns-2, para facilitar la interpretación por *TraceGraph* las trazas son adecuadas a un nuevo formato por el convertidor de archivos *Trace Converter*.

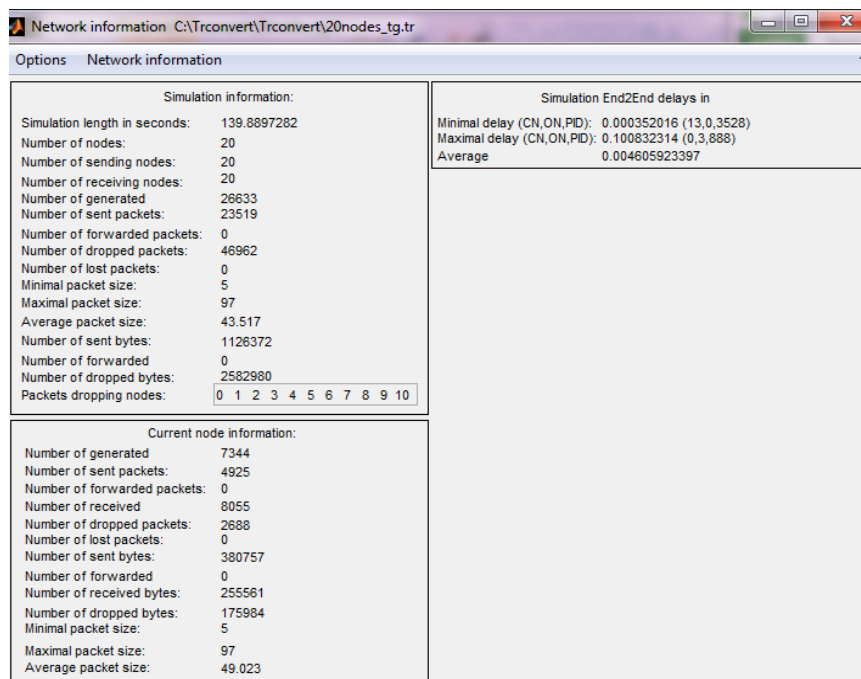


Figura. 3.8. Herramienta TraceGraph, Ventana Network information

La herramienta *TraceGraph* mediante las ventanas *Trace graph 2.05*, *Graphs* y *Network Information* permite obtener gráficas 2D o 3D de varios parámetros como el *Throughput*, *Delay*, *Jitter*, etc. Así como la impresión de datos estadísticos importantes de la red, se puede ilustrar en la Figura 3.8.

A partir de los archivos de trazas extraídos de las simulaciones y con la ayuda del *software* mencionado *TraceGraph*, obtenemos las gráficas del *throughput* versus tiempo de simulación generadas en unidades de bits por longitud de intervalo de tiempo del inglés (*Length of Time Interval* - TIL), correspondientes a los escenarios con topología física estrella, distribución de nodos randómico y parámetros de transmisión definidos en la Tabla 3.4, se muestran las curvas organizadas en la Figura 3.9.

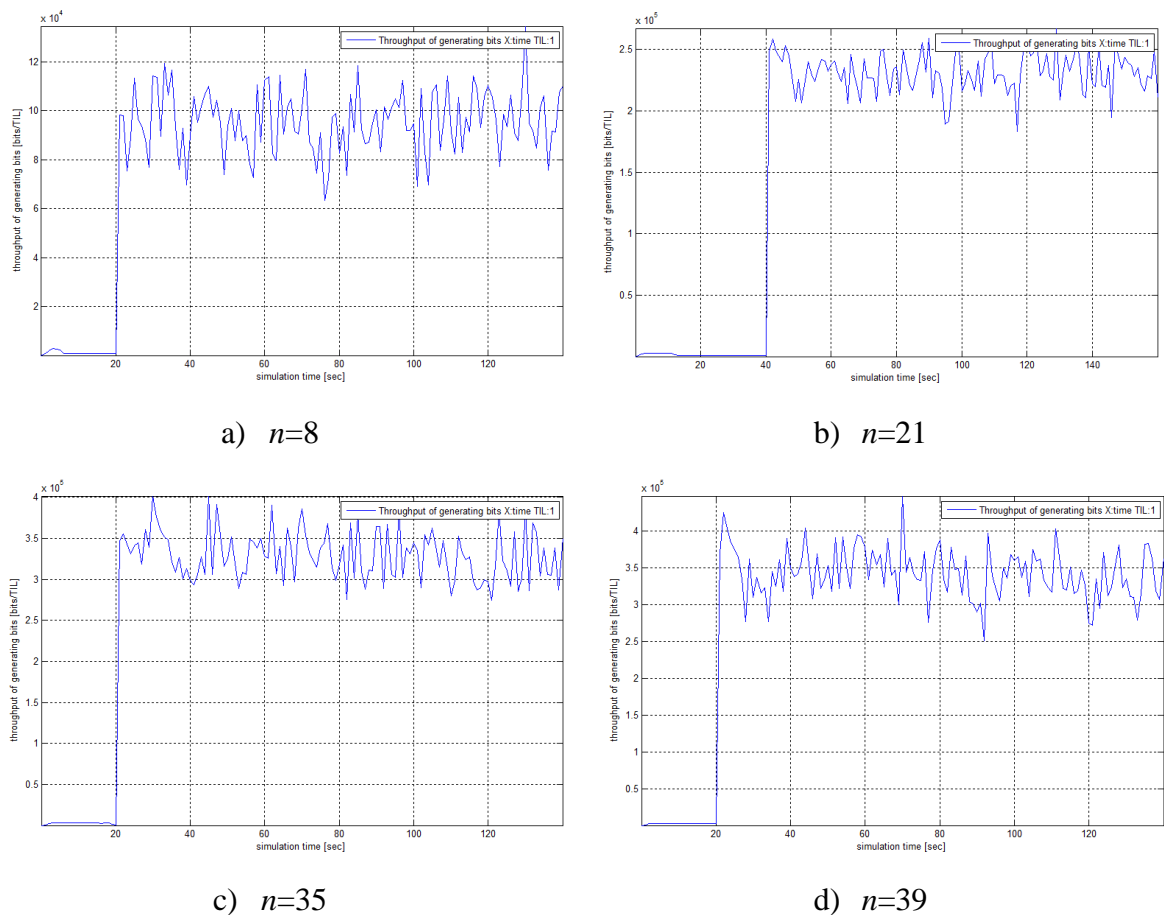


Figura. 3.9. Throughput de bits generados en unidades de bits/TIL, Escenarios: a) 8 nodos, b) 21 nodos, c) 35 nodos, y d) 39 nodos

En la Figura 3.9. a), se muestra que el rendimiento aumenta hasta un valor promedio de 9×10^4 bits / TIL, manteniéndose dentro de los 120 segundos en el rango de 7×10^4 a 12×10^4 bits / TIL. Es necesario tomar en cuenta que no todos los bits generados son enviados y es por esto que a pesar que las curvas muestran un rendimiento alto se puede observar en la Tabla 3.6 la cantidad de bytes caídos y perdidos, extraídos por medio de *TraceGraph* en la ventana *Network Information*.

Los valores del *Throughput* respecto a la densidad de nodos (n) mostrado son calculados con ayuda de la ecuación (2.1).

Tabla. 3.6.
Resultados de Simulaciones ns-2 (TraceGraph)

Resultados Simulados								
n	Tiempo Simulación	Paquetes			Bytes		Throughput [bits/s]	Retardo AVG (Seg)
		Generado	Enviado	Caídos	Enviados	Caídos		
2	119.89	2024	2022	2	60737	180	4052.91	0.002949
3	119.89	3046	3038	69	119207	6637	7954.73	0.004000
4	119.89	3990	3969	110	165765	9817	11061.45	0.003366
5	119.89	5701	5586	501	246210	38550	16429.48	0.004131
6	119.89	7418	7201	1019	322759	69457	21537.32	0.003977
7	119.89	9506	9178	2042	415121	131772	27701.12	0.004195
8	119.89	11207	10801	3275	499464	201022	33329.53	0.004461
9	119.89	13352	12764	5458	593605	327853	39610.87	0.004477
10	119.89	15155	14397	8293	674232	481840	44991.78	0.004742
11	119.89	15945	14998	9748	705596	564886	47083.58	0.004663
12	119.89	17535	16392	13046	778448	751634	51945.32	0.004749
13	119.89	19217	17852	16862	848784	959824	56638.95	0.004687
14	119.89	20104	18475	20098	876169	1135006	58466.34	0.004715
15	119.89	20961	19089	23521	911595	1320375	60830.46	0.004616
16	119.89	22648	20555	28656	984899	1600378	65721.31	0.005147
17	119.89	23202	20916	31571	1004855	1756660	67053.49	0.004795
18	119.89	24156	21519	37083	1027677	2045267	68577.49	0.004631
19	119.91	25026	22146	39699	1061960	2186146	70846.50	0.004828
20	119.89	26633	23519	46962	1126372	2582980	75161.86	0.004606
21	119.89	27176	23670	50588	1152419	2767452	76900.16	0.005221
22	119.91	26836	23014	55005	1093679	3000948	72963.15	0.004989
23	119.89	27318	23171	58038	1093150	3177529	72945.77	0.004808
24	119.89	28835	24349	65975	1159809	3580203	77393.91	0.004965

Continúa →

<i>n</i>	Tiempo Simulación	Paquetes			Bytes		Throughput [bits/s]	Retardo AVG (Seg)
		Generado	Enviado	Caídos	Enviados	Caídos		
25	119.89	28886	24022	70934	1114823	3848664	74392.20	0.004672
26	119.89	30651	25668	79506	1213942	4287573	81005.55	0.004820
27	119.89	30039	24495	81711	1149033	4407358	76673.81	0.005153
28	119.89	31864	26325	88327	1236141	4762971	82487.10	0.005015
29	119.89	31054	24219	97394	1104528	5242723	73704.23	0.005494
30	119.89	32018	25123	102940	1154145	5530161	77015.95	0.005081
31	119.89	32114	24619	110569	1104475	5938895	73700.10	0.005358
33	119.89	31248	22037	124132	917454	6690465	61221.10	0.006019
35	119.89	32671	23340	133086	981002	7134054	65461.79	0.005584
37	119.89	35056	24395	156629	1016275	8375522	67814.64	0.005868
39	119.88	33552	21648	165924	816715	8881920	54500.02	0.006560

Así mismo mediante la ecuación (2.2) se calcula el *Throughput* Normalizado respecto a la tasa de transmisión máxima de 250 kbps. Y para determinar el congestionamiento de la red hallamos el número de paquetes recibidos y enviados, finalmente se determina el parámetro *Delivery Ratio* que nos permite conocer el desempeño de la red respecto a los paquetes generados, se muestra en la Tabla 3.7.

$$Delivery\ Ratio = \frac{Número\ de\ Paquetes\ Recibidos}{Número\ de\ Paquetes\ Enviados} \times 100\% \quad (3.4)$$

Tabla. 3.7.

Resultados de Simulaciones (Calculados)

Resultados Simulados						
<i>n</i>	Paquetes		Throughput Normalizado	Complemento de Throughput Normalizado	Delivery Ratio (%)	Paquetes Perdidos (%)
	Recibidos	Perdidos				
2	2020	2	0.02	0.98	99.90	0.10
3	3030	8	0.03	0.97	99.74	0.26
4	3948	21	0.04	0.96	99.47	0.53
5	5471	115	0.07	0.93	97.94	2.06
6	6984	217	0.09	0.91	96.99	3.01
7	8850	328	0.11	0.89	96.43	3.57
8	10395	406	0.13	0.87	96.24	3.76
9	12176	588	0.16	0.84	95.39	4.61
10	13639	758	0.18	0.82	94.74	5.26
11	14051	947	0.19	0.81	93.69	6.31

Continúa \longrightarrow

<i>n</i>	Paquetes		Throughput Normalizado	Complemento de Throughput Normalizado	Delivery Ratio (%)	Paquetes Perdidos (%)
	Recibidos	Perdidos				
12	15249	1143	0.21	0.79	93.03	6.97
13	16487	1365	0.23	0.77	92.35	7.65
14	16846	1629	0.23	0.77	91.18	8.82
15	17217	1872	0.24	0.76	90.19	9.81
16	18462	2093	0.26	0.74	89.82	10.18
17	18630	2286	0.27	0.73	89.07	10.93
18	18882	2637	0.27	0.73	87.75	12.25
19	19266	2880	0.28	0.72	87.00	13.00
20	20405	3114	0.30	0.70	86.76	13.24
21	20164	3506	0.31	0.69	85.19	14.81
22	19192	3822	0.29	0.71	83.39	16.61
23	19024	4147	0.29	0.71	82.10	17.90
24	19863	4486	0.31	0.69	81.58	18.42
25	19158	4864	0.30	0.70	79.75	20.25
26	20685	4983	0.32	0.68	80.59	19.41
27	18951	5544	0.31	0.69	77.37	22.63
28	20786	5539	0.33	0.67	78.96	21.04
29	17384	6835	0.29	0.71	71.78	28.22
30	18228	6895	0.31	0.69	72.56	27.44
31	17124	7495	0.29	0.71	69.56	30.44
33	12826	9211	0.24	0.76	58.20	41.80
35	14009	9331	0.26	0.74	60.02	39.98
37	13734	10661	0.27	0.73	56.30	43.70
39	9744	11904	0.22	0.78	45.01	54.99

3.4 ANÁLISIS DE RESULTADOS

El primer parámetro que se analiza es el *Throughput*, el mismo que indica la cantidad de datos transmitidos en bytes dentro del periodo de simulación determinado que es de 120 segundos, de esta manera nos permite conocer el nivel de rendimiento de la red, que es igual a la capacidad que tiene la red para manipular toda la información que transita en ella.

Utilizando el *Toolbox Curve Fitting* de MATLAB[®] se logra obtener una función aproximada del *Throughput*, retardo y paquetes perdidos de la red versus el número de nodos, esta herramienta facilita el ajuste de curvas generando una función de acuerdo a

nuestras necesidades garantizando un límite de confianza del 95%, ésta puede ser un polinomio de grado n así como una función lineal todo depende del análisis que se realicen al momento de configurar los parámetros necesarios para seleccionar la función más adecuada, en la siguiente Figura se muestra la ventana principal de la herramienta donde se visualiza el resultado gráfico y algebraico.

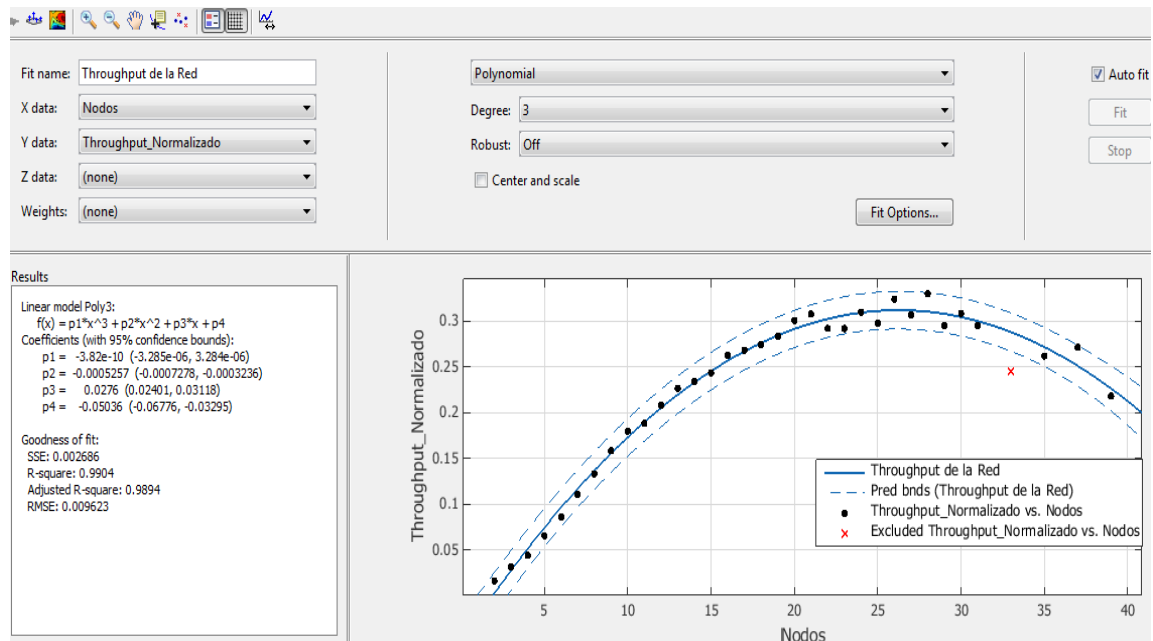


Figura. 3.10. Ventana principal Curve Fitting

Para hallar una función que relacione *Throughput* y el número de nodos se requiere ingresar los datos correspondientes a las dos métricas mencionadas en un archivo M-File de MATLAB para luego cargarlos desde la herramienta *Curve Fitting* como se muestra en la Figura 3.10, adicionalmente seleccionamos a que modelo de función se ajustarán los datos ingresados, en este caso se elige de tipo polinomio, sin embargo, es necesario realizar un análisis para definir cuál es el grado que se le asignará al polinomio.

El mejor ajuste de curva y en si el grado del polinomio seleccionado dentro de la herramienta *Curve Fitting* se determina mediante el método visual y analizando las estadísticas de bondad de ajuste de la curva, éstas se visualizan en el panel de resultados; la suma de errores cuadráticos (SSE), R-cuadrado, R-cuadrado Ajustado y el error cuadrático medio (RMSE).

La curva presentada en la Figura 3.11 a) corresponde a los parámetros ingresados en la Figura 3.10, aquí se ilustra un ajuste preciso del polinomio de tercer grado, incluso el trazado de residuos mostrado en la Figura 3.11 b) se acercan más a cero, no obstante en la Tabla 3.8 se realiza una comparación entre polinomios de diferente grado con el objetivo de analizar las estadísticas de bondad de ajuste y determinar finalmente el grado de la función.

Tabla. 3.8.

Estadísticas de bondad de ajuste

Datos de ajuste				
Estadísticas de bondad de ajuste	Grado de Polinomio	Throughput vs Nodos	Retardo vs Nodos	Paquetes Perdidos vs Nodos
La suma de errores cuadráticos (SSE)	2	0.002686	2.903e-06	6.214e+05
	3	0.002686	5.086e-07	5.509e+05
	4	0.002636	4.115e-07	1.119e+05
R-cuadrado Ajustado	2	0.9897	0.7459	0.9980
	3	0.9894	0.9536	0.9982
	4	0.9892	0.9609	0.9996
Error cuadrático medio (RMSE)	2	0.009462	0.0003408	143.9
	3	0.009623	0.0001456	137.8
	4	0.009702	0.0001338	65.61

La suma de errores cuadráticos según [52], mide la desviación total de los valores de respuesta a partir del ajuste a los valores resultantes. Cuando el valor es cercano a cero indica que el modelo tiene un componente de error aleatorio más pequeño, y que el ajuste será más útil para la predicción. Para los tres casos expuestos los valores que se acerca más a cero corresponden al polinomio de grado 4 con 0.002636, 4.115e-07 y 4.937e+05.

El R-cuadrado ajustado es la estadística que actúa como mejor indicador, éste puede tomar cualquier valor inferior o igual a 1, con un valor cercano a 1 indica un mejor ajuste. Los valores negativos pueden ocurrir cuando el modelo contiene términos que no ayudan a predecir la respuesta. Los valores cercanos a 1 en el caso de Retardo vs Nodos y Paquetes Perdidos vs Nodos recaen en el polinomio de cuarto grado con 0.9609 y 0.9983 respectivamente, mientras que para el caso de *Throughput* vs Nodos corresponde al

polinomio de grado 2 con un valor de R-cuadrado ajustado igual a 0.9897, esto equivale a un ajuste del 98.97% de la variación total en los datos sobre el promedio.

El error cuadrático medio (RMSE) se conoce también como el error estándar de ajuste o desviación cuadrática media, es una medida de uso frecuente de la diferencia entre los valores pronosticados por el modelo y los datos realmente obtenidos en las simulaciones realizadas en el capítulo anterior, además sirve para agregar dichas diferencias en una sola medida de poder predictivo y al igual que con SSE, un valor de RMSE más cerca de 0 indica un ajuste más útil para la predicción. El RMSE cercano a cero en el caso de *Throughput vs Nodos* es 0.009462 que corresponde al polinomio de grado 2 mientras que para los dos casos restantes pertenecen a un polinomio de grado 4.

Una vez realizado el análisis de los datos extraídos en la Tabla 3.8 es preciso determinar cuál es el grado del polinomio con el cual se trabajará a partir de este momento, por lo tanto para el primer caso que es *Throughput vs Nodos* se escoge un polinomio de segundo orden dado que dos de tres de las estadísticas indican que éste tipo de polinomio se ajusta de forma más cercana a los datos modelados. En el segundo y tercer caso se determina como polinomio idóneo al de cuarto orden porque de forma gráfica y numérica nos muestra que se ajusta de manera ideal a los datos de la red ingresados a la herramienta.

La Tabla 3.9 ilustra las funciones polinómicas que relacionan las métricas de calidad de red con respecto al número de nodos, las mismas son de segundo y cuarto orden.

Tabla. 3.9.

Funciones de métricas de la red en función del número de nodos

Función algebraica de una variable	
Throughput	$f(x) = -0.0005257x^2 + 0.0276x - 0.05036$ (3.5)
Retardo	$f(x) = -6.115e - 09x^4 + 7.733e - 07x^3 - 3.046e - 05x^2 + 0.0004865x + 0.00203$ (3.6)
Paquetes Perdidos	$f(x) = 0.00472x^4 - 0.3392x^3 + 15.03x^2 - 44.39x + 4.622$ (3.7)

A pesar que gráficamente la curva de la Figura 3.11 a) perteneciente al polinomio de grado 3 se ajusta de manera adecuada por sus residuos cercanos a cero, se demostró de

forma numérica que el polinomio de un grado menor se ajusta de forma más exacta. A continuación se analizan las gráficas resultantes de dichas funciones.

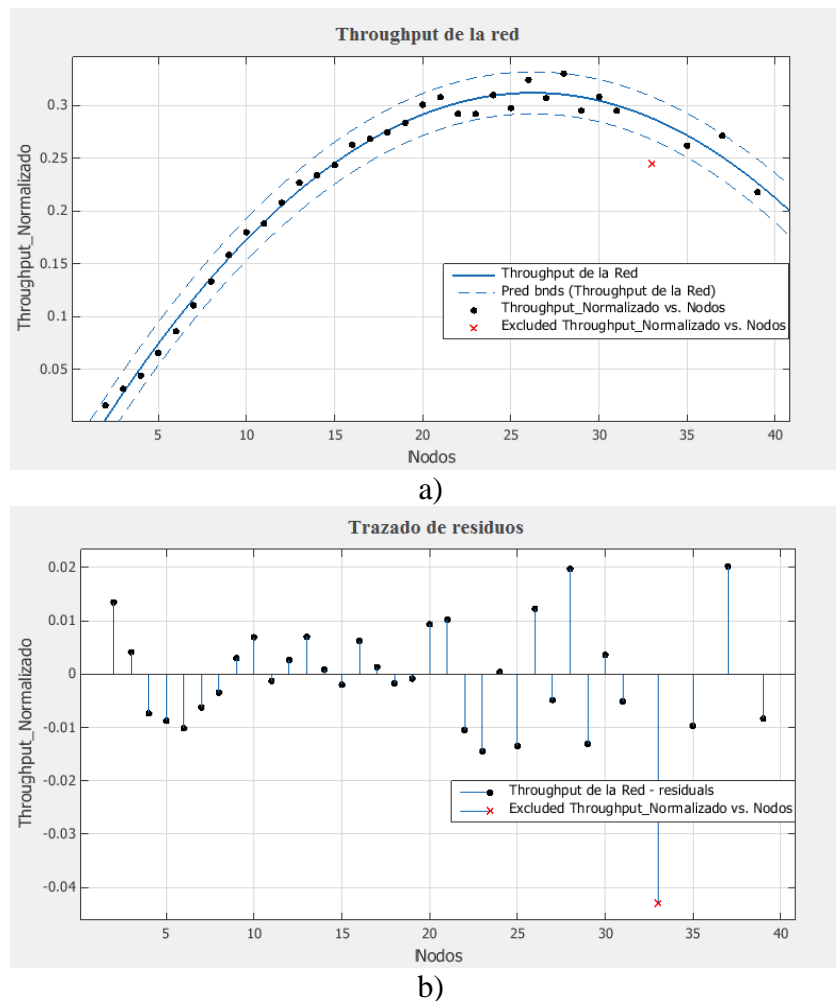


Figura. 3.11. Rendimiento de la red: a) Throughput de la red, b) Trazado de Residuos.

El comportamiento gráfico del *Throughput* en relación al número de nodos determina que el incremento del número de nodos también hace que tienda a elevarse la tasa de transmisión, sin embargo ésta alcanza un rango máximo entre 25 y 28 nodos, con un número de nodos mayor a este rango el canal se satura y tiende a disminuir notoriamente, esto se debe a que en la red existe solo un nodo coordinador encargado de recibir y almacenar la información, la cantidad de escenarios simulados es idónea para determinar de manera más acertada la tendencia que toma la curva de la tasa de transmisión de la red.

Con respecto al Retardo y Pérdida de Bytes notamos que los datos son directamente proporcionales al número de nodos ya que a mayor número de nodos existe más cantidad de caída de bytes y por ende aumenta el retardo de la red.

En la Figura 3.12 se ilustra las curvas del retardo y paquetes perdidos de la red, además se observa los límites de confianza del 95% trazados con líneas entre cortada, pues representa que los coeficientes generados por el polinomio de cuarto orden tienen ese porcentaje de confianza, se descartó los puntos que no se encuentran dentro de los límites, con esto se consigue que la función resultante se ajuste mejor a los datos modelados.

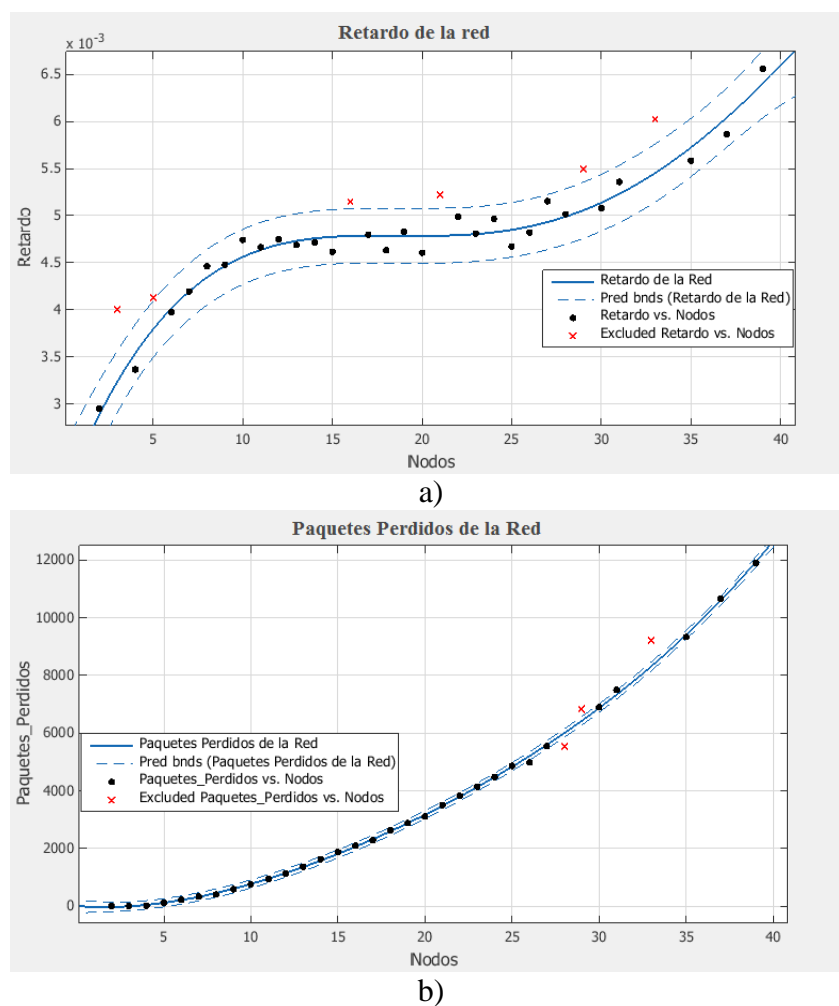


Figura. 3.12. Métricas de calidad de la red a) Retardo, b) Paquetes Perdidos

Es necesario recalcar que el objetivo principal de este trabajo es encontrar un punto o rango óptimo de nodos donde el *Throughput* sea el máximo y el retardo el mínimo, por lo tanto se requiere encontrar una función que relacione tres variables; el número de nodos y

las dos ya mencionas, para ello se aplica el mismo procedimiento anterior que se utilizó para encontrar la función objetivo en una sola variable pero en este caso aplicado a tres.

Respecto al grado de los polinomios evaluados se realiza un análisis de las estadísticas de bondad y método visual, se determina que el polinomio que mejor se ajusta corresponde al de cuarto grado, a continuación se muestra la expresión.

$$f(x,y) = 13.66 - 8.177x + 4.556y + 0.9576x^2 - 0.9681xy + 10.81y^2 + 1.555x^3 + 1.533x^2y - 2.91xy^2 - 6.116y^3 + 4.086x^4 + 9.492x^3y + 1.08x^2y^2 + 0.9793xy^3 + 1.08y^4 \quad (3.8)$$

Donde x es el complemento del *Throughput* normalizado, y es el retardo de la red y $f(x,y)$ corresponde a la función objetivo que evaluada en el punto (x,y) proporciona un escalar equivalente al número de nodos óptimo.

En la siguiente Figura se ilustra la superficie que representa la función perteneciente a la ecuación (3.8), se puede notar claramente los puntos marcados sobre la superficie que van graficándose de forma creciente, adicionalmente es preciso acentuar que los datos del *Throughput* ingresados al *Curve Fitting* para este caso son el complemento de los datos originales.

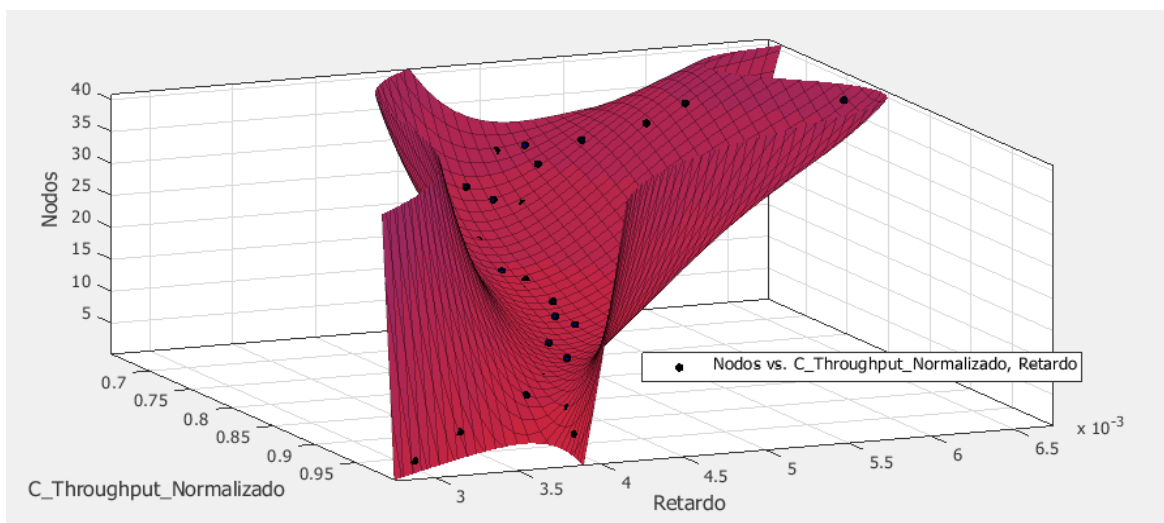


Figura. 3.13. Superficie Nodos vs Throughput vs Retardo de la red

La superficie que se muestra en la Figura 3.13 tiende a crecer ya que el aumento del retardo y tasa de transmisión de toda la red hace que se incrementen de igual forma los paquetes perdidos, ya que a partir del número de nodos 28 la pérdida de paquetes sobrepasa el 25% de lo permitido por el Geofísico de la Escuela Politécnica Nacional, por lo mencionado se toma como restricción que el rango de nodos al igual que las otras variables se incrementen hasta alcanzar una pérdida de paquetes máxima del 25%, si verificamos en la Tabla 3.7 los datos de paquetes perdidos extraídos de las simulaciones encontramos que el porcentaje máximo permitido llega hasta el número de nodos 28.

La restricción encontrada de los paquetes perdidos ayuda a minimizar el área de la superficie es decir el dominio x, y donde se requiere encontrar el punto mínimo absoluto que maximice la función objetivo.

Se define la restricción de la siguiente manera,

$$f(x, y) \leq 28 \quad (3.9)$$

Por lo tanto,

$$\begin{aligned} -8.177x + 4.556y + 0.9576x^2 - 0.9681xy + 10.81y^2 + 1.555x^3 + 1.533x^2y - \\ 2.91xy^2 - 6.116y^3 + 4.086x^4 + 9.492x^3y + 1.08x^2y^2 + 0.9793xy^3 + \\ 1.08y^4 \leq 14.34 \end{aligned} \quad (3.10)$$

La inecuación (3.10) no lineal queda planteada y será utilizada en el siguiente capítulo como parte de las restricciones a las que está sujeta la función (3.8).

CAPÍTULO 4

OPTIMIZACIÓN DEL NÚMERO DE SENSORES INALÁMBRICOS

4.1 APLICACIÓN DE TÉCNICAS DE OPTIMIZACIÓN

La optimización de una función objetivo de una o varias variables se puede llevar a cabo utilizando diferentes herramientas, en nuestro caso se hará uso del *Toolbox* de optimización de MATLAB®, es el más usado en la mayoría de aplicaciones relacionados a la optimización de funciones, además provee algoritmos para resolver diferentes tipos de problemas de optimización, con restricciones o sin ellas; los cuales incluyen programación lineal, programación cuadrática, minimización (maximización) de funciones de una o varias variables, solución de sistemas de ecuaciones no lineales, entre otros.

El *toolbox* de optimización de MATLAB® es un conjunto de funciones (almacenadas en m-files) basados en algoritmos matemáticos que posee una interfaz gráfica que facilita a los usuarios menos experimentados hacer uso de esta herramienta, sin necesidad de hacer uso de código. Hay que tomar en cuenta que MATLAB siempre busca minimizar funciones, por tanto si se requiere resolver un problema de maximización es necesario multiplicar la función objetivo por -1 , de esta forma, los valores obtenidos en la minimización de la nueva función son los equivalentes a los de la maximización de la función objetivo original [53].

La capacidad y el alcance del *toolbox* de Optimización se pueden ampliar combinando éste con otros *toolboxes*, o creando ficheros. En las Tablas 4.1 se muestra una descripción de todas las funciones disponibles en MATLAB en relación a la búsqueda de un mínimo local.

Tabla. 4.1.

Funciones de minimización [54]

Tipo y descripción	Notación	Función
Minimización escalar: Encuentra el mínimo de una función de una variable	$\text{Min}_a f(a)$ Para $a_1 < a < a_2$	fminbnd
Minimización sin restricciones: Encuentra el mínimo de una función de varias variables sin restricciones.	$\text{Min}_x f(x)$ Donde x es un vector	fminunc, fminsearch
Programación lineal: Resuelve un problema de programación lineal.	$\text{Min}_x f^T(x)$ Sujeto a $A \cdot x = b, Aeq \cdot x = beq, lb = x = ub$ A, Aeq son matrices, f, x, b, beq, lb, ub son vectores.	linprog
Programación cuadrática: Resuelve un problema de programación cuadrática como el que se especifica.	$\text{Min}_x \frac{1}{2} x^T H x + f^T(x)$ Sujeto a $A \cdot x = b, Aeq \cdot x = beq, lb = x = ub$ H, A, Aeq son matrices, f, x, b, beq, lb, ub son vectores.	quadprog
Minimización con restricciones: Encuentra el mínimo de una función no lineal de varias variables con restricciones.	$\text{Min}_x f(x)$ Sujeto a $c(x) = 0, ceq(x) = 0, A \cdot x = b,$ $Aeq \cdot x = beq, lb = x = ub$ A, Aeq son matrices, f, x, b, beq, lb, ub son vectores, $f(x), c(x)$ y $ceq(x)$ pueden ser funciones no lineales que devuelven un vector.	fmincon
Problema de mínimo-máximo: Resuelve un problema de mínimo y máximo	$\text{Min}_x \max_{\{F_i\}} \{F_i(x)\}$ Sujeto a $c(x) = 0, ceq(x) = 0, A \cdot x = b,$ $Aeq \cdot x = beq, lb = x = ub$ A, Aeq son matrices, x, b, beq, lb, ub son vectores, $F(x), c(x)$ y $ceq(x)$ pueden ser funciones no lineales que devuelven un vector.	fminimax
Minimización semi-infinita: Encuentra el mínimo de una función de varias variables con restricciones semi-infinitas.	$\text{Min}_x f(x)$ Sujeto a $A \cdot x = b, Aeq \cdot x = beq, lb = x = ub$ tal que $K1(x, w1) = 0;$ $K2(x, w2)0; \dots; Kn(x, wn) = 0$ A, Aeq son matrices. f, x, b, beq, lb, ub son vectores. $c(x)$ y $ceq(x), Ki(x, wi)$ son funciones que devuelven un vector. $f(x), c(x), ceq(x)$ pueden ser funciones no lineales. Los vectores (o matrices) $Ki(x, wi) \leq 0$ son funciones continuas tanto en x como en wi . Las variables wi son vectores de dimensión dos.	fseminf

Recordemos que, en el *Toolbox* de optimización de MATLAB® se hace una clasificación de los algoritmos de optimización internos que se usan para ejecutar esta herramienta y que las funciones de optimización no son más que rutinas con extensión .m

que se llevan a cabo mediante un conjunto de sentencias y algoritmos de optimización específicos, la aplicación de los algoritmos elegidos para este proyecto depende exclusivamente de las restricciones a las que esté sujeta F ya sean lineales o no. Para un correcto uso de las rutinas de minimización “fminbnd”, “fminunc”, “fmincon”, etc. en cualquier programación que se desarrolle, es importante conocer qué argumentos se reciben y devuelven, así como el significado de cada uno de estos parámetros con los que las funciones trabajan. A continuación hay detalles importantes de la sintaxis que usan las funciones, que son empleadas en este proyecto, empezando por la función o “fminbnd”.

Tabla. 4.2.

Función fminbnd, MATLAB® [55]

Sintaxis “fminbnd”
x = fminbnd(fun,x1,x2) x = fminbnd(fun,x1,x2) devuelve un valor x que es un minimizador local de la función que se describe en fun en el intervalo $x_1 < x < x_2$.
x = fminbnd(fun,x1,x2,options) x = fminbnd(fun,x1,x2,options) minimiza con los parámetros de optimización especificado en la estructura <i>options</i> . Se puede definir estos parámetros utilizando la función optimset. “fminbnd” utiliza <i>options</i> para ingresar campos con estructura: <ul style="list-style-type: none"> • Display. - Nivel de visualización 'off ' o 'none' muestra ninguna salida; 'iter' muestra la salida en cada iteración; 'final' muestra sólo el resultado final; 'notify' (por defecto) muestra la salida sólo si la función no converge. • MaxFunEvals.- Número máximo de evaluaciones de la función, un entero positivo. El valor predeterminado es 500. • MaxFunEvals - Número máximo de evaluaciones de la función permitidos. • MaxIter - Número máximo de iteraciones permitidas. • TolX - tolerancia Terminación de x.
x = fminbnd(problem) encuentra el mínimo de <i>problem</i> , cuando <i>problem</i> es una estructura con los argumentos de entrada : <ul style="list-style-type: none"> • Objective - Función Objetiva • x1- Punto final izquierdo • x2- Punto final derecho • solver- 'fminbnd' • options- Estructura <i>Options</i> creado usando optimset.
[x,fval] = fminbnd(...) [x,fval] = fminbnd(...) devuelve el valor de la función objetivo calculado fun en x .
[x,fval,exitflag] = fminbnd(...) [x,fval,exitflag] = fminbnd(...) devuelve un valor exitflag que describe la condición de salida de fminbnd.

Continúa →

Sintaxis “fminbnd”

[x,fval,exitflag,output] = fminbnd(...) **[x,fval,exitflag,output] = fminbnd(...)** devuelve una estructura output que contiene información acerca de la optimización:

- output.algorithm - El algoritmo utilizado.
- output.funcCount - El número de evaluaciones de la función.
- output.iterations - El número de iteraciones realizadas.

La función “fminunc” intenta encontrar un mínimo de una función escalar de varias variables que está libre de condiciones, a partir de una estimación inicial. Esto se conoce generalmente como optimización no lineal sin restricciones [56]. La sintaxis para la utilización de este algoritmo está definida como se muestra en la siguiente Tabla 4.5.

Tabla. 4.3.

Función fminunc, MATLAB® [55]

Sintaxis “fminunc”

x = fminunc(fun,x0) comienza en el punto x0 e intenta encontrar un mínimo local x de la función descrita en fun . x0 puede ser un escalar, vector o matriz.

x = fminunc(fun,x0,options) minimiza con las opciones de optimización especificados en options . Se usa la sentencia optimset para establecer estas opciones.

x = fminunc(problem) considera el mínimo para problem , cuando problem es una estructura descrita en los argumentos de entrada . Se crea la estructura mediante la exportación de un problema de aplicación de optimización.

[x,fval]=fminunc(...) **[x,fval]=fminunc(...)** vuelve en fval el valor de la función objetivo fun en la solución x .

[x,fval,exitflag]=fminunc(...) **[x,fval,exitflag]=fminunc(...)** devuelve un valor exitflag que describe la condición de salida.

[x,fval,exitflag,output]=fminunc(...) **[x,fval,exitflag,output]=fminunc(...)** devuelve una estructura output que contiene información acerca de la optimización.

[x,fval,exitflag,output,grad]=fminunc(...) **[x,fval,exitflag,output,grad] = fminunc(...)** vuelve en grad el valor del gradiente de fun en la solución x .

[x,fval,exitflag,output,grad,hessian]=fminunc(...) **[x,fval,exitflag,output,grad,hessian] = fminunc(...)** devuelve en hessian el valor de la matriz Hessiana de la función objetivo fun en la solución x .

El propósito de la función “fmincon” es encontrar el mínimo ó máximo del valor escalar que devuelve una función de varias variables no lineal sujeta a una serie de restricciones. Normalmente se conoce este tipo de problemas como optimización no lineal con restricciones o programación no lineal. Se debe implementar la función objetivo en un M-file para que se realicen los cálculos oportunos para obtener el valor escalar que

devuelve la función. Se puede describir de forma breve la sintaxis para el manejo los parámetros de entrada de la función de minimización en la Tabla 4.6.

Tabla. 4.4.

Función fmincon, MATLAB® [55]

Sintaxis “fmincon”
x = fmincon (fun,x0,A,b) empieza con los valores especificados por el vector de inicio x0, y encuentra un valor mínimo del vector x para la función que ha sido descrita en fun, y tal que se cumpla la inecuación lineal $A \cdot x = b$.
x = fmincon (fun,x0,A,b,Aeq,beq) encuentra el mínimo de la función fun, sujeta a la condición de igualdad $Aeq \cdot x = beq$, además de la inecuación $A \cdot x = b$. Cuando no existan condiciones de desigualdad, se debe establecer las variables $A=[]$, y $b=[]$.
x = fmincon(fun,x0,A,b,Aeq,beq,lb,lu) encuentra el mínimo de la función fun, sujeta a las condiciones $A \cdot x = b$ y $Aeq \cdot x < beq$. Ahora se definen unos límites inferiores y superiores para los valores de las variables de diseño, el vector x, así que la solución que halla la rutina estará comprendida en el rango $lb = x = lu$. Cuando no se usen las restricciones de igualdad de la función fun habrá que establecer $Aeq=[]$, y $beq=[]$.
x = fmincon(fun,x0,A,b,Aeq,beq,lb,lu,nonlcon) encuentra el mínimo de la función fun, para que se cumplan las anteriores restricciones y además sujeta la minimización de las inecuaciones no lineales $c(x)$ o $ceq(x)$ definidas en la función nonlcon. fmincon minimiza tal que se cumpla que $c(x)=0$ y $ceq(x)=0$. Cuando no se usen límites inferiores y/o superiores para los valores del vector de parámetros, habrá que establecer $lb=[]$ y $lu=[]$.
x = fmincon(fun,x0,A,b,Aeq,beq,lb,lu,nonlcon,options) minimiza la función objetivo, teniendo en cuenta los parámetros de optimización especificados en la estructura <i>options</i> . Si no se usan las restricciones del tipo $c(x)=0$ y $ceq(x)=0$, establecer $nonlcon=[]$.
x = fmincon(fun,x0,A,b,Aeq,beq,lb,lu,nonlcon,options,P1,P2,...) pasa parámetros adicionales a la función objetivo, sin necesidad de tener variables globales. Cuando los argumentos de la función $A,b,Aeq,beq,lb,lu,nonlcon$ y <i>options</i> , no sean necesarios, habrá que establecerlos como matrices vacías.
[x,fval] = fmincon (...) , devuelve el valor de la función objetivo fun, evaluada en x.
[x,fval,exitflag]=fmincon (...) , devuelve el valor de la función fun, evaluada en x y una bandera exitflag que indica la condición de salida o finalización de la función fmincon.
[x,fval,exitflag,output] = fmincon (...) , devuelve el valor de la función objetivo fun, evaluada en x, la condición de salida, exitflag, de la función de minimización, y una estructura llamada output que contiene información sobre la minimización llevada a cabo por la función fmincon.
[x,fval,exitflag,output,lambda] = fmincon (...) , devuelve los mismos valores anteriores y además una estructura llamada lambda que contiene los múltiplos Lagrangianos en el vector obtenido x.
[x,fval,exitflag,output,lambda,grad] = fmincon (...) , devuelve los mismos valores anteriores y además el valor del gradiente de la función fun en la solución obtenida x.
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon (...) , devuelve los mismos valores anteriores y además el valor del Hessian de la función fun en la solución obtenida x.

La estructura de optimización *options* contiene parámetros que pueden influir en la visualización de la solución que se obtenga en el proceso de minimización o maximización. En la Tabla 4.7 se describe las entradas que pueden tomar la estructura de la sentencia *options*.

Tabla. 4.5.

Parámetros de Options [55]

Entrada	Descripción
Algorithm	Elija el algoritmo de optimización: <ul style="list-style-type: none"> • 'interior-point' (por defecto) • 'trust-region-reflective' • 'SQP' • 'active-set'
DerivativeCheck	Comparar derivados suministrados por el usuario (gradientes de objetivo o limitaciones) para finitas-diferenciación derivados. Las opciones son 'on' o el defecto, 'off'.
Diagnostics	Muestra información de diagnóstico acerca de la función que se minimiza o resuelto. Las opciones son 'on' o el defecto, 'off'.
DiffMaxChange	Cambio máximo en variables para gradientes de diferencias finitas (un escalar positivo). El valor predeterminado es Inf.
DiffMinChange	Cambio mínimo en variables para gradientes de diferencias finitas (un escalar positivo). El valor predeterminado es 0.
Display	'off' o 'none' muestra ninguna salida. 'iter' salida en cada iteración, y da el mensaje de salida por defecto. 'iter-detailed' muestra la salida en cada iteración, y da el mensaje de salida técnica. 'notify' muestra la salida sólo si la función no converge, y da el mensaje de salida por defecto. 'notify-detailed' muestra la salida sólo si la función no converge, y da el mensaje de salida técnica. 'final' (por defecto) muestra sólo el resultado final, y da el mensaje de salida por defecto. 'final-detailed' muestra sólo el resultado final, y da el mensaje de salida técnica.
FinDiffRelStep	Escalar o factor de tamaño de paso vectorial. Cuando se establece FinDiffRelStep a un vector v.
FinDiffType	Diferencias finitas, utilizados para estimar los gradientes, son o bien 'forward' (por defecto), o 'central' (centrado). 'central' tiene el doble de evaluaciones de la función, pero debe ser más precisa.
FunValCheck	Compruebe si los valores de la función objetivo y las restricciones son válidos. 'on' muestra un error cuando la función o limitaciones objetivo devuelven un valor que es complejo, Inf, o NaN. El valor por defecto, 'off', no muestra ningún error.
GradConstr	Gradiente para las funciones de restricciones no lineales definidos por el usuario. Cuando se establece en 'on', fmincon espera que la función de limitación tenga cuatro salidas, como se describe en nonlcon en el Argumentos de entrada. El algoritmo La trust-region-reflective no acepta restricciones no lineales.
GradObj	Gradiente para la función de objetivo definido por el usuario.

Continúa →

Entrada	Descripción
Maxiter	Número máximo de iteraciones permitido, un entero positivo. El valor por defecto para todos los algoritmos es 400; para el algoritmo 'interior-point' por defecto es 1000.
OutputFcn	Especifica una o más funciones definidas por el usuario. El valor predeterminado es ninguno ([]).
PlotFcns	El valor predeterminado es ninguno ([]). @optimplotx grafica el punto actual @optimplotfunccount grafica la función count @optimplotfval grafica el valor de la función @optimplotconstrviolation traza la restricción de violación máxima @optimplotstepsize grafica el tamaño del paso @optimplotfirstorderopt grafica la medida optimalidad de primer orden
TolCon	Tolerancia en la violación de restricción, un escalar positivo. El valor predeterminado es 1e-6.
TolFun	Tolerancia de terminación en el valor de la función, un escalar positivo. El valor predeterminado es 1e-6.
TolX	Tolerancia de terminación de x , un escalar positivo. El valor por defecto para todos los algoritmos es 1e-6; para el algoritmo 'interior-point' es 1e-10.
TypicalX	Típicos valores x. El número de elementos en TypicalX es igual al número de elementos en x0, el punto de partida. El valor por defecto es 1 (número de variables, 1). fmincon utiliza TypicalX para ampliar las diferencias finitas para la estimación del gradiente. El algoritmo 'trust-region-reflective' utiliza TypicalX sólo para la opción DerivativeCheck.
UseParallel	true, estima los gradientes en paralelo. Deshabilitado por defecto con el valor por defecto, false. Trust-region-reflective requiere un gradiente, por lo tanto UseParallel no se aplica.

Por otro lado, los algoritmos que son desarrollados por el conjunto de funciones de optimización de MATLAB están basados en métodos de optimización matemáticos que pueden ser algoritmos de media escala o bien algoritmos de amplia escala. Esta distinción entre algoritmos se hace para diferenciar los algoritmos clásicos de los algoritmos de amplia escala, los cuales han sido diseñados para abordar problemas con más eficiencia.

Algoritmos de media escala

Entre los algoritmos de media escala nos encontramos con los conocidos como método de búsqueda simple *Nelder-Mead* y método *quasi-Newton* o de métrica variable *Broyden Fletcher Goldfarb Shanno* (BFGS) usados en la minimización sin restricciones. También se encuentran entre los algoritmos de media escala los métodos de programación cuadrática secuencial del inglés (*Sequential Quadratic Programming - SQP*) usados en las funciones de minimización con restricciones, problemas de mínimo-máximo, alcance de

objetivos y optimización semi-infinita. Y por último, nos encontramos los algoritmos que usan los problemas de mínimos cuadrados no lineales, tales como los métodos de Gauss-Newton y Levenberg-Marquardt [54].

Algoritmos de amplia escala

Los principales algoritmos de amplia escala son iterativos, es decir, que generan una secuencia de soluciones aproximadas. En cada iteración se resuelve un sistema lineal de forma aproximada. Entre este tipo de algoritmos se encuentran los problemas de límites con restricciones que son resueltos usando métodos de Newton. También problemas de igualdades con restricciones para los cuales se usa una iteración de gradiente conjugado preconditionado [54].

Además, es importante tener en cuenta que cada una de las funciones de optimización en MATLAB diferencian los dos tipos de algoritmos, los de amplia escala no son aplicables a todo tipo de problemas que se puedan formular, por lo que permite a *options* diferenciar el uso o no de estos algoritmos con el parámetro de entrada 'LargeScale', si se fija en "off" se usa de media escala y "on" el de amplia escala, el segundo cubre los casos en los que solamente hay límites con restricciones o sólo igualdades lineales. En el caso de usar los métodos de gran escala para resolver un problema es necesario definir el gradiente de la función a optimizar, siendo opcional en el caso de los de media escala. El cálculo del Gradiente puede convertirse en un problema, no obstante con la ayuda de MATLAB el cálculo computacional se vuelve un paso regular en el proceso de optimización.

El problema que se pretende resolver en este capítulo, requiere de forma general minimizar una variable y maximizar otra, sin embargo MATLAB maximiza o bien minimiza pero no se puede ejecutar las dos acciones al mismo tiempo es por esto que se realiza una minimización de la variable x que corresponde al complemento del *Throughput_Normalizado*, y de la variable y perteneciente a los datos de *End-to-End Delay*, las cuales son variables de las que depende la función objetivo sujeta a serie condiciones, por lo tanto se trata de un problema de optimización no lineal multivariable,

por ello las funciones elegidas para encontrar un número óptimo de nodos sensores han sido las funciones de minimización “fminunc” y “fmincon”.

En primera instancia, en el siguiente apartado se optimiza por separado las curvas del *Throughput*, Retardo y Paquetes Perdidos con respecto al número de nodos, este proceso corresponde a un problema de optimización escalar de una sola variable, al mismo tiempo es un problema no lineal por tratarse de un polinomio de segundo y cuarto grado como se describió en la Tabla 3.9, se elige como función a utilizar “fminbnd”.

4.1.1. OPTIMIZACIÓN ESCALAR NO LINEAL

4.1.1.1. Solución usando Interfaz Gráfica

Para llevar a cabo la optimización de las funciones objetivo (ecuación 3.5, 3.6, y 3.7) de una única variable se utiliza “fminbnd” que tiene como propósito encontrar el mínimo de la función dentro de un intervalo fijo x (nodos). Como ya se mencionó MATLAB permite dar solución a problemas de optimización mediante dos vías: Interfaz gráfica y por medio de comandos. En este apartado se describe de forma puntual el manejo de parámetros de entrada y opciones de la interfaz gráfica del *Optimization Toolbox*TM.

Para ejecutar el *toolbox* de optimización accedemos a través del menú *Start* → *Toolboxes* → *Optimization* → *Optimization tool (optimtool)*, o desde la línea de comandos *optimtool*. Desde el mismo menú es posible seleccionar la ayuda proporcionada por cada *toolbox* o ver ejemplos de uso del mismo.

Como se puede observar en la Figura 4.1, la ventana de la herramienta consta de tres secciones principalmente: *Problem Setup and Results* para seleccionar el tipo de *solver* (Método de solución), sus parámetros, y donde posteriormente se muestran los resultados del proceso de optimización, la segunda sección, *Options*, contiene las opciones disponibles para usar en cada herramienta con respecto al número de iteraciones, configuración para algoritmos de larga escala, gráficos de la función, etc, y la tercera parte *Quick Reference*, es un menú de ayuda.

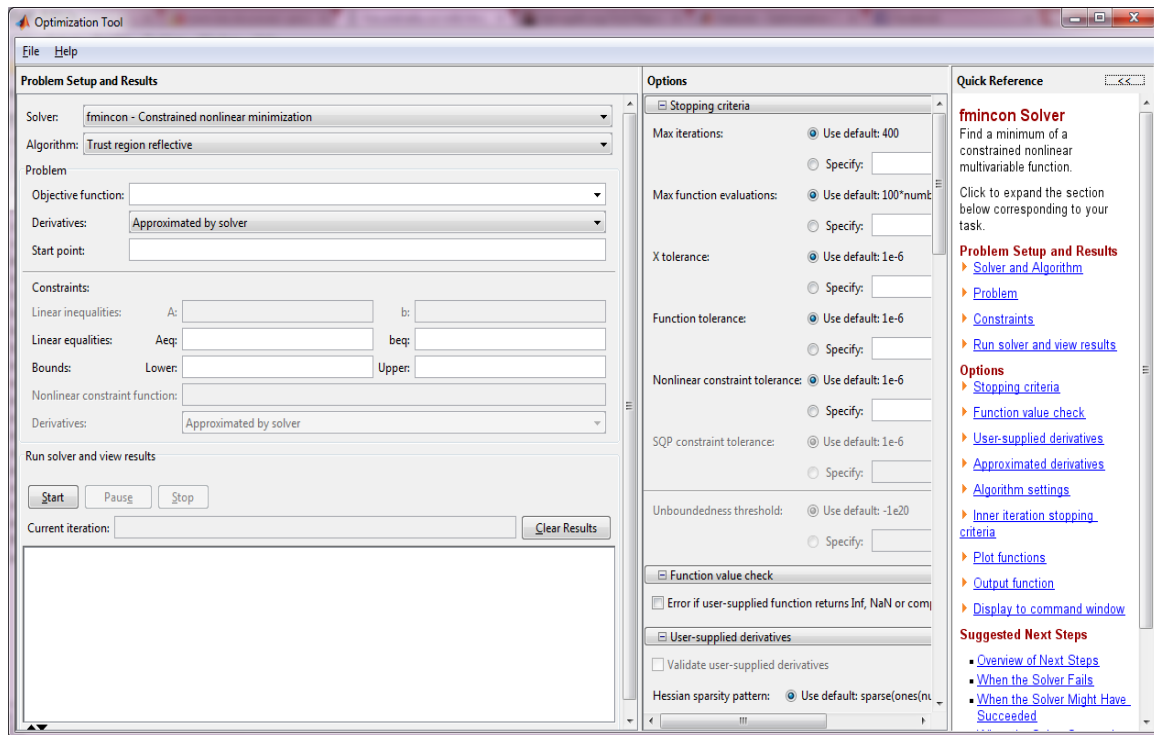


Figura. 4.1. Ventana principal, Optimization Toolbox™ MATLAB®.

Optimizando de forma individual la función del *Throughput* respecto al número de nodos, se emplea la función “fminbnd”, la cual se basa en el algoritmo matemático de búsqueda de la sección dorada y de interpolación parabólica (Método de Newton), la función sólo necesita ser definida para x en el intervalo $x_1 < x < x_2$. Si el mínimo se produce realmente en x_1 ó x_2 , “fminbnd” devuelve un punto x en el interior del intervalo (x_1, x_2) .

Disponiendo de la interfaz gráfica en la ventana *Problem Setup and Results* introducimos las variables que requiere el *solver* es decir el tipo de función o rutina a usar que para este caso es “fminbnd”, adicionalmente ingresamos los extremos del intervalo donde se desea buscar el punto óptimo, y en la ventana *options* conservamos los valores que vienen por defecto. Al presionar el botón *Start* se inicia el proceso de optimización, en el visor de resultados se observa los valores óptimos de la variable x , al evaluar la función objetivo en este valor alcanza su máximo, se muestra en la Figura 4.2.

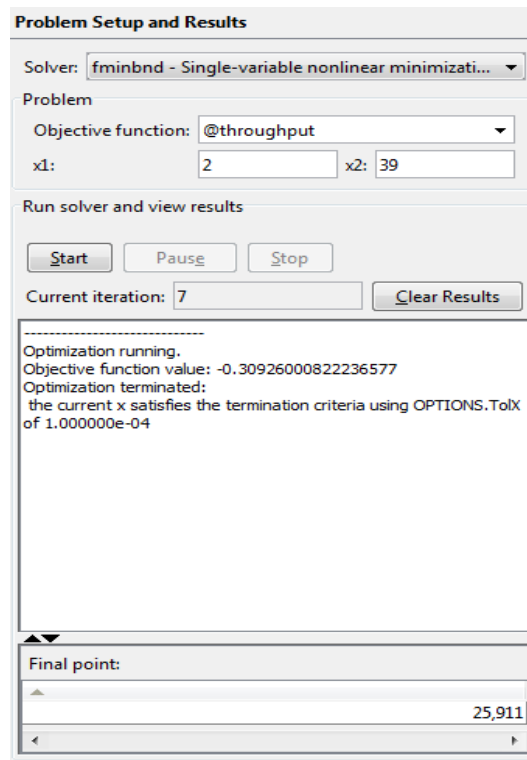


Figura. 4.2. Solución usando la función fminbnd sobre interfaz gráfica.

Es necesario definir el polinomio o función objetivo en un M-File para poder acceder a ella desde la herramienta de optimización anteponiendo @ seguido del nombre como se ilustra en la Figura 4.2. El valor obtenido por el *solver* de la función evaluada en el punto óptimo es negativo debido a que fue multiplicado por -1 para lograr la maximización, para nuestro interés no se toma en cuenta el signo. El punto encontrado por la herramienta es 25.911 que equivale a un número óptimo de nodos igual a 26.

De forma similar optimizamos las funciones pertenecientes al *End-to-End Delay* y *packet loss*, después de haber ejecutado la optimización no lineal univariable, se desea corrobora que los puntos máximos encontrados coincidan con los resultados gráficos ilustrados en las Figuras 3.13 y 3.14.

4.1.1.2. Solución usando Línea de Comandos

Al hacer uso de la función “fminbnd” desde la línea de comandos se espera obtener resultados similares al método gráfico. La ventaja de utilizar este mecanismo es que los

resultados pueden utilizarse como parámetros de otras funciones o para la implementación de un programa más complejo.

La sintaxis usada para llamar a esta función es:

```
x = fminbnd(fun,x1,x2,options)
```

Donde *fun* es la función objetivo, declarada en un fichero (*.m), *x1* y *x2* son los límites del intervalo en el que se desea encontrar el mínimo (máximo). “fminbnd” minimiza con mediante los parámetros de optimización de la estructura *options* especificados en la Tabla 4.4. Con ayuda del comando *optimset* se logra manipular o cambiar los valores de su estructura. Para nuestro caso se tiene:

```
% -----
%Optimización de una Variable(optim_throughput.m)-----
x1= 2; %Puntos iniciales
x2= 39;

options = optimset('Display','iter');
[x,fval,exitflag,output]= fminbnd(@throughput,x1,x2,options)

% -----
% -----
```

Al ejecutar el archivo desde la ventana de comandos obtenemos la siguiente información del proceso de optimización:

```
>> optim_throughput
  Func-count      x          f(x)        Procedure
      1         16.1327   -0.258082   initial
      2         24.8673   -0.310894   golden
      3         30.2655   -0.303426   golden
      4         26.2507   -0.3119    parabolic
      5         26.2507   -0.3119    parabolic
      6         26.2507   -0.3119    parabolic

Optimization terminated: the current x satisfies the
termination criteria using OPTIONS.TolX of 1.000000e-04

x =
    26.2507

fval =
   -0.3119

exitflag =
         1

output =
    iterations: 5
```

```

funcCount: 6
algorithm: 'golden section search, parabolic
interpolation'
message: [1x111 char]

```

Esta función tiene limitaciones sobre las soluciones que retorna, ya que si la función objetivo contiene varios mínimos en la región evaluada esta devuelve sólo uno de los mínimos; además, el *solver* es válido únicamente para valores en el dominio de los números reales. Sin embargo para el problema de optimización que buscamos resolver no limita en ningún sentido porque se maneja netamente valores reales.

Los datos obtenidos tras la optimización indican que se trata de una maximización por el signo negativo del valor óptimo que alcanza la función, El valor 26.2507 y -0.3119 representan gráficamente el punto más alto de la curva del *Throughput* presentada en la Figura 3.13 a). La maximización de las curvas del Retardo y Pérdida de Paquetes se realiza bajo el mismo concepto.

4.1.2. OPTIMIZACIÓN MULTIVARIABLE NO LINEAL

Una vez optimizadas las funciones de una variable, es hora de pasar a la optimización de la función de interés de dos variables (*Throughput* y *End-to-End Delay*), para ello se reitera que el alcance del proyecto plantea la aplicación de técnicas de optimización apropiadas y flexibles, pues con la comparación de estas técnicas se determina el nivel de ajuste que tienen cada uno de ellas hacia nuestro problema a resolver y con esto definir el mejor método, ya que los puntos máximos y mínimos hallados en la superficie 3D, directamente nos conduce a encontrar el número óptimo de sensores donde el valor de *Throughput* es estable y suficiente para lograr una monitorización adecuada de señales sísmicas.

El problema de optimización no lineal de varias variables se puede definir mediante la siguiente expresión,

$$\begin{aligned}
 f(x,y) = & 13.66 - 8.177x + 4.556y + 0.9576x^2 - 0.9681xy + 10.81y^2 + \\
 & 1.555x^3 + 1.533x^2y - 2.91xy^2 - 6.116y^3 + 4.086x^4 + 9.492x^3y + \\
 & 1.08x^2y^2 + 0.9793xy^3 + 1.08y^4
 \end{aligned} \tag{3.8}$$

Donde, $f(x, y)$ es una tercera variable z igual al número de nodos, x es el valor del *Throughput* y y es el retardo promedio de la red.

A continuación se define tres casos de optimización posibles para la función objetivo (ecuación 3.8) dependiendo de las restricciones planteadas:

Caso 1.

Minimizar la función

$$f(x, y),$$

$$s. a. \quad x_1 \leq x \leq x_2, \quad y_1 \leq y \leq y_2 \quad (4.1)$$

Donde, $f(x, y)$ es el número de nodos, x_1, \dots, x_2 son los valores de *Throughput* y y_1, \dots, y_2 es el retardo promedio de la red. En este primer caso la restricción del dominio puede o no ser tomar en cuenta, al fin que se trata de los valores extremos de la superficie.

Caso 2.

Minimizar la función

$$f(x, y), \quad s. a. \quad g(x, y) \quad (4.2)$$

Donde, $g(x, y)$ es la restricción no lineal (ecuación 3.10) análoga al máximo porcentaje de paquetes perdidos.

Cuando se requieran dar restricciones de desigualdad el *toolbox* de Matlab siempre toma restricciones del tipo menor que ($c \leq b$). Si se requieren desigualdades de tipo mayor que $c \geq b$, se debe multiplicar por -1 así pues se deja en la forma menor que, para que pueda ser entendida por el *toolbox*. Por tanto tenemos que $c \geq b$ es equivalente a $a - c \leq -b$.

Caso 3

Minimizar la función

$$f(x, y), \quad s. a. \quad \begin{cases} g(x, y) \\ G_1(x), G_2(y) \end{cases} \quad (4.3)$$

Sobre la superficie proyectada de $f(x, y)$ en la Figura 4.3, se fija un punto $P(x, y)$ límite donde ésta tienda a decrecer de forma considerable, convirtiéndolo en restricciones gráficas $G_1(x)$ y $G_2(y)$ descrita de la siguiente manera:

$$G_1(x) \leq 0 \approx -x + 0.8892 \leq 0 ; G_2(y) \leq 0 \approx -y + 0.004195 \leq 0 \tag{4.4}$$

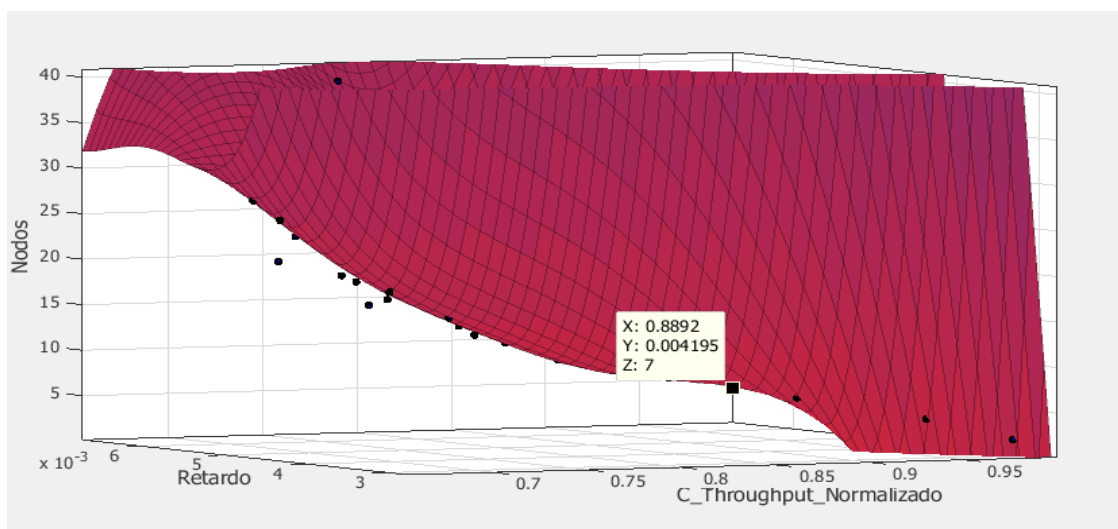


Figura. 4.3. Restricción gráfica en $P(x,y)$.

En la Tabla 4.8 se realiza un resumen de todas las restricciones de desigualdades no lineales definidas hasta el momento.

Tabla. 4.6.

Restricciones de desigualdad no lineal

Descripción	Expresión
Restricción del 25% de paquetes perdidos. (Caso2)	$-14.34 - 8.177x + 4.556y + 0.9576x^2 - 0.9681xy + 10.81y^2 + 1.555x^3 + 1.533x^2y - 2.91xy^2 - 6.116y^3 + 4.086x^4 + 9.492x^3y + 1.08x^2y^2 + 0.9793xy^3 + 1.08y^4 \leq 0$
Restricción gráfica de tipo $Ax \leq b$.(Caso 3)	$-x + 0.8892 \leq 0$
Restricción gráfica de tipo $Ay \leq b$.(Caso 3)	$-y + 0.004195 \leq 0$

4.1.2.1. Solución usando Interfaz Gráfica

– Aplicación de Métodos de Optimización - Primer Caso

Para desarrollar el proceso de minimización (maximización) sin restricciones de la función de varias variables, MATLAB ofrece las funciones “fminunc” y “fminsearch” que solo requieren información de la función y el dominio donde se extiende, las dos opciones componen sentencias y algoritmos similares por lo tanto se utilizará solo una, “fminunc” se toma como primer método a ejecutar.

La función “fminunc” puede hacer uso de dos tipos de algoritmos matemáticos modificando los parámetros de entrada del campo *options*: Algoritmo de región de confianza o Trust-Region donde se requiere el gradiente de la función, es activado utilizando la sentencia *optimset*, se basa en el método de Newton interior reflectante. Y el segundo donde cada iteración implica la solución aproximada de un sistema lineal de gran tamaño utilizando el método de gradientes conjugados pre-condicionados (PCG).

El segundo algoritmo matemático aplicable es el Quasi-Newton, éste es un método de búsqueda de óptimos en funciones derivables ya que construye una aproximación de la curvatura de la función no lineal utilizando sólo información del gradiente, evitando por lo tanto calcular de forma explícita la matriz hessiana.

Como ya se mencionó la función a optimizar debe ser guarda en un archivo (*.m) junto a la información del gradiente y matriz hessiana si así lo requiere, para nuestro proceso de minimización solo se hace uso del gradiente (primera derivada), se muestra a continuación las líneas de código.

```
% -----
% Función de Objetivo-----
function [f,g] = myfun1(x) % myfun1 tiene dos parámetros de salida

f = ( 13.66 -8.177*x(1) + 4.556*x(2) + 0.9576 *x(1).^2 -0.9681*x(1)*x(2)
+ 10.81*x(2).^2 + 1.555*x(1).^3 + 1.533*x(1).^2*x(2) -2.91*x(1)*x(2).^2 -
6.116*x(2).^3 + 4.086*x(1).^4 + 9.492*x(1).^3*x(2)+1.08*x(1).^2*x(2).^2 +
0.9793*x(1)*x(2).^3 + 1.08*x(2).^4);

if nargin > 1 % Calcula el gradiente
```



```

g(1)=(2043*x(1).^3)/125 + (7119*x(1).^2*x(2))/250 + (933*x(1).^2)/200 +
(54*x(1)*x(2).^2)/25 + (1533*x(1)*x(2))/500 + (1197*x(1))/625 +
(9793*x(2).^3)/10000 - (291*x(2).^2)/100 - (9681*x(2))/10000 -
8177/1000;

g(2)=(2373*x(1).^3)/250 + (54*x(1).^2*x(2))/25 + (1533*x(1).^2)/1000 +
(29379*x(1)*x(2).^2)/10000 - (291*x(1)*x(2))/50 - (9681*x(1))/10000 +
(108*x(2).^3)/25 - (4587*x(2).^2)/250 + (1081*x(2))/50 + 1139/250;
end
end
-----
-----

```

En la ventana del optimizador seleccionamos como *solver* a utilizar “fminunc” de larga escala, invocamos a la función objetivo, y definimos un punto inicial que gráficamente debe ser aproximado al punto óptimo.

En la Figura 4.4 se ilustra la configuración de parámetros y los resultados obtenidos.

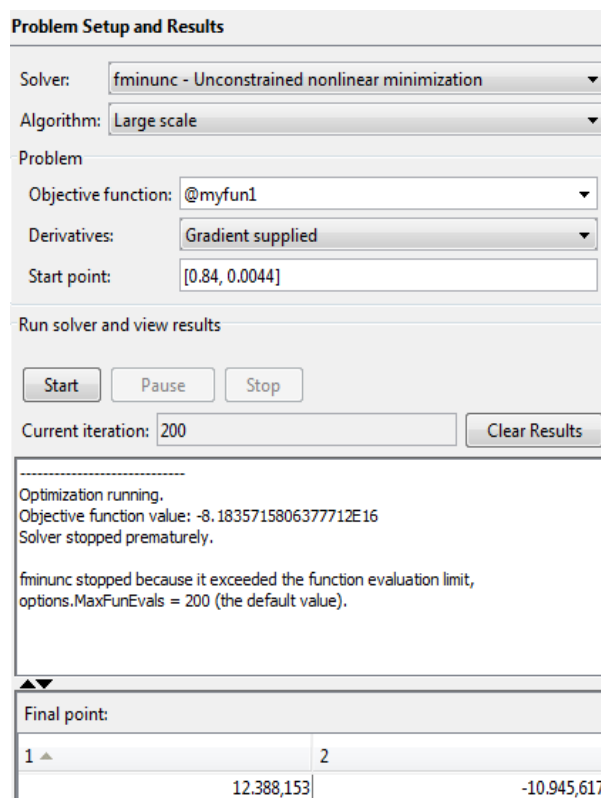


Figura. 4.4. Solución gráfica usando la función fminunc.

El los valores encontrados de x y y donde la función alcanza su punto más alto son totalmente fuera del rango analizado, “fminunc” sobre la herramienta no solicita información de los límites donde debe buscar el mínimo y cuando ejecuta las iteraciones

sobrepasa el área de la superficie de la Figura 4.3 hallando uno de los extremos mínimos, donde la función objetivo es negativa a pesar que se la está minimizando.

– Aplicación de Métodos de Optimización - Segundo Caso

La minimización de una función multivariable sujeta a una serie de restricciones, ya sean de desigualdad o igualdad, lineales o no, puede ser procesada mediante la función “fmincon”, los argumentos de entrada para esta función son específicos y ayudan a la búsqueda de un mínimo local, por el contrario, si se busca un óptimo global es necesario emplear “patternsearch”, que encuentra el mínimo de una función utilizando un patrón de búsqueda.

En este apartado se emplea la función “fmincon” como método de optimización, se programa para trabajar con algoritmos de media escala tal como: SQP, Quasi-Newton, y line-search.

La restricción correspondiente al caso presentado en la Tabla 4.7 debe guardarse en un archivo (*.m), se la escribe en modo de función, como sigue a continuación:

```
% -----
% Función de restricciones-----
function [C, Ceq]=nonlcon(x)

    C= [-14.34-8.177*x(1)+4.556*x(2)+0.9576*x(1).^2-0.9681*x(1)*x(2)+
        10.81*x(2).^2+1.555*x(1).^3+1.533*x(1).^2*x(2)-2.91*x(1)*x(2).^2-
        6.116*x(2).^3+4.086*x(1).^4+9.492*x(1).^3*x(2)+1.08*x(1).^2*x(2).^2+
        0.9793*x(1)*x(2).^3+ 1.08*x(2).^4];

    Ceq=[];

End
% -----
% -----
```

El archivo tiene el mismo nombre de la función “nonlcon”, desde el campo de funciones de restricciones no lineales del *Toolbox* se invoca a esta función de restricción anteponiendo @ al nombre, en el campo *Derivatives* especificamos el uso o no del gradiente, en la Figura 4.5 se presenta el proceso de optimización en base a éste, donde es

obligatorio ingresar los datos de *Bounds* o límites de la superficie, las matrices vacías (no se cuenta con restricciones de igualdad o desigualdad) y el tipo de algoritmo.

Problem Setup and Results

Solver: **fmincon - Constrained nonlinear minimization**

Algorithm: **Active set**

Problem

Objective function: **@myfun1**

Derivatives: **Gradient supplied**

Start point: **[0.94 0.0033]**

Constraints:

Linear inequalities: A: **[]** b: **[]**

Linear equalities: Aeq: **[]** beq: **[]**

Bounds: Lower: **[0.67 0.002769]** Upper: **[0.999 0.0051]**

Nonlinear constraint function: **@nonlcon**

Derivatives: **Approximated by solver**

Run solver and view results

Start **Pause** **Stop**

Current iteration: **2** **Clear Results**

Optimization running.
Objective function value: 9.923039545269551
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in feasible directions, to within the default value of the function tolerance, and constraints are satisfied to within the default value of the constraint tolerance.

Final point:

1	2
0.67	0.003

Figura 4.5. Solución gráfica con la función fmincon.

Corremos el optimizador y empleando el gradiente de la función tenemos como respuesta que el número óptimo de nodos es 9.92304 equivalente a 10 en el punto $x = 0.67$ y $y = 0.003$. Adicionalmente, en el panel de *Constraints* (Restricciones) se puede elegir si se utiliza datos del gradiente de las funciones de restricción, para nuestro programa no se maneja esta información, se le permite al *solver* utilizar una aproximación.

Bajos las mismas condiciones se ejecuta la optimización de la función objetivo sin la intervención de su gradiente, pues se pretende validar la eficiencia de la *Toolbox* de Optimización de MATLAB respecto a este parámetro, se ilustra en la Figura 4.6.

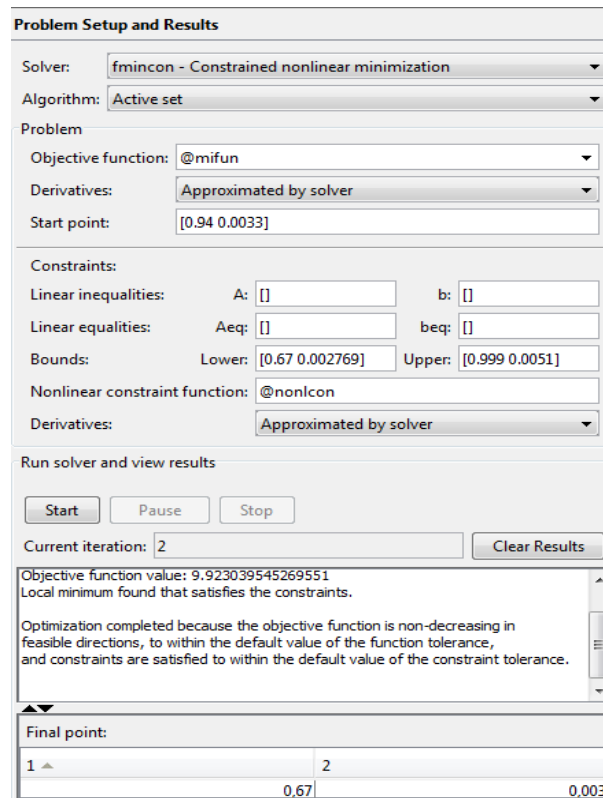


Figura. 4.6. Solución gráfica sin intervención del Gradiente.

– Aplicación de Métodos de Optimización - Tercer Caso

La optimización en el tercer caso se diferencia del anterior en el incremento del número de restricciones, el tipo de *constraints* siguen siendo de desigualdad no lineal por lo tanto se añade estas al archivo (nonlcon.m) de la siguiente manera,

```
% -----
% Función de restricciones-----

function [C, Ceq]=nonlcon(x)

C= [-14.34-8.177*x(1)+4.556*x(2)+0.9576*x(1).^2-0.9681*x(1)*x(2)+
10.81*x(2).^2+1.555*x(1).^3+1.533*x(1).^2*x(2)-2.91*x(1)*x(2).^2-
6.116*x(2).^3+4.086*x(1).^4+9.492*x(1).^3*x(2)+1.08*x(1).^2*x(2).^2+
0.9793*x(1)*x(2).^3+ 1.08*x(2).^4; -x(1)+0.8892; -x(2)+0.004195];

Ceq=[];

End
% -----
% -----
```

En la Figura 4.7 se muestra el proceso de optimización sujeto a todas las restricciones descritas en el transcurso del desarrollo del proyecto.

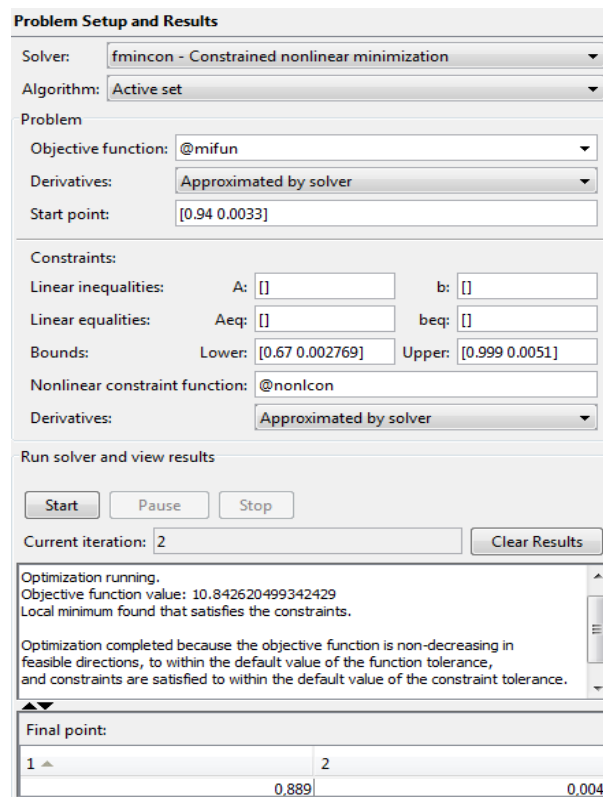


Figura. 4.7. Solución gráfica sujeta a restricciones.

El proceso de optimización al incrementar el número de restricciones es similar, sin embargo las restricciones que añadimos producen su efecto ya que limitan o acotan la región de búsqueda, es decir es menor al caso anterior, el valor máximo de la función es 10.8426 que equivale aproximadamente a 11 nodos, en la Figura 4.7 se observa que el resultado obtenido no varía de forma brusca respecto al anterior.

4.1.2.2. Solución usando Línea de Comandos

El siguiente programa desarrollado para el primer caso de optimización sin restricciones contiene pocos argumentos de entrada a la función “fminunc”, se fija un punto inicial por donde el optimizador empieza a buscar el mínimo, los argumentos de entrada para *options* nos ayudan a visualizar la información más detallada acerca del proceso y las iteraciones ejecutadas:

```

% -----
%Optimización Multivariable(optim_fminunc.m)-----
x0 = [0.84, 0.0044]; % condiciones iniciales
options = optimset('Diagnostics', 'on','Display', 'iter', 'GradObj','on',
'Hessian', 'off', 'LargeScale', 'on', 'TolFun', 1E-12,'TolX', 1E-15 );

[x,fval,exitflag,output] = fminunc(@myfun1,x0,options)

% -----
% -----

```

Tras ejecutar la programación nos encontramos con el siguiente resultado en la ventana de comandos, se detalla el tipo de algoritmo matemático que emplea la rutina para encontrar el mínimo de la función algebraica, además indica el número de iteraciones y el valor que va tomando la función en cada iteración que se ejecutan.

```
>> optim_fminunc
```

Diagnostic Information

Number of variables: 2

Functions

Objective and gradient: myfun1

Hessian: finite-differencing (or Quasi-Newton)

Algorithm selected large-scale: trust-region Newton

End diagnostic information

Iteration	f(x)	Norm of step	First-order optimality	CG-iterations
0	10.4691		10.5	
1	9.32366	1.02779	15	1
2	-12033.8	10	7.37e+03	1
3	-944818	20	1.16e+05	1
4	-2.70458e+07	40	1.14e+06	1
5	-5.62485e+08	80	1.15e+07	1
32	-3.74976e+13	84.0012	4.65e+10	1
199	-8.0185e+16	84.0012	1.46e+13	1
200	-8.18357e+16	84.0012	1.48e+13	1

Solver stopped prematurely.

fminunc stopped because it exceeded the function evaluation limit, options.MaxFunEvals = 200 (the default value).

```

x =
    1.0e+04 *
    1.2388 -1.0946

fval =
    -8.1836e+16

exitflag =
    0

output =
    iterations: 200
    funcCount: 201
    cgiterations: 200
    firstorderopt: 1.4839e+13
    algorithm: 'large-scale: trust-region Newton'
    message: [1x142 char]
    constrviolation: []

```

Para el segundo y tercer caso la programación cambia ya que los elementos de entrada de “fmincon” son más complejos de definir, sin embargo en la solución gráfica se especifica los mismos parámetros excepto los de *options* que se seleccionaron en base a la Tabla 4.7.

```

% -----
%Optimización Multivariable con restricciones (optim_fmincon.m) -----
x0=[0.94 0.0033] % Condiciones Iniciales

options = optimset('Diagnostics', 'on','Display','iter','Algorithm',
'active-set');

[x,fval,exitflag,output]= fmincon(@mifun,x0,[],[],[],[],[0.67 0.002769],...
[0.999 0.0051],'nonlcon',options)

% -----
% -----

```

El proceso de optimización además de la información ya mencionada en los casos anteriores, presenta el tipo de restricciones que se utiliza para limitar el área de búsqueda del mínimo.

```
>> optim_fmincon
```

Diagnostic Information

Number of variables: 2

Functions

Objective: mifun
 Gradient: finite-differencing
 Hessian: finite-differencing (or Quasi-Newton)
 Nonlinear constraints: nonlcon
 Nonlinear constraints gradient: finite-differencing

Constraints

Number of nonlinear inequality constraints: 1
 Number of nonlinear equality constraints: 0

Number of linear inequality constraints: 0
 Number of linear equality constraints: 0
 Number of lower bound constraints: 2
 Number of upper bound constraints: 2

Algorithm selected medium-scale: SQP, Quasi-Newton, line-search

End diagnostic information

Iter	F-count	Max f(x)	Line search constraint	Directional steplength	First-order derivative	optimality
0	3	11.3441	-0.000531			
1	6	9.92304	0	1	-11.4	11

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in feasible directions, to within the default value of the function tolerance, and constraints are satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

Active inequalities (to within options.TolCon = 1e-06):

lower	upper	ineqlin	ineqnonlin
		1	
		2	

x =
 0.6700 0.0028


```
fval =  
  
9.9230  
  
exitflag =  
  
1  
  
output =  
  
    iterations: 2  
    funcCount: 6  
    lssteplength: 1  
    stepsize: 0  
    algorithm: 'medium-scale: SQP, Quasi-Newton, line-search'  
    firstorderopt: 0  
    constrviolation: 0  
    message: [1x783 char]
```

El valor más alto que toma la función resulto 9.9230 que comparado a la solución obtenido en el método gráfico es estrechamente similar.

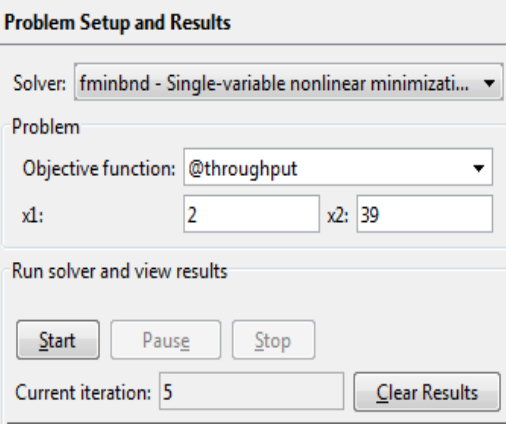
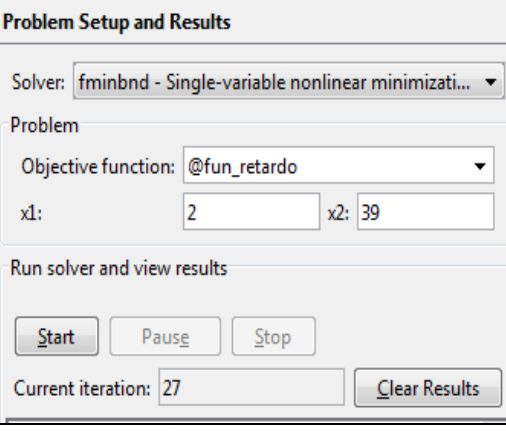
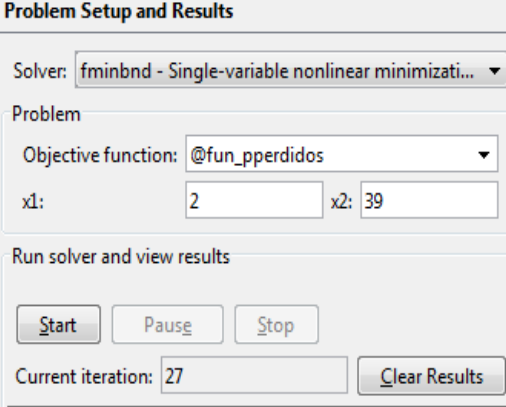
4.2 TABULACIÓN DE RESULTADOS

En este segmento se realiza la tabulación de los valores obtenidos tras la aplicación de técnicas de optimización de funciones de una y varias variables.

4.2.1 RESULTADOS DE OPTIMIZACIÓN NO LINEAL DE FUNCION UNIVARIABLE

En la Tabla 4.9 se presenta la configuración de argumentos de entrada de la *Optimization Toolbox*TM de MATLAB® y programación realizada que se llevó a cabo para la optimización numérica escalar de la funciones del *Throughput*, retardo promedio y paquetes perdidos, mediante la función “fminbnd”.

Tabla. 4.7.
Configuración y programación de parámetros en fminbnd

Interfaz gráfica	Línea de Comandos
	<pre>disp(' OPTIMIZACIÓN DE FUNCIÓN DE UNA VARIABLE') disp('-----') disp(' x=x(1)=Número de nodos sensores') disp(' y=f(x)=Throughput de la red') x1= 2;%Puntos iniciales x2= 39; options = optimset('Display','iter'); [x,fval,exitflag,output]= fminbnd(@throughput,x1,x2,options)</pre>
<ul style="list-style-type: none"> - Los campos x_1 y x_2 son los extremos del intervalo x donde se busca el punto máximo. - La variación de los argumentos de entrada de <i>Options</i> no afecta al valor resultante. 	
	<pre>disp(' OPTIMIZACIÓN DE FUNCIÓN DE UNA VARIABLE') disp('-----') disp(' x=x(1)=Número de nodos sensores') disp(' y=f(x)=Retardo de la red') x1= 2;%Puntos iniciales x2= 39; options = optimset('Display','iter'); [x,fval,exitflag,output]= fminbnd(@fun_retardo,x1,x2,options)</pre>
<ul style="list-style-type: none"> - La variable x es común para las tres funciones, representa el número de nodos, este intervalo encierra 34 puntos equivalente al número de escenarios simulados bajo $ns-2$. 	
	<pre>disp(' OPTIMIZACIÓN DE FUNCIÓN DE UNA VARIABLE') disp('-----') disp(' x=x(1)=Número de nodos sensores') disp(' y=f(x)=Paquetes Perdidos de la red') x1= 2;%Puntos iniciales x2= 39; options = optimset('Display','iter'); [x,fval,exitflag,output]= fminbnd(@fun_pperdidos,x1,x2,options)</pre>
<ul style="list-style-type: none"> - Los parámetros x, $fval$, $exitflag$, $output$ no influyen en el resultado, sin embargo proporciona información más detallada del resultado. 	

Cuando se encuentra un punto óptimo utilizando las dos opciones que ofrece MATLAB es importante solicitar información detallada del proceso que realiza la

herramienta, como ya se mencionó, esto se realiza mediante modificación de las entradas a la estructura y sintaxis de las rutinas. Los resultados obtenidos tras iniciar la búsqueda del máximo en la herramienta de optimización y al ejecutar cada uno de los *script*, se muestra en la siguiente Tabla.

Tabla. 4.8.
Resultados de Optimización de funciones de una variable

Resultados				
Función	fval = y = f(x) (Función evaluada en el punto óptimo)		Punto (x) (Número de Nodos)	
	LC (Línea de Comandos)	IG (Interfaz Gráfica)	LC (Línea de Comandos)	IG (Interfaz Gráfica)
Throughput	0.3119	0.3119	26.2507	26.251
Retardo	0.0064	0.006523	38.9999	39
Perdida de Paquetes	1.1932*e+04	1.19324*e+04	38.9999	39

Tabla. 4.9.
Datos del proceso de Optimización

Función	Algorithm (Algoritmo)	Iterations (Nº Iteraciones)	Exitflag (Bandera de salida)
Throughput	'golden section search, parabolic interpolation'	5	1
Retardo	'golden section search, parabolic interpolation'	27	1
Perdida de Paquetes	'golden section search, parabolic interpolation'	27	1

La Tabla 4.11 contiene información de salida relevante del proceso de optimización, con ayuda del parámetro de entrada 'Algorithm' de *options*, MATLAB nos permite conocer el algoritmo matemático que utiliza la función, así como el número de iteraciones o evaluaciones que realiza se puede conocer con más claridad y exactitud que función es más eficiente.

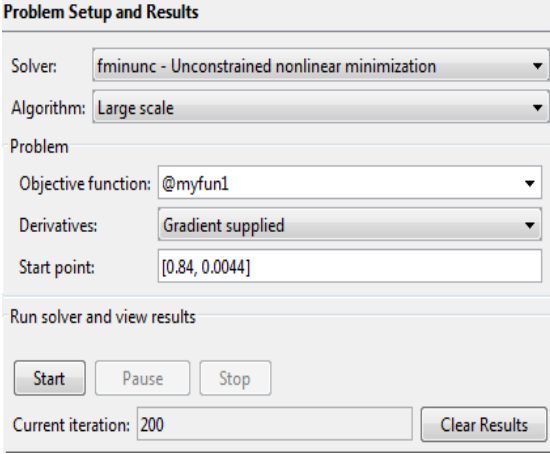
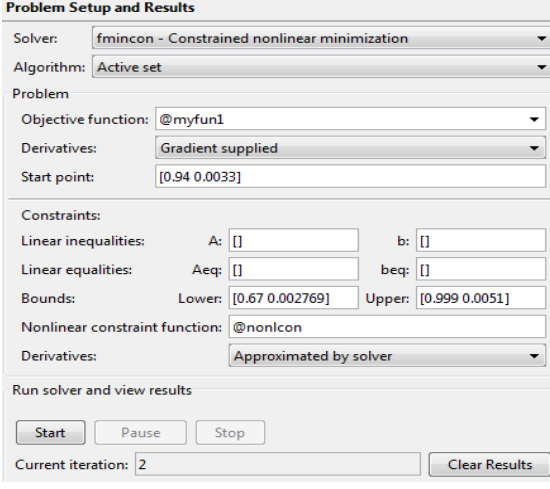
4.2.2 RESULTADOS DE OPTIMIZACIÓN NO LINEAL DE FUNCION MULTIVARIABLE

La siguiente información descrita en la Tabla 4.12 nos permite encontrar el punto mínimo de la función objetivo dependiente de las variables x (*Throughput*) y y (*End-to-*

End Delay), para ello es importante reiterar que los valores que toma la variable x son el complemento del *Throughput* normalizado respecto a la tasa de transmisión 250 kbps.

Tabla. 4.10.

Configuración y programación de parámetros en `fminunc` y `fmincon`

Interfaz gráfica	Línea de Comandos
	<pre>disp(' OPTIMIZACIÓN DE FUNCIÓN DE VARIAS VARIABLES') disp(' x=x(1)=Throughput de la red') disp(' y=x(2)=Retardo de la red') disp(' z=f(x,y)=Número de nodos sensores') disp(' Condiciones Iniciales') x0 = [0.84, 0.0044]; options = optimset('Diagnostics', 'on','Display','iter', 'GradObj','on','Hessian','off', 'LargeScale','on','TolFun', 1E- 6,'TolX', 1E-6); [x,fval,exitflag,output] =fminunc(@myfun1,x0,options)</pre>
<ul style="list-style-type: none"> - La herramienta de optimización permite seleccionar el tipo de algoritmo que emplea la función “fminunc”. - El cálculo del gradiente es opcional, la herramienta de optimización puede aproximar ese valor sin presentar errores. - Para optimizar con “fminunc” se utilizó la información del gradiente, 'GradObj', 'on' - La tolerancia de terminación de la función y variable x 'TolFun', $1E-6$, 'TolX', $1E-6$ son definidos por <i>default</i>. 	
	<pre>disp(' OPTIMIZACIÓN DE FUNCIÓN DE VARIAS VARIABLES') disp(' x=x(1)=Throughput de la red') disp(' y=x(2)=Retardo de la red') disp(' z=f(x,y)=Número de nodos sensores') disp(' Condiciones Iniciales') x0=[0.94 0.0033] options = optimset('Diagnostics', 'on','Display','iter','Algorithm', 'active-set'); [x,fval,exitflag,output]= fmincon(@mifun,x0,[],[],[],[],[0.6 7 0.002769],[0.999 0.0051],'nonlcon',options)</pre>
<ul style="list-style-type: none"> - El punto inicial es escogido de forma aleatoria, sin embargo es recomendable tomar un punto que gráficamente se aproxime al mínimo. - La activación de 'Diagnostics', 'on' permite visualizar de forma resumida la cantidad y tipo de funciones que intervienen en el proceso. - El uso del algoritmo 'Algorithm', 'active-set' permite utilizar varios algoritmos (SQP, Quasi-Newton, line-search) para buscar el mínimo óptimo. 	

En la Tabla 4.13 se muestran los valores obtenidos, los resultados son hallados mediante las funciones “fminunc” y “fmincon”, las cuales aplican distintas sentencias y algoritmos matemáticos para hallar el punto óptimo (PO) que corresponde al número de nodos.

Tabla. 4.11.

Resultados de Optimización de funciones multivariables sujetas a Desigualdades no lineales (DNL), mediante línea de Comandos (LC) e Interfaz Gráfica (IG)

Resultados							
Caso	Nº de Restricciones	z = f(x,y) (PO)		Punto x (Throughput)		Punto y (Retardo)	
		LC	IG	LC	IG	LC	IG
1 (fminunc)	Ninguna	8.1836*e +16	8.1836*e +16	1.2388* e+04	1.2388* e+04	-1.0946* e+04	-1.0946* e+04
2 (fmincon)	Límites x y y	9.9230	9.9230	0.6700	0.6700	0.0028	0.003
2 (fmincon)	1 (DNL)	9.9230	9.9230	0.6700	0.6700	0.0028	0.003
3 (fmincon)	3 (DNL)	10.8426	10.8426	0.8892	0.889	0.0042	0.004

Los resultados presentados en la Tabla 4.13 de los casos planteados 1, 2, y 3 de optimización numérica son hallados mediante los algoritmos matemáticos que se describen a continuación.

Tabla. 4.12.

Datos del proceso de Optimización

Caso	Algorithm (Algoritmo)	Nº Iteraciones	Bandera de salida
1 (fminunc)	'medium-scale: Quasi-Newton line search'	4	-3
2 (fmincon)	'medium-scale: SQP, Quasi-Newton, line-search'	2	1
2 (fmincon)	'medium-scale: SQP, Quasi-Newton, line-search'	2	1
3 (fmincon)	'medium-scale: SQP, Quasi-Newton, line-search'	2	1

4.3 RESULTADOS

4.3.1 REPRESENTACIÓN GRÁFICA Y ANÁLISIS

Los parámetros que van a ser analizados en esta sección para determinar el número de nodos óptimo en redes de sensores inalámbricos aplicadas a Monitorización Volcánica son:

- *Throughput* de la red respecto al punto óptimo (Número de nodos)
- Retardo de la red respecto al punto óptimo (Número de nodos)
- Pérdida de paquetes de la red respecto al punto óptimo (Número de nodos)
- Nodos en función del *Throughput* y Retardo

4.3.1.1. *Throughput* de la red respecto al punto óptimo (Número de nodos)

Los valores de *Throughput* de la Tabla 3.6 extraídos de la simulación de la Red de Sensores Inalámbricos muestran claramente el comportamiento creciente que toma la función frente al número de nodos, por eso en el Capítulo 3 se dedujo que la función que representa esta relación es de segundo orden, y es por este motivo que la herramienta de optimización realiza un menor esfuerzo y aplica algoritmos o métodos matemáticos de optimización simple como Interpolación Parabólica y Búsqueda de la Sección Dorada como se describe en la Tabla 4.11.

Con los valores obtenidos en la Tabla 4.10 se determina que el valor x (nodos) donde el *Throughput* alcanza su máximo valor es 26, por lo tanto la tasa máxima de transmisión de paquetes de la WSN es 81 kbps que equivale al 32.4% de la tasa de transmisión teórica definida por el estándar Zigbee/IEEE 802.15.4, si ubicamos gráficamente el punto óptimo 26 tenemos el resultado gráfico ilustrado en la Figura 4.8, validamos gráficamente que se trata de un óptimo absoluto. Por lo tanto con 26 nodos se alcanza el máximo rendimiento de la red, no obstante, hay que tomar en cuenta que después de este punto el *Throughput* tiende a decaer prolongadamente.

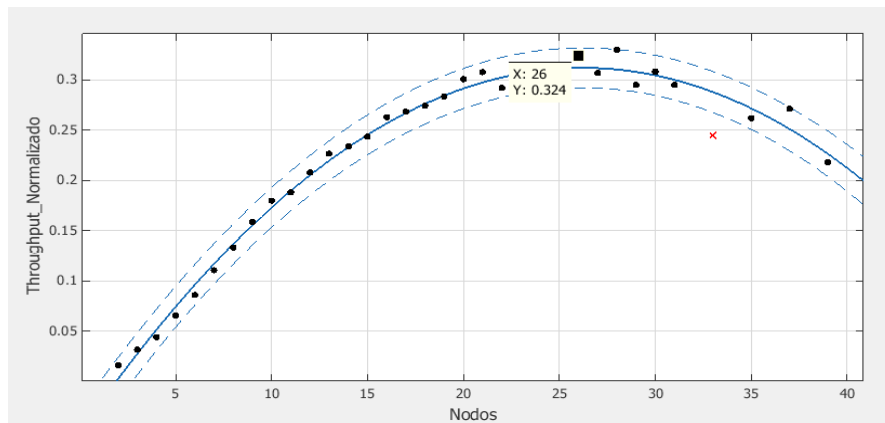


Figura. 4.8. Resultado gráfico Optimización Univariable, Throughput vs Nodos

4.3.1.2. Retardo de la red respecto al punto óptimo (Número de nodos)

Los valores de la Tabla 4.11 muestran que los algoritmos empleados por el optimizador son los mismos debido a que la forma más rápida y fácil de optimizar funciones univariadas es mediante “fminbnd”, sin embargo, el número de iteraciones muestra que en esta ocasión se realizó un mayor número de búsquedas y esto se debe a que los valores del retardo de los escenarios simulados tienden a formar una curva creciente y decreciente en diferentes intervalos.

En la Tabla 4.10 se observa claramente que el valor del retardo máximo corresponde a un número de nodos igual a 39, es decir a mayor número de nodos el retardo es máximo, el valor hallado por el optimizador corresponde a un máximo absoluto, por lo tanto, si ubicamos este punto sobre la curva presentada en la Figura 4.9, verificamos la teoría.

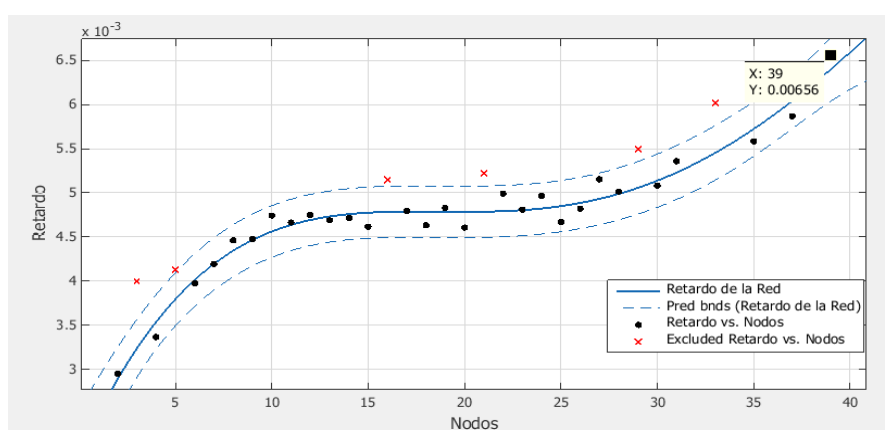


Figura. 4.9. Resultado gráfico Optimización Univariable, Retardo vs Nodos

4.3.1.3. Pérdida de paquetes de la red respecto al punto óptimo (Número de nodos)

Según la Tabla 4.10, las dos opciones de optimización permitidas por MATLAB devuelven el punto $1.1932e+04$, siendo éste el punto máximo que alcanza la función de paquetes perdidos, este valor depende de la variable número de nodos igual a 39, siendo un óptimo absoluto, por lo tanto se corrobora que a mayor número de nodos mayor será el número de paquetes perdidos. No obstante, en el Capítulo 3 se limitó al 25%, es decir la red de sensores inalámbricos puede permitir una pérdida de paquetes máximo del 25%, de lo contrario el *Delivery Ratio* estaría por debajo del 60% que influye directamente en la disminución de la tasa de transmisión de la red.

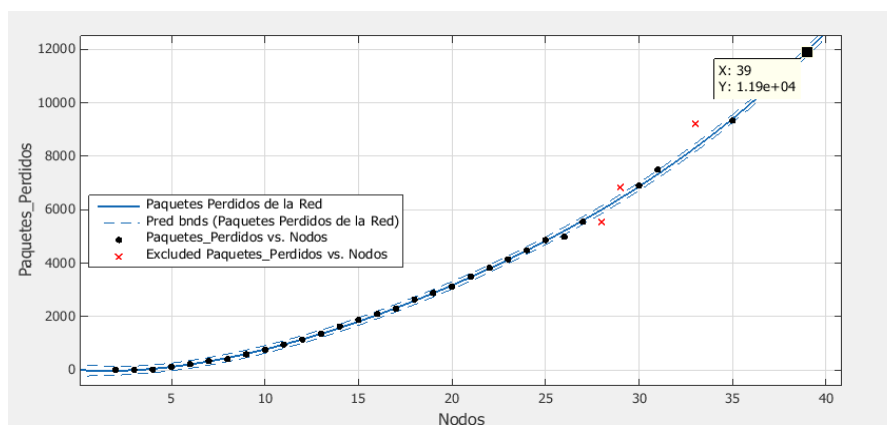


Figura. 4.10. Resultado gráfico Optimización Univariable, Paquetes Perdidos vs Nodos

Si verificamos los valores de paquetes perdidos de la Tabla 3.6 podemos determinar que a partir del nodo 28 se sobrepasa el porcentaje límite, por lo tanto se descarta el intervalo de nodos (29, 30, 31, ...,39).

4.3.1.4. Nodos en función del *Throughput* y Retardo

Mediante un cuadro comparativo presentado en la Tabla 4.15 se verifica, que si se requiere resolver un problema de optimización no lineal multivariable la función “fminunc” no es una buena opción ya que al no solicitar los límites de la superficie donde se desea buscar el máximo (mínimo) como argumento de entrada, está inicia una búsqueda a partir del punto inicial y sale del área de nuestro interés y proporciona óptimos (máximos o mínimo) de la superficie completa, es por eso que como punto óptimo tenemos

8.1836e+16 que es totalmente lejano al punto inicial (aproximación gráfica del punto óptimo) que ingresamos. Por lo tanto, sobre los valores de la función z de la Tabla 4.15, se verifica que la función “fmincon” proporciona un resultado dentro de la superficie de interés, ya que al permitir la introducción de restricciones para la optimización se condiciona al comportamiento de la función y conduce al *solver* a buscar el óptimo exclusivamente entre estos límites.

Tabla. 4.13.

Comparación de Resultados del proceso de optimización obtenidos mediante Líneas de Comandos (LC) e Interfaz Gráfica (IG).

Cuadro Comparativo			
Caso	Nº de Restricciones	fval = z = f(x,y) (Número de Nodos óptimo)	
		LC	IG
1 (fminunc)	Ninguna	8.1836e+16	8.1836e+16
2 (fmincon)	Límites x y y	9.9230	9.9230
2 (fmincon)	1 (Desigualdad no lineal)	9.9230	9.9230
3 (fmincon)	3 (Desigualdad no lineal)	10.8426	10.8426

Se tiene un valor máximo aproximado de la función de 10 que representa el mínimo número de nodos, es decir, si se planificara un futuro despliegue en sitio de una WSN habrá de reducir 29 nodos, con esto se minimizaría costos, energía de la red y se redujera el trabajo de la instalación física de los sensores sobre la superficie del Volcán. El resultado gráfico obtenido se muestra en la siguiente Figura 4.11.

La representación gráfica del resultado permite visualizar la posición del óptimo (Mínimo) sobre la superficie y de este modo validar que exclusivamente se trata de un óptimo local ubicado en la región acotada por las restricciones de desigualdad, a las que está sujeta la función objetivo.

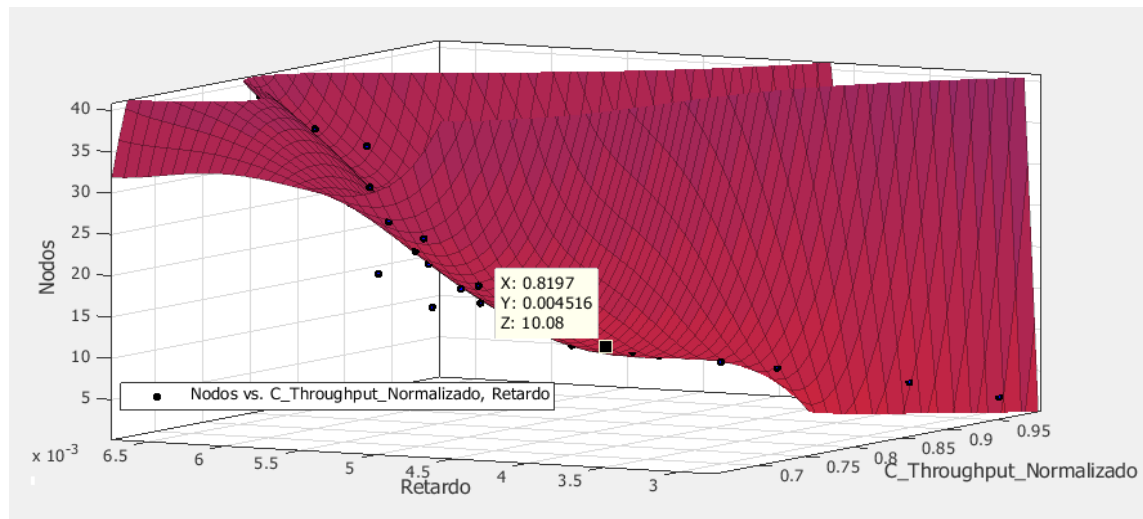


Figura. 4.11. Resultado gráfico de Optimización Multivariable

Si analizamos los tipos de algoritmos matemáticos presentados en la Tabla 4.14 que emplea la función, “fmincon” verificamos que a diferencia de “fminunc”, éste aplica uno adicional el algoritmo SQP (Programación cuadrática secuencial), ya que se requiere resolver un problemas donde la función objetivo y las restricciones son dos veces continuamente diferenciable, también con esto se logra minimizar el tiempo de respuesta ya que realiza menos iteraciones. Por los motivos planteados se determina que esta función es una mejor alternativa para optimizar funciones no lineales multivariable.

CAPÍTULO 5

DISCUSIÓN

5.1 PRINCIPALES CONTRIBUCIONES

- La contribución más significativas del presente trabajo de investigación, es la metodología desarrollada y empleada para encontrar el óptimo número de nodos en una red de sensores inalámbricos para la monitorización volcánica de señales sísmicas, que maximice la variable del rendimiento η y minimice las variables de PL y EED. Este proceso consistió en extraer los valores de las métricas QoS mencionadas de los 34 escenarios simulados, y en base a éstas se dedujo una expresión algebraica polinómica definida como función objetivo, el orden del polinomio se determinó mediante un análisis de estadísticas de bondad de ajuste (Error Cuadrático Medio, Suma de Errores Cuadrados y R-cuadrado Ajustado), seguido se introdujo dicha función al *solver* y mediante algoritmos de optimización se obtuvo el resultado numéricos dentro de los valores que se estimaba.
- Este proyecto viene siendo una contribución importante a las investigaciones que se han venido ejecutando durante años por el Centro de Investigaciones de Redes Ad-Hoc “CIRAD”, del Departamento de Eléctrica y Electrónica de la Universidad de las Fuerzas Armadas – ESPE, entorno a la monitorización de señales sísmicas del Volcán Cotopaxi, como un respaldo más a las investigaciones realizadas bajo el estándar IEEE 802.15.4 se desarrolló un modelamiento de optimización que se aplicó para la minimización de nodos de una WSN, sin embargo se podría utilizar como guía para resolver cualquier otro problema de optimización relacionado.

5.2 CONCLUSIONES

- En Base al estudio del marco conceptual de las WSN, el estándar Zigbee/IEEE 802.15.4 y las características físicas y lógicas de los módulos XBee-PRO (S2), se demostró que estos dispositivos frente a aplicaciones de Monitorización Volcánica de variables ambientales, brindan resultados que satisfacen los requisitos de *Hardware* necesarios para un posterior despliegue en escenarios reales, y ofrece prometedores avances para futuros proyectos dentro del Departamento.
- Con la implementación de una distribución de nodos sensores sobre la red de tipo aleatoria y generación de un evento sísmico Volcánico de Largo Período, se logró establecer un ambiente de simulación más real ya que no depende de un patrón de implementación uniforme, y se acoge más a las características del terreno Volcánico, esto conllevó a obtener información más efectiva la cual validó el desempeño de la WSN. Además, si se planificara un futuro despliegue en escenario real la implementación física de la red sobre la superficie del Volcán sería más factible.
- Como resultado de utilizar la *Toolbox* Curve Fitting de MATLAB® para obtención de la función de objetivo y mediante el análisis de estadísticas de bondad de ajuste se encontró que dicha función corresponde a un polinomio de cuarto orden, por lo tanto se tiene un problema de optimización no lineal multivariable, es así que, mediante el estudio y evaluación de los métodos de ONL proporcionados por la *Toolbox* de optimización de MATLAB® para problemas de optimización de funciones univariables y multivariables, se determinó que la función “fmincon” en comparación a “fminbnd” y “fminunc” ejecuta menos iteraciones y emplea algoritmos de optimización más complejos pero efectivos, que buscaron el punto óptimo (mínimo) en la región acotada por las restricciones de desigualdad.
- Con el desarrollo del proyecto se cumplió el objetivo principal de encontrar el óptimo número de nodos, mediante el análisis de las métricas QoS y evaluación de los métodos de optimización de funciones no lineal (Cuasi - Newton, SQP, Búsqueda de la sección Dorada, e Interpolación Parabólica), permitieron encontrar

un valor óptimo aproximado de 10 nodos sensores que de forma particular resulta válido solo para el escenario planteado, donde el nivel de η de la WSN corresponde a 65.72kbps, el EED promedio mínimo es 5.15 ms y la PL es igual a 2093, correspondiente al 11% de pérdidas de paquetes debido al uso de restricciones de acotación de la región de búsqueda, por lo tanto se garantiza que la PL efectivamente sea menor al 25%.

5.3 TRABAJOS FUTUROS

- Para comprobar la eficiencia del modelado de optimización creado en este proyecto, es recomendable hacer uso de métodos de optimización numérica más complejos existentes en la misma herramienta tal como “*patternsearch*” sobre el mismo problema planteado, pues este método encuentra el mínimo de una función utilizando un patrón de búsqueda (ps) que no requiere el gradiente del problema para ser optimizado y se puede utilizar en funciones que no son continuas o diferenciables. Además existen herramientas de optimización cada vez más sofisticadas en las últimas versiones de las hojas de cálculo de utilización generalizada de Excel, y a diferencia de la herramienta utilizada en este trabajo, esta nueva herramienta no requiere una función objetivo algebraica que relacione las variables a optimizar, sino que utiliza los datos numéricos de las variables directamente sobre las celdas especificadas para las variables de decisión y la función objetivo del *solver*, por lo tanto esto abre una nueva visión hacia un posible trabajo futuro, que consista inicialmente en modelar el problema de optimización resuelto en este trabajo por medio del *solver* de optimización de Excel, permitiendo comparar y validar los resultados a obtener. Finalmente, con la validación del resultado de este trabajo por al menos un método o herramienta más se podría en un futuro realizar pruebas de *test-bed* en sitio de un escenario con 10 nodos sensores con el objetivo de corroborar que efectivamente con esa cantidad de nodos se alcanza un *Throughput* estable y un mínimo retardo.
- La relevancia de los problemas de optimización en el mundo tecnológico y empresarial ha generado la introducción de diversos tipos de herramientas de optimización cada vez más eficaces sobre diferentes plataformas de *software*,

constituyendo una alternativa para solucionar problemas de optimización de cualquier índole, tomando en cuenta lo mencionado y considerando que el consumo energético es una de las métricas que actúa como uno de los principales indicadores del desempeño de la red, se recomienda en un futuro plantear un problema de optimización en relación al comportamiento de este parámetro, frente al rendimiento (*Throughput*) y al número de nodos de una WSN sobre los escenarios generados en este trabajo o similares, al resolver dicho problema se podría llegar a alargar el tiempo de vida útil de las baterías de cada nodo sensor, obteniendo una mejora en la eficiencia energética de una red en funcionamiento. Con el desarrollo de un posterior proceso de optimización se condicionaría las tres variables mencionadas, y en caso de que las variables representen una curva o superficie convexa se podría hacer uso de la herramienta CVX de Matlab, que se utiliza exclusivamente para problemas de optimización convexa u optimización LMI (desigualdad de la matriz lineal).

REFERENCIAS BIBLIOGRÁFICAS

- [1] I. F. Logroño, *Implementación de un Sistema de Monitorización de Señales Sísmicas del Volcán Cotopaxi Empleando una Red de Sensores Inalámbricos*, Escuela Politécnica del Ejército, 2011.
- [2] J. V. Hernández, *Redes inalámbricas de sensores: Una nueva arquitectura eficiente y robusta basada en jerarquía dinámica de grupos*, Universidad Politécnica de Valencia, 2010.
- [3] C. N. Chio, *Redes de sensores inalámbricos*, II Congreso Internacional de Ingeniería Mecatrónica – UNAB, 2009.
- [4] D. C. Restrepo, D. A. Ovalle y A. J. Montoya, 'Manejo de bases de datos en redes de sensores inalámbricos', *Revista Avances en sistemas e informática*, vol. VI, nº 1, 2009.
- [5] A. Los Santos, *Aplicación de las Redes de Sensores en el entorno vehicular*, Universidad de Vigo, España, 2009.
- [6] J. A. Moñino, *Aplicación del Modelado Específico de Dominio a las Redes de Sensores Inalámbricos*, Universidad Politécnica de Cartagena, 2007.
- [7] G. Werner, M. Welsh, M. Ruiz, J. Johnson, M. Ruiz y J. Lees, *Monitoring Volcanic Eruptions with a Wireless Sensor Network*, Harvard University, University of New Hampshire & University of North Carolina, 2004.
- [8] Moteiv Corporation , 'Projects: Tmote Sky', TIK Computer Engineering Group, 13 Noviembre 2006. [En línea]. Available: <http://www.snm.ethz.ch/Projects/TmoteSky>. [Último acceso: 23 Abril 2015].
- [9] G. Werner, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz y J. Lees, 'Deploying a Wireless Sensor Network on an Active Volcano', *IEEE Internet Computing*, pp. 18-25, 2006.

- [10] S. F. Saigua y M. V. Villafuerte, *Evaluación de las Topologías Físicas de WSN Mediante la Implementación de un Prototipo de Medición de Variables Ambientales para el "G.E.A.A-ESPOCH"*, Escuela Superior Politécnica de Chimborazo, 2014.
- [11] E. E. Flores, *Redes de Sensores Inalámbricas Aplicado a la Medicina*, Universidad de Cantabria, 2012.
- [12] D. Gascón, 'Redes de Sensores Inalámbricos- la tecnología invisible', *Revista MIT Technology Review*, pp. 53-55, 2010.
- [13] G. P. Ricardo y J. Mora, *Guía para la implementación de redes de sensores inalámbricos*, Universidad Pontificia Bolivariana, 2010.
- [14] A. B. Corral, *Diseño e implementación de un entorno de simulación para redes de sensores inalámbricos*, Colombia: Universidad Politécnica de Cartagena, 2005.
- [15] J. Mayné, *IEEE 802.15.4 y ZigBee*, Silica an Avnet Company, 2007.
- [16] C. L. Ramiro, *Diseño de una arquitectura para redes de sensores con soporte para aplicaciones de detección de eventos*, Universidad Politécnica de Valencia, 2012.
- [17] O. Ortiz, *Caracterización de calidad de servicio en redes inalámbricas de sensores*, Universidad Politécnica de Madrid, 2010.
- [18] E. S. Reinoso, *Diseño e implementación de un prototipo para una red de domótica y seguridad para un hogar utilizando el estándar IEEE 802.15.4 ZigBee*, Quito: XXII Jornadas en Ingeniería Eléctrica y Electrónica Escuela Politécnica Nacional, 2008.
- [19] W. E. Barriga y F. P. Zúñiga, *Tecnologías Inalámbricas de Corto Alcance Estándar ZigBee y Bluetooth*, Quito: Universidad Politécnica Salesiana, 2006.
- [20] G. Campos, *Caracterización Del Consumo en Redes ZIGBEE/802.15.4*, Universidad de Málaga, 2009.

- [21] S. Serna, 'Especificación de perfil zigbee para monitoreo y control de plantas industriales', *Revista Tecnológica, Instituto Tecnológico Metropolitano*, n° 23, pp. 172-175, 2009.
- [22] J. P. Dignani, *Análisis Del Protocolo Zigbee*, Universidad Nacional de la Plata, 2011.
- [23] IEEE Computer Society, *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, LAN/MAN Standards Committee, New York, 2006.
- [24] K. Lakshminarayanan, V. Padmanabhan y J. Padhye, 'Bandwidth Estimation in Broadband Access Networks', Octubre 2004. [En línea]. Available: <http://www.aqualab.cs.northwestern.edu>. [Último acceso: 23 Marzo 2015].
- [25] D. P. Tobón y N. G. Gómez, 'Análisis de métricas de calidad de servicio para la configuración del protocolo CSMA/CA en redes de sensores inalámbricas de área corporal', *Ingeniería y Desarrollo*, vol. XXX, n° 1, pp. 101-124, 2012.
- [26] A. Cánovas, *Diseño y Desarrollo de un Sistema de Gestión Inteligente integrado de servicios de IPTV estándar estereoscópico y HD basado en QoE*, Universidad Politécnica de Valencia, 2013.
- [27] Barcelona, Universitat Pompeu Fabra, 'Máximos y mínimos de funciones de varias', [En línea]. Available: <http://www.dtic.upf.edu/~gharo/anum/apunts/hessian.pdf>. [Último acceso: 25 Marzo 2015].
- [28] R. Cantu, *Programación no Lineal*, Universidad Autónoma de Nuevo León, 1996.
- [29] J. A. Caballero, *Introducción a la optimización numérica*, Universidad de Alicante, 2011.
- [30] J. García, *Tutorial de Analisis Numérico-Interpolación: Fórmulas de Newton en diferencias finita*, Universidad de Las Palmas de Gran Canaria, 2000.
- [31] D. Valle, *Método de la secante*, Universidad Autónoma del Estado de México, 2007.

- [32] M. C. Vélez, C. A. Castro y J. Maya, 'Algoritmo de búsqueda aleatoria para la programación de la producción en un taller de fabricación', *Revista Universidad EAFIT*, vol. XXXIX, n° 131, pp. 76-86, 2003.
- [33] J. C. Raimundez, *Métodos Modernos*, Gallego: Universidad de Vigo, 2005.
- [34] J. A. Conesa, Curso básico de análisis térmico, Editorial Club Universitario, 2000.
- [35] J. A. Caballero, *Métodos numéricos de optimización: problemas de optimización sin restricciones*, Universidad de Alicante, 2011.
- [36] J. M. Oviedo, *Optimización Multivariante*, 2003.
- [37] E. Garza, *Optimización Con Restricciones de Igualdad*, Tecnológico de Monterrey, 2009.
- [38] Universidad Técnica Federico Santa María, *Máximos y mínimos mediante el método de Lagrange*, 2010.
- [39] E. J. Cobos, *Estudio de las Rdes Sensoriales como una Nueva Alternativa de Comunicación Inalámbrica*, Escuela Politécnica Del Ejército, 2007.
- [40] M. B. Tituaña, *Implementación de un Prototipo de Comunicaciones Mediante la utilización del Estándar IEEE 802.15.4*, Escuela Politécnica del Ejército, 2010.
- [41] Digi International Company, *XBee®/XBee-PRO® ZB RF Modules*, 2012.
- [42] H. S. Martinez, *Evaluación y Análisis de un Prototipo de Sensores Inalámbricos para la Monitorización de infraestructuras Críticas De Puentes*, Quito: Universidad de las Fuerzas Armadas - ESPE, 2014.
- [43] M. Greis, *Tutorial for the Network Simulator NS*, VINT group, 2006.
- [44] H. D. Ortiz, *Estudio de los Efectos de Sitio para la Construcción de un Índice de Actividad Sísmica en el Volcán Cotopaxi*, Quito: Escuela Politécnica del Ejército, 2013.
- [45] A. E. Cordova, *Estudios de Micro-Sismicidad para Proyectos Geotérmicos:*

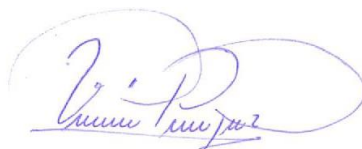
- Chacana y Chachimbiro*, Escuela Politécnica Nacional, 2013.
- [46] F. Borda, *Actividad Ondas Sísmicas*, Institución Educativa técnica San Bartolome, 2013.
- [47] Nariño - Proyecto Cooperativo Internacional , La transición océano continente en el suroeste de Colombia, Tomo II, Colombia: Guadalupe Ltda., 1973.
- [48] T. Utsu, Y. Ogata y R. S. Matsu'ura, *The Centenary of the Omori Formula for a Decay Law of Aftershock Activity*, Japan: Institute of Statistical Mathematics, Minato-ku, Tokyo 106., 1995.
- [49] J. L. Millones, 'Localización de un Sismo Utilizando una Estación de Tres Componentes', 2005. [En línea]. Available: http://sisbib.unmsm.edu.pe/bibvirtual/monografias/Basic/millones_jj/contenido.htm. [Último acceso: 05 Diciembre 2014].
- [50] J. R. Orellana, 'Resumen de la Actividad del Volcan Cotopaxi Durante en año 2009', 2015. [En línea]. Available: <http://www.igepn.edu.ec/informes-volcanicos/cotopaxi/cotopaxi-anuales/2009-10>. [Último acceso: 2 Abril 2015].
- [51] M. C. Acosta, *Estudio del Estandar IEEE 802.15.4 "ZIGBEE" para Comunicaciones Inalámbricas de Área Personal de Bajo Consumo de Energía y su comparación con el estandar IEEE 802.15.1 "BLUETOOTH"*, Quito: Escuela Politécnica Nacional, 2006.
- [52] MathWorks, Inc, 'Curve Fitting Toolbox', 2015. [En línea]. Available: http://in.mathworks.com/help/pdf_doc/curvefit/curvefit.pdf. [Último acceso: 6 Abril 2015].
- [53] I. Cabezas y J. Páez, *Matlab Toolbox de optimización Aplicaciones en ciencias económicas*, Bogotá: Facultad de Ciencias Económicas Universidad Nacional de Colombia, 2010.
- [54] G. M. Vizcaíno, 'Automatización del Proceso de Diseño de un Circuito Electrónico con la Herramienta MATLAB y el simulador de Circuitos Spectre', Marzo 2006. [En línea]. Available: <http://bibing.us.es/proyectos/>. [Último acceso: 10 Abril 2015].

-
- [55] MathWorks, Inc, 'Optimization Toolbox', 2015. [En línea]. Available: <http://www.mathworks.com/help/optim/index.html>. [Último acceso: 2 Abril 2015].
- [56] S. Abad, 'Detección de Anomalías Cardíacas con Aprendizaje Automático (Machine Learning)-Análisis con Regresión Logística (Logistic Regression) ', 15 Enero 2014. [En línea]. Available: <http://samuelabad1991.blogspot.com>. [Último acceso: 4 Abril 2015].
- [57] E. S. Reinoso, *Diseño e Implementación de un Prototipo para una Red de Domótica y Seguridad para un Hogar Utilizando el estándar IEEE 802.15.4 "Zigbee"*, Quito: Escuela Politécnica Nacional, 2008.

ACTA DE ENTREGA

El presente Proyecto de grado fue entregado al Departamento de Eléctrica y Electrónica y reposa en la Universidad de las Fuerzas Armadas – ESPE desde:

Sangolquí, 25 de Mayo de 2015

ELABORADO POR:

VERÓNICA ISABEL POAQUIZA YUMBULEMA

CI: 050286457-2

AUTORIDAD:


Ing. Paúl Bernal
**DIRECTOR DE LA CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**