



UNIVERSIDAD NACIONAL DE LOJA

**FACULTAD DE LA ENERGÍA LAS INDUSTRIAS Y LOS
RECURSOS NATURALES NO RENOVABLES**

**CARRERA DE ELECTRÓNICA Y
TELECOMUNICACIONES**

**ESTUDIO COMPARATIVO DE LAS MÉTRICAS DE CALIDAD DE
SERVICIO (QoS) EN LOS PROTOCOLOS DE ENCAMINAMIENTO
REACTIVOS DSR Y AODV EN REDES MANET**

***TESIS PREVIA A OPTAR POR EL TITULO
DE INGENIERO EN ELECTRÓNICA Y
TELECOMUNICACIONES***

AUTOR:

× **Diego Andrés Sánchez Carrión.**

DIRECTOR:

× **Ing. John Jossimar Tucker Yépez, Mg. Sc.**

Loja – Ecuador

2019

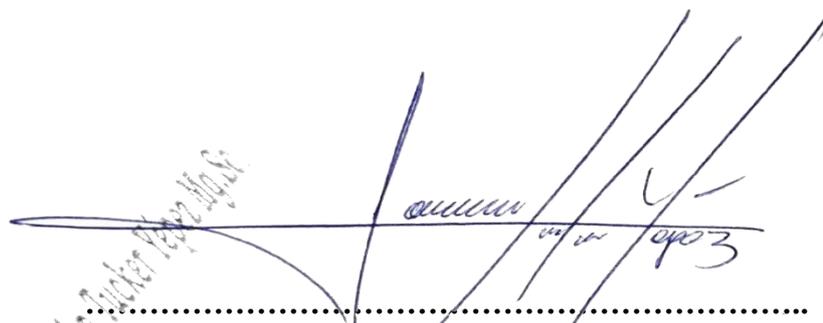
CERTIFICACIÓN

Ing. John Jossimar Tucker Yépez, Mg. Sc.
DIRECTOR DE TESIS

CERTIFICA:

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación cuyo tema versa en **“ESTUDIO COMPARATIVO DE LAS MÉTRICAS DE CALIDAD DE SERVICIO (QoS) EN LOS PROTOCOLOS DE ENCAMINAMIENTO REACTIVOS DSR Y AODV EN REDES MANET”**, previa a la obtención del título de **Ingeniero en Electrónica y Telecomunicaciones**, realizado por el señor: **Diego Andrés Sánchez Carrión** la misma que cumple con la reglamentación y políticas de investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

Loja, 28 de junio 2019



Ing. John Jossimar Tucker Yépez Mg. Sc.
DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN

AUTORÍA

Yo **DIEGO ANDRÉS SÁNCHEZ CARRIÓN**, declaro ser el autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional – Biblioteca Virtual.

Firma: -----

Cedula: 1150354031

Fecha: 12 de julio de 2019

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

DIEGO ANDRES SANCHEZ CARRION, declaro ser autor de la tesis titulada: “**ESTUDIO COMPARATIVO DE LAS MÉTRICAS DE CALIDAD DE SERVICIO (QoS) EN LOS PROTOCOLOS DE ENCAMINAMIENTO REACTIVOS DSR Y AODV EN REDES MANET**”, como requisito para optar al grado de: **INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los doce días del mes de julio del dos mil diecinueve

Firma: -----



Autor: Diego Andrés Sánchez Carrión

Cédula: 1150354031

Dirección: Loja,

Correo Electrónico: dasanchezc@unl.edu.ec

Teléfono: 2574086

Celular: 0979415075

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. John Jossimar Tucker Yépez. Mg. Sc.

Tribunal de Grado: Ing. Andy Fabricio Vega León. Mg. Sc.

Ing. Franklin Gustavo Jiménez Peralta. Mg. Sc.

Ing. Pedro Andrés Delgado Guerrero. Mg. Sc.

DEDICATORIA

Dedico este trabajo primeramente a Dios por guiarme y brindarme la oportunidad de llegar a este punto de culminación de mis estudios universitarios y lograrlo de la mejor manera.

Con todo cariño lo dedico a mi padre Mario Sánchez y madre Livia Carrión quienes me han ayudado a ser todo lo que soy, inculcándome buenos valores y principios que me han servido para cumplir cada una de mis metas.

A mi Hermana Mónica, mi hermano Fernando, mi sobrina Megan y demás familiares que son mi inspiración para día a día ir superándome y me han brindado todo su cariño siempre.

Mi dedicatoria también a mis Amigos, y a excompañeros de la universidad y colegio por brindarme su amistad, apoyo y ser parte fundamental en toda mi formación.

AGRADECIMIENTO

Agradezco primeramente a Dios por regalarme la vida, una maravillosa Familia, y por todas las formas en que me ha bendecido en cada momento de mi vida, por ser la luz y guía en cada situación del camino recorrido en esta carrera.

A mis padres que me se han esforzado diariamente para brindarme los estudios y además me han brindado sus enseñanzas, y me guían siempre alentándome a brindar lo mejor de mí para cumplir todas y cada una de mis metas, en esta ocasión mi carrera profesional.

A mi hermana y hermano, por su apoyo incondicional, en todos los aspectos de mi vida especialmente en estos años de vida universitaria.

Al ing. John Tucker por su experiencia, asesoría y apoyo, a través de lo cual pude concretar mi proyecto de titulación y de la misma manera el proceso de titulación.

A mis compañeros y amigos con quienes compartí varias vivencias que dejaron bastas experiencias que me ayudaron a llegar a este momento de la finalización de mi carrera.

A la planta docente que a partir de los conocimientos y experiencias impartidos durante años de vida universitaria me brindaron enseñanzas académicas, valores y experiencias de vida, y que gracias a esto hoy puedo concretar con profesionalismo mi proyecto de titulación.

TABLA DE CONTENIDO

CERTIFICACIÓN.....	II
AUTORÍA	III
CARTA DE AUTORIZACIÓN.....	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
TABLA DE CONTENIDO	VII
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS.....	XII
ACRÓNIMOS.....	XIII
1. TÍTULO.....	1
2. RESUMEN	2
ABSTRACT	3
3. INTRODUCCIÓN.....	4
4. REVISIÓN LITERARIA.....	5
4.1. REDES INALÁMBRICAS.....	5
4.2. Estándar IEEE 802.11x	6
4.3. Redes Ad Hoc.....	8
4.4. Redes Móviles Ad Hoc (MANET)	9
4.5. Aplicaciones de MANET	13
4.6. Protocolos de encaminamiento en redes convencionales	15
4.6.1. Vector Distancia.....	15
4.6.2. Estado de enlace	15
4.7. Protocolos de encaminamiento para redes ad hoc	16
4.7.1. Proactivos	16
4.7.2. Reactivos	16
4.7.3. Híbridos.....	16
4.8. Métricas de Calidad de Servicio QoS.....	17
4.8.1. Tasa de Paquetes perdidos (PLR).....	17
4.8.2. Retardo (Delay).....	17
4.8.3. Jitter (Variación del Retardo de Paquetes)	17
4.8.4. Consumo de Energía	18
4.8.5. Ancho de Banda	18

4.8.6.	Rendimiento	18
4.8.7.	Throughput	18
4.9.	Protocolos de Autoconfiguración	18
4.10.	Modelos de Movilidad.....	19
4.11.	Modelo Random Waypoint.....	19
4.12.	Protocolo AODV (Ad hoc On demand Distance Vector)	19
4.13.	Protocolo DSR (Dynamic Source Routing)	21
5.	MATERIALES Y MÉTODOS.....	23
5.1.	Materiales	23
5.2.	Método Cualitativo	23
5.3.	Modelo de Movilidad	24
5.4.	Escenarios	24
5.5.	Herramientas de Software.....	26
5.5.1.	NS-3	26
5.5.1.1.	Cálculos	27
5.5.1.2.	Wireshark.....	28
5.5.1.3.	NetAnim.....	29
5.5.1.4.	FlowMonitor.....	30
5.5.1.5.	PythonViz	32
5.5.1.6.	V4Ping	33
5.6.	Estándar.....	34
5.7.	Protocolos de Encaminamiento	36
5.7.1.	Métricas.....	36
5.7.1.1.	Throughput.....	36
5.7.1.2.	Delay	37
5.7.1.3.	Jitter.....	37
5.8.	Estructura del Código.....	37
5.8.1.	Inclusión de Librerías	37
5.8.2.	Definición de espacios de nombres	38
5.8.3.	Función Principal.....	38
5.8.4.	Diseño de la Topología	39
5.8.5.	Modelo de Movilidad	39
5.8.6.	Características de Encaminamiento	40
5.8.7.	Aplicaciones.....	41

5.8.8.	Obtención de Métricas	42
5.8.9.	Simulador	44
6.	RESULTADOS	45
6.1.	Escenario 1: Red MANET con 25 Nodos	45
6.1.1.	Throughput	47
6.1.2.	Delay	48
6.1.3.	Jitter	49
6.2.	Escenario 2: Red MANET con 50 Nodos	50
6.2.1.	Throughput	52
6.2.2.	Delay	52
6.2.3.	Jitter	53
6.3.	Escenario 3: Red MANET con 100 Nodos	54
6.3.1.	Throughput	56
6.3.2.	Delay	57
6.3.3.	Jitter	57
7.	DISCUSIÓN	59
8.	CONCLUSIONES	61
9.	RECOMENDACIONES	63
10.	BIBLIOGRAFÍA	64
11.	ANEXOS	66
	ANEXO 1: CÓDIGO FUENTE	66
	ANEXO 2: RESULTADOS GENERALES DE LA SIMULACIÓN	74
	ANEXO 3: ANÁLISIS DE PLAGIO EN URKUND	91

ÍNDICE DE FIGURAS

Figura 1. Modos de Operación del estándar 802.11x.....	8
Figura 2. Modo de Operación de las MANET	10
Figura 3. Area de la ciudad de Loja utilizada para simulación	25
Figura 4. Línea de código para obtener ficheros de Wireshark.....	28
Figura 5.Ficheros .pcap creados a partir de la ejecución del código NS3.....	28
Figura 6. Ejecución de ficheros .pcap en Wireshark.	29
Figura 7. Líneas de código para obtener un archivo xml de NetAnim	29
Figura 8. Representación visual de la Red utilizando NetAnim.....	30
Figura 9. Línea de código para obtener instalar el Monitor de Flujo	31
Figura 10. Datos de Flujo del archivo xml interpretados en Python	31
Figura 11. Forma de Ejecutar el programa junto a PythonViz	32
Figura 12. Gráfica de simulación de una Red en PythonViz.....	32
Figura 13. Gráfica de comunicación de dos nodos en PythonViz.....	33
Figura 14. Líneas de Código para usar V4Ping en NS-3	34
Figura 15. Gráfica de simulación de la Red en PythonViz	34
Figura 16. Librerías incluidas en el código NS-3 para (a) AODV y (b) DSR	37
Figura 17. Espacios de nombres necesarios para el código NS-3 de (a) AODV y (b) DSR.....	38
Figura 18. Líneas de Código de la Función Principal.....	38
Figura 19. Líneas de Código para crear la topología de la red	39
Figura 20. Líneas de Código para añadir movilidad	40
Figura 21. Líneas de Código para crear Direcciones IP e instalar Internet y encaminamiento AODV	41
Figura 22. Líneas de Código para crear Direcciones IP e instalar Internet y encaminamiento DSR.....	41
Figura 23. Líneas de Código para crear la aplicación UDP	42
Figura 24. Líneas de Código para realizar los cálculos de métricas de QoS.....	43
Figura 25. Líneas de Código para la ejecución de la simulación.....	44
Figura 26. Fase de Descubrimiento de ruta del escenario con 25 nodos	46
Figura 27. Fase de envío de paquetes del escenario con 25 nodos.....	46

Figura 28. Resultados de Throughput para cada Protocolo en el escenario con 25 nodos.....	48
Figura 29. Resultados de Delay para cada Protocolo en el escenario 25 nodos	48
Figura 30. Resultados de Jitter para cada Protocolo en el escenario de 25 nodos..	49
Figura 31. Fase de Descubrimiento de ruta del escenario con 50 nodos	50
Figura 32. Fase de envío de paquetes del escenario con 50 nodos.....	51
Figura 33. Resultados de Throughput para cada Protocolo en el escenario con 50 nodos.....	52
Figura 34. Resultados de Throughput para cada Protocolo en el escenario con 50 nodos.....	53
Figura 35. Resultados de Jitter para cada Protocolo en el escenario con 50 nodos.....	53
Figura 36. Fase de Descubrimiento de ruta del escenario con 100 nodos	55
Figura 37. Fase de Envío de Paquetes del escenario con 100 nodos.....	56
Figura 38. Resultados de Throughput para cada Protocolo en el escenario con 100 nodos.....	56
Figura 39. Resultados de Delay para cada Protocolo en el escenario con 100 nodos.....	57
Figura 40. Resultados de Jitter para cada Protocolo en el escenario con 100 nodos.....	57
Figura 41. Resultado del Análisis de Plagio	91

ÍNDICE DE TABLAS

Tabla 1: Detalles sobre algunas de las revisiones del estándar 802.11	7
Tabla 2. Materiales Utilizados en el trabajo	23
Tabla 3: Parámetros del Primer Escenario de Simulación	45
Tabla 4: Valores de Métricas de QoS del escenario con 25 nodos	47
Tabla 5: Comparativa de las métricas de QoS de cada Protocolo en el escenario de 25 nodos.....	49
Tabla 6: Parámetros del Escenario de Simulación con 50 nodos	50
Tabla 7: Valores de Métricas de QoS del escenario con 50 nodos	51
Tabla 8: Comparativa de las métricas de QoS en cada protocolo en el escenario con 50 nodos	54
Tabla 9: Parámetros del Escenario de Simulación con 100 nodos	54
Tabla 10: Valores de métricas de QoS del Escenario de Simulación con 100 nodos.....	55
Tabla 11: Comparativa de las métricas de QoS en cada protocolo en el escenario con 100 nodos	58
Tabla 12: Comparativa General de las métricas de QoS en cada protocolo	59
Tabla 13: Datos de un Escenario DSR, en los primeros 5 segundos de Simulación	76
Tabla 14: Datos de un Escenario DSR durante 5 segundos a mitad de la Simulación.....	79
Tabla 15: Datos de un Escenario DSR en los 5 segundos finales de la Simulación.	82
Tabla 16: Datos de un Escenario AODV en los primeros 5 segundos de Simulación	85
Tabla 17: Datos de un Escenario AODV 5 segundos a mitad de la Simulación	87
Tabla 18: Datos de un Escenario AODV en 5 segundos al final de la Simulación..	90

ACRÓNIMOS

ACK	Acknowledgement o Acuse de Recibo
AODV	Ad-Hoc on Demand Distance Vector
AP	Access Point
ARPANET	Advanced Research Projects Agency Network
BAN	Body Area Network
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DAD	Duplicate Address Detection
DARPA	Defense Advanced Research Projects Agency
DSDV	Destination-Sequenced Distance-Vector Routing
DSR	Dynamic Source Routing
GHz	Giga Hertz
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LAN	Local Area Network
LPWAN	Low Power Wide Area Network
MAC	Media Access Control
MANET	Mobile Ad-Hoc Network
Mbps	Mega Bits por segundo
NETANIM	Network Animator
NETT	Net Traversal Time
NS-3	Network Simulator 3
PDT	Path Discovery Time
PLR	Packet Loss Ratio

QoS	Quality of Service
RERR	Route Error
RFC	Request for Comments
RIP	Routing Information Protocol
RREP	Route Reply
RREQ	Route Request
RTP	Real-Time Transport Protocol
SPF	Shortest Path First
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network

1. TÍTULO

ESTUDIO COMPARATIVO DE LAS MÉTRICAS DE CALIDAD DE SERVICIO (QoS) EN LOS PROTOCOLOS DE ENCAMINAMIENTO REACTIVOS DSR Y AODV EN REDES MANET

2. RESUMEN

El presente Trabajo de Investigación se efectuó a partir de la identificación del problema, planteamiento de objetivos, revisión de Literatura, elección de la metodología, definiendo dentro de la misma varios pasos a realizar para la posterior exposición y análisis de resultados, todos estos procesos fueron fundamentales para el desarrollo de conclusiones y recomendaciones en cuanto al tema y objetivos planteados.

Dentro de la revisión de literatura se realizó una investigación general de las Redes Ad-Hoc y las posibles formas de aplicarlas, además de información de los protocolos de encaminamiento planteados en el tema, partiendo de esta información se pudo determinar el funcionamiento y ventajas de las redes ad-hoc y el uso de los protocolos de Encaminamiento DSR y AODV para su posterior comparación basada en algunas de sus métricas de calidad de servicio.

Siguiendo el proceso metodológico planteado se realizó la obtención de resultados a través del análisis de las métricas de calidad de servicio escogidas, las métricas fueron obtenidas a partir de la edición de código fuente en lenguaje C++ que se puede encontrar en los archivos del software de simulación de redes NS-3, dentro de la programación se plantea un área determinada, y además la movilidad de los nodos dentro de este espacio, según el modelo Random WayPoint, posteriormente las métricas obtenidas fueron de utilidad para la evaluación de cada protocolo de encaminamiento en los respectivos escenarios. Luego se procedió a la comparación de los dos protocolos propuestos (DSR, AODV) en base a las métricas (Throughput, Delay, Jitter) en 3 escenarios con diferente densidad de nodos en cada uno de ellos. Por último, se plantea basados en estas comparaciones cuál de estos protocolos es el más adecuado para diferentes escenarios en Redes MANET.

Palabras clave: Ad-Hoc, AODV, Delay, DSR, Encaminamiento, Jitter, MANET, Métricas, Movilidad, Nodos, NS-3, QoS, Random Waypoint, Redes, Throughput.

ABSTRACT

This Research was carried out from the identification of the problem, setting objectives, Literature review, choice of methodology, defining within it several steps to be used for subsequent exposure and analysis of results. All these processes were fundamental for the development of conclusions and recommendations regarding the subject and objectives.

Regarding literature review we made a general research of networks Ad-Hoc and the possible forms to be apply, as well the information of the routing protocols proposed in this topic, it was possible to determine the operation and advantages of the ad-hoc networks and the use of the DSR and AODV routing protocols for later comparison based on some of their metrics quality of service.

Following the methodological process proposed, we proceeded to obtain results through the analysis of the selected quality of service metrics, the metrics were obtained from the edition code sources of the programing language C++ which are available in the files of the NS-3 network simulation software, within the programming a certain area is proposed, and also the movements of the nodes inside this space, according the model Random Waypoint, later the metrics obtain were used for the evaluation of each protocol of routing in the respective scenarios. Then we proceeded to the comparison of the two proposed protocols (DSR, AODV) based on the metrics (Throughput, Delay, Jitter) in three scenarios with different density of nodes in each of them. Finally, we place based in these comparisons which one of the protocols is the most suitable for different scenarios in MANET Networks.

Keywords: Ad-Hoc, AODV, Delay, DSR, Routing, Jitter, MANET, Metrics, Mobility, Nodes, NS-3, QoS, Random Waypoint, Networks, Throughput.

3. INTRODUCCIÓN

Con el paso del tiempo y la evolución de la tecnología, las redes inalámbricas se convierten cada vez más en piezas fundamentales en el uso cotidiano de la tecnología, y debido a su evolución y principalmente la de los dispositivos móviles, poco a poco también se empieza a convertir en una necesidad la implementación de movilidad en las redes, esto no se refiere a la movilidad existente en los dispositivos conectados a la red sino a poder crear una red completa que se mantenga a pesar de la movilidad quitando la necesidad de un solo dispositivo administrador de la red.

A partir de la necesidad de movilidad nace el concepto de redes Ad-Hoc las cuales en su concepción incluyen la movilidad como una característica propia de este tipo de redes, creando así un tipo de tecnología capaz de establecer redes en cualquier lugar sin necesidad de infraestructura de red existente. Las redes Ad-Hoc crean también la necesidad de desarrollar nuevos protocolos de encaminamiento que se acoplen a las características de este tipo de redes, teniendo en cuenta los posibles problemas que se pueden presentar en este tipo de redes como la eficiencia del protocolo y el tiempo que a estos les tome realizar la entrega de paquetes.

Debido a la necesidad de determinar un protocolo apropiado para este tipo de redes se ha establecido los objetivos planteados en el anteproyecto la realización de una comparativa entre algunas métricas de calidad de servicio de dos protocolos de encaminamiento reactivo dentro de una red MANET para determinar cuál de ellos se puede considerar mejor en escenarios con diferente densidad de nodos, esto se lograra través del uso de una Herramienta de software libre llamada NS-3 la cual será de utilidad para la simulación de los escenarios planteados y la comparación de los Protocolos.

Dentro de las simulaciones se analizara el comportamiento de dos protocolos de encaminamiento en redes MANET con diferentes densidades de nodos utilizando un modelo de movilidad llamado Random Waypoint para simular un movimiento aleatorio dentro de un área específica como lo sería el movimiento de varias personas dentro de un área determinada, para finalmente realizar los cálculos y obtener los datos necesarios para la comparativa de los protocolos determinando cuál de ellos fue el mejor en los escenarios dispuestos.

4. REVISIÓN LITERARIA

4.1. REDES INALÁMBRICAS

El inicio de las redes inalámbricas remonta al año de 1971, año en el cual se obtiene la primera red de área local inalámbrica (WLAN) en la universidad de Hawaii, dicha red se denominó ALOHANET fue extendida sobre cuatro islas permitiendo así la comunicación bidireccional de computadores de siete campus diferentes por medio de una topología de estrella, al igual que la primera red cableada (ARPANET)¹, la red ALOHA se construyó con fondos de la Agencia de Proyectos de Investigación Avanzados de Defensa (DARPA)², ALOHANET utilizaba radio paquete o packet radio que consiste en el envío de señales digitales a través de radio que son reensambladas al llegar al destino final.

Luego en 1979 en una fábrica de suiza se publican resultados de un experimento usando tecnología infrarroja, y los resultados de este experimento se han considerado como punto de partida de la evolución de lo que hoy conocemos como redes inalámbricas.

Las redes inalámbricas se caracterizan por brindar a los usuarios libre movilidad dentro del rango del dispositivo administrador la red además es considerada como alternativa a las conexiones cableadas.(Bernal, Cortés, & Leguizamón, 2018)

Estas redes se dividen en dos grupos:

El primer grupo se caracteriza por una estructura centralizada con dispositivos encargados de administrar y delegar funciones de conmutación y encaminamiento a otros dispositivos dispuestos para dichas funciones, estos dispositivos administradores de la red se conocen con Access Point o AP. En este grupo la red depende de la posición del dispositivo administrador (AP) y su longitud de onda, con lo cual se determina la cobertura y alcance de la red, por lo cual la movilidad de los usuarios se ve limitada a un radio determinado por el AP.

El segundo grupo no tiene la necesidad de la existencia de un solo dispositivo administrador actuando como intermediario de los usuarios por lo cual no se necesita de

¹ ARPANET: Advanced Research Projects Agency Network

² DARPA: Agencia del Departamento de Defensa de E.E.U.U encargado de Investigar y desarrollar nuevas tecnologías de utilidad militar

dispositivos pertenecientes a las capas 2 y 3 sino que cada uno de los nodos dentro de la red actúan como usuarios y como router al mismo tiempo es decir cada nodo puede enviar y encaminar datos dentro de una red colaborativa conocida como redes Ad-Hoc

4.2. Estándar IEEE 802.11x

Luego del surgimiento y la gran acogida de la tecnología Ethernet alrededor del mundo se hizo mayor la necesidad de brindar conectividad en lugares en los cuales la comunicación a través de redes LAN cableadas no era óptimo para la red, motivo por el cual se pretende la creación de un nuevo estándar conocido como 802.11x en el año de 1990, esta estándar tiene sus inicios como un grupo de trabajo dentro del estándar IEEE 802 de manera que los dispositivos a los cuales se les dificulta la conectividad a través de redes cableadas puedan integrarse a la red de manera inalámbrica utilizando esta tecnología.

La primera versión del estándar 802.11x se denominó 802.11 Legacy que surge en el año de 1997, en el cual se especifican velocidades de 1 Mbps hasta 2 Mbps, esta versión utiliza la banda de frecuencia de 2.4GHz y como medio físico usa señales infrarrojas y el método de Acceso Múltiple con Detección de Portadora y Prevención de Colisiones (CSMA/CA)³ para el acceso al canal.

Es un estándar que define las especificaciones tanto a nivel físico como de enlace de datos (específicamente en la subcapa MAC), que se debe tomar en cuenta al momento de implementar una red LAN inalámbrica.

El estándar 802.11x cuenta con gran variedad de revisiones y cada revisión se diferencia principalmente por las técnicas de modulación que utiliza, la banda de frecuencias en las que opera y la calidad de servicio QoS que brinda. (Rodríguez, 2015)

³ CSMA/CA: método de acceso múltiple el cual permite a varias estaciones ocupar el mismo medio de transmisión, realizando un anuncio antes de cada transmisión para así evitar colisiones

En la tabla 1 se muestra algunas de las revisiones de este estándar

Revisión	Banda de Frecuencia	Velocidad Máxima	Características
802.11 (Legacy)	2.4 GHz	2 Mbps	Primer Estándar 802.11
802.11b	2.4 GHz	11 Mbps	Pierde velocidad al trabajar a máxima velocidad
802.11a	5 GHz	54 Mbps	No obtiene tanta acogida como 802.11b debido a su tardía aparición en el mercado
802.11g	2.4 GHz	54 Mbps	Altamente aceptado debido a su interoperabilidad con 802.11b
802.11n	2.4 GHz y 5 GHz	500 Mbps	Introduce el concepto de MIMO (múltiples entradas – múltiples salidas)
802.11p	5.9 GHz y 6.2GHz	-	Ideal para ser aplicado en automóviles, para comunicaciones de corto alcance
802.11ac	5GHz	6.9 Mbps	Alcance de hasta 400 metros

Tabla 1: Detalles sobre algunas de las revisiones del estándar 802.11

Fuente: Autor

El estándar 802.11x contempla 2 tipos de operación:

El primer modo de operación llamado modo infraestructura se centra en la comunicación de los dispositivos de la red a través de un dispositivo central como lo es un punto de

acceso (AP) es decir el dispositivo actúa como administrador de la red y provee servicios a los dispositivos de comunicación conectados a él.

El segundo modo conocido como ad hoc, fue creado con el fin de proveer comunicación a grandes grupos de usuarios de una manera fácil y rápida sin la necesidad de una infraestructura de la que dependan, sino que cada nodo sea autónomo.

En la figura 1 se muestra los modos de operación que se pueden presentar según el estándar 802.11x. (Rodríguez, 2015)

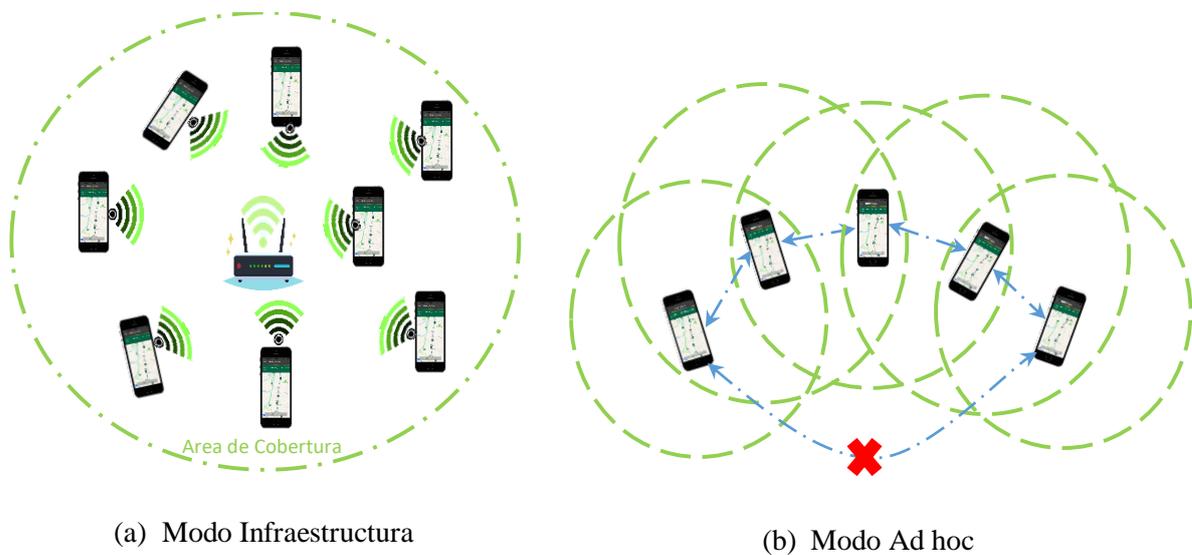


Figura 1. Modos de Operación del estándar 802.11x

Fuente: Autor

4.3. Redes Ad Hoc

Entre los años 1980 y 1990 se realizaron variadas investigaciones sobre redes ad hoc las cuales se centraron principalmente para ser usada con propósitos militares con el objetivo de que las unidades militares se pudieran comunicar con dispositivos de radio de forma cooperativa además de poseer características de movilidad, de modo que cada nodo pueda funcionar como conmutador de paquete o como un terminal.

En los años 90 debido a la mayor demanda de las redes de comunicaciones, así como al incremento de dispositivos móviles y portátiles que permiten la comunicación, y la creación de tecnologías inalámbricas como Wifi, bluetooth, entre otros se planteó las redes ad hoc como una solución a las comunicaciones modernas pretendiendo que con

este tipo de redes se pueda crear una ampliación de cobertura de las comunicaciones inalámbricas.

Las redes ad hoc son redes que no cuentan con una infraestructura central de la cual depende la red, sino que cada dispositivo es utilizado como punto de acceso o encaminador, estas redes pueden tener distintas topologías es decir que cada nodo puede desplazarse y salir de una red y al mismo tiempo formar un nuevo enlace con otro nodo dentro de su área de cobertura y debido a sus características de independencia pueden ser utilizadas en cualquier lugar.

A pesar de que los nodos pertenecientes a una red ad hoc son por lo general móviles no excluyen la posibilidad de inserción de nodos fijos dentro de la misma.

Debido a su movilidad uno de los mayores problemas de las redes ad hoc es el encaminamiento dentro de las mismas por lo cual este tema ha sido blanco de varias investigaciones para determinar cuál es el mejor protocolo aplicable a este tipo de redes.(Quintana & Uribe, 2008)

4.4. Redes Móviles Ad Hoc (MANET)

Las redes Móviles ad Hoc (MANET) por sus siglas en inglés (Mobile Ad-hoc Networks) se refieren a las redes que funcionan a partir del trabajo en conjunto de todos los nodos involucrados que en el caso de este tipo de redes son dispositivos móviles que soportan tecnologías como Wifi 802.11, bluetooth, Zigbee, LPWAN, entre otras.

Las Redes MANET no necesitan del establecimiento de una topología estática, de una estructura fija ni tampoco de dispositivos intermediarios como router, switches o Access point.

Los dispositivos móviles poseen características capaces de ampliar el campo de acción de las redes MANET, debido a que poseen autonomía y movilidad además de estar entre los dispositivos más utilizados en la actualidad, las MANET operan haciendo uso de la

estructura del modelo TCP/IP debido a que realizan transmisiones entre pares (peer to peer)⁴ además dichas redes funcionan como una gran red multipunto.

Las MANET realizan transmisiones basados en adyacencias entre nodos vecinos enviando mensajes en modo de inundación o flooding a los vecinos dentro de su área de cobertura y a su vez si el nodo destino no era ninguno de estos nodos cada vecino realiza un flooding a sus vecinos repitiendo el proceso hasta llegar al nodo destino. (Bernal et al., 2018)

En la Figura 2 se puede observar el modo de operación de una MANET normal, en la figura 2(a) se puede observar la red con la que se va a trabajar en la que se establece el nodo A como origen y el nodo F como destino en la Figura 2(b) se observa el proceso de descubrimiento de ruta iniciando con el reconocimiento de los vecinos del nodo A

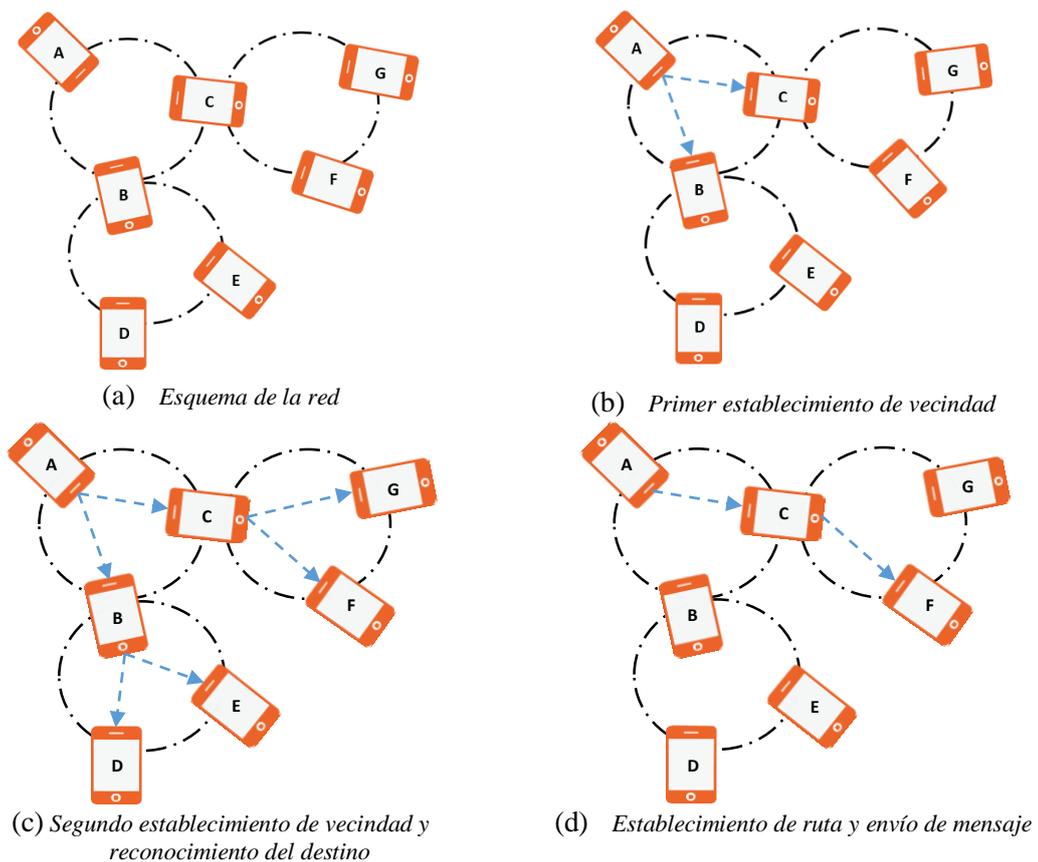


Figura 2. Modo de Operación de las MANET

Fuente: Autor

⁴ Las redes peer to peer (P2P) son redes en las que todos los nodos participantes se consideran como iguales sin la necesidad de existencia de servidores ni clientes fijos, sino que cada nodo puede tomar posición tanto de cliente como de servidor

Como ninguno de los nodos adyacentes al nodo A es el destino se procede a la Figura 2(c) en donde se realiza el reconocimiento de vecinos de los nodos B y C dentro de los cuales se encuentra el destino por lo que se procede a la Figura 2(d) en la cual ya se establece la ruta y procede al envío del mensaje del nodo A al nodo F usando el nodo C como intermediario.

Debido a que las MANET operan de manera distinta a las redes convencionales se hace necesario el uso de protocolos con funciones especializadas aplicadas a este tipo de redes para proveer el funcionamiento óptimo de la red.

Las principales ventajas de las redes MANET son:

- Flexibilidad: las redes MANET pueden ser instaladas de forma sencilla y dinámica es decir que pueden ser instaladas en cualquier lugar o cualquier tipo de situación dependiendo de la necesidad de los nodos que pertenecerán a la red
- Movimiento de los nodos: una de las principales ventajas de una MANET es la libertad que poseen sus nodos para movilizarse dentro de la red permitiendo ampliar el rango de cobertura de la red así mismo permitiendo que los usuarios realicen otras tareas mientras sus dispositivos siguen dentro de la red.
- Descentralización: es considerada una ventaja debido a que no se depende de un solo dispositivo para el encaminamiento de los mensajes, sino que cada nodo puede realizar la función de encaminador es decir cada uno de los nodos poseen autonomía dentro de la red.
- Escalabilidad: debido a que la red no posee una topología estática cualquier nodo puede entrar o salir de la red sin entorpecer el funcionamiento general de la red, esto es posible además porque los dispositivos pertenecientes a la red están en constante movimiento.
- Economía: la instalación de redes MANET no representa costos elevados debido a que no depende de dispositivos externos para administrar la red que por lo general son los que implican mayores gastos dentro de redes convencionales.
- Multi-salto: esta característica dentro de las redes MANET es necesaria debido a que los dispositivos móviles dependen de su batería por lo cual sus señales son de baja potencia teniendo como resultado un área de cobertura pequeña, el Multi-salto ayuda a que a pesar de su escasa área de cobertura los nodos pertenecientes

a la red puedan lograr la comunicación enviando mensajes de reconocimiento a sus nodos adyacentes y estos a los suyos sucesivamente hasta que el mensaje llegue a su destino.

- Formación Autónoma: debido a que las comunicaciones en las redes MANET en cada nodo se realizan entre pares (peer to peer) los nodos forman una topología dinámica en el momento que se integran a la red, de este modo los dispositivos son autónomos de la intervención humana ya que no necesitan ser configurados en el momento de entrar o salir de una MANET.
- Organización autónoma: debido a la topología dinámica que crean los nodos en la red y la movilidad de los mismos las MANET posibilitan los nodos puedan auto-organizarse para que la entrada o salida de dispositivos en la red no afecte el funcionamiento de los demás nodos.(Bernal et al., 2018)

Los principales problemas a superar dentro de MANET son:

- Energía: es uno de los principales problemas que pueden surgir dentro de una MANET debido a que los dispositivos móviles por lo general poseen baterías con un tiempo de vida reducido razón por la cual se provoca dentro de la red caída de nodos, así como roturas de enlaces, afectando el funcionamiento de la red.
- Nuevos protocolos: se puede considerar como un reto dentro de las MANET la necesidad de creación de nuevos protocolos capaces de soportar las características de movilidad, topología dinámica y autonomía de las MANET, pudiendo ser estos protocolos de encaminamiento, protocolos de autoconfiguración, entre otros. El objetivo de estos protocolos se debe centrar en el descubrimiento y mantenimiento de rutas y la eliminación de rutas en desuso además de asegurar que los dispositivos no reciban elevados niveles de consumo de CPU y no consumir mucha energía al mismo tiempo.
- Enlaces Rotos: debido a la movilidad de los nodos en la red se considera un gran problema el hecho de que los nodos pueden entrar y salir de la red causando en ocasiones la rotura de enlaces haciendo necesario volver a calcular las rutas ya establecidas.
- Ancho de banda y medio de transmisión: las MANET pueden ser afectadas por interferencias debidas a que el medio de transmisión que se ocupa es el aire además debido a que son redes inalámbricas poseen anchos de banda reducidos.

- Seguridad: al ser en este caso el aire un medio en el cual todos los nodos comparten su información en las MANET las comunicaciones realizadas dentro de ella pueden ser fácilmente interceptadas por cualquier persona que pueda acceder a la red y obtener así los mensajes q se envíen dentro de la red.
- Procesamiento limitado: comúnmente los dispositivos móviles utilizados en una red ad hoc son pequeños y poseen una capacidad de CPU mínima por lo que se dificulta el procesamiento de algunas funciones de los nodos, como operaciones de seguridad, además que este bajo nivel de procesamiento impide a estos dispositivos soportar los procedimientos de cálculo de métricas y descubrimiento de rutas que poseen los dispositivos de capa 3 en las redes tradicionales.(Bernal et al., 2018)

4.5. Aplicaciones de MANET

Existen varios escenarios en los que las MANET pueden resultar de gran ayuda a continuación se menciona algunos de ellos:

- Educación

Las MANET pueden ser aplicadas dentro del ámbito de la educación en diferentes escenarios como lo puede ser para la ejecución de clases mediante la creación de grupos de trabajo o en prácticas siendo en este caso el maestro un supernodo que puede supervisar el progreso de los estudiantes dentro de la red además de poder aplicar a través de este tipo de redes plataformas e-learning, además se puede usar en la generación de conferencias virtuales así como para hacer posible la ampliación de la red de campus universitarios, aprovechando de esta manera varias de las ventajas de las MANET como el bajo costo que esto implicaría a las instituciones educativas además de proveer un despliegue de la red de una forma sencilla, las aplicaciones de las MANET en educación resultan mucho más útiles y a la vez necesarias en instituciones en las cuales no existe infraestructura de telecomunicaciones como: una escuela rural, para lo cual resulta un reto para las instituciones realmente el hecho de implementar los dispositivos móviles como material de aprendizaje y no como una distracción para los estudiantes.
- Escenarios de emergencia

Las redes MANET llegarían a ser de gran utilidad al implementarse en escenarios en los cuales han ocurrido catástrofes como terremotos, inundaciones, huracanes o tsunamis y han resultado varios daños a la infraestructura de telecomunicaciones del sitio afectado dejando sin comunicación a las personas perjudicadas por lo cual una MANET crearía una red de respaldo mientras se puede reparar la infraestructura de la red evitando así perder comunicación en ningún momento. (Tandazo, Carrión, Gonzalez, & Cueva, 2015)

- Operaciones de búsqueda

Comúnmente dentro de las operaciones de búsqueda se pretende dar informes constantes del progreso y dichas búsquedas pueden ser realizadas en lugares con escasa o nula infraestructura de telecomunicaciones para lo cual podemos utilizar una MANET para evitar perder comunicación con los implicados en la búsqueda.

- Misional

En ocasiones como en reuniones, conferencias, centros comerciales u otros lugares se necesitan de redes para compartir información o permitir comunicación entre dispositivos dentro del lugar, en estos casos el despliegue y ejecución de una red tradicional centralizada con dispositivos de capas 2 y 3 tiende a ser costoso además de complicado y en algunas ocasiones se necesitan solo para un momento conformando esto un gasto innecesario por lo cual una posible solución a estos casos son las redes MANET debido a su simplicidad y facilidad de despliegue ayudando en escenarios temporales brindando también la oportunidad de desmontar la red al terminar el evento.

- Redes Militares

Las redes militares siempre necesitan constantemente de las comunicaciones inalámbricas teniendo en cuenta la movilidad de los campamentos, soldados, vehículos, etc. Así se puede considerar de gran utilidad la utilización de MANET para sus comunicaciones evitando la creación de infraestructuras de telecomunicaciones.

- Sensores

En la actualidad los sensores son instalados en varios de los equipos del hogar desde detectores de humo hasta electrodomésticos los cuales pueden enviar notificaciones de su estado, así como también redes como las BAN (Body Area

Network) que monitorean las funciones vitales de una persona, a través de las MANET se facilitaría el envío de estas notificaciones aun estando en un lugar alejado para mantener a las personas al tanto de esta información. (Bernal et al., 2018)

4.6. Protocolos de encaminamiento en redes convencionales

La clasificación de los protocolos de encaminamiento de redes convencionales se puede dividir en 2 tipos los cuales son Vector Distancia y Estado de Enlace.

4.6.1. Vector Distancia

El enrutamiento de vector de distancia se basa en actualizaciones periódicas creando una tabla en cada nodo, cada tabla es considerada como un vector que proporciona la ruta con la mejor distancia hacia el destino y especifica el enlace que se puede utilizar para llegar a dicho destino, los protocolos de vector distancia están basados en el algoritmo de Bellman-Ford, la distancia de un nodo a su destino se refiere a la métrica con la cual funcione el protocolo como por ejemplo en el caso del protocolo RIP donde su métrica está basada en el número de saltos, el enrutamiento de vector distancia es de gran utilidad en redes pequeñas debido a que, para redes más grandes sus actualizaciones periódicas podrían causar una gran cantidad de recursos usados innecesariamente.

4.6.2. Estado de enlace

El enrutamiento de estado de enlace soluciona uno de los grandes inconvenientes del enrutamiento de vector distancia el cual es el envío constante de actualizaciones que tienden a crear un uso innecesario de recursos, en el caso de los protocolos de estado de enlace las actualizaciones solo se realizan cuando son necesarias como por ejemplo cuando un nodo entra o sale de la red, los protocolos de encaminamiento de estado de enlace se basan en el algoritmo de Dijkstra también conocido como SPF (Shortest Path First) o primero la ruta más corta, otra diferencia respecto a los protocolos de vector distancia, los protocolos de estado de enlace poseen 3 tipos de tablas una de vecindad, una de topología y una de encaminamiento por lo cual tienen una visión total de la topología de la red a diferencia de los protocolos de vector distancia que solo poseen

una visión parcial de la topología. (Ledesma, Baluja, & Coya, 2014)(Bernal et al., 2018)(Coya Rey & Ledesma Quiñones, 2014)

4.7. Protocolos de encaminamiento para redes ad hoc

Dentro de los protocolos de encaminamiento para redes ad hoc se puede encontrar 3 diferentes tipos: reactivos, proactivos e híbridos.

4.7.1. Proactivos

Al igual que los protocolos de vector distancia, los proactivos basan su funcionamiento en el envío de actualizaciones periódicas a todos los demás nodos para realizar el descubrimiento de rutas por lo cual estas redes presentan retardos mínimos al momento del envío de paquetes; una gran ventaja de estos protocolos es que no crean cargas adicionales para realizar el descubrimiento de rutas ya que cuentan con carga fija que incluye el coste de creación de ruta, este tipo de protocolos al igual que los de vector distancia presentan problemas al usarse en redes grandes perdiendo características de escalabilidad y rendimiento conforme la red se hace más grande.

4.7.2. Reactivos

Los protocolos reactivos a diferencia de los proactivos operan bajo demanda es decir que el descubrimiento de rutas solo se realiza en el momento en que se necesita un camino, haciendo uso de inundaciones a los nodos vecinos disminuyendo en parte la sobrecarga que pueden crear los protocolos proactivos al mantener todas las rutas periódicamente, pero logra esto creando una desventaja la cual es el retardo que se crea al momento de buscar rutas nuevas que no están disponibles.

4.7.3. Híbridos

Los protocolos híbridos se caracterizan por la reducción de la sobrecarga que poseen los proactivos y la reducción del retardo de descubrimiento de ruta que poseen los protocolos reactivos es decir son llamados híbridos debido a que poseen tanto características de protocolos proactivos como de protocolos reactivos. (Maria A. Murazzo., Rodríguez, & Martínez, 2008)(Coya Rey & Ledesma Quiñones, 2014)

4.8. Métricas de Calidad de Servicio QoS

Las métricas se consideran una forma cuantitativa de demostrar si un componente o proceso posee una característica dada, en el caso del encaminamiento una métrica es un valor que ayuda a determinar la mejor ruta para el envío de paquetes.

La calidad de servicio QoS es conocida como la capacidad de una red para proveer diferentes niveles de servicio con varios tipos de tráfico permitiendo así diferenciar y categorizar servicios de mayor o menor importancia para dar prioridad a las aplicaciones que más necesiten de la red.

Las métricas de calidad de servicio son parámetros que hacen posible determinar la calidad de la transmisión realizada para garantizar una comunicación fiable algunas de las más utilizadas en redes ad hoc son las siguientes:

4.8.1. Tasa de Paquetes perdidos (PLR)

Esta métrica hace referencia al porcentaje de paquetes que pueden perderse en el trayecto del envío debido a interferencias u otros factores como lo puede ser que el buffer este lleno en el momento que llega el paquete.

4.8.2. Retardo (Delay)

Esta métrica brinda información sobre el tiempo promedio que demora un paquete en llegar a su destino, este retardo que sufren los paquetes puede ser causado por varios factores como buffer lleno, o encolamiento de paquetes además se puede considerar que el paquete puede haber tomado una ruta más larga hacia su destino.

4.8.3. Jitter (Variación del Retardo de Paquetes)

Es la variación que presentan los paquetes en el retardo con el que llegan al destino, esta métrica surge debido a que el Delay de cada paquete no es el mismo por lo cual se vuelve necesario saber cómo se comporta esta variación en cada paquete para determinar rangos promedio para esperar recibir un paquete. (Gómez et al., 2014)(Rodríguez, 2015)

4.8.4. Consumo de Energía

El consumo de energía promedio por paquete recibido es una métrica que proporciona el valor promedio de energía necesaria para completar una transferencia de paquetes, esta métrica es muy importante en las MANET debido a que los dispositivos participantes poseen baterías que pueden poseer poco tiempo de vida.

4.8.5. Ancho de Banda

Es considerado como la cantidad máxima de datos (bits) que se puede enviar de un extremo a otro dependiendo del medio que se utilice.

4.8.6. Rendimiento

Es considerado como el ancho de banda luego de ser afectado por la cantidad de usuarios conectados a la red además de otros parámetros como distancia y tráfico.

4.8.7. Throughput

Es la probabilidad de éxito en la entrega de un mensaje sin considerar los encabezados del paquete, ACK, retransmisiones, etc., es decir es la probabilidad de entregas de los datos puros del usuario hasta el destino. (Flores, Iliana, Ángel, & Carlos, 2017)(Rodríguez, 2015)

4.9. Protocolos de Autoconfiguración

Los protocolos de autoconfiguración son protocolos que permiten a las MANET mantener sus características de auto-creación, auto-organización, y auto-administración entre los cuales se pueden destacar 3 tipos:

Estado completo: los nodos conocen el estado de toda la red y las direcciones son asignadas de manera centralizada o distribuida.

Sin estado: cada nodo elige aleatoriamente su dirección y se asegura que su dirección no esté ocupada por medio de un proceso de detección de dirección duplicada (DAD).

Híbridos: combinan los protocolos mencionados anteriormente para mejorar la autoconfiguración. (Ledesma et al., 2014)

4.10. Modelos de Movilidad

Movilidad aleatoria: en este modelo los nodos se mueven de forma libre y sin restricción alguna; cada nodo define su destino, velocidad y dirección independientemente de los demás.

Movilidad en Patrones: en este modelo los nodos se mueven siguiendo un patrón específico como círculo o malla.

Movilidad controlada: en este modelo los nodos realizan movimientos con propósitos específicos.

4.11. Modelo Random Waypoint

De acuerdo con el modelo de destino aleatorio o Random Waypoint (RWP) los nodos de una red ad hoc se desplazan en línea recta y con velocidad constante entre dos puntos elegidos al azar dentro del espacio limitado para los movimientos. Así, el nodo determina las coordenadas del siguiente destino mediante una variable aleatoria uniformemente distribuida en cada dirección. Una vez que se alcanza un destino y previamente a elegir el siguiente, el modelo RWP permite pausas, normalmente caracterizadas con un valor constante. Igualmente, en la mayor parte de las implementaciones, la velocidad constante de cada trayecto se suele decidir también a través de una distribución uniforme. (Casilari & Triviño, 2014).

El Modelo RWP no se encuentra enfocada a ningún escenario de aplicación específico por lo tanto se considera un modelo generalista, debido a su simplicidad es utilizado comúnmente para realizar pruebas.

4.12. Protocolo AODV (Ad hoc On demand Distance Vector)

Este protocolo está basado en DSDV y DSR, posee características que permiten que sus nodos puedan encontrar y mantener rutas hacia su destino, AODV es un protocolo reactivo es decir las rutas se establecen solo cuando se necesitan, se utiliza vectores de distancia y distancias medidas en saltos, el protocolo AODV soporta unicast y broadcast y se encuentra descrito en el RFC 3561.

Los paquetes que envía el protocolo AODV no tienen conocimiento de la red en la que trabajan ya que solo conocen la ruta de origen a destino, este protocolo no mantiene las rutas para cada nodo, sino que como su nombre lo dice trabaja bajo demanda es decir las rutas se descubren según se vayan necesitando.

El protocolo AODV tiene la capacidad de crear tablas de enrutamiento en cada nodo evitando así que los paquetes deban cargar con las rutas. Todas las rutas incluidas en la tabla de AODV contienen un número de secuencia para identificarla y un temporizador o lifetime, lo cual ayuda a que el protocolo pueda determinar qué información es nueva y cual es antigua para así evitar bucles y transmisiones en rutas caducas, todos los nodos pertenecientes a la red deben mantener actualizado su número de secuencia para asegurar el correcto funcionamiento de AODV.

El protocolo AODV hace posible el mantenimiento de rutas mediante mensajes HELLO los cuales ayudan a informar si existen fallos en la red es decir si un nodo pierde el enlace con su vecino es porque no ha recibido dicho mensaje, a pesar de ser de gran utilidad estos mensajes pueden provocar un aumento en el consumo de ancho de banda de la red.

Este protocolo puede ser utilizado en redes que no puedan presentar nodos maliciosos debido a que AODV no presenta ningún tipo de seguridad, sino que confía en cada uno de los nodos dentro de la red.

En el protocolo AODV se asume que la primera ruta descubierta es la más corta lo cual no será siempre una realidad por lo cual se puede terminar eligiendo una ruta más larga.

Entre las principales ventajas del protocolo AODV se puede considerar el hecho de que no crea tráfico adicional en la transmisión y no requiere mucho espacio de almacenamiento de datos. Además del consumo de ancho de banda ocasionado por el envío de mensajes HELLO se puede considerar una desventaja de este protocolo que ocupa mayor tiempo que otros protocolos para el descubrimiento y establecimiento de rutas. (Perkins, Belding-Royer, & Das, 2003)(Murazzo, Rodríguez, & Martínez, 2009)(Gómez et al., 2014)

4.13. Protocolo DSR (Dynamic Source Routing)

El protocolo DSR se presenta en el año 1994 por Johnson y Maltz proponiendo este protocolo como una solución al correcto encaminamiento en redes MANET.

Es un protocolo de encaminamiento reactivo que consta de la fase de descubrimiento de ruta bajo demanda y una fase de mantenimiento de ruta, los paquetes de petición utilizados en el proceso de descubrimiento se almacenan en la cabecera del mensaje cuando este paquete llega a un nodo que conoce el destino o cuando llega al nodo destino se envía una respuesta al nodo origen incluyendo la información de los nodos por los cuales ha llegado, por lo tanto los paquetes llevan en su cabecera la información de la ruta completa a seguir.

El protocolo DSR es capaz de adaptarse a realizar cambios en el encaminamiento rápidamente cuando los nodos tienden a aumentar su velocidad, experimentando una sobrecarga mínima.

Los nodos que utilizan el protocolo DSR poseen una tabla cada uno llamada tabla de rutas o Route cache a partir de la cual el nodo determinara la ruta hacia el nodo destino o realizara el descubrimiento en caso de que no haya una ruta existente a ese destino a este proceso se denomina Route Discovery, mientras se realiza este proceso los paquetes se mantienen dentro de un buffer donde pueden esperar o ser descartados de esta manera este protocolo designa la tarea de detección de pérdidas y solicitudes de retransmisión a protocolos de capas superiores.

Este protocolo posee un mecanismo opcional llamado overhearing que mediante la observación del tráfico de datos ayuda a obtener información adicional de encaminamiento permitiendo a los nodos obtener un mayor conocimiento de la topología de red y optimizando de esta manera la elección de la ruta más corta, a pesar del beneficio que ofrece también necesita que los nodos intermedios utilicen mayor cantidad de procesamiento. (Johnson, Hu, & Maltz, 2007)(Subiela & Fernández, 2009)(Chamba Macas, 2014)

Los protocolos AODV y DSR utilizan 3 tipos de mensaje en el proceso de descubrimiento y mantenimiento de rutas los cuales son:

Route Request (RREQ): es enviado mediante broadcast por un nodo que solicita una ruta a un destino, un identificador junto a la dirección IP de origen, identifica unívocamente un RREQ, este identificador aumenta su valor en uno cada vez que un emisor envía un Route Request, la RREQ se almacena durante un tiempo PDT y así no puede ser procesado más de una vez por cada nodo, cuando una estación recibe un RREQ no conocido configura una ruta de vuelta al origen.

Route Reply (RREP): es enviado por el destino hacia el origen, este espera un tiempo NETT para que llegue el RREP, en caso de no recibirlo se vuelve a enviar un RREQ hasta un número determinado de intentos RREQ TRIES, cuando se recibe el RREP en la estación origen la ruta queda establecida y se puede enviar mensajes, opcionalmente se puede también utilizar mensajes RREP ACK.

Route Error (RERR): En caso de existir enlaces rotos se debe avisar de dicho error mediante un mensaje de broadcast RERR que al ser recibido por un nodo este invalida la ruta por la cual llego el mensaje. (Gómez et al., 2014)(Chamba Macas, 2014; Ferro, Rodriguez, & Gonzalez, 2016)

5. MATERIALES Y MÉTODOS

5.1. Materiales

En la tabla 2 se detalla los materiales utilizados para el presente proyecto de investigación.

Cantidad	Descripción
1	Computadora
1	Simulador de redes NS-3
1	Editor de Texto (Visual Studio Code)
1	Visualizador de redes (PythonViz)

Tabla 2. Materiales Utilizados en el trabajo

Fuente: Autor

5.2. Método Cualitativo

El método Cualitativo es aquel que se basa en el estudio de las cualidades de un evento o fenómeno objeto de investigación para que a partir de las mismas se pueda lograr una mayor comprensión del proyecto y concluir satisfactoriamente los resultados, los cuales no serán generalizables.

En la presente investigación se utilizará una metodología con enfoque cualitativo debido a que se realizará una comparativa basándose en las características de las métricas de calidad de servicio de los protocolos de encaminamiento DSR y AODV además teniendo en cuenta que los datos obtenidos en la investigación no podrán ser generalizados debido a que se habla de un entorno ad hoc en la cual cada uno de los nodos se moverán de manera aleatoria.

Además del método cualitativo se plantea un enfoque mayormente teórico ya que para la realización de la investigación se tomarán en cuenta cuestiones ya investigadas con anterioridad y a partir de ellas se logrará obtener nuevos datos a partir de simulaciones dando también un enfoque empírico para el estudio comparativo.

La investigación se realizará mediante un proceso de comparación en varios escenarios con distintas densidades de nodos dentro de redes Ad hoc encaminadas según los protocolos DSR y AODV tomando en cuenta para cada escenario las métricas de calidad de servicio: Throughput, Delay, Jitter.

5.3. Modelo de Movilidad

En el presente proyecto se ha planteado el uso del modelo de movilidad Random Waypoint dado que es un modelo aleatorio que a diferencia del modelo Random Walk presenta un tiempo de pausa además de la velocidad de los nodos y tiempo de movilidad, por lo tanto, es un modelo ideal para representar varias personas o dispositivos moviéndose dentro de un área determinada, que en este caso serán los nodos dentro de la red MANET.

Tomando en cuenta que uno de los objetivos específicos de esta investigación se basa en estudiar la utilidad de este modelo de movilidad dentro de una red de personas en un escenario de búsqueda y rescate, las dimensiones para estos espacios de emergencia no son definidas, sin embargo esta evaluación se realizara mediante el estudio del comportamiento de los nodos dentro de la red utilizando este modelo para simular la interacción de nodos a distintas velocidades determinando movimientos aleatorios como lo son los que se pueden dar en un evento emergente dentro del cual las personas pueden estar aglomeradas en un espacio específico o realizar búsquedas por distintas posiciones dentro del escenario, cabe recalcar se puede tomar como ejemplo de estas situaciones: asaltos, accidentes de tránsito, desastres naturales, emergencias médicas, etc.

A continuación, se presenta las características físicas del escenario de simulación para posibles situaciones de emergencia e incluso de actividades cotidianas.

5.4. Escenarios

Para este trabajo se han planteado 3 escenarios en los cuales se evaluará el comportamiento de la red, utilizando 25 nodos para el primer escenario, 50 para el segundo y 100 nodos para el ultimo escenario los cuales se eligieron basado en las capacidades de procesamiento tanto del software NS-3 como de la computadora utilizada, además de ser densidades de nodos comúnmente usadas en los trabajos de investigación referentes a redes Ad-Hoc debido a que estas densidades permiten la evaluación de los protocolos de una manera eficiente sin crear sobrecargas innecesarias dentro de la red para la evaluación de los protocolos en condiciones normales e igualitarias para ambos.

En este caso el área simulada en cada escenario será de 500x500 metros cuadrados con forma cuadrangular el cual fue elegido de manera que permita simular espacios medianamente concurridos. Como se puede apreciar en (Rodríguez, 2015) existen 3 tipos

de escenarios para redes Ad Hoc (pequeño, mediano y grande), se toma en cuenta las características de cada tipo de escenario para la elección de este debido a que un escenario pequeño sirve para simular escenarios en los cuales la densidad y movilidad de los nodos es baja, a diferencia de los escenarios medianos los cuales permiten simular escenarios con mayor número de nodos y velocidades de los mismos, además de que se utilizan específicamente para redes MANET como por ejemplo en aeropuertos, campus universitarios congresos, etc., por último los escenarios grandes se caracterizan por utilizar más de 150 nodos que supera las densidades planteadas y soportadas para un funcionamiento eficiente de la simulación y se usan principalmente para la unión de dos o más edificios. De esta manera se elige un escenario mediano debido a que se simulara una red manet, permitiéndonos posicionar las diferentes densidades de nodos planteadas y permita apreciar de mejor manera los envíos y movimientos de los nodos tanto visualmente como en el procesamiento interno del software para evitar fallos.

Para la creación de dicho escenario se ha tomado como referencia la ciudad de Loja, esta área estará limitada para el largo del escenario por las calles 18 de noviembre y 24 de mayo y para el ancho del escenario las calles Azuay y José Antonio Eguiguren, tomando en cuenta que este sector es muy concurrido y la movilidad de las personas se podría tomar como tipo aleatorio, cumpliendo con las características planteadas para los nodos que se simulará.

La Figura 3 muestra el área delimitada en Google Maps



Figura 3. Área de la ciudad de Loja utilizada para simulación

Fuente: Google Maps.

5.5.Herramientas de Software

Los protocolos de encaminamiento para redes Ad hoc aún se encuentran en investigación por lo cual para estudiarlos se necesita de software especializado en el tema de simulación de redes que permita observar las características de los paquetes así como de la red misma tanto de manera visual como de manera cuantificada, este tipo de software al ser una herramienta de código abierto permite su uso en cualquier sistema operativo, pero es recomendable su uso en distribuciones libres como lo es Linux debido a que fueron creados a partir de estas.

5.5.1. NS-3

Para la elección del software de simulación se tomó en cuenta la investigación realizada por (Rodríguez, 2015) dentro de la cual se encuentra la tabla 3 en la que se muestra varios criterios sobre cada simulador y se demuestra la razón de elegir al simulador NS-3 por sobre otros.

Valor	OPNET	OMNET	NS2	GNS3	NS3	NC-TUNS
Uso Investigativo	Alto	Alto	Alto	Bajo	Alto	Alto
Tipo de Licencia	Comercial	Libre	Libre	Libre/ Comercial	Libre	Libre
Curva de Aprendizaje	Alto	Alto	Alto	Bajo	Alto	Alto
Plataformas de Soporte	Windows, Unix	Windows, Unix	Windows, Mac, Unix	Windows, Mac, Unix	Windows, Mac, Unix, Linux	Linux
Interfaz Gráfica	Alto	Medio	Bajo	Alto	Medio	Bajo
Graficación de Resultados	Buena	Aceptable	No tiene	Limitado	Alto	Aceptable

Tecnologías de nivel 2 y 3 que soporta	Alto	Alto	Alto	Bajo	Medio	Medio
Trafico que permite modelar	Alto	Alto	Medio	Bajo	Alto	Alto
Soporte en redes inalámbricas	Bueno Medio	Medio	Bueno	Nulo	Alto	Aceptable

Tabla 3: Criterios para la elección del simulador

Fuente: (Rodríguez, 2015)

Teniendo en cuenta también que NS-3 es utilizado principalmente para la simulación de:

- Redes Ad Hoc
- Redes WiMAX, WiFi, LTE
- Redes de Sensores
- Redes Fijas TCP/IP

NS-3 es una herramienta que a través de código en lenguaje C++ permite la simulación de diferentes tipos de redes; en el caso de este proyecto se utilizara código para realizar la simulación de una red MANET, usando el modelo de movilidad Random Waypoint para escenarios con distintas densidades de nodos para cada protocolo de encaminamiento (AODV y DSR), estos códigos permitirán brindar características a los nodos propias de este tipo de red y además la recopilación de datos a través de distintos métodos como puede ser obtener datos por medio de cálculos dentro del mismo código, llamar librerías o crear ficheros que puedan ser procesados por otro tipo de herramienta como: Wireshark, NetAnim o PythonViz.

5.5.1.1. Cálculos

Se puede obtener información de una simulación agregando líneas de código capaces de realizar los cálculos que se desean, para realizar este tipo de simulación se deberá tener el conocimiento suficiente del tipo de dato que se desea recopilar

además de conocimiento del lenguaje de programación para utilizar las funciones que brinda el programa como tal en ese aspecto.

5.5.1.2. Wireshark

Es una Herramienta de Software externa a NS-3 con la cual se puede obtener acceso a los datos de la simulación realizando capturas de los paquetes o flujos de datos que realiza NS-3 al ejecutarse, estas capturas se obtienen al habilitar dentro del código la creación de ficheros con extensión .pcap y su instalación en todos los nodos de la simulación lo que permitirá que todos los flujos de paquetes se guarden en estos ficheros para posteriormente ser interpretados mediante la herramienta de software wireshark.

```
internet.EnablePcapIpv4All("PCAP");
```

Figura 4. Línea de código para obtener ficheros de Wireshark

Fuente: Autor

Al ejecutar el programa NS-3 habiendo incluido el comando de la Figura 4. Se crearán los ficheros con extensión .pcap correspondientes a los flujos de datos de cada nodo dentro de la red.

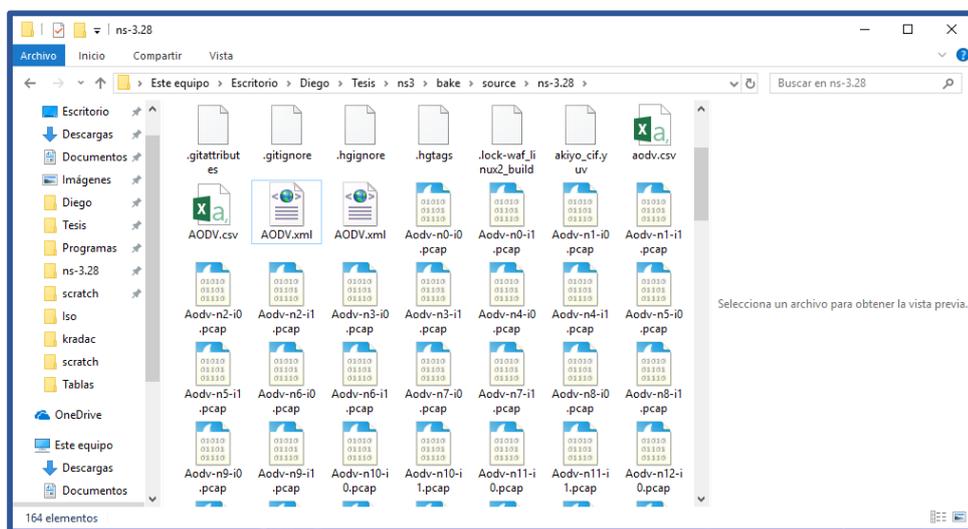


Figura 5. Ficheros .pcap creados a partir de la ejecución del código NS3.

Fuente: Autor

En la Figura 5 se muestra varios de los ficheros creados para su posterior análisis utilizando el software Wireshark.

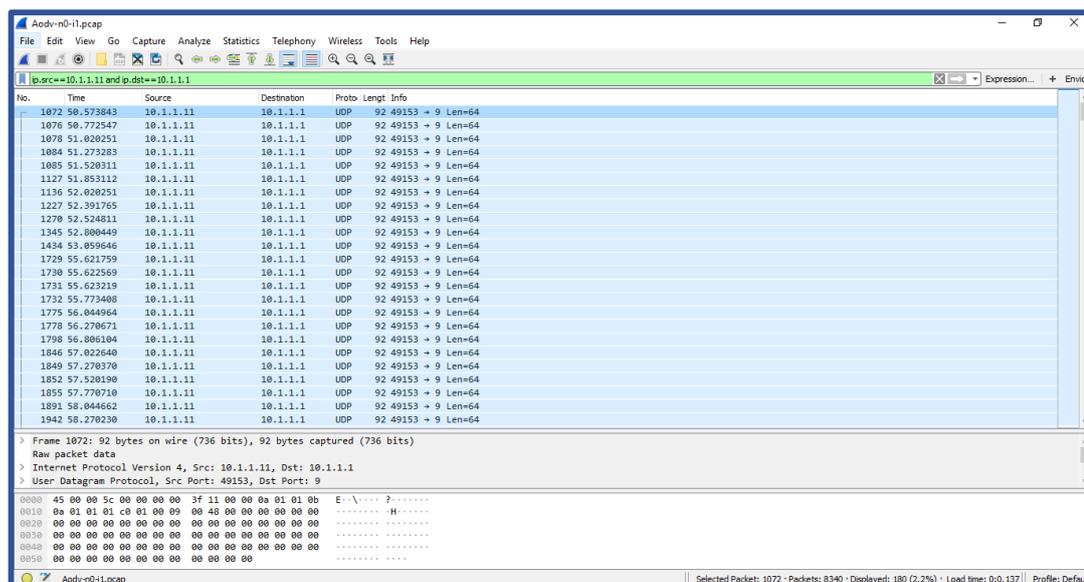


Figura 6. Ejecución de ficheros .pcap en Wireshark.

Fuente: Autor

Al abrir cualquiera de los ficheros se podrá observar dentro de wireshark todos los flujos de datos que han realizado los nodos dentro de la red, así como se muestra en la figura 6, pudiendo obtener datos específicos de ellos como: el protocolo utilizado, las direcciones ip, el tipo de paquete enviado, entre otros.

5.5.1.3. NetAnim

NetAnim es una herramienta que viene dentro del paquete de NS-3 la cual permite el análisis de manera visual de la simulación NS-3 a través de un archivo .xml que guardará datos de la red como en este proyecto: la movilidad, el protocolo de transporte, dirección IP, protocolo de encaminamiento por defecto solo se activará la visualización de la simulación a través de estos datos pero se puede obtener más información de estos activando la función de metadatos la cual proporcionará más datos sobre los flujos de paquetes de toda la simulación así como una gráfica de flujos.

```
AnimationInterface anim ("aadv.xml");
anim.EnablePacketMetadata(true);
```

Figura 7. Líneas de código para obtener un archivo xml de NetAnim

Fuente: Autor

Al ejecutar el programa de NS-3 incluyendo las líneas de código mostradas en la figura 7 se crea un archivo con extensión .xml el cual se usa para su análisis con la

herramienta NetAnim, además se activa la obtención de metadatos para además de visualizar el comportamiento de la red obtener datos específicos de la misma.

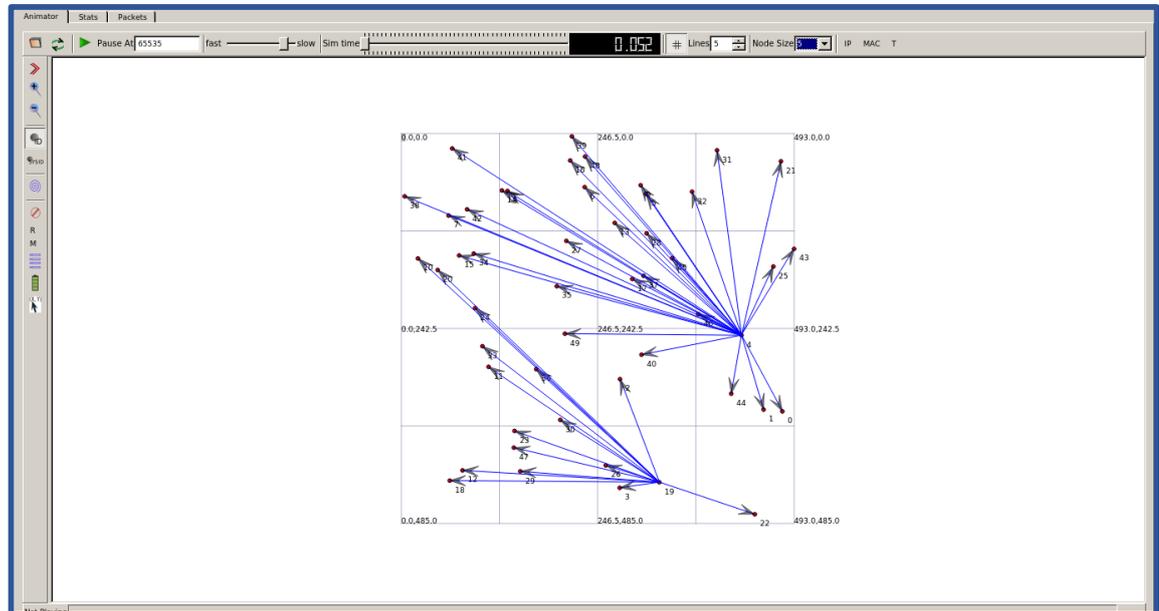


Figura 8. Representación visual de la Red utilizando NetAnim

Fuente: Autor

Se puede observar en la Figura 8 que la simulación de NetAnim muestra la relación entre todos los nodos de la red, teniendo en cuenta un entorno Ad-Hoc en el cual los nodos guardan relación entre todos y no muestra una comunicación directamente a un nodo específico; se torna un tanto difícil distinguir cuales son los flujos de paquetes que se desea observar y cuáles son los mensajes de la fase de descubrimiento y de mantenimiento de ruta de cada protocolo de encaminamiento.

5.5.1.4. FlowMonitor

FlowMonitor es una librería de NS-3 que permite la obtención de varios datos del envío y recepción de paquetes como lo son bytes transmitidos y recibidos, retardos, Jitter, Throughput, tasa de pérdida de paquetes, entre otras es uno de los métodos de captura de datos más directos y completos dentro de las librerías NS-3, está basado en un archivo de Python (.py) dentro del cual se encuentran las fórmulas para la obtención de las diferentes métricas antes mencionadas, para esto se debe llamar a la librería e instalarla a todos los nodos, por ultimo crear un archivo .xml que guardará todos los datos del monitor de flujo para luego ser interpretado a través del archivo de Python, cabe recalcar que a pesar de que FlowMonitor es un método

eficiente, no funciona para todos los tipos de flujo de datos, por lo tanto antes de usarlo se debe saber de qué tipo es el flujo que se desea analizar.

```
Ptr<FlowMonitor> flowmon;  
FlowMonitorHelper flowmonHelper;  
flowmon = flowmonHelper.InstallAll ();  
flowmon->SetAttribute("DelayBinWidth", DoubleValue(0.01));  
flowmon->SetAttribute("JitterBinWidth", DoubleValue(0.01));  
flowmon->SetAttribute("PacketSizeBinWidth", DoubleValue(1));
```

Figura 9. Línea de código para obtener instalar el Monitor de Flujo

Fuente: Autor

Luego de la ejecución del programa incluyendo el código de la Figura 9 se obtendrán un archivo con extensión .xml el cual guardará los datos de flujo de cada uno de los nodos de la red.

```
FlowID: 511 (UDP 10.1.1.34/654 --> 10.1.1.33/654)  
TX bitrate: 7.08 kbit/s  
RX bitrate: 7.08 kbit/s  
Mean Delay: 30.93 ms  
Packet Loss Ratio: 0.00 %  
TX Bytes: 240.00 bytes  
RX Bytes: 240.00 bytes  
Hop Count: 1.00 hops  
TX Packets: 5.00 packets  
RX Packets: 5.00 packets  
Throughput: 0.35 bps  
Jitter: 8936.6376 us  
FlowID: 512 (UDP 10.1.1.36/654 --> 10.1.1.33/654)  
TX bitrate: 3.73 kbit/s  
RX bitrate: 3.73 kbit/s  
Mean Delay: 116.10 ms  
Packet Loss Ratio: 0.00 %  
TX Bytes: 96.00 bytes  
RX Bytes: 96.00 bytes  
Hop Count: 1.00 hops  
TX Packets: 2.00 packets  
RX Packets: 2.00 packets  
Throughput: 0.14 bps  
Jitter: 71358.6770 us
```

Figura 10. Datos de Flujo del archivo xml interpretados en Python

Fuente: Autor

Los datos obtenidos en el archivo xml del monitor de flujo se pueden analizar a partir de la ejecución de un fichero de Python que interpretará los datos del archivo xml y realizará los cálculos de métricas a partir de los mismos mostrando todos los datos de flujos de paquetes en la terminal, en la figura 10 se puede observar todos los valores obtenidos de esta ejecución.

5.5.1.5. PythonViz

PythonViz es una herramienta de Python que permite la visualización de datos, en este caso se lo puede usar conjuntamente con NS-3 para habilitar la visualización de la red en tiempo real permitiendo ver los nodos, así como los envíos que realiza la red simulada, además permite la visualización de datos específicos de los nodos como, el tiempo de simulación, el número de paquetes transmitidos, numero de paquetes recibidos además del tamaño total de los paquetes transmitidos y recibidos por cada nodo. Para esto se debe simplemente llamar a esta herramienta desde la terminal junto a los comandos que se utilizan para correr la simulación de NS-3 como lo muestra la figura 11.

```
Tesis/ns3/bake/source/ns-3.28$ ./waf --run comparacion --vis
```

Figura 11. Forma de Ejecutar el programa junto a PythonViz

Fuente: Autor

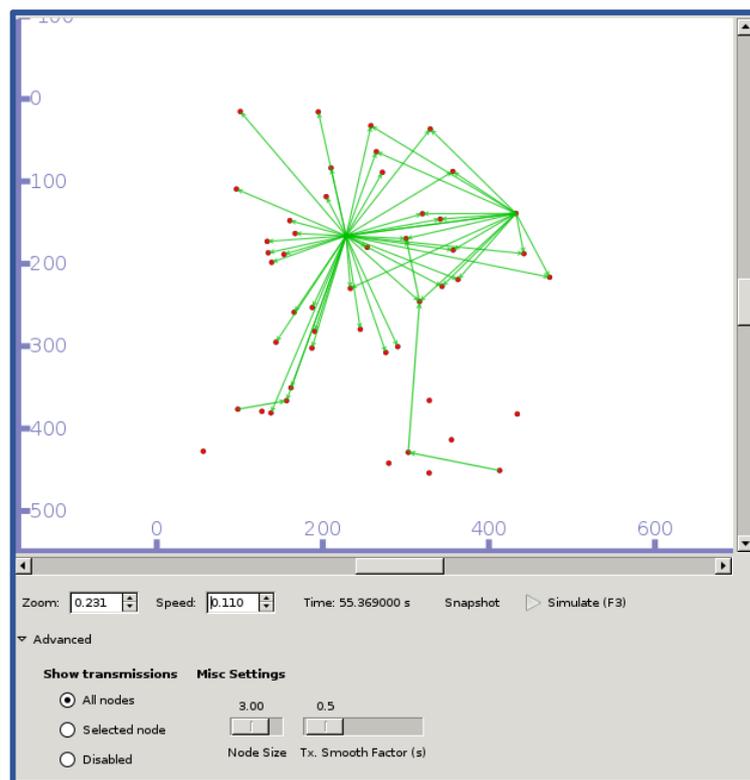


Figura 12. Gráfica de simulación de una Red en PythonViz

Fuente: Autor

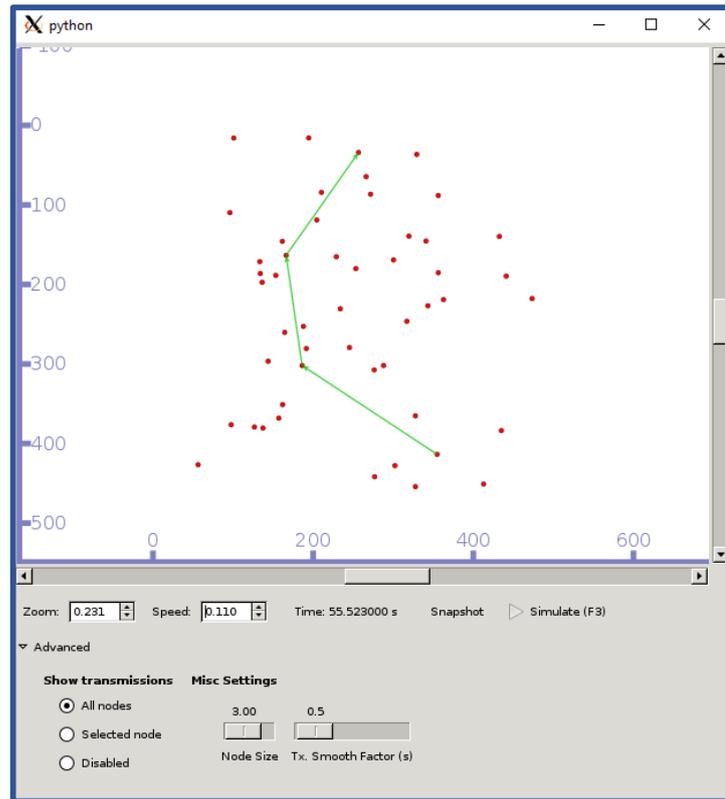


Figura 13. Gráfica de comunicación de dos nodos en PythonViz

Fuente: Autor

Luego de ejecutar el comando de ejecución del programa integrándolo a PythonViz se obtiene una gráfica de la red como se observa en la figura 12, en esta se muestra los flujos de mensajes de descubrimiento y mantenimiento de ruta y los flujos de paquetes de datos por separado, la figura 12 muestra varios envíos de descubrimiento de ruta y la figura 13 se puede observar un envío de paquetes entre 2 nodos específicos, a partir de esto podemos decir que PythonViz permite observar de una mejor manera estos flujos y permite diferenciarlos en este tipo de simulación, a diferencia de NetAnim que no permitía distinguir estos dos tipos de envíos.

5.5.1.6. V4Ping

Es una librería incluida en el software NS-3 que se utiliza para realizar un ping a una dirección específica, esta librería nos brinda el mismo resultado que se obtiene al utilizar el comando ping en Windows o Linux, para utilizar esta librería V4Ping dentro de NS-3 se debe declarar la librería dentro del programa que se desea utilizar y luego realizar la llamada de la aplicación ping de esta librería y proporcionarle datos del nodo que envía y hacia qué dirección, luego se instala esta aplicación a todos los nodos de la red, los resultados que brinda esta librería serán datos de

tiempo de envío, TTL, número de secuencia y al final muestra los datos de paquetes enviados, paquetes recibidos y la tasa de pérdida de paquetes que presento la simulación.

```
V4PingHelper ping (interfaces.GetAddress (size - 1));
ping.SetAttribute ("Verbose", BooleanValue (true));

ApplicationContainer p = ping.Install (nodes.Get (0));
p.Start (Seconds (0));
p.Stop (Seconds (totalTime) - Seconds (0.001));
```

Figura 14. Líneas de Código para usar V4Ping en NS-3

Fuente: Autor

Luego de ejecutar el programa NS-3 incluyendo las líneas de código de la Figura 14, se obtiene dentro de la terminal los datos antes mencionados como se muestra en la Figura 15.

```
Creating 10 nodes 100 m apart.
Starting simulation for 100 s ...
PING 10.0.0.10 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=7 ttl=56 time=2101 ms
64 bytes from 10.0.0.10: icmp_seq=8 ttl=56 time=1110 ms
64 bytes from 10.0.0.10: icmp_seq=9 ttl=56 time=110 ms
64 bytes from 10.0.0.10: icmp_seq=10 ttl=56 time=12 ms
64 bytes from 10.0.0.10: icmp_seq=11 ttl=56 time=9 ms
64 bytes from 10.0.0.10: icmp_seq=12 ttl=56 time=10 ms
64 bytes from 10.0.0.10: icmp_seq=13 ttl=56 time=12 ms
64 bytes from 10.0.0.10: icmp_seq=14 ttl=56 time=9 ms
64 bytes from 10.0.0.10: icmp_seq=15 ttl=56 time=10 ms
64 bytes from 10.0.0.10: icmp_seq=16 ttl=56 time=10 ms
64 bytes from 10.0.0.10: icmp_seq=17 ttl=56 time=11 ms
64 bytes from 10.0.0.10: icmp_seq=18 ttl=56 time=10 ms
64 bytes from 10.0.0.10: icmp_seq=19 ttl=56 time=14 ms
64 bytes from 10.0.0.10: icmp_seq=20 ttl=56 time=11 ms
64 bytes from 10.0.0.10: icmp_seq=21 ttl=56 time=11 ms
64 bytes from 10.0.0.10: icmp_seq=22 ttl=56 time=10 ms
64 bytes from 10.0.0.10: icmp_seq=23 ttl=56 time=14 ms
64 bytes from 10.0.0.10: icmp_seq=24 ttl=56 time=12 ms
64 bytes from 10.0.0.10: icmp_seq=25 ttl=56 time=9 ms
64 bytes from 10.0.0.10: icmp_seq=26 ttl=56 time=12 ms
64 bytes from 10.0.0.10: icmp_seq=27 ttl=56 time=11 ms
64 bytes from 10.0.0.10: icmp_seq=28 ttl=56 time=11 ms
64 bytes from 10.0.0.10: icmp_seq=29 ttl=56 time=10 ms
64 bytes from 10.0.0.10: icmp_seq=30 ttl=56 time=11 ms
64 bytes from 10.0.0.10: icmp_seq=31 ttl=56 time=10 ms
64 bytes from 10.0.0.10: icmp_seq=32 ttl=56 time=11 ms
64 bytes from 10.0.0.10: icmp_seq=33 ttl=56 time=12 ms
--- 10.0.0.10 ping statistics ---
100 packets transmitted, 27 received, 73% packet loss, time 99999ms
rtt min/avg/max/mdev = 9/132.7/2101/446.6 ms
```

Figura 15. Gráfica de simulación de la Red en PythonViz

Fuente: Autor

5.6. Estándar

Dentro del presente trabajo de investigación se toma en cuenta el uso del estándar 802.11b para brindar las características inalámbricas a la red, en este caso se elige este estándar por encima de los demás debido a que es uno de los estándares que menos potencia ocuparía dentro de la red siendo esto una gran ventaja debido a que los dispositivos a simular serán dispositivos móviles los cuales presentan baterías con un tiempo limitado de vida, por otro lado el este estándar brinda una señal más estable debido al uso de

modulación de espectro ensanchado, con hasta 100m de cobertura y con menos requerimientos para brindar su máxima velocidad de 11Mbps teniendo en cuenta esto su alcance se ve menos degradado al usarse en grandes distancias y obstáculos, permitiendo de esta manera centrarnos en el estudio de los protocolos de encaminamiento lo cual es el objetivo de esta investigación sin la necesidad de preocuparnos a gran escala por deficiencias dentro de cuestiones de capa física ni de enlace de datos a diferencia de otros estándares que a pesar de permitir mayores velocidades pueden así mismo crear deficiencias en cuanto a cobertura y variaciones en las velocidades de transmisión debido al uso de altos valores de potencia lo cual dificultaría evaluar las características de encaminamiento en igualdad de condiciones para todos los nodos ocasionando valores muy aleatorios e incluso fallos que no conciernen al estudio de los protocolos y sus métricas de QoS.

Estándar	Velocidad máxima (Mbps)	Distancia máxima (m)	Señal	Consumo	Interferencias
802.11a	54	120	-	Mayor consumo	-
802.11b	11	150	Estable debido al bajo consumo	Bajo consumo	-
802.11g	54	150	Depende de la potencia	Bajo consumo	Bluetooth
802.11n	150/300	300	Depende de la distancia y potencia	Mayor consumo	-

Tabla 4: Criterios para la elección de la versión del estándar 802.11

Fuente: Autor

Como se puede observar en la tabla 4 el estándar 802.11n puede ser tomado como el que presenta mejores características en comparación a los demás, lamentablemente este estándar puede variar su velocidad dependiendo del nivel de señal que reciba a lo largo de la distancia que permite y la potencia del dispositivo que ocupa es bastante alta, a

diferencia del estándar 802.11b que debido a su bajo consumo permite mantener estabilidad en su velocidad de transmisión a lo largo de los 150 metros de cobertura, siendo este estándar una solución simple para enviar paquetes pequeños que sería el caso de un escenario de emergencia y rescate donde solo se pretende el envío de mensajes que contienen solo texto sin desgastar la batería del dispositivo; por otro lado el estándar 802.11g presenta además de un alto consumo de energía un gran inconveniente el cual es la posibilidad de tener interferencias causadas por dispositivos bluetooth.

5.7. Protocolos de Encaminamiento

Los protocolos de encaminamiento que se simularán en el presente proyecto serán AODV y DSR que se han elegido como objetivo de estudio para determinar cuál de ellos presenta mejores características comparando ambos en varios escenarios con distintas densidades de nodos.

En cada uno de estos protocolos de encaminamiento se considerará algunas de sus métricas de calidad de servicio QoS como parámetros de comparación para poder determinar la eficiencia del protocolo y posteriormente decidir cuál de ellos se considera el mejor en cada escenario.

5.7.1. Métricas

El Principal Objetivo del presente trabajo es el estudio y comparación de las métricas de calidad de servicio en cada uno de los protocolos, por lo tanto, la elección de estas métricas se considera una parte importante en el proceso de investigación, en este caso se ha escogido 3 métricas para trabajar las cuales serán:

5.7.1.1. Throughput

Se ha considerado Throughput como una de las métricas de calidad de servicio a evaluar debido a que es un parámetro considerado relevante en el caso de esta investigación ya que nos permitirá observar la tasa efectiva de datos recibidos en cada uno de los escenarios, teniendo en cuenta que uno de los factores más importantes dentro de una red es la recepción eficaz de datos, evaluar este parámetro será de gran ayuda para determinar dentro de los escenarios cuál de los protocolos nos permite recibir una mayor cantidad de datos de forma eficaz.

5.7.1.2. Delay

El Delay es una métrica que nos permitirá observar dentro de la red el tiempo promedio que le toma a cada paquete de datos en ser recibido por un nodo, este parámetro será de gran ayuda para determinar que protocolo y en que escenarios es más rápida la transmisión y recepción de paquetes, teniendo en cuenta uno o varios nodos transmitiendo al mismo tiempo dentro de la red.

5.7.1.3. Jitter

Por último, se ha escogido como métrica de calidad de servicio a evaluar el Jitter promedio de la red que nos permitirá determinar que tanto varia el retardo en promedio para cada paquete, este parámetro se considera importante debido a que según esta métrica se podrá determinar cuál de los protocolos y escenarios contienen valores más constantes de retardo dentro de la red lo que además podría permitir predecir los tiempos comunes de llegada de un paquete dentro de esa red.

5.8. Estructura del Código

El código que se utilizara para la presente investigación está basado en el ejemplo de código incluido en NS-3 llamado manet-routing-compare.cc, con algunas modificaciones necesarias para cumplir con los objetivos de este trabajo, las partes principales de este código se explicaran brevemente a continuación.

5.8.1. Inclusión de Librerías

En el presente proyecto se deberá importar un total de 9 librerías para permitir utilizar funciones necesarias para la óptima simulación de la red las cuales se pueden observar en la figura 16 (a) para AODV y (b) para DSR.

<pre>#include <fstream> #include <iostream> #include "ns3/core-module.h" #include "ns3/network-module.h" #include "ns3/internet-module.h" #include "ns3/mobility-module.h" #include "ns3/wifi-module.h" #include "ns3/aodv-module.h" #include "ns3/applications-module.h"</pre>	<pre>#include <fstream> #include <iostream> #include "ns3/core-module.h" #include "ns3/network-module.h" #include "ns3/internet-module.h" #include "ns3/mobility-module.h" #include "ns3/wifi-module.h" #include "ns3/dsr-module.h" #include "ns3/applications-module.h"</pre>
(a)	(b)

Figura 16. Librerías incluidas en el código NS-3 para (a) AODV y (b) DSR

Fuente: Autor

5.8.2. Definición de espacios de nombres

Los espacios de nombres son funciones que permiten el uso de palabras reservadas de ciertas librerías dentro del código que se está creando, en la figura 17 se muestra los espacios de nombres usados en el presente trabajo de tesis.

```
using namespace ns3;  
using namespace std;
```

(a)

```
using namespace ns3;  
using namespace dsr;  
using namespace std;
```

(b)

Figura 17. Espacios de nombres necesarios para el código NS-3 de (a) AODV y (b) DSR

Fuente: Autor

5.8.3. Función Principal

La función principal es la primera función que se ejecuta al iniciar la simulación, en ella se debe incluir las funciones y clases que se efectúan dentro del programa y el orden en que lo harán, en este caso se llena los títulos del archivo de salida que se va a crear y se realizará una llamada a la función que inicia los procesos y cálculos de la red.

```
int  
main (int argc, char *argv[])  
{  
    RoutingExperiment experiment;  
    string CSVfileName = experiment.CommandSetup (argc,argv);  
  
    ofstream out (CSVfileName.c_str ());  
    out << "Tiempo,"<<  
        "Throughput," <<  
        "Paquetes Recibidos," <<  
        "Direccion Fuente," <<  
        "Direccion Destino,"<<  
        "Protocolo,"<<  
        "Delay,"<<  
        "Jitter"<<  
    endl;  
    out.close ();  
  
    int nSinks = 5;  
    double txp = 7.5;  
  
    experiment.Run (nSinks, txp, CSVfileName);  
}
```

Figura 18. Líneas de Código de la Función Principal

Fuente: Autor

5.8.4. Diseño de la Topología

La topología de la red será variable dado que se simulará un entorno Ad-Hoc y los nodos se encontrarán en constante movimiento, se deberá instalar características propias de una red común, en este caso se utilizará una red wifi usando el estándar 802.11b ya que es un estándar que permite llegar a grandes distancias manteniendo características de transmisión constantes debido a que no requieren de mucha potencia de transmisión permitiendo dejar un poco de lado las cuestiones de la capa de enlace de datos centrándonos en el objeto de estudio de esta investigación que son los protocolos de encaminamiento de la capa de red a diferencia de otros estándares que a pesar de proveer mayores velocidades y distancias de transmisión, pueden variar debido a que en una red Ad hoc dependerán de la potencia con la cual trabaje el dispositivo móvil, se utiliza una tasa de transmisión de 2Mbps que es la máxima que permite el estándar, con estos datos se creará los nodos y dispositivos contenedores con las características antes mencionadas.

```
string rate ("2048bps");
string phyMode ("DsssRate1Mbps");

NodeContainer adhocNodes;
adhocNodes.Create (nWifis);

WifiHelper wifi;
wifi.SetStandard (WIFI_PHY_STANDARD_80211b);

YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
YansWifiChannelHelper wifiChannel;
wifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss ("ns3::FriisPropagationLossModel");
wifiPhy.SetChannel (wifiChannel.Create ());

WifiMacHelper wifiMac;
wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
                               "DataMode",StringValue (phyMode),
                               "ControlMode",StringValue (phyMode));

wifiPhy.Set ("TxPowerStart",DoubleValue (txp));
wifiPhy.Set ("TxPowerEnd", DoubleValue (txp));

wifiMac.SetType ("ns3::AdhocWifiMac");
NetDeviceContainer adhocDevices = wifi.Install (wifiPhy, wifiMac, adhocNodes);
```

Figura 19. Líneas de Código para crear la topología de la red

Fuente: Autor

5.8.5. Modelo de Movilidad

El modelo de movilidad estará dado dentro de un área rectangular en este caso de 500x500 metros, los nodos se moverán de manera aleatoria siguiendo el modelo

Random Waypoint en el cual se deberá especificar la velocidad máxima de los nodos en m/s y el tiempo máximo de pausa de los mismos, en este caso serán de 20m/s de velocidad y 0.5s de pausa debido a que se va a trabajar en una red MANET conformada por diferentes tipos de usuarios, que pueden ser personas con dispositivos móviles o incluso drones, esto permitirá que el programa simule comunicaciones desde el mejor, hasta el peor de los casos sin adentrarnos en un tema diferente de movilidad como lo serían las redes VANET que necesitan simular un modelo diferente considerando obstáculos, estos valores solo serán referenciales, la simulación tomará valores aleatorios entre los límites mencionados es decir cada nodo podrá simular un tipo diferente de usuario.

```

MobilityHelper mobilityAdhoc;
int64_t streamIndex = 0; // used to get consistent mobility across scenarios

ObjectFactory pos;
pos.SetTypeId ("ns3::RandomRectanglePositionAllocator");
pos.Set ("X", StringValue ("ns3::UniformRandomVariable[Min=0.0|Max=500.0]"));
pos.Set ("Y", StringValue ("ns3::UniformRandomVariable[Min=0.0|Max=500.0]"));

Ptr<PositionAllocator> taPositionAlloc = pos.Create ()->GetObject<PositionAllocator> ();
streamIndex += taPositionAlloc->AssignStreams (streamIndex);

stringstream ssSpeed;
ssSpeed << "ns3::UniformRandomVariable[Min=0.0|Max=" << nodeSpeed << "]";
stringstream ssPause;
ssPause << "ns3::ConstantRandomVariable[Constant=" << nodePause << "]";
mobilityAdhoc.SetMobilityModel ("ns3::RandomWaypointMobilityModel",
                                "Speed", StringValue (ssSpeed.str ()),
                                "Pause", StringValue (ssPause.str ()),
                                "PositionAllocator", PointerValue (taPositionAlloc));
mobilityAdhoc.SetPositionAllocator (taPositionAlloc);
mobilityAdhoc.Install (adHocNodes);
streamIndex += mobilityAdhoc.AssignStreams (adHocNodes, streamIndex);

```

Figura 20. Líneas de Código para añadir movilidad

Fuente: Autor

5.8.6. Características de Encaminamiento

Las características necesarias para el encaminamiento son el tema principal de investigación del presente trabajo en esta parte del código NS-3 se designará un rango de direcciones IP que podrán tomar los nodos dentro de la red, estas direcciones IP serán designadas a los nodos según su número ID en orden, el rango de direcciones que se toma para este trabajo será “10.1.1.0” con máscara “255.255.255.0” es decir que dentro del segmento de red 10.1.1.x tendremos un total de 254 direcciones

disponibles para los nodos, en este caso se utilizará un máximo de 100 nodos y por lo tanto se asignaran direcciones desde 10.1.1.1 hasta 10.1.1.100.

```
AodvHelper aodv;  
  
Ipv4ListRoutingHelper list;  
InternetStackHelper internet;  
list.Add (aodv, 100);  
m_protocolName = "AODV";  
internet.SetRoutingHelper (list);  
internet.Install (ad hocNodes);  
  
NS_LOG_INFO ("assigning ip address");  
  
Ipv4AddressHelper addressAdhoc;  
addressAdhoc.SetBase ("10.1.1.0", "255.255.255.0");  
Ipv4InterfaceContainer ad hocInterfaces;  
ad hocInterfaces = addressAdhoc.Assign (ad hocDevices);
```

Figura 21. Líneas de Código para crear Direcciones IP e instalar Internet y encaminamiento AODV

Fuente: Autor

Además de crear las direcciones IP dentro de las características de red se añadirá el encaminamiento. Los protocolos AODV y DSR se instalarán junto con la capacidad de internet en cada uno de los nodos. En el caso de DSR se instala internet y luego el protocolo por separado como se observa en las Figuras 21 y 22.

```
DsrHelper dsr;  
DsrMainHelper dsrMain;  
Ipv4ListRoutingHelper list;  
InternetStackHelper internet;  
m_protocolName = "DSR";  
internet.Install (ad hocNodes);  
dsrMain.Install (dsr, ad hocNodes);  
  
NS_LOG_INFO ("assigning ip address");  
  
Ipv4AddressHelper addressAdhoc;  
addressAdhoc.SetBase ("10.1.1.0", "255.255.255.0");  
Ipv4InterfaceContainer ad hocInterfaces;  
ad hocInterfaces = addressAdhoc.Assign (ad hocDevices);
```

Figura 22. Líneas de Código para crear Direcciones IP e instalar Internet y encaminamiento DSR

Fuente: Autor

5.8.7. Aplicaciones

La aplicación determinará la manera en que se realiza el envío de paquetes en la red y, el protocolo de transporte que se utilizará (TCP, UDP, RTP) en este trabajo es OnOffAplicattion como el método de envío, este método pertenece al protocolo

UDP, esta aplicación consiste en realizar envíos cada cierto tiempo dentro de la simulación, en este caso se tomara un tiempo de 1 segundo para los envíos.

```
OnOffHelper onoff1 ("ns3::UdpSocketFactory",Address ());
onoff1.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable[Constant=1.0]"));
onoff1.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable[Constant=0.0]"));
```

Figura 23. Líneas de Código para crear la aplicación UDP

Fuente: Autor

5.8.8. Obtención de Métricas

La obtención de datos de métricas se dará a través de cálculos realizados dentro del código fuente a utilizar debido a que varias de las herramientas aplicables a NS-3 no son recomendables para este tipo de protocolos ya que no pueden reconocer el tipo de flujos, especialmente del protocolo DSR, así mismo se eligió como método visual PythonViz debido a que permite observar de mejor manera los flujos.

Los cálculos que se realizarán serán los necesarios para obtener las tres métricas de QoS mencionadas anteriormente: Throughput, Delay y Jitter, las fórmulas que se adecuarán dentro del código son las siguientes:

Para el Throughput se deberá convertir la cantidad de datos a bytes y dividirla para el tiempo de simulación, obteniendo un resultado en kbps

$$Th = \frac{bytes * 8}{Ts * 1000} \quad \text{Ecuación 1}$$

Donde:

bytes = Total de Bytes de cada Paquete

Ts = Tiempo de Simulación

Para el cálculo del Delay o retardo, se obtendrá primeramente los tiempos de llegada de cada paquete, de esta manera se establece el Delay mediante la diferencia entre el tiempo actual y tiempo previo de recepción de paquetes, la fórmula será la siguiente

$$Dly = T_A - T_P \quad \text{Ecuación 2}$$

Donde:

T_A = Tiempo Actual de Simulación o de Recepción

T_P = Tiempo Previo de Simulación o Recepción

En cuanto al cálculos del Jitter se obtendrá a partir de los datos calculados de retardo o Delay, realizando una diferencia entre Delay actual y Delay previo, para

posteriormente a este resultado restarle el valor de Jitter previo y por ultimo dividirlo para 16 para reducción de ruido según el RFC 3393 que trata sobre la métrica de variación de retardo de paquetes IP.

$$Jitt = \frac{[(Dly_A - Dly_P) - Jitt_P]}{16} \quad \text{Ecuación 3}$$

Donde:

Jitt = Jitter

Dly_A = Delay del paquete Actual

Dly_P = Delay del paquete Previo

Jitt_P = Jitter del paquete Previo

Todas estas fórmulas han sido agregadas dentro del código fuente de cada protocolo analizado y la forma de cálculo se puede observar en la figura 24.

```
void
RoutingExperiment::ReceivePacket (Ptr<Socket> socket)
{
    Ptr<Packet> packet;
    Address senderAddress;

    while ((packet = socket->RecvFrom (senderAddress)))
    {
        a=socket->GetNode ()->GetId ();
        bytesTotal += packet->GetSize ();
        packetsReceived += 1;
        delaySum = ((Simulator::Now ().GetSeconds ())-m_time0)*1000;
        m_time0 = (Simulator::Now ().GetSeconds ());
        jitterSum = (delaySum - jitter0);
        m_jitter0 = abs(abs(jitterSum) - m_jitter0) / 16;
        m_jitter=m_jitter0;
        jitter0 = delaySum;
        time=jitter0;
        NS_LOG_UNCOND (PrintReceivedPacket (socket, packet, senderAddress));
    }
}

void
RoutingExperiment::CheckThroughput ()
{
    double kbs = (bytesTotal * 8.0) / (1000*(time/1000));

    bytesTotal = 0;
    jitterSum = 0;
    delaySum = 0;
    packetsReceived = 0;
    Simulator::Schedule (Seconds (0.001), &RoutingExperiment::CheckThroughput, this);
}
```

Figura 24. Líneas de Código para realizar los cálculos de métricas de QoS

Fuente: Autor

5.8.9. Simulador

Por último, se dará al programa características de simulación añadiendo al código definido las 3 líneas de código que se muestran en la figura 25 dentro de las cuales se inicia la ejecución de la simulación con el comando Run, se define el tiempo en que la simulación finalizará su ejecución y por último se utiliza la función “Destroy” para poder borrar los datos de simulación para que la próxima simulación inicie desde cero.

```
Simulator::Stop (Seconds (TotalTime));  
Simulator::Run ();  
Simulator::Destroy ();
```

Figura 25. Líneas de Código para la ejecución de la simulación

Fuente: Autor

6. RESULTADOS

6.1. Escenario 1: Red MANET con 25 Nodos

Dentro del Primer escenario de simulación se tendrá los parámetros especificados en la tabla 3 y se calcularán todos los valores de métricas de calidad de servicio (QoS) mencionadas anteriormente en la metodología.

Parámetros	Valor
Numero de nodos	25
Dimensión del escenario [m]	500x500
Modelo de movilidad	Random Waypoint
Tiempo total de Simulación [s]	100
Velocidad máxima de nodos [m/s]	20
Tiempo de Pausa [s]	0.5
Estándar	802.11b
Tasa de Transmisión de datos [Mbps]	2
Velocidad máxima de datos [Mbps]	11
Potencia de Transmisión [dBm]	7.5
Modulación	DSSS
Encaminamiento	AODV, DSR

Tabla 5: Parámetros del Primer Escenario de Simulación

Fuente: Autor

Todos los valores de la tabla 3 se aplicarán en los protocolos DSR y AOD, dentro del escenario en ambos se tiene una fase de descubrimiento de ruta como se muestra en la figura 26.

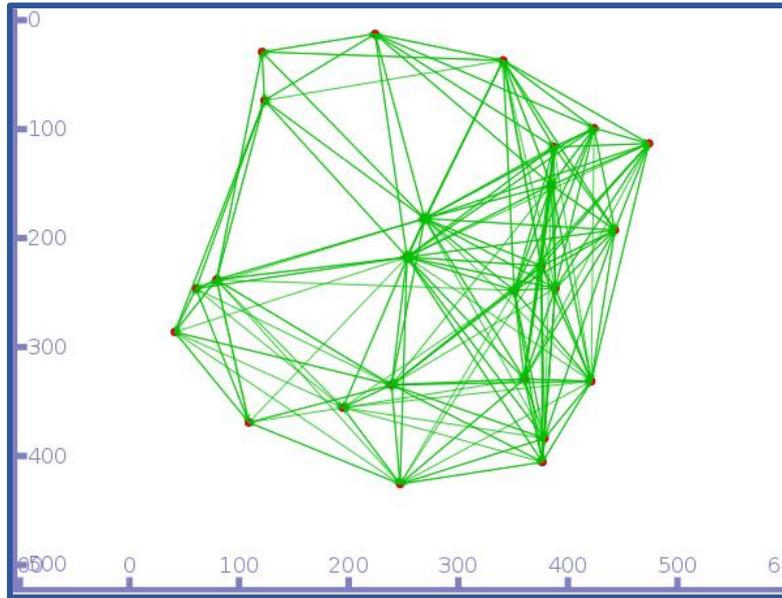


Figura 26. Fase de Descubrimiento de ruta del escenario con 25 nodos

Fuente: Autor

A continuación, tenemos la fase de envío de paquetes que se muestra en la figura 27.

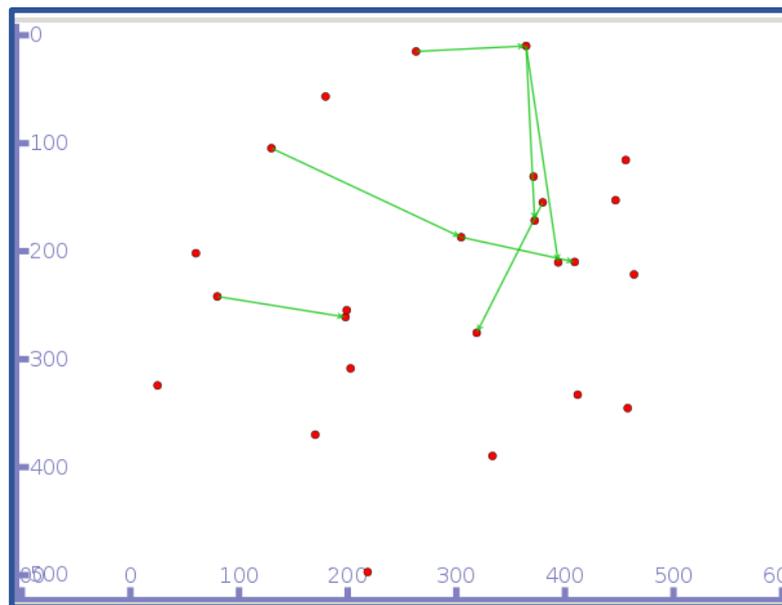


Figura 27. Fase de envío de paquetes del escenario con 25 nodos

Fuente: Autor

La fase de descubrimiento de ruta y de envío de paquetes que se muestran en las figuras 26 y 27 se presentarán en ambos protocolos con la diferencia de que luego de la fase de descubrimiento de AODV, las rutas se mantienen mediante el envío constante de mensajes broadcast llamados Hello, en cambio DSR solo realiza broadcast en la fase de

descubrimiento ya que mantiene la ruta guardando los datos de encaminamiento en la cabecera del paquete enviado.

En cuanto a la fase de envío de paquetes cada protocolo trabaja de manera diferente debido a esto se obtiene los siguientes datos generales mostrados en la tabla 3 sobre las Métricas de QoS mencionadas en la metodología.

Nodo	AODV			DSR		
	Throughput [Kbps]	Delay [ms]	Jitter [ms]	Throughput [Kbps]	Delay [ms]	Jitter [ms]
1	10,821	247,575	1,065	2,582	251,866	1,717
2	2,049	250,823	0,312	2,320	262,474	2,853
3	2,055	285,104	0,929	10,529	279,062	7,232
4	2,072	332,048	1,632	2,045	252,182	0,487
5	2,049	250,534	0,390	2,287	257,061	1,970
6	1,993	953,462	28,782	2,344	257,510	2,599
7	2,042	264,336	0,783	2,042	252,182	0,430
8	2,000	272,589	3,268	2,346	288,848	6,970
9	1,997	432,285	9,815	2,037	259,301	1,484

Tabla 6: Valores de Métricas de QoS del escenario con 25 nodos

Fuente: Autor

A partir de los datos de estas métricas, se procede a la comparación de las mismas.

6.1.1. Throughput

Dentro de los Datos rescatados de la simulación de una red de 25 nodos en se considera que el protocolo con mayor Throughput será el más efectivo en el escenario, de la figura 28 se obtiene que DSR tiende a tener valores de Throughput más altos en 6 de los 9 nodos evaluados, por tanto, en un escenario de 25 nodos podemos decir que DSR es capaz de recibir más datos que AODV de manera efectiva.

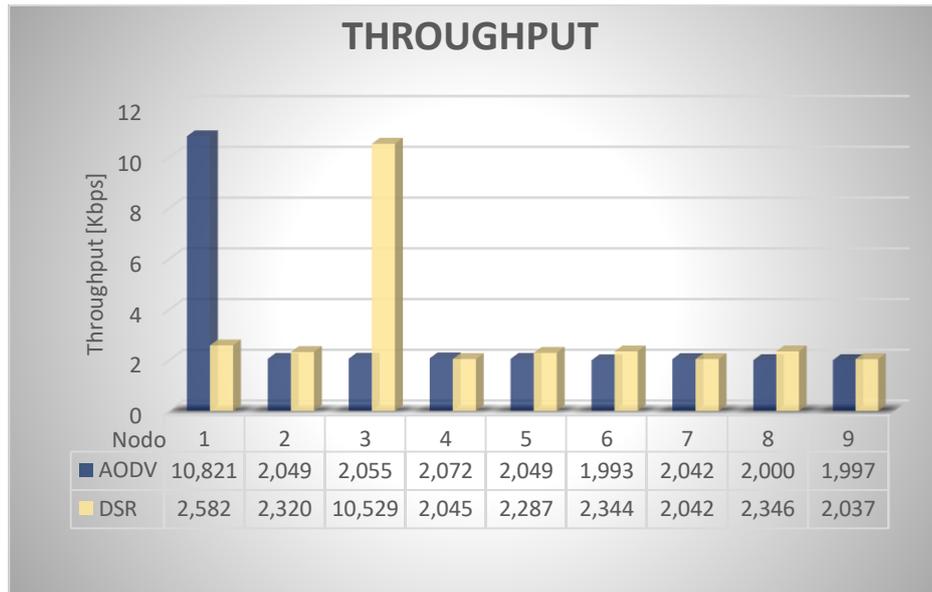


Figura 28. Resultados de Throughput para cada Protocolo en el escenario con 25 nodos

Fuente: Autor

6.1.2. Delay

Dentro de los datos rescatados en este escenario, como se muestra la figura 29, los valores de Delay del protocolo AODV son más altos en 5 de los nodos evaluados a diferencia de DSR, por lo tanto, según lo evaluado aquí, DSR tiene una mayor efectividad en cuanto al tiempo de transmisión ya que sus valores de retardo promedio son más bajos, es decir el envío y recepción de paquetes en DSR se da de manera más rápida que en AODV.

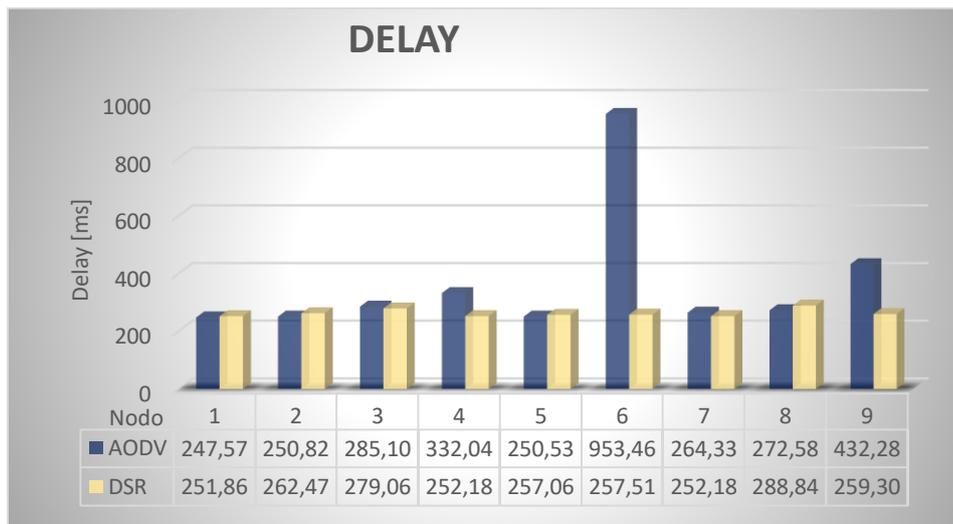


Figura 29. Resultados de Delay para cada Protocolo en el escenario 25 nodos

Fuente: Autor

6.1.3. Jitter

El Jitter es una métrica de QoS que se considera buena entre más baja sea, teniendo en cuenta que es una variación de retardo, es decir el protocolo se considerara mejor mientras más constante sea su retardo, en cuanto a los datos de la figura 30 el protocolo con un Jitter más bajo en 5 de los nodos es AODV a pesar de que en algunos casos el Jitter de AODV supera por mucho al de DSR.

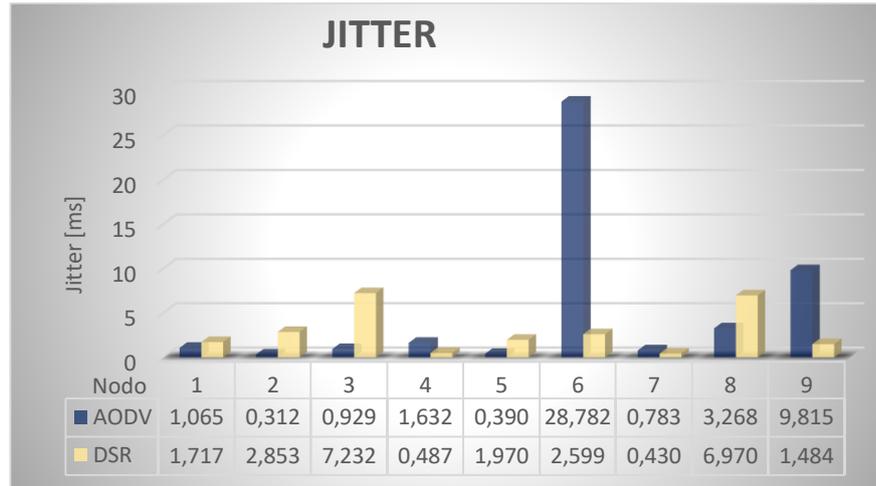


Figura 30. Resultados de Jitter para cada Protocolo en el escenario de 25 nodos

Fuente: Autor

A continuación, en la tabla 5 se muestra de manera general la comparación del comportamiento de los protocolos en cuanto a sus métricas de QoS en el escenario con 25 nodos.

	AODV	DSR
Throughput	Bajo	Alto
Delay	Alto	Bajo
Jitter	Bajo	Alto

Tabla 7: Comparativa de las métricas de QoS de cada Protocolo en el escenario de 25 nodos

Fuente: Autor

Según los datos de la tabla 5 AODV es superado por DSR en dos de las métricas evaluadas las cuales son Throughput y Delay mientras AODV presenta un mejor rendimiento en cuestiones de Jitter cabe mencionar que en las métricas relacionadas al tiempo como lo son el Delay y Jitter el indicador Bajo simboliza un mejor

rendimiento del protocolo a diferencia del Throughput que se considera mejor cuando el indicador es Alto.

6.2. Escenario 2: Red MANET con 50 Nodos

Dentro del segundo escenario se configuran los siguientes parámetros para la simulación.

Parámetros	Valor
Numero de nodos	50
Dimensión del escenario [m]	500x500
Modelo de movilidad	Random Waypoint
Tiempo total de Simulación [s]	100
Velocidad máxima de nodos [m/s]	20
Tiempo de Pausa [s]	0.5
Estándar	802.11b
Tasa de Transmisión de datos [Mbps]	2
Velocidad máxima de datos [Mbps]	11
Potencia de Transmisión [dBm]	7.5
Modulación	DSSS
Encaminamiento	AODV, DSR

Tabla 8: Parámetros del Escenario de Simulación con 50 nodos

Fuente: Autor

En la figura 31 se muestra la red en su fase de descubrimiento de ruta.

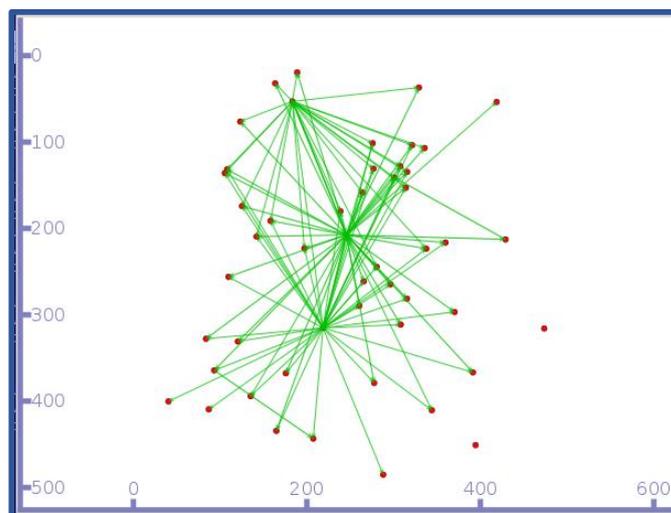


Figura 31. Fase de Descubrimiento de ruta del escenario con 50 nodos

Fuente: Autor

A continuación, se presenta en la figura 32 la fase de envío de paquetes del segundo escenario.

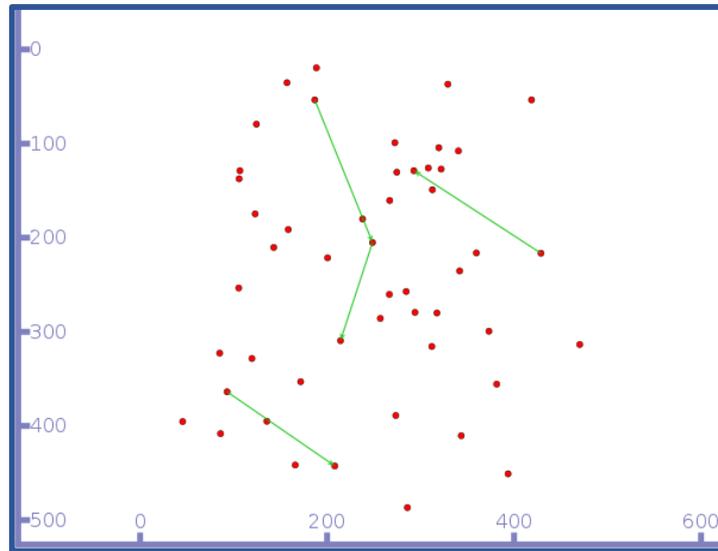


Figura 32. Fase de envío de paquetes del escenario con 50 nodos

Fuente: Autor

En el presente escenario se lograron rescatar los siguientes datos de métricas de QoS presentados en la tabla 7.

Nodo	AODV			DSR		
	Throughput [Kbps]	Delay [ms]	Jitter [ms]	Throughput [Kbps]	Delay [ms]	Jitter [ms]
1	2,648	327,948	11,370	2,200	256,560	2,664
2	9,107	1106,235	105,447	2,287	256,497	3,283
3	8,568	310,056	10,204	7,569	268,761	3,847
4	19,552	290,805	8,895	5,545	275,947	6,260
5	18,781	260,993	5,790	4,830	259,179	4,241
6	17,914	348,974	15,562	2,160	256,103	1,787
7	5,287	306,391	11,132	4,983	253,483	1,896
8	9,768	328,659	10,454	3,009	269,452	3,593
9	3,396	250,534	2,913	2,040	254,114	0,592

Tabla 9: Valores de Métricas de QoS del escenario con 50 nodos

Fuente: Autor

6.2.1. Throughput

Dentro de la figura 33, se puede observar que los valores de Throughput del protocolo AODV tienden a ser mayores en comparación a los de DSR; por lo tanto, a diferencia del escenario con 25 nodos, el protocolo DSR se ha quedado atrás en cuanto a la entrega efectiva de paquetes y el protocolo AODV sobresale en este ambiente con un número mayor de nodos presentando en los 9 casos expuestos un mayor valor de Throughput.

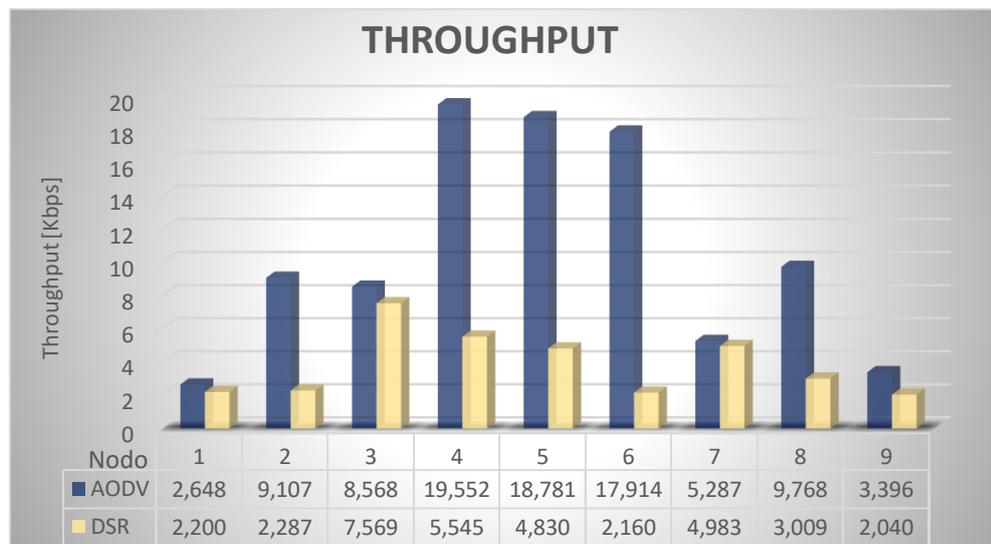


Figura 33. Resultados de Throughput para cada Protocolo en el escenario con 50 nodos.

Fuente: Autor

6.2.2. Delay

Se puede observar en la figura 34 en lo que respecta a valores promedio de Delay el protocolo AODV presenta valores de retardo mayores a los del protocolo DSR por lo tanto en cuanto a Delay los resultados de este escenario concuerdan con los del anterior por lo que se obtiene que DSR es más efectivo en cuanto a rapidez en la entrega de paquetes, esto se da en 8 de los 9 nodos evaluados.

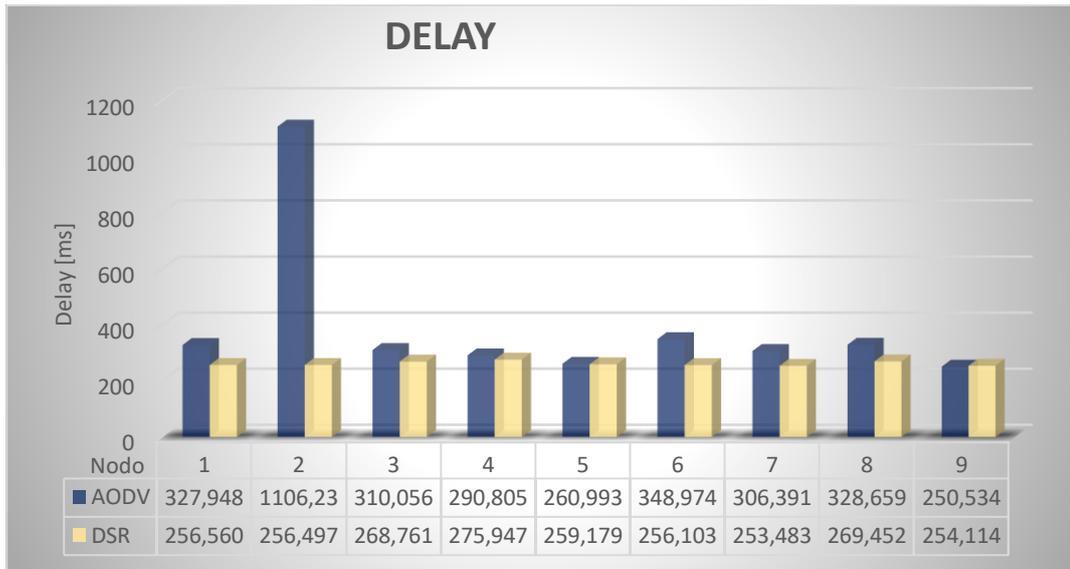


Figura 34. Resultados de Throughput para cada Protocolo en el escenario con 50 nodos.

Fuente: Autor

6.2.3. Jitter

Se puede apreciar en la figura 35 que los valores de Jitter en varios casos son menores en el escenario con protocolo DSR que en el protocolo AODV; por tanto, el protocolo DSR destaca una vez más en cuanto a valores más constantes de retardo en todos los nodos evaluados.

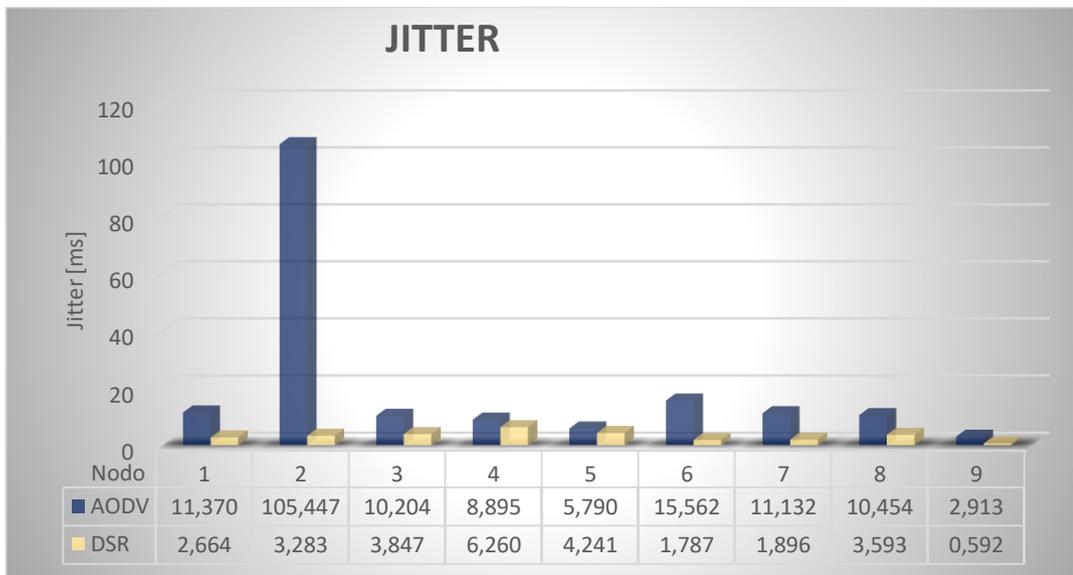


Figura 35. Resultados de Jitter para cada Protocolo en el escenario con 50 nodos.

Fuente: Autor

A continuación, en la tabla 8 se muestra de manera general la comparación del comportamiento de los protocolos en cuanto a sus métricas de QoS en el escenario con 50 nodos.

	AODV	DSR
Throughput	Alto	Bajo
Delay	Alto	Bajo
Jitter	Alto	Bajo

Tabla 10: Comparativa de las métricas de QoS en cada protocolo en el escenario con 50 nodos

Fuente: Autor

Como se puede observar en la tabla 8 el protocolo DSR presenta mejores características en las métricas Delay y Jitter es decir que la entrega de paquetes contiene un retardo Bajo y más constante que AODV mientras el Throughput nos muestra que a diferencia del escenario con 25 nodos AODV muestra mayor efectividad en la entrega de datos.

6.3. Escenario 3: Red MANET con 100 Nodos

En cuanto al tercer escenario se presenta los siguientes parámetros en la tabla 7.

Parámetros	Valor
Numero de nodos	100
Dimensión del escenario [m]	500x500
Modelo de movilidad	Random Waypoint
Tiempo total de Simulación [s]	100
Velocidad máxima de nodos [m/s]	20
Tiempo de Pausa [s]	0.5
Estándar	802.11b
Tasa de Transmisión de datos [Mbps]	2
Velocidad máxima de datos [Mbps]	11
Potencia de Transmisión [dBm]	7.5
Modulación	DSSS
Encaminamiento	AODV, DSR

Tabla 11: Parámetros del Escenario de Simulación con 100 nodos

Fuente: Autor

En la tabla 9 se muestran todos los datos de métricas de QoS recogidas para cada protocolo.

Nodo	AODV			DSR		
	Throughput [Kbps]	Delay [ms]	Jitter [ms]	Throughput [Kbps]	Delay [ms]	Jitter [ms]
1	4,877	281,536	9,147	2,605	266,826	4,086
2	12,010	330,012	17,297	2,766	262,613	3,645
3	6,247	253,672	5,236	2,045	250,559	0,214
4	11,989	335,468	16,377	2,073	252,858	1,591
5	59,158	497,264	41,968	3,863	274,169	6,897
6	11,310	261,283	4,927	2,045	250,559	0,184
7	5,544	324,917	13,185	2,768	256,767	3,564
8	3,667	317,784	11,386	10,093	258,427	3,187
9	4,370	288,053	7,090	3,083	252,440	1,232

Tabla 12: Valores de métricas de QoS del Escenario de Simulación con 100 nodos

Fuente: Autor

Se puede Observar en la Figura 36 la fase de descubrimiento de ruta de los nodos con un total de 100 nodos.

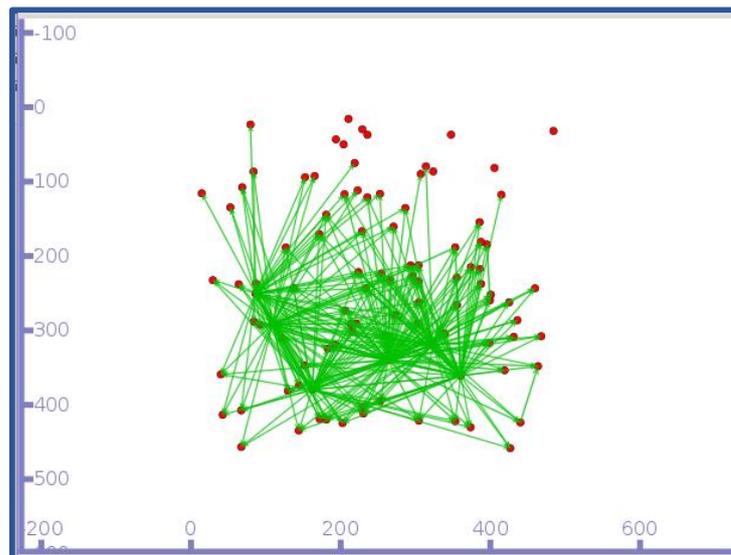


Figura 36. Fase de Descubrimiento de ruta del escenario con 100 nodos

Fuente: Autor

A continuación, se presenta la fase de envío de paquetes, del presente escenario en la figura 37.

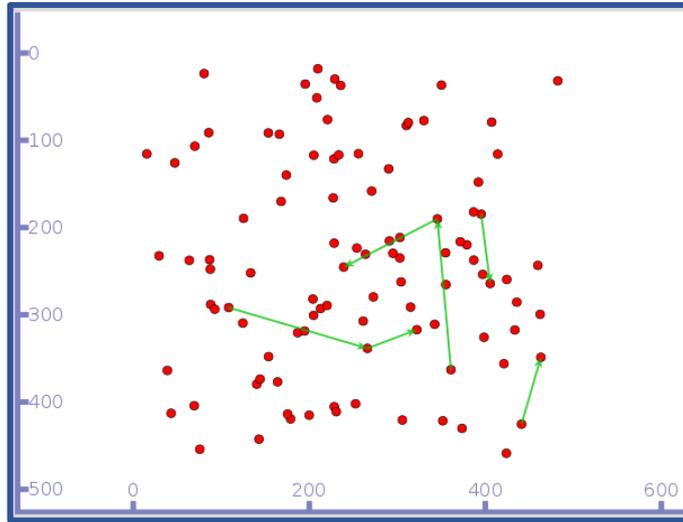


Figura 37. Fase de Envío de Paquetes del escenario con 100 nodos

Fuente: Autor

6.3.1. Throughput

En el caso de este último escenario se puede observar según la figura 38, que la métrica de Throughput en el protocolo AODV tiende a ser mejor que en DSR en 8 de los nodos evaluados teniendo con este, dos escenarios en los que AODV supera a DSR en esta métrica, es decir que la recepción efectiva de paquetes se puede considerar mejor con el Protocolo AODV en los casos con mayor densidad de nodos.

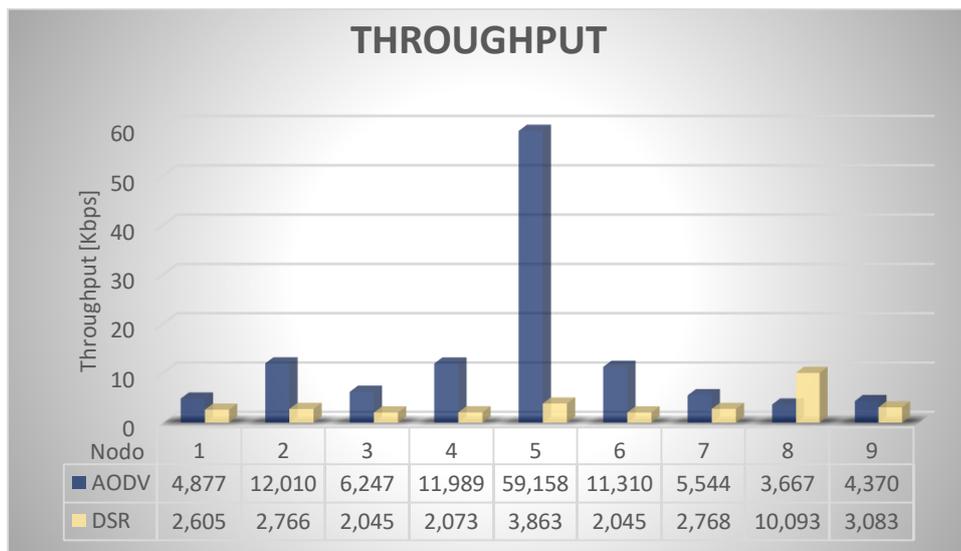


Figura 38. Resultados de Throughput para cada Protocolo en el escenario con 100 nodos

Fuente: Autor

6.3.2. Delay

El Delay evaluado en el escenario con 100 se muestra en la figura 39, en la cual se puede observar claramente que los valores de retardo al igual que en los dos escenarios anteriores tiende a ser menores utilizando DSR, en el caso de este protocolo esta métrica es más baja en los 9 nodos evaluados.

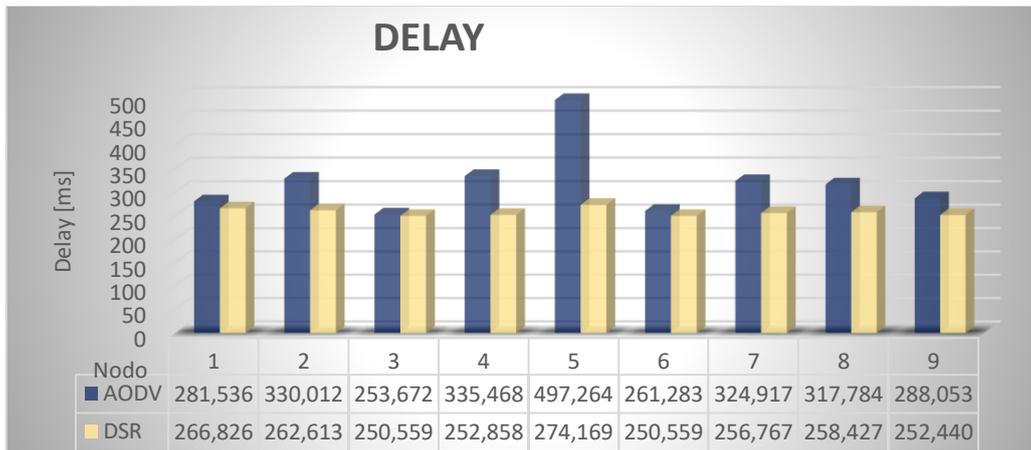


Figura 39. Resultados de Delay para cada Protocolo en el escenario con 100 nodos

Fuente: Autor

6.3.3. Jitter

Por último, la métrica de Jitter según se puede observar en la figura 40 los valores más bajos de variación de retardo son los del protocolo DSR en los 9 nodos, a partir de esto se determina que los valores de Jitter tienden a ser más constantes en DSR a diferencia de AODV que presenta más variación haciendo más difícil estimar tiempos de transmisión de este protocolo.

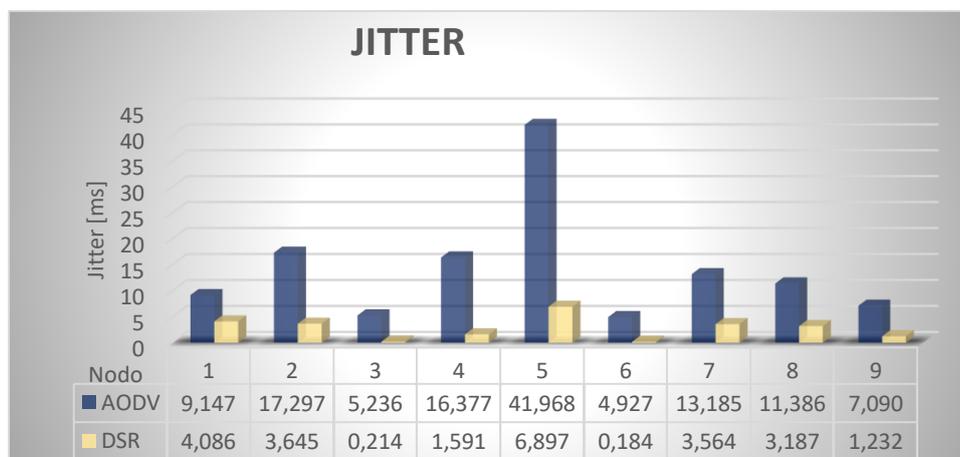


Figura 40. Resultados de Jitter para cada Protocolo en el escenario con 100 nodos.

Fuente: Autor

A continuación, en la tabla 11 se muestra de manera general la comparación del comportamiento de los protocolos en cuanto a sus métricas de QoS en el escenario con 100 nodos.

	AODV	DSR
Throughput	Alto	Bajo
Delay	Alto	Bajo
Jitter	Alto	Bajo

Tabla 13: Comparativa de las métricas de QoS en cada protocolo en el escenario con 100 nodos

Fuente: Autor

En la tabla 11 se muestra que los valores del escenario con 100 nodos presenta a AODV como el protocolo con mayor tasa de entrega efectiva de paquete o Throughput pero su retardo y variación del mismo se encuentran en valores más altos que DSR por lo tanto se puede determinar que mientras mayor sea el número de nodos de la red AODV es más confiable para aplicaciones en las cuales se necesita una mayor fidelidad de datos recibidos mientras DSR se considera mejor para casos en los que se necesite mayor velocidad de entrega sin importar si algunos de los datos enviados no llegan a su destino.

Dentro de los 3 escenarios evaluados se observa varios picos en los valores de las diferentes métricas esto debido a que existen variaciones en los tiempos de transmisión de los paquetes ya que se toma un promedio de aproximadamente 200 paquetes enviados un tiempo de transmisión muy bajo por la cercanía de los nodos o cambios en su movilidad pueden crear grandes cambios dentro de estas 3 métricas ya que sus fórmulas dependen de este tiempo.

Finalmente, se ha podido notar que AODV es el protocolo que destaca en cuanto a Throughput, esto se debe a que DSR mantiene en su cabecera los datos de encaminamiento por lo cual tiende a crear sobrecarga en el paquete en escenarios con gran densidad de nodos aumentando así la posibilidad de que los datos no lleguen completos a su destino, por otro lado el Delay y Jitter tienden a ser mayores en AODV debido a que este al mantener sus rutas enviando periódicamente mensajes Hello resulta utilizar demasiados recursos, además de que este considera que la primera ruta que encuentra es la más corta lo cual no siempre es cierto, pudiendo terminar incluso enviando el paquete por el camino más extenso creando retardos innecesarios y muy variables.

7. DISCUSIÓN

Esta investigación tiene gran importancia debido a que se pudo constatar que en un escenario simulado con 25 nodos y con modelo de movilidad Random Waypoint el envío de paquetes mediante el protocolo DSR presenta mejores características en cuanto a las métricas de QoS: Throughput (Rendimiento) y Delay (Retardo), en cambio AODV muestra un Jitter (Variación de Retardo) mucho más eficiente en esta red con bajo número de usuarios, mientras que en los escenarios con mayor densidad de nodos El protocolo AODV presenta mejoras en cuanto al Throughput es decir la tasa de datos efectiva de transferencia tiende a ser mejor, pero a pesar de presentar mejoras en esta métrica, aún permanece por debajo de DSR en cuanto a Delay y Jitter es decir que en casos en que la red presenta mayor cantidad de nodos AODV se considera un protocolo efectivo para aplicaciones que requieran de entrega eficaz de los datos sin importar el tiempo que tarden estos en llegar, mientras DSR se puede considerar mejor para aplicaciones en las que se requiera mayor velocidad en el envío y recepción de paquetes y se permita cierto porcentaje de pérdidas de datos, todos estos resultados se resumen en la tabla 12.

	25 Nodos		50 Nodos		100 Nodos	
	AODV	DSR	AODV	DSR	AODV	DSR
Throughput	Bajo	Alto	Alto	Bajo	Alto	Bajo
Delay	Alto	Bajo	Alto	Bajo	Alto	Bajo
Jitter	Bajo	Alto	Alto	Bajo	Alto	Bajo

Tabla 14: Comparativa General de las métricas de QoS en cada protocolo

Fuente: Autor

Los resultados antes mencionados se obtuvieron a través de métricas de QoS calculadas para cada paquete enviado por los nodos dentro de la red durante 50 segundos, por lo que se obtiene una extensa lista de datos lo cual dificultaría la evaluación de los mismos, debido a esto se obtuvo un valor promedio de la información de cada usuario para poder realizar las respectivas comparaciones de los protocolos de mejor manera.

Según los escenarios planteados en la investigación de (Murazzo, Rodríguez, & Martínez, 2009) el protocolo AODV supera a DSR en cuanto a la métrica de retardo o Delay a diferencia de la presente investigación en la cual se puede observar que con una mayor densidad de nodos DSR presenta un menor retardo promedio que AODV. En un análisis más a fondo se puede citar a (Rodríguez, 2015) en el cual se puede observar que en simulaciones con el modelo Random Waypoint en escenarios con 25 y 50 nodos con un

total de 100 segundos de simulación se determina que el protocolo con mayor Throughput además de menor Delay y Jitter es AODV en ambos escenarios mientras que en la presente investigación los valores tienen diferencias considerables especialmente en cuanto a Delay y Jitter ya que en los 3 escenarios planteados con un tiempo de simulación de 50 segundos y densidades de 25, 50, y 100 nodos el protocolo DSR mantiene valores más bajos de Delay y Jitter mientras que, en cuanto a Throughput los resultados obtenidos son aproximados, ya que AODV supera a DSR en los escenarios con 50 y 100 nodos.

Dentro de la presente investigación se puede denotar varias limitaciones en el proceso de simulación como lo son la falta de información específica sobre algunas herramientas de software, así mismo varias de las incluidas en el simulador no cumplen con las características necesarias para este tipo de investigación, como: FlowMonitor, V4ping, entre otras, que permiten la visualización de métricas de calidad de servicio a partir de datos de los paquetes obtenidos mediante sondas que se añaden al envío, pero estas son capaces de seguir flujos específicos de paquetes por lo tanto no reconocen algunos otros, en este caso no era posible para las sondas detectar la información del protocolo DSR ya que este incluye en el encabezado del paquete información de encaminamiento, en cambio otras herramientas como Wireshark muestran algunas fallas en la detección de los datos por lo que algunos paquetes DSR se consideran como malformados o incluso no se detectan.

A partir del presente trabajo se puede plantear nuevas investigaciones dentro del campo de Redes como lo podrían ser la investigación de nuevos métodos de recuperación de datos para los protocolos DSR y AODV o la evaluación de otro tipo de métricas considerando una de ellas el consumo de energía de los nodos dentro de la red MANET.

8. CONCLUSIONES

- El software de simulación NS-3 en conjunto con herramientas como PythonViz, facilitan de gran manera la simulación de varios escenarios de redes Ad-Hoc, dando la posibilidad de brindar una gran variedad de características a estas redes para realizar pruebas, como en este caso la evaluación de los protocolos de encaminamiento.
- El modelo de movilidad Random Waypoint representa una herramienta bastante simple y de gran utilidad para la realización de pruebas de protocolos de encaminamiento dentro de redes MANET para la representación del movimiento de varias personas como nodos dentro de un área determinada, a distintas velocidades demostrando a través de la observación del movimiento de los nodos la utilidad que esta herramienta puede presentar para simular el comportamiento de personas dentro de un espacio específico, en este caso llevándolo a un ámbito más aplicable como lo sería un escenario de emergencia y rescate debido a que en estos la movilidad que presentan las personas se da de manera aleatoria sin un patrón específico que seguir sino tratando de realizar comunicación entre todos los individuos dentro del área.
- Los valores de Throughput de las redes simuladas son más eficientes en AODV que en DSR en escenarios en los cuales el número de nodos es mayor, mientras DSR tiende a tener valores más eficientes en el escenario con menor número de nodos esto debido a que mientras mayor es el trayecto de un paquete DSR más grande será su cabecera ya que es ahí donde guarda sus rutas e información de encaminamiento, a diferencia de AODV que mantiene dentro de cada nodo la tabla de encaminamiento permitiendo que sus paquetes sean más pequeños y que la mayoría del contenido del paquete sean los datos.
- Los valores de Delay del protocolo DSR son los más eficaces en los escenarios simulados con 25, 50 y 100 nodos, concluyendo de esta manera que este protocolo puede ser utilizado en escenarios en los que se requiere un menor retardo de entrega de paquetes.
- El Jitter simulado dentro de la red tiende a ser menor en el protocolo DSR que en AODV a pesar del aumento de la densidad de los nodos en cada escenario, en conclusión, los valores de retardo en una red con protocolo DSR son más

constantes que en AODV por lo tanto se puede aproximar de mejor manera los tiempos de entrega en el protocolo DSR que en AODV.

9. RECOMENDACIONES

- Se recomienda Utilizar la versión más estable de la distribución de Linux que se pretenda utilizar ya que de esto puede depender todas las herramientas de NS-3 que se pueda utilizar.
- Los Protocolos Reactivos al mantener diferentes formas de mantenimiento de ruta no siempre se pueden evaluar de la misma forma o con la misma herramienta por lo tanto se recomienda recolectar suficiente información sobre cada protocolo a evaluar.
- Se debe prestar especial atención al hecho de que varias de las herramientas de obtención de métricas que se pueden utilizar dentro de NS-3 trabajan por medio de sondas las cuales no están capacitadas del todo para protocolos de redes Ad-hoc por lo cual protocolos como DSR no son reconocidos por estas sondas y otros protocolos a pesar de ser reconocidos, pueden presentar errores en la obtención de datos.
- Para la representación visual de las simulaciones de ser necesarias se recomienda el uso de la herramienta PythonViz debido a que esta herramienta permite una visualización de mejor manera de los diferentes tipos de envío que presentan los protocolos de encaminamiento.

10. BIBLIOGRAFÍA

- Baluja, W., Ledesma, T. O., & Coya, L. (2016). New solution for the creation of MANETs based on personal devices. *IEEE Latin America Transactions*, 14(3), 1480–1487. <https://doi.org/10.1109/TLA.2016.7459638>
- Bernal, M. A., Cortés, W. S., & Leguizamón, M. Á. (2018). MANET: Advantages, challenges and applications for education. *Sistemas y Telemática*, 15(43), 45–63. <https://doi.org/10.18046/syt.v15i43.2764>
- Casilari, E., & Triviño, A. (2014). Análisis de la estabilidad de modelos de movilidad en simulaciones de redes ad hoc, (May), 9. Recuperado de <https://www.researchgate.net/publication/242484201>
- Chamba Macas, F. V. (2014). Implementación de protocolos de enrutamiento reactivo y proactivos para redes inalámbricas móviles Ad-hoc, 1–125.
- Collaguazo, K. (2017). *Plan De Mejora Continúa Basado En El Estudio De La Red Local Inalámbrica (Wlan) Actual De La Universidad Técnica Del Norte*.
- Coya, L., Ledesma, T., & Baluja, W. (2014). Protocolos de enrutamiento aplicables a redes MANET. *Revista Telemática*, 13(3), 59–74. Recuperado de <http://www.revistatelematica.cujae.edu.cu/index.php/tele/article/view/170>
- Coya Rey, L., & Ledesma Quiñones, T. O. (2014). Selección del protocolo de enrutamiento para la implementación de una manet. *Information Technology Journal*, (November 2014). Recuperado de https://www.researchgate.net/profile/Walter_Baluja_Garcia/publication/281411352_SELECCION_DEL_PROTOCOLO_DE_ENRUTAMIENTO_PARA_LA_IMPLEMENTACION_DE_UNA_MANET/links/55e6261b08aeb1a7ccd65ee/SELECCION-DEL-PROTOCOLO-DE-ENRUTAMIENTO-PARA-LA-IMPLEMENTACION-DE-UN
- Ferro, R., Rodríguez, A., & Gonzalez, R. (2016). *Propuesta de un Protocolo de comunicaciones MANET entre Robots Móviles Usando teoría de Nodos*.
- Flores, T., Iliana, E., Ángel, M., & Carlos, A. R. (2017). Análisis del parámetro Throughput en una red Ad hoc y MANET en el estándar, 3(7), 1–9.
- Gómez, P., Posada, G., Vallejo, M., Pérez, P., Ayde, M., & Velásquez, V. (2014). Evaluación del desempeño del protocolo de enrutamiento AODV para diferentes escenarios de redes de sensores inalámbricos. *Universidad del Norte*, 32(1), 80–101.
- Johnson, D., Hu, Y., & Maltz, U. D. (2007). *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. Request for Comments: 4728 Rice*. Recuperado de <https://www.rfc-editor.org/rfc/pdf/rfc4728.txt.pdf>
- Ledesma, T., Baluja, W., & Coya, L. (2014). Autoconfiguración en MANETs. *TELEMÁTICA*, 13.
- Maria A. Murazzo., Rodríguez, N. R., & Martínez, M. (2008). *Evaluación del retardo de los protocolos de ruteo reactivos para redes Manet*. *Revista Ingeniería Electrónica, Automática y Comunicaciones ISSN: 1815-5928* (Vol. 29). Facultad de Ingeniería Eléctrica, Instituto Superior Politécnico José Antonio Echeverría. Recuperado de <http://rielac.cujae.edu.cu/index.php/riec/article/view/20/6>
- Michel, F. E., Díaz, A., & Calafate, C. T. (2007). *Comparativo entre los protocolos de encaminamiento AODV y DSR*. Recuperado de <http://caii.itmexicali.edu.mx/download/michel07comparativo.pdf>

- Miranda, K., García-nájera, A., & López-jaimés, A. (2017). Algoritmos de Autodespliegue Para Redes de relevos móviles, (December).
- Morán, N. (2016). *Estudio de las Tecnologías Inalámbricas Metro Mesh, WI-MAX y WI-FI para Implementar un ISP para el Sector El Retorno de la Ciudad de Ibarra*. Universidad Técnica del Norte. Recuperado de http://repositorio.utn.edu.ec/bitstream/123456789/1379/1/02_ICA_286_TESIS.pdf
- Murazzo, M. A., Rodríguez, N. R., & Martínez, M. (2009). *Evaluación Y Simulación Del Rendimiento De Los Protocolos De Ruteo Para Manet Bajo Restricciones De QoS*. San Juan. Recuperado de http://sedici.unlp.edu.ar/bitstream/handle/10915/19663/Documento_completo.pdf?sequence=1&isAllowed=y
- Ontaneda, I. (2017). *Medición del Consumo de Energía en un Nodo Sensor Inalámbrico en la Transmisión de Video sobre IPv6 en Aplicaciones de Domótica*. *Journal of Experimental Psychology: General*.
- Patricia, D., & Vallejo, T. (2012). Análisis de métricas de calidad de servicio para la configuración del protocolo CSMA / CA en redes de sensores inalámbricas de área corporal Analysis of quality of service metrics for CSMA / CA protocol configuration in wireless body area networks. *Ingeniería y Desarrollo*, 30(1), 101–124.
- Perkins, C., Belding-Royer, E., & Das, S. (2003). *Ad hoc On-Demand Distance Vector (AODV) Routing. Request for Comments: 3561*. Recuperado de <https://www.rfc-editor.org/rfc/pdf/rfc3561.txt.pdf>
- Quintana, L., & Uribe, G. (2008). Estudio De Las Redes Inalámbricas Conectadas En Modo Ad-Hoc Teniendo En Cuenta El Impacto Que Estas Generan En La Atenuación De La Señal. *Animal Genetics*, 39(5), 561–563.
- Rodríguez, G. (2015). *Análisis y Simulación de protocolos de enrutamiento adecuados en diferentes escenarios para redes AdHoc, mediante la herramienta Ns-3*. Universidad Nacional De Loja, Loja. Recuperado de http://dspace.unl.edu.ec/jspui/bitstream/123456789/11585/1/Rodríguez_Pineda%20Gabriela_Maribel.pdf?fbclid=IwAR04rCIb7pbSE5QkjEWyMTssfk7Ffm69HhkTNpW_xWMKZN_ViSVWAaAg-To
- S, D. A., P, P. L., & R, D. R. (2016). Estudio comparativo sobre simulación de escenarios de protocolos SAODV y AODV, 29(Julio), 88–105.
- Saumelt, M., & Castro, H. (2007). Análisis del Desempeño del Protocolo DSR Bajo Diferentes Modelos de Movilidad. *Epsilon*, 89–104. Recuperado de <http://autoorganizacioncreativa.net/FelipeRivas-AutoorganizacionCreativa-2010.pdf>
- Sepúlveda Gómez, J. A. (2017). Impacto de la Dependencia de Rango Largo en el QoS del Tráfico de Video.
- Subiela, R., & Fernández, A. (2009). Simulación De Protocolos De Encaminamiento En Redes Móviles Ad Hoc Con Ns-2. *Network*, 1–4.
- Tandazo, R. T., Carrión, S. C., Gonzalez, P. L., & Cueva, K. M. (2015). Evaluación de modelos de movilidad para redes móviles en escenarios de emergencia y rescate: AVM, UPM, Pursue, Waypoint y Brownian. *2015 10th Iberian Conference on Information Systems and Technologies, CISTI 2015*, (June 2016). <https://doi.org/10.1109/CISTI.2015.7170358>

11. ANEXOS

ANEXO 1: CÓDIGO FUENTE

```
// Definición de Librerías
#include <fstream>
#include <iostream>
#include "NS-3/core-module.h"
#include "NS-3/network-module.h"
#include "NS-3/internet-module.h"
#include "NS-3/mobility-module.h"
#include "NS-3/wifi-module.h"
#include "NS-3/AODV-module.h"
#include "NS-3/applications-module.h"

//Definición de Espacios de Nombres
using namespace NS-3;
using namespace std;

class RoutingExperiment //definición de la clase Routing Experiment
{
public: //definición de variables y funciones públicas de la clase
Routing Experiment
    RoutingExperiment ();
    void Run (int nSinks, double txp, string CSVfileName);

private: // Definición e Inicialización de Variables y Funciones privadas
de la clase Routing Experiment
    Ptr<Socket> SetupPacketReceive (Ipv4Address addr, Ptr<Node> node);
    void ReceivePacket (Ptr<Socket> socket);
    void CheckThroughput ();

    uint32_t port;
    uint32_t bytesTotal;
    uint32_t packetsReceived;
    string m_CSVfileName;
    int m_nSinks;
    string m_protocolName;
    double m_txp;
    double m_time0=80;
    double m_time1=80;
    double m_time2=80;
    double m_time3=80;
    double m_time4=80;
    double time = 0;
    double m_jitter=0;
    double m_jitter0=0;
```

```

double m_jitter1=0;
double m_jitter2=0;
double m_jitter3=0;
double m_jitter4=0;
double jitter = 0;
double jitter0 = 0;
double jitter1 = 0;
double jitter2 = 0;
double jitter3 = 0;
double jitter4 = 0;
double jitterSum = 0;
double delaySum = 0;
uint32_t m_protocol;
double a;

};

//Definición de la Función Routing Experiment
RoutingExperiment::RoutingExperiment ()
: port (9),
  bytesTotal (0),
  packetsReceived (0),
  m_CSVfileName ("AODV.csv")
{
}
// Definición de la Función para la impresión de datos de recepción de
paquetes, en la terminal
static inline string
PrintReceivedPacket (Ptr<Socket> socket, Ptr<Packet> packet, Address
senderAddress)
{
  ostringstream oss;

  oss << Simulator::Now ().GetSeconds () << " " << socket->GetNode ()-
>GetId ();

  if (InetSocketAddress::IsMatchingType (senderAddress))
  {
    InetSocketAddress addr = InetSocketAddress::ConvertFrom
(senderAddress);
    oss << " received one packet from " << addr.GetIpv4 ();

  }
  else
  {
    oss << " received one packet!";
  }
}

```

```

    return oss.str ();
}
//Definición de la Función ReceivePacket para la obtención de datos y
cálculo de Delay y Jitter
void
RoutingExperiment::ReceivePacket (Ptr<Socket> socket)
{
    Ptr<Packet> packet;
    Address senderAddress;

    while ((packet = socket->RecvFrom (senderAddress)))
    {
        a=socket->GetNode ()->GetId ();
        bytesTotal += packet->GetSize ();
        packetsReceived += 1;
        if (a==0){
            delaySum = ((Simulator::Now ().GetSeconds ())-m_time0)*1000;
            m_time0 = (Simulator::Now ().GetSeconds ());
            jitterSum = (delaySum - jitter0);
            m_jitter0 = abs(abs(jitterSum) - m_jitter0) / 16;
            m_jitter=m_jitter0;
            jitter0 = delaySum;
            time=jitter0;
        } else if (a==1){
            delaySum = ((Simulator::Now ().GetSeconds ())-m_time1)*1000;
            m_time1 = (Simulator::Now ().GetSeconds ());
            jitterSum = (delaySum - jitter1);
            m_jitter1 = abs(abs(jitterSum) - m_jitter1) / 16;
            m_jitter=m_jitter1;
            jitter1 = delaySum;
            time=jitter1;
        } else if (a==2){
            delaySum = ((Simulator::Now ().GetSeconds ())-m_time2)*1000;
            m_time2 = (Simulator::Now ().GetSeconds ());
            jitterSum = (delaySum - jitter2);
            m_jitter2 = abs(abs(jitterSum) - m_jitter2) / 16;
            m_jitter=m_jitter2;
            jitter2 = delaySum;
            time=jitter2;
        } else if (a==3){
            delaySum = ((Simulator::Now ().GetSeconds ())-m_time3)*1000;
            m_time3 = (Simulator::Now ().GetSeconds ());
            jitterSum = (delaySum - jitter3);
            m_jitter3 = abs(abs(jitterSum) - m_jitter3) / 16;
            m_jitter=m_jitter3;
            jitter3 = delaySum;
            time=jitter3;
        }
    }
}

```

```

    } else if (a==4){
        delaySum = ((Simulator::Now ().GetSeconds ())-m_time4)*1000;
        m_time4 = (Simulator::Now ().GetSeconds ());
        jitterSum = (delaySum - jitter4);
        m_jitter4 = abs(abs(jitterSum) - m_jitter4) / 16;
        m_jitter=m_jitter4;
        jitter4 = delaySum;
        time=jitter4;
    }
    NS_LOG_UNCOND (PrintReceivedPacket (socket, packet,
senderAddress));
    }
}
//Definición de la Función CheckThroughput para la obtención de datos y
cálculo de Throughput
void
RoutingExperiment::CheckThroughput ()
{
    double kbs = (bytesTotal * 8.0) / (1000*(time/1000));
    bytesTotal = 0;

    //Impresión de datos en archivo CSV
    ofstream out (m_CSVfileName.c_str (), ios::app);

    out << (Simulator::Now ().GetSeconds ()) << ", "
        << kbs << ", "
        << packetsReceived << ", "
        << "10.1.1." << a+1 << ", "
        << "10.1.1." << a+1 << ", "
        << m_protocolName << ", "
        << delaySum << ", "
        << m_jitter
        << endl;

    out.close ();
    jitterSum = 0;
    delaySum = 0;
    packetsReceived = 0;
    Simulator::Schedule (Seconds (0.001),
&RoutingExperiment::CheckThroughput, this);
}

//Definición de la Función SetupPacketReceive para guardar los datos del
paquete
Ptr<Socket>
RoutingExperiment::SetupPacketReceive (Ipv4Address addr, Ptr<Node> node)
{

```

```

TypeId tid = TypeId::LookupByName ("NS-3::UdpSocketFactory");
Ptr<Socket> sink = Socket::CreateSocket (node, tid);
InetSocketAddress local = InetSocketAddress (addr, port);
sink->Bind (local);
sink->SetRecvCallback (MakeCallback (&RoutingExperiment::ReceivePacket,
this));

    return sink;
}
//Definición de la Función Principal
int
main (int argc, char *argv[])
{
    RoutingExperiment experiment;
    string CSVfileName = "AODV.csv";

    ofstream out (CSVfileName.c_str ());
    out << "Tiempo," <<
    "Throughput," <<
    "Paquetes Recibidos," <<
    "Dirección Fuente," <<
    "Dirección Destino," <<
    "Protocolo," <<
    "Delay," <<
    "Jitter" <<
    endl;
    out.close ();

    int nSinks = 1;
    double txp = 7.5;

    experiment.Run (nSinks, txp, CSVfileName);
}

//Definición de la Función de Ejecución de la Simulación
void
RoutingExperiment::Run (int nSinks, double txp, string CSVfileName)
{
    Packet::EnablePrinting ();
    m_nSinks = nSinks;
    m_txp = txp;
    m_CSVfileName = CSVfileName;

    int nWifis = 50;

    double TotalTime = 100.0;
    string rate ("2048bps");

```

```

string phyMode ("DsssRate11Mbps");
string tr_name ("AODV");
int nodeSpeed = 20; //in m/s
int nodePause = 0.5; //in s
m_protocolName = "protocol";

//Definición de Valores de Configuración de la Red
Config::SetDefault ("NS-3::OnOffApplication::PacketSize", StringValue
("64"));
Config::SetDefault ("NS-3::OnOffApplication::DataRate", StringValue
(rate));
Config::SetDefault ("NS-3::WifiRemoteStationManager::NonUnicastMode",
StringValue (phyMode));

//Creación de Nodos
NodeContainer Ad-HocNodes;
Ad-HocNodes.Create (nWifis);
//Características de Capa 2 de la Red
WifiHelper wifi;
wifi.SetStandard (WIFI_PHY_STANDARD_80211b);
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
YansWifiChannelHelper wifiChannel;
wifiChannel.SetPropagationDelay ("NS-
3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss ("NS-3::FriisPropagationLossModel");
wifiPhy.SetChannel (wifiChannel.Create ());

WifiMacHelper wifiMac;
wifi.SetRemoteStationManager ("NS-3::ConstantRateWifiManager",
"DataMode", StringValue (phyMode),
"ControlMode", StringValue (phyMode));

//Definición de Potencia de Transmisión de la Red
wifiPhy.Set ("TxPowerStart", DoubleValue (txp));
wifiPhy.Set ("TxPowerEnd", DoubleValue (txp));

wifiMac.SetType ("NS-3::Ad-HocWifiMac");
//Creación de Dispositivos e instalación de Características de Capa 2
NetDeviceContainer Ad-HocDevices = wifi.Install (wifiPhy, wifiMac, Ad-
HocNodes);

//Definición del Escenario y modelo de Movilidad
MobilityHelper mobilityAd-Hoc;
int64_t streamIndex = 0;

ObjectFactory pos;
pos.SetTypeId ("NS-3::RandomRectanglePositionAllocator");

```

```

    pos.Set ("X", StringValue ("NS-
3::UniformRandomVariable[Min=0.0|Max=500.0]"));
    pos.Set ("Y", StringValue ("NS-
3::UniformRandomVariable[Min=0.0|Max=500.0]"));
    Ptr<PositionAllocator> taPositionAlloc = pos.Create ()-
>GetObject<PositionAllocator> ();
    streamIndex += taPositionAlloc->AssignStreams (streamIndex);

    stringstream ssSpeed;
    ssSpeed << "NS-3::UniformRandomVariable[Min=0.0|Max=" << nodeSpeed <<
    " ]";
    stringstream ssPause;
    ssPause << "NS-3::ConstantRandomVariable[Constant=" << nodePause <<
    " ]";
    mobilityAd-Hoc.SetMobilityModel ("NS-3::RandomWaypointMobilityModel",
                                     "Speed", StringValue (ssSpeed.str ()),
                                     "Pause", StringValue (ssPause.str ()),
                                     "PositionAllocator", PointerValue
(mobilityAd-Hoc));
    mobilityAd-Hoc.SetPositionAllocator (taPositionAlloc);
    mobilityAd-Hoc.Install (Ad-HocNodes);
    streamIndex += mobilityAd-Hoc.AssignStreams (Ad-HocNodes, streamIndex);
    NS_UNUSED (streamIndex);

    //Definición de Características de capa 3 de la Simulación
(Direccionamiento, Protocolo de Encaminamiento)
    AODVHelper AODV;
    Ipv4ListRoutingHelper list;
    InternetStackHelper internet;
    list.Add (AODV, 100);
    m_protocolName = "AODV";
    internet.SetRoutingHelper (list);
    internet.Install (Ad-HocNodes);

    Ipv4AddressHelper addressAd-Hoc;
    addressAd-Hoc.SetBase ("10.1.1.0", "255.255.255.0");
    Ipv4InterfaceContainer Ad-HocInterfaces;
    Ad-HocInterfaces = addressAd-Hoc.Assign (Ad-HocDevices);

    //Definición de Características de capa de transporte
    OnOffHelper onoff1 ("NS-3::UdpSocketFactory",Address ());
    onoff1.SetAttribute ("OnTime", StringValue ("NS-
3::ConstantRandomVariable[Constant=1.0]"));
    onoff1.SetAttribute ("OffTime", StringValue ("NS-
3::ConstantRandomVariable[Constant=0.0]"));

    //Ejecución de Envíos dentro de la Red

```

```

for (int i = 0; i < nSinks; i++)
{
    Ptr<Socket> sink = SetupPacketReceive (Ad-HocInterfaces.GetAddress
(i), Ad-HocNodes.Get (i));

    AddressValue remoteAddress (InetSocketAddress (Ad-
HocInterfaces.GetAddress (i), port));
    onoff1.SetAttribute ("Remote", remoteAddress);

    Ptr<UniformRandomVariable> var =
CreateObject<UniformRandomVariable> ();
    ApplicationContainer temp = onoff1.Install (Ad-HocNodes.Get (i +
10));
    temp.Start (Seconds (var->GetValue (50.0,51.0)));
    temp.Stop (Seconds (TotalTime));
}
//Llamada a la función CheckThroughput
CheckThroughput ();

//Comandos de Inicio y Finalización de Ejecución de la Simulación
Simulator::Stop (Seconds (TotalTime));
Simulator::Run ();
Simulator::Destroy ();
}

```

ANEXO 2: RESULTADOS GENERALES DE LA SIMULACIÓN

DSR

Tiempo	Throughput	Paquetes Recibidos	Dirección Fuente	Dirección Destino	Protocolo	Delay	Jitter
50,397	1,291	1	10,1,1,14	10,1,1,4	DSR	396,692	24,793
50,632	2,181	1	10,1,1,14	10,1,1,4	DSR	234,741	8,572
50,777	0,659	1	10,1,1,12	10,1,1,2	DSR	776,853	48,553
50,882	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,418
50,942	3,116	1	10,1,1,12	10,1,1,2	DSR	164,287	35,251
51,018	0,503	1	10,1,1,11	10,1,1,1	DSR	1017,290	63,581
51,062	0,482	1	10,1,1,15	10,1,1,5	DSR	1061,250	66,328
51,132	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,026
51,191	2,051	1	10,1,1,12	10,1,1,2	DSR	249,680	3,134
51,261	2,106	1	10,1,1,11	10,1,1,1	DSR	243,106	44,413
51,307	2,089	1	10,1,1,15	10,1,1,5	DSR	245,091	46,865
51,382	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,002
51,446	2,010	1	10,1,1,12	10,1,1,2	DSR	254,786	0,123
51,511	2,050	1	10,1,1,11	10,1,1,1	DSR	249,800	2,357
51,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	2,622
51,562	9,953	1	10,1,1,11	10,1,1,1	DSR	51,443	12,250
51,563	479,725	1	10,1,1,11	10,1,1,1	DSR	1,067	2,383
51,632	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
51,694	2,069	1	10,1,1,12	10,1,1,2	DSR	247,487	0,448
51,738	0,295	1	10,1,1,13	10,1,1,3	DSR	1737,560	108,598
51,739	469,995	1	10,1,1,13	10,1,1,3	DSR	1,089	101,742
51,762	2,576	1	10,1,1,11	10,1,1,1	DSR	198,783	12,208
51,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,164
51,882	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
51,979	1,795	1	10,1,1,12	10,1,1,2	DSR	285,291	2,335
52,012	2,051	1	10,1,1,11	10,1,1,1	DSR	249,660	2,417
52,035	1,732	1	10,1,1,13	10,1,1,3	DSR	295,564	12,046
52,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,010
52,07	14,485	1	10,1,1,13	10,1,1,3	DSR	35,348	15,511
52,12	10,265	1	10,1,1,13	10,1,1,3	DSR	49,879	0,061

52,132	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
52,189	7,367	1	10,1,1,13	10,1,1,3	DSR	69,503	1,223
52,195	86,601	1	10,1,1,13	10,1,1,3	DSR	5,912	3,898
52,217	2,151	1	10,1,1,12	10,1,1,2	DSR	238,067	2,806
52,261	2,055	1	10,1,1,11	10,1,1,1	DSR	249,127	0,118
52,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,001
52,382	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
52,437	2,115	1	10,1,1,13	10,1,1,3	DSR	242,050	14,515
52,442	2,277	1	10,1,1,12	10,1,1,2	DSR	224,809	0,653
52,512	2,036	1	10,1,1,11	10,1,1,1	DSR	251,413	0,136
52,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
52,632	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
52,687	2,050	1	10,1,1,13	10,1,1,3	DSR	249,800	0,423
52,764	2,030	1	10,1,1,11	10,1,1,1	DSR	252,213	0,042
52,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
52,815	1,372	1	10,1,1,12	10,1,1,2	DSR	373,061	9,225
52,882	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
52,936	2,058	1	10,1,1,13	10,1,1,3	DSR	248,821	0,035
52,963	3,453	1	10,1,1,12	10,1,1,2	DSR	148,271	13,473
53,012	2,068	1	10,1,1,11	10,1,1,1	DSR	247,587	0,287
53,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
53,132	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
53,186	2,050	1	10,1,1,13	10,1,1,3	DSR	249,780	0,058
53,192	2,238	1	10,1,1,12	10,1,1,2	DSR	228,808	4,192
53,261	2,058	1	10,1,1,11	10,1,1,1	DSR	248,807	0,058
53,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
53,382	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
53,436	2,049	1	10,1,1,13	10,1,1,3	DSR	249,840	0,000
53,441	2,056	1	10,1,1,12	10,1,1,2	DSR	248,987	0,999
53,512	2,039	1	10,1,1,11	10,1,1,1	DSR	251,093	0,139
53,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
53,632	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
53,686	2,047	1	10,1,1,13	10,1,1,3	DSR	250,160	0,020
53,761	2,056	1	10,1,1,11	10,1,1,1	DSR	249,067	0,118

53,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
53,882	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
53,937	2,041	1	10,1,1,13	10,1,1,3	DSR	250,880	0,044
54,011	2,050	1	10,1,1,11	10,1,1,1	DSR	249,700	0,032
54,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
54,132	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
54,186	2,054	1	10,1,1,13	10,1,1,3	DSR	249,220	0,101
54,262	2,040	1	10,1,1,11	10,1,1,1	DSR	250,973	0,078
54,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
54,382	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
54,436	2,047	1	10,1,1,13	10,1,1,3	DSR	250,080	0,047
54,51	2,057	1	10,1,1,11	10,1,1,1	DSR	248,867	0,127
54,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
54,632	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
54,686	2,046	1	10,1,1,13	10,1,1,3	DSR	250,280	0,010
54,763	2,029	1	10,1,1,11	10,1,1,1	DSR	252,386	0,212
54,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
54,882	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
54,925	0,345	1	10,1,1,12	10,1,1,2	DSR	1484,220	77,140
54,936	2,051	1	10,1,1,13	10,1,1,3	DSR	249,620	0,041
54,954	17,462	1	10,1,1,12	10,1,1,2	DSR	29,322	86,110

Tabla 15: Datos de un Escenario DSR, en los primeros 5 segundos de Simulación

Fuente: Autor

Tiempo	Throughput	Paquetes Recibidos	Dirección Fuente	Dirección Destino	Protocolo	Delay	Jitter
70,019	1,997	1	10,1,1,11	10,1,1,1	DSR	256,336	0,436
70,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
70,132	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
70,184	2,058	1	10,1,1,13	10,1,1,3	DSR	248,807	0,096
70,19	2,047	1	10,1,1,12	10,1,1,2	DSR	250,080	0,003
70,262	2,100	1	10,1,1,11	10,1,1,1	DSR	243,804	0,756
70,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
70,382	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
70,435	2,046	1	10,1,1,13	10,1,1,3	DSR	250,280	0,086

70,441	2,043	1	10,1,1,12	10,1,1,2	DSR	250,673	0,037
70,527	1,938	1	10,1,1,11	10,1,1,1	DSR	264,154	1,225
70,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
70,632	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
70,686	2,036	1	10,1,1,13	10,1,1,3	DSR	251,493	0,070
70,691	2,051	1	10,1,1,12	10,1,1,2	DSR	249,587	0,066
70,764	2,153	1	10,1,1,11	10,1,1,1	DSR	237,785	1,572
70,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
70,882	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
70,935	2,060	1	10,1,1,13	10,1,1,3	DSR	248,547	0,180
70,941	2,048	1	10,1,1,12	10,1,1,2	DSR	249,940	0,018
71,031	1,920	1	10,1,1,11	10,1,1,1	DSR	266,657	1,706
71,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
71,132	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
71,185	2,049	1	10,1,1,13	10,1,1,3	DSR	249,920	0,075
71,191	2,046	1	10,1,1,12	10,1,1,2	DSR	250,260	0,019
71,268	2,163	1	10,1,1,11	10,1,1,1	DSR	236,761	1,762
71,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
71,382	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
71,435	2,047	1	10,1,1,13	10,1,1,3	DSR	250,080	0,005
71,452	1,963	1	10,1,1,12	10,1,1,2	DSR	260,765	0,655
71,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
71,615	1,474	1	10,1,1,11	10,1,1,1	DSR	347,397	6,805
71,632	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
71,684	2,052	1	10,1,1,13	10,1,1,3	DSR	249,540	0,033
71,691	2,143	1	10,1,1,12	10,1,1,2	DSR	238,875	1,327
71,764	3,436	1	10,1,1,11	10,1,1,1	DSR	149,004	11,974
71,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
71,882	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
71,934	2,048	1	10,1,1,13	10,1,1,3	DSR	250,060	0,030
71,942	2,036	1	10,1,1,12	10,1,1,2	DSR	251,453	0,703
72,013	2,058	1	10,1,1,11	10,1,1,1	DSR	248,822	5,490
72,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
72,132	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000

72,184	2,047	1	10,1,1,13	10,1,1,3	DSR	250,160	0,004
72,192	2,051	1	10,1,1,12	10,1,1,2	DSR	249,640	0,069
72,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
72,382	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
72,434	2,048	1	10,1,1,13	10,1,1,3	DSR	250,020	0,008
72,441	2,057	1	10,1,1,12	10,1,1,2	DSR	248,967	0,038
72,545	0,962	1	10,1,1,11	10,1,1,1	DSR	532,003	17,356
72,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
72,632	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
72,684	2,048	1	10,1,1,13	10,1,1,3	DSR	250,040	0,001
72,695	2,016	1	10,1,1,12	10,1,1,2	DSR	253,986	0,311
72,764	2,336	1	10,1,1,11	10,1,1,1	DSR	219,197	18,466
72,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
72,882	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
72,934	2,049	1	10,1,1,13	10,1,1,3	DSR	249,900	0,009
72,942	2,068	1	10,1,1,12	10,1,1,2	DSR	247,527	0,384
73,014	2,047	1	10,1,1,11	10,1,1,1	DSR	250,100	0,777
73,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
73,132	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
73,185	2,045	1	10,1,1,13	10,1,1,3	DSR	250,380	0,029
73,191	2,058	1	10,1,1,12	10,1,1,2	DSR	248,827	0,057
73,264	2,046	1	10,1,1,11	10,1,1,1	DSR	250,260	0,039
73,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
73,382	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
73,436	2,039	1	10,1,1,13	10,1,1,3	DSR	251,153	0,046
73,442	2,037	1	10,1,1,12	10,1,1,2	DSR	251,313	0,152
73,515	2,041	1	10,1,1,11	10,1,1,1	DSR	250,891	0,037
73,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
73,632	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
73,684	2,061	1	10,1,1,13	10,1,1,3	DSR	248,407	0,169
73,692	2,051	1	10,1,1,12	10,1,1,2	DSR	249,600	0,098
73,764	2,057	1	10,1,1,11	10,1,1,1	DSR	248,889	0,123
73,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
73,882	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000

73,935	2,046	1	10,1,1,13	10,1,1,3	DSR	250,200	0,102
73,941	2,058	1	10,1,1,12	10,1,1,2	DSR	248,807	0,043
74,015	2,044	1	10,1,1,11	10,1,1,1	DSR	250,540	0,096
74,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
74,132	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
74,185	2,047	1	10,1,1,13	10,1,1,3	DSR	250,100	0,000
74,191	2,047	1	10,1,1,12	10,1,1,2	DSR	250,180	0,083
74,265	2,043	1	10,1,1,11	10,1,1,1	DSR	250,585	0,003
74,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
74,382	2,048	1	10,1,1,14	10,1,1,4	DSR	250,000	0,000
74,435	2,048	1	10,1,1,13	10,1,1,3	DSR	250,000	0,006
74,441	2,048	1	10,1,1,12	10,1,1,2	DSR	250,020	0,005
74,514	2,056	1	10,1,1,11	10,1,1,1	DSR	249,001	0,099
74,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
74,64	1,986	1	10,1,1,14	10,1,1,4	DSR	257,870	0,492
74,685	2,048	1	10,1,1,13	10,1,1,3	DSR	250,020	0,001
74,69	2,052	1	10,1,1,12	10,1,1,2	DSR	249,540	0,030
74,764	2,054	1	10,1,1,11	10,1,1,1	DSR	249,314	0,013
74,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
74,886	2,076	1	10,1,1,14	10,1,1,4	DSR	246,600	0,674
74,935	2,048	1	10,1,1,13	10,1,1,3	DSR	249,980	0,002
74,941	2,046	1	10,1,1,12	10,1,1,2	DSR	250,300	0,046

Tabla 16: Datos de un Escenario DSR durante 5 segundos a mitad de la Simulación

Fuente: Autor

Tiempo	Throughput	Paquetes Recibidos	Dirección Fuente	Dirección Destino	Protocolo	Delay	Jitter
95,033	2,147	1	10,1,1,11	10,1,1,1	DSR	238,521	2,178
95,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
95,134	2,048	1	10,1,1,14	10,1,1,4	DSR	250,040	0,003
95,186	2,032	1	10,1,1,13	10,1,1,3	DSR	251,913	0,412
95,191	2,049	1	10,1,1,12	10,1,1,2	DSR	249,820	0,209
95,267	2,185	1	10,1,1,11	10,1,1,1	DSR	234,362	0,124
95,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000

95,414	1,825	1	10,1,1,14	10,1,1,4	DSR	280,600	1,910
95,434	2,066	1	10,1,1,13	10,1,1,3	DSR	247,827	0,230
95,44	2,050	1	10,1,1,12	10,1,1,2	DSR	249,720	0,007
95,519	2,029	1	10,1,1,11	10,1,1,1	DSR	252,339	1,116
95,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
95,641	2,255	1	10,1,1,14	10,1,1,4	DSR	227,004	3,230
95,685	2,046	1	10,1,1,13	10,1,1,3	DSR	250,300	0,140
95,691	2,045	1	10,1,1,12	10,1,1,2	DSR	250,420	0,043
95,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
95,884	2,108	1	10,1,1,14	10,1,1,4	DSR	242,836	0,788
95,891	1,378	1	10,1,1,11	10,1,1,1	DSR	371,540	7,380
95,935	2,043	1	10,1,1,13	10,1,1,3	DSR	250,673	0,015
95,94	2,052	1	10,1,1,12	10,1,1,2	DSR	249,520	0,054
96,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
96,119	2,243	1	10,1,1,11	10,1,1,1	DSR	228,256	8,494
96,137	2,024	1	10,1,1,14	10,1,1,4	DSR	252,952	0,583
96,191	2,044	1	10,1,1,12	10,1,1,2	DSR	250,500	0,058
96,194	1,980	1	10,1,1,13	10,1,1,3	DSR	258,568	0,493
96,264	3,530	1	10,1,1,11	10,1,1,1	DSR	145,046	4,670
96,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
96,394	1,994	1	10,1,1,14	10,1,1,4	DSR	256,732	0,200
96,434	2,130	1	10,1,1,13	10,1,1,3	DSR	240,419	1,104
96,442	2,037	1	10,1,1,12	10,1,1,2	DSR	251,373	0,051
96,543	1,834	1	10,1,1,11	10,1,1,1	DSR	279,156	8,090
96,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
96,634	2,132	1	10,1,1,14	10,1,1,4	DSR	240,156	1,024
96,687	2,025	1	10,1,1,13	10,1,1,3	DSR	252,826	0,706
96,692	2,051	1	10,1,1,12	10,1,1,2	DSR	249,660	0,104
96,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
96,885	2,041	1	10,1,1,14	10,1,1,4	DSR	250,896	0,607
96,94	2,061	1	10,1,1,12	10,1,1,2	DSR	248,467	0,068
96,944	1,994	1	10,1,1,13	10,1,1,3	DSR	256,815	0,205
97,057	2,044	1	10,1,1,15	10,1,1,5	DSR	250,474	0,030
97,153	1,910	1	10,1,1,14	10,1,1,4	DSR	268,019	1,032

97,185	2,122	1	10,1,1,13	10,1,1,3	DSR	241,332	0,955
97,191	2,046	1	10,1,1,12	10,1,1,2	DSR	250,200	0,104
97,276	0,699	1	10,1,1,11	10,1,1,1	DSR	732,621	27,836
97,307	2,052	1	10,1,1,15	10,1,1,5	DSR	249,526	0,057
97,385	2,207	1	10,1,1,14	10,1,1,4	DSR	232,001	2,187
97,436	2,040	1	10,1,1,13	10,1,1,3	DSR	251,040	0,547
97,442	2,033	1	10,1,1,12	10,1,1,2	DSR	251,793	0,093
97,511	2,179	1	10,1,1,11	10,1,1,1	DSR	234,946	29,365
97,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,026
97,641	2,000	1	10,1,1,14	10,1,1,4	DSR	256,008	1,364
97,686	2,048	1	10,1,1,13	10,1,1,3	DSR	250,060	0,027
97,693	2,041	1	10,1,1,12	10,1,1,2	DSR	250,893	0,050
97,761	2,043	1	10,1,1,11	10,1,1,1	DSR	250,633	0,855
97,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,002
97,938	2,038	1	10,1,1,13	10,1,1,3	DSR	251,173	0,068
97,941	2,063	1	10,1,1,12	10,1,1,2	DSR	248,167	0,167
97,987	1,480	1	10,1,1,14	10,1,1,4	DSR	345,865	5,531
98,012	2,042	1	10,1,1,11	10,1,1,1	DSR	250,780	0,044
98,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
98,144	3,258	1	10,1,1,14	10,1,1,4	DSR	157,151	11,449
98,191	2,052	1	10,1,1,12	10,1,1,2	DSR	249,487	0,072
98,192	2,010	1	10,1,1,13	10,1,1,3	DSR	254,715	0,217
98,277	1,932	1	10,1,1,11	10,1,1,1	DSR	265,079	0,891
98,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
98,437	2,096	1	10,1,1,13	10,1,1,3	DSR	244,292	0,638
98,44	2,053	1	10,1,1,12	10,1,1,2	DSR	249,440	0,002
98,489	1,481	1	10,1,1,14	10,1,1,4	DSR	345,705	11,069
98,519	2,118	1	10,1,1,11	10,1,1,1	DSR	241,727	1,404
98,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
98,669	2,851	1	10,1,1,14	10,1,1,4	DSR	179,579	9,691
98,688	2,037	1	10,1,1,13	10,1,1,3	DSR	251,333	0,400
98,691	2,043	1	10,1,1,12	10,1,1,2	DSR	250,600	0,072
98,761	2,119	1	10,1,1,11	10,1,1,1	DSR	241,601	0,080
98,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000

98,94	2,034	1	10,1,1,13	10,1,1,3	DSR	251,773	0,002
98,943	2,029	1	10,1,1,12	10,1,1,2	DSR	252,307	0,102
98,985	1,621	1	10,1,1,14	10,1,1,4	DSR	315,829	7,910
99,011	2,049	1	10,1,1,11	10,1,1,1	DSR	249,920	0,515
99,057	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
99,138	3,340	1	10,1,1,14	10,1,1,4	DSR	153,299	9,664
99,185	2,090	1	10,1,1,13	10,1,1,3	DSR	244,941	0,427
99,191	2,070	1	10,1,1,12	10,1,1,2	DSR	247,313	0,306
99,262	2,037	1	10,1,1,11	10,1,1,1	DSR	251,373	0,059
99,307	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
99,435	2,047	1	10,1,1,13	10,1,1,3	DSR	250,180	0,301
99,441	2,046	1	10,1,1,12	10,1,1,2	DSR	250,220	0,163
99,485	1,477	1	10,1,1,14	10,1,1,4	DSR	346,577	11,476
99,512	2,044	1	10,1,1,11	10,1,1,1	DSR	250,520	0,050
99,557	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
99,686	2,040	1	10,1,1,13	10,1,1,3	DSR	250,933	0,028
99,694	2,022	1	10,1,1,12	10,1,1,2	DSR	253,166	0,174
99,763	2,045	1	10,1,1,11	10,1,1,1	DSR	250,347	0,008
99,807	2,048	1	10,1,1,15	10,1,1,5	DSR	250,000	0,000
99,849	1,406	1	10,1,1,14	10,1,1,4	DSR	364,260	0,388
99,937	2,040	1	10,1,1,13	10,1,1,3	DSR	251,013	0,003
99,941	2,076	1	10,1,1,12	10,1,1,2	DSR	246,674	0,395

Tabla 17: Datos de un Escenario DSR en los 5 segundos finales de la Simulación

Fuente: Autor

AODV

Tiempo	Throughput	Paquetes Recibidos	Dirección Fuente	Dirección Destino	Protocolo	Delay	Jitter
50,349	1,468	1	10,1,1,15	10,1,1,5	AODV	348,685	21,793
50,561	0,913	1	10,1,1,14	10,1,1,4	AODV	560,636	35,040
50,585	0,877	1	10,1,1,11	10,1,1,1	AODV	584,098	36,506
50,598	2,057	1	10,1,1,15	10,1,1,5	AODV	248,906	4,874
50,665	0,771	1	10,1,1,13	10,1,1,3	AODV	664,433	41,527
50,783	2,577	1	10,1,1,11	10,1,1,1	AODV	198,703	21,806
50,798	2,159	1	10,1,1,14	10,1,1,4	AODV	237,159	18,027

50,833	3,037	1	10,1,1,13	10,1,1,3	AODV	168,560	28,397
50,836	0,612	1	10,1,1,12	10,1,1,2	AODV	835,963	52,248
50,846	2,063	1	10,1,1,15	10,1,1,5	AODV	248,222	0,262
51,021	2,777	1	10,1,1,12	10,1,1,2	AODV	184,355	37,460
51,031	2,067	1	10,1,1,11	10,1,1,1	AODV	247,704	1,700
51,048	2,048	1	10,1,1,14	10,1,1,4	AODV	250,000	0,324
51,088	2,014	1	10,1,1,13	10,1,1,3	AODV	254,196	3,577
51,096	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,095
51,273	2,028	1	10,1,1,12	10,1,1,2	AODV	252,512	1,919
51,284	2,023	1	10,1,1,11	10,1,1,1	AODV	253,032	0,227
51,298	2,048	1	10,1,1,14	10,1,1,4	AODV	250,000	0,020
51,334	2,079	1	10,1,1,13	10,1,1,3	AODV	246,224	0,275
51,346	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,006
51,525	2,035	1	10,1,1,12	10,1,1,2	AODV	251,620	0,064
51,531	2,073	1	10,1,1,11	10,1,1,1	AODV	247,028	0,361
51,552	2,018	1	10,1,1,14	10,1,1,4	AODV	253,691	0,229
51,598	2,033	1	10,1,1,15	10,1,1,5	AODV	251,852	0,115
51,615	1,823	1	10,1,1,13	10,1,1,3	AODV	280,879	2,149
51,8	2,061	1	10,1,1,14	10,1,1,4	AODV	248,480	0,311
51,85	4,347	2	10,1,1,13	10,1,1,3	AODV	235,553	2,699
51,864	1,538	1	10,1,1,11	10,1,1,1	AODV	332,801	5,338
51,9	1,364	1	10,1,1,12	10,1,1,2	AODV	375,424	7,734
52,031	3,063	1	10,1,1,11	10,1,1,1	AODV	167,139	10,020
52,048	2,066	1	10,1,1,14	10,1,1,4	AODV	247,829	0,021
52,133	1,812	1	10,1,1,13	10,1,1,3	AODV	282,606	2,772
52,177	1,566	1	10,1,1,15	10,1,1,5	AODV	326,936	4,707
52,353	2,899	1	10,1,1,15	10,1,1,5	AODV	176,622	9,100
52,391	1,493	1	10,1,1,14	10,1,1,4	AODV	342,966	5,945
52,403	1,378	1	10,1,1,11	10,1,1,1	AODV	371,514	12,147
52,536	3,848	1	10,1,1,11	10,1,1,1	AODV	133,046	14,145
52,554	3,141	1	10,1,1,14	10,1,1,4	AODV	163,010	10,876
52,601	2,070	1	10,1,1,15	10,1,1,5	AODV	247,360	3,852
52,772	0,801	1	10,1,1,13	10,1,1,3	AODV	639,547	22,136
52,81	2,001	1	10,1,1,14	10,1,1,4	AODV	255,894	5,126

52,811	1,858	1	10,1,1,11	10,1,1,1	AODV	275,638	8,028
52,874	1,872	1	10,1,1,15	10,1,1,5	AODV	273,487	1,392
53,059	2,061	1	10,1,1,14	10,1,1,4	AODV	248,454	0,145
53,07	1,975	1	10,1,1,11	10,1,1,1	AODV	259,197	0,526
53,098	2,289	1	10,1,1,15	10,1,1,5	AODV	223,705	3,024
53,196	0,395	1	10,1,1,12	10,1,1,2	AODV	1295,290	57,009
53,231	1,116	1	10,1,1,13	10,1,1,3	AODV	458,869	9,909
53,262	16,884	1	10,1,1,13	10,1,1,3	AODV	30,324	26,165
53,304	2,083	1	10,1,1,14	10,1,1,4	AODV	245,846	0,154
53,313	4,374	1	10,1,1,12	10,1,1,2	AODV	117,062	70,076
53,332	7,243	1	10,1,1,13	10,1,1,3	AODV	70,688	0,887
53,346	2,061	1	10,1,1,15	10,1,1,5	AODV	248,429	1,356
53,548	2,098	1	10,1,1,14	10,1,1,4	AODV	244,015	0,105
53,549	2,167	1	10,1,1,12	10,1,1,2	AODV	236,322	3,074
53,597	2,046	1	10,1,1,15	10,1,1,5	AODV	250,225	0,028
53,618	1,794	1	10,1,1,13	10,1,1,3	AODV	285,475	13,369
53,803	2,008	1	10,1,1,14	10,1,1,4	AODV	254,959	0,677
53,857	1,966	1	10,1,1,15	10,1,1,5	AODV	260,416	0,635
53,874	1,999	1	10,1,1,13	10,1,1,3	AODV	256,179	0,995
54,048	2,091	1	10,1,1,14	10,1,1,4	AODV	244,856	0,589
54,101	2,253	1	10,1,1,13	10,1,1,3	AODV	227,264	1,745
54,108	2,042	1	10,1,1,15	10,1,1,5	AODV	250,783	0,562
54,298	2,048	1	10,1,1,14	10,1,1,4	AODV	250,000	0,285
54,337	2,175	1	10,1,1,13	10,1,1,3	AODV	235,398	0,399
54,346	2,146	1	10,1,1,15	10,1,1,5	AODV	238,576	0,728
54,549	2,039	1	10,1,1,14	10,1,1,4	AODV	251,153	0,054
54,558	0,507	1	10,1,1,12	10,1,1,2	AODV	1009,150	48,110
54,566	68,846	1	10,1,1,12	10,1,1,2	AODV	7,437	59,601
54,571	103,497	1	10,1,1,12	10,1,1,2	AODV	4,947	3,569
54,583	2,075	1	10,1,1,13	10,1,1,3	AODV	246,785	0,687
54,596	2,047	1	10,1,1,15	10,1,1,5	AODV	250,148	0,678
54,77	2,574	1	10,1,1,12	10,1,1,2	AODV	198,942	11,902
54,822	1,880	1	10,1,1,14	10,1,1,4	AODV	272,413	1,325
54,858	1,955	1	10,1,1,15	10,1,1,5	AODV	261,842	0,689

54,987	1,268	1	10,1,1,13	10,1,1,3	AODV	403,631	9,760
---------------	-------	---	-----------	----------	------	---------	-------

Tabla 18: Datos de un Escenario AODV en los primeros 5 segundos de Simulación

Fuente: Autor

Tiempo	Throughput	Paquetes Recibidos	Dirección Fuente	Dirección Destino	Protocolo	Delay	Jitter
70,096	2,042	1	10,1,1,11	10,1,1,1	AODV	250,741	9,437
70,099	2,253	1	10,1,1,15	10,1,1,5	AODV	227,211	1,609
70,102	1,766	1	10,1,1,14	10,1,1,4	AODV	289,993	7,131
70,27	0,439	1	10,1,1,13	10,1,1,3	AODV	1166,140	56,006
70,272	0,811	1	10,1,1,12	10,1,1,2	AODV	631,141	4,166
70,281	2,766	1	10,1,1,11	10,1,1,1	AODV	185,097	3,513
70,298	2,609	1	10,1,1,14	10,1,1,4	AODV	196,208	5,416
70,338	7,533	1	10,1,1,13	10,1,1,3	AODV	67,969	65,135
70,346	2,071	1	10,1,1,15	10,1,1,5	AODV	247,239	1,151
70,52	2,065	1	10,1,1,12	10,1,1,2	AODV	247,909	23,692
70,531	2,048	1	10,1,1,11	10,1,1,1	AODV	249,960	3,834
70,551	2,022	1	10,1,1,14	10,1,1,4	AODV	253,199	3,223
70,596	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,101
70,798	1,920	1	10,1,1,11	10,1,1,1	AODV	266,689	0,806
70,808	1,999	1	10,1,1,14	10,1,1,4	AODV	256,112	0,019
70,84	1,596	1	10,1,1,12	10,1,1,2	AODV	320,759	3,072
70,963	1,397	1	10,1,1,15	10,1,1,5	AODV	366,461	7,273
71,041	2,556	1	10,1,1,12	10,1,1,2	AODV	200,346	7,334
71,061	1,944	1	10,1,1,11	10,1,1,1	AODV	263,404	0,155
71,074	1,923	1	10,1,1,14	10,1,1,4	AODV	266,307	0,636
71,099	3,757	1	10,1,1,15	10,1,1,5	AODV	136,278	13,932
71,129	0,648	1	10,1,1,13	10,1,1,3	AODV	790,325	41,076
71,281	2,325	1	10,1,1,11	10,1,1,1	AODV	220,184	2,692
71,3	2,270	1	10,1,1,14	10,1,1,4	AODV	225,600	2,504
71,355	2,000	1	10,1,1,15	10,1,1,5	AODV	255,955	6,609
71,368	2,138	1	10,1,1,13	10,1,1,3	AODV	239,432	31,864
71,531	2,048	1	10,1,1,11	10,1,1,1	AODV	250,003	1,695
71,548	2,058	1	10,1,1,14	10,1,1,4	AODV	248,782	1,292
71,583	2,386	1	10,1,1,13	10,1,1,3	AODV	214,612	0,440

71,596	2,122	1	10,1,1,15	10,1,1,5	AODV	241,306	0,502
71,802	2,020	1	10,1,1,14	10,1,1,4	AODV	253,506	0,214
71,834	2,037	1	10,1,1,13	10,1,1,3	AODV	251,384	2,271
71,836	1,682	1	10,1,1,11	10,1,1,1	AODV	304,480	3,299
71,846	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,512
72,031	2,625	1	10,1,1,11	10,1,1,1	AODV	195,080	6,631
72,048	2,077	1	10,1,1,14	10,1,1,4	AODV	246,494	0,425
72,085	2,039	1	10,1,1,13	10,1,1,3	AODV	251,146	0,127
72,096	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,032
72,281	2,045	1	10,1,1,11	10,1,1,1	AODV	250,380	3,042
72,298	2,048	1	10,1,1,14	10,1,1,4	AODV	250,000	0,193
72,331	2,083	1	10,1,1,13	10,1,1,3	AODV	245,791	0,327
72,346	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,002
72,538	1,999	1	10,1,1,11	10,1,1,1	AODV	256,107	0,168
72,57	1,884	1	10,1,1,14	10,1,1,4	AODV	271,784	1,349
72,577	0,333	1	10,1,1,12	10,1,1,2	AODV	1536,200	83,032
72,583	2,034	1	10,1,1,13	10,1,1,3	AODV	251,664	0,347
72,597	2,045	1	10,1,1,15	10,1,1,5	AODV	250,337	0,021
72,769	2,660	1	10,1,1,12	10,1,1,2	AODV	192,476	78,793
72,782	2,098	1	10,1,1,11	10,1,1,1	AODV	243,993	0,747
72,799	2,236	1	10,1,1,14	10,1,1,4	AODV	228,935	2,594
72,831	2,061	1	10,1,1,13	10,1,1,3	AODV	248,376	0,184
72,846	2,051	1	10,1,1,15	10,1,1,5	AODV	249,663	0,041
73,02	2,042	1	10,1,1,12	10,1,1,2	AODV	250,680	1,287
73,031	2,053	1	10,1,1,11	10,1,1,1	AODV	249,420	0,293
73,048	2,054	1	10,1,1,14	10,1,1,4	AODV	249,281	1,110
73,08	2,052	1	10,1,1,13	10,1,1,3	AODV	249,460	0,056
73,13	1,805	1	10,1,1,15	10,1,1,5	AODV	283,735	2,127
73,269	2,055	1	10,1,1,12	10,1,1,2	AODV	249,140	0,016
73,281	2,048	1	10,1,1,11	10,1,1,1	AODV	250,020	0,019
73,298	2,048	1	10,1,1,14	10,1,1,4	AODV	250,000	0,024
73,331	2,044	1	10,1,1,13	10,1,1,3	AODV	250,440	0,058
73,346	2,367	1	10,1,1,15	10,1,1,5	AODV	216,265	4,084
73,52	2,043	1	10,1,1,12	10,1,1,2	AODV	250,620	0,092

73,532	2,044	1	10,1,1,11	10,1,1,1	AODV	250,540	0,031
73,55	2,034	1	10,1,1,14	10,1,1,4	AODV	251,782	0,110
73,581	2,044	1	10,1,1,13	10,1,1,3	AODV	250,460	0,002
73,596	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	1,853
73,77	2,048	1	10,1,1,12	10,1,1,2	AODV	250,040	0,031
73,781	2,049	1	10,1,1,11	10,1,1,1	AODV	249,920	0,037
73,798	2,063	1	10,1,1,14	10,1,1,4	AODV	248,219	0,216
73,831	2,053	1	10,1,1,13	10,1,1,3	AODV	249,400	0,066
73,846	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,116
74,02	2,046	1	10,1,1,12	10,1,1,2	AODV	250,200	0,008
74,031	2,048	1	10,1,1,11	10,1,1,1	AODV	250,040	0,005
74,048	2,048	1	10,1,1,14	10,1,1,4	AODV	250,000	0,098
74,08	2,050	1	10,1,1,13	10,1,1,3	AODV	249,760	0,018
74,098	2,034	1	10,1,1,15	10,1,1,5	AODV	251,671	0,097
74,27	2,048	1	10,1,1,12	10,1,1,2	AODV	249,940	0,016
74,281	2,052	1	10,1,1,11	10,1,1,1	AODV	249,540	0,031
74,298	2,048	1	10,1,1,14	10,1,1,4	AODV	250,000	0,006
74,331	2,044	1	10,1,1,13	10,1,1,3	AODV	250,520	0,046
74,346	2,062	1	10,1,1,15	10,1,1,5	AODV	248,329	0,203
74,519	2,053	1	10,1,1,12	10,1,1,2	AODV	249,380	0,034
74,532	2,044	1	10,1,1,11	10,1,1,1	AODV	250,500	0,058
74,55	2,033	1	10,1,1,14	10,1,1,4	AODV	251,874	0,117
74,581	2,050	1	10,1,1,13	10,1,1,3	AODV	249,760	0,045
74,596	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,092
74,77	2,046	1	10,1,1,12	10,1,1,2	AODV	250,220	0,050
74,782	2,048	1	10,1,1,11	10,1,1,1	AODV	250,000	0,028
74,798	2,063	1	10,1,1,14	10,1,1,4	AODV	248,126	0,227
74,831	2,046	1	10,1,1,13	10,1,1,3	AODV	250,240	0,027
74,846	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,006

Tabla 19: Datos de un Escenario AODV 5 segundos a mitad de la Simulación

Fuente: Autor

Tiempo	Throughput	Paquetes Recibidos	Dirección Fuente	Dirección Destino	Protocolo	Delay	Jitter
95,049	1,869	1	10,1,1,12	10,1,1,2	AODV	273,893	1,059

95,052	2,234	1	10,1,1,14	10,1,1,4	AODV	229,205	2,707
95,077	1,733	1	10,1,1,11	10,1,1,1	AODV	295,516	2,763
95,086	2,021	1	10,1,1,13	10,1,1,3	AODV	253,389	0,059
95,118	1,882	1	10,1,1,15	10,1,1,5	AODV	271,982	27,924
95,27	2,317	1	10,1,1,12	10,1,1,2	AODV	220,935	3,244
95,278	66,532	1	10,1,1,12	10,1,1,2	AODV	7,696	13,125
95,289	2,414	1	10,1,1,11	10,1,1,1	AODV	212,136	5,039
95,299	2,074	1	10,1,1,14	10,1,1,4	AODV	246,844	0,933
95,334	2,061	1	10,1,1,13	10,1,1,3	AODV	248,467	0,304
95,346	2,245	1	10,1,1,15	10,1,1,5	AODV	228,018	1,003
95,519	2,126	1	10,1,1,12	10,1,1,2	AODV	240,869	13,753
95,532	2,109	1	10,1,1,11	10,1,1,1	AODV	242,728	1,597
95,549	2,047	1	10,1,1,14	10,1,1,4	AODV	250,100	0,145
95,583	2,058	1	10,1,1,13	10,1,1,3	AODV	248,784	0,001
95,598	2,035	1	10,1,1,15	10,1,1,5	AODV	251,610	1,412
95,769	2,048	1	10,1,1,12	10,1,1,2	AODV	250,060	0,285
95,782	2,046	1	10,1,1,11	10,1,1,1	AODV	250,280	0,372
95,799	2,048	1	10,1,1,14	10,1,1,4	AODV	250,060	0,007
95,83	2,067	1	10,1,1,13	10,1,1,3	AODV	247,664	0,070
95,846	2,061	1	10,1,1,15	10,1,1,5	AODV	248,390	0,113
96,058	1,856	1	10,1,1,11	10,1,1,1	AODV	275,821	1,573
96,075	1,858	1	10,1,1,14	10,1,1,4	AODV	275,584	1,595
96,089	1,977	1	10,1,1,13	10,1,1,3	AODV	258,948	0,701
96,094	1,573	1	10,1,1,12	10,1,1,2	AODV	325,403	4,691
96,096	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,094
96,269	2,937	1	10,1,1,12	10,1,1,2	AODV	174,357	9,147
96,281	2,292	1	10,1,1,11	10,1,1,1	AODV	223,359	3,181
96,299	2,284	1	10,1,1,14	10,1,1,4	AODV	224,136	3,116
96,333	2,106	1	10,1,1,13	10,1,1,3	AODV	243,148	0,944
96,346	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,006
96,523	2,013	1	10,1,1,12	10,1,1,2	AODV	254,312	4,425
96,531	2,049	1	10,1,1,11	10,1,1,1	AODV	249,860	1,458
96,549	2,046	1	10,1,1,14	10,1,1,4	AODV	250,260	1,438
96,581	2,065	1	10,1,1,13	10,1,1,3	AODV	247,984	0,243

96,596	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,000
96,769	2,083	1	10,1,1,12	10,1,1,2	AODV	245,828	0,254
96,783	2,031	1	10,1,1,11	10,1,1,1	AODV	252,076	0,047
96,799	2,048	1	10,1,1,14	10,1,1,4	AODV	250,000	0,074
96,831	2,048	1	10,1,1,13	10,1,1,3	AODV	250,000	0,111
96,846	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,000
97,019	2,048	1	10,1,1,12	10,1,1,2	AODV	249,960	0,242
97,25	1,136	1	10,1,1,14	10,1,1,4	AODV	450,704	12,539
97,274	2,005	1	10,1,1,12	10,1,1,2	AODV	255,401	0,325
97,298	0,996	1	10,1,1,11	10,1,1,1	AODV	514,204	16,380
97,349	1,019	1	10,1,1,15	10,1,1,5	AODV	502,446	15,778
97,354	103,479	1	10,1,1,15	10,1,1,5	AODV	4,948	30,108
97,396	5,216	1	10,1,1,11	10,1,1,1	AODV	98,167	24,979
97,431	2,833	1	10,1,1,14	10,1,1,4	AODV	180,741	16,089
97,502	0,763	1	10,1,1,13	10,1,1,3	AODV	671,146	26,315
97,524	2,048	1	10,1,1,12	10,1,1,2	AODV	249,968	0,319
97,548	3,368	1	10,1,1,11	10,1,1,1	AODV	152,024	1,805
97,587	5,975	1	10,1,1,13	10,1,1,3	AODV	85,697	34,946
97,596	2,110	1	10,1,1,15	10,1,1,5	AODV	242,606	12,972
97,769	2,093	1	10,1,1,12	10,1,1,2	AODV	244,670	0,311
97,78	2,201	1	10,1,1,11	10,1,1,1	AODV	232,653	4,927
97,83	2,107	1	10,1,1,13	10,1,1,3	AODV	243,037	7,650
97,846	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,349
97,85	1,222	1	10,1,1,14	10,1,1,4	AODV	419,067	13,890
98,019	2,048	1	10,1,1,12	10,1,1,2	AODV	250,000	0,314
98,064	2,392	1	10,1,1,14	10,1,1,4	AODV	214,070	11,944
98,074	1,742	1	10,1,1,11	10,1,1,1	AODV	293,891	3,519
98,081	2,040	1	10,1,1,13	10,1,1,3	AODV	251,021	0,021
98,096	2,048	1	10,1,1,15	10,1,1,5	AODV	250,000	0,022
98,271	2,035	1	10,1,1,12	10,1,1,2	AODV	251,652	0,084
98,299	2,177	1	10,1,1,14	10,1,1,4	AODV	235,158	0,572
98,331	2,054	1	10,1,1,13	10,1,1,3	AODV	249,259	0,109
98,351	2,010	1	10,1,1,15	10,1,1,5	AODV	254,692	0,292
98,519	2,058	1	10,1,1,12	10,1,1,2	AODV	248,748	0,176

98,551	2,031	1	10,1,1,14	10,1,1,4	AODV	252,136	1,025
98,557	1,060	1	10,1,1,11	10,1,1,1	AODV	482,926	11,595
98,596	2,087	1	10,1,1,15	10,1,1,5	AODV	245,308	0,568
98,602	1,888	1	10,1,1,13	10,1,1,3	AODV	271,203	1,365
98,77	2,039	1	10,1,1,12	10,1,1,2	AODV	251,076	0,134
98,784	2,262	1	10,1,1,11	10,1,1,1	AODV	226,395	15,309
98,801	2,051	1	10,1,1,14	10,1,1,4	AODV	249,660	0,091
98,847	2,090	1	10,1,1,13	10,1,1,3	AODV	244,954	1,555
98,848	2,036	1	10,1,1,15	10,1,1,5	AODV	251,469	0,350
99,019	2,059	1	10,1,1,12	10,1,1,2	AODV	248,664	0,142
99,047	1,941	1	10,1,1,11	10,1,1,1	AODV	263,768	1,379
99,055	2,011	1	10,1,1,14	10,1,1,4	AODV	254,638	0,305
99,082	2,177	1	10,1,1,13	10,1,1,3	AODV	235,179	0,514
99,096	2,060	1	10,1,1,15	10,1,1,5	AODV	248,531	0,162
99,269	2,048	1	10,1,1,12	10,1,1,2	AODV	249,980	0,073
99,28	2,197	1	10,1,1,11	10,1,1,1	AODV	233,020	1,836
99,299	2,098	1	10,1,1,14	10,1,1,4	AODV	243,986	0,647
99,465	1,389	1	10,1,1,15	10,1,1,5	AODV	368,621	7,496
99,517	1,178	1	10,1,1,13	10,1,1,3	AODV	434,466	12,423
99,707	2,114	1	10,1,1,15	10,1,1,5	AODV	242,194	7,433
99,831	0,912	1	10,1,1,12	10,1,1,2	AODV	561,480	19,464
99,856	3,435	1	10,1,1,15	10,1,1,5	AODV	149,065	5,356
99,907	6,663	1	10,1,1,12	10,1,1,2	AODV	76,839	29,074
99,914	1,289	1	10,1,1,13	10,1,1,3	AODV	397,324	1,545
99,942	18,008	1	10,1,1,13	10,1,1,3	AODV	28,432	22,959

Tabla 20: Datos de un Escenario AODV en 5 segundos al final de la Simulación

Fuente: Autor

ANEXO 3: ANÁLISIS DE PLAGIO EN URKUND

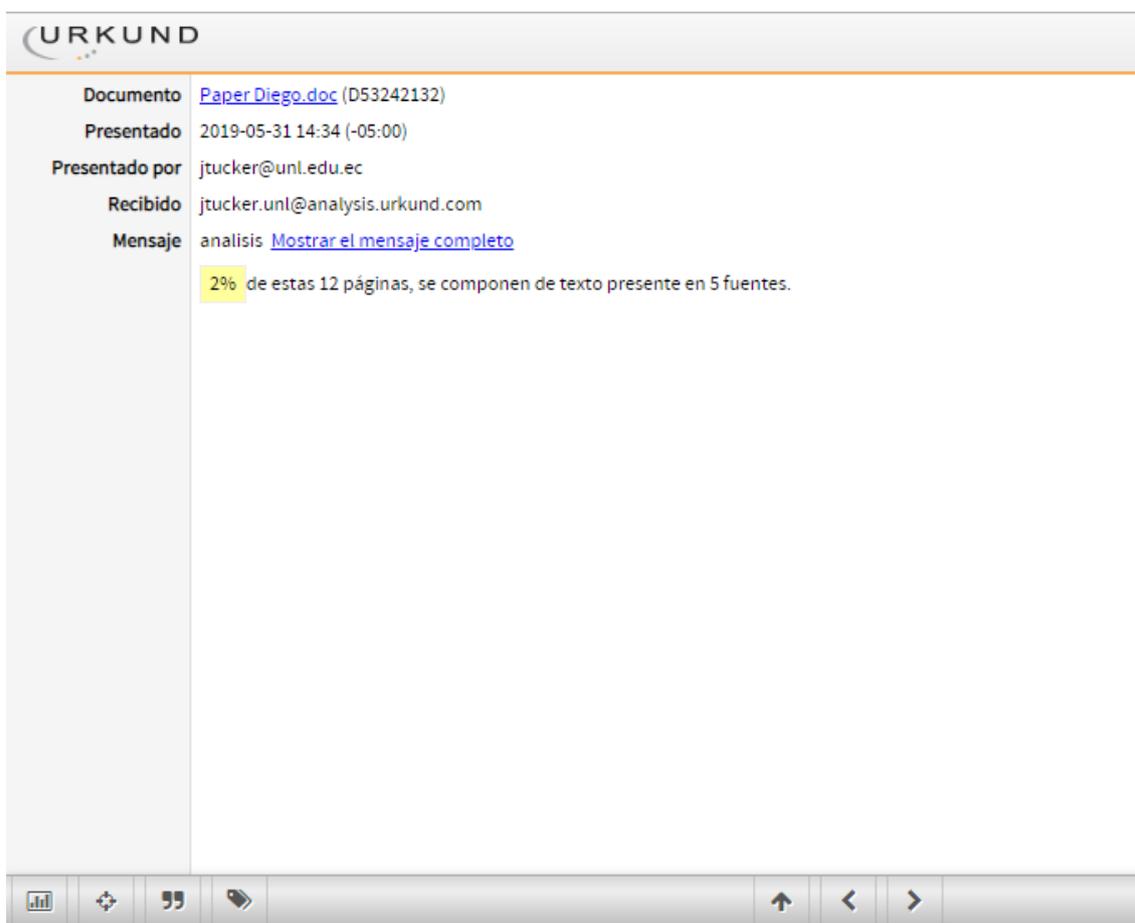


Figura 41. Resultado del Análisis de Plagio

Fuente: Urkund