



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

EVALUACIÓN DE ESTÁNDARES DE COMUNICACIÓN EN UNA RED DE SENSORES INALÁMBRICOS PARA LA DETECCIÓN DE INUNDACIONES EN EL AFLUENTE DEL RÍO DAMAS DE ALLURIQUÍN.

ING. ABEL ISAÍAS GALLEGOS ÁLAVA

Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo,
presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH,
como requisito parcial para la obtención del grado de:

MAGISTER EN SISTEMAS DE TELECOMUNICACIONES

Riobamba-Ecuador

Enero, 2019



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DEL TRABAJO DE TITULACIÓN CERTIFICA QUE:

El **Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo**, titulado: “EVALUACIÓN DE ESTÁNDARES DE COMUNICACIÓN EN UNA RED DE SENSORES INALÁMBRICOS PARA LA DETECCIÓN DE INUNDACIONES EN EL AFLUENTE DEL RÍO DAMAS DE ALLURIQUÍN”, de responsabilidad del Sr. Abel Isaías Gallegos Álava, ha sido prolijamente revisado y se autoriza su presentación.

Tribunal:

Ing. Oswaldo Geovanny Martínez Guashima. M. Sc.

PRESIDENTE

FIRMA

Ing. Gonzalo Fernando Olmedo Cifuentes. PH. D.

DIRECTOR DE TESIS

FIRMA

Ing. Freddy Roberto Acosta Buenaño. M. Sc.

MIEMBRO DEL TRIBUNAL

FIRMA

Ing. Tony Fernando Flores Pulgar. M. Sc.

MIEMBRO DEL TRIBUNAL

FIRMA

Riobamba, Enero 2019.

DERECHOS INTELECTUALES

Yo, Abel Isaías Gallegos Álava, declaro que soy responsable de las ideas, doctrinas y resultados expuestos en el **Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo**, y que el patrimonio intelectual generado por la misma pertenece exclusivamente a la Escuela Superior Politécnica de Chimborazo.

Abel Isaías Gallegos Álava
No. Cédula: 171821031-1

©2019, Abel Isaías Gallegos Álava

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

DECLARACIÓN DE AUTENTICIDAD

Yo, Abel Isaías Gallegos Álava, declaro que el presente proyecto de investigación es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación de Maestría.

Abel Isaías Gallegos Álava
No. Cédula: 171821031-1

DEDICATORIA

Dedico este trabajo a mis padres Rafael Gallegos y Narcisa Álava quienes con su paciencia y consejos me han guiado para ser mejor cada día; a mi Tía Rebeca Gallegos quien estuvo siempre pendiente como mi segunda madre en el transcurso de toda la maestría. A mi familia por su confianza, a mis amigos por sus consejos y a mi novia Liseth por su apoyo incondicional y su paciencia.

Abel I. Gallegos A.

AGRADECIMIENTO

A Dios por permitirme culminar un paso más en el camino profesional y siempre darme sus bendiciones y protección.

A mis padres por siempre estar pendientes de mí y ser el pilar fundamental en mi vida.

Al Ing. Gonzalo Olmedo PhD. Por siempre guiarme con sus consejos durante el desarrollo del proyecto.

Al Ing. Tony Flores e Ing. Freddy Acosta, por el total apoyo recibido y desinteresado.

Al Ing. Oswaldo Martínez, por su predisposición y guía desde el inicio hasta la culminación de la maestría.

A los docentes que fueron parte de la Maestría en Sistemas de Telecomunicaciones segunda cohorte, que dentro y fuera del aula impartieron su conocimiento y experiencia.

Abel I. Gallegos A.

CONTENIDO

RESUMEN	xiii
ABSTRACT	xiv
CAPÍTULO I	1
1. PLANTEAMIENTO DEL PROBLEMA	1
1.1 Antecedentes	1
1.2 Problema de Investigación	2
1.3 Formulación del Problema	3
1.4 Sistematización del Problema	3
1.6 Objetivos de la Investigación	4
1.6.1 Objetivo General	4
1.6.2 Objetivos Específicos	4
1.7 Hipótesis	5
CAPÍTULO II	8
2. MARCO TEÓRICO	8
2.1 Redes De Sensores Inalámbricas	8
2.1.1 Nodos Sensores	9
2.1.1.1 Microcontrolador	9
2.1.1.2 Sistema de Radio	9
2.1.1.3 Sensor	10
2.1.1.4 Sistema de Alimentación de Energía	10
2.1.1.5 Gateway	10
2.1.1.6 Estación Base	10
2.2 Topologías típicas utilizadas en las RSI	11
2.3 Estándares De Comunicación	13
2.3.1 IEEE 802.15.1 Bluetooth	13
2.3.1.1 Modo de Operación BR/EDR	15
2.3.1.2 Modo de Operación Bluetooth Low Energy	16
2.3.1.3 Modo de Operación AMP	18
2.3.2 IEEE 802.11n WI-FI	20
2.3.2.1 Nivel Físico	20
2.3.2.2 Nivel de Acceso al Medio	21
2.3.2.3 Protocolos de Control de Acceso al Medio	21
2.3.3 IEEE 802.15.4 ZIG-BEE	22

2.3.3.1	<i>Arquitectura del dispositivo de la LR-WPAN</i>	24
2.3.3.2	<i>Capa Física de IEEE 802.15.4</i>	25
2.3.3.3	<i>Aplicaciones de Tecnología Zigbee</i>	25
2.4	Tipos de Sensores	25
2.4.1	<i>Sensor de Nivel de Río (Moradel Fonseca)</i>	25
2.4.2	<i>Sensor de Nivel de Líquidos tipo radar</i>	26
2.4.3	<i>Sensor de Nivel Seba PULS 15</i>	27
2.4.4	<i>Sensor e+ WATER L</i>	27
CAPÍTULO III		29
3.	ESCENARIO Y SIMULADOR	29
3.1	Escenario	29
3.2	Simuladores	31
3.3	Network Simulator 3 (NS3) – Generalidades	32
3.3.1	<i>La Web</i>	33
3.3.2	<i>Mercurial</i>	33
3.3.3	<i>WAF</i>	33
3.3.4	<i>Ambiente de desarrollo</i>	33
3.3.5	<i>Condiciones para la ejecución del simulador NS3</i>	34
CAPÍTULO IV		35
4.	SIMULACIÓN	35
4.1	Selección del Estándar	35
4.1.1	<i>Simulación 802.11n Wi-Fi</i>	35
4.1.2	<i>Simulación 802.15.1 Bluetooth</i>	36
4.1.3	<i>Simulación 802.15.4 Zigbee</i>	37
4.2	Elección de Escenarios	38
4.2.1	<i>Escenario 1 - Número de Nodos 29, Distancia entre nodos 100m (OLSR)</i>	39
4.2.2	<i>Escenario 2 - Número de Nodos 29, Distancia entre nodos 100m (AODV)</i>	40
4.2.3	<i>Escenario 3 - Número de Nodos 29, Distancia entre nodos 100m (DSDV)</i>	42
4.2.4	<i>Escenario 4 - Número de Nodos 15, Distancia entre nodos 200m (OLSR)</i>	43
4.2.5	<i>Escenario 5 - Número de Nodos 15, Distancia entre nodos 200m (AODV)</i>	45
4.2.6	<i>Escenario 6 - Número de Nodos 15, Distancia entre nodos 200m (DSDV)</i>	46
4.3	Análisis de Resultados	47
CONCLUSIONES		51
RECOMENDACIONES		52
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE FIGURAS

Figura 1-2	Esquema interno de un Nodo Sensor Inalámbrico	9
Figura 2-2	Correspondencia Estándares IEEE, Capa Física y MAC	13
Figura 3-2	Eventos de Aviso en Bluetooth Low Energy	17
Figura 4-2	Eventos de Aviso en Bluetooth Low Energy	19
Figura 5-2	Protocolos de Control de Acceso al Medio	22
Figura 6-2	Modelo de Capas ZIGBEE.....	23
Figura 7-2	Modelo de Capas ZIGBEE.....	24
Figura 8-2	Sensor de Nivel de río	26
Figura 9-2	Sensor tipo radar VEGAPULS 61	26
Figura 10-2	Sensor tipo varilla e+ water.....	27
Figura 11-3	Esquema Ruta de Red sobre Río Damas	29
Figura 12-3	Río Damas	30
Figura 13-3	Río Damas	31
Figura 14-3	Eclipse IDE	34
Figura 15-4	Comparación de Resultado Estándar 802.11n.....	35
Figura 16-4	Comparación de Resultado Estándar 802.15.1 BLE	36
Figura 17-4	Comparación de Resultado Estándar 802.15.4 ZIGBEE.....	37
Figura 18-4	Throughput OLSR - 29 Sensores - Estándar 802.11n distancia 100m.....	39
Figura 19-4	Archivo PCAP OLSR Nodo 1 Distancia 100m.....	39
Figura 20-4	Archivo XML OLSR Nodos 29 Distancia 100m en NetAnim.....	40
Figura 21-4	Throughput AODV - 29 Sensores - Estándar 802.11n distancia 100m	41
Figura 22-4	Archivo PCAP AODV Nodos 29 Distancia 100m.....	41
Figura 23-4	Throughput DSDV- 29 Sensores - Estándar 802.11n distancia 100m	42
Figura 24-4	Archivo PCAP DSDV Nodos 29 Distancia 100m	43
Figura 25-4	Throughput OLSR - 15 Sensores - Estándar 802.11n distancia 200m.....	44
Figura 26-4	Archivo PCAP OLSR Nodo 1 Distancia 200m.....	44
Figura 27-4	Throughput AODV - 15 Sensores - Estándar 802.11n distancia 200m	45
Figura 28-4	Archivo PCAP AODV Nodos 15 Distancia 200m.....	45
Figura 29-4	Throughput DSDV - 15 Sensores - Estándar 802.11n distancia 200m	46
Figura 30-4	Archivo PCAP DSDV Nodos 15 Distancia 200m	47

ÍNDICE DE TABLAS

Tabla 1-2: Comparativa Frecuencia ZIGBEE.....	22
Tabla 2-2: Comparación Sensores	28
Tabla 3-3: Comparación Simuladores de Redes	32
Tabla 4-4: Throughput de acuerdo a Escenarios.....	48
Tabla 5-4: Parámetros de Red (DELAY, TX BITRATE, RX BITRATE)	49
Tabla 6-4: Parámetros de Red (TX PACKETS, RX PACKETS, JITTER)	50
Tabla 7-4: Resumen Ponderaciones.....	50

ÍNDICE DE ANEXOS

ANEXO A	Información confidencial proporcionada por el GAD Municipal Santo Domingo
ANEXO B	Fotos del Río Damas
ANEXO C	802.15.1 BLUETOOTH – Programación NS3
ANEXO D	802.11n WIFI Programación NS3
ANEXO E	802.15.4 ZIGBEE - Programación NS3
ANEXO F	Escenarios Programación NS3

RESUMEN

El objetivo de este proyecto fue evaluar a nivel de Red los estándares de comunicación 802.15.1 Bluetooth, 802.15.4 Zigbee y 802.11n Wi-Fi, a través de la simulación de una de Red de Sensores Inalámbricos (RSI). Se identificaron los componentes principales de una RSI, los tipos de sensores utilizados en un Sistema de Alerta Temprana (SAT) para la detección de inundaciones en Ríos, las topologías de red, el nivel físico, de acceso al medio y de red de los estándares de comunicación antes listados, adicional se verificó el SAT de Alluriquín actual, mismo que funciona a través de 4 nodos sensores y la red GSM. Se realizaron visitas técnicas al afluente del Río Damas en la Parroquia Alluriquín de Santo Domingo, y se obtuvieron los parámetros de Red a ser utilizados en la simulación. Se escogió al simulador Network Simulator 3 (NS3) que permitió comparar los estándares antes mencionados. Como resultado de la simulación se pudo observar que el protocolo 802.11n tiene mejores características de transmisión a mayor distancia y throughput, en comparación con los estándares 802.15.1 y 802.15.4. Luego se propuso seis distintos escenarios de red para escoger el mejor protocolo de ruteo, configurando los parámetros: protocolos (OLSR - *Optimized Link State Routing Protocol*, AODV - *Ad hoc On-Demand Distance Vector*, DSDV - *Destination-Sequenced Distance-Vector Routing*), protocolo de transporte (*UDP - User Datagram Protocol*), modo de propagación, tamaño de paquetes de datos, número de nodos, distancia entre nodos, entre otros. Se recomienda una nueva RSI a través del estándar 802.11n utilizando el protocolo AODV, con una distancia entre nodos de 100m y 29 nodos sensores, como optimización de transmisión del actual SAT de Alluriquín, permitiendo la redundancia de nodos y la no dependencia de la red GSM para la transmisión de los datos.

Palabras claves: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <TELECOMUNICACIONES>, <RED DE SENSORES INALÁMBRICOS (RSI)>, <NETWORK SIMULATOR 3 (NS3) >, < SISTEMA DE ALERTA TEMPRANA (SAT) >, < ESTÁNDARES DE TELECOMUNICACIONES>

ABSTRACT

The present investigation was carried out with the objective for evaluating at the Network level the communication standards 802.15.1 Bluetooth, 802.15.4 Zigbee and 802.11n Wi-Fi, through the simulation of a Wireless Sensor Network (WSN). The main components from WSN were identified as the type of sensors used in an Early Warning System (EWS) for detection of floods in rivers, the network topologies, at physical level, access to the medium and network of standards of communication before listings. Additional, it was verified the EWS from Alluriquín current; same that works through 4 sensor nodes and Global System for Mobile Communication (GSM). The network Technical visits made to the affluent at "Damas" River in the parish of Alluriquín in Santo Domingo, and the parameters of the network were obtained to be used in the simulation. The simulator Network Simulator 3 (NS3) was chosen to compare the aforementioned standards. As a result of the simulation, it was observed that the 802.11n protocol has better transmission characteristics at greater distance and throughput, in comparison with the 802.15.1 and 802.15.4. Then, it was proposed 6 different network scenarios for choosing the best routing protocol, configuring the parameters: protocols (OLSR- Optimized Link State Routing Protocol, AODV - Ad hoc On-Demand Distance Vector, DSDV - Destination-Sequenced Distance-Vector Routing) , transport protocol (UDP - User Datagram Protocol), mode of propagation, size of data packets, number of nodes, distance between nodes, and so on. It is recommended a new WSN through the 802.11n standard using the AODV protocol, with a distance between two nodes of 100m and 29 sensor nodes, as transmission optimization of the current EWS from Alluriquín, allowing the redundancy of nodes and the non-dependence of GSM network for the transmission of data.

Keywords: TECHNOLOGY AND SCIENCE OF ENGINEERING, TELECOMMUNICATIONS, WIRELESS SENSORS NETWORK (WSN), NETWORK SIMULATOR 3 (NS3), EARLY WARNING SYSTEM (EWS), TELECOMMUNICATIONS STANDARDS.

CAPÍTULO I

1. PLANTEAMIENTO DEL PROBLEMA

1.1 Antecedentes

Según el Plan Nacional del Buen Vivir objetivo 3 hace referencia a que “... Ecuador es un país con un elevado índice de vulnerabilidad ante factores de origen natura y antrópico. Durante 1900 y 2009 se registraron 65 desastres de gran magnitud (CRED, 2013) el 60%, provocado por fenómenos hidro-meteorológicos (sequías, inundaciones, deslizamientos húmedos) y el 40%, por eventos geofísicos (sismos, erupciones volcánicas y deslizamientos secos). Los sismos generaron pérdidas económicas de USD 1 500 millones durante el periodo.

Durante los últimos 25 años, grandes desastres afectaron los países de la región andina. Según la CEPAL (2003), el 33% de las pérdidas directas e indirectas (vidas humanas, infraestructura social y productiva) en la región fue causado por eventos naturales. En nuestro país, el fenómeno de El Niño, durante los episodios más severos (1982- 1983 y 1997-1998), causó pérdidas estimadas por USD 3 900 millones” (Senplades, 2013)

La gobernación de Santo Domingo de los Tsáchilas a través de su página web oficial indica con fecha de 26 de abril de 2016 que: “Aproximadamente a las 12h00 de la madrugada de hoy la parroquia Alluriquín perteneciente al cantón Santo Domingo sufrió una emergencia ocasionada por el desbordamiento del río Damas. A la 01h00 la gobernadora de la provincia, Doris Merino Figueroa, activó el Comité de Operaciones de Emergencia (COE) Provincial para de inmediato brindar el contingente necesario.

Técnicos de la Secretaría de Gestión de Riesgos (SGR) llegaron hasta el lugar para analizar los daños causados por el desbordamiento y para evacuar a las personas que se encontraban en riesgo. Hasta el momento aproximadamente 200 personas han sido evacuadas del lugar, 40 casas fueron destruidas por completo y existen cuatro fallecidos y un desaparecido” (Tsáchilas, 2016)

El presente proyecto pretende realizar la evaluación de los estándares de comunicación en la red de sensores inalámbricos para la detección de inundaciones en el afluente del río damas en

Alluriquín. Con esto se pretende alertar y crear planes de contingencia para prevenir desastres que tengan afectaciones principalmente a la vida humana de la población de Alluriquín zona urbana y por quienes a diario transitan por este sector de la vía Atoag Santo Domingo.

Actualmente el GAD Municipal de Santo Domingo, posee la monitorización del afluente del río damas a través de tres sensores ubicados según el Anexo A, adjunto al presente documento. Estos sensores determinan el nivel del agua, la temperatura del ambiente, la humedad del aire y el acumulado de lluvia. Se dispone del acceso remoto vía una IP pública que da el acceso a cada estación que debe ser monitorizada por el sistema de gestión de riesgos de Santo Domingo y por el ECU-911, institución encargada de brindar soporte ante las emergencias suscitadas dentro de la provincia.

1.2 Problema de Investigación

En el afluente del río damas en Alluriquín parroquia rural del cantón Santo Domingo, existe el Sistema de Alerta Temprana (SAT) por sus siglas, el mismo consta de tres estaciones, cada una de las cuales posee 4 (cuatro) dispositivos sensores: nivel de agua, pluviómetro, humedad, temperatura y 1 (un) dispositivo tipo cámara de video. Dichas estaciones se encuentran distanciadas la primera a 200 m de Alluriquín, la segunda a 4,7 km y la tercera a 8,7 km. Cada estación además de componerse de los sensores tiene un módulo de comunicación de datos el mismo que internamente tiene dos routers GSM¹, el primero para la transmisión de datos de los sensores y el segundo para el tráfico de la cámara de video. Las estaciones se comunican con un servidor ubicado en las instalaciones del Municipio en el parque central de Santo Domingo, a través de la red GSM de Movistar. En el servidor se tienen tres tipos de alertas: normal, preventivo y evacuación inmediata. Cada estación envía las alertas en un lapso de 5 minutos. Estas alertas con los datos de los sensores son grabadas en el servidor en una Base de datos. Adicional el ECU-911 de Santo Domingo tiene el acceso a dichas alertas y datos en la IP² 200.105.237.150, proporcionando un usuario y contraseña para poder ingresar a dicho sistema. Las estaciones utilizan baterías recargables y paneles solares para su funcionamiento.

En dicho sistema de alerta temprana existen tres estaciones donde cada una posee un sólo sensor de nivel de agua, por lo que hay la posibilidad de que alguno falle con lo que no podría llevarse a cabo un control verdadero y eficaz de los niveles del río, consecuentemente el probable desborde del mismo sin su respectiva notificación.

¹ Global System for Mobile

² Internet Protocol

1.3 Formulación del Problema

Dentro del Actual Sistema de Alerta Temprana para la detección de inundaciones en el afluente del río Damas, existen tres sensores los cuales se detallan en el anexo A. No existe redundancia, por lo que si alguno de los tres sensores fallan, no se verificaría el nivel de agua en el río y por lo tanto la alerta ante la posible inundación no se realizaría. Las consecuencias de este fallo podrían desembocar en inundaciones como la sucedida el día 26 de abril de 2016, provocando incluso la pérdida de vidas humanas. (Tsáchilas, 2016)

¿Se puede mejorar el sistema de Alerta temprana para detectar inundaciones en el afluente del río damas de Alluriquín, a través de la propuesta de una red de sensores inalámbricos y la evaluación de los estándares de comunicación 802.11n, 802.15.1 y 802.15.4, permitiendo una red con mayor estabilidad en relación a la posible falla de un sensor, con el menor retardo y menor consumo de energía?

1.4 Sistematización del Problema

- a) ¿Qué magnitud está siendo registrada y monitorizada en cada nodo sensor en el actual Sistema de Alerta Temprana?
- b) ¿Qué tipo de sensor está siendo utilizando?
- c) ¿Qué problemas se presentarían en caso de fallo de un sensor del Actual Sistema de Alerta Temprana?
- d) ¿Cuál es el escenario, topología y protocolo de comunicación de la red de sensores inalámbricos, que optimice la transmisión en tiempo real de los nodos sensores y la estación base del Sistema de Alerta Temprana?
- e) ¿Cuál de los estándares de comunicación propuestos, permite a la red de sensores inalámbricos: estabilidad, mayor redundancia, menor consumo de energía y menor retardo, tomando en cuenta la transmisión en tiempo real del Sistema de Alerta Temprana, para la detección de inundaciones en el río damas de Alluriquín?

1.5 Justificación de la Investigación

La presente investigación propone como objetivo principal evaluar estándares de comunicación en una red de sensores inalámbricos para la detección de inundaciones en el afluente del río damas de Alluriquín. El objetivo número 3 del Plan Nacional del Buen Vivir indica

“...Entre los derechos para mejorar la calidad de vida se incluyen el acceso al agua y a la alimentación (art. 12), a vivir en un ambiente sano (art. 14), a un hábitat seguro y saludable...” (Senplades, 2013). Además también hace referencia a lo siguiente dentro de la sección de Gestión de Riesgos del objetivo 3 “...Los desastres a nivel mundial se duplicaron en las dos últimas décadas y alcanzaron más de 400 por año. La vulnerabilidad ha aumentado en muchos países y Ecuador no es la excepción. Una de sus causas es el incremento de los riesgos urbanos debido a los elevados índices de urbanización, incluyendo asentamientos no planificados e inseguros en áreas inundables, costeras y montañosas...”. (Senplades, 2013).

La utilización de la tecnología para lograr la prevención y gestión de riesgos determinaría que la propuesta de este proyecto contribuiría al mejoramiento de la calidad de vida de la población de Alluriquín en la provincia Santo Domingo de los Tsáchilas, proponiendo una solución tecnológica aplicada a un problema real existente.

Una RSI³ es un tipo de red de comunicación en la que los principales equipos de red son sensores que tienen procesamiento interno y que además de conectarse entre ellos, permiten la medición de distintas magnitudes físicas en el ambiente por lo general de difícil acceso.

1.6 Objetivos de la Investigación

1.6.1 Objetivo General

Evaluar estándares de comunicación en una nueva propuesta de red de sensores inalámbricos a través de un simulador, que permita a un Sistema de Alerta Temprana la detección de inundaciones en el afluente del río damas de Alluriquín, optimizando recursos de la red.

1.6.2 Objetivos Específicos

- Fundamentar el tipo de datos a transmitir, su modo de propagación y las características de los sensores a ser utilizados dentro de la simulación, tomando como referencia las características de los sensores del actual Sistema de Alerta Temprana.
- Describir los componentes principales de la red de sensores inalámbricos y sus funciones dentro de la misma.

³ Red de Sensores Inalámbricos

- Diseñar la red de sensores inalámbricos y realizar la simulación de distintos escenarios de red en tiempo real, entre nodos y la estación base de la red, utilizando los estándares de comunicación 802.11n, 802.15.1 y 802.15.4.
- Analizar los resultados de las simulaciones de la red de sensores inalámbricos y determinar cuál es el mejor escenario y estándar para el intercambio de datos en la red tomando en cuenta su rendimiento, retardo, redundancia y throughput.
- Identificar mediante resultados de simulación cuál estándar de comunicación sería el propicio para la mayor estabilidad y el menor consumo de energía eléctrica de la red.

1.7 Hipótesis

a. Hipótesis general

La evaluación de los estándares de comunicación en una RSI para la detección de inundaciones en el afluente del río damas en Alluriquín, permitirá escoger una red más estable, con menor retardo y menor consumo de energía.

b. Hipótesis específicas

El Análisis del tipo de datos a transmitir, determinará qué sensores deberán ser utilizados en la RSI.

La construcción del marco teórico, permitirá aplicar la tecnología y técnicas existentes para determinar cuál es el estándar mejor aplicado para la transmisión de datos en la RSI.

La propuesta de una nueva red de sensores inalámbricos y la evaluación de los estándares de comunicación 802.11n, 802.15.1 y 802.15.4, con sus respectivos cambios en frecuencia y ancho de banda, permitirá la comunicación en tiempo real del sistema de Alerta Temprana, obteniendo: mayor rendimiento y throughput, menor retardo y consumo de energía en el proceso de transmisión y recepción de datos entre los nodos sensores, la estación base de la RSI, así como redundancia de sensores por posibles fallos.

c. Identificación de variables

Hipótesis general:

Variable independiente:

Evaluación de los estándares de comunicación en una red de sensores inalámbricos, que permita la comunicación en tiempo real de un Sistema de Alerta Temprana, para la detección de inundaciones en el afluente del río damas en Alluriquín

Variable Dependiente:

Diseñar la red y escoger un escenario con mayor rendimiento y *throughput*, con menor retardo y consumo de energía, así como redundancia de sensores por posibles fallos.

Hipótesis específica 1

Variable independiente:

Análisis de tipos de datos

Variable Dependiente:

Sensores a utilizar.

Hipótesis específica 2

Variable independiente:

El desarrollo del marco teórico adecuado.

Variable Dependiente:

Estándar mejor aplicado para la transmisión de datos en la RSI.

Hipótesis específica 3

Variable independiente:

Diseño de la red de sensores y evaluación de estándares de comunicación con sus respectivos cambios en frecuencia y ancho de banda.

Variable Dependiente:

Mayor rendimiento y *throughput*, y menor retardo y consumo de energía en el proceso de transmisión y recepción de datos entre los nodos sensores y la estación base de la red de sensores inalámbricos, así como redundancia de sensores por posibles fallos.

Operación de variables

Se recabará información de la simulación de la red de sensores inalámbricos, con distintos escenarios de red, los datos serán obtenidos en hojas de cálculo y gráficas que mostrarán la relación entre la distancia, y la eficiencia de la interconexión de los saltos de comunicación y transmisión entre los sensores, en base al *throughput*, retardo, redundancia y consumo de energía.

CAPÍTULO II

2. MARCO TEÓRICO

2.1 Redes De Sensores Inalámbricas

Las Redes de Sensores Inalámbricas (RSI) son un tipo especial de redes dentro de las redes inalámbricas de área personal de baja velocidad, principalmente caracterizadas por su baja capacidad de energía eléctrica, por lo general la situación geográfica donde se encuentra cada uno de los sensores que son parte de la red es de difícil acceso. La gran acogida de este tipo de redes se da con el avance de implementación de dispositivos electrónicos manejados con microcontroladores de tamaño muy pequeño, con altas prestaciones y muy bajo consumo de energía, propicio para poder utilizar estos dispositivos en múltiples aplicaciones. (Erazo P, 2015).

Este tipo de redes son muy flexibles ya que permiten monitorear sistemas complejos, donde el humano no tiene la posibilidad de coexistir con ambientes hostiles. Además de ser autoconfigurables, es decir que si existe un cambio en la topología de red ya sea por daño de algún nodo o por el agotamiento de la batería de uno de ellos, dicha red no deja de funcionar y permite seguir captando datos del entorno donde se encuentran instalados los sensores.

Las funciones más importantes de los sensores son:

- La comunicación multi-salto en una red que permite el uso de un protocolo de ruteo entre nodos sensores.
- Muchos nodos sensores que por lo general tienen las mismas funciones y cooperan entre ellos.
- Normalmente permiten cambios en la topología de la red debido a las fallas por pérdida, retiro o adición de nodos (Chacin, 2008).

2.1.1 Nodos Sensores

Los dispositivos llamados nodos sensores tienen internamente un microcontrolador, un sistema de radio, un sensor y un sistema de alimentación de energía como se observa en la Figura 1-2. Dichos nodos son encargados de sensar el medio ambiente donde se encuentren ubicados, procesar la información captada y transmitirla hacia otro nodo o hacia el *Gateway*.

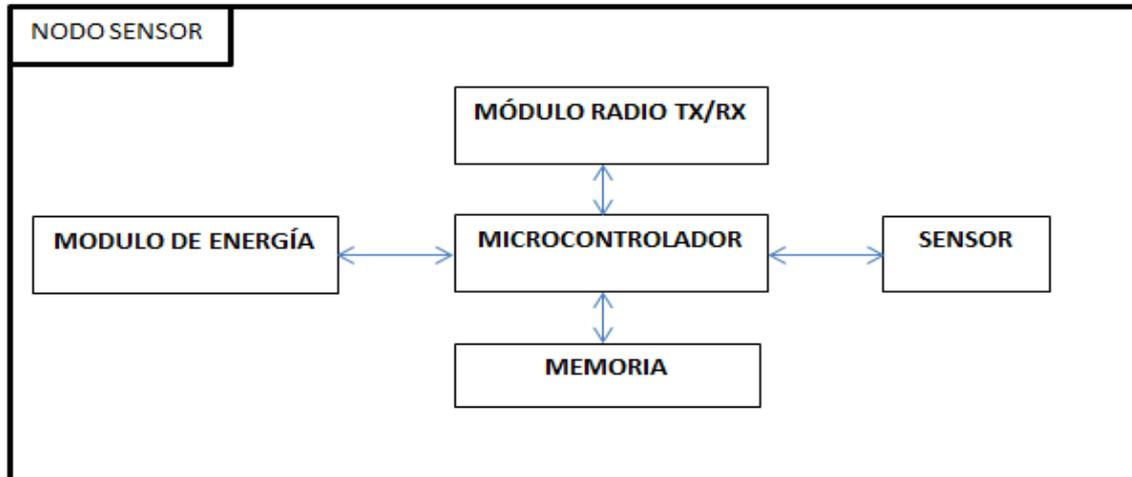


Figura 1-2 Esquema interno de un Nodo Sensor Inalámbrico

Fuente: GALLEGOS, Abel, 2018

2.1.1.1 Microcontrolador

Los microcontroladores ampliamente son utilizados en múltiples aplicaciones electrónicas ya que internamente tienen los componentes de un computador, con baja capacidad de memoria, de procesamiento y con posibilidad de comunicación a través de sus puertos de entrada y salida, Conversores Analógicos Digitales, y demás componentes.

2.1.1.2 Sistema de Radio

El sistema de radio es el que permite la transmisión y recepción de datos de cada nodo sensor para la comunicación de la red. Los principales estándares de comunicación utilizados en las redes de sensores inalámbricos son 802.11n, 802.15.1 y 802.15.4. El presente proyecto hace énfasis principal en las prestaciones que tiene cada uno de éstos estándares para obtener una red flexible y que permita la óptima comunicación de la red.

2.1.1.3 Sensor

Es el componente electrónico y/o físico de sensado, el mismo permite sensar la medida del ambiente, la cual va a depender de la aplicación en la que va a ser utilizada la RSI. La característica principal de estos sensores es el bajo consumo de energía eléctrica.

2.1.1.4 Sistema de Alimentación de Energía

Estos dispositivos electrónicos requieren de un módulo de alimentación de energía, la capacidad de consumo va a depender de las prestaciones del microcontrolador y del sensor, así como de la frecuencia de funcionamiento del radio y el protocolo de ruteamiento de la información a ser transmitida en la red. Por lo general los elementos de alimentación van a ser baterías que van a depender del origen del electrolito, las cuales pueden ser de Zinc, Litio, Níquel y otros. Se están investigando otros tipos de alimentación de energía que utilicen fuentes renovables, como es el caso de la energía solar.

2.1.1.5 Gateway

Es el equipo de red que permite a los nodos sensores enviar la información hacia redes de distinto tipo tales como el Internet, una red LAN⁴ o intranets. También se lo conoce como puerta de enlace y es el que va a interconectar los nodos sensores con la estación base. Este tipo de conexión por lo general se lo realiza a través de la pila de protocolos TCP⁵/IP.

2.1.1.6 Estación Base

Es el equipo de datos u ordenador que va a permitir la recolección de información desde toda la red, por lo general hacia una base de datos que tenga una relación con la aplicación que quiera dársele a la RSI. La comunicación entre la estación base y los dispositivos Gateway se la puede realizar mediante cable ya sea con tecnología Ethernet o Fibra óptica o mediante una red inalámbrica sea esta una red móvil o un enlace de tipo microondas, todo dependerá de la tecnología que se tenga a disposición y de los recursos económicos con los cuales se disponga (Fernández M, 2009).

⁴ Local Area Network

⁵ Transmission Control Protocol

2.2 Topologías típicas utilizadas en las RSI

Dentro de la red de sensores cada componente se puede definir según su funcionalidad en la misma, estos componentes pueden ser *FFD*⁶ y *RFD*⁷.

FFD dispone de memoria y recursos de red necesarios para poder ejecutar todas las funciones dentro del estándar de comunicación escogido para la red, mientras que RFD ejecuta el protocolo de acceso al medio ya que únicamente tiene su interfaz física hacia el modem inalámbrico. Los dispositivos podrán actuar en la topología de la red de la siguiente forma:

Coordinador de Red

Es un dispositivo de tipo FFD el cual tiene la responsabilidad de seleccionar los parámetros clave de la red, a la vez que podrá almacenar información relevante tal como claves de seguridad de la red.

Nodo Ruteador

Es un dispositivo FFD que permite la comunicación entre otros nodos ruteadores y los nodos terminales.

Nodo Terminal

Es un dispositivo del tipo RFD, que permite únicamente la comunicación con funciones reducidas de red, justas para la comunicación entre nodos sensores.

En las Redes de Sensores Inalámbricas existen tres tipos de redes bien definidas según su estructura:

- Redes planas: En los que todos los nodos tienen la misma función, es decir pueden ser nodos ruteadores y terminales a la vez.
- Redes jerárquicas: Donde existen nodos de distinto nivel, unos llamados de cabecera o ruteadores y otros llamados terminales.

⁶ Full Function Device

⁷ Reduced Function Device

- Redes basadas en localización: Donde cada nodo posee la información de su posición como ejemplo en un sistema de coordenadas, y emplea la información de sus nodos pares para transmitir la información por la red. (Capella H, 2010)

La topología de la red va a ser determinada por la combinación de los nodos antes expuestos, para lo cual se pueden generar las siguientes topologías:

Topología en Estrella

La característica principal es que el nodo terminal directamente interactúa con el nodo coordinador de la red, dicha topología usualmente tiene un alcance entre 30 y 100 metros. Todos los nodos sensores tienen la misma función. Los nodos finales no intercambian información entre ellos ya que es una red de un solo salto. Esta topología es la que consume menos energía en total de la red, pero el problema es que si existe un fallo en el nodo coordinador la red se cae por completo.

Topología en Malla

La característica fundamental en esta topología es que los nodos se pueden comunicar entre ellos, es decir son un conjunto de nodos ruteadores comunicados principalmente a través de multi-saltos, los cuales permiten la interconexión de la red con el nodo coordinador. El consumo de energía es más elevado debido al multi-salto de la información y porque cada nodo ruteador tiene la posibilidad de comunicarse con sus nodos vecinos y con el nodo coordinador. Este tipo de topología es tolerante a fallos debido a que si un nodo falla se puede auto-configurar la red para que la información viaje por otro camino.

Topología Híbrida Malla – Estrella

Esta topología es una combinación de las ventajas de las dos topologías anteriores, ya que permite un mejor aprovechamiento de la energía en la red, así como la ventaja de combinar el multi-salto de información entre nodos ruteadores con la posibilidad de ubicar nodos terminales para expandir la red hacia lugares más lejanos, donde se pueda ubicar nodos sensores de menor precio. Dentro de este proyecto se ha escogido la topología malla porque se requiere que todos los nodos permitan multi-salto de información.

2.3 Estándares De Comunicación

A nivel de la capa física, en las redes RSI existen varios tipos de estándares de comunicación, los mismos permiten la creación de una variedad de sistemas que de acuerdo a sus características como distancia entre nodos, ancho de banda e incluso la disponibilidad en el mercado para adquirirlos, hace que se puedan realizar varias comparaciones entre estos. Se citaran a continuación tres.

2.3.1 IEEE 802.15.1 Bluetooth

Bluetooth es parte de la familia de estándares para redes locales de bajo alcance. La relación que existe entre este estándar con los otros estándares se encuentra ubicado en la Figura 2-2, basados únicamente en el Control de Acceso al Medio y la capa física de una WPAN ⁸

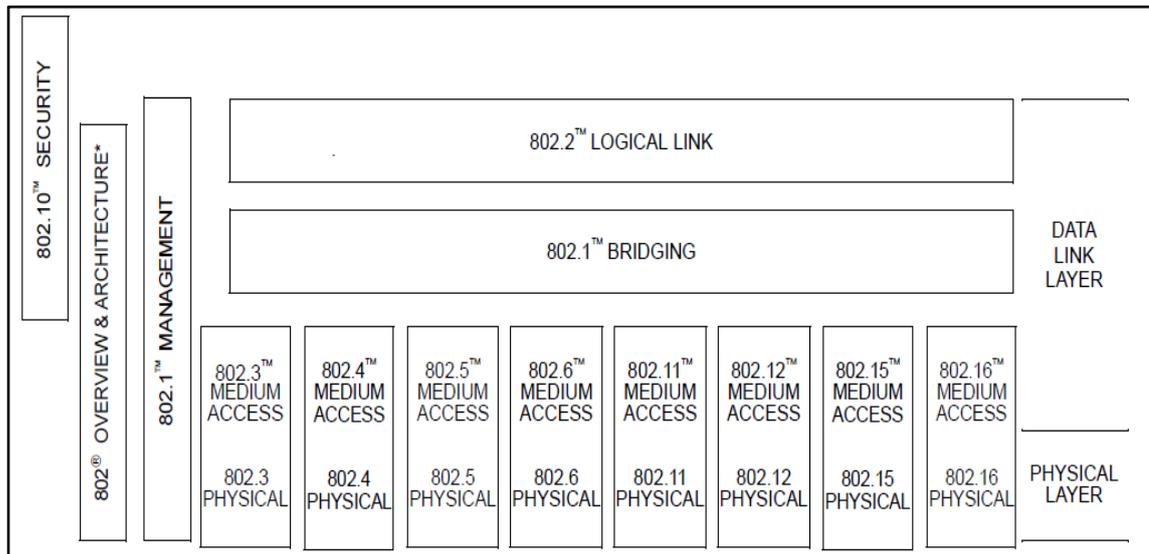


Figura 2-2 Correspondencia Estándares IEEE, Capa Física y MAC

Fuente: IEEE Std 802.15.1-2002

Se debe entender que este estándar tiene varios aspectos relevantes, de los cuales se explica a continuación:

La autenticación se la realiza a través de dos procedimientos: el primero basado en la autenticación del Protocolo de Manejo de Enlace LMP⁹ utilizado cuando existe llave entre dispositivos; el segundo basado en el emparejamiento a través de este LMP sin embargo este proceso se lo realiza cuando no existe llave entre dispositivos.

⁸ Wireless Personal Area Network

⁹ Link Management Protocol

Al hablar del estándar Bluetooth hay que tener en cuenta que directamente existe correlación con una red inalámbrica de área personal WPAN¹⁰, debido básicamente a tres factores fundamentales que son:

- Niveles de potencia y área de cobertura;
- Control del medio;
- Tiempo de vida de la red (Society, 2002).

El estándar de comunicación Bluetooth en su versión 5.0 es un sistema de radio comunicación de corto alcance que reemplaza a cables y que entre sus principales funciones clave se tiene la robustez, el bajo consumo de potencia y el bajo costo de los dispositivos. Muchas de las funciones del núcleo son opcionales y esto permite que cada producto sea diferenciado por sus aplicaciones y construcción.

Dentro de la versión 5.0 existen las tecnologías: BR¹¹, LE¹² y AMP¹³. Las dos primeras tecnologías permiten el descubrimiento de dispositivos, el establecimiento y mecanismos de conexión. La primera incluye un mejoramiento de rata de datos llamado EDR¹⁴ que se combina con el control de Acceso al Medio (MAC) y la capa Física (PHY). Ésta permite conexiones asíncronas o síncronas de 721,2 kbps para BR y 2,1 Mbps para EDR.

El Núcleo Bluetooth tiene controladores primarios y secundarios. Un controlador primario se puede configurar de las siguientes formas:

- Un controlador BR/EDR que incluye el dispositivo de Radio, Banda Base, Control de Enlace y una Interfaz Controlador de Host (HCI) opcional.
- Un controlador LE que incluye la capa física LE PHY, Capa de Enlace y una Interfaz Controlador de Host (HCI) opcional.
- Una combinación entre BR/EDR y LE, dentro de un controlador simple.

¹⁰ Wireless Personal Area Network

¹¹ Basic Rate

¹² Low Energy

¹³ Alternate MAC/PHYs

¹⁴ Enhanced Data Rate

2.3.1.1 *Modo de Operación BR/EDR*

La operación en BR/EDR ¹⁵ de la capa física se la realiza sobre la banda ISM de 2,4GHz. El sistema emplea el espectro ensanchado por salto de frecuencias FHSS por sus siglas en inglés para combatir la interferencia a través de distintas portadoras. El modo BR usa modulación por frecuencia binaria para minimizar la complejidad del transmisor. La tasa de símbolo es 1 Mega-símbolo por segundo, soportando una tasa de datos de 1 Mbps, mientras que en el modo EDR, la tasa es 2 o 3 Mbps.

En una operación típica de comunicación vía Bluetooth, el canal físico de radio es compartido por el grupo de dispositivos mediante un patrón de salto de frecuencia, el cual lo sincroniza un reloj común. El dispositivo master es el que provee la sincronización, mientras que los demás dispositivos son llamados esclavos. Un grupo de dispositivos sincronizados por un mismo máster se denomina una PICONET.

Los dispositivos de una PICONET tienen un patrón común de salto de frecuencia pseudorandómica, en un grupo de 79 frecuencias separadas 1 MHz en la banda ISM.

El canal físico es subdividido en unidades de tiempo llamados SLOT. Los datos son transmitidos entre dispositivos, mediante paquetes guardados en cada slot. La tecnología Bluetooth provee el efecto full dúplex de transmisión a través del uso del esquema TDD¹⁶.

El canal físico lo forman un dispositivo máster y un esclavo. El enlace es bidireccional, lo que provee el transporte de paquetes entre el dispositivo máster y el esclavo. En el caso de una conexión tipo esclavo broadcast, el enlace provee un transporte unidireccional entre el máster y probablemente un sin número de esclavos. Una PICONET no puede estar formada solo entre dispositivos esclavos.

El tráfico de datos es enviado a través de enlaces lógicos multiplexados sobre el enlace físico, ocupando los SLOT asignados por el administrador de recursos.

El control de banda base, capa física y datos de usuario es llevado sobre enlaces lógicos a través del protocolo administrador del enlace LMP¹⁷. La función del administrador de enlaces utiliza LMP para controlar la operación de dispositivos en una PICONET.

¹⁵ Basic Rate / Enhanced Data Rate

¹⁶ Time-Division Duplex

¹⁷ Manager Protocol

Sobre la banda base la capa L2CAP provee un canal basado en abstracción para aplicaciones y servicios. Suministra segmentación y ensamblaje de datos de aplicaciones y Multiplexa y demultiplexa canales sobre el canal de enlace lógico compartido. (Bluetooth Sig, 2016)

2.3.1.2 *Modo de Operación Bluetooth Low Energy*

Opera en la banda ISM 2,4GHz. Al igual que BR/EDR emplea el espectro ensanchado por salto de frecuencias para combatir la interferencia y desvanecimiento a través de múltiples portadoras. La tasa de símbolo es de 1 Mega-símbolo por segundo, cuando 1 símbolo representa 1 bit, por lo tanto la tasa de datos es de 1 Mbps, conocido como LE 1M PHY. La tasa de 1 Mega-símbolo por segundo puede opcionalmente soportar codificación de corrección de errores, dentro de dos esquemas; el primero cuando $S=2$, lo que significa que dos símbolos representan un bit, por lo que la tasa de datos será dividida en 2, es decir 500 kbps, y el segundo esquema donde $S=8$, por lo que la tasa de datos sería 125 kbps. También existe opcionalmente el modo LE 2M, que representa 2 Mega-símbolos.

Este modo tiene dos tipos de acceso: TDMA¹⁸ y FDMA¹⁹. En FDMA se usan cuarenta (40) canales físicos separados por 2 MHz. Treinta y siete (37) canales son secundarios y de datos, mientras que los tres (3) restantes son de aviso primario. En TDMA un dispositivo transmite un paquete en un tiempo predeterminado y el correspondiente responde con un paquete después de un predeterminado intervalo.

Los canales físicos son divididos en unidades de tiempo llamados eventos. Los datos son transmitidos entre dispositivos LE a través de paquetes que son llevados en dichos eventos. Existen cuatro tipos de eventos: Aviso, Aviso extendido, Aviso Periódico y Conexión.

¹⁸ Acceso Múltiple por División de Tiempo

¹⁹ Acceso Múltiple por División de Código

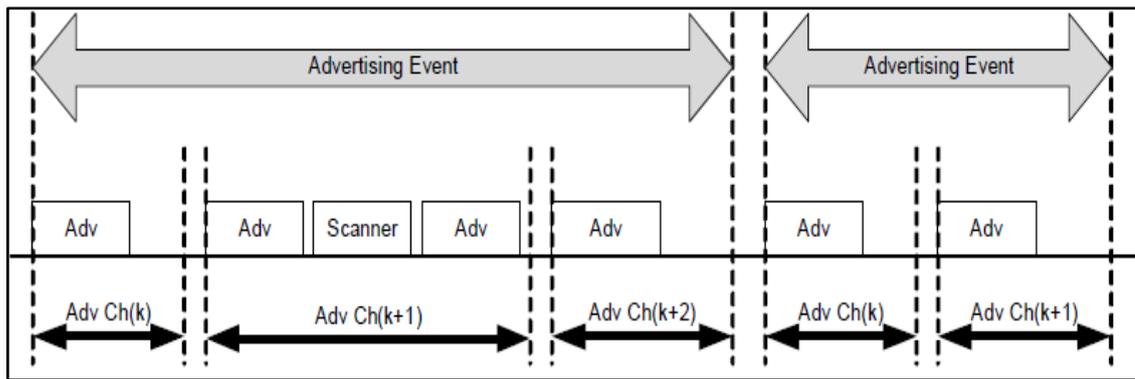


Figura 3-2 Eventos de Aviso en Bluetooth Low Energy

Fuente: (Bluetooth Sig, 2016)

Los dispositivos que transmiten paquetes de aviso sobre canales físicos de aviso son llamados anunciantes. Los dispositivos que reciben paquetes de aviso sobre canales de aviso sin intención de conectarse al dispositivo de anuncio son llamados escáneres. Al iniciar cada evento de aviso, el anunciante envía un paquete correspondiente con el tipo de evento. Dependiendo del tipo de paquete, el escáner puede dar una respuesta al anunciante sobre el mismo canal físico de aviso. El canal físico de aviso cambia el siguiente paquete de aviso enviado en el mismo evento. El anunciante puede terminar el evento en cualquier instante durante el mismo. El primer canal físico de aviso es utilizado para el inicio del siguiente evento.

Los dispositivos en una Piconet utilizan un patrón de salto de frecuencia que es determinado por un algoritmo contenido en un campo que es proporcionado por el iniciador del servicio a través del mensaje de respuesta de conexión. Este salto de frecuencias se da en la Banda ISM de forma pseudoaleatoria.

Dentro del canal físico se forma un enlace entre dispositivo maestro y esclavo. El enlace físico activo es el que proporciona transporte bidireccional de paquetes. En cada enlace físico siempre debe haber un dispositivo master y un esclavo, no se puede dar entre dos esclavos. El enlace físico es usado por uno o más enlaces lógicos, el tráfico de cada enlace lógico es enviado y multiplexado a través del enlace físico por el administrador de recursos.

El protocolo de Enlace Lógico, la capa física y los datos de usuario son transmitidos a través del Enlace Lógico (*Link Layer*). Todos los dispositivos activos en una Piconet por defecto tienen una conexión lógica asíncrona (LE ACL) por sus siglas en inglés que transporta la señalización del protocolo de enlace lógico.

En la parte superior de la capa de enlace se encuentra la capa L2CAP, la cual provee los servicios de fragmentación y desfragmentación de datos de aplicación, multiplexación y demultiplexación de canales, que permite a las capas superiores de aplicación y servicios interactuar con las capas inferiores (Física y Enlace), de igual manera que en BR/EDR.

En la parte superior de L2CAP, residen dos protocolos adicionales. El primero es el protocolo de manejo de, el cual utiliza un canal fijo L2CAP para brindar seguridad entre dispositivos. El segundo es el protocolo de atributo, que permite enviar pequeñas cantidades de datos sobre un canal fijo L2CAP, y es utilizado por los dispositivos para determinar capacidades y servicios de los otros dispositivos de la PICONET. El protocolo de atributo también es utilizado en BR/EDR.

2.3.1.3 *Modo de Operación AMP*

El modo Alternativo (*Alternate MAC/PHY*), son controladores secundarios en Bluetooth. La radio BR/DR se utiliza para el descubrimiento, asociación, establecimiento y mantenimiento de la conexión de la radio. Cuando se enlazan las capas L2CAP de los dispositivos, los administradores AMP pueden descubrir los AMP que estén disponibles en el otro dispositivo. Cuando existe un AMP común entre dos dispositivos, el núcleo del sistema proporciona los mecanismos para que el tráfico de datos de los controladores del BR/EDR pase a un controlador AMP.

La capa de adaptación de protocolo se encuentra en el controlador AMP sobre las capas PHY y MAC. Este protocolo es responsable de mapear el comportamiento de Bluetooth según lo especificado en HCI²⁰ hacia los protocolos PHY y MAC correspondientes

Dentro de los controladores AMP pueden crearse o moverse los canales L2CAP. También pueden volver a la radio BR/EDR cuando las capacidades AMP no son necesarias o cuando el tiempo de enlace físico AMP cumple su límite. Los AMP pueden conectarse o desconectarse según sea necesario el ahorro de energía.

²⁰ Host Controller Interface

El funcionamiento de Bluetooth se entiende de una mejor forma a través de un modelo de capas en el que se encuentra: Canal, Enlace Físico, Transporte Lógico, Enlace Lógico y el canal L2CAP. En la figura 2-6 se puede apreciar los distintos niveles de conexión e interacción del modelo antes mencionado. Se puede observar cada uno de los modos de Operación (BR/EDR, LE y AMP).

Para el funcionamiento de IoT, el Internet de las cosas, se ha desarrollado el estándar BLE, el mismo que permite que los sensores tengan la capacidad de ahorro de energía y que se puedan comunicar de forma oportuna para interactuar en la red. Cada dispositivo dentro de una red en especial domótica y de seguridad, tiene la capacidad de encontrarse en modo ahorro de energía hasta que deba transmitir ya sea a intervalos de tiempo o cuando exista una alarma específica, que haga al dispositivo interactuar y avisar del evento por el cual se ha despertado.

BLE y ZIGBEE al ser de la misma familia de protocolos (IEEE 802.15), se encuentran dentro de los estándares más comunes en la implementación de las Redes de Sensores Inalámbricos (WSN).

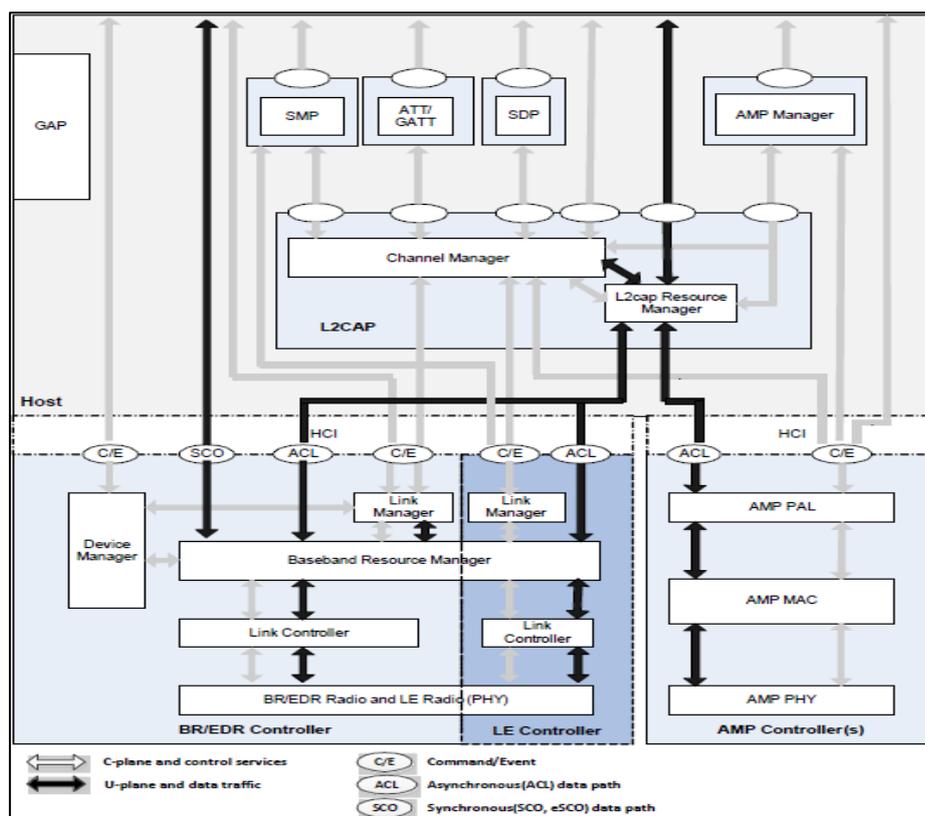


Figura 4-2 Eventos de Aviso en Bluetooth Low Energy

Fuente: (Bluetooth Sig, 2016)

Dentro de los logros más importantes en la versión 5.0 de Bluetooth se encuentra el área de cobertura, misma que supera en pruebas los 350 metros. (Woolley, 2018)

2.3.2 IEEE 802.11n WI-FI

El estándar IEEE 802.11n tiene como característica principal sus modos de comunicación con alto throughput, es decir un alto rendimiento de la capacidad de tasa de bits en la red, además es uno de los estándares de comunicación más usado en las redes inalámbricas, debido a su cobertura y a su tipo de multiplexación (OFDM), que permite al usuario de red obtener mayor velocidad en la comunicación de datos en comparación a estándares como el Bluetooth o ZIGBEE.

2.3.2.1 Nivel Físico

Dentro de las principales características a nivel de la capa física se tienen las siguientes:

- Las tasas de datos son superiores debido al uso de la Multiplexación por División Espacial, asociado a la utilización del arreglo de antenas MIMO y canales de 20 o 40 MHz.
- Para una transmisión de radio utiliza la multiplexación por división de frecuencias ortogonales (OFDM), con un tiempo de símbolo $T_s=4\text{ms}$, en un canal de 20 MHz con 52 subportadoras, a una tasa máxima de 65 Mbps.
- A través de MIMO puede alcanzar una velocidad de transmisión de hasta 600 Mbps, ya que utiliza la transmisión y recepción por varios arreglos de antenas aprovechando la propagación multi-camino, lo que conlleva una mayor tasa de transmisión.
- A través de la característica llamada “Transmit Beamforming”, transmite varias señales de radio con un desfase entre múltiples antenas que se unifican en el receptor. Con esto aumenta la señal recibida incrementando la Relación Señal Ruido (SNR). Ésta técnica solo puede ser utilizada cuando hay un único receptor, ya que si se tuviera varios receptores sería muy complicado calcular los desfases de manera correcta.

- Multiplexación por División Espacial: Multiplexa una señal de un ancho de banda determinada en n señales de menor ancho de banda iguales, a través de MIMO.
- El intervalo de guarda es menor (800ns), lo que reduce la Interferencia Intersimbólica.
- Código de verificación de baja densidad (LDPC): códigos lineales de bloque donde se utilizan muchos ceros y pocos unos, lo que en la matriz de paridad se traduce en baja densidad. (Chávez, 2009)

2.3.2.2 *Nivel de Acceso al Medio*

Cuando se habla de Acceso al medio, se tiene que entender que representa la capa de enlace de datos, dicha capa se encuentra representada formalmente por varios tipos de Acceso dependiendo de la tecnología que se vaya a seguir. Específicamente en la Capa MAC se proporciona el control de transmisión de datos. Al igual que en Ethernet, en este tipo de redes (Inalámbricas) existen colisiones, por lo que 802.11n utiliza CSMA-CA²¹. Para detectar si el canal está libre o no, para esto se utiliza el CCA²², que permite verificar el estado del canal, y una vez que esté libre, envía mensajes de aviso para poder transmitir. En el caso de existir colisiones se esperan tiempos aleatorios para poder ocupar nuevamente el canal.

2.3.2.3 *Protocolos de Control de Acceso al Medio*

Los protocolos de control de acceso al medio se dividen en dos grupos: libres de contención y protocolos basados en contención. Los de libre contención a su vez se dividen por asignamiento de espacio fijo y dinámico. Dentro del asignamiento fijo de espacio se tienen Acceso Múltiple por división de Frecuencia, Acceso Múltiple por División de Tiempo y Acceso Múltiple por división de Código, mismos que dividen el canal de comunicación ya sea en frecuencia, por tiempo o por código respectivamente y siempre tienen un slot o espacio dedicado a cada usuario. Los de asignación por espacio dinámico son los basados en encuestas, Paso de token y basados en reserva. Los de asignación basados en encuestas tienen un dispositivo que debe controlar el envío y recepción de las mismas para poder asignarle el canal de comunicación. El paso de token en cambio tiene una trama llamada token, el mismo que pasa de dispositivo en dispositivo y el que tiene dicha trama tiene la posibilidad de transmitir. El basado en reserva

²¹ Acceso Múltiple por Detección de Portadora con Prevención de Colisiones

²² Clear Channel Assessment

utiliza intervalos de tiempo estáticos para que los nodos reserven el futuro acceso al medio de comunicación en función a la demanda. (Dargie, 2010)

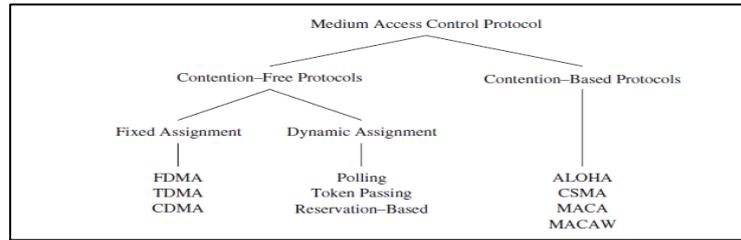


Figura 5-2 Protocolos de Control de Acceso al Medio

Fuente: Dargie, 2010

Como se puede observar en la Figura 5-2, en el estándar 802.11n el protocolo de Acceso al Medio utilizado es basado en Contención, específicamente el CSMA-CA.

2.3.3 IEEE 802.15.4 ZIG-BEE

El estándar IEEE 802.15.4 asociado a ZIG-BEE se encuentra dentro de las tecnologías de Acceso al medio para una red de sensores inalámbricos de Low Rate WPAN, el mismo que opera en una tasa mínima de 20 kbps en frecuencias 868/915MHz (Europa y Norte América) y una máxima de 250 kbps en frecuencias 2,4GHz libre para todo el mundo como indica la Tabla 2-1. El acceso al medio está dado por CSMA-CA²³. Es un protocolo de acceso muy flexible, siendo uno de los más utilizados en las RSJ. El alcance entre dispositivos ZIG-BEE se encuentra entre 10 y 150m.

Tabla 1-2: Comparativa Frecuencia ZIGBEE

Física	Banda de Frecuencia	Números de Canales	Parámetros de propagación		Parámetros de los datos		
			Velocidad de procesamiento	Modulación	Bits por segundo	Cantidad de Información	Modulación
868/915 MHz	868-870 MHz	0	300 kchip/s	BPSK	20 kb/s	20 kband	BPSK
	902-928 MHz	1 a 10	600 kchip/s	BPSK	40 kb/s	40 kband	BPSK
2,4 GHz	2,4-2,4835 GHz	11 a 26	2,0 Mchip/s	O-QPSK	250 kb/s	62,5 kband	16-ary Orthogonal

Fuente: (Vera R, Barbosa J, & Pabón G, 2017)

²³ Acceso al Medio por detección de portadora con prevención de colisión

Hay que recalcar que el estándar IEEE 802.15.4 es la base para el control de acceso al medio (MAC) y para el manejo de la capa física (PHY) en el estándar ZIGBEE. En la figura 6-2 se puede observar el modelo de capas seguido por el estándar ZIGBEE 3.0, para lo cual la Alianza ZIGBEE ha sido la responsable de crear los demás protocolos de red, transporte y aplicación para los dispositivos basados en esta tecnología. Dicha Alianza es conformado por empresas que han venido desarrollando no solo la base para los protocolos ZIGBEE sino también ha venido investigando y desarrollando los dispositivos específicamente para usos industriales, construcción, de transporte, de salud, de domótica para hogares, y control de ciudades. Se espera que para 2018 existan en el mercado 1 billón de unidades de sensores inalámbricos fijos, lo que constituirá a ZIGBEE como el líder mundial en las RSI.

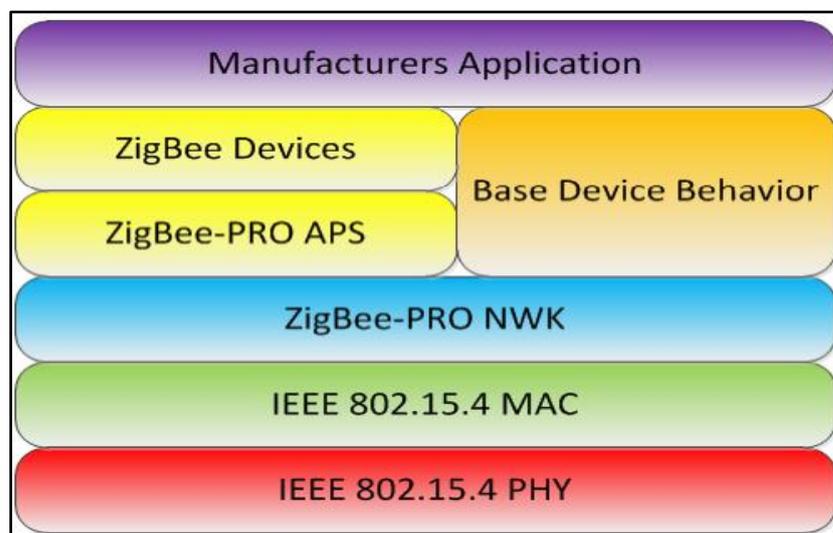


Figura 6-2 Modelo de Capas ZIGBEE

Fuente: <http://www.zigbee.org/>

Zigbee está basado en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (WPAN²⁴) fue diseñado para ser utilizado en comunicaciones inalámbricas que requieren corto alcance y limitada potencia, es una tecnología de baja tasa de transferencia, bajo consumo de energía, la principal función es la de crear una topología de red jerárquica para que n dispositivos se comuniquen incluso se adiciona características como la autenticación, encriptación.

²⁴ Wireless Personal Área Network

2.3.3.1 Arquitectura del dispositivo de la LR-WPAN

El estándar IEEE 802.15.4 establece las características de la capa física y de la capa de control de acceso al medio para redes inalámbricas de área personales, las ventajas de LR- WPAN es la facilidad de instalación, transferencia de datos confiables, operación de corto alcance costo extremadamente bajo, y una duración de batería permitida, manteniendo al mismo tiempo una pila de protocolo simple y flexible.

El dispositivo se compone de la capa física que contiene el transceptor de radio frecuencia junto con su mecanismo de control de bajo nivel y una subcapa MAC que provee acceso al canal físico para todos los tipos de transferencia.

El 802.2 control de enlace lógico IEEE(LLC) puede acceder a la subcapa MAC a través de la subcapa de convergencia específica de servicio (SSCS) como la mayoría de otros protocolos 802, no se especifica no describí las capas superiores de la pila.

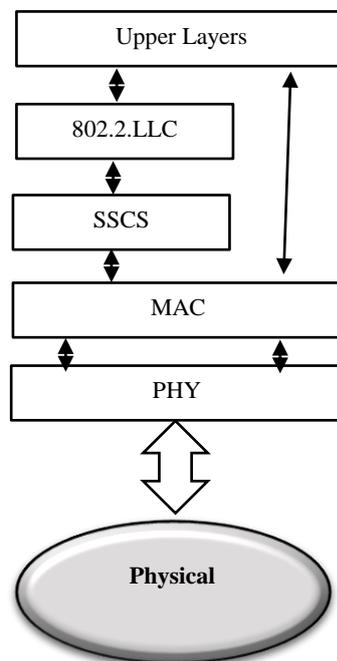


Figura 7-2 Modelo de Capas ZIGBEE

Fuente: (Vera R, Barbosa J, & Pabón G, 2017)

2.3.3.2 *Capa Física de IEEE 802.15.4*

La capa física facilita dos servicios datos y gestión el servicio de datos permite la transmisión y recepción de unidades de datos de protocolos a través del canal de radio físico. La característica de la capa física son activación y desactivación de transceptor de radio, la detención de energía e indicación de la calidad del enlace, selección del canal, evaluación clara y la transmisión y recepción de paquetes a través del medio físico. (Vera R, Barbosa J, & Pabón G, 2017)

2.3.3.3 *Aplicaciones de Tecnología Zigbee*

Se menciona que las aplicaciones de tecnología Zigbee son diversas van desde las aplicaciones de automatización del hogar, control industrial electromedicina en control de tráfico, en sistemas de alerta, localización y seguimientos de personas, control de accesos, detención de fugas, sistemas de alerta, de perímetros de seguridad, optimización de recursos, control de gasto energético etc.

2.4 **Tipos de Sensores**

Los sensores utilizados para medir el nivel de agua en un río se llaman pluviómetros. Existe varios modelos de sensores, los mismos que difieren de su fabricante siendo los más relevantes los que se verifican a continuación.

2.4.1 *Sensor de Nivel de Río (Moradel Fonseca)*

Es un sensor de nivel de río de tipo inalámbrico, con alarma sonora. Tiene una altura de 2m extensible hasta 8m. Su tiempo de respuesta es de 200 ms. El espaciado de medición entre nivel es de 30 cm. Utiliza batería de 12V con carga automática solar a través de un panel de 12 V a 5W en el receptor y un panel solar a 6V con su respectiva batería en el transmisor. El transmisor y el receptor transmiten en RF de 433 MHz. El equipo medidor de nivel de agua está construido en tubo PVC con flotadores Reed-Switch de Propileno. Tiene una pantalla LCD en el receptor. El cerebro de este circuito es un microcontrolador de 8 bits. (Moradel F, 2014)

En la figura 8-2 se aprecia la estructura física y de funcionamiento del sensor de nivel de río antes mencionado. Como se puede apreciar, la transmisión y recepción de los datos del sensor son enviadas vía radio frecuencia y están organizados a través de un microcontrolador que hace las veces de mini computador, permitiendo el control y presentación de los datos.

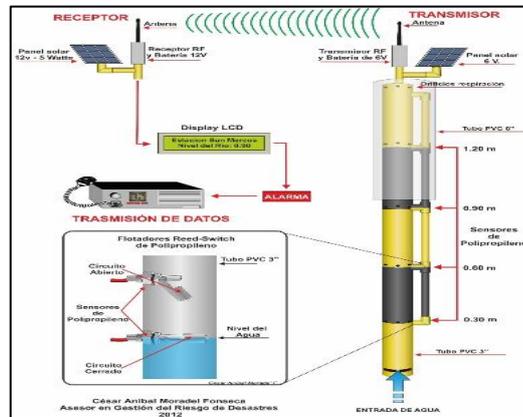


Figura 8-2 Sensor de Nivel de río

Fuente: (Moradel F, 2014)

2.4.2 Sensor de Nivel de Líquidos tipo radar

Este tipo de sensor utiliza un radar en banda K, con una precisión de 2 mm. Tiene salida analógica de 4 a 20 mA. Soporte de temperatura entre -40 y $+80^{\circ}\text{C}$. Utilizan una antena tipo trompeta plástica. Su medición es hasta 35m, tensión de trabajo entre 9,6 y 35 VDC. Su configuración es posible vía Bluetooth a través de smartphome IOS y Android, a través de la aplicación Vega Tools. Permite interconexión con PC a través de la interface VEGA CONNECT vía USB. También puede conectarse a un módulo llamado PLICSCOM (VEGADIS 82) para GSM. El tiempo de medición del radar es de 2 ms. (Vega, 2018)



Figura 9-2 Sensor tipo radar VEGAPULS 61

Fuente: (Vega, 2018)

En la Figura 9-2 se puede verificar el sensor tipo Radar VEGAPULS 61, el mismo que fue descrito en esta sección.

2.4.3 Sensor de Nivel Seba PULS 15

Este tipo de sensor es utilizado actualmente por el Sistema de Alerta Temprana de Alluriquín, su salida es analógica de 4 a 20 mA. Su rango de medición es de 15m con precisión de 2 mm. La conexión puede darse vía remota a través de GSM/GPRS. Se alimenta con un panel solar a 12 VDC, con una batería de 18 Ah/12V. Contiene un *datalogger* que permite guardar los datos y distribuirlos a través de la red GSM. La temperatura de operación se da entre -40 y 80 °C. El consumo de energía es de 185 mA en modo *idle* y 255 mA en modo Transmisión (Domingo, 2016).

2.4.4 Sensor e+ WATER L

Es un sensor de nivel de agua tipo varilla, con un rango entre 0 y 2m y precisión de 3mm. Intervalo de resolución es de 0,1cm. Tiene un componente e-sense con conexión modem de campo y modem para PC tipo GSM, con un máximo de 8 sensores por modem. Su temperatura de operación entre -20 y 80°C. Existen intervalos de monitoreo que pueden ir de varios segundos a horas. Utiliza baterías de litio de 3,6VDC/ 2,3 Ah. Aproximadamente la batería tiene una duración de 6 años con muestreo de la señal cada 10 min. En la Figura 10-2 se puede observar como el sensor tipo varilla se encuentra instalado en la fuente de agua a medir.



Figura 10-2 Sensor tipo varilla e+ water

Fuente: (Eijkelkamp, 2018)

A continuación, se detallan las principales características de cada sensor:

Tabla 2-2: Comparación Sensores

Sensor / Característica	Moradel Fonseca	Sensor de Nivel de Líquidos tipo radar	Sensor de Nivel Seba PULS 15	Sensor e+ WATER L
Distancia (m)	8	35	15	2
Precisión (mm)	300	2	2	1
Voltaje Nominal (VDC)	6	9,6 - 35	12	3,6
Costo	Bajo	Elevado	Elevado	Elevado
Medio de Transmisión	Radio	GSM	GSM	GSM

Fuente: GALLEGOS, Abel, 2018

De acuerdo a la tabla 2-2, se puede determinar que por el costo y el tipo de transmisión, el mejor sensor propuesto es el Sensor de Nivel de Río (Moradel Fonseca), aunque su precisión no es la mejor, pero es muy aceptable.

CAPÍTULO III

3. ESCENARIO Y SIMULADOR

3.1 Escenario

La cuenca del Río Damas que puede afectar a la Parroquia Alluriquín tiene una extensión de aproximadamente 3 km desde la población hacia la cordillera, El escenario se presenta a continuación, teniendo en cuenta que el Río Damas tiene Irregularidad en su Terreno, sin embargo se va a considerar distancias lineales múltiples de 400 metros con el objetivo de facilitar la simulación.

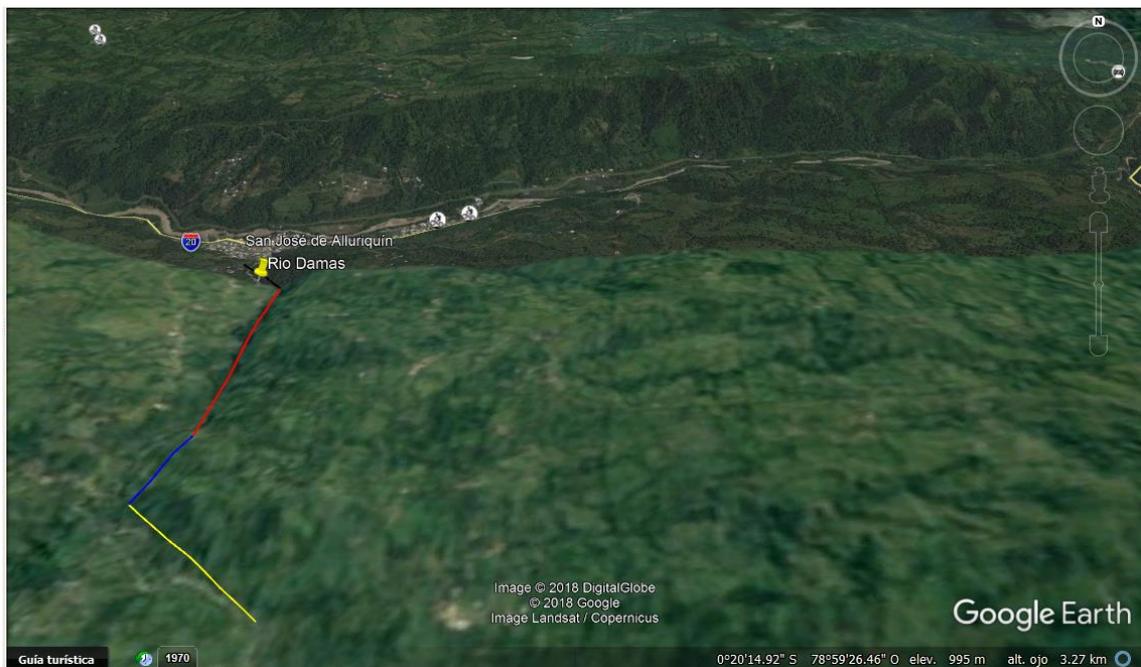


Figura 11-3 Esquema Ruta de Red sobre Río Damas

Fuente: Google Earth 7.3.1.4507

En la figura 11-3 se puede observar que existen 4 líneas o rutas, las cuales se han diferenciado con distintos colores que representan la posibilidad de seguir un camino pseudo-lineal de la señal de la red de sensores. La primera de color negro que es la más cercana a la Población de Alluriquín, misma que tiene una longitud de 400 metros. La segunda de color rojo es la de mayor longitud y es de 1200 metros. La tercera es de 400 metros con distintivo de color azul. La

última de color amarillo es la más lejana a la cabecera Parroquial de Alluriquín y cubre una longitud de 800 metros. El total analizado de longitud del río va a ser de 2800 metros.

Se puede observar en las siguientes imágenes cómo es la vegetación alrededor del Río Damas. Para esto se debe tomar en consideración el canal de comunicación, el tipo de dispositivo, la ganancia de la antena, el tamaño del paquete a enviarse, el número de nodos que me permitan alcanzar los 2800 metros de longitud a estudiar, si existe o no la posibilidad de asignación de dirección de red dinámica, adicionalmente del protocolo de ruteo para la comunicación entre nodos y un método de propagación adecuado con sus respectivas características de pérdidas en el proceso de transmisión de los datos. Así como también considerar la posibilidad de que existan nodos en los cuales se requiera la instalación de mástil para obtener una altura adecuada, minimizando así las pérdidas del canal.



Figura 12-3 Río Damas

Fuente: GALLEGOS, Abel, 2018

Como se puede observar en la Figura 12-3 el caudal del Río es bastante alto en el invierno. La imagen 13-3 a continuación fue tomada por el autor de la tesis en época de verano, comprobando las coordenadas del lugar ya que en invierno es difícil el acceso al mismo. Se pueden verificar más fotos en el ANEXO B - FOTOS DEL RÍO DAMAS.



Figura 13-3 Río Damas

Fuente: GALLEGOS, Abel, 2018

3.2 Simuladores

Para lograr una simulación lo más real posible se deben considerar aspectos esenciales, los cuales se verificaron en el párrafo anterior, se enumeran los siguientes:

- El simulador debe permitir la simulación de los tres estándares de comunicación propuestos: 802.11n, 802.15.1 y 802.15.4.
- Debe permitir configurar los siguientes parámetros: número de nodos, distancia entre nodos, canal de comunicación, modo de propagación, ganancia de la antena, direccionamiento, protocolos de red, protocolos de transporte, tamaño del paquete en Bytes y otros.
- Debe permitir mostrar resultados de forma que se puedan interpretar de una manera sencilla y con bastante exactitud.

Ante esta situación se presenta un resumen de las principales características de tres simuladores: Network Simulator 2, Network Simulator 3 y OMNET++.

Tabla 3-3: Comparación Simuladores de Redes

Característica	Network Simulator 2	Network Simulator 3	OMNET++
Lenguaje de Programación	C++, Interface OTcl	C++, Phyton	C++
Animador de Interfaz	NAM (.nam animación)	NetAnim (.xml animación)	GUI
Analizador de Datos	.tr para trazas .xg para gráficos	.tr para trazas .plt gnuplot para graficos .pcap permite WireShark	Gráficas y estadísticas a través de su herramienta GUI
Emulador	Permite solo simulación	Permite simulación y emulación	Permite simulación y emulación
Nivel de Aprendizaje	Bajo	Alto	Alto
Tráfico que soporta	Medio	Alto	Alto
Soporte en Redes inalámbricas	Bueno	Alto	Bueno
Licencia	Opensource	Opensource	Opensource

Fuente: GALLEGOS, Abel, 2018

De acuerdo a la tabla 3-3, se puede verificar que existe una ventaja bastante grande del Simulador NS3, debido a que permite la creación de archivos .pcap, los mismos pueden interpretarse a través de la herramienta para Redes “WireShark”. Adicional que el nivel de aprendizaje, tráfico que soporta y nivel de soporte para redes inalámbricas es alto. Por lo que para el desarrollo de ésta investigación se escoge al Network Simulator 3.

A continuación se presentan las principales características y funcionalidades del NS3.

3.3 Network Simulator 3 (NS3) – Generalidades

El software NS3, es un simulador de eventos discretos para redes de comunicaciones, el mismo que está escrito en C++ y PHYTON. Esta versión fue creada en el año 2006 y no guarda compatibilidad con su antecesor el NS2, puesto que fue reescrito totalmente desde cero.

Está escrito sobre código abierto, y es una plataforma donde se encuentran todos los recursos para los investigadores y quienes deseen contribuir con mejorar estándares y pruebas en redes de telecomunicaciones.

Los principales recursos que se tiene para la investigación y ejecución del NS3 son los siguientes:

3.3.1 *La Web*

Existe variada información dentro de la web oficial de NS3. Dentro de ella se pueden encontrar manuales, tutoriales, ejemplos e incluso artículos científicos desarrollados en base a las principales arquitecturas de red.

3.3.2 *Mercurial*

Es un complejo sistema de manejo que organiza internamente toda la codificación del NS3. El proyecto NS3 usa a Mercurial como un administrador de recursos. Mercurial permite organizar, gestionar y administrar los cambios en el código y documentación que conlleva un proyecto en NS3.

3.3.3 *WAF*

Existen diversas maneras de compilar un programa en NS3. Uno de los más conocidos y utilizados es WAF. Este compilador se encuentra construido en PHYTON, sin embargo no necesariamente debe existir la codificación en dicho lenguaje para poder compilar cualquier código en NS3, sino que debe cumplir con las principales características de un código en C++ para poderlo compilar y ejecutar.

3.3.4 *Ambiente de desarrollo*

Los scripts en NS3 son escritos normalmente en C++ o Phyton, adicionalmente los modelos que se ejecutan en cada simulación en su mayor parte son en C++, dichos modelos son orientado a objetos. Para poder editar o escribir nuevos scripts en C++ para NS3, se pueden utilizar varios programas dentro de Linux. Uno de los más conocidos y utilizados para esta función es ECLIPSE, el mismo que es un compilador de proyectos en C++ y que permite tener todas las funciones necesarias para el desarrollador.

3.3.5 Condiciones para la ejecución del simulador NS3

Para este proyecto se utilizaron condiciones necesarias que permitieron un correcto ambiente de desarrollo de nuevas API para NS3 o a su vez se utilizó los scripts de ejemplo o de recurso existentes en el directorio del Simulador.

El simulador se instaló en el sistema operativo Linux Mint 18. El editor para los scripts en C++ utilizado fue el Eclipse Versión Oxygen Release (4.7.0) como se muestra en la figura 14-3.



Figura 14-3 Eclipse IDE

Fuente: (Foundation, 2018)

CAPÍTULO IV

4. SIMULACIÓN

4.1 Selección del Estándar

Para escoger el mejor estándar y poder tomar un modelo que se ajuste a las circunstancias acordes al Ambiente en el Río Damas, se procede a verificar cuál de los tres estándares (802.11n, 802.15.1 y 802.15.4) permite que pueda existir distancia mayor entre nodos, lo que por el complicado acceso al lugar sería de ayuda para la probable instalación de los nodos sensores. A continuación, se muestran los resultados de la simulación en los tres estándares, teniendo como base el número de nodos, la distancia que existe entre cada uno de ellos y el mayor *throughput*:

4.1.1 Simulación 802.11n Wi-Fi

En la simulación realizada en el Software NS3, según el ANEXO 802.11n WIFI - PROGRAMACIÓN NS3, se tienen los siguientes resultados considerando al estándar 802.11n en el primer modo HtMcs0 de 6,5 Mbps, canal de comunicación configurado con pérdidas de acuerdo al espacio libre, datos enviados a través del protocolo UDP²⁵, una carga útil de 1442 Bytes, distancia variable entre 10 y 230 metros. Dentro de cada simulación se ha normalizado el *Throughput* con valor máximo de 1, para poder comparar los distintos estándares. En la simulación el número de nodos fue de 10, 20, 50, 100 y 200.

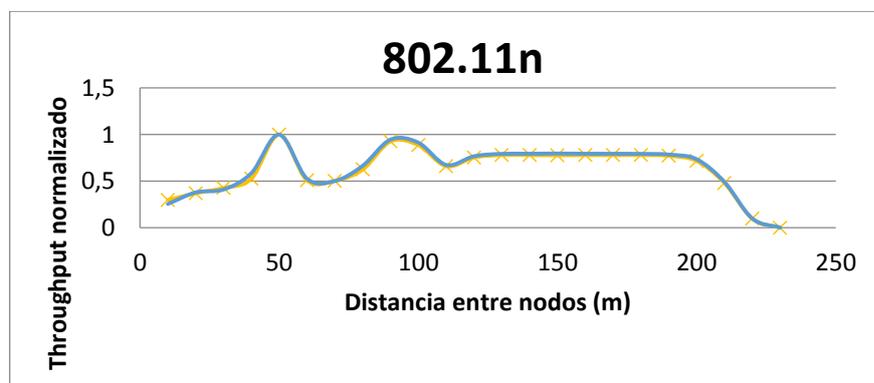


Figura 15-4 Comparación de Resultado Estándar 802.11n

Fuente: GALLEGOS, Abel, 2018

²⁵ User Datagram Protocol

Como se puede observar en la figura 15-4, se verifica que el mayor *throughput* se da aproximadamente a los 50 y 100 metros de distancia, teniendo un porcentaje aproximado al 100%. El cálculo se lo realizó de acuerdo al mayor resultado de *throughput* obtenido en cada simulación, por tal razón es que en ciertos casos existe un *throughput* mayor a 1. Adicional a entre los 120 y 200 m existe un *throughput* aproximado constante del 80%. Los puntos de más bajo *throughput* existen cuando los nodos se encuentran entre 10 y 40 m, entre 60 y 80m, y cuando sobrepasan los 200m de distancia entre nodos.

La curva es bastante parecida en cada simulación, es decir el *throughput* no depende del número de nodos, sino de la distancia en la que se instalen cada uno de ellos. La configuración óptima en el caso del 802.11n es no mayor a 200 m entre nodos.

Los datos de la simulación fueron tabulados según los resultados del ANEXO 802.11n WIFI - PROGRAMACIÓN NS3.

4.1.2 Simulación 802.15.1 Bluetooth

En la simulación realizada en el Software NS3, según ANEXO 802.15.1 BLE - PROGRAMACIÓN NS3, se tienen los siguientes resultados considerando al estándar 802.15.1, canal de comunicación con pérdidas características de acuerdo al espacio libre configurado según el radio 802.15.1, una carga útil de 20 Bytes, distancia variable entre 1 y 100 metros. Dentro de cada simulación se tiene como resultado el *throughput* normalizado como máximo valor 1.

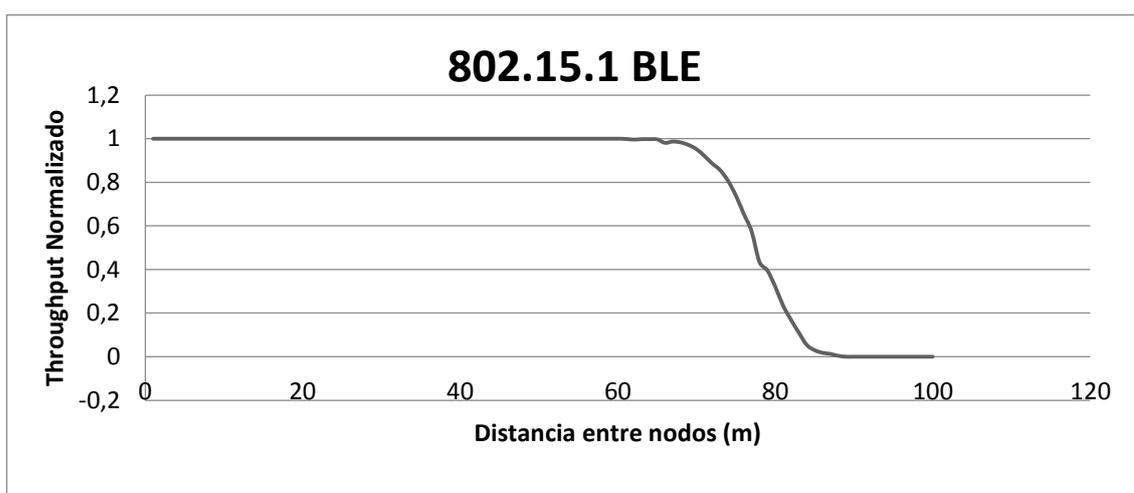


Figura 16-4 Comparación de Resultado Estándar 802.15.1 BLE

Fuente: GALLEGOS, Abel, 2018

Como se puede observar en la figura 16-4, se verifica que el mayor throughput se da aproximadamente entre 1 y 60 metros de distancia, teniendo un porcentaje aproximado al 100%. Adicional a entre los 61 y 70 m existe un throughput a la baja de hasta el 80%. A partir de los 71 m en adelante decrece hasta llegar a casi el 10%, mientras que si sobrepasan los 85m de distancia entre nodos, el throughput tiende a 0.

La curva es prácticamente idéntica en cada simulación, es decir el throughput no depende del número de nodos al igual que en el estándar 802.11n, sino de la distancia en la que se instalen cada uno de ellos. La ubicación óptima de los nodos sensores en el caso del 802.5.1 BLE y de acuerdo a los resultados, se daría en una distancia máxima de 80m.

Los datos de la simulación se tabularon según los resultados del ANEXO 802.15.1 BLE - PROGRAMACIÓN NS3.

4.1.3 Simulación 802.15.4 Zigbee

En la simulación realizada en el Software NS3, según el ANEXO 802.15.4 ZIGBEE - PROGRAMACIÓN NS3, se tienen los siguientes resultados considerando al estándar 802.15.4, canal de comunicación con pérdidas de acuerdo al espacio libre, configurado según el radio 802.15.4, una carga útil de 20 Bytes, distancia variable entre 1 y 140 metros. Dentro de cada simulación se tiene como resultado el throughput normalizado como máximo valor 1.

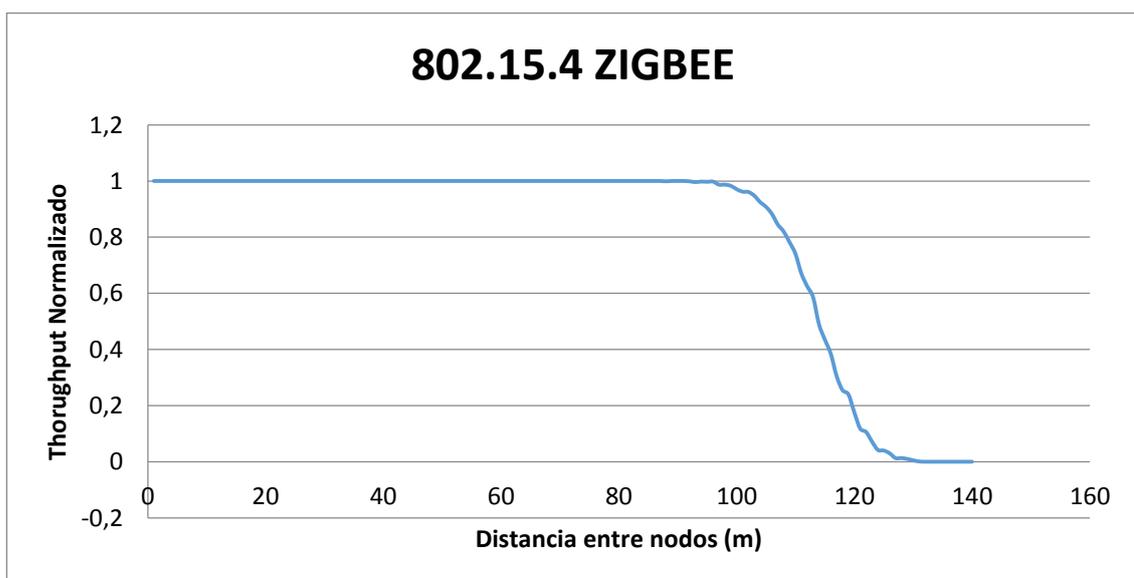


Figura 17-4 Comparación de Resultado Estándar 802.15.4 ZIGBEE

Fuente: GALLEGOS, Abel, 2018

Como se puede observar en la figura 17-4, se verifica que el mayor throughput se da entre 1 y 100 metros de distancia, teniendo un porcentaje de rendimiento de red del 100%. Adicional a entre los 101 y 110 m existe un throughput a la baja de hasta el 80%. A partir de los 111 m hasta los 120m decrece hasta llegar a casi el 20%, mientras que si sobrepasan los 120m de distancia entre nodos, el throughput tiende a 0.

La curva es prácticamente idéntica en cada simulación, es decir el throughput no depende del número de nodos al igual que en los estándares 802.11n y 802.15.1, sino de la distancia en la que se instalen cada uno de ellos. La ubicación óptima de los nodos sensores en el caso del 802.5.4 ZIGBEE y de acuerdo a los resultados, sería un máximo de distancia entre ellos de 120m.

Los datos de la simulación se tabularon de acuerdo a los resultados del ANEXO 802.15.4 ZIGBEE - PROGRAMACIÓN NS3.

Se escoge como estándar para los escenarios a construir al 802.11n debido a que tiene mayor distancia entre cada nodo sensor, adicionalmente es de los tres estándares el que permite la configuración de protocolos de ruteo, direccionamiento a nivel de red, mientras que 802.15.1 y 802.15.4 no permiten la configuración de protocolos de ruteo y sólo permiten direccionamiento a nivel MAC.

4.2 Elección de Escenarios

Como se pudo observar en la sección 3.1 se consideró tramos múltiplos de 400 m y un máximo tramo estudiado en el Río de 2800 m. Para esto se propone el estudio de los sensores a distancias de 100m con un máximo número de nodos de 29 (Nodos raíz o coordinador 0 y Nodo recurso 28), y 200m con un máximo número de nodos de 15 (Nodo raíz o coordinador 0 y Nodo recurso 15). Adicionalmente para cada una de las propuestas se verificarán tres protocolos de ruteo: AODV, DSDV y OLSR. El archivo base se encuentra en el ANEXO ESCENARIOS - PROGRAMACIÓN NS3.

Para la simulación y recolección de resultados se utilizaron las siguientes herramientas:

- Network Simulator V 3.19 y su herramienta WAF (Simulación);
- WireShark Versión 2.4.5 (Tratamiento de Archivos PCAP);
- NetAnim y FlowMonitor (Animación y verificación de Escenarios, saltos y otros parámetros de red);

- Tracemetrics 1.4.0 (Tratamiento de archivos de trazas).

4.2.1 Escenario 1 - Número de Nodos 29, Distancia entre nodos 100m (OLSR)

En este escenario, se propone que existan 29 Nodos sensores, ubicados con una separación entre ellos de 100 m. El Nodo raíz es el Nodo 0, mismo que va a encontrarse lo más cercano a la población Alluriquín. El nodo 28 es el nodo recurso, es decir el más lejano, el que cubrirá los 2800m de río.

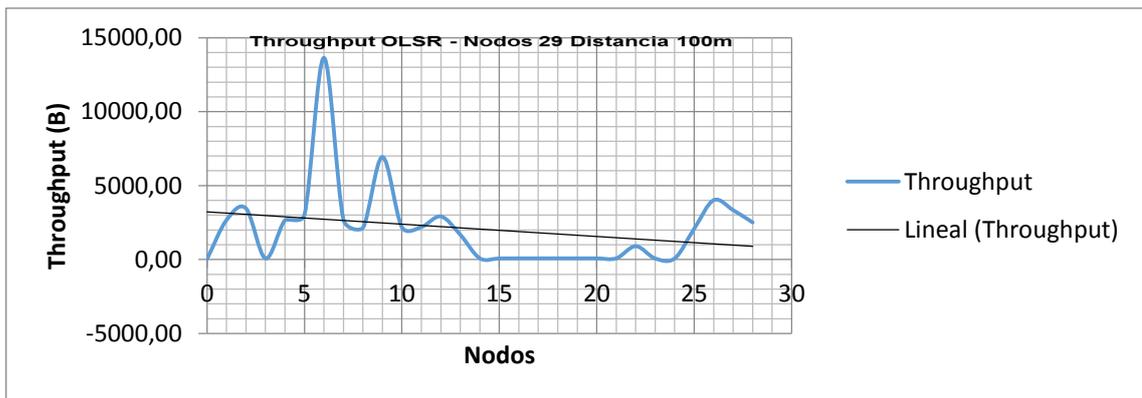


Figura 18-4 Throughput OLSR - 29 Sensores - Estándar 802.11n distancia 100m

Fuente: GALLEGOS, Abel, 2018

Como se muestra en la figura 18-4, existe variación en el throughput (medido en Bytes), utilizando el Protocolo OLSR, con una media decreciente a medida que se aleja del nodo raíz entre 3KBytes y 1KBytes.

No.	Time	Source	Destination	Protocol	Length	Info
88	30.050815	192.168.1.3	192.168.1.255	OLSR v1	177	OLSR (IPv4) Packet, Length: 84 Bytes
89	30.050832	00:00:00_00:00:04	00:00:00_00:00:07	ARP	93	192.168.1.4 is at 00:00:00:00:00:04
90	30.052592	00:00:00_00:00:07	00:00:00_00:00:07	802.11	38	Acknowledgement, Flags=0.....
91	30.057572	00:00:00_00:00:04	Broadcast	ARP	93	Who has 192.168.1.1? Tell 192.168.1.4
92	30.057842	00:00:00_00:00:01	00:00:00_00:00:04	ARP	91	192.168.1.1 is at 00:00:00:00:00:01
93	30.058442	00:00:00_00:00:01	00:00:00_00:00:04	ARP	91	192.168.1.1 is at 00:00:00:00:00:01
94	30.058634	00:00:00_00:00:01	00:00:00_00:00:01	802.11	38	Acknowledgement, Flags=0.....
95	30.060192	192.168.1.29	192.168.1.1	802.11	36	Acknowledgement, Flags=0.....

```

> Frame 95: 1093 bytes on wire (8744 bits), 1093 bytes captured (8744 bits)
> Radiotap Header v0, Length 27
  802.11 radio information
    PHY type: 802.11n (7)
    MCS index: 0
    Bandwidth: 20 MHz (0)
    Short GI: False
    Greenfield: False
    FEC: BEC (0)
    Number of STBC streams: 0
    Number of extension spatial streams: 0
    Data rate: 6.5 Mb/s
    Channel: 1
    Frequency: 2412MHz
    Signal strength (dBm): -90dBm
    Noise level (dBm): -94dBm
    TSF timestamp: 30072787
  [Duration: 1352µs]
  IEEE 802.11 QoS Data, Flags: .....
    Logical-link Control
    Internet Protocol Version 4, Src: 192.168.1.29, Dst: 192.168.1.1
    User Datagram Protocol, Src Port: 49153, Dst Port: 80
    QUIC (Quick UDP Internet Connections)
  
```

Figura 19-4 Archivo PCAP OLSR Nodo 1 Distancia 100m

Fuente: GALLEGOS, Abel, 2018

De acuerdo a la Figura 19-4, se puede verificar que los parámetros de transmisión son los siguientes: El tipo de medio físico es 802.11n, con índice MCS 0, Ancho de Banda de 20MHz, Tasa de Datos de 6,5Mbps, Canal 1, Frecuencia del Canal 2412 MHz, Intensidad de Señal de -90dBm, nivel de ruido de -94dBm y una duración total de la transmisión de 1352 μ s. La Dirección IP de origen es 192.168.1.29 (Nodo 28) y el destino 192.168.1.1 (Nodo 0 o raíz). Se utilizó como protocolo de transporte UDP, puerto origen 49153 y destino 80.

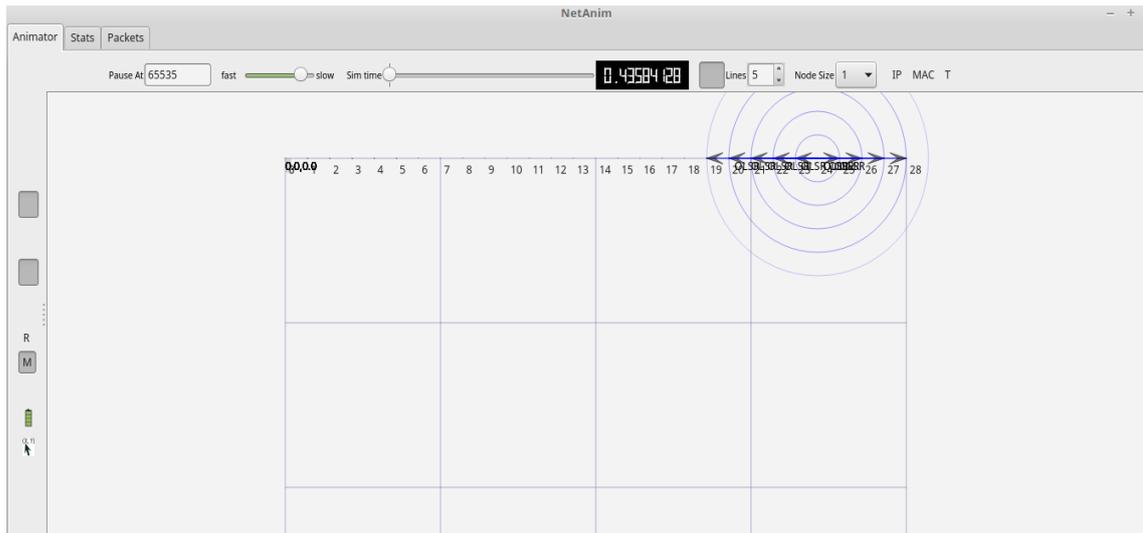


Figura 20-4 Archivo XML OLSR Nodos 29 Distancia 100m en NetAnim

Fuente: GALLEGOS, Abel, 2018

En la figura 20-4 se observa cómo se comporta el salto de paquetes entre nodos, a través del protocolo OLSR, verificando 29 nodos (Nodos del 0 al 28). Se toma en cuenta una distancia lineal, dado que en el peor de los casos la transmisión se daría en línea recta y teniendo en cuenta que el ángulo que forman los cambios en los tramos estudiados en el Río Damas es mayor a 90 grados, significa que la suma de los lados que forman el mayor ángulo no es mayor al valor del lado opuesto al ángulo antes descrito.

4.2.2 Escenario 2 - Número de Nodos 29, Distancia entre nodos 100m (AODV)

En este escenario, se propone que existan 29 Nodos sensores, ubicados con una separación entre ellos de 100 m. El Nodo raíz es el Nodo 0, mismo que va a encontrarse lo más cercano a la población Alluriquín. El nodo 28 es el nodo recurso, es decir el más lejano, el que estará más alejado a los 2800m de río.

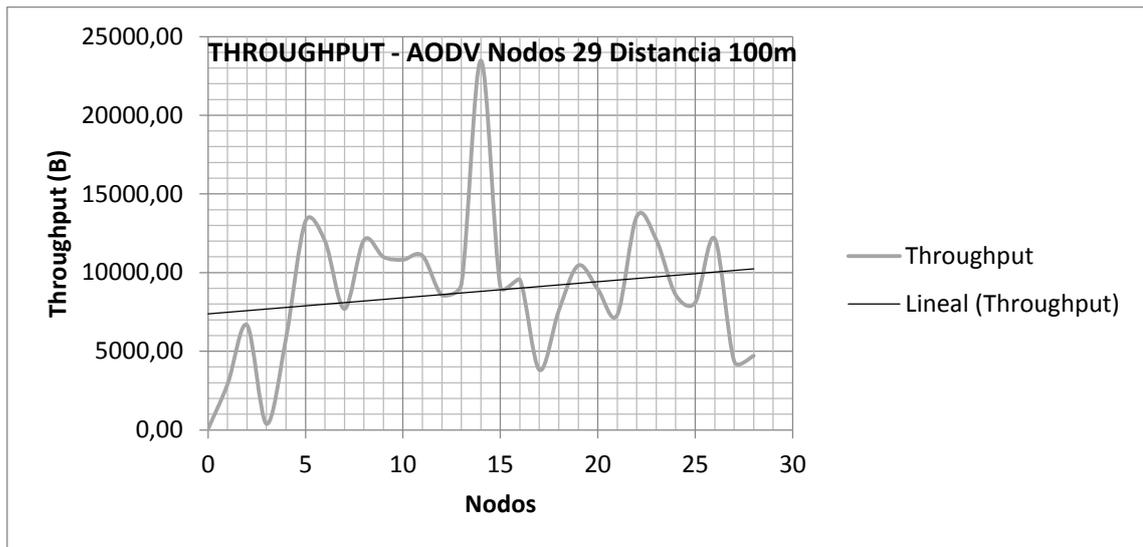


Figura 21-4 Throughput AODV - 29 Sensores - Estándar 802.11n distancia 100m

Fuente: GALLEGOS, Abel, 2018

Como se muestra en la figura 21-4, existe variación en el throughput (medido en Bytes), utilizando el Protocolo AODV, con variación creciente entre 7,5KBytes y 10KBytes.

No.	Time	Source	Destination	Protocol	Length	Info
145	31.972197		00:00:00_00:00:01 (.. 802.11)	802.11	38	Acknowledgement, Flags=0.....
146	31.972441	192.168.1.2	192.168.1.1	AODV	113	Route Reply, D: 192.168.1.29, O: 192.168.1.1 Hcnt=11 DSN=5 Lifetime=4719
147	31.972451		00:00:00_00:00:02 (.. 802.11)	802.11	36	Acknowledgement, Flags=0.....
148	31.973637	00:00:00_00:00:02	Broadcast	ARP	93	Who has 192.168.1.5? Tell 192.168.1.2
149	31.973959		00:00:00_00:00:05 (.. 802.11)	802.11	38	Acknowledgement, Flags=0.....
150	31.974173	192.168.1.2	192.168.1.1	802.11	113	Route Reply, D: 192.168.1.1, O: 192.168.1.29 Hcnt=1 DSN=0 Lifetime=1030
151	31.975086		00:00:00_00:00:06 (.. 802.11)	802.11	38	Acknowledgement, Flags=0.....


```

> Frame 150: 113 bytes on wire (904 bits), 113 bytes captured (904 bits)
> Radiotap Header v0, Length 27
4 802.11 radio information
  PHY type: 802.11n (7)
  MCS index: 0
  Bandwidth: 20 MHz (0)
  Short GI: False
  Greenfield: False
  FEC: BEC (0)
  Number of STBC streams: 0
  Number of extension spatial streams: 0
  Data rate: 6.5 Mb/s
  Channel: 1
  Frequency: 2412MHz
  Signal strength (dBm): -81dBm
  Noise level (dBm): -94dBm
  TSF timestamp: 31978323
  [Duration: 148µs]
  IEEE 802.11 QoS Data, Flags: .....
  Logical-Link Control
  Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.5
  User Datagram Protocol, Src Port: 654, Dst Port: 654
  Ad hoc On-demand Distance Vector Routing Protocol, Route Reply, Dest IP: 192.168.1.1, Orig IP: 192.168.1.29, Lifetime=1030
  
```

Figura 22-4 Archivo PCAP AODV Nodos 29 Distancia 100m

Fuente: GALLEGOS, Abel, 2018

De acuerdo a la Figura 22-4, se puede verificar que los parámetros de transmisión son los siguientes: El tipo de medio físico es 802.11n, con índice MCS 0, Ancho de Banda de 20MHz, Tasa de Datos de 6,5Mbps, Canal 1, Frecuencia del Canal 2412 MHz, Intensidad de Señal de -81dBm, nivel de ruido de -94dBm y una duración total de la transmisión de 148 μ s. La Dirección IP de origen es 192.168.1.29 (Nodo 28) y el destino 192.168.1.1 (Nodo 0 o raíz). Se utilizó como protocolo de transporte UDP, puerto origen 49153 y destino 80. En esta ocasión se verifica una trama AODV, la cual pasa desde el origen 192.168.1.2 al destino 192.168.1.5, con los puertos origen y destino UPD 654.

4.2.3 Escenario 3 - Número de Nodos 29, Distancia entre nodos 100m (DSDV)

En este escenario, se propone que existan 29 Nodos sensores WI-FI, ubicados con una separación entre ellos de 100 m. El Nodo raíz es el Nodo 0, mismo que va a encontrarse lo más cercano a la población Alluriquín. El nodo 28 es el nodo recurso, es decir el más lejano, el que estará a 2800m del nodo raíz.

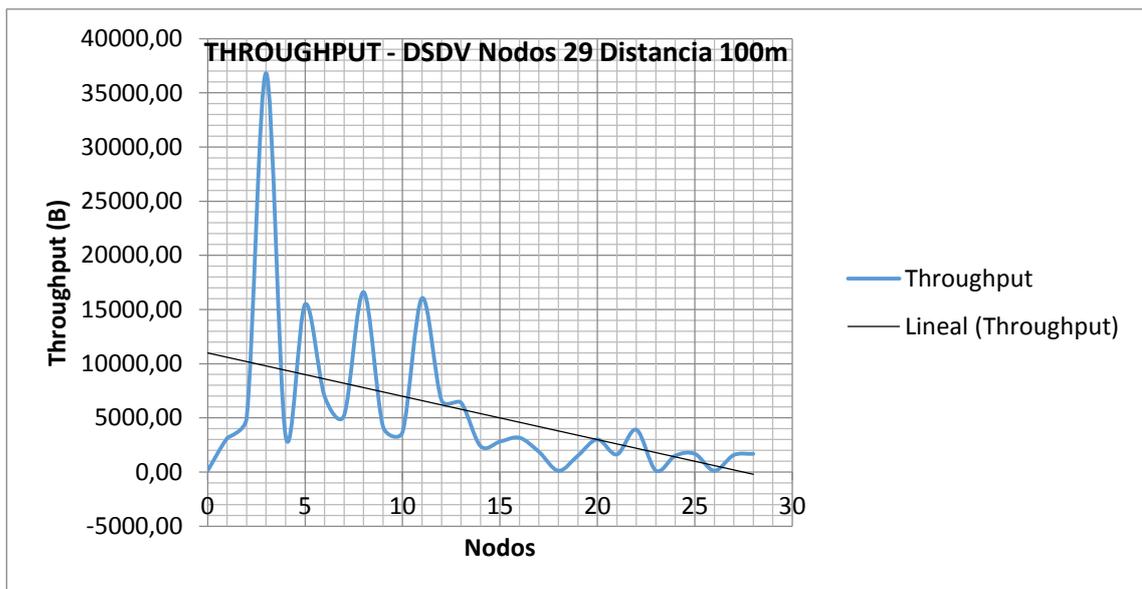


Figura 23-4 Throughput DSDV- 29 Sensores - Estándar 802.11n distancia 100m

Fuente: GALLEGOS, Abel, 2018

Como se muestra en la figura 23-4, existe variación en el throughput (medido en Bytes), utilizando el Protocolo DSDV, con una media que baja desde 10KBytes hasta 0 Bytes a medida que la red de nodos es más lejana, es decir la comunicación entre el nodo coordinador y el recurso es nula.

No.	Time	Source	Destination	Protocol	Length	Info
378	30.125191	192.168.1.3	192.168.1.255	packetbb	117	
379	30.125649	192.168.1.2	192.168.1.255	packetbb	117	
380	30.125938	192.168.1.4	192.168.1.255	packetbb	117	
381	30.126048	192.168.1.1	192.168.1.255	packetbb	115	
382	30.126496	192.168.1.1	192.168.1.255	packetbb	115	
383	30.127254	192.168.1.4	192.168.1.255	packetbb	117	
384	30.127702	192.168.1.2	192.168.1.255	packetbb	117	


```

Frame 384: 117 bytes on wire (936 bits), 117 bytes captured (936 bits)
  Radiotap Header v0, Length 27
    802.11 radio information
      PHY type: 802.11n (7)
      MCS index: 0
      Bandwidth: 20 MHz (0)
      Short GI: False
      Greenfield: False
      FEC: BEC (0)
      Number of STBC streams: 0
      Number of extension spatial streams: 0
      Data rate: 6.5 Mb/s
      Channel: 1
      Frequency: 2412MHz
      Signal strength (dBm): -81dBm
      Noise level (dBm): -94dBm
      TSF timestamp: 30128066
      [Duration: 152µs]
    IEEE 802.11 QoS Data, Flags: .....
    Logical-Link Control
    Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.255
    User Datagram Protocol, Src Port: 269, Dst Port: 269
    PacketBB Protocol
  
```

Figura 24-4 Archivo PCAP DSDV Nodos 29 Distancia 100m

Fuente: GALLEGOS, Abel, 2018

De acuerdo a la Figura 24-4, se puede verificar que los parámetros de transmisión son los siguientes: El tipo de medio físico es 802.11n, con índice MCS 0, Ancho de Banda de 20MHz, Tasa de Datos de 6,5Mbps, Canal 1, Frecuencia del Canal 2412 MHz, Intensidad de Señal de -81dBm, nivel de ruido de -94dBm y una duración total de la transmisión de 152 µs. En este caso WireShark no permite verificar más datos de la transmisión a través de este protocolo, como se puede observar en la Figura 4-10 existen fallas en comunicación (WireShark color rojo).

4.2.4 Escenario 4 - Número de Nodos 15, Distancia entre nodos 200m (OLSR)

En este escenario, se propone 15 Nodos sensores, ubicados con una separación lineal entre ellos de 200 m. El Nodo raíz es el Nodo 0, mismo que va a encontrarse lo más cercano a la población Alluriquín. El nodo 14 es el nodo recurso, es decir el más lejano, el que estará más alejado a los 2800m de río.

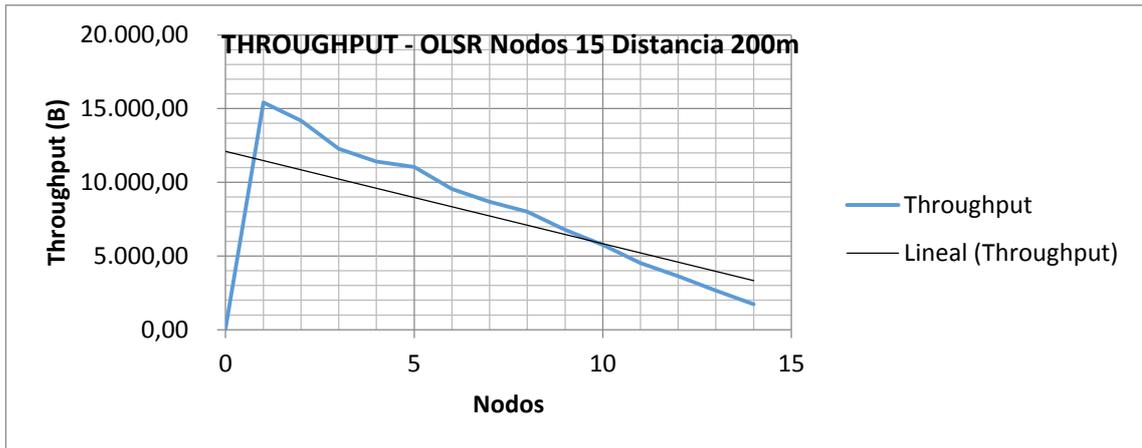


Figura 25-4 Throughput OLSR - 15 Sensores - Estándar 802.11n distancia 200m

Fuente: GALLEGOS, Abel, 2018

Como se muestra la figura 25-4, existe variación en el *throughput* (medido en Bytes), utilizando el Protocolo OLSR, con una media que va bajando a medida que se aleja del nodo raíz de entre 12KBytes hasta los 3KBytes aproximadamente.

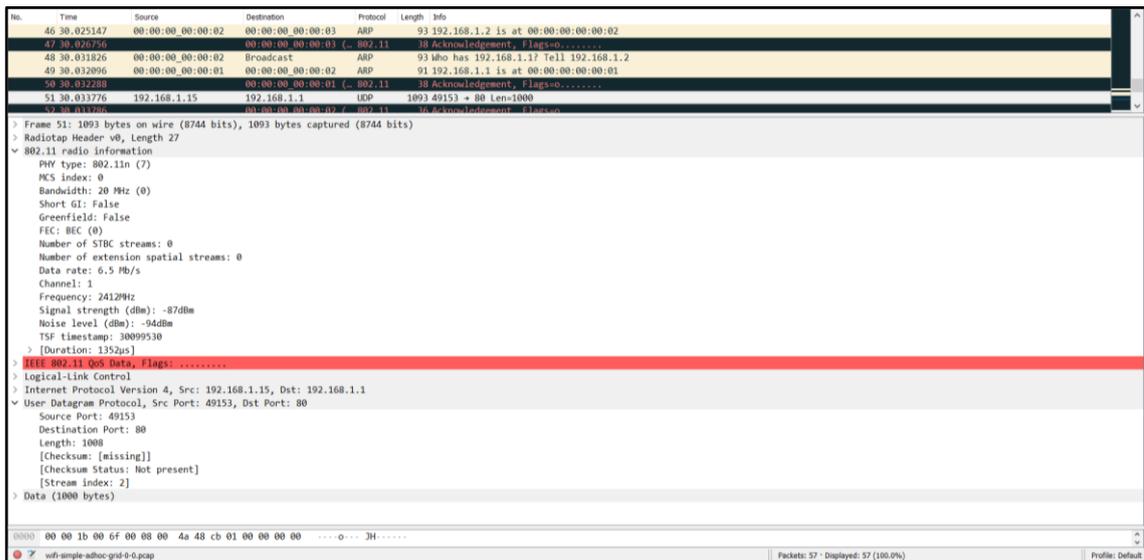


Figura 26-4 Archivo PCAP OLSR Nodo 1 Distancia 200m

Fuente: GALLEGOS, Abel, 2018

De acuerdo a la Figura 26-4, se puede verificar que los parámetros de transmisión son los siguientes: El tipo de medio físico es 802.11n, con índice MCS 0, Ancho de Banda del canal 20MHz, Tasa de Datos de 6,5Mbps, Canal 1, Frecuencia del Canal 2412 MHz, Intensidad de Señal de -87dBm, nivel de ruido de -94dBm y una duración total de la transmisión de 1352 μ s. La Dirección IP de origen es 192.168.1.15 (Nodo 14) y el destino 192.168.1.1 (Nodo 0 o raíz).

Se utilizó como protocolo de transporte UDP, puerto origen 49153 y destino 80 y el Protocolo de ruteo OLSR.

4.2.5 Escenario 5 - Número de Nodos 15, Distancia entre nodos 200m (AODV)

En este escenario, se propone que existan 15 Nodos sensores, ubicados con una separación entre ellos de 200 m. El Nodo raíz es el Nodo 0, mismo que va a encontrarse lo más cercano a la población Alluriquín. El nodo 14 es el nodo recurso, es decir el más lejano, el que estará más alejado a los 2800m de río, protocolo de ruteo AD-hoc On Demand Vector (AODV).

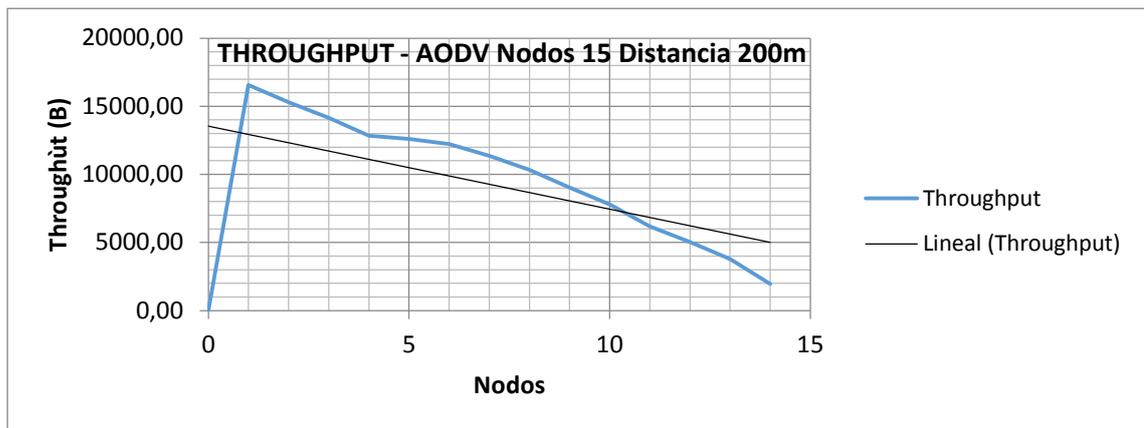


Figura 27-4 Throughput AODV - 15 Sensores - Estándar 802.11n distancia 200m

Fuente: GALLEGOS, Abel, 2018

Como se muestra en la figura 27-4, existe variación en el throughput (medido en Bytes), utilizando el Protocolo AODV, con una media entre 14KBytes y 5Kbytes.

No.	Time	Source	Destination	Protocol	Length	Info
68	31.994804	192.168.1.2	192.168.1.1	AODV	113	Route Reply, D: 192.168.1.15, O: 192.168.1.1 Hcnt=13 DSN=5 Lifetime=4559
69	31.994814		00:00:00_00:00:02 (- 802.11	38	Acknowledgement, Flags=0.....	
70	31.998039	00:00:00_00:00:02	00:00:00_00:00:03 (- 802.11	93	Who has 192.168.1.3? Tell 192.168.1.2	
71	31.998600		00:00:00_00:00:03 (- 802.11	38	Acknowledgement, Flags=0.....	
73	32.001376	00:00:00_00:00:02	00:00:00_00:00:03	ARP	93	192.168.1.2 is at 00:00:00:00:00:02
74	32.001837		00:00:00_00:00:03 (- 802.11	38	Acknowledgement, Flags=0.....	
75	32.003000	192.168.1.2	192.168.1.255	AODV	113	Route Reply, D: 192.168.1.2, O: 192.168.1.2 Hcnt=0 DSN=0 Lifetime=2000

```

Frame 72: 113 bytes on wire (904 bits), 113 bytes captured (904 bits)
  Radiotap Header v0, Length 27
    802.11 radio information
      PHY type: 802.11n (7)
      MCS index: 0
      Bandwidth: 20 MHz (0)
      Short GI: False
      Greenfield: False
      FEC: BEC (0)
      Number of STBC streams: 0
      Number of extension spatial streams: 0
      Data rate: 6.5 Mb/s
      Channel: 1
      Frequency: 2412MHz
      Signal strength (dBm): -87dBm
      Noise level (dBm): -94dBm
      TSF timestamp: 32013198
      [Duration: 148µs]
    IEEE 802.11 QoS Data, Flags: .....
      Logical Link Control
      Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.3
      User Datagram Protocol, Src Port: 654, Dst Port: 654
      Ad hoc On-demand Distance Vector Routing Protocol, Route Reply, Dest IP: 192.168.1.1, Orig IP: 192.168.1.15, Lifetime=1055
  
```

Figura 28-4 Archivo PCAP AODV Nodos 15 Distancia 200m

Fuente: GALLEGOS, Abel, 2018

De acuerdo a la Figura 28-4, se puede verificar que los parámetros de transmisión son los siguientes: El tipo de medio físico es 802.11n, con índice MCS 0, Ancho de Banda del canal 20MHz, Tasa de Datos de 6,5Mbps, Canal 1, Frecuencia del Canal 2412 MHz, Intensidad de Señal de -87dBm, nivel de ruido de -94dBm y una duración total de la transmisión de 148 μ s. La Dirección IP de origen es 192.168.1.15 (Nodo 14) y el destino 192.168.1.1 (Nodo 0 o raíz). Se utilizó como protocolo de transporte UDP, puerto origen 654 y destino 654. En esta ocasión se verifica una trama AODV, la cual se verifica la última línea de la Figura 28-4.

4.2.6 Escenario 6 - Número de Nodos 15, Distancia entre nodos 200m (DSDV)

En este escenario, se propone que existan 15 Nodos sensores, ubicados con una separación entre ellos de 200 m. El Nodo raíz es el Nodo 0, mismo que va a encontrarse lo más cercano a la población Alluriquín. El nodo 14 es el nodo recurso, es decir el más lejano, el que estará más alejado a los 2800m de río, protocolo de ruteo DSDV.

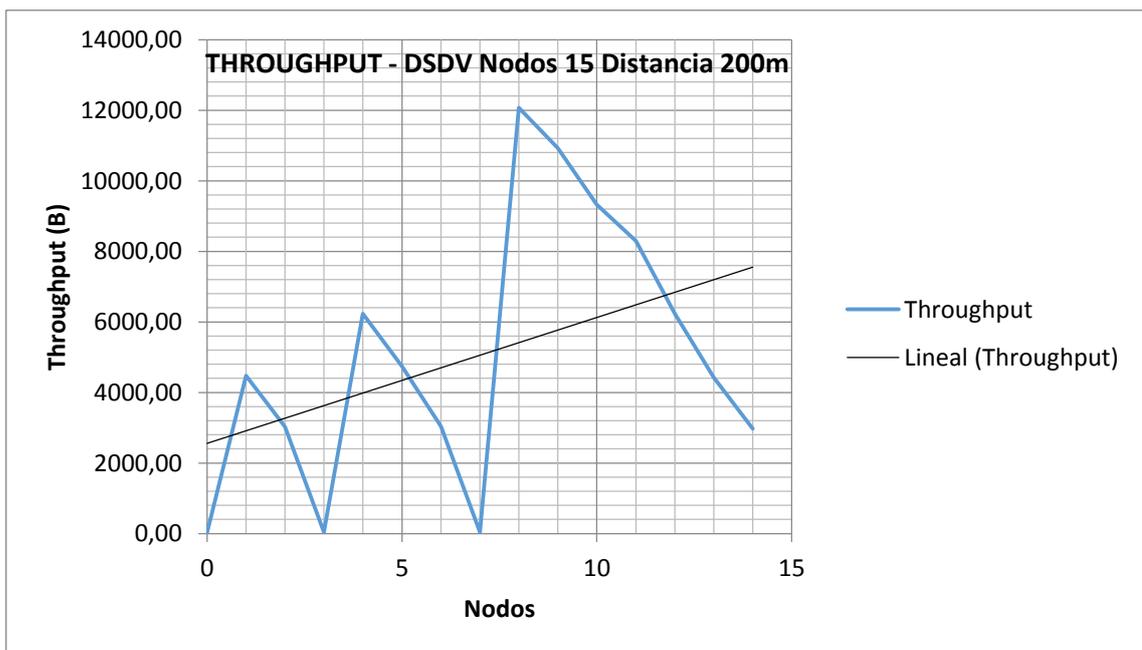


Figura 29-4 Throughput DSDV - 15 Sensores - Estándar 802.11n distancia 200m

Fuente: GALLEGOS, Abel, 2018

Como se muestra en la figura 29-4, existe variación en el throughput (medido en Bytes), utilizando el Protocolo DSDV, con una media creciente entre 2KBytes y 8KBytes.

No.	Time	Source	Destination	Protocol	Length	Info
52	30.054067	192.168.1.2	192.168.1.255	packet...	117	
53	30.054375	192.168.1.1	192.168.1.255	packet...	115	
54	30.099939	00:00:00_00:00:02	00:00:00_00:00:03	ARP	93	192.168.1.2 is at 00:00:00:00:00:02
55	30.101548	00:00:00_00:00:02	00:00:00_00:00:03	(... 802.11	38	Acknowledgement, Flags=0...
56	30.106718	00:00:00_00:00:02	Broadcast	ARP	93	Who has 192.168.1.1? Tell 192.168.1.2
57	30.106808	00:00:00_00:00:01	00:00:00_00:00:02	ARP	91	192.168.1.1 is at 00:00:00:00:00:01
58	30.106999	00:00:00_00:00:01	00:00:00_00:00:01	(... 802.11	38	Acknowledgement, Flags=0...
60	30.108617	00:00:00_00:00:01	00:00:00_00:00:02	(... 802.11	36	Acknowledgement, Flags=0...


```

Frame 59: 1093 bytes on wire (8744 bits), 1093 bytes captured (8744 bits)
Radiotap Header v0, Length 27
802.11 radio information
  PHY type: 802.11n (7)
  MCS index: 0
  Bandwidth: 20 MHz (0)
  Short GI: False
  Greenfield: False
  FEC: BEC (0)
  Number of STBC streams: 0
  Number of extension spatial streams: 0
  Data rate: 6.5 Mb/s
  Channel: 1
  Frequency: 2412MHz
  Signal strength (dBm): -87dBm
  Noise level (dBm): -94dBm
  TSF timestamp: 30108999
  [Duration: 1352µs]
IEEE 802.11 QoS Data, Flags: .....
Logical-Link Control
Internet Protocol Version 4, Src: 192.168.1.15, Dst: 192.168.1.1
User Datagram Protocol, Src Port: 49153, Dst Port: 80
QUIC (Quick UDP Internet Connections)
  
```

Figura 30-4 Archivo PCAP DSDV Nodos 15 Distancia 200m

Fuente: GALLEGOS, Abel, 2018

De acuerdo a la Figura 30-4, se puede verificar que los parámetros de transmisión son los siguientes: El tipo de medio físico es 802.11n, con índice MCS 0, Ancho de Banda del canal 20MHz, Tasa de Datos de 6,5Mbps, Canal 1, Frecuencia del Canal 2412 MHz, Intensidad de Señal de -87dBm, nivel de ruido de -94dBm y una duración total de la transmisión de 1352 μ s. La Dirección IP de origen es 192.168.1.15 (Nodo 14) y el destino 192.168.1.1 (Nodo 0 o raíz). Se utilizó como protocolo de transporte UDP, puerto origen 49153 y destino 80. En esta ocasión se verifica una trama QUIC, la cual se puede observar en la última línea de la Figura 30-4.

4.3 Análisis de Resultados

Para el análisis de los resultados se tomará en cuenta los parámetros: Distancia, Número de Nodos, Protocolo de Ruteo, Retardo y Rendimiento de la Red.

Tabla 4-4: Throughput de acuerdo a Escenarios

Protocolo Ruteo – Número de Nodos	OLSR – Nodos 29 (Escenario o 1)	AODV – Nodos 29 (Escenario 2)	DSDV – Nodos 29 (Escenario 3)	OLSR – Nodos 15 (Escenario 4)	AODV – Nodos 15 (Escenario 5)	DSDV – Nodos 15 (Escenario 6)
Distancia(m)	Throughput (Bytes)	Throughput (Bytes)	Throughput (Bytes)	Throughput (Bytes)	Throughput (Bytes)	Throughput (Bytes)
100	2670,39	2902,62	3122,68	-	-	-
200	3460,21	6660,86	4975,08	15.425,69	16568,17	4476,38
300	79,22	385,24	36818,79	-	-	-
400	2656,50	5824,17	3314,48	14.188,47	15308,39	3031,29
500	3024,52	13269,45	15482,08	-	-	-
600	13639,93	12007,20	6981,35	12.286,68	14158,69	40,20
700	2704,50	7690,65	5221,97	-	-	-
800	2147,78	12066,92	16606,18	11.410,77	12843,66	6233,43
900	6938,45	10996,07	4188,81	-	-	-
1000	2191,52	10817,73	3674,02	11.048,36	12594,00	4734,92
1100	2200,45	11073,83	16044,27	-	-	-
1200	2916,84	8559,72	6621,53	9.556,87	12223,93	3033,49
1300	1675,23	9256,41	6409,96	-	-	-
1400	84,86	23478,23	2392,47	8.680,96	11362,56	38,68
1500	79,22	9091,67	2801,33	-	-	-
1600	82,79	9553,91	3163,49	8.014,19	10341,50	12064,87
1700	82,10	3814,13	1863,25	-	-	-
1800	84,03	7583,44	118,29	6.776,97	9046,31	10928,70
1900	81,00	10451,89	1511,95	-	-	-
2000	84,30	8942,78	3007,00	5.749,98	7796,66	9327,77
2100	84,30	7314,63	1620,90	-	-	-
2200	904,25	13571,32	3903,05	4.513,86	6188,07	8291,83
2300	75,50	12088,17	130,93	-	-	-
2400	83,20	8578,38	1512,50	3.639,06	5025,94	6229,02
2500	2089,33	8094,27	1672,42	-	-	-
2600	4018,86	12156,98	118,98	2.663,53	3787,43	4422,14
2700	3340,84	4365,00	1565,53	-	-	-
2800	2511,82	4711,36	1674,89	1.726,70	1972,81	2973,88
PROMEDIO	2142,57	9118,11	5589,94	8.263,01	9944,15	5416,19
PONDERACIÓN	6	2	4	3	1	5

Fuente: GALLEGOS, Abel, 2018

La Ponderación 1 corresponde al mejor escenario de acuerdo a los valores promedio y la ponderación 6 al peor. Como se puede observar en la Tabla 4-4, el mayor throughput corresponde al escenario 5 con el protocolo AODV con 15 Nodos, el segundo pertenece al escenario 2 con el protocolo AODV con 29 Nodos. El peor *Throughput* corresponde a OLSR con 29 Nodos.

Como se verifica en la Tabla 5-4, según los parámetros de Red (DELAY, TX BITRATE, RX BITRATE), el escenario que tiene mejor rendimiento corresponde al Escenario 4 con el Protocolo OLSR y 15 Nodos. El peor escenario corresponde a los Escenarios 1, 2, 3 y 6, puesto que no completan la transmisión entre el nodo raíz y el nodo recurso.

Tabla 5-4: Parámetros de Red (DELAY, TX BITRATE, RX BITRATE)

Transmisión UDP	UDP 192.168.1.29/49153---- >192.168.1.1/80			UDP 192.168.1.15/49153---- >192.168.1.1/80		
	OLSR – Nodos 29 (Escenario 1)	AODV – Nodos 29 (Escenario 2)	DSDV – Nodos 29 (Escenario 3)	OLSR – Nodos 15 (Escenario 4)	AODV – Nodos 15 (Escenario 5)	DSDV – Nodos 15 (Escenario 6)
Mean Delay	Sin Valor	Sin Valor	Sin Valor	154.665ms	1117.84ms	Sin Valor
Tx bitrate	12.4286kbps	12.4286kbps	12.4286kbps	12.4286kbps	12.4286kbps	12.4286kbps
Rx bitrate	0kbps	0kbps	0kbps	12.7713kbps	8.85301kbps	0kbps
PONDERACIÓN	3	3	3	1	2	3

Fuente: GALLEGOS, Abel, 2018

Como se verifica en la Tabla 6-4, según los parámetros de Red (TX PACKETS, RX PACKETS, JITTER), el escenario que tiene mejor rendimiento corresponde al Escenario 5 con el Protocolo OLSR y 15 Nodos. El peor escenario corresponde a los Escenarios 1, 2, 3 y 6.

Tabla 6-4: Parámetros de Red (TX PACKETS, RX PACKETS, JITTER)

Transmisión UDP	UDP 192.168.1.29/49153---->192.168.1.1/9			UDP 192.168.1.15/49153---->192.168.1.1/9		
Protocolo Ruteo – Número de Nodos	OLSR – Nodos 29 (Escenario 1)	AODV – Nodos 29 (Escenario 2)	DSDV – Nodos 29 (Escenario 3)	OLSR – Nodos 15 (Escenario 4)	AODV – Nodos 15 (Escenario 5)	DSDV – Nodos 15 (Escenario 6)
txPackets	29	29	29	29	29	29
rxPackets	0	0	0	17	15	0
jitterSum	Sin Valor	Sin Valor	Sin Valor	1.32705e+08n s	1.04304e+10n s	Sin Valor
PONDERACIÓN	3	3	3	1	2	3

Fuente: GALLEGOS, Abel, 2018

De acuerdo la ponderación Total verificada en la Tabla 7-4, existe muy poca diferencia entre los escenarios 4 y 5, ya que el escenario 4 a pesar de no alcanzar el mejor throughput, tiene mayor rendimiento en cuanto a paquetes enviados, y menor delay que el escenario 5. Mientras que el escenario 5 obtiene un mejor throughput pero obtiene una menor tasa de transmisión. Adicionalmente se puede verificar que entre el escenario 2 y el 5, se tiene el mismo protocolo de enrutamiento con diferentes números de nodos, pero ambos con mayor throughput.

Tabla 7-4: Resumen Ponderaciones

Transmisión UDP	UDP 192.168.1.29/49153---->192.168.1.1/9			UDP 192.168.1.15/49153---->192.168.1.1/9		
Protocolo Ruteo – Número de Nodos	OLSR – Nodos 29 (Escenario 1)	AODV – Nodos 29 (Escenario 2)	DSDV – Nodos 29 (Escenario 3)	OLSR – Nodos 15 (Escenario 4)	AODV – Nodos 15 (Escenario 5)	DSDV – Nodos 15 (Escenario 6)
PONDERACIÓN (Tabla 4-1)	6	2	4	3	1	5
PONDERACIÓN (Tabla 4-2)	3	3	3	1	2	3
PONDERACIÓN (Tabla 4-3)	3	3	3	1	2	3
TOTAL	12	8	10	5	5	11

Fuente: GALLEGOS, Abel, 2018

CONCLUSIONES

Las condiciones climáticas como humedad, temperatura y la vegetación se han caracterizado en los parámetros de red y se ha escogido el modo de propagación de acuerdo al espacio libre. Para esto los datos a transmitir son los niveles de agua del río medidos en cada uno de los nodos sensores, así como las direcciones IP de cada uno, como referencia de la distancia total existente entre el nodo coordinador y el nodo recurso.

Una RSI, es una red flexible, adaptable y que permite la utilización de varios tipos de estándares para la transmisión de datos de los nodos sensores. Existen en el mercado gran variedad de tipos de sensores acorde a la aplicación que se vaya a estudiar. La optimización o creación de un SAT, se la puede realizar a través de una RSI.

De los tres estándares analizados (802.11n, 802.15.1, y 802.15.4) para la interconexión de la Red de Sensores Inalámbricos propuesta en el Río Damas, el que tiene mayores prestaciones en cuanto a distancia y throughput es el 802.11n conocido como WI-FI.

La herramienta NS3 permite la configuración de los escenarios de acuerdo a la realidad del Río Damas y a los parámetros de configuración de los estándares de comunicación y de la Red de Sensores Inalámbricos.

De acuerdo a los resultados de las simulaciones en base a los escenarios propuestos, el que mayor rendimiento en relación al retardo y jitter es el Escenario que maneja el protocolo OLSR de 15 nodos con una distancia de separación de 200 metros, sin embargo en base el escenario con 15 nodos a una distancia de separación de 200 metros y a través del protocolo AODV permite el mayor throughput de red, además que genera menor tráfico a la llegada de los paquetes en el receptor, debido a ser un protocolo reactivo. Adicionalmente se puede realizar una combinación entre los escenarios de 100 y 200 metros tomando en cuenta el protocolo AODV, ya que el throughput para dichos escenarios es el mayor de los tres estándares analizados (802.11n, 802.15.1, y 802.15.4) para la interconexión de la Red de Sensores Inalámbricos propuesta en el Río Damas, el que tiene mayores prestaciones en cuanto a distancia y *throughput* es el 802.11n.

RECOMENDACIONES

Para la implementación de la nueva red de Sensores Inalámbricos Propuesta, se recomienda que la interconexión se la realice a través del estándar 802.11n y el protocolo AODV, el mismo que tiene las mejores prestaciones de acuerdo a este estudio.

De acuerdo a las condiciones de acceso se recomienda que cada radio transmisor 802.11n se encuentre instalado sobre un mástil de al menos 3 metros para reducir las pérdidas por obstáculos.

Extender el estudio a otros ríos de similares características con el objeto de plantear nuevas redes de sensores inalámbricos que permitan alertas hacia la población.

Para una mayor comprensión de los datos obtenidos en la simulación, utilizar herramientas adicionales para la animación de escenarios de red con sus respectivos parámetros de configuración y análisis de resultados

BIBLIOGRAFÍA

- Bluetooth Sig, P. (06 de Diciembre de 2016). Bluetooth Core Specification. *Covered Core Package Version: 5.0*.
- Capella H, J. V. (Abril de 2010). Redes inalámbricas de sensores: una nueva arquitectura eficiente y robusta basada en jerarquía dinámica de grupos. Valencia, España.
- Chacin, R. B. (2008). Revisión del proceso de identificación de Nodos en las Wireless Sensor Networks. *TELEMATIQUE*, 1-23.
- Chávez, C. N. (14 de Octubre de 2009). Evaluación de la tecnología IEEE 802.11n con la plataforma OPNET. Catalunya, España.
- Dargie, W. W. (2010). *Fundamentals of Wireless Sensor Networks : Theory and Practice*. Wiley.
- Domingo, G. M. (2016). *Implementación de un Sistema de Monitoreo y Alerta Temprana para el río Damas de la parroquia Alluriquín*. Santo Domingo.
- Eijkelkamp. (2018). *EIJKELKAMP*. Recuperado el 12 de diciembre de 2017, de www.eijkelkamp.com
- Erazo P, J. A. (2015). Evaluación de una red de Sensores Inalámbrica para detección de Incendios Forestales. *Revista Tecnológica ESPOL – RTE*.
- Fernández M, R. O. (2009). Redes inalámbricas de sensores: teoría y aplicación práctica. *Universidad de La Rioja, Servicio de Publicaciones*, 17-50.
- Foundation, E. (2018). *Eclipse*. Recuperado el 15 de enero de 2018, de www.eclipse.org
- Moradel F, C. A. (2014). *Gestión del Riesgo de Desastres GRD*. Recuperado el 15 de Noviembre de 2017, de <http://www.grdesastres.totalh.net>
- Sacarelo V, G. F. (2014). Diseño y evaluación de una red de sensores para analizar entornos acuáticos. *MASKANA, I+D+ingeniería*.
- Senplades. (2013). *Plan Nacional de Desarrollo / Plan Nacional para el Buen Vivir 2013 - 2017. Versión resumida*. Quito.

- Society, I. C. (2002). *Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)*. The Institute of Electrical and Electronics Engineers, Inc.
- Son, J. R. (2011). Simulation based Functional and Performance Evaluation of Robot Components and Modules. *International Conference on Information Science and Applications*, 1-7.
- Tsáchilas, G. D. (26 de Abril de 2016). *GOBERNACIÓN SANTO DOMINGO*. Recuperado el 14 de Diciembre de 2016, de <http://gubernacionsdtsachilas.gob.ec/alluriquin-en-emergencia-por-desbordamiento-del-rio-damas-la-madrugada-de-hoy/>
- Vega. (2018). *VEGA*. Recuperado el 10 de Diciembre de 2017, de <https://www.vega.com>
- Vera R, C. A., Barbosa J, J. E., & Pabón G, D. (2017). La Tecnología ZigBee estudio de las características de la capa física. *Scientia Et Technica*, 238-245.
- Woolley, M. (2018). *Bluetooth 5 Go Faster. Go Further*. Recuperado el 02 de 05 de 2018, de Bluetooth SIG: <https://www.bluetooth.com/>

ANEXOS

ANEXO A. Información confidencial proporcionada por el GAD Municipal Santo Domingo

Dentro de este Anexo, se encuentra información confidencial proporcionada por el GAD Municipal Santo Domingo de los Tsáchilas y se debe mantener la reserva del caso.

CONTRATO DE SUBASTA INVERSA ELECTRONICA No. 039-2016

Objeto: “IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y ALERTA TEMPRANA PARA EL RÍO DAMAS DE LA PARROQUIA ALLURIQUIN”

Los equipos utilizados en el Sistema de Alerta Temprana por cada estación son los siguientes:

- Caja IP67 con soporte de acero inoxidable.
- Datalogger Vaisala QLM201C
- Tarjeta DSU232 Vaisala
- Módulo QSA224DC Lote: L291
- Tarjeta CF 2GB Transcend
- Modem GSM/GPRS Sierra Wireless LS300
- Modem GSM/GPRS Sierra Wireless Fastrack Extend
- Regulador de Carga Morningstar SUNSAVER SS-10L-12V
- Sensor de Temperatura y Humedad Vaisala HMP155
- Protector de Radiación Solar Vaisala DTR503A
- Panel Solar 100W SL080-12M100
- Batería de ION de Litio 18Ah / 12V
- Sensor de Lluvia Texas TR-525M
- Sensor de Nivel Seba
- UPS XMART SUPRA701
- Cámara IP, mini PTZ tipo ojo de águila TD-9632E2, Zoom 20X, giro 360°
- Cerramiento de 3x3
- Poste de 12 metros
- Soportes para instalación y montaje para equipos.

La ubicación de cada estación sensor se encuentra en la siguiente tabla:

Tabla 1. Ubicación de las estaciones en el SAT Alluriquín. Fuente: GAD Municipal Santo Domingo

ESTACIÓN	DENOMINACION	DISTANCIA DESDE EL CENTRO POBLADO	ALTITUD M.S.N.M.
1	DAMAS_1	200 METROS	761
2	DAMAS_2	4,7 KILOMETROS	809
3	DAMAS_3	8,7 KILOMETROS	1150

Fuente: GALLEGOS, Abel, 2018

Sensores, Registro y Monitoreo constante (cada cinco minutos):

- Nivel del río Damas
- Precipitación (lluvias)
- Temperatura del ambiente
- Humedad en el ambiente

Niveles del caudal del río para el cambio de alerta:

Tabla 2. Nivel de caudal y Alertas en el SAT Alluriquín. Fuente: GAD Municipal Santo Domingo

ESTACIÓN	DENOMINACIÓN	ALERTAS		
		VERDE	AMARILLA	ROJA
1	DAMAS_1	0,00 – 2,49 metros	2,50 – 3,49 metros	>3,50 metros
2	DAMAS_2	0,00 – 2,49 metros	2,50 – 3,49 metros	>3,50 metros
3	DAMAS_3	0,00 – 0,99 metros	1,00 – 1,99 metros	>2,00 metros

Fuente: GALLEGOS, Abel, 2018

Niveles de alerta: Configura tres niveles de alerta:

- VERDE: normalidad en el caudal del río
- AMARILLA: precaución, preparación de la comunidad para una posible evacuación
- ROJA: evacuación inmediata

Capturas de Pantalla del Sistema Actual de Alerta Temprana Acceso vía web.

Tag	Value	Unit	Timestamp	Target	Low-low	Low limit	High limit	High-high
NIVEL DE AGUA	1.39	m	30-abr.-2018 23:59:00	0.50	0.00	0.00	2.50	3.50
ACUMULADO DE LLUVIA	0.00	mm	30-abr.-2018 23:59:00	0.00	0.00	0.00	2.00	3.00
TEMPERATURA DEL AIRE	21.20	C	30-abr.-2018 23:59:00	0.00	0.00	0.00	40.00	50.00
HUMEDAD DEL AIRE	100.00	%	30-abr.-2018 23:59:00	0.00	0.00	0.00	100.00	100.00
VOLT BAT	13.26	V	30-abr.-2018 23:59:00	12.09	11.00	12.00	14.00	15.00

Imagen 1 Datos del Río Damas 1

Fuente: GALLEGOS, Abel, 2018

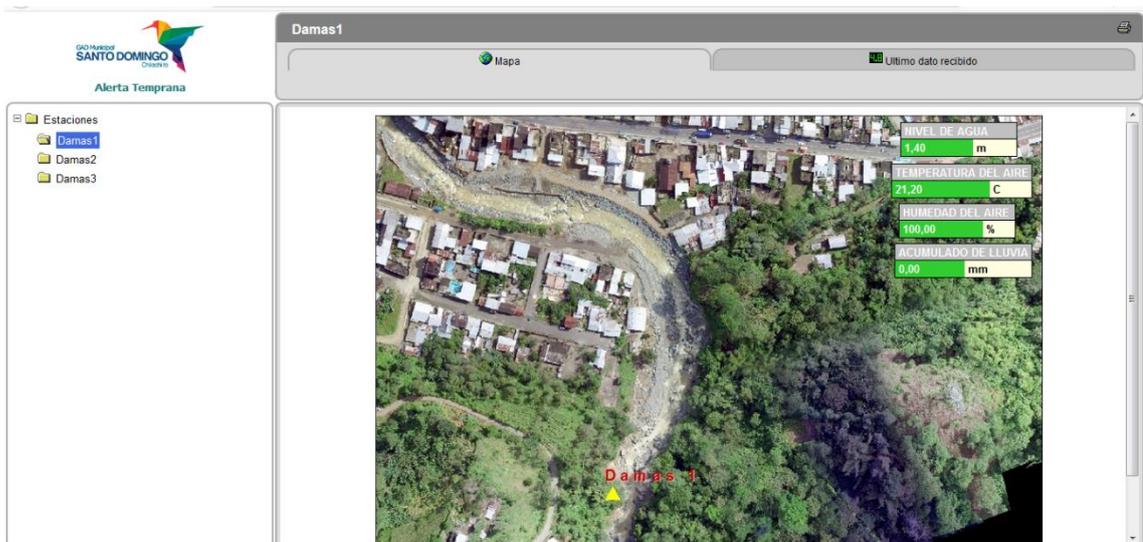


Imagen 2 Mapa del Río Damas 1

Fuente: GALLEGOS, Abel, 2018

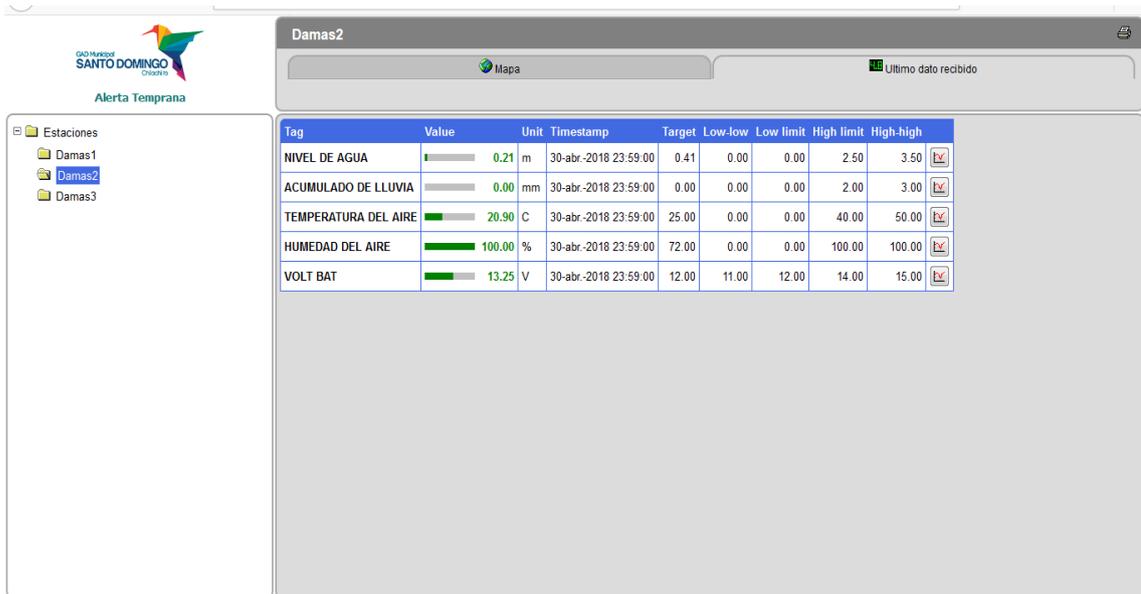


Imagen 3 Datos del Río Damas 2

Fuente: GALLEGOS, Abel, 2018

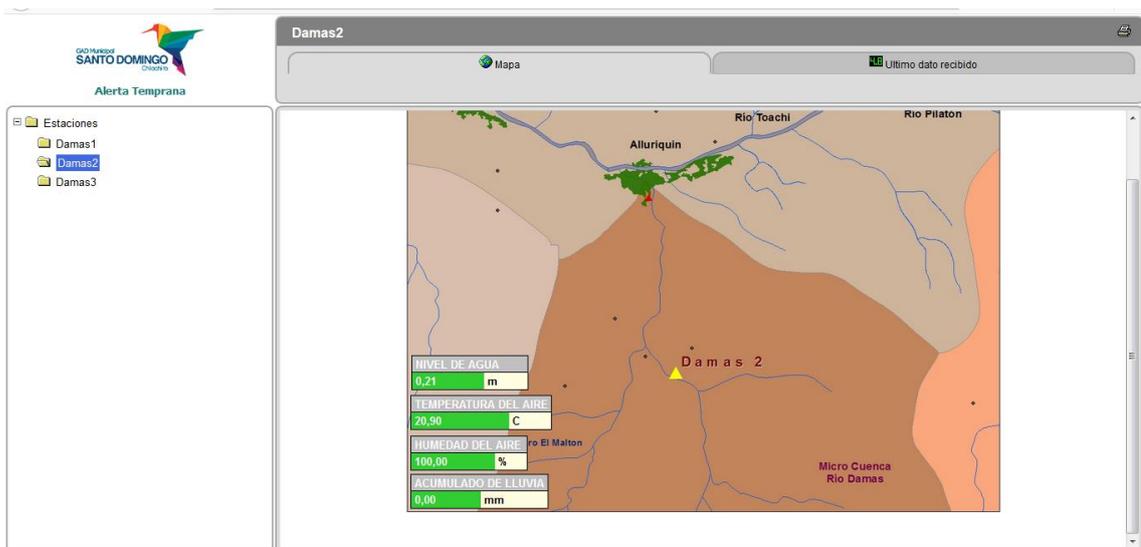


Imagen 4 Mapa del Río Damas 2

Fuente: GALLEGOS, Abel, 2018

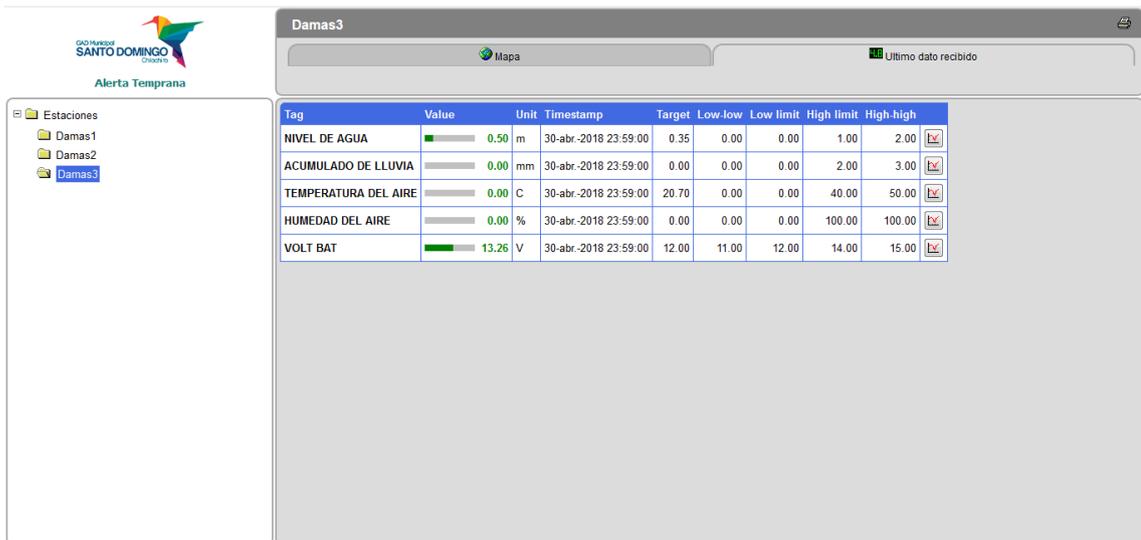


Imagen 5 Datos del Río Damas 3

Fuente: GALLEGOS, Abel, 2018

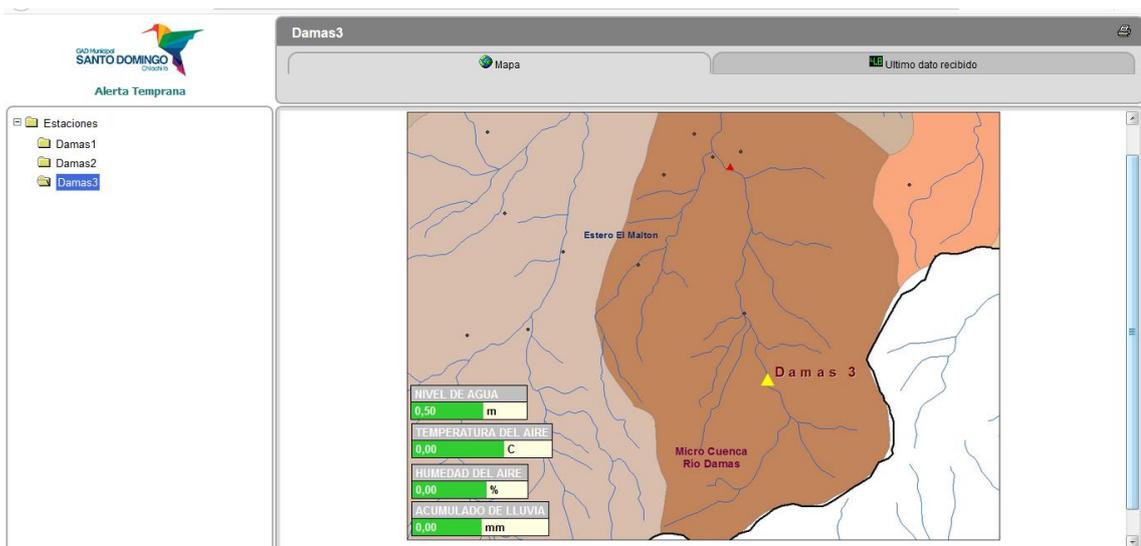


Imagen 6 Mapa del Río Damas 3

Fuente: GALLEGOS, Abel, 2018

ANEXO B. FOTOS DEL RÍO DAMAS



Imagen 7 Río Damas

Fuente: GALLEGOS, Abel, 2018



Imagen 8 Río Damas

Fuente: GALLEGOS, Abel, 2018



Imagen 9 Río Damas

Fuente: GALLEGOS, Abel, 2018



Imagen 10 Verificación de Geo-posicionamiento Río damas

Fuente: GALLEGOS, Abel, 2018

ANEXO C- 802.15.1 BLUETOOTH – PROGRAMACIÓN NS3

ARCHIVO “ble-error-distance-plot.cc”

```
#include <ns3/test.h>
#include <ns3/log.h>
#include <ns3/callback.h>
#include <ns3/packet.h>
#include <ns3/simulator.h>
#include <ns3/ble-error-model.h>
#include <ns3/propagation-loss-model.h>
#include <ns3/ble-net-device.h>
#include <ns3/spectrum-value.h>
#include <ns3/ble-spectrum-value-helper.h>
#include <ns3/ble-mac.h>
#include <ns3/node.h>
#include <ns3/net-device.h>
#include <ns3/single-model-spectrum-channel.h>
#include <ns3/multi-model-spectrum-channel.h>
#include <ns3/mac16-address.h>
#include <ns3/constant-position-mobility-model.h>
#include <ns3/uinteger.h>
#include <ns3/nstime.h>
#include <ns3/abort.h>
#include <ns3/command-line.h>
#include <ns3/gnuplot.h>
#include <fstream>
#include <iostream>
#include <string>
#include <vector>
using namespace ns3;
using namespace std;
static uint32_t g_received = 0;
NS_LOG_COMPONENT_DEFINE ("BleErrorDistancePlot");
static void BleErrorDistanceCallback(McpsDataIndicationParams params,
    Ptr<Packet> p) {
    g_received++;
}
```

```

int main(int argc, char *argv[]) {
    std::ostringstream os;
    std::ofstream berfile("802.15.1-psr-distance.plt");
    int minDistance = 1;
    int maxDistance = 300; // meters
    int increment = 1;
    int maxPackets = 1000;
    int packetSize = 20;
    double txPower = -5; //dBm
    uint32_t channelNumber = 11;
    int N = 200; // numero de nodos
    CommandLine cmd;
    cmd.AddValue("txPower", "transmit power (dBm)", txPower);
    cmd.AddValue("packetSize", "packet (MSDU) size (bytes)", packetSize);
    cmd.AddValue("channelNumber", "channel number", channelNumber);
    cmd.Parse(argc, argv);
    os << "Packet (MSDU) size = " << packetSize << " bytes; tx power = "
        << txPower << " dBm; channel = " << channelNumber;
    Gnuplot psrplot = Gnuplot("802.15.1-psr-distance1.eps");
    Gnuplot2dDataset psrdataset("802.15.1-psr-vs-distance");
    for (int i = 1; i < N; i++) {
        Ptr < Node > n0 = CreateObject<Node>();
        Ptr < Node > n1 = CreateObject<Node>();
        Ptr < BleNetDevice > dev0 = CreateObject<BleNetDevice>();
        Ptr < BleNetDevice > dev1 = CreateObject<BleNetDevice>();
        dev0->SetAddress(Mac16Address("00:01"));
        dev1->SetAddress(Mac16Address("00:02"));
        Ptr < MultiModelSpectrumChannel > channel = CreateObject<
            MultiModelSpectrumChannel>();
        Ptr < LogDistancePropagationLossModel > model = CreateObject<
            LogDistancePropagationLossModel>();
        channel->AddPropagationLossModel(model);
        dev0->SetChannel(channel);
        dev1->SetChannel(channel);
        n0->AddDevice(dev0);
        n1->AddDevice(dev1);
        Ptr < ConstantPositionMobilityModel > mob0 = CreateObject<

```

```

        ConstantPositionMobilityModel>());
dev0->GetPhy()->SetMobility(mob0);
Ptr < ConstantPositionMobilityModel > mob1 = CreateObject<
        ConstantPositionMobilityModel>());
dev1->GetPhy()->SetMobility(mob1);
BleSpectrumValueHelper svh;
Ptr < SpectrumValue > psd = svh.CreateTxPowerSpectralDensity(txPower,
        channelNumber);
dev0->GetPhy()->SetTxPowerSpectralDensity(psd);
McpsDataIndicationCallback cb0;
cb0 = MakeCallback(&BleErrorDistanceCallback);
dev1->GetMac()->SetMcpsDataIndicationCallback(cb0);
McpsDataRequestParams params;
params.m_srcAddrMode = SHORT_ADDR;
params.m_dstAddrMode = SHORT_ADDR;
params.m_dstPanId = 0;
params.m_dstAddr = Mac16Address("ff:ff");
params.m_msduHandle = 0;
params.m_txOptions = 0;
Ptr < Packet > p;
mob0->SetPosition(Vector(0, 0, 0));
mob1->SetPosition(Vector(minDistance, 0, 0));
for (int j = minDistance; j < maxDistance;) {
    for (int i = 0; i < maxPackets; i++) {
        p = Create < Packet > (packetSize);
        Simulator::Schedule(Seconds(i), &BleMac::McpsDataRequest,
                dev0->GetMac(), params, p);
    }
    Simulator::Run();
    NS_LOG_DEBUG(
        "Received " << g_received << " packets for distance "
<< j);

    psrdataset.Add(j, g_received / 1000.0);
    g_received = 0;
    j += increment;
    mob1->SetPosition(Vector(j, 0, 0));
}

```

```
        psrplot.AddDataset(psrdataset);
    }
    psrplot.SetTitle(os.str());
    psrplot.SetTerminal("postscript eps color enh \"Times-BoldItalic\");
    psrplot.SetLegend("distance (m)", "Packet Success Rate (PSR)");
    psrplot.SetExtra(
        "set xrange [0:200]\n\
set yrange [0:1]\n\
set grid\n\
set style line 1 linewidth 5\n\
set style increment user");
    psrplot.GenerateOutput(berfile);
    berfile.close();
    Simulator::Destroy();
    return 0;
}
```

ANEXO D - 802.11n WIFI- PROGRAMACIÓN NS3

ARCHIVO "myhelperwifi.cpp"

```
/*
 * myhelperwifi.cpp
 *
 * Created on: 11 abr. 2018
 * Author: manager
 */
#include "myhelperwifi.h"
#include "myLog.cpp"
double totalenergia=0;
void RemainingEnergy(double oldValue, double remainingEnergy) {
    NS_LOG_UNCOND(Simulator::Now ().GetSeconds () << "s Current remaining energy
= " << remainingEnergy << "J");
}
/// Trace function for total energy consumption at node.
void TotalEnergy(double oldValue, double totalEnergy) {
    NS_LOG_UNCOND(Simulator::Now ().GetSeconds () << "s Total energy consumed
by radio = " << totalEnergy << "J");
    totalenergia= totalEnergy;
    //std::cout << "ENERGIA PARCIAL" << totalEnergy << std::endl;
}
myhelperwifi::myhelperwifi() {
    // TODO Auto-generated constructor stub
}
myhelperwifi::~myhelperwifi() {
    // TODO Auto-generated destructor stub
}
static int EnviadosWifi = 0;//
static int RecibidosWifi = 0;
static void GenerateTraffic(Ptr<Socket> socket, uint32_t pktSize,
uint32_t pktCount, Time pktInterval) {
if (pktCount > 0) {
socket->Send(Create<Packet>(pktSize));
Simulator::Schedule(pktInterval, &GenerateTraffic, socket, pktSize,
pktCount - 1, pktInterval);
```

```

EnviadosWifi++;
std::cout << "GENERA TRAFICO : " << EnviadosWifi << std::endl;
} else {
socket->Close();
}
}

void ReceivePacket(Ptr<Socket> socket) {
while (socket->Recv()) {
NS_LOG_UNCOND("Received one packet!");
RecibidosWifi++;
std::cout << "RECIBE TRAFICO " << RecibidosWifi << std::endl;
}
}

NodeContainer myhelperwifi::Build(int nodos, int distancia, int perdida,
double simulationtime) {
myLog log;
//std::string phyMode("DsssRate1Mbps");
std::vector <std::string> modes;
modes.push_back ("HtMcs0");
//double distance = 500; // m
//uint32_t packetSize = 1000; // bytes
//uint32_t numPackets = 10;
Config::SetDefault("ns3::TcpSocket::SegmentSize",
UIntegerValue(payloadSize));
//uint32_t numNodes = 25; // by default, 5x5
//uint32_t sinkNode = 0;
//uint32_t sourceNode = 24;
double interval = 1.0; // seconds
bool verbose = true;
bool tracing = true;
// Convert to time object
Time interPacketInterval = Seconds(interval);
// disable fragmentation for frames below 2200 bytes
Config::SetDefault("ns3::WifiRemoteStationManager::FragmentationThreshold",
StringValue("2200"));
// turn off RTS/CTS for frames below 2200 bytes
Config::SetDefault("ns3::WifiRemoteStationManager::RtsCtsThreshold",

```

```

        StringValue("2200"));
// Fix non-unicast data rate to be the same as that of unicast
Config::SetDefault("ns3::WifiRemoteStationManager::NonUnicastMode",
        StringValue("HtMcs0"));
log.say("WIFI", "Creando nodos");
equiposwifi.Create(nodos);
log.say("WIFI", "Creando wifi");
// The below set of helpers will help us to put together the wifi NICs we want
WifiHelper wifi;
if (verbose) {
        wifi.EnableLogComponents(); // Turn on all Wifi logging
}

YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default();
// set it to zero; otherwise, gain will be added
wifiPhy.Set("RxGain", DoubleValue(-10));
// ns-3 supports RadioTap and Prism tracing extensions for 802.11b
wifiPhy.SetPcapDataLinkType(YansWifiPhyHelper::DLT_IEEE802_11_RADIO);
YansWifiChannelHelper wifiChannel;
wifiChannel.SetPropagationDelay("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss("ns3::FriisPropagationLossModel");
wifiPhy.SetChannel(wifiChannel.Create());
// Add an upper mac and disable rate control
WifiMacHelper wifiMac;
wifi.SetStandard(WIFI_PHY_STANDARD_80211n_2_4GHZ);
Config::SetDefault ("ns3::LogDistancePropagationLossModel::ReferenceLoss",
DoubleValue (40.046));
wifi.SetRemoteStationManager("ns3::ConstantRateWifiManager", "DataMode",
        StringValue("HtMcs0"), "ControlMode", StringValue("HtMcs0"));
// Set it to adhoc mode
wifiMac.SetType("ns3::AdhocWifiMac");
NetDeviceContainer redequiposwifi = wifi.Install(wifiPhy, wifiMac,
        equiposwifi);
log.say("WIFI", "Creando movil");
MobilityHelper mobility;
mobility.SetPositionAllocator("ns3::GridPositionAllocator", "MinX",
        DoubleValue(0.0), "MinY", DoubleValue(0.0), "DeltaX",

```

```

        DoubleValue(0), "DeltaY", DoubleValue(distancia), "GridWidth",
        UIntegerValue(50), "LayoutType", StringValue("RowFirst"));
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.Install(equiposwifi);
log.say("WIFI", "Creando OLSR");
// Enable OLSR
OlsrHelper olsr;
Ipv4StaticRoutingHelper staticRouting;
log.say("WIFI", "Creando internet/ipv4");
Ipv4ListRoutingHelper list;
list.Add(staticRouting, 0);
list.Add(olsr, 10);
log.say("WIFI", "MODULO ENERGIA");
/** Energy Model */
/*****
*****/
/* energy source */
BasicEnergySourceHelper basicSourceHelper;
// configure energy source
basicSourceHelper.Set("BasicEnergySourceInitialEnergyJ", DoubleValue(3));
// install source
EnergySourceContainer sources = basicSourceHelper.Install(equiposwifi);
/* device energy model */
WifiRadioEnergyModelHelper radioEnergyHelper;
// configure radio energy model
radioEnergyHelper.Set("TxCurrentA", DoubleValue(0.0174));
radioEnergyHelper.Set("RxCurrentA", DoubleValue(0.0197));
// install device model
DeviceEnergyModelContainer deviceModels = radioEnergyHelper.Install(
        redequiposwifi, sources);
/*****
*****/

InternetStackHelper internet;
internet.SetRoutingHelper(list); // has effect on the next Install ()
internet.Install(equiposwifi);

```

```

Ipv4AddressHelper ipv4;
NS_LOG_INFO("Assign IP Addresses.");
ipv4.SetBase("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer i = ipv4.Assign(redequipoWifi);
log.say("WIFI", "Creando server");
uint16_t port = 9;
UdpServerHelper server(port);
/** connect trace sources **/
/*****
*****/
// all sources are connected to node 1
// energy source
Ptr<BasicEnergySource> basicSourcePtr = DynamicCast<BasicEnergySource>(
    sources.Get(0));
basicSourcePtr->TraceConnectWithoutContext("RemainingEnergy",
    MakeCallback(&RemainingEnergy));
Ptr<DeviceEnergyModel> basicRadioModelPtr =
    basicSourcePtr-
>FindDeviceEnergyModels("ns3::WifiRadioEnergyModel").Get(
    0);
NS_ASSERT(basicRadioModelPtr != NULL);
basicRadioModelPtr->TraceConnectWithoutContext("TotalEnergyConsumption",
    MakeCallback(&TotalEnergy));
serverApp = server.Install(equipoWifi.Get(0));
serverApp.Start(Seconds(0.0));
serverApp.Stop(Seconds(simulationTime + 1));
for (int j = 1; j < nodos; j++) {
    UdpClientHelper client(i.GetAddress(0), port);
    client.SetAttribute("MaxPackets", UintegerValue(4294967295u));
    client.SetAttribute("Interval", TimeValue(Time("0.001"))); //packets/s
    client.SetAttribute("PacketSize", UintegerValue(payloadSize));
    ApplicationContainer clientApp = client.Install(equipoWifi.Get(j));
    clientApp.Start(Seconds(1.0));
    clientApp.Stop(Seconds(simulationTime + 1));
}
if (tracing == true) {
    log.say("WIFI", "Creando trazas");
}

```

```

        AsciiTraceHelper ascii;
        wifiPhy.EnableAsciiAll(
            ascii.CreateFileStream("wifi-simple-adhoc-grid.tr"));
        wifiPhy.EnablePcap("wifi-simple-adhoc-grid", redequipoSwifi);
        // Trace routing tables
        Ptr<OutputStreamWrapper> routingStream = Create<OutputStreamWrapper>(
            "wifi-simple-adhoc-grid.routes", std::ios::out);
        olsr.PrintRoutingTableAllEvery(Seconds(2), routingStream);
        Ptr<OutputStreamWrapper> neighborStream =
Create<OutputStreamWrapper>(
            "wifi-simple-adhoc-grid.neighbors", std::ios::out);
        olsr.PrintNeighborCacheAllEvery(Seconds(2), neighborStream);
        // To do-- enable an IP-level trace that shows forwarding events only
    }
//Simulator::Schedule(Seconds(5.0), &GenerateTraffic, source, packetSize,numPackets,
interPacketInterval);
    // Give OLSR time to converge-- 30 seconds perhaps
    return equipoSwifi;
}
void myhelperwifi::Report(double simulationtime, int distancia, int numnodos) {
    uint64_t totalPacketsThrough =
        DynamicCast<UdpServer>(serverApp.Get(0))->GetReceived();
    double throughput = totalPacketsThrough * payloadSize * 8
        / (simulationtime * 1000000); //Mbit/s
    std::cout << "TOTAL PAQUETES RECIBIDOS: " << totalPacketsThrough
        << std::endl;
    std::cout << "TOTAL THROUGHPUT: " << throughput << " Mbps " << std::endl;
    std::cout << "TOTAL ENERGIA CONSUMIDA: " << totalenergia << " Joules " <<
std::endl;
    std::cout << "DISTANCIA ENTRE NODOS: " << distancia/2 << " metros " <<
std::endl;
    std::cout << "NUMERO DE NODOS: " << numnodos << std::endl;
}

```

ARCHIVO "myhelperwifi.h"

```
/*
 * myhelperwifi.h
 *
 * Created on: 11 abr. 2018
 * Author: manager
 */
#ifndef SCRATCH_MYHELPERWIFI_H_
#define SCRATCH_MYHELPERWIFI_H_
#include "ns3/core-module.h"
#include "ns3/applications-module.h"
#include "ns3/wifi-module.h"
#include "ns3/mobility-module.h"
#include "ns3/internet-module.h"
#include "ns3/gnuplot.h"
#include "ns3/olsr-helper.h"
#include "ns3/gnuplot.h"
#include "ns3/netanim-module.h"
#include "myLog.h"
#include "ns3/energy-module.h"
using namespace ns3;
class myhelperwifi {
private :
    NodeContainer equiposwifi;
    //int NumNodosWifi=0;
    //int DistanciaNodosWifi=0;
    ApplicationContainer serverApp;
    uint32_t payloadSize = 1472; //bytes
public:
    void Report(double simulationtime, int distancia, int numnodos);
    myhelperwifi();
    virtual ~myhelperwifi();
    NodeContainer Build(int nodos, int distancia, int perdida,double simulationtime);
};
#endif /* SCRATCH_MYHELPERWIFI_H_ */
```

ARCHIVO "Log.cpp"

```
/*
 * Log.cpp
 *
 * Created on: 12 abr. 2018
 * Author: manager
 */
#include "myLog.h"
#include <iostream>
using namespace ns3;
myLog::myLog() {
    // TODO Auto-generated constructor stub
}
myLog::~myLog() {
    // TODO Auto-generated destructor stub
}
void myLog::say(std::string Tag, std::string Message) {
std::cout << "TAG : " << Tag << " MESSAGE : " << Message << std::endl;
}
}
```

ARCHIVO "Log.h"

```
/*
 * Log.h
 *
 * Created on: 12 abr. 2018
 * Author: manager
 */
#ifndef SCRATCH_MYLOG_H_
#define SCRATCH_MYLOG_H_
#include <string>
using namespace ns3;
class myLog {
public:
    myLog();
    virtual ~myLog();
    void say(std::string Tag, std::string Message); };
#endif /* SCRATCH_MYLOG_H_ */
```

ANEXO E - 802.15.4 ZIGBEE - PROGRAMACIÓN NS3

ARCHIVO "lr-wpan-error-distance-plot.cc"

```
#include <ns3/test.h>
#include <ns3/log.h>
#include <ns3/callback.h>
#include <ns3/packet.h>
#include <ns3/simulator.h>
#include <ns3/lr-wpan-error-model.h>
#include <ns3/propagation-loss-model.h>
#include <ns3/lr-wpan-net-device.h>
#include <ns3/spectrum-value.h>
#include <ns3/lr-wpan-spectrum-value-helper.h>
#include <ns3/lr-wpan-mac.h>
#include <ns3/node.h>
#include <ns3/net-device.h>
#include <ns3/single-model-spectrum-channel.h>
#include <ns3/multi-model-spectrum-channel.h>
#include <ns3/mac16-address.h>
#include <ns3/constant-position-mobility-model.h>
#include <ns3/uinteger.h>
#include <ns3/nstime.h>
#include <ns3/abort.h>
#include <ns3/command-line.h>
#include <ns3/gnuplot.h>
#include <fstream>
#include <iostream>
#include <string>
#include <vector>
using namespace ns3;
using namespace std;
static uint32_t g_received = 0;
NS_LOG_COMPONENT_DEFINE ("LrWpanErrorDistancePlot");
static void LrWpanErrorDistanceCallback(McpsDataIndicationParams params,
    Ptr<Packet> p) {
    g_received++;
```

```

}
int main(int argc, char *argv[]) {
    std::ostringstream os;
    std::ofstream berfile("802.15.4-psr-distanceLS.plt");
    int minDistance = 1;
    int maxDistance = 200; // distancia maxima
    int increment = 1;
    int maxPackets = 1000;
    int packetSize = 20; //Bytes
    double txPower = -5; //dBm decibelios milli
    uint32_t channelNumber = 11;
    int N = 200;// numero de nodos
    CommandLine cmd;
    cmd.AddValue("txPower", "transmit power (dBm)", txPower);
    cmd.AddValue("packetSize", "packet (MSDU) size (bytes)", packetSize);
    cmd.AddValue("channelNumber", "channel number", channelNumber);
    cmd.Parse(argc, argv);
    os << "Packet (MSDU) size = " << packetSize << " bytes; tx power = "
        << txPower << " dBm; channel = " << channelNumber;
    Gnuplot psrplot = Gnuplot("802.15.4-psr-distanceLS.eps");
    Gnuplot2dDataset psrdataset("802.15.4-psr-vs-distanceLS");
    for (int i = 1; i < N; i++) {
        Ptr < Node > n0 = CreateObject<Node>();
        Ptr < Node > n1 = CreateObject<Node>();
        Ptr < LrWpanNetDevice > dev0 = CreateObject<LrWpanNetDevice>();
        Ptr < LrWpanNetDevice > dev1 = CreateObject<LrWpanNetDevice>();
        dev0->SetAddress(Mac16Address("00:01"));
        dev1->SetAddress(Mac16Address("00:02"));
        Ptr < MultiModelSpectrumChannel > channel = CreateObject<
            MultiModelSpectrumChannel>();
        Ptr < LogDistancePropagationLossModel > model = CreateObject<
            LogDistancePropagationLossModel>();
        channel->AddPropagationLossModel(model);
        dev0->SetChannel(channel);
        dev1->SetChannel(channel);
        n0->AddDevice(dev0);
        n1->AddDevice(dev1);
    }
}

```

```

Ptr < ConstantPositionMobilityModel > mob0 = CreateObject<
    ConstantPositionMobilityModel>();
dev0->GetPhy()->SetMobility(mob0);
Ptr < ConstantPositionMobilityModel > mob1 = CreateObject<
    ConstantPositionMobilityModel>();
dev1->GetPhy()->SetMobility(mob1);
LrWpanSpectrumValueHelper svh;
Ptr < SpectrumValue > psd = svh.CreateTxPowerSpectralDensity(txPower,
    channelNumber);
dev0->GetPhy()->SetTxPowerSpectralDensity(psd);
McpsDataIndicationCallback cb0;
cb0 = MakeCallback(&LrWpanErrorDistanceCallback);
dev1->GetMac()->SetMcpsDataIndicationCallback(cb0);
McpsDataRequestParams params;
params.m_srcAddrMode = SHORT_ADDR;
params.m_dstAddrMode = SHORT_ADDR;
params.m_dstPanId = 0;
params.m_dstAddr = Mac16Address("ff:ff");
params.m_msduHandle = 0;
params.m_txOptions = 0;
Ptr < Packet > p;
mob0->SetPosition(Vector(0, 0, 0));
mob1->SetPosition(Vector(minDistance, 0, 0));
for (int j = minDistance; j < maxDistance;) {
    for (int i = 0; i < maxPackets; i++) {
        p = Create < Packet > (packetSize);
        Simulator::Schedule(Seconds(i),
&LrWpanMac::McpsDataRequest,
                                dev0->GetMac(), params, p);
    }
    Simulator::Run();
    NS_LOG_DEBUG(
        "Received " << g_received << " packets for distance "
<< j);
    psrdataset.Add(j, g_received / 1000.0);
    g_received = 0;
    j += increment;
}

```

```
        mob1->SetPosition(Vector(j, 0, 0));
    }
    psrplot.AddDataset(psrdataset);
}
psrplot.SetTitle(os.str());
psrplot.SetTerminal("postscript eps color enh \"Times-BoldItalic\");
psrplot.SetLegend("distance (m)", "Packet Success Rate (PSR)");
psrplot.SetExtra(
    "set xrange [0:200]\n\
set yrange [0:1]\n\
set grid\n\
set style line 1 linewidth 5\n\
set style increment user");
    psrplot.GenerateOutput(berfile);
    berfile.close();
    Simulator::Destroy();
    return 0;
}
```

ANEXO F– ESCENARIOS PROGRAMACIÓN NS3

ARCHIVO “AG3.cc”

```
//AG3
```

```
// Created on: 12 jul. 2018
```

```
// Author: manager
```

```
#include "ns3/core-module.h"
```

```
#include "ns3/applications-module.h"
```

```
#include "ns3/mobility-module.h"
```

```
#include "ns3/wifi-module.h"
```

```
#include "ns3/internet-module.h"
```

```
#include "ns3/network-module.h"
```

```
#include "ns3/olsr-helper.h"
```

```
#include "ns3/netanim-module.h"
```

```
#include "ns3/olsr-helper.h"
```

```
#include "ns3/flow-monitor-module.h"
```

```
using namespace ns3;
```

```
NS_LOG_COMPONENT_DEFINE("WifiSimpleAdhocGrid");
```

```
void ReceivePacket(Ptr<Socket> socket) {
```

```
    while (socket->Recv()) {
```

```
        NS_LOG_UNCOND("Received one packet!");
```

```
    }
```

```
}
```

```
static void GenerateTraffic(Ptr<Socket> socket, uint32_t pktSize,
```

```
    uint32_t pktCount, Time pktInterval) {
```

```
    if (pktCount > 0) {
```

```
        socket->Send(Create < Packet > (pktSize));
```

```
        Simulator::Schedule(pktInterval, &GenerateTraffic, socket, pktSize,  
            pktCount - 1, pktInterval);
```

```
    } else {
```

```
        socket->Close();
```

```
    }
```

```
}
```

```
int main(int argc, char *argv[]) {
```

```
    //std::string phyMode ("DsssRate1Mbps");
```

```
    std::vector < std::string > modes;
```

```

modes.push_back("HtMcs0");
double distance = 200; // m
uint32_t packetSize = 1000; // bytes
uint32_t numPackets = 1;
int numNodes = 15; // by default, 5x5
uint32_t sinkNode = 0;
uint32_t sourceNode = 14;
double interval = 1.0; // seconds
bool verbose = true;
bool tracing = true;
double distanciatotal = sourceNode * distance;
CommandLine cmd;
//cmd.AddValue("phyMode", "Wifi Phy mode", phyMode);
cmd.AddValue("distance", "distance (m)", distance);
//cmd.AddValue("packetSize", "size of application packet sent", packetSize);
//cmd.AddValue("numPackets", "number of packets generated", numPackets);
//cmd.AddValue("interval", "interval (seconds) between packets", interval);
//cmd.AddValue("verbose", "turn on all WifiNetDevice log components", verbose);
cmd.AddValue("tracing", "turn on ascii and pcap tracing", tracing);
cmd.AddValue("numNodes", "number of nodes", numNodes);
cmd.AddValue("sinkNode", "Receiver node number", sinkNode);
cmd.AddValue("sourceNode", "Sender node number", sourceNode);
cmd.Parse(argc, argv);
// Convert to time object
Time interPacketInterval = Seconds(interval);
// disable fragmentation for frames below 2200 bytes
Config::SetDefault("ns3::WifiRemoteStationManager::FragmentationThreshold",
StringValue("2200"));
// turn off RTS/CTS for frames below 2200 bytes
Config::SetDefault("ns3::WifiRemoteStationManager::RtsCtsThreshold",
StringValue("2200"));
// Fix non-unicast data rate to be the same as that of unicast
Config::SetDefault("ns3::WifiRemoteStationManager::NonUnicastMode",
StringValue("HtMcs0"));
NodeContainer c;
c.Create(numNodes);
// The below set of helpers will help us to put together the wifi NICs we want

```

```

WifiHelper wifi;
if (verbose) {
    wifi.EnableLogComponents(); // Turn on all Wifi logging
}
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default();
// set it to zero; otherwise, gain will be added
wifiPhy.Set("RxGain", DoubleValue(-10));
// ns-3 supports RadioTap and Prism tracing extensions for 802.11b
wifiPhy.SetPcapDataLinkType(YansWifiPhyHelper::DLT_IEEE802_11_RADIO);
YansWifiChannelHelper wifiChannel;
wifiChannel.SetPropagationDelay("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss("ns3::FriisPropagationLossModel");
wifiPhy.SetChannel(wifiChannel.Create());
// Add an upper mac and disable rate control
WifiMacHelper wifiMac;
wifi.SetStandard(WIFI_PHY_STANDARD_80211n_2_4GHZ);
Config::SetDefault("ns3::LogDistancePropagationLossModel::ReferenceLoss",
    DoubleValue(40.046));
wifi.SetRemoteStationManager("ns3::ConstantRateWifiManager", "DataMode",
    StringValue("HtMcs0"), "ControlMode", StringValue("HtMcs0"));
// Set it to adhoc mode
wifiMac.SetType("ns3::AdhocWifiMac");
NetDeviceContainer devices = wifi.Install(wifiPhy, wifiMac, c);
MobilityHelper mobility;
mobility.SetPositionAllocator("ns3::GridPositionAllocator", "MinX",
    DoubleValue(0.0), "MinY", DoubleValue(0.0), "DeltaX",
    DoubleValue(distance), "DeltaY", DoubleValue(0.0), "GridWidth",
    UIntegerValue(5), "LayoutType", StringValue("ColumnFirst"));
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.Install(c);
// Enable OLSR
OlsrHelper olsr;
Ipv4StaticRoutingHelper staticRouting;
Ipv4ListRoutingHelper list;
list.Add(staticRouting, 0);
list.Add(olsr, 10);
InternetStackHelper internet;

```

```

internet.SetRoutingHelper(list); // has effect on the next Install ()
internet.Install(c);
Ipv4AddressHelper ipv4;
NS_LOG_INFO("Assign IP Addresses.");
ipv4.SetBase("192.168.1.0", "255.255.255.0");
Ipv4InterfaceContainer i = ipv4.Assign(devices);
TypeId tid = TypeId::LookupByName("ns3::UdpSocketFactory");
Ptr < Socket > recvSink = Socket::CreateSocket(c.Get(sinkNode), tid);
InetSocketAddress local = InetSocketAddress(Ipv4Address::GetAny(), 80);
recvSink->Bind(local);
recvSink->SetRecvCallback(MakeCallback(&ReceivePacket));
Ptr < Socket > source = Socket::CreateSocket(c.Get(sourceNode), tid);
InetSocketAddress remote = InetSocketAddress(i.GetAddress(sinkNode, 0), 80);
source->Connect(remote);
if (tracing == true) {
    AsciiTraceHelper ascii;
    wifiPhy.EnableAsciiAll(
        ascii.CreateFileStream("wifi-simple-adhoc-grid.tr"));
    wifiPhy.EnablePcap("wifi-simple-adhoc-grid", devices);
    // Trace routing tables
    Ptr < OutputStreamWrapper > routingStream = Create < OutputStreamWrapper
        > ("wifi-simple-adhoc-grid.routes", std::ios::out);
    olsr.PrintRoutingTableAllEvery(Seconds(2), routingStream);
    Ptr < OutputStreamWrapper > neighborStream = Create
        < OutputStreamWrapper
        > ("wifi-simple-adhoc-grid.neighbors", std::ios::out);
    olsr.PrintNeighborCacheAllEvery(Seconds(2), neighborStream);
    // To do-- enable an IP-level trace that shows forwarding events only
}
// Give OLSR time to converge-- 30 seconds perhaps
Simulator::Schedule(Seconds(30.0), &GenerateTraffic, source, packetSize,
    numPackets, interPacketInterval);
AnimationInterface anim("wifi2.xml");
for (int i = 0; i < numNodes; i++) {
    anim.SetConstantPosition(c.Get(i), i * distance, i * 0, 0.0);
    anim.EnablePacketMetadata(true);
    anim.SetMobilityPollInterval(Seconds(1));
}

```

```
}  
// Output what we are doing  
NS_LOG_UNCOND(  
    "Testing from node " << sourceNode << " to " << sinkNode  
    << " with grid distance " << distance);  
Simulator::Stop(Seconds(33.0));  
Simulator::Run();  
std::cout << "Comunicacion entre nodo final " << sourceNode  
    << " hacia nodo raiz " << sinkNode << " distancia entre nodos "  
    << distance << " distancia total " << distanciatotal << " metros "  
<<std::endl;  
    Simulator::Destroy();  
    return 0;  
}
```