



**UNIVERSIDAD
NACIONAL DE
LOJA**

PFC-N-CIS-019



Área de la Energía, las Industrias y los Recursos Naturales no Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

**“Análisis y Simulación de protocolos de
enrutamiento adecuados en diferentes
escenarios para redes AdHoc, mediante
la herramienta Ns-3”**

*“Tesis previa la obtención del título
de Ingeniera en Sistemas”*

Autor:

Rodríguez-Pineda, Gabriela-Maribel

Director:

Ing. Mgs. Padilla-Encalada, Alex-Vinicio

Loja-Ecuador

2015

Certificación del Director

Ing. Mgs. Padilla-Encalada, Alex-Vinicio

DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS

CERTIFICA:

Que la señora Gabriela Maribel Rodríguez Pineda, egresada de la carrera de Ingeniería en Sistemas y cuyo tema versa sobre “ANÁLISIS Y SIMULACIÓN DE PROTOCOLOS DE ENRUTAMIENTO ADECUADOS EN DIFERENTES ESCENARIOS PARA REDES ADHOC, MEDIANTE LA HERRAMIENTA NS-3”, ha sido monitoreado, revisado y orientado bajo mi asesoramiento, con pertinencia y con la rigurosidad científica que el trabajo de investigación debe cumplir, por lo cual autorizo su presentación y sustentación.

Loja, 13 de Mayo de 2015.



Ing. Mgs. Padilla-Encalada, Alex-Vinicio

Director del Proyecto de Titulación

Autoría

Yo **GABRIELA MARIBEL RODRIGUEZ PINEDA**, declaro ser autora del presente proyecto de titulación 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:



Cédula: 1104566805

Fecha: 27 de Julio del 2015

**Carta de autorización de Tesis por parte de la autora,
para la consulta, reproducción parcial o total y
publicación electrónica del texto completo.**

Yo **GABRIELA MARIBEL RODRIGUEZ PINEDA**, declaro ser autora de la tesis titulada: **“ANÁLISIS Y SIMULACIÓN DE PROTOCOLOS DE ENRUTAMIENTO ADECUADOS EN DIFERENTES ESCENARIOS PARA REDES ADHOC, MEDIANTE LA HERRAMIENTA NS-3”**, como requisito para optar el grado de: **INGENIERA EN SISTEMAS**; 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 veintisiete días del mes de Julio del dos mil quince.

Firma:



Autor: Gabriela Maribel Rodríguez Pineda

Dirección: Loja (Sauces Norte)

Teléfono: (07) 2136024

Celular: 0989224433

Correo Electrónico: gmrodriguezp@unl.edu.ec - 6abyta.rodriguez@gmail.com

Director de Tesis: Ing. Alex-Vinicio, Padilla-Encalada, Mgs.

Tribunal de Grado: 1 Ing. Walter-Rodrigo, Tene-Ríos, Mg. sc.

2 Ing. Auliria-Virginia, Torres-Ontaneda, Mg. sc.

3 Ing. Marco-Augusto, Ocampo-Carpio, Mg. sc.

Agradecimiento

A mis padres que me ofrecen sus esfuerzos diarios y sus enseñanzas, que están arraigadas en mi vida y que me guían siempre a brindar lo mejor de mí a los demás, mediante una meta cumplida, esta vez mi carrera profesional.

A mi hija, mi impulso, mi alegría, por su sacrificio diario a mi lado, con el que he alcanzado este logro en mi carrera profesional.

A mis hermanas y hermano, por su apoyo incondicional, en estos años de vida universitaria, sobre todo a mi hermana mayor por ser guía y ejemplo en mi vida y carrera profesional.

Al ing. Alex Padilla por su experiencia, asesoría y rigurosidad científica, con la que pude concretar mi proyecto de titulación y con ello el proceso de titulación.

A la planta de docentes que con sus conocimientos y experiencias impartidos en los años de vida universitaria me brindaron enseñanzas académicas, valores y experiencias de vida, y que gracias a su conocimiento hoy puedo llevar a cabo con profesionalismo mi proyecto de titulación.

Dedicatoria

A mis padres, Luz y Saúl

A mi hija, Emily

A mis hermanas y hermano, Soledad-Luisa y David

Cesión de Derechos

Gabriela Maribel Rodríguez Pineda, autora del presente proyecto de titulación, autoriza a la Universidad Nacional de Loja, al Área de la Energía, las Industrias y los Recursos Naturales No Renovables y por consecuente a la carrera de Ingeniería en Sistemas, hacer uso total o parcial del contenido del mismo, en medida de lo que se vea conveniente.

a. Título

“Análisis y Simulación de Protocolos de enrutamiento adecuados en diferentes escenarios para redes AdHoc, mediante la herramienta Ns-3”

b. Resumen

Para la realización del presente proyecto de titulación se siguieron las etapas de: identificación de problema, planteamiento de objetivos, estado del arte, conjuntamente con el listado de tareas definidas en la metodología, que permitieron ofrecer una solución al problema planteado mediante fases.

Se realizó un análisis a fondo de la información recolectada sobre las redes Ad-Hoc, haciendo uso de los métodos deductivo e inductivo, en donde se pudo identificar sus principales inconvenientes, entre los cuales está el de los llamados nodos ocultos, que se asocia principalmente al inadecuado enrutamiento, razón por lo que se inició el estudio del uso de protocolos acorde a estas redes, con el uso de la herramienta de Simulación de Redes Ns-3 (OpenSource), una de las mejores alternativas para redes Ad-Hoc, y para la gestión de la misma en un entorno simulado.

Se procedió con la ejecución de las fases que plantea la metodología, partiendo desde un análisis de las redes Ad-Hoc y sus protocolos, continuando con la especificación de dos escenarios y cuatro protocolos los que se realizaron en la etapa de simulación que involucra la codificación en lenguaje de programación (c++), lo que dio como resultado un conjunto de métricas que sirvieron para evaluar el desempeño de la red de acuerdo al escenario y al protocolo utilizado. Luego de esto se procedió a realizar una comparación de las métricas entre los cuatro protocolos utilizados (AODV, DSR, OLSR, DSDV), en los dos escenarios y a realizar una propuesta en base a este estudio, orientada a la estabilización de las redes Ad-Hoc, basándonos en mejorar en este caso el enrutamiento, con el uso de protocolos adecuados.

Summary

For the accomplishment of the present certification Project the following steps were followed: first identification of the problem, objectives approach, state of the art, along with the list of tasks that define the methodology, which allowed to offer a solution to the problems through phases.

A deep analysis of the information collected related to the Ad, hoc networks took place, using the deductive and inductive methods, where it was possible to identify the main drawbacks, among them there is the so- called hidden nodes, which is mainly associated with the wrong type of routing, and therefore the study of protocol usage was initiated according to these networks, using the network simulation tools spanish words “Simulación de Redes” Ns-3 (open source), one of the best alternatives for Ad-Hoc networks and to manage them in a simulated environment.

We proceeded with the execution of the phases which is totally planted by the methodology, starting from a deep analysis of the Ad-Hoc networks and it's protocols, continuing with the specification of two stages and four protocols, which they were performed on the simulation stage which also involves a coding programming language (c++), which resulted a set of metrics that were used to evaluate the performance of the networks according to the stage and the protocols that were used. After this we proceeded to perform a comparison of the metrics and the four protocols used key words: (AODV, DSR, OLSR, DSDV), on two stages and in order to make a proposal based on this study, aimed to the stabilization of Ad-Hoc networks, based on improving routing in this case, using the appropriate protocols.

Índice de Contenidos

Certificación del Director	i
Autoría	ii
Carta de autorización de Tesis por parte de la autora	iii
Agradecimiento	iv
Dedicatoria	v
Cesión de Derechos	vi
a. Título	vii
b. Resumen	viii
Summary	ix
Índice de Contenidos	xi
Índice de Figuras	xvi
Índice de Tablas	xx
c. Introducción	22
d. Revisión de Literatura	23
1. Redes	23
1.1 Definición de Red	23
1.2 Clasificación de Redes de acuerdo a su área de Cobertura	23
1.3 Redes Inalámbricas y su clasificación	25
1.4 Tecnologías WLAN IEEE 802.11	26
1.5 Modelo OSI	29
1.5.1 Capa Física	29
1.5.2 Capa de Enlace	30
1.5.2.1 Subcapa de Control Lógico de enlace (Logical Link LLC) ..	30
1.5.2.2 Subcapa de Control de Acceso al Medio (Media Access Control MAC) ..	30
2. Inicios de las redes Ad-Hoc	30

2.1 Definición de Red Ad-Hoc	31
2.2 Estructura de las Redes Ad-Hoc	32
2.3 Características de las redes Ad-Hoc	32
2.4 Tipos de Redes Ad-Hoc	34
2.5 Campos de Aplicación de las Redes Ad-Hoc	34
2.6 Protocolos de Enrutamiento de las Redes Ad-Hoc.....	35
2.6.1 Clasificación de Protocolos en redes Ad-Hoc	35
2.6.1.1 AODV (Ad-hoc On Demand Vector)	36
2.6.1.2 DSDV (Destination Sequenced Distance Vector)	36
2.6.1.3 DSR (Dynamic Source Routing)	37
2.6.1.4 TORA (Temporally Ordered Routing ALgorithm)	37
2.6.1.5 OLSR (Optimized link State Routing Protocol)	38
2.6.1.6 ZRP (Zone Routing Protocol)	38
2.7 Comparación entre Arquitectura regular TCP/IP y AdHoc	39
2.8 Modos de Transmisión en redes Ad-Hoc.....	39
2.8.1 Unicast.....	40
2.8.2 Multicast.....	40
2.9 Tramas en Redes AdHoc	40
2.9.1 Tramas de manejo de Conexión.....	40
2.9.2 Tramas de Control.....	42
2.9.3 Tramas de datos	42
2.10 Forma de Funcionamiento de las redes Ad-Hoc.....	43
2.11 Tipos de escenarios para Redes Adhoc	44
3. Network Simulator NS-3.....	44
3.1 Comparativa entre Simuladores de Red.....	45
3.2 Métricas de Rendimiento.....	48
3.2.1 Throughput.....	48

3.2.2 Tasa promedio de entrega	49
3.2.3 Retardo promedio de extremo a extremo	49
e. Materiales y Métodos.....	50
1. Métodos	50
2. Técnicas.....	50
3. Metodología	51
f. Resultados.....	53
Fase 1: Análisis de las Redes Ad-Hoc	53
1.1 Estadísticas.....	53
1.2 Principales causas de inestabilidad detectadas por enrutamiento en redes Ad-Hoc.	58
Fase 2: Modelado de la Topología	59
2.1 Selección de Escenarios	59
2.2 Selección de Protocolos.....	60
2.3 Evaluación de Herramientas para Simulación	62
2.4 Características principales de ns3.....	64
2.5 Tecnologías permitidas en ns3.....	65
Fase 3: Simulación y Evaluación.....	65
3.1 Simulación	65
3.2 Especificación de Software	66
3.3 Especificación de Hardware	66
3.4 Selección de equipo	66
3.5 Costos.....	67
3.6 Beneficios a obtener.....	67
3.7 Módulos de ns3.....	68
3.8 Estructura de un script en NS3.....	69
3.9 Análisis en NS-3.....	72

3.9.1 Información analizada	72
3.9.2 Datos recogidos	73
3.10 Instalación y Configuración de NS3.....	73
3.11 Codificación de la Topología	83
3.12 Obtención de métricas y animación de la Red.....	87
3.13 Análisis de las métricas obtenidas.....	98
3.13.1 Escenario de prueba uno	99
3.13.1.1 Time First Tx Packets.....	99
3.13.1.2 Time First Rx Packets	101
3.13.1.3 Time Last Tx Packets	103
3.13.1.4 Time Last Rx Packets.....	104
3.13.1.5 Delay.....	106
3.13.1.6 Jitter	107
3.13.1.7 Tx Bytes	109
3.13.1.8 Rx Bytes.....	110
3.13.1.9 Tx Packets	112
3.13.1.10 Rx Packets	113
3.13.1.11 Lost Packets.....	115
3.13.1.12 Cálculo de Troughput	116
3.13.1.13 Cálculo de Tasa de Entrega	117
3.13.2 Escenario de prueba dos	118
3.13.2.1 Time First Tx Packets.....	118
3.13.2.2 Time First Rx Packets	119
3.13.2.3 Time Last Tx Packets	121
3.13.2.4 Time Last Rx Packets.....	122
3.13.2.5 Delay.....	124
3.13.2.6 Jitter	125
3.13.2.7 TxBytes	127
3.13.2.8 RxBytes.....	128
3.13.2.9 TxPackets	130
3.13.2.10 RxPackets	131
3.13.2.11 Lost Packets.....	133
3.13.2.12 Cálculo de Troughput	134

3.13.2.13 Cálculo de Tasa de Entrega	135
3.13.1 Propuesta del mejor protocolo para el primer escenario.....	136
3.13.2 Propuesta del mejor protocolo para el segundo escenario	137
g. Discusión	139
1. Desarrollo de la Propuesta Alternativa	139
2. Valoración Técnica Económica Ambiental.....	140
h. Conclusiones	142
i. Recomendaciones	144
j. Bibliografía.....	145
k. Anexos.....	149
Anexo 1: Certificación de la traducción hecha al Resumen del Informe Final.....	149
Anexo 2: Descripción de siglas	150
Anexo 3: Licencia Creative Commons	151

Índice de Figuras

Figura 1: Topología de red de Infraestructura básica.....	23
Figura 2: Tipos de Redes.....	24
Figura 3: Clasificación de las redes inalámbricas.....	25
Figura 4: Evolución del estándar 802.11.....	26
Figura 5: Capas del Modelo OSI.....	29
Figura 6: Red ALOHA.....	31
Figura 7: WLan Ad-Hoc.	31
Figura 8: Clasificación de las Redes Ad-Hoc.	34
Figura 9: Clasificación de Protocolos de enrutamiento de redes Ad-Hoc.	35
Figura 10. Proceso de enrutamiento de AODV.	36
Figura 11: Proceso de enrutamiento de TORA.	37
Figura 12: Proceso de enrutamiento de OLSR.....	38
Figura 13: Proceso de enrutamiento de ZRP.	38
Figura 14: Comparación entre arquitecturas de protocolos TCP/IP y Ad-Hoc.	39
Figura 15: Estructura de un fichero de traza en Ad-Hoc.....	47
Figura 16: Aplicación por capas de acuerdo a la clase en Ns3.	48
Figura 17: Formula para el cálculo de Througput.....	49
Figura 18: Formula para el cálculo de Tasa de entrega.	49
Figura 19: Formula para el cálculo de Retardo.	49
Figura 20. Equipamiento tecnológico en el hogar a nivel nacional.	53
Figura 21. Nivel de crecimiento de la adquisición de celulares en el hogar.....	54
Figura 22. Porcentaje de acceso a internet según el área.	54
Figura 23. Uso de las tecnologías para acceso a internet.....	55
Figura 24. Uso de internet en los 12 últimos meses.....	55
Figura 25. Lugar de uso de Internet.	56
Figura 26. Frecuencia de uso de Internet.....	56
Figura 27. Porcentaje de personas con celular activado a internet.	57
Figura 28. Porcentaje de personas con celular inteligente.	57
Figura 29: Problema de Nodos Ocultos en Manets.	58
Figura 30: Protocolos seleccionados.	60
Figura 31: Organización del simulador Ns3.	64
Figura 32: Tipo de Aplicaciones en Ns3.....	71
Figura 33: Instalación de gcc, g++ y python.	74

Figura 34: Instalación de mercurial.	74
Figura 35: Directorio de instalación de Ns3.	75
Figura 36: Descarga de bake usando mercurial.	75
Figura 37: Añadir bake a la ruta.	75
Figura 38: Comprobación de prerequisitos instalados.	76
Figura 39: Instalación de dependencias incumplidas.	76
Figura 40: Comprobación de prerequisitos instalados por segunda ocasión.	77
Figura 41: Configuración de módulos para Ns3.	77
Figura 42: Módulos específicos para la configuración incumplidos.	78
Figura 43: Instalación de módulos específicos para la configuración.	79
Figura 44: Visualización de módulos específicos instalados.	79
Figura 45: Descarga y construcción de módulos.	80
Figura 46: Prueba para verificar la construcción de Ns3.	80
Figura 47: Prueba para verificar la instalación de Ns3.	81
Figura 48: Descarga de NetAnim.	81
Figura 49: Construcción de NetAnim.	81
Figura 50: Interfaz de NetAnim.	82
Figura 51: Interfaz de nodos en NetAnim.	82
Figura 52: Módulos incluidos en la simulación.	83
Figura 53: Definición de nombres Ns3 y DSR.	83
Figura 54: Registro de notas utilizados en la simulación.	84
Figura 55: Función principal de la Simulación.	84
Figura 56: Creación de Nodos (capa física).	85
Figura 57: Fijación de canal Wifi (Capa de Enlace).	85
Figura 58: Instalación de modo Ad-Hoc.	85
Figura 59: Segmento de comparación de protocolo.	86
Figura 60: Asignación de direcciones Ip (Capa de Red).	86
Figura 61: Uso de clase de Aplicaciones (Capa de Aplicación).	87
Figura 62: Inicio y fin de la simulación.	87
Figura 63: Líneas de código para uso de NetAnim.	87
Figura 64: Envío de broadcast con OLSR en escenario uno.	88
Figura 65: Envío de broadcast con OLSR en escenario dos.	88
Figura 66: Envío de broadcast con AODV en escenario uno.	89
Figura 67: Sectorización de envío de broadcast con AODV en escenario uno.	89
Figura 68: Sectorización de envío de broadcast con AODV en escenario dos.	90

Figura 69: Mayor sectorización de envío de broadcast con AODV en escenario dos..	90
Figura 70: Bajo envío de broadcast con DSDV en escenario uno.....	91
Figura 71: Envío de broadcast con DSDV en escenario uno.....	91
Figura 72: Escenario dos sin envío de broadcast usando DSDV.	92
Figura 73: Escenario dos con envío de broadcast usando DSDV.	92
Figura 74: Movilidad inicial escenario uno y sin broadcast usando DSR.	93
Figura 75: Movilidad de los nodos escenario uno y sin broadcast usando DSR.....	93
Figura 76: Envío de broadcast en escenario uno usando DSR.	94
Figura 77: Movilidad inicial escenario dos y sin broadcast usando DSR.	94
Figura 78: Movilidad de los nodos escenario dos y sin broadcast usando DSR.	95
Figura 79: Envío de broadcast en escenario dos usando DSR.	95
Figura 80: Líneas de código para uso de FlowMonitor.....	96
Figura 81: Archivos generados .pcap.....	96
Figura 82: Análisis de archivos .pcap en wireshark.....	97
Figura 83: Archivo .xml generados en la simulación.	97
Figura 84: Resultados de las métricas obtenidas en Netanim.	98
Figura 85: Resultados de las métricas obtenidas en Netanim.	99
Figura 86: Línea estadística de la métrica TxPackets escenario I.....	100
Figura 87: Promedio de la métrica Time First TxPackets escenario I.....	101
Figura 88: Línea estadística de la métrica Time First RxPackets escenario I.....	102
Figura 89: Promedio de la métrica Time First RxPackets escenario I.	102
Figura 90: Línea estadística de la métrica Time Last TxPackets escenario I,	103
Figura 91: Promedio de la métrica Time Last TxPackets escenario I.	104
Figura 92: Línea estadística de la métrica Time Last RxPackets escenario I.	105
Figura 93: Promedio de la métrica Time Last RxPackets escenario I.	105
Figura 94: Línea estadística de la métrica Delay escenario I.	106
Figura 95: Promedio de la métrica Delay escenario I.	107
Figura 96: Línea estadística de la métrica Jitter escenario I.	108
Figura 97: Promedio de la métrica Jitter escenario I.	108
Figura 98: Línea estadística de la métrica TxBytes escenario I.	109
Figura 99: Promedio de la métrica TxBytes escenario I.	110
Figura 100: Línea estadística de la métrica RxBytes escenario I.	111
Figura 101: Promedio de la métrica RxBytes escenario I.	111
Figura 102: Línea estadística de la métrica TxPackets escenario I.	112
Figura 103: Promedio de la métrica TxPackets escenario I.	113

Figura 104: Línea estadística de la métrica RxPackets escenario I.....	114
Figura 105: Promedio de la métrica RxPackets escenario I	114
Figura 106: Línea estadística de la métrica LostPackets escenario I.....	115
Figura 107: Promedio de la métrica LostPackets escenario I.....	116
Figura 108: Cálculo de Troughput escenario I.....	117
Figura 109: Cálculo de Tasa de entrega escenario I.....	117
Figura 110: Línea estadística de la métrica Time First TxPackets escenario II.	118
Figura 111: Promedio de la métrica Time First TxPackets escenario II.....	119
Figura 112: Línea estadista de la métrica Time First RxPackets escenario II.....	120
Figura 113: Promedio de la métrica Time First RxPackets escenario II.....	120
Figura 114: Línea estadística de la métrica Time Last TxPackets escenario II.....	121
Figura 115: Promedio de la métrica Time Last TxPackets escenario II.....	122
Figura 116: Línea estadística de la métrica Time Last RxPackets escenario II.	123
Figura 117: Promedio de la métrica Time Last RxPackets escenario II.....	123
Figura 118: Línea estadística de la métrica Delay escenario II.	124
Figura 119: Promedio de la métrica Delay escenario II.....	125
Figura 120: Línea estadística de la métrica Jitter escenario II.....	126
Figura 121: Promedio de la métrica Jitter escenario II.	126
Figura 122: Línea estadística de la métrica TxBytes escenario II.....	127
Figura 123: Promedio de la métrica TxBytes escenario II.	128
Figura 124: Línea estadística de la métrica RxBytes escenario II.	129
Figura 125: Promedio de la métrica RxBytes escenario II.....	129
Figura 126: Línea estadística de la métrica TxPackets escenario II.	130
Figura 127: Promedio de la métrica TxPackets escenario II.....	131
Figura 128: Línea estadística de la métrica RxPackets escenario II.....	132
Figura 129: Promedio de la métrica RxPackets escenario II.	132
Figura 130: Línea estadística de la métrica LostPackets escenario II.	133
Figura 131: Promedio de la métrica LostPackets escenario II.....	134
Figura 132: Cálculo de Troughput escenario II.....	135
Figura 133: Cálculo de Tasa de entrega escenario II.....	135

Índice de Tablas

TABLA I. CLASIFICACIÓN DEL ESTÁNDAR 802.11	26
TABLA II. PRINCIPALES CARACTERÍSTICAS DE LAS REDES AD-HOC	33
TABLA III. TRAMAS DE CONEXIÓN ESTÁNDAR 802.11	40
TABLA IV. TRAMAS DE CONTROL ESTÁNDAR 802.11	42
TABLA V. CARACTERÍSTICAS DE NS VERSIÓN 3 SEGÚN LA PLATAFORMA	45
TABLA VI. ELECCIÓN DE UN SIMULADOR DEL RED.....	46
TABLA VII. DESCRIPCIÓN DEL ESCENARIO DE PRUEBA UNO.....	59
TABLA VIII. DESCRIPCIÓN DEL ESCENARIO DE PRUEBA DOS.....	60
TABLA XI. COMPARACIÓN ENTRE PROTOCOLOS SELECCIONADOS	62
TABLA X. MÉTRICAS PARA ELECCIÓN DEL SIMULADOR	63
TABLA XI. ESPECIFICACIÓN SOFTWARE	66
TABLA XII. ESPECIFICACIÓN HARDWARE	66
TABLA XIII. EQUIPO DE TRABAJO	67
TABLA XIV. COSTOS DEL PROYECTO	67
TABLA XV. ANÁLISIS DE LA MÉTRICA TIME FIRST TX PACKETS ESCENARIO I	100
TABLA XVI. ANÁLISIS DE LA MÉTRICA TIME FIRST RX PACKETS ESCENARIO I	101
TABLA XVII. ANÁLISIS DE LA MÉTRICA TIME LAST TX PACKETS ESCENARIO I	103
TABLA XVIII. ANÁLISIS DE LA MÉTRICA TIME LAST RX PACKETS ESCENARIO I	104
TABLA XIX. ANÁLISIS DE LA MÉTRICA DELAY ESCENARIO I	106
TABLA XX. ANÁLISIS DE LA MÉTRICA JITTER ESCENARIO I	107
TABLA XXI. ANÁLISIS DE LA MÉTRICA TX BYTES ESCENARIO I	109
TABLA XXII. ANÁLISIS DE LA MÉTRICA RX BYTES ESCENARIO I	110
TABLA XXIII. ANÁLISIS DE LA MÉTRICA TX PACKETS ESCENARIO I	112
TABLA XXIV. ANÁLISIS DE LA MÉTRICA RX PACKETS ESCENARIO I	113
TABLA XXV. ANÁLISIS DE LA MÉTRICA LOST PACKETS ESCENARIO I	115
TABLA XXVI. CÁLCULO DE TROUGHPUT ESCENARIO I.....	116
TABLA XXVII. CÁLCULO DE TASA DE ENTREGA ESCENARIO I.....	117
TABLA XXVIII. ANÁLISIS DE LA MÉTRICA TIMEFIRSTTXPACKETS ESCENARIO II	118

TABLA XXIX. ANÁLISIS DE LA MÉTRICA TIMEFIRSTXPACKETS ESCENARIO II	119
TABLA XXX. ANÁLISIS DE LA MÉTRICA TIMELASTTXPACKETS ESCENARIO II	121
TABLA XXXI. ANÁLISIS DE LA MÉTRICA TIMELASTXPACKETS ESCENARIO II.....	122
TABLA XXXII. ANÁLISIS DE LA MÉTRICA DELAY ESCENARIO II	124
TABLA XXXIII. ANÁLISIS DE LA MÉTRICA JITTER ESCENARIO II.....	125
TABLA XXXIV. ANÁLISIS DE LA MÉTRICA TX BYTES ESCENARIO II	127
TABLA XXXV. ANÁLISIS DE LA MÉTRICA RX BYTES ESCENARIO II.....	128
TABLA XXXVI. ANÁLISIS DE LA MÉTRICA TX PACKETS ESCENARIO II	130
TABLA XXXVII. ANÁLISIS DE LA MÉTRICA RX PACKETS.....	131
TABLA XXXVIII. ANÁLISIS DE LA MÉTRICA LOST PACKETS ESCENARIO II	133
TABLA XXXIX. CÁLCULO DE TROUGHPUT ESCENARIO II	134
TABLA XXXX. CÁLCULO DE TASA DE ENTREGA ESCENARIO II	135
TABLA XXXXI. COMPARACIÓN DE MÉTRICAS ENTRE PROTOCOLOS ESCENARIO DE PRUEBA UNO.....	136
TABLA XXXXII. COMPARACIÓN DE MÉTRICAS ENTRE PROTOCOLOS ESCENARIO DE PRUEBA DOS.....	138
TABLA XXXXIII. VALORACIÓN ECONÓMICA	140
TABLA XXXXIV. TABLA DE SIGNIFICADOS	150

c. Introducción

Actualmente las redes inalámbricas han tomado fuerza tanto en innovación de tecnología, uso y penetración las que claramente favorecen muchas actividades de nuestro entorno cotidiano como los negocios y el comportamiento social, con mayor impulso gracias a la fuerza de Internet.

Debido a la evolución de la tecnología sobretodo de los dispositivos móviles, se está creando mayores usos a las tecnologías que usan bandas de frecuencia, las que no involucran mayor esfuerzo e infraestructura y que son la mejor alternativa en términos de portabilidad, usabilidad y precio, las Manet (Mobile Ad-Hoc Network) constituyen una tecnología ideal para el establecimiento instantáneo de la comunicación entre nodos que son móviles y en lugares en donde no es posible construir un backbone de comunicaciones debido a inconvenientes económicos o a la falta de tiempo para la construcción de la infraestructura requerida, ejemplo de estos escenarios son las operaciones militares o las operaciones de rescate en caso de incendios, terremotos o inundaciones.

El principal problema que se plantea es el uso de protocolos inadecuados, que afectan a la estabilidad de las redes Ad-Hoc, y al utilizarlos se obtiene mejora de la eficiencia en el paso de información en tiempo y calidad

En el presente proyecto de fin de carrera, se realiza un estudio de los protocolos de enrutamiento para Redes Ad-Hoc, realizando primeramente revisión literaria a profundidad, para comprender las ventajas y desventajas del proyecto y del tema de estudio, con el uso de metodologías y técnicas que aporten al buen desarrollo del proyecto, para poder evaluar las métricas resultantes de la simulación en Network Simulator Ns3, logrando resultados en base a la experimentación en dos escenarios, y proponer el mejor protocolo según el escenario, orientado a la mejora del enrutamiento y traducido en cierta forma en la estabilización de las redes Ad-Hoc. Finalmente se realiza la valoración de recursos utilizados en el proyecto, acompañados de conclusiones y recomendaciones que aporten a los nuevos interesados en el tema.

d. Revisión de Literatura

1. Redes

1.1 Definición de Red

En la figura 1 se ilustra primeramente la base de la tesis, una red básica que en los términos más simples se define como un conjunto de dispositivos interconectados que intercambia información y comparten recursos entre sí a través de un medio.[1]

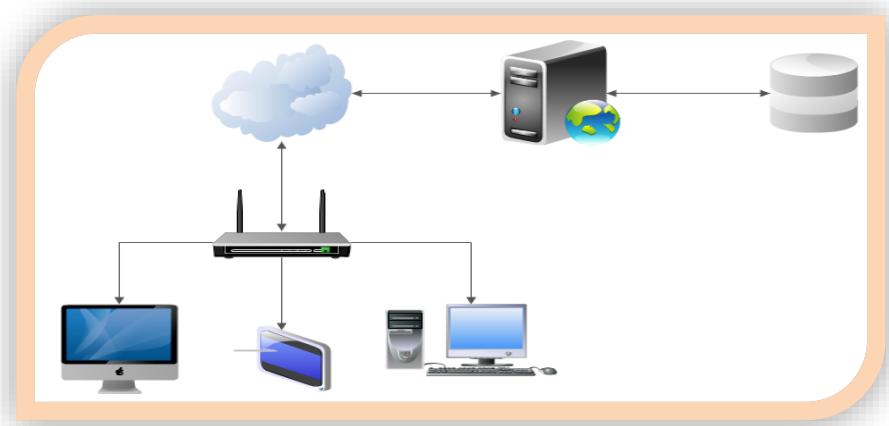


Figura 1: Topología de red de Infraestructura básica.

1.2 Clasificación de Redes de acuerdo a su área de Cobertura

La clasificación de las redes se ilustra en la figura 2, dicha clasificación se basa en un conjunto de características que se describen a continuación:

- **PAN (PERSONAL AREA NETWORK) O RED DE ÁREA PERSONAL:**
Tiene un rango de alcance de hasta los 10m, conocido como el estándar 802.15 Bluetooth Su tasa de transmisión se encuentran entre los 10 bps hasta los 10Mbps, en la frecuencia 2,4Ghz este tipo de redes se caracterizan por su baja complejidad en el diseño, bajo costo y reducido consumo de potencia[1].

- **LAN (LOCAL AREA NETWORK) O RED DE ÁREA LOCAL:** Bajo el estándar 802,11 Wifi, es una red cuyo rango de alcance es aproximadamente de 10m a 150m, con frecuencia 2.4 y 5 Ghz, se distinguen por su tamaño (tiempos de transmisión limitados y conocidos.), su tecnología de transmisión operan de 10 a 100Mbps. Las nuevas LAN funcionan hasta 10 Gbps y su topología se encuentran dos principales topologías: bus y anillo[1].
- **MAN (METROPOLITAN AREA NETWORK) O RED DE ÁREA METROPOLITANA:** Esta red utiliza el estándar 802,16 WiMax, su rango de alcance llega hasta los 50m, opera sobre la frecuencia 2.11 Ghz es una red de alta velocidad (banda ancha) que da cobertura en un área geográfica más extensa, proporciona capacidad de integración de múltiples servicios mediante la transmisión de datos, voz y video, ofrecen velocidades de 10Mbps, 20Mbps, 45Mbps, 75Mbps, sobre pares de cobre y 1Gbps, 10Gbps en fibra óptica[1].
- **WAN (WIDE AREA NETWORK) O RED DE ÁREA AMPLIA:** Ofrece velocidades de trasmisión hasta 384Kbps, con frecuencia 0.9, 1.8 y 2.1 GHz, se extiende sobre un área geográfica extensa empleando medios de comunicación poco habituales, como satélites, cables interoceánicos, fibra óptica, etc. Utiliza medios públicos. La diferencia de las demás es que proporciona un medio de transmisión a larga distancia de datos, voz, imágenes, videos, sobre grandes áreas geográficas que pueden llegar a extenderse hacia un país, un continente o el mundo entero, cubre distancias desde 100km hasta 1000 km. Un ejemplo claro de esta red es Internet[1].



Figura 2: Tipos de Redes.

1.3 Redes Inalámbricas y su clasificación

Las redes inalámbricas se dividen en dos categorías: basadas en infraestructura preestablecida y redes Ad-Hoc. Actualmente nos encontramos con los siguientes tipos de redes inalámbricas las que se aprecian gráficamente en la figura 3:

- **WPAN (Wireless Personal Area Network - Red inalámbrica de ámbito personal):** Cubren un área del tamaño de una habitación. Tradicionalmente este tipo de redes fue basado en infrarrojos que permiten la comunicación entre dos elementos (ordenadores portátiles, PDAs, etc.) a baja velocidad y a una distancia cercana[2].
- **WLAN (Wireless Local Area Network - Red inalámbrica de ámbito local):** Son las redes que cubren el ámbito de una casa, una oficina o el edificio de una empresa[2].
- **WWAN (Wireless Wide Area Network - Red inalámbrica de área extensa):** Son las redes cuyo ámbito cubre áreas más amplias por ejemplo: una ciudad. Por su gran tamaño, estas redes son usadas a gran escala por las empresas de telefonía móvil o ISPs (Internet Service Providers)[2].

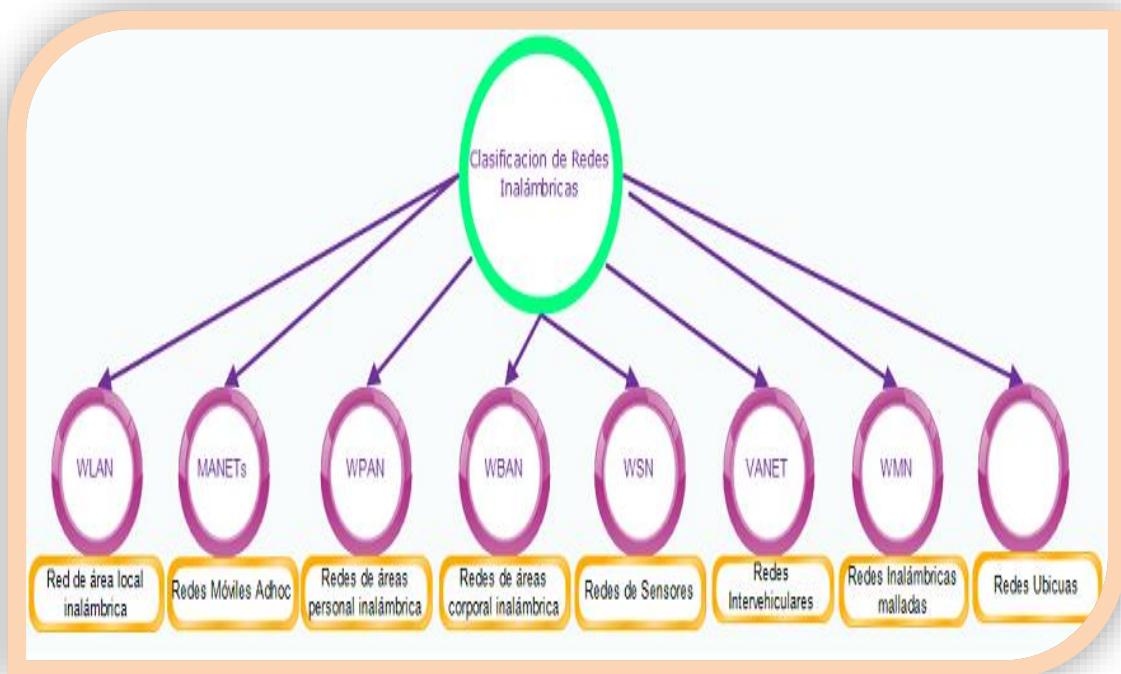


Figura 3: Clasificación de las redes inalámbricas.

1.4 Tecnologías WLAN IEEE 802.11

La especificación IEEE 802.11 (ISO/IEC 802-11) es un estándar internacional que define las características de una red de área local inalámbrica (WLAN). Wi-Fi estándar de fidelidad inalámbrica, es el nombre otorgado por la Wi-Fi Alliance, anteriormente WECA (Wireless Ethernet Compatibility Alliance), grupo que garantiza la compatibilidad entre dispositivos que utilizan el estándar 802.11, en la figura 4 se puede ver la evolución del estándar en línea de tiempo desde sus inicios en el año 1997 hasta el 2012[2].

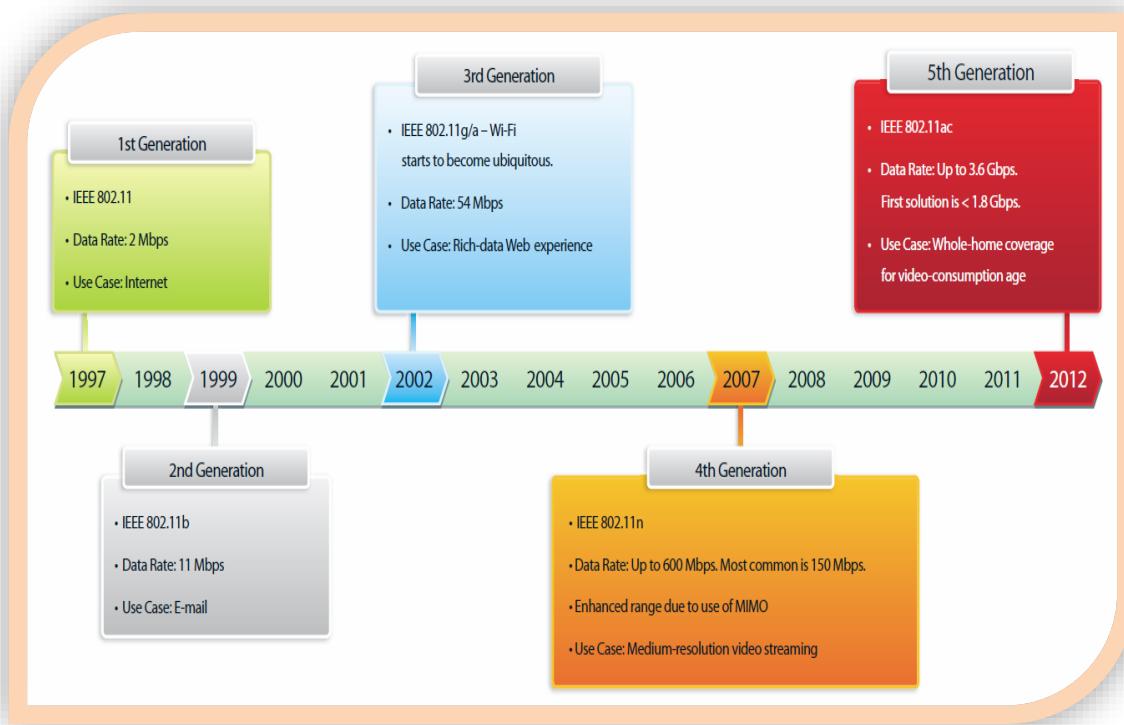


Figura 4: Evolución del estándar 802.11.

En la **TABLA I** se menciona algunas de las divisiones del estándar para cubrir áreas específicas de origen inalámbrico[3].

TABLA I. CLASIFICACIÓN DEL ESTÁNDAR 802.11

Estándar	Nombre	Descripción
802.11a	WifiS	Este estándar admite un ancho de banda de 54Mbps, provee ocho canales de radio en la banda de frecuencia de 5 GHz.
802.11b	Wifi	Es el más utilizado, ofrece un rendimiento de 11Mbps, tiene un alcance de hasta 300m en un

		espacio abierto, usa el rango de frecuencia de 2,4GHz, con tres canales de radio disponibles.
802.11c	Combinación de 802.11 y 802.1d	Es una versión modificada del estándar 802.1d que permite combinar 802.1d con dispositivos compatibles con 802.11(a nivel de enlace de datos)
802.11d	Internacionalización	Es un complemento del estándar 802.11 que está pensado para permitir el uso internacional de las redes 802.11 locales.
802.11e	Mejora de calidad del servicio	Destinado a mejorar la calidad del servicio en el nivel de la capa de enlace de datos, su objetivo es definir los requerimientos de ancho de banda de acuerdo al paquete y el retardo de la transmisión para permitir mejores transmisiones de audio y video.
802.11f	Itinerancia	Utiliza el protocolo IAPP que le permite a un usuario itinerante cambiarse claramente de un punto de acceso a otro mientras esta en movimiento sin importar que marcas de puntos de acceso se usan en la infraestructura de red.
802.11g		Este estándar admite un ancho de banda de 54Mbps, en el rango de frecuencia de 2.4 GHz. Es compatible con el estándar 802.11b, lo que implica que los dispositivos pueden funcionar sin problemas si admiten cualquiera de estos estándares.
802.11h		Su objetivo es unir el estándar 802.11 y el estándar europeo HiperLan 2, y cumplir las regulaciones relacionadas con el uso de las frecuencias y el rendimiento energético.
802.11i		Destinado a mejorar la seguridad en la transferencia de datos (administra y distribuye claves, cifrado y autenticación). Este estándar se basa en AES (estándar de cifrado avanzado) y puede cifrar transmisiones que se ejecutan en las tecnologías 802.11 ^a , 802.11b y 802.11g

802.11j		Parecido a estándar 802.11h, pero este destinado para la regulación japonesa
802.11k		Permite a los commutadores y puntos de acceso inalámbricos calcular y valorar los recursos de radiofrecuencias de los clientes en una WLAN, mejorando así su gestión.
802.11n		Ratificado por la IEEE el 11 de septiembre de 2009, con una velocidad de 600Mbps en capa física.
802.11p		Opera en el espectro de frecuencia de 5,90GHz y de 6,20 GHz especialmente indicado para automóviles. La base de las comunicaciones dedicadas de corto alcance (DSRC) en Norteamérica, la tecnología DSRC permitirá el intercambio de datos entre vehículos e infraestructura en carretera.
802.11r		Su principal característica es permitir a la red que establezca los protocolos de seguridad que identifican a un dispositivo en el nuevo punto de acceso antes de que abandone el actual lo que permite que la transición entre los nodos demore menos de 50 milisegundos como VOIP.
802.11v		Para la gestión de todas las redes de forma centralizada, o distribuida a través de un mecanismo de capa 2, mecanismos de ahorro de energía con dispositivos de mano VOIP Wi-Fi, posicionamiento, proporcionar nuevo servicios dependientes de la ubicación, temporización y coexistencia con mecanismos para reducción de interferencia entre diferentes tecnologías en un mismo dispositivo.
802.11w		Este estándar podrá proteger las redes contra la interrupción causada por los sistemas malévolos que crean peticiones desasociadas que parecen ser enviadas por el equipo valido.

1.5 Modelo OSI

El estándar 802.11 establece los niveles inferiores del modelo OSI, es decir la Capa Física y la Capa de Enlace, para conexiones inalámbricas que utilizan ondas electromagnéticas, estas capas se describen gráficamente en la figura 5 haciendo una corta descripción de su función dentro del modelo[3].

El estándar 802.11 es el primer estándar y permite un ancho de banda de 1 a 2 Mbps. El estándar original se ha modificado para optimizar el ancho de banda (incluidos los estándares 802.11a, 802.11b y 802.11g, denominados estándares físicos 802.11) o para especificar componentes de mejor manera con el fin de garantizar mayor seguridad o compatibilidad[4].



Figura 5: Capas del Modelo OSI.

1.5.1 Capa Física

Abreviada como PHY que ofrece tres tipos de codificación de la información, la capa física del estándar 802.11 ofrece dos técnicas para la transmisión de frecuencias de radio y una especificación para transmisión infrarrojas[3].

1.5.2 Capa de Enlace

Está compuesta por dos subcapas LLC control de enlace lógico, y MAC control de acceso al medio.

1.5.2.1 Subcapa de Control Lógico de enlace (Logical Link LLC)

Esta sub-capa es exactamente igual a la capa LLC utilizada por las redes cableadas del tipo 802,3 con un sistema de direccionamiento de 48 bits idénticos (MAC Address). Esto permite simplificar al extremo los puentes (bridges) entre los dos tipos de red[3].

1.5.2.2 Subcapa de Control de Acceso al Medio (Media Access Control MAC)

El método de acceso múltiple es la llamada función de distribución Coordinada (Distributed Coordination Function DCF) utiliza el conocido método de acceso múltiple por censado de portadora con prevención de colisiones (Carrier Sense Multiple Access/Collision Avoidance, CMSA/CA), requiere que cada nodo inalámbrico escuche el medio compartido para saber si otros nodos se encuentran trasmitiendo, si el canal está desocupado el nodo puede trasmitir , caso contrario el nodo sigue escuchando hasta que la transmisión finalice para luego ejecutar el procedimiento, evitando la monopolización de algunos nodos[3].

2. Inicios de las redes Ad-Hoc

Las redes Ad-Hoc tienen sus inicios en 1968 con la red **ALOHA** que se utilizó para la interconexión de centros educativos en Hawái, aunque se emplearon estaciones fijas, el protocolo ALOHA se prestaba a la gestión distribuida de acceso a canales, un protocolo de salto simple (single-hop) que no soportaba el encaminamiento de manera inherente, en su lugar, cada nodo tenía que estar al alcance de todos los demás nodos participantes[5].

Inspirada por ALOHA, DARPA comenzó a trabajar, en 1973, en la PRnet (packet radio network) una red de salto múltiple, cuyos nodos cooperaban en la retransmisión del tráfico en beneficio mutuo para llegar a estaciones distantes que de otra manera habrían estado fuera de su alcance, con estas técnicas se pudo observar que se aumentaba la capacidad de la red, ya que el dominio espacial podía ser reutilizado para sesiones de salto múltiple concurrentes pero físicamente separadas. Cuando se estaba desarrollando IEEE 802.11n un estándar para redes de área local inalámbricas (wireless

local área network - WLAN) el Instituto de Ingeniería Eléctrica y Electrónica (Institute for Electrical and Electronic Engineering - IEEE) sustituye el término red de paquetes de radio por el de red Ad-Hoc[5].

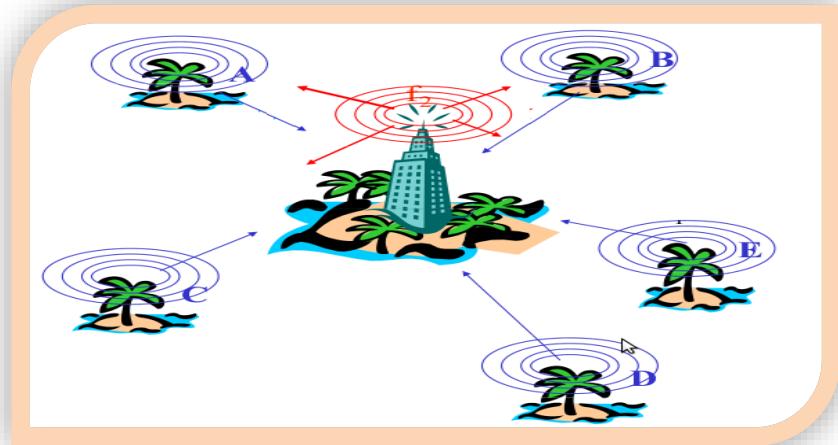


Figura 6: Red ALOHA.

2.1 Definición de Red Ad-Hoc

En este tipo de red los nodos pueden moverse de forma libre y no dependen de un control central preestablecido, un nodo puede establecer un enlace con otros nodos siempre y cuando se encuentre dentro de su rango de transmisión[6].

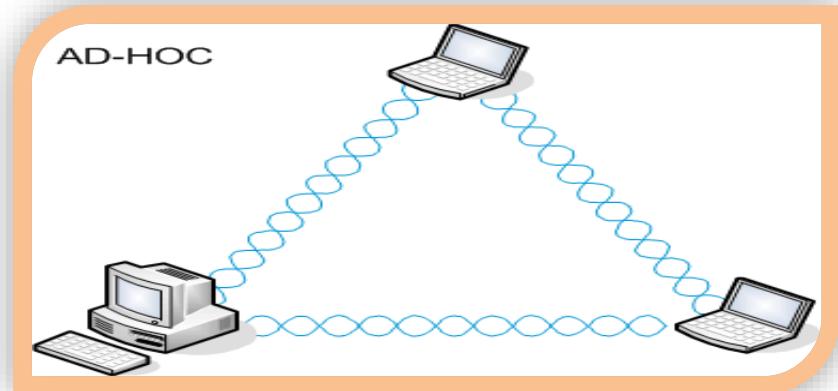


Figura 7: WLAN Ad-Hoc.

La figura 7 muestra una forma básica donde opera una red Ad-Hoc conformada por tres nodos móviles interconectados entre sí a través de enlaces inalámbricos de

comunicación, las MANET pueden ser de uno a más saltos conformadas sin infraestructura preestablecida, sin control centralizado.

2.2 Estructura de las Redes Ad-Hoc

Las Redes Ad-Hoc utilizan múltiples tecnologías y protocolos existentes los que permiten establecer comunicaciones inalámbricas, básicamente una red 802.11 se puede encontrar los siguientes componentes, siendo algunos no necesarios para desplegarla[7]:

- **Estación (Station, STA):** Dispositivo equipado con una interfaz 802.11 por ejemplo PDA, PC, Teléfono móvil, etc.
- **Punto de Acceso (Access Point, AP):** Son dispositivos que interconectan varias STA formando de esta manera una red, en ocasiones incorporan la funcionalidad de routers o enrutadores con capacidad para encaminar paquetes de datos y asignar direcciones Ip.
- **Portal:** Dispositivo que interconecta un red 802.11 con otra se puede encontrar el PORTAL y el AP en el mismo dispositivo físico.
- **Sistema de Distribución (Distribution System, DS):** Varios AP's interconectados conformando una columna vertebral (Backbone), conocida como sistema de distribución.
- **BSS (Basic Service Set):** Es el conjunto conformado por AP's y STA's asociadas, las que están identificadas por su BSSID (identificador de 6 bits).
- **ESS (Extended Service Set):** El conjunto de WLAN's interconectadas, formada por cada BSS, AP's, STA's y ESS.

2.3 Características de las redes Ad-Hoc

El propósito de las redes móviles Ad-Hoc es el de apoyar operaciones robustas y eficientes, incorporando un enrutamiento funcional en los nodos móviles. Debido a que las MANET se pueden construir espontáneamente sin necesidad de una infraestructura existente, estas redes presentan restricciones especiales que se deben considerar para el diseño de soluciones. En la **TABLA II** se describen algunas características[8]:

TABLA II. PRINCIPALES CARACTERÍSTICAS DE LAS REDES AD-HOC

Característica	Descripción
Topología dinámica	Debido a que los nodos de una red móvil Ad-Hoc son libres de moverse arbitrariamente, la topología de la red (típicamente multi-hop) puede cambiar rápida y aleatoriamente. Este comportamiento dinámico dificulta el establecimiento de la conectividad en la red, la cual se debe mantener para permitir que los servicios de comunicación operen sin interrupciones.
Operación en forma distribuida	El término distribuido hace referencia a que los nodos en una MANET, tienen un conocimiento local sobre su entorno. Es decir, cada nodo solo conoce la información sobre los nodos vecinos que se encuentran dentro de su radio de transmisión y no tiene un conocimiento global de la red.
Ancho de banda limitado	El ancho de banda disponible en MANET es menor, comparado con el ancho de banda disponible en las redes de infraestructura preestablecida.
Fluctuación de los enlaces	El efecto de una alta tasa de error de bits (información errada), es más significativo en una red móvil Ad-Hoc de múltiples saltos, ya que el agregado de todos los errores en los enlaces afecta significativamente la calidad de la información recibida.
Operación de energía limitada	Los nodos en una MANET requieren de una batería para obtener la energía que asegure su funcionamiento. Por lo tanto la energía es uno de los recursos que se busca optimizar con el fin de limitar el consumo de la potencia.

2.4 Tipos de Redes Ad-Hoc

La gráfica a continuación desglosa los tipos de redes Ad-Hoc, mencionando algunas de las características por lo que se hace esta división[8].



Figura 8: Clasificación de las Redes Ad-Hoc.

2.5 Campos de Aplicación de las Redes Ad-Hoc

A continuación se mencionan algunos de los campos de aplicación de las redes Ad-Hoc[8]:

- **Redes Tácticas:** Comunicación en operaciones militares, campos de batalla automatizados.
- **Redes de Sensores:** Recogida de datos en tiempo real, generalmente correlacionados en espacio y tiempo.
- **Servicio de salvamento y emergencia:** Operaciones de búsqueda y rescate, o sustitución de redes con infraestructura en caso de desastres naturales.
- **Entornos Comerciales:** Comercio electrónico, acceso remoto a bases de datos centralizadas, oficina móvil, servicios vehiculares.
- **Redes para particulares y empresas:** Redes de Área local WLAN, para oficinas u hogares.
- **Ocio:** Juegos multiusuario, mascotas robots, accesorios a internet en exteriores.
- **Servicios de localización:** Para seguimiento e información.

- **Otros:** Extensión de la cobertura en un sitio de construcción, acceso inalámbrico público en zonas urbanas.

2.6 Protocolos de Enrutamiento de las Redes Ad-Hoc

En el caso de las redes Ad-Hoc se deben utilizar algoritmos de enrutamiento adecuados a ciertas características que estas poseen, por ejemplo la topología dinámica, en donde cada nodo debe tener la capacidad también de ser router.

2.6.1 Clasificación de Protocolos en redes Ad-Hoc

Los protocolos de las redes Adhoc se dividen en tres grupos, los que se pueden ver gráficamente en la siguiente figura y se describen a continuación[8]:

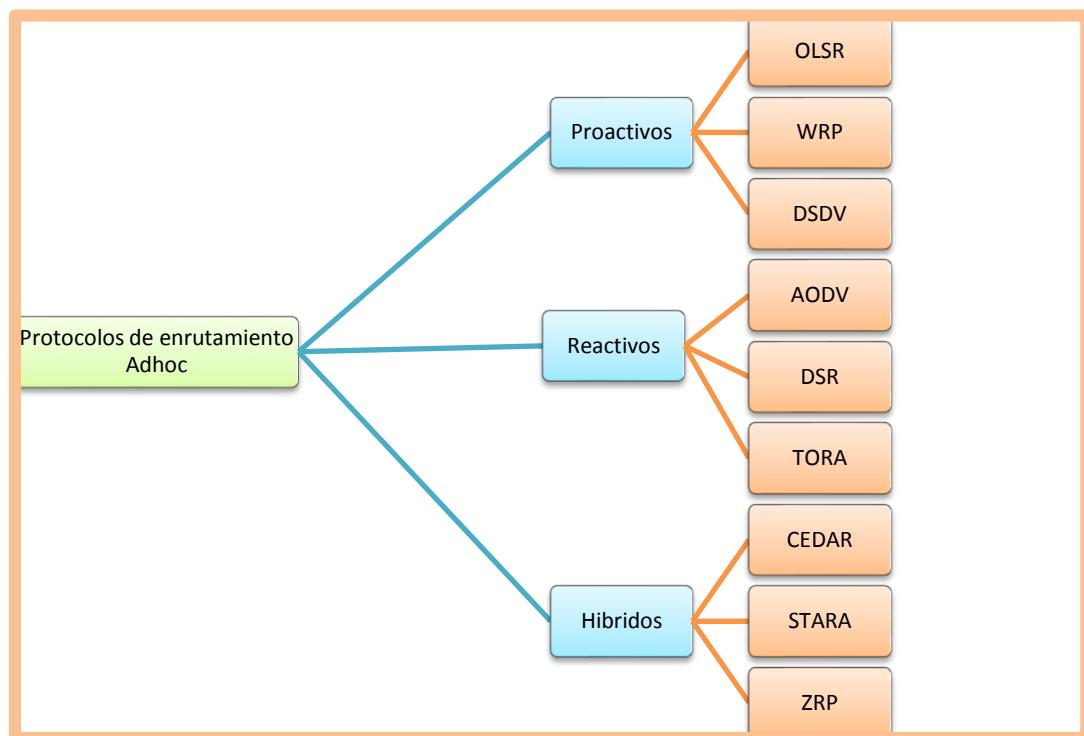


Figura 9: Clasificación de Protocolos de enrutamiento de redes Ad-Hoc.

- **Proactivos:** Mantienen constantemente actualizada la información de direccionamiento, a través de intercambio de paquetes. Permite tener un direccionamiento disponible a cada petición de enrutamiento. Su principal problema es la sobrecarga en la red por el tráfico que produce[8].
- **Reactivos:** Este tipo de algoritmo invoca únicamente el envío de tráfico de descubrimiento de ruta solo cuando el paquete efectivamente se ha de transmitir, por lo que no existe sobrecarga[8].

- **Híbridos:** Busca unir las ventajas de los anteriores, haciendo que cada nodo tenga conocimiento solo de sus nodos adyacentes (enrutamiento proactivo)[9].

A continuación se describen algunos protocolos de direccionamiento en redes Adhoc que son frecuentes:

2.6.1.1 AODV (Ad-hoc On Demand Vector)

Es un protocolo reactivo que soporta enrutamiento multidifusión (multicast) basado en el algoritmo vector distancia. Crean un número de secuencia de destino (destination sequence number), esto hace que los nodos sepan cuando se actualiza un recorrido evitando así la formación de lazos (loop) en el camino, ya que aquel nodo que deba elegir el siguiente nodo elegirá el próximo de su secuencia mayor correspondiente a la información de enrutamiento más reciente[9].

Usa mensajes llamados RREQ (Route Request), RREP (Route Replies) y RERR (Route Errors) enviados y recibidos mediante UDP, en este tipo de direccionamiento cada nodo tiene conocimiento local de sus vecinos[9].

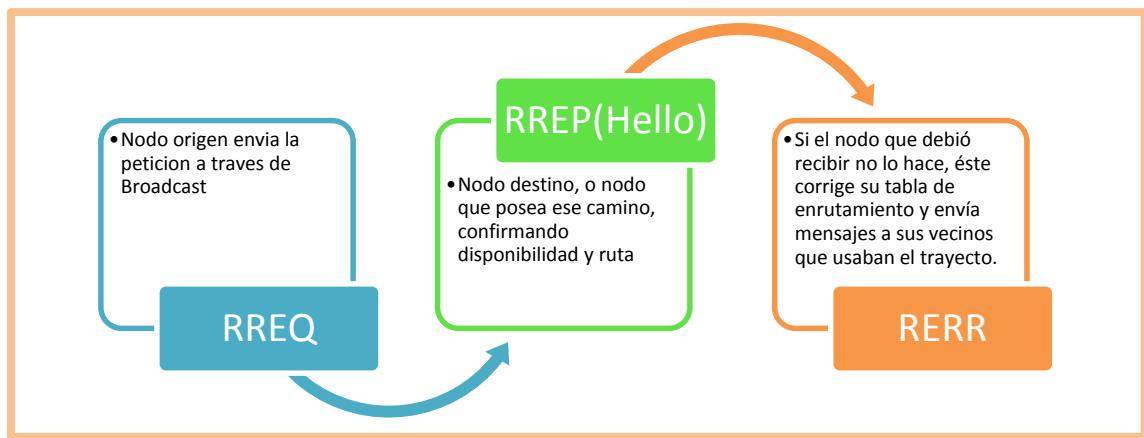


Figura 10. Proceso de enrutamiento de AODV.

2.6.1.2 DSDV (Destination Sequenced Distance Vector)

Pertenece a los algoritmos de vector distancia, es un protocolo proactivo que enruta de manera que cada nodo posea una tabla de enrutamiento que indica cuantos saltos (hop) necesita dependiendo del destino, realizando actualización de tablas mediante intercambio de información entre nodos cercanos, además de aplicar algoritmos de camino mínimo a menor costo, dependiendo del número de secuencia (sequence number), mientras el número de secuencia más alto se considera más fiable el camino

al encontrar dos caminos con número de secuencia igual, se verifica el menor coste dependiendo el número de saltos[9].

2.6.1.3 DSR (Dynamic Source Routing)

Es un protocolo reactivo, que utiliza Source Routing y el mecanismo On Demand, en donde cada nodo fuente u origen conoce paso a paso (hop by hop) la ruta para llegar al destino, gracias a la memoria de enrutamiento (route cache) que memoriza todos los caminos[9].

Utilizan de igual manera los mensajes RREQ, RREP y RERR como en AODV, el nodo origen envía mensajes RREQ, el origen o nodos dentro de la ruta le contestaran con RREP y en caso de no ser posible la comunicación durante la transmisión se enviaran RERR, pero en este caso todos los nodos utilizan su memoria actualizando sus tablas, y excluyendo el enlace que tiene error, este protocolo hace uso de su memoria de direccionamiento y fuente de direccionamiento de manera que se pueda evitar los lazos (loop)[9].

2.6.1.4 TORA (Temporally Ordered Routing ALgorithm)

Es un protocolo reactivo, este proporciona al nodo remitente múltiples trayectos para llegar al destino, cada nodo tendrá una copia del TORA para cada destino en donde cada uno de ellos tiene la libertad de crear, mantener y cancelar trayectos de enrutamiento, se asocia a cada nodo un peso de manera de ir de mayor a menor, mientras que los caminos descubiertos con paquetes QRY (Querry) vienen actualizados con aquellos de tipo UDP (Update)[9].

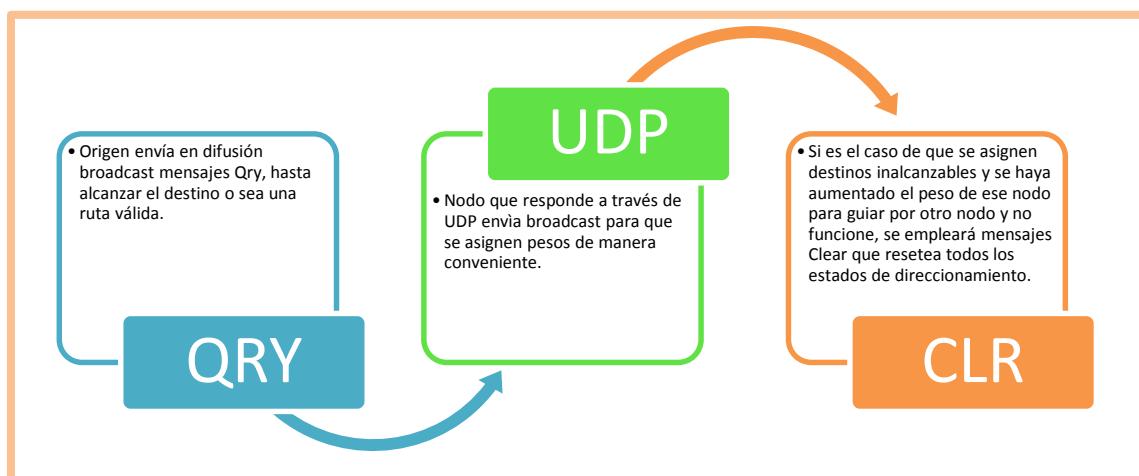


Figura 11: Proceso de enrutamiento de TORA.

2.6.1.5 OLSR (Optimized link State Routing Protocol)

Es un protocolo de encaminamiento proactivo diseñado para redes Adhoc altamente dinámicas y con ancho de banda limitado, provee un mecanismo eficientemente de diseminación (flooding) de información parcial del estado de la red empleando un conjunto de nodos especiales MPR (Multipoint Relays)[9].

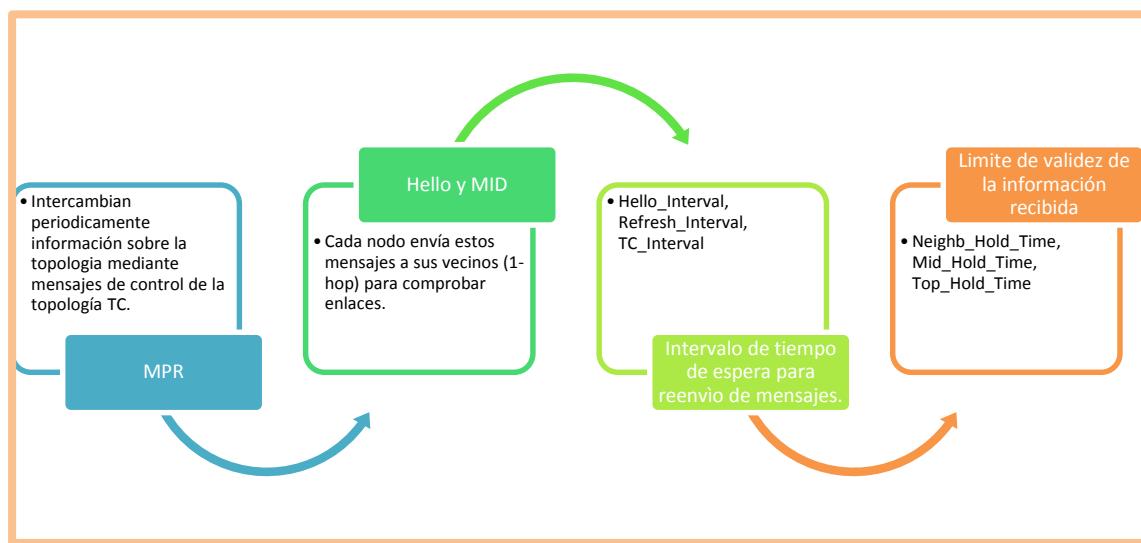


Figura 12: Proceso de enrutamiento de OLSR.

2.6.1.6 ZRP (Zone Routing Protocol)

Es un protocolo Hibrido, busca reducir la sobrecarga de control de protocolos en enrutamiento proactivo y disminuir la latencia por el descubrimiento de rutas en protocolos reactivos, se define como radio de zona lo que hace es dividir la topología por zonas y crear para cada zona tablas de enrutamiento, cuando un nodo hace la petición se verifica en su tabla si el destino se encuentra en su zona, si no es así se envía a comprobar en otra zona si está allí el destino o existe otro camino, si está en su zona envía la contestación al emisor, sino difunde el mensaje entre sus vecinos[9].

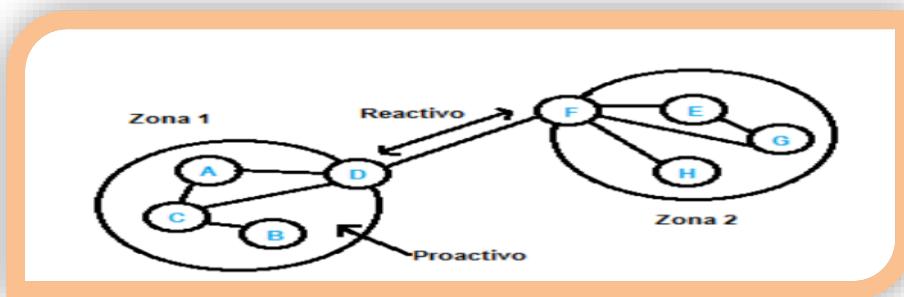


Figura 13: Proceso de enrutamiento de ZRP.

2.7 Comparación entre Arquitectura regular TCP/IP y AdHoc

La arquitectura de los protocolos TCP/IP, es una arquitectura que ha demostrado ser fiable y adaptable a cualquier medio. Por lo tanto, sería interesante poder aplicar a redes AdHoc y así mantener la compatibilidad con internet. Pero las características propias de redes AdHoc sobre todo la de topología variable hacen que ciertos protocolos se degraden notablemente, razón por la cual se utilizan protocolos especializados en redes AdHoc. A continuación se hace una comparativa gráfica entre la arquitectura de los protocolos TCP/IP (tcp, udp) y AdHoc (ahr), en la figura 14[10].

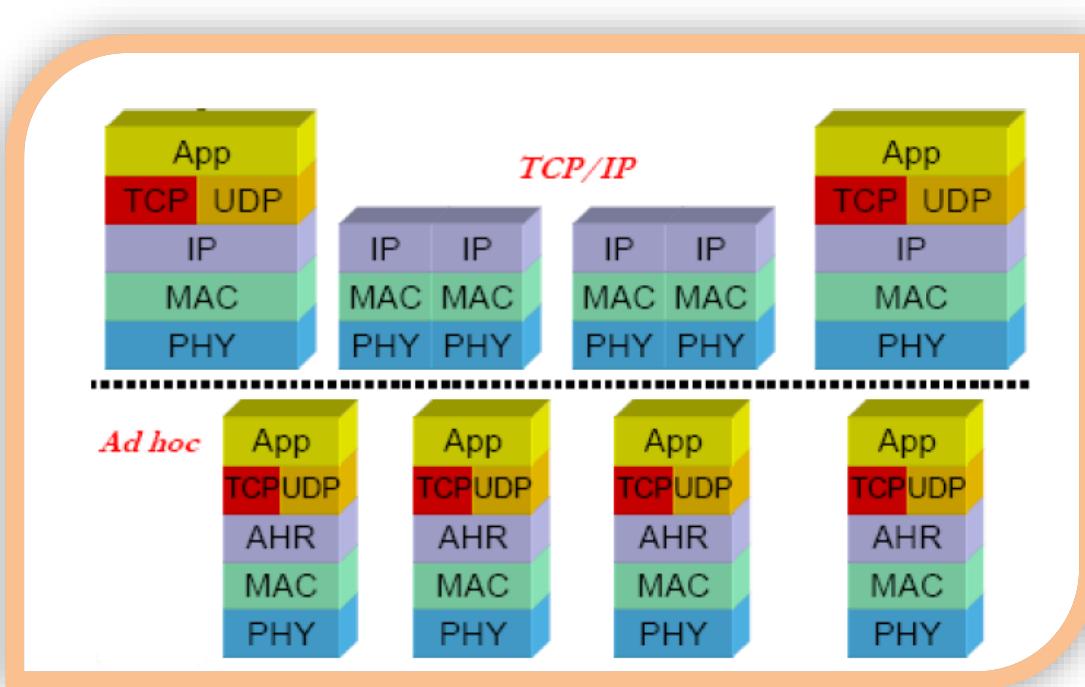


Figura 14: Comparación entre arquitecturas de protocolos TCP/IP y Ad-Hoc.

2.8 Modos de Transmisión en redes Ad-Hoc

El modelo de transmisión atiende que destinatarios deben recibir los paquetes. En un canal inalámbrico, todos los nodos dentro del rango de cobertura del nodo origen recibirán los paquetes, estableciendo un modelo de comunicación broadcast[10].

No obstante, se puede establecer otros mecanismos para restringir el grupo de destinatarios. A continuación se hace mención a los modos de transmisión más frecuentes:

2.8.1 Unicast

Se refiere a la comunicación uno a uno, de un nodo origen a un único nodo destino. La mayor parte de los protocolos de encaminamiento para redes Ad-Hoc se ubica dentro de esta categoría, ya que se trata del modo de operación normal en cualquier red. Los protocolos que lo utilizan son: **DSDV, DSR, AODV, TORA, ZRP, OLSR[10]**.

2.8.2 Multicast

Este modelo de comunicación se emplea cuando un nodo necesita enviar información idéntica a varios destinos. Un nodo puede incorporarse o disgregarse de un grupo multicast cuando lo deseé, pero en contrapartida, los nodos deben mantener información adicional sobre los nodos que se adhieren o abandonan su grupo multicast. Existe solo un protocolo en redes AdHoc que utiliza difusión multicast llamado **MAODV[10]**.

2.9 Tramas en Redes AdHoc

Las tramas (frames) del estándar IEEE 802.11, para analizar el funcionamiento de una WLAN, se puede decir en forma general que el estándar 802.11 define una serie paquetes que son usados por los nodos y los AP para establecer la comunicación entre ellos y mantener el enlace entre ellos.

Cada trama tiene un campo de control que define la versión del protocolo 802.11, el tipo de trama, la dirección MAC del origen y del destino, el número de secuencia de la trama y una secuencia de redundancia para detección de errores[11].

2.9.1 Tramas de manejo de Conexión

Permiten a los nodos establecer y mantener la comunicación entre ellos. En la **TABLA III** se hace referencia a los diferentes subtipos de tramas de conexión[11].

TABLA III. TRAMAS DE CONEXIÓN ESTÁNDAR 802.11

Trama	Descripción
Tramas de Autenticación	La autenticación es el proceso por el cual un Access Point acepta o rechaza la identidad de un nodo que pretende conectarse con él. El nodo inicia el procedimiento enviando una trama de autenticación, si la autenticación es Abierta, el AP simplemente contesta con una trama de respuesta afirmativa o negativa. Si el AP tiene definido el tipo opcional de

	Autenticación por frase de paso compartida (Shared Key Authentication), el AP responde con una trama de respuesta conteniendo una frase de texto. El nodo deberá ahora enviar una versión encriptada de la palabra de paso usando su clave WEP para encriptar. El AP se asegura que el nodo tiene la clave WEP correcta desencriptando y comparando la frase de texto con la que envió previamente.
Tramas de Desautenticación	Es una trama enviada por un nodo a otro nodo para terminar la conexión segura entre ellos.
Tramas de solicitud de asociación	La asociación en 802.11 habilita a un AP a disponer de recursos para establecer una conexión con un nodo estación. El nodo estación comienza la solicitud de asociación enviando una trama de este tipo. Esta trama contiene información sobre el nodo estación como las velocidades soportadas y el SSID al cual desea asociarse. El AP evalúa el requerimiento del nodo y si decide aceptar reserva un espacio de memoria para permitir el intercambio de datos y establece un número de asociación (Association ID) para el nodo.
Trama de respuesta de Asociación	Es la trama con el que el AP responde a una solicitud de asociación. La trama contiene información relativa a la asociación en cuestión como el Association ID, las velocidades aceptadas, etc.
Tramas de Reasociación	Las tramas de solicitud de re asociación se envían cuando un nodo se mueve y encuentra otro AP con mayor señal (Beacon Signal) que el actual al que está asociado. El nuevo AP al recibir esta señal coordina a través del DS (Red cableada probablemente) el envío de los paquetes que pudieran estar pendientes en el viejo AP para transmitirlos al nodo y luego envía una trama de respuesta de re asociación con los nuevos datos de Association ID y las velocidades aceptadas.
Trama de desasociación	Es una trama que suele enviar un nodo estación cuando desea cancelar la asociación en forma ordenada. Esta trama instruye al AP para que libere la memoria y el Association ID relacionado a este nodo.

Trama de balizamiento (beacon)	Es una trama que el AP envía periódicamente para anunciar su presencia y recabar información tales como el SSID y otros parámetros que le indican al AP si los nodos siguen a su alcance. Los nodos estación permanentemente escanean los canales de radio escuchando los beacons para establecer a cual AP conviene asociarse.
Tramas de prueba	Las tramas de requerimiento de prueba son los que envían los nodos estación para saber que otras radios están al alcance. Al recibirlos, el otro extremo responde con una trama de respuesta a la prueba conteniendo capacidades, velocidades soportadas, etc.

2.9.2 Tramas de Control

Son tramas que dan asistencia a la transferencia entre estaciones inalámbricas, y ayudan a eliminar el problema de nodos ocultos, se pueden observar en la **TABLA IV[11]**.

TABLA IV. TRAMAS DE CONTROL ESTÁNDAR 802.11

Trama	Descripción
Tramas RTS	Implementan la función RTS para salvaguardar la presencia de nodos ocultos.
Tramas CTS	Implementan la función CTS para salvaguardar la presencia de nodos ocultos
Trama ACK	Implementan la función de confirmación de recepción de tramas de datos sin error.

2.9.3 Tramas de datos

Son las tramas que transportan la información entre los nodos y los AP[11].

2.10 Forma de Funcionamiento de las redes Ad-Hoc

Para el correcto funcionamiento de una red Ad-Hoc se debe tener en cuenta algunas propiedades entre las que se puede mencionar las siguientes[9]:

- **Escalabilidad:** El desempeño de la red debe mantenerse a medida que se agregan nuevos nodos a la red. El intercambio constante de información asociado a los cambios de la topología de la red conlleva un importante uso del ancho de banda disponible; adicionalmente, factores tales como ruido e interferencia, acceso múltiple y desvanecimiento limitan aún más dicho ancho de banda. Esto es especialmente crítico conforme el número de nodos aumenta: asumiendo una red con N nodos dotados de antenas omnidireccionales, las estrategias de auto organización deben hacer un uso lento de este recurso. Adicionalmente, el intercambio de los mensajes de autenticación necesarios para garantizar la seguridad conforme nuevos nodos son incorporados también conlleva consumo de ancho de banda.
- **Conservación de energía:** Los nodos deben tener la capacidad de apagarse o de reducir su consumo de energía cuando la situación lo permita; en consecuencia, un objetivo de las estrategias de auto-organización es prolongar el funcionamiento de la red mediante una administración adecuada de la energía de las baterías.
- **Sincronización:** En muchas situaciones es necesario que los nodos de la red dispongan de una misma referencia de tiempo. Este es un problema resuelto en las redes cableadas convencionales, sin embargo es un reto vigente en las redes ad-hoc.
- **Adaptabilidad:** El desempeño de la red debe mantenerse cuando los nodos o los enlaces entre los mismos dejen de operar por alguna razón (agotamiento de las baterías, destrucción del nodo, ocultamiento, etc.)
- **Auto-reparación:** Debe ser capaz de detectar, localizar y reparar fallas de manera automática; estas fallas pueden deberse a desaparición de nodos, pérdida momentánea de los enlaces, sobrecargas, etc.
- **Auto-optimización:** Debe tener la capacidad de escoger los parámetros óptimos de funcionamiento de la red en respuesta a los cambios en las condiciones ambientales (por ejemplo, el cambio en el número de nodos).

2.11 Tipos de escenarios para Redes Adhoc

La variación del tipo de escenarios viene dado por el número de nodos que lo conforman y otras características que pueden modificarse dependiendo de las condiciones que se quiera establecer dentro del escenario, además se puede clasificar de acuerdo al uso que puede ser VANET(Redes Vehiculares), Domésticas(Domótica), etc.[12].

- **Escenario Pequeño.**- Este tipo de escenario nos permite simular gran cantidad de situaciones de redes MANET. Por ejemplo podrían ser redes creadas dentro de cafeterías, museos, salas de exposiciones o conferencias o incluso una escuela o universidad. Este escenario se caracteriza por tener pocos nodos y con una baja movilidad ya que se tratan de personas que van andando o están quietas.
- **Escenario Mediano.**- Este segundo escenario tiene unas dimensiones bastante mayores al anterior. Como ejemplos de redes MANET que puedan encajar con este modelo de escenario, pueden ser: aeropuertos, campus universitarios, congresos, etc. Estos entornos ya son mayores y con mucha más gente, por lo tanto aumentará el número de nodos. La movilidad será parecida al del caso anterior, personas andando o paradas.
- **Escenario Grande.**- Características Mayores a las de los escenarios anteriores con más de 150 nodos, se utiliza principalmente en la unión de dos o más edificios. La movilidad será parecida al del caso anterior, personas andando o paradas, pero se hace un poco más compleja ya que se conforma en su mayoría por nodos móviles, y pocos estáticos.

3. Network Simulator NS-3

Ns-3 es software libre, se ofrece bajo la versión 2 de la GNU General Public License. Cuenta con dos versiones ns-2 y ns-3 que en general son incompatibles. Comenzó a desarrollarse en 1989 como una variante del simulador de red REAL. En 1995, Ns-3 había ganado el apoyo de DARPA, el proyecto Vint de LBL, Xerox PARC, UCB y USC/ISI[13].

Ns-3 ahora es desarrollado en colaboración entre una serie de investigadores e instituciones, incluida la SAMAN (con el apoyo de DARPA), CONSER (a través de

la NSF), y ICIR (antes ACIRI). Sun Microsystems y la UCB Daedelus y Carnegie Mellon también han aportado grandes contribuciones[13].

Es un simulador de eventos discretos de red, que tiene como principales objetivos lograr un mayor énfasis en los niveles 2 y 4 del OSI. En relación con la usabilidad y adaptabilidad, NS-3 está escrito solo en C++. Las plataformas que soporta esta herramienta, tanto de escritorio como de servidor, son: i386, x86-64, Linux, OS X, FreeBSD, Solaris y Windows. Usa NAM para la animación del escenario de red, que es parte de NS-2 desde el principio, y ha logrado hacer un extendido en ambientes Wireless para la generación de resultados[13].

Se usa principalmente en ambientes educativos y de investigación. Permite simular tanto protocolos unicast como multicast y se utiliza intensamente en la investigación de redes móviles ad-hoc. Implementa una amplia gama de protocolos tanto de redes cableadas como de redes inalámbricas. La versión actual, ns-3, está diseñada para soportar todo el flujo de trabajo de la simulación desde la configuración hasta la recolección y análisis de tramas[13].

3.1 Comparativa entre Simuladores de Red

A continuación la **TABLA V**, ayuda a despejar algunas dudas de porque se utiliza Ns3, muestra las diferentes características y en qué plataformas están soportadas, en general refleja el estatus de ns-3[14]:

TABLA V. CARACTERÍSTICAS DE NS VERSIÓN 3 SEGÚN LA PLATAFORMA

Descripción	Linux	FreeBSD	Mac OS X
Construcción optimizada	X	X	X
Enlaces de Python	X	X	X
Enhebrado	X	X	X
Simulador en tiempo real	X	X	
Dispositivo Emulador de Red	X		

Tap-Bridge	X		
Cuna de Simulación de Red	X		
Construcción en estática	X	X	X

Algunas recomendaciones para la elección de un simulador de Red[14]:

- Tipo de licencia (Soporte, documentación)
- Uso específico del simulador(módulos, ajuste a requerimientos)
- Generación y procesamiento de resultados

En la siguiente **TABLA VI** se hace referencia a los diferentes tipos de características a tomar en cuenta en la elección de un simulador de red, y la comparativa entre los simuladores existentes[14].

TABLA VI. ELECCIÓN DE UN SIMULADOR DEL RED

Valor	OPNET	OMNET	NS2	GNS3	NS3	NC-TUNS
Uso investigativo	Alto	Alto	Alto	Bajo	Alto	Alto
Tipo de licencia	Comercial	Libre	Libre	Comercial/ Libre	Libre	Libre
Curva de aprendizaje	Alto	Alto	Alto	Bajo	Alto	Alto
Plataforma de soporta	Windows, Unix	Windo ws, Unix	Windows , Mac, Unix	Windows, Mac, Unix	Windows , Mac, Unix, Linux	Linux
Interfaz Gráfica	Alto	Medio	Bajo	Alto	Medio	Bajo
Graficación de resultados	Buena	Acepta ble	No tiene	Limitado	Alto	Acepta ble

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	Bueno	Nulo	Alto	Aceptable

Ns-3 es utilizado principalmente para la simulación de[14]:

- Redes Ad-Hoc
- Redes WiMax, Wifi, Lte
- Redes de Sensores
- Redes fijas TCP/IP

Las trazas Clases en ns3 tienen la siguiente arquitectura que se clasifica de la siguiente manera[14]:

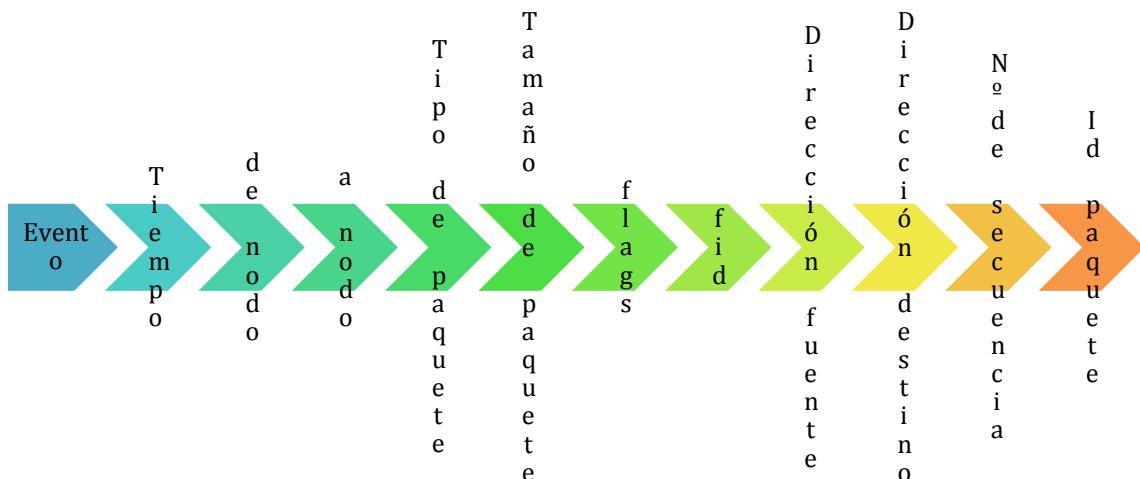


Figura 15: Estructura de un fichero de traza en Ad-Hoc.

Donde cada campo se refieren a[14]:

- **Evento:** Se refiere al evento y puede tomar valore r recibido, d eliminado, + colocado en la cola, - sacado de la cola.
- **Tiempo:** valor de medida
- **De node:** nodo de donde viene el paquete, no necesariamente el de origen

- **Hacia node:** siguiente nodo al que va el paquete
- **Tipo de paquete:** Clasificación
- **Tamaño del paquete**
- **Flags:** banderas
- **Fid:** Id del flujo
- **Dirección fuente:** Nodo de origen de paquete, el formato es nodo-puerto
- **Numero de secuencia:** Para el enrutamiento
- **Id del paquete**

Con Ns3 se manejan una serie de clases que hacen referencia a las capas como si haríamos una instalación en una red de infraestructura en un laboratorio, en la figura 16 se describe las clases de acuerdo a la capa que le corresponde[15]:



Figura 16: Aplicación por capas de acuerdo a la clase en Ns3.

3.2 Métricas de Rendimiento

Durante el proceso de Simulación se obtienen algunos valores como resultado de la eficiencia del enrutamiento los cuales nos servirán para medir el rendimiento de la red en general, los que están definidos y nos orientaran a encontrar en este caso en mejor protocolo[10].

3.2.1 Throughput

Se define como la máxima tasa de transferencia que se puede alcanzar en una transmisión entre elementos de las terminales (end-to-end). En otras palabras es la

cantidad total de información recibida en los nodos receptores por unidad de tiempo y se mide en bits por segundos (bps), su ecuación se aprecia a continuación[10]:

$$\text{Throughput} = \frac{\text{Datos entregados exitosamente}}{\text{Tiempo en la transmisión}} [\text{kbytes/s}]$$

Figura 17: Formula para el cálculo de Throughput.

3.2.2 Tasa promedio de entrega

La tasa promedio de entrega es la fracción de paquetes que llegan a su destino por el número total de paquetes enviados en la red, regularmente se expresan en porcentaje, la ecuación para el cálculo es la siguiente[10]:

$$\text{Tasa de entrega} = \frac{\text{Paquetes recibidos}}{\text{Paquetes enviados}} \times 100$$

Figura 18: Formula para el cálculo de Tasa de entrega.

3.2.3 Retardo promedio de extremo a extremo

El retardo promedio extremo a extremo se expresa como el tiempo promedio que requiere todos los paquetes de datos que han sido entregados al destino, donde el retardo del paquete i-ésimo se expresa como $T_{rj} - T_{sj}$. Asumiendo que j paquetes entregados a su destino, mostrando en la siguiente ecuación la manera de calcularlo[10]:

$$\text{Retardo} = \frac{\sum j T_{rj} - T_{sj}}{\text{Paquetes Entregados}} [s]$$

Figura 19: Formula para el cálculo de Retardo.

e. Materiales y Métodos

1. Métodos

El presente Proyecto de Fin de carrera se lo desarrolló utilizando diferentes métodos científicos de investigación y recolección de bibliografía acorde al tema, con el fin de obtener la información y cumplir con los objetivos planteados, para lo cual se utilizaron los métodos científicos descritos a continuación:

Método Inductivo.- Este método permitió encontrar los problemas tanto en redes Adhoc, como en los que causan inestabilidad en este tipo de redes y mediante los cuales se puede ofrecer una solución al problema planteado.

Método Deductivo.- Con el cual se obtuvo la información necesaria para el desarrollo del proyecto. Este método se lo utilizó en la fase de recopilación de la información, que permitió esclarecer conceptos; así también como para determinar el objetivo general del proyecto de titulación, para permitir el acoplamiento de los diferentes protocolos a un mismo escenario cumpliendo con los requisitos y características determinadas en la topología, además permitió especificar cuáles serán los escenarios de estudio para resolver el problema de investigación.

Método Científico.- Se utilizó en la obtención y análisis de información, además en la etapa de resultados para el análisis de las métricas obtenidas y que de esta manera pudieron ofrecer conclusiones a partir de este análisis.

2. Técnicas

Para la recolección de la información útil para el desarrollo del proyecto, se ha considerado el uso de las siguientes técnicas de investigación:

Entrevista: Se utilizó esta técnica para la obtención de información de fuentes seguras, como lo son doctores en el área de Redes y Telecomunicaciones obteniendo de esta manera claridad en sobre la situación actual de este tipo de Redes.

Análisis documental: Se utilizó con el fin de conocer y completar la estructura del proyecto, mediante el análisis de información cualitativa y cuantitativa brindando de esta manera un soporte firme para el desarrollo y propuesta de soluciones.

3. Metodología

El presente Proyecto de Fin de Carrera se lo desarrolló utilizando, lineamientos propuestos por la autora ya que no existen metodologías definidas para el desarrollo de proyectos en Redes, por adaptarse de mejor manera a los distintos escenarios propuestos y al desarrollo en general del proyecto, cuyas fases se obtuvieron con la guía del director de tesis, las que permitieron definir un proceso ordenado para dar solución al problema del proyecto de titulación, la misma que está compuesta por las fases que se detallan a continuación:

Análisis de las Redes AdHoc.- En esta fase se comienza por dar respuesta a un grupo muy concreto de interrogantes, que permitirán definir los parámetros a considerar en los escenarios, las mismas que son:

¿Qué son las Redes Adhoc y cómo funcionan?

¿De qué manera funciona el enrutamiento en las redes AdHoc?

¿Qué son los protocolos de enrutamiento y que diferencias existen entre los protocolos regulares?

¿Qué especificaciones deben cumplir las topologías para adaptarse a un escenario Adhoc?

¿Cómo puedo medir el rendimiento de la red, utilizando protocolos Adhoc?

¿En base a qué se puede seleccionar un mejor rendimiento?

¿Esta solución provoca mejoras en la estabilidad de redes AdHoc?

Fase que determina el objetivo general y los específicos del proyecto de titulación, que brindan una visión clara de las expectativas de rendimiento de una Red AdHoc.

Fase de Modelado de la Topología.- Esta fase comprende la definición de los requerimientos técnicos de la red en general, estableciendo específicamente ampliación y numero de nodos en la red, además de los protocolo y un estudio a fondo de cómo funcionan estos protocolos, resaltando que se hizo la elección de dos escenarios en donde se aplicaron cuatro protocolos entre proactivos y reactivos, dándonos pautas para los resultados finales.

Fase de Simulación y evaluación.- En esta etapa se establece la instalación, configuración de la herramienta, así como de la creación de código desarrollando la Red especificada y obteniendo las métricas que ayudaron en el planteamiento de soluciones., mediante la denominada programación de redes.

En esta fase también se realizaran pruebas para poder obtener resultados a través de las distintas pruebas que asegurar el perfecto funcionamiento en la configuración, obteniendo las métricas necesarias para el análisis, evaluación y propuesta del mejor protocolo dependiendo de su rendimiento en las simulaciones, lo que garantiza un mejor envío de paquetes y por lo tanto proporciona más estabilidad en la Red.

f. Resultados

Fase 1: Análisis de las Redes Ad-Hoc

1.1 Estadísticas

En el último censo a nivel nacional en el proyecto denominado “Tecnologías de la Información y Comunicación TIC's 2013” realizado por el Instituto Nacional de Estadísticas y Censos, se puede obtener datos estadísticos el uso, frecuencia, tipo de red utilizada, etc. Teniendo en cuenta que el uso y penetración de las redes inalámbricas va creciendo constantemente con más fuerza gracias al uso del internet y de los dispositivos inalámbricos[16].

En los siguientes gráficos se puede visualizar un análisis de los años 2010, 2011, 2012 y 2013 los cuales son el principal motivador para el estudio de este proyecto y que da la pauta de importancia para la simulación y análisis de los protocolos de redes Inalámbricas, en este caso las redes Manet o redes AdHoc, con la finalidad de aportar a la estabilidad en este tipo de redes que avanza día a día[16].

- **Equipamiento tecnológico del hogar a nivel nacional**

La figura 20 muestra que el 18,1% tienen al menos un computador portátil, 9,1 puntos más que lo registrado en 2010. Y un 27,5% de los hogares tiene computador de escritorio, 3,5 puntos más que en 2010. Esto refleja que las redes inalámbricas, viene creciendo tomando como referencia la adquisición de computadores portátiles, que usan frecuentemente dispositivos para conexión inalámbrica[16].

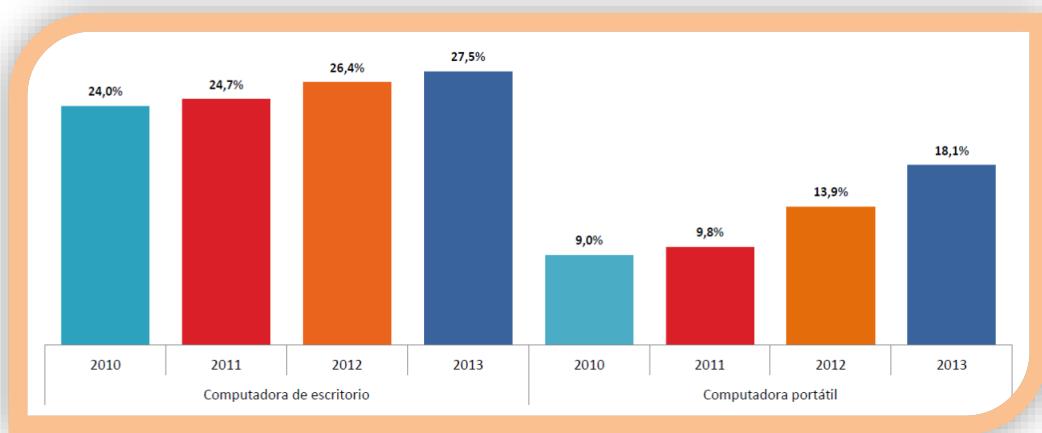


Figura 20. Equipamiento tecnológico en el hogar a nivel nacional.

- **Hogares que tienen teléfono fijo y/o celular a nivel nacional.**

En la figura 21 se observa que el 86,4% de hogares posee al menos un teléfono celular, 36,7 puntos más que lo registrado en el 2010. Lo que sigue manteniendo la opinión de que las redes inalámbricas están en auge[16].

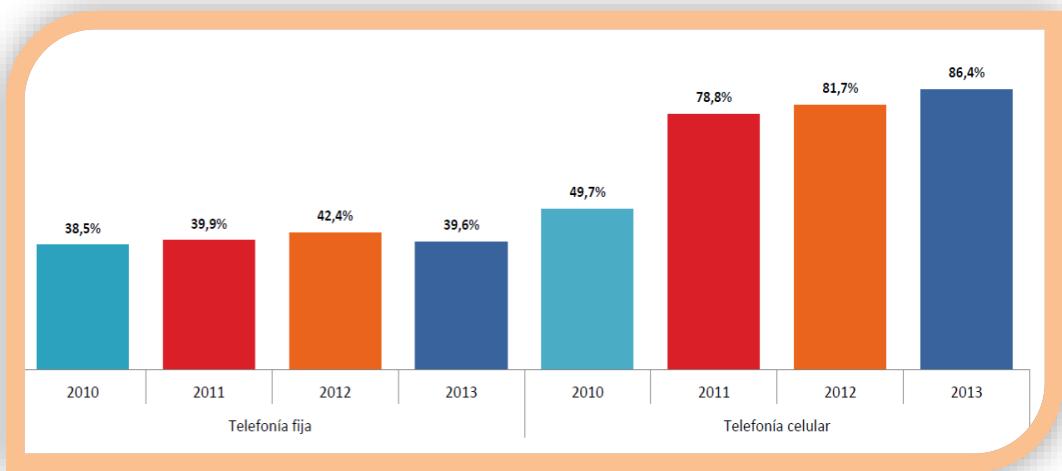


Figura 21. Nivel de crecimiento de la adquisición de celulares en el hogar.

- **Acceso a internet según el área**

En la figura 22 se ubican los porcentajes de cobertura de internet de acuerdo al área siendo el 28,3% de los hogares a nivel nacional quienes cuentan con acceso a internet, creciendo tanto en zonas rurales como urbanas, 16,5 puntos más de crecimiento a nivel nacional[16].

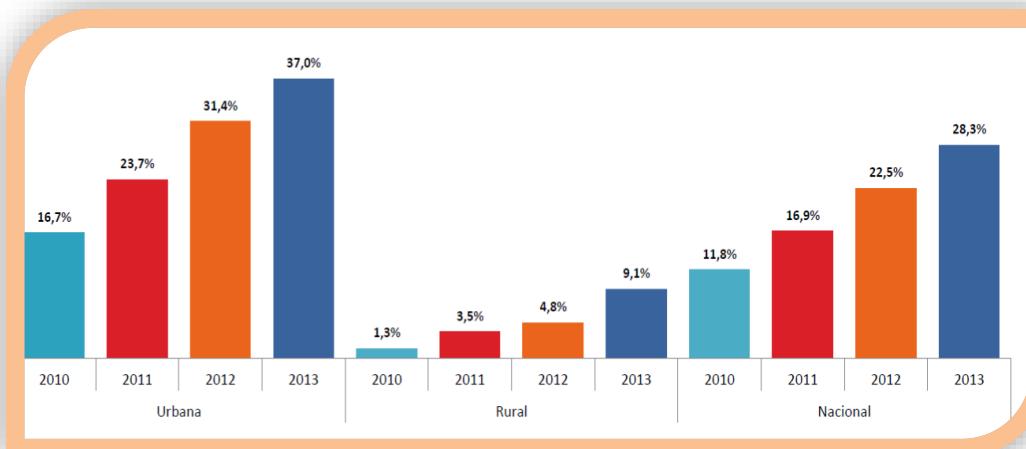


Figura 22. Porcentaje de acceso a internet según el área.

- **Hogares que tienen acceso a internet a nivel nacional**

En la figura 23 se pueden evidenciar el crecimiento de las redes a nivel inalámbrico, lo que no ha ocurrido con las demás tecnologías como son modem y cable. El 28,3% de los hogares tienen acceso a internet, entre ellos el 31,4% acceden por medios inalámbricos y que desde el 2010, ha venido creciendo de una manera imponente[16].

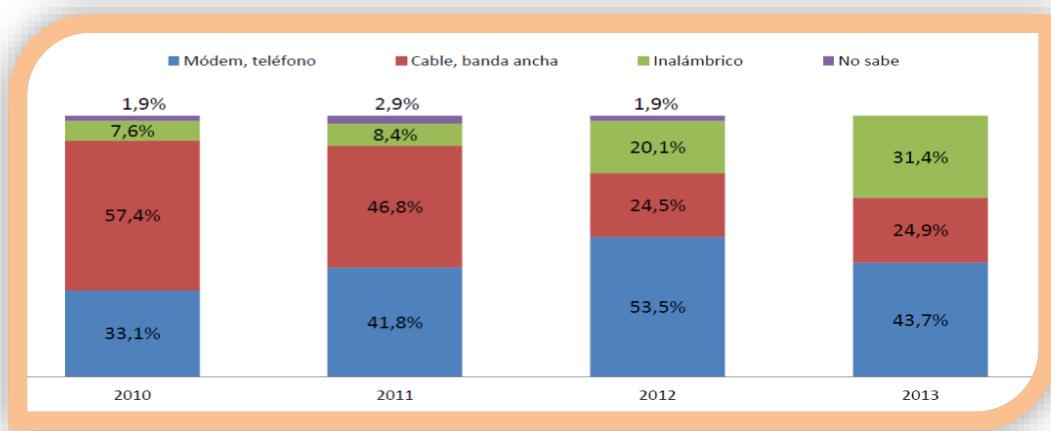


Figura 23. Uso de las tecnologías para acceso a internet.

- **Porcentaje de personas que han utilizado internet en los últimos 12 meses**

En la figura 24 se puede apreciar que el 40,4% de la población de Ecuador ha utilizado internet en los 12 últimos meses. En el área urbana el 47,6% frente al 25,3% del área rural[16].

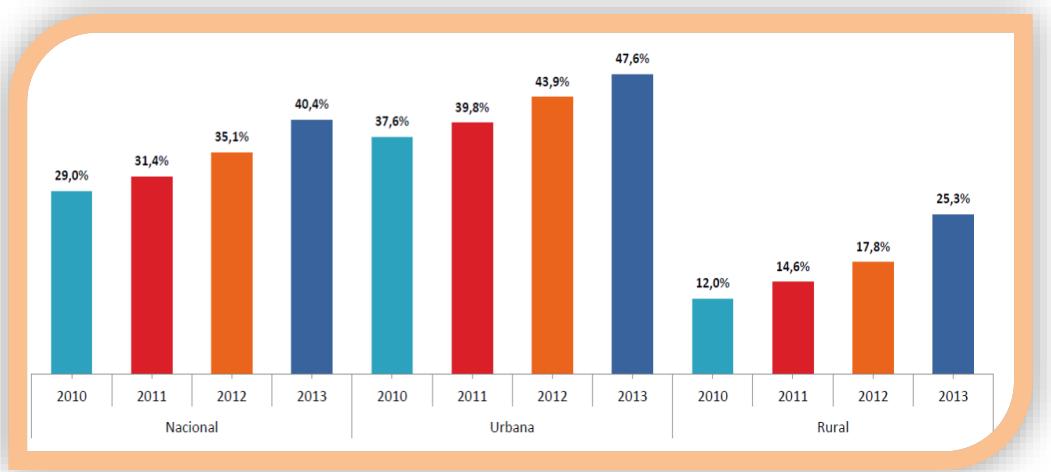


Figura 24. Uso de internet en los 12 últimos meses.

- **Lugar de uso de Internet por área.**

En la figura 25 se puede apreciar que el lugar de uso del internet es con gran afluencia el de centro de acceso público que por lo general es mediante red inalámbrica[16].

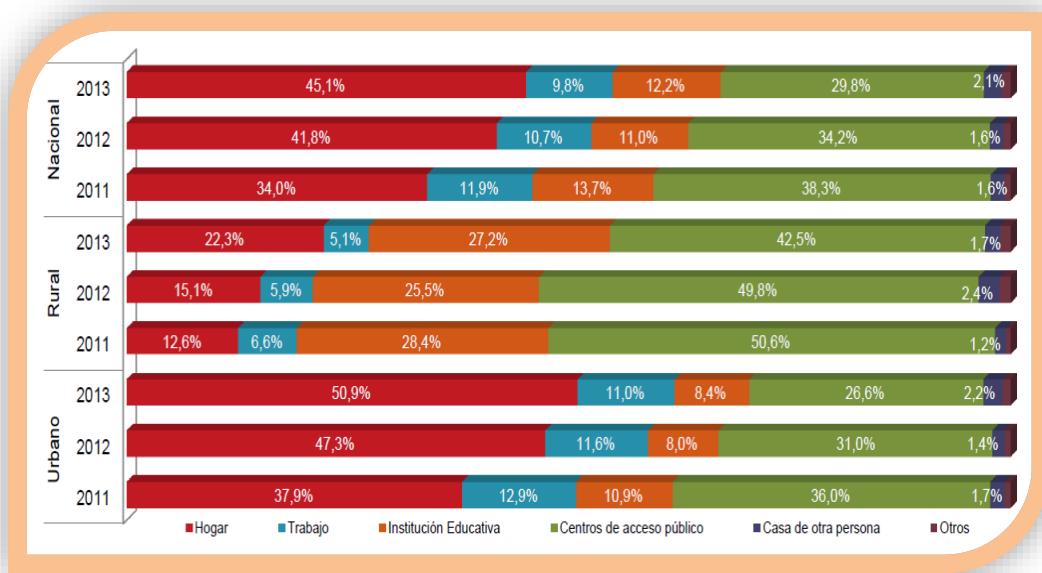


Figura 25. Lugar de uso de Internet.

- **Frecuencia de uso de Internet a nivel nacional**

Luego de constatar el crecimiento que tienen las redes inalámbricas, se puede decir que el tema de la estabilidad es de gran importancia, además debido al uso frecuente que tiene, siendo el 64,0% el porcentaje de personas en 2013 que usan diariamente internet según la figura 26 [16].

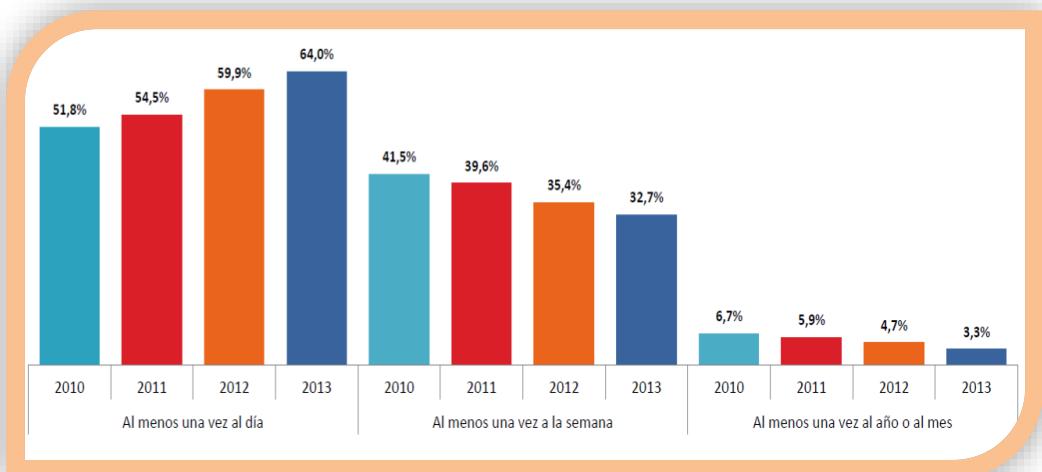


Figura 26. Frecuencia de uso de Internet.

- **Porcentaje de personas que tienen teléfono celular activado a internet por provincia.**

En la figura 27 se puede apreciar que la provincia con mayor número de teléfonos celulares activados a internet es Pichincha con un 60,9% y la que tiene menor porcentaje con un 37,4%[16].

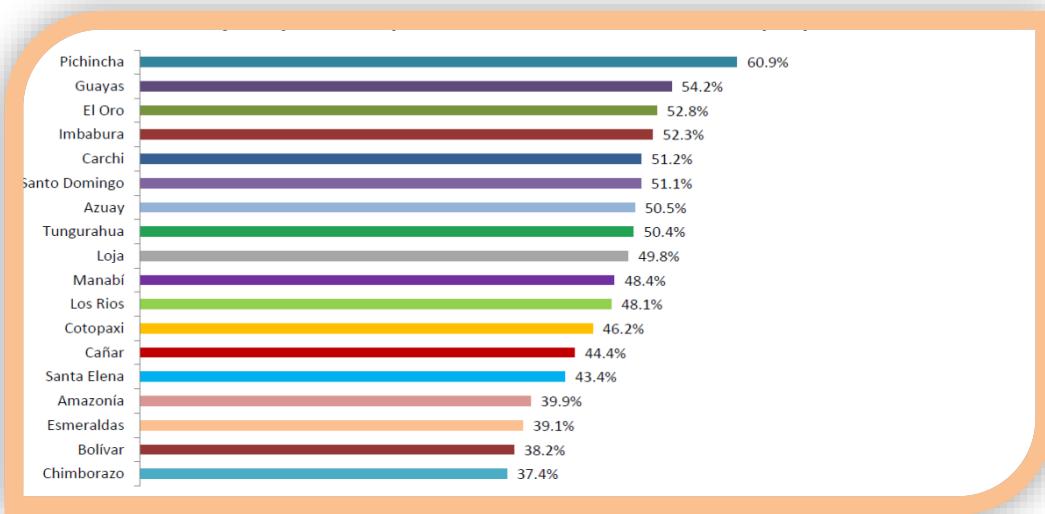


Figura 27. Porcentaje de personas con celular activado a internet.

- **Porcentaje de personas que tienen teléfono inteligente a nivel nacional**

El porcentaje de personas con teléfono inteligente viene creciendo desde 2011 como lo muestra la figura 28 llegando desde 8,4% a 16,9% de crecimiento en dos años, lo que usualmente hace suponer que también usan las redes inalámbricas[16].

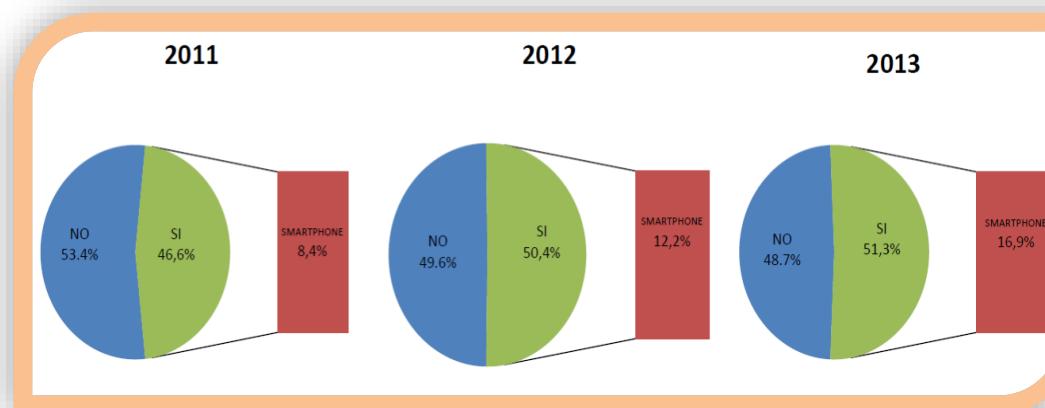


Figura 28. Porcentaje de personas con celular inteligente.

1.2 Principales causas de inestabilidad detectadas por enrutamiento en redes Ad-Hoc.

En esta etapa se reconoció el problema que contribuye a la inestabilidad en las Redes Ad-Hoc, por sus características propias y por el tipo de enrutamiento utilizado. Enfocándose precisamente en que los problemas de fondo por falta de estabilidad se ocasionan por variantes en la topología que es muy común en este tipo de redes, lo que a su vez causa el llamado problema de los nodos ocultos que es el principal causante de inestabilidad, ocurre cuando al creer que un canal está libre, pero en realidad está ocupado por otro nodo que no escucha, causa de esta manera que los paquetes colisionen y no se obtenga respuesta de recepción del paquete[17].

El ejemplo a continuación detalla de mejor manera el problema de nodos ocultos, en la figura 29, la estación A transmite a la estación B, la estación C no puede determinar el estado del canal entre A y B y por lo tanto transmite a B, de esta manera hay interferencia en el canal y colisionan los paquetes transmitidos[17].

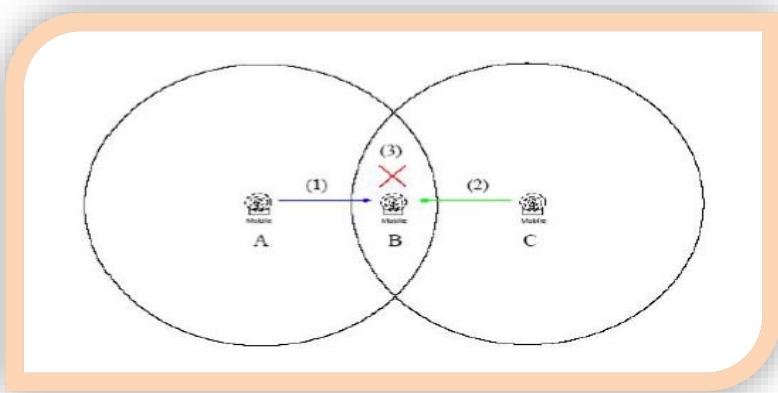


Figura 29: Problema de Nodos Ocultos en Manets.

En temas de forma se determina que las características de los dispositivos también son una fuente de inestabilidad en este tipo de redes, características como que ocupan un ancho de banda limitado, lo que no ocurre de forma muy frecuente en redes con infraestructura, y en la obtención de energía para su funcionamiento ya que por lo general obtienen su energía de una batería lo que afecta de forma general en el uso de su potencia máxima[17].

La inestabilidad además radica en la forma de enrutamiento que despeja las dudas sobre si el tipo de protocolo afecta la estabilidad, ya que en métricas muy bajas como es el retardo, numero de paquetes recibidos, el nivel máximo de transmisión o tasa máxima de transmisión pueden variar tanto por el problema de nodos ocultos como por las características de hardware y de software de los dispositivos inalámbricos.

Dentro de este punto se ha decidido, hacer uso de dos escenarios ya que el factor determinante dentro de una topología Ad-Hoc es el número de nodos, agregadas a estas características propias de la topología, y a cuatro protocolos que serán los que se usan con mayor frecuencia, dos proactivos y dos reactivos.

Fase 2: Modelado de la Topología

2.1 Selección de Escenarios

Para la simulación de las redes Ad-Hoc, se ha tomado en cuenta dos escenarios uno pequeño y otro mediano, con las mismas características de topología y los protocolos utilizados los cuales se detallan en la **TABLA VII** y **TABLA VIII**, ya que se desea apreciar el trabajo que hace cada protocolo de acuerdo en este caso al tráfico que se genera, lo que varía gracias al número de nodos en la red, en el primer escenario con 25 nodos y el segundo con 50, esperando poder apreciar las diferencias en su rendimiento para su posterior análisis y propuesta de soluciones.

TABLA VII. DESCRIPCIÓN DEL ESCENARIO DE PRUEBA UNO.

Parámetro	Valor
Cantidad de nodos	25
Dimensión del escenario	300x1500m
Tiempo total de simulación	200s
Tamaño de paquetes	1024b
Tasa de trasmisión	1mb/s
Cantidad de paquetes	1000
Protocolo	AODV-DSR-DSDV-OLSR

TABLA VIII. DESCRIPCIÓN DEL ESCENARIO DE PRUEBA DOS.

Parámetro	Valor
Cantidad de nodos	50
Distancia máxima entre nodos	300x1500m
Tiempo total de simulación	200s
Tamaño de paquetes	1024b
Tasa de trasmisión	1mb/s
Cantidad de paquetes	1000
Protocolo	AODV-DSR-DSDV-OLSR

2.2 Selección de Protocolos

En las tablas anteriores se detallaron cuatro protocolos los cuales fueron utilizados en los dos escenarios, ya que ayudan al problema de nodos ocultos por su algoritmo de enrutamiento, son los más utilizados en Redes Ad-Hoc, dos de estos son proactivos y dos reactivos los que se pueden observar en la figura 30. A demás cabe recalcar que de estos protocolos seleccionados, OLSR, DSDV y AODV tiene módulos definidos en Ns-3 y DRS se implementa con clases que pueden utilizarse para su simulación, y estos son los soportados por el simulador Ns-3[18].

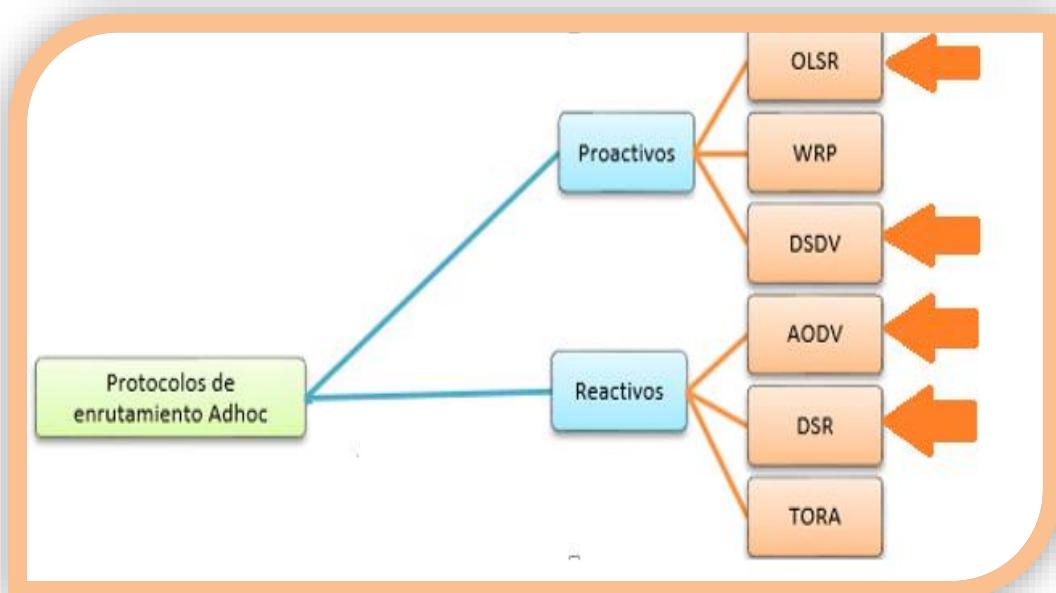


Figura 30: Protocolos seleccionados.

A continuación se realiza una comparativa de las características entre los protocolos de enrutamiento para redes Ad-Hoc, en donde se precisa explicar algunos criterios a tomar en cuenta para la selección de los mismos, además cabe resaltar que son los soportados por Ns-3, por lo que no se ha tomado en cuenta los demás protocolos en la elección[18]:

- **Métricas de enrutamiento.**- Parámetros que determinan la selección de la ruta.
- **Libre de lazos.**- Define si el protocolo es capaz de evitar la ocurrencia de lazos en la red.
- **Escalabilidad.**- Define si el protocolo permite la inserción de nuevos nodos en la red sin que se afecte su rendimiento.
- **Confiabilidad.**- Define si el protocolo garantiza la entrega de la información a su respectivo destino.
- **Balance de carga.**- Proceso de distribuir la carga y el tráfico uniformemente en la red para evitar sobrecargas.
- **Control de congestión.**- Proceso de asignación de los recursos de red, de manera que pueda operar a un nivel de rendimiento aceptable cuando la demanda excede o es cercana a la capacidad de los recursos, como ancho de banda, espacio en memoria, capacidad de procesamiento, etc.
- **Troughput.**- Razón promedio de entregas exitosas de paquetes en un canal de comunicación.
- **Implementación.**- Desarrollo de versiones de los protocolos para plataformas de computadoras y dispositivos de comunicación.

En la **TABLA XI**. Se detalla las características anteriormente descritas para realizar una comparativa entre los protocolos de Enrutamiento para redes Ad-Hoc[18].

TABLA XI. COMPARACIÓN ENTRE PROTOCOLOS SELECCIONADOS

Protocolo	Métrica más utilizada	Libre de lazos	Escalabilidad	Confiabledad	Balance de Carga	Control de congestión	Throughput	Implementación
OLSR	Saltos(Camino más corto)	Si	No	Si	No	No	Disminuye con la movilidad	Linux, Android,Ns-3
AODV	Saltos(Camino más corto)	Si	No	Si	No	Si	Disminuye con la movilidad	Ns-3, c++, Matlab, Linux.
DSR	Saltos(Camino más corto)	Si	No	Si	No	Si	Disminuye con la movilidad	Ns-3, Linux
DSDV	Saltos(Camino más corto)	Si	No	No	No	Si	Disminuye con la movilidad	Ns-3, Linux

2.3 Evaluación de Herramientas para Simulación

En esta sección la **TABLA X** resalta las características principales para tomar en cuenta en la elección de un simulador de red, destaca el simulador OPNET como el mejor simulador ya que tiene valores altos en todas las características pero la razón por la que no se lo ha seleccionado es que tiene una licencia de tipo comercial y no cuenta con una versión libre que es lo que actualmente se trata de fomentar. GNS3 se elimina en primera instancia ya que presenta valores no competitivos casi en todas sus características, luego de esto se elimina por comparación NS-2 por valor bajo en interfaz gráfica y de resultados, así mismo NC-TUNS por interfaz gráfica y valor aceptable en soporte de tecnologías inalámbricas[19].

Quedando para la siguiente comparativa OMNET y NS-3 las dos con valores iguales en Uso investigativo, tipo de licencia, curva de aprendizaje, Interfaz gráfica, tráfico a modelar, pero siendo una pauta para la elección de NS-3 las características que resaltan como:

- **Plataforma que soporta.**- Siendo aplicable en Windows, Unix, **Mac, Linux**
- **Graficacion de resultados.**- Con valor **alto**, a diferencia de OMNET que tiene aceptable.

- **Soporte estándar 802.11.**- Siendo Ns-3, el software libre con valor alto en esta característica y razón principal para su elección, ya que permite la simulación de redes inalámbricas ampliamente.

TABLA X. MÉTRICAS PARA ELECCIÓN DEL SIMULADOR

Valor	OPNET	OMNET	NS2	GNS3	NS3	NC-TUNS
Uso investigativo	Alto	Alto	Alto	Bajo	Alto	Alto
Tipo de licencia	Comercial	Libre	Libre	Comercial/ Libre	Libre	Libre
Curva de aprendizaje	Alto	Alto	Alto	Bajo	Alto	Alto
Plataforma de soporta	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
Traficó que permite modelar	Alto	Alto	Medio	Bajo	Alto	Alto
Soporte estándar 802.11	Alto	Medio	Bueno	Nulo	Alto	Aceptable

2.4 Características principales de ns3

El Simulador Ns3 no es compatible con su versión anterior Ns2. Está escrito en su mayoría en lenguaje C++, adicional a este el lenguaje Python que también ha aportado para su desarrollo, actualmente este simulador genera archivos tipo PCAP, los cuales pueden ser leído o interpretado por un capturador de paquetes, entre los más conocidos está el Wiresharkg[20].

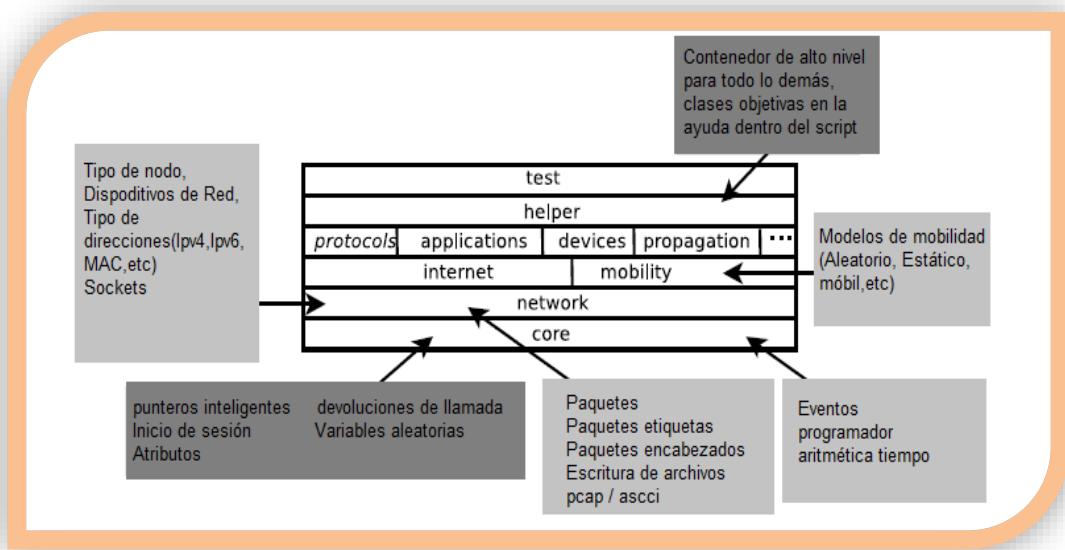


Figura 31: Organización del simulador Ns3.

En la figura 31 se aprecia la organización de Ns3 y el funcionamiento de cada módulo, el núcleo de la simulación se lleva a cabo en el directorio `src/core`, y la base se utiliza para construir el motor de simulación `src/simulator`, estos tres primeros módulos están destinados a formar un núcleo genérico de simulación que puede ser utilizado por diferentes tipos de redes, no basadas en internet[20].

En el tercer nivel están los más importantes Nodo y Movilidad, los que serán contenidos dentro de *DevicesContainers*, encargados de guardar los datos configurados en los nodos y dispositivos inalámbricos, y para el control de movilidad manejan módulos específicos que catalogan a los nodos como móviles o fijos[20].

Dos tipos de dispositivos de redes especiales trabajan para apoyar la configuración de la red en diferentes etapas de la simulación, en conjunto con los modelos asociados con Internet incluida en el API de sockets utilizados por las aplicaciones de internet[20].

Y por último los auxiliares y de análisis, que contienen comandos fundamentales para que el núcleo pueda desarrollar las aplicaciones configuradas en los niveles superiores.

2.5 Tecnologías permitidas en ns3

Esta herramienta permite el desarrollo de modelos de alto desempeño, lo que habilita el uso de la herramienta como emulador, soporta simulaciones de Redes o tecnologías tales como[20]:

- Sobre IPV4
- Sobre IPV6
- Wireless (Adhoc, Wifi, WiMax,etc)
- Algoritmos de enrutamiento (OLSR, AODV, etc)

Fase 3: Simulación y Evaluación

3.1 Simulación

Mediante la herramienta de simulación Ns3, se obtuvo resultados con los que luego de un análisis, se procedió a sacar conclusiones, el porqué del uso de un simulador es que gracias a este se puede realizar experimentación sin trabajar directamente con un sistema real, o en ocasiones cuando no se cuenta con dicho sistema.

Pero que por otro lado recrean gran parte de la realidad, existe un campo amplio del uso de simuladores como ejemplo de ellos están:

- Creación de nuevos fármacos.- simular interacción entre compuestos químicos
- Simuladores de vuelo.- para pilotaje de aviones
- Seguridad.- Impacto de choques y seguridad de los ocupantes.

Dentro del área informática se puede destacar su uso en:

- Sistemas distribuidos
- Redes de ordenadores
- Bases de datos
- Multiprogramación

Es el paso previo a la implantación del sistema en este caso de red, que lo realizamos en vista de costos de viabilidad y complejidad del proyecto, determinando la factibilidad del estudio en un entorno simulado[21].

3.2 Especificación de Software

El software a utilizar para el desarrollo del proyecto de titulación es el que se describe en la **TABLA XI**, acompañado de algunas dependencias para trabajar con Ns3.

TABLA XI. ESPECIFICACIÓN SOFTWARE

Parámetro	Software	Versión
Sistema Operativo	Xubuntu	14.04.02
Simulador de redes	NS3	3.23
Herramienta de Análisis y Animación	NetAnim	3.106
Herramienta de Análisis de datos	Wireshark	1.10.6
Dependencias para Ns-3		
Python	Requerimiento mínimo para trabajar con C++, Python y NS3	
C++	Requerimiento mínimo para trabajar con C++ y NS3	
Mercurial	Requerimiento para trabajar con los repositorios de desarrollo de NS3	
QT4-devel	Necesario para trabajar con el animador NetAnim	

3.3 Especificación de Hardware

En la **TABLA XII** se aprecia las características a utilizar en cuanto al hardware necesario para trabajar en el proyecto de titulación.

TABLA XII. ESPECIFICACIÓN HARDWARE

Parámetro	Hardware
Equipo	Laptop Toshiba, Satellite
Memoria Ram	12 GB DDR3
Procesador	i5-3210m 2.5 Ghz
Disco Duro	100 GB

3.4 Selección de equipo

El presente proyecto de fin de carrera se lo realizó en conjunto, para poder obtener resultados de calidad al finalizar el proyecto. El presente proyecto es realizado por la postulante, y el equipo que interviene está conformado por profesionales dentro de la rama de redes y telecomunicaciones, que serán un gran soporte al momento de ahondar en el tema. El equipo se detalla en la **TABLA XIII**.

TABLA XIII. EQUIPO DE TRABAJO

Integrante	Actividad
Gabriela Rodríguez	Diseño, codificación y análisis de la Red.
Ing. Alex Padilla	Director y tutor del proyecto
Ing. Hernán Samaniego	Conocedor del tema sobre proyectos en Ns3

3.5 Costos

Los requisitos económicos que se presentan en la tabla para el presente proyecto de fin de carrera son asumidos en su totalidad por la postulante y se describen en la **TABLA XIV**.

TABLA XIV. COSTOS DEL PROYECTO

Actividad	Duración (horas)	Valor	Total
Análisis de Redes Adhoc	285	10	2850
Modelado de la Topología	150	10	1500
Simulación y Evaluación	354	10	3540
Total:			7890

3.6 Beneficios a obtener

Gracias al estudio descrito en el presente proyecto se dará la pauta para el análisis del rendimiento de red, además de que se puede prestar para implantar tomando en cuenta como base la simulación, se puede poner el caso de estudio de una red Ad-Hoc en la universidad dotando de características deseables para el rendimiento de la red como son:

- Estabilidad en la red real, tomando en cuenta las medidas de desempeño y el escenario de la Universidad que es en gran escala.
- Valores bajos o nulos en métricas como perdida de paquetes, colisiones, retardo en la entrega de paquetes, etc.
- Mayor éxito en la taza de paquetes entregados.

- Menor coste, o número de saltos que dará otro punto extra al rendimiento de la red.

3.7 Módulos de ns3

Diferentes entidades se implementan en diferentes módulos. Cada módulo tiene "atributos". Las funcionalidades de los módulos son proporcionados por diferentes funciones miembro de C++. Hay helper API disponible con cada módulo. Estos métodos Helpers hace que la programación sea más fácil y a nivel bajo de API, se puede utilizar dentro del código y tiene todavía mayor facilidad de[15]:

- Extender fácilmente un módulo para añadir nueva características / funcionalidades
- Es fácil de añadir nuevo módulo

Los módulos que se pueden incluir dentro del código en Ns-3 son:

- Antenna
- AODV
- Applications
- Bridge
- Buildings
- Config-store
- Core
- Csma
- Csma-layout
- DSDV
- DSR
- Emu
- Energy
- Fid-net.device
- Flow monitor
- Internet
- Lte
- Mesh
- Mobility
- Mpi

- Netanim (python)
- Network
- Nix-vector-routing
- OLSR
- Point-to-point
- Point-to-point-layout
- Propagation
- Sixlowpan
- Spectrum
- Stats
- Tap-bridge
- Test (no python)
- Topology-read
- Uan
- Virtual-net-device
- Visualizer
- Wave
- Wifi
- wimax

3.8 Estructura de un script en NS3

A continuación se describe una estructura y funcionamiento de cada sección que conforma un script escrito en C++, para Ns-3[22]:

- **Texto estándar**

En la primera línea se debe especificar el estilo de codificación que se utilizará en el código fuente, además de esto se puede especificar información adicional sobre la licencia de distribución del software Ns3, a continuación se especifica la línea de código[22].

```
/*-*-Mode: C++; c-file-style:"gnu"; indent-tabs-mode:ni;-*-*/
```

- **Módulos de inclusión**

La manera correcta de iniciar la escritura del código es realizando la declaración de los módulos de inclusión como se aprecia a continuación[22]

```
#include "ns3/core/core-module.h"  
  
#include "ns3/simulator-module.h"  
  
#include "ns3/node-module.h"  
  
#include "ns3/helper-module.h"
```

- **Definición de nombres en Ns-3**

La siguiente línea en el script a ser desarrollado es la declaración de nombres, lo que permite realizar un manejo más global de las variables incluso para integrar funciones de otros códigos y lo realizamos de la siguiente manera[22]

Using namespace ns3;

- **Registro o notas**

Se lo realiza con la siguiente línea de código, y se lo utiliza para indicar los procesos que siguen en la documentación del denominado espacio Doxygen que maneja ns3 enfocado en el manejo de registros (logging)[22].

NS_LOG_COMPONENT_DEFINE (“PrimerScript”);

- **Función Principal**

Es similar a la manera de iniciar un programa en lenguaje C++, es aconsejable que durante esta línea se especifique también la especificación sobre los registros (logging) para que pueda dar información sobre la simulación, la declaración de la función principal y los logginn en el programa tendrá la siguiente sintaxis[22]

Int

Main (int argc, char *argv[])

{

LogComponentEnable(“UdpEchoClientApplication”, LOG_LEVEL_INFO);

LogComponentEnable (“UdpEchoServerApplication”, LOG_LEVEL_INFO);

- **Clases para el diseño de topologías**

Existe una gran variedad de clases que se utilizan en el simulador las que permiten realizar los procesos de declaración de variables, tipos de comunicación que se van a realizar y redes de datos a ser configuradas, protocolos, etc.[22].

Las clases más utilizadas se listan a continuación:

- ✓ NodeContainer
- ✓ PointToPointHelper
- ✓ NetDeviceContainer
- ✓ InternetStackHelper
- ✓ Ipv4AddressHelper

- **Aplicaciones**

Existe de igual manera gran número de aplicaciones que se pueden realizar mediante implementación de un script en el cual se manejan protocolos de comunicaciones y formas diferentes de transmisión de datos, las más conocidas son UDP y TCP, a continuación en la figura 32 se describen las aplicaciones más utilizadas en Ns3[22].

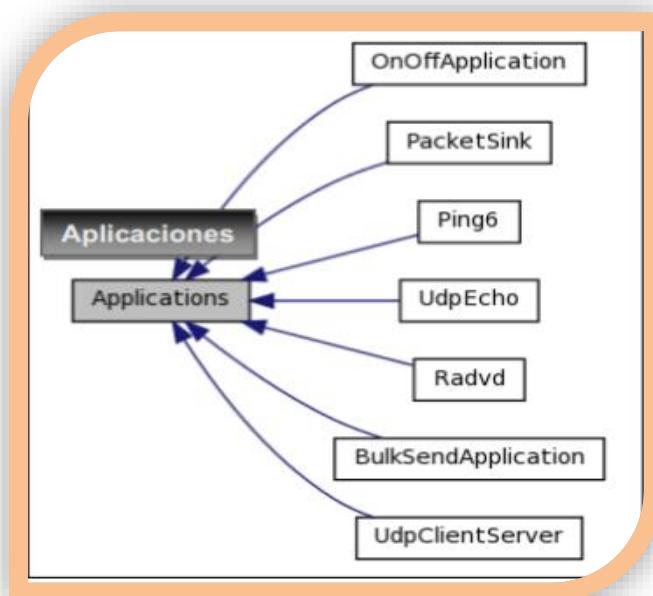


Figura 32: Tipo de Aplicaciones en Ns3.

- **Simulador**

En este punto lo que se busca realizar, propiamente la ejecución del programa que se ha escrito, ejecutando cuando lea esta instrucción todas las tareas programadas siguiendo el orden establecido. Se realiza con la siguiente línea de código.

Simulator::Run();

Después de que todos los eventos hayan sido ejecutados en su totalidad y no existe ningún evento adicional, el sistema regresa a la sentencia de ejecución de ***Simulator::Run();*** Luego de esto tenemos que ejecutar la línea que permite destruir todos los objetos auxiliares que se han creado durante la simulación y que en adelante serán considerados como “elementos basura”. A continuación se describe la línea de código para lo mencionado[22].

Simulator::Destroy

3.9 Análisis en NS-3

El módulo para realizar análisis en Ns3 es ***FlowMonitor*** está diseñado de forma modular y puede ser adaptado por subclases ***ns3 :: FlowProbe*** y ***ns3 :: FlowClassifier***[23]

3.9.1 Información analizada

Por el momento las dos subclases están disponibles para IPv4 e IPv6. Cada subclase clasifica los paquetes en cuatro puntos:

- Cuando se envía un paquete (SendOutgoing IPv [4,6] rastros)
- Cuando se envía un paquete (UnicastForward IPv [4,6] rastros)
- Cuando se recibe un paquete (LocalDeliver IPv [4,6] trazas)
- Cuando se deja caer un paquete (gota IPv [4,6] rastros)

Puesto que los paquetes realizan un seguimiento a nivel IP, cualquier retransmisión causada por protocolos L4 (por ejemplo, TCP) será visto por la subclase como un nuevo paquete. La etiqueta en cada paquete llevará los datos básicos de mismo, útiles para la clasificación del paquete[24].

3.9.2 Datos recogidos

Los datos que recoge **flowmonitor** son los descritos a continuación[24]:

- **timeFirstTxPacket:** cuando se transmitió el primer paquete en el flujo
- **timeLastTxPacket:** cuando se transmitió el último paquete en el flujo
- **timeFirstRxPacket:** cuando el primer paquete en el flujo fue recibido por un nodo de extremo
- **timeLastRxPacket:** cuando se recibió el último paquete en el flujo
- **delaySum:** la suma de todos los retrasos de extremo a extremo para todos los paquetes recibidos de la corriente
- **jitterSum:** la suma de toda demora jitter de extremo a extremo (variación del retardo) valores de todos los paquetes recibidos del flujo.
- **txBytes, txPackets:** número total de bytes transmitidos / paquetes para el flujo
- **rxBytes, rxPackets:** número total de bytes recibidos / paquetes para el flujo
- **lostPackets:** número total de paquetes que se supone que se pierde (no se informa más de 10 segundos)
- **timesForwarded:** el número de veces que un paquete ha sido según los informes remitidos
- **delayHistogram, jitterHistogram, packetSizeHistogram:** versiones histograma de la demora, jitter y tamaños de paquetes, respectivamente
- **packetsDropped, bytesDropped:** el número de paquetes y bytes perdidos, divididos según el código de razón de la pérdida (definido en la subclase).

3.10 Instalación y Configuración de NS3

NS-3 puede correr en diversas plataformas. Para la ejecución de las simulaciones realizadas en este estudio se ha escogido un Sistema Operativo basado en GNU/Linux, XUbuntu en su versión 00 para 64 bits. La instalación de NS-3 es relativamente sencilla para un usuario experimentado con sistemas GNU/Linux[19]. Las notas de instalación se pueden ver en su página oficial[25].

Existen dos maneras de instalar Ns3 son:

- ✓ **Bake:** Es una herramienta hecha en Python para construir software distribuido, desarrollado para el proyecto ns-3, pero en general podría ser utilizado para ayudar en el desarrollo de otros proyectos de código abierto que requieren la integración de múltiples proyectos.

- ✓ **Tarball:** Archivos distribuidos como archivo .tar, un formato de archivo que puede combinar varios archivos en un solo archivo llamado “tarball”, por lo general comprimidos.
- **Instalación de dependencias básicas y para el uso de Bake**

A continuación se explica la manera de instalar las dependencias necesarias para Ns3. La figura 33 explica gráficamente como instalar las dependencias gcc, g++ y python en sus versiones más recientes, mediante el comando **`$sudo apt-get install gcc g++ python`[26]**

- ✓ **Gcc.-** Conjunto de compiladores denominado así por sus siglas GNU Compiler Colección
- ✓ **G++.-** Conjunto de compiladores de C++, GNU C++
- ✓ **Python.-** Lenguaje de programación interpretado

```
tesis@tesis:~$ sudo apt-get install gcc g++ python
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
gcc ya está en su versión más reciente.
python ya está en su versión más reciente.
fijado python como instalado manualmente.
Paquetes sugeridos:
  g++-multilib g++-4.8-multilib gcc-4.8-doc libstdc++6-4.8-dbg
  libstdc++-4.8-doc
Se instalarán los siguientes paquetes NUEVOS:
  g++ g++-4.8 libstdc++-4.8-dev
0 actualizados, 3 se instalarán, 0 para eliminar y 3 no actualizados.
Necesito descargar 8.090 kB de archivos.
Se utilizarán 29,6 MB de espacio de disco adicional después de esta operación.
```

Figura 33: Instalación de gcc, g++ y python.

La figura 34 muestra la instalación de mercurial que es una dependencia más para la instalación, el cual es un sistema de control de versiones multiplataforma implementado principalmente con python, se lo instala mediante el código **`$sudo apt-get install mercurial`**

```
tesis@tesis:~$ sudo apt-get install mercurial
[sudo] password for tesis:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  mercurial-common
Paquetes sugeridos:
  qct vim emacs kdifft3 kdifft3-qt kompare meld tkcvs mgdiff python-mysqldb
  python-pygments
Se instalarán los siguientes paquetes NUEVOS:
  mercurial mercurial-common
0 actualizados, 2 se instalarán, 0 para eliminar y 3 no actualizados.
Necesito descargar 1.558 kB de archivos.
Se utilizarán 8.515 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] 
```

Figura 34: Instalación de mercurial.

En la figura 35 se puede apreciar el directorio en donde se instala Ns3 que es **/Home/tesis/bake**

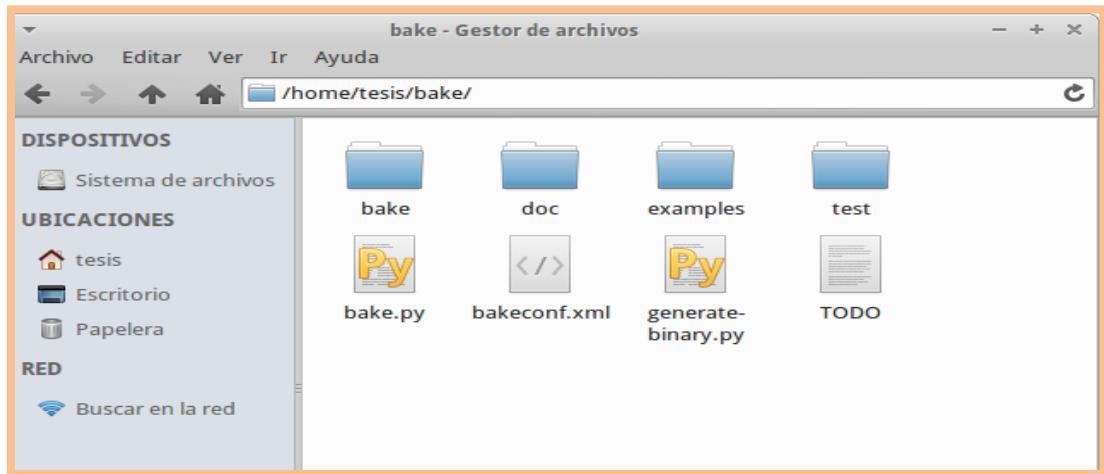


Figura 35: Directorio de instalación de Ns3.

En la figura 36 se aprecia el código para descargar **bake** usando mercurial, requisito que ya cumplimos anteriormente, con el siguiente comando **\$hg clone <http://code.nsnam.org/bake>**

```
terminal
$ hg clone http://code.nsnam.org/bake
destination directory: bake
requesting all changes
adding changesets
adding manifests
adding file changes
added 352 changesets with 811 changes to 63 files
updating to branch default
45 files updated, 0 files merged, 0 files removed, 0 files unresolved
```

Figura 36: Descarga de bake usando mercurial.

Luego de realizado este proceso se agrega el **bake** a la ruta, se lo realiza mediante los comandos que están en la figura 37.

```
terminal
$ export BAKE_HOME=`pwd`/bake
$ export PATH=$PATH:$BAKE_HOME
$ export PYTHONPATH=$PYTHONPATH:$BAKE_HOME
```

Figura 37: Añadir bake a la ruta.

Para saber que dependencia o requisito falta por instalar para el correcto funcionamiento de Ns3 se puede hacer uso del comando que se muestra en la figura 38 y se detalla a continuación ***bake.py check***, lo que muestra un listado de requisitos cubiertos ya que nos aparece la palabra **Ok** en aquellos que ya están, y la palabra **missing** en aquellas dependencias que aún faltan por instalar, en este caso las dependencias que faltan son **Cvs, Git, Bazaar, Unrar Tool, 7z ata compression, Utility, cmake y autoreconf tool.**

```
Terminal - tesis@tesis: ~
Archivo Editar Ver Terminal Pestañas Ayuda
tesis@tesis:~$ export BAKE_HOME=`pwd`/bake
tesis@tesis:~$ export PATH=$PATH:$BAKE_HOME
tesis@tesis:~$ export PYTHONPATH=$PYTHONPATH:$BAKE_HOME
tesis@tesis:~$ bake.py check
> Python - OK
> GNU C++ compiler - OK
> Mercurial - OK
> CVS - is missing
> GIT - is missing
> Bazaar - is missing
> Tar tool - OK
> Unzip tool - OK
> Unrar tool - is missing
> 7z data compression utility - is missing
> XZ data compression utility - OK
> Make - OK
> cMake - is missing
> patch tool - OK
> autoreconf tool - is missing

> Path searched for tools: /usr/local/sbin /usr/local/bin /usr/sbin /usr/bin /sbin /bin /usr/games /usr/local/games /home/tesis/bake /home/tesis/bake bin /home/tesis/bake /home/tesis/bake
tesis@tesis:~$
```

Figura 38: Comprobación de prerequisitos instalados.

Las dependencias incumplidas se instalan con el comando como lo muestra la figura 39.

```
Terminal - tesis@tesis:~  
Archivo Editar Ver Terminal Pestañas Ayuda  
tesis@tesis:~$ sudo apt-get install cvs git bzr unrar p7zip cmake dh-autoreconf  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes extras:  
autoconf automake autopoint autotools-dev build-essential cmake-data  
debhelper dh-apparmor dpkg-dev fakeroot git-man libalgorithm-diff-perl  
libalgorithm-diff-xs-perl libalgorithm-merge-perl libassuan0 liberror-perl  
libfakeroot libgpgmell libjs-jquery libltdl-dev libmail-sendmail-perl  
libsigsegv2 libsys-hostname-long-perl libtool m4 po-debconf python-bzrlib  
python-configobj python-gpgme python-keyring python-launchpadlib  
python-lazr.restfulclient python-lazr.uri python-oauth python-paramiko  
python-secretstorage python-simplejson python-wadllib  
Paquetes sugeridos:  
autoconf2.13 autoconf-archive gnu-standards autoconf-doc bzr-doc bzrtools  
python-bzrlib.tests codeblocks eclipse mksh rcs dh-make apparmor-easyprof  
debian-keyring git-daemon-run git-daemon-sysvinit git-doc git-el git-email  
git-gui gitk gitweb git-arch git-bzr git-cvs git-mediawiki git-svn gpgsm  
javascript-common libtool-doc automaken gfortran fortran95-compiler gcj-jdk  
p7zip-full libmail-box-perl python-bzrlib-dbg python-kerberos python-pycurl  
gir1.2-gnomekeyring-1.0 python-gdata python-keyczar python-kde4  
python-testresources python-secretstorage-doc  
Se instalarán los siguientes paquetes NUEVOS:  
autoconf automake autopoint autotools-dev build-essential bzr cmake  
cmake-data cvs debhelper dh-apparmor dh-autoreconf dpkg-dev fakeroot git  
git-man libalgorithm-diff-perl libalgorithm-diff-xs-perl  
libalgorithm-merge-perl libassuan0 liberror-perl libfakeroot libgpgmell  
libjs-jquery libltdl-dev libmail-sendmail-perl libsigsegv2  
libsys-hostname-long-perl libtool m4 p7zip po-debconf python-bzrlib  
python-configobj python-gpgme python-keyring python-launchpadlib  
python-lazr.restfulclient python-lazr.uri python-oauth python-paramiko  
python-secretstorage python-simplejson python-wadllib unrar  
0 actualizados, 45 se instalarán, 0 para eliminar y 3 no actualizados.  
Necesito descargar 15,2 MB de archivos.  
Se utilizarán 67,9 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n] 
```

Figura 39: Instalación de dependencias incumplidas.

Luego de la instalación de las dependencias incumplidas procedemos a comprobar nuevamente con el comando **bake.py check** con lo que se puede dar cuenta de que ya están cumplidas todas las dependencias necesarias con la palabra **Ok** como lo muestra la figura 40.

```
Terminal - tesis@tesis: ~
Archivo Editar Ver Terminal Pestañas Ayuda
tesis@tesis:~$ bake.py check
> Python - OK
> GNU C++ compiler - OK
> Mercurial - OK
> CVS - OK
> GIT - OK
> Bazaar - OK
> Tar tool - OK
> Unzip tool - OK
> Unrar tool - OK
> 7z data compression utility - OK
> XZ data compression utility - OK
> Make - OK
> cMake - OK
> patch tool - OK
> autoreconf tool - OK

> Path searched for tools: /usr/local/sbin /usr/local/bin /usr/sbin /usr/bin /sbin /bin /usr/games /usr/local/games /home/tesis/bake /home/tesis/bake bin /home/tesis/bake /home/tesis/bake
tesis@tesis:~$
```

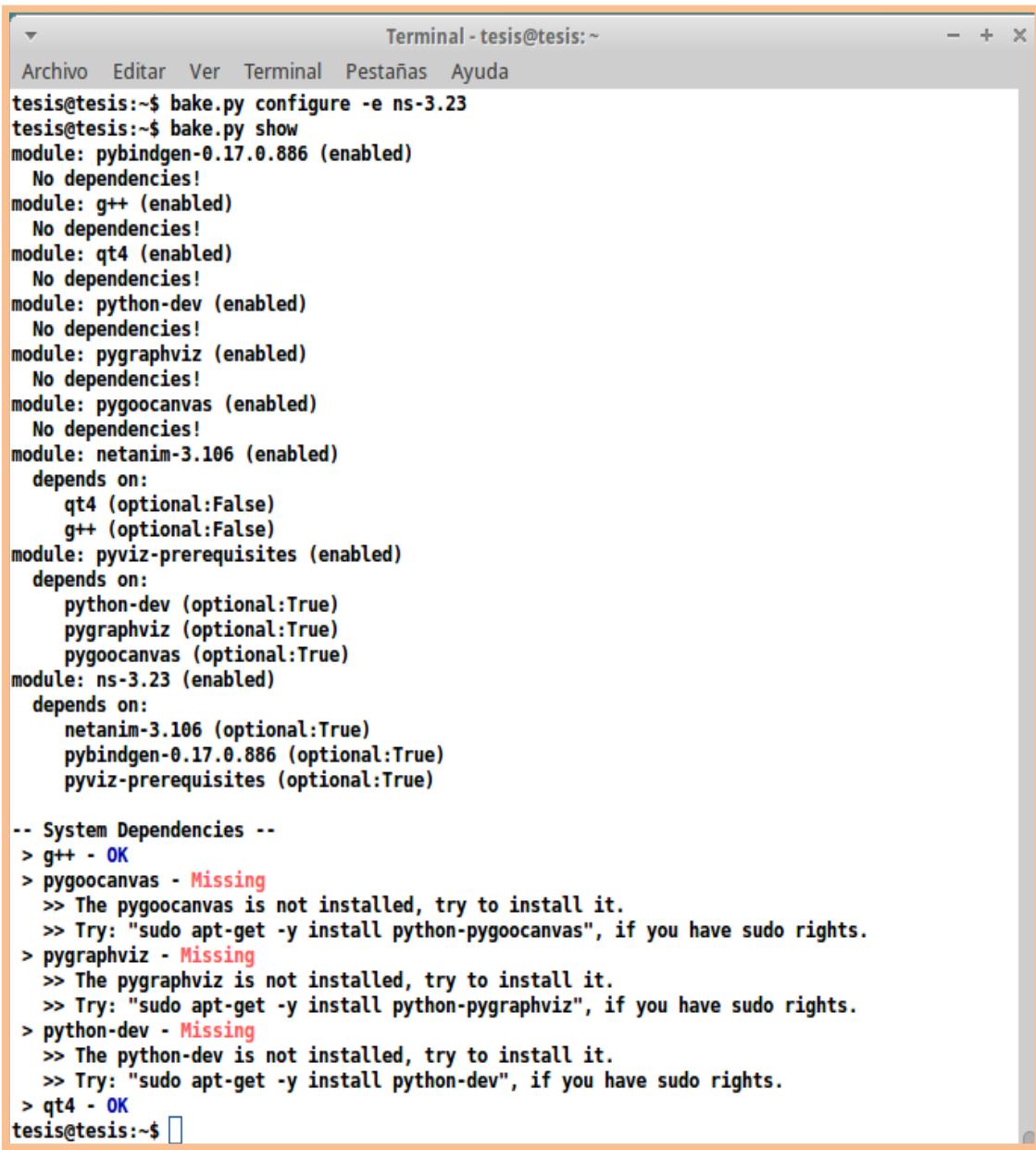
Figura 40: Comprobación de prerequisitos instalados por segunda ocasión.

Antes de la descarga y la construcción de ns-3 bake necesita para configurar que le informe cuáles son los módulos que desea agregar al ns-3, por ejemplo puede ser la distribución estándar. Para realizar esta acción tecleamos el siguiente comando **\$bake.py configure -e ns3.23** como lo muestra la figura 41.

```
Terminal - tesis@tesis: ~
Archivo Editar Ver Terminal Pestañas Ayuda
tesis@tesis:~$ bake.py configure -e ns-3.23
tesis@tesis:~$
```

Figura 41: Configuración de módulos para Ns3.

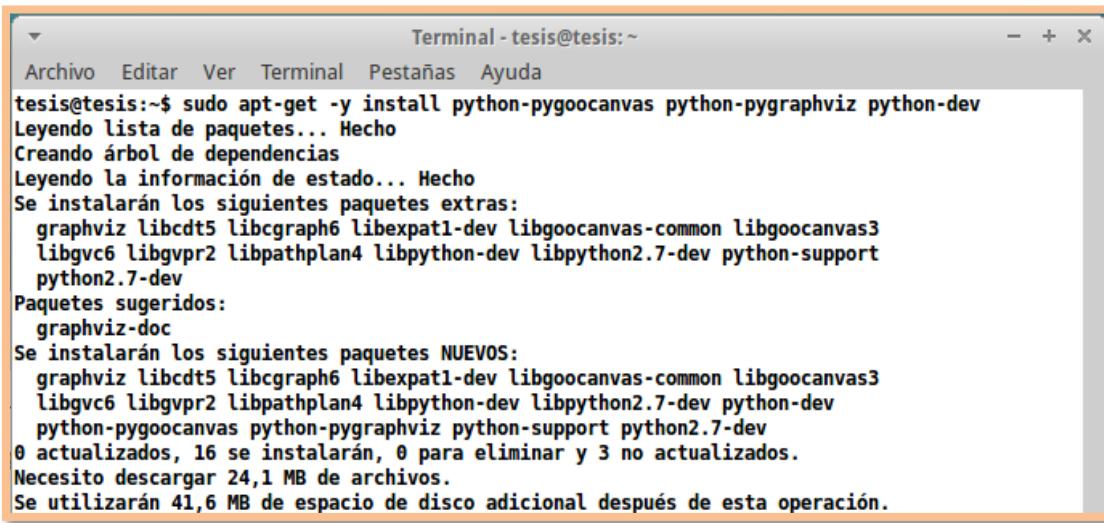
Luego de la configuración podemos ver los módulos específicos añadidos para la configuración y los que aún se encuentran sin instalar con el comando **bake.py show** en este caso muestra en la figura 42 los módulos específicos para configurar sin instalar **pygoocanvas**, **pygraphviz** y **python-dev**.



```
Terminal - tesis@tesis:~  
Archivo Editar Ver Terminal Pestañas Ayuda  
tesis@tesis:~$ bake.py configure -e ns-3.23  
tesis@tesis:~$ bake.py show  
module: pybindgen-0.17.0.886 (enabled)  
    No dependencies!  
module: g++ (enabled)  
    No dependencies!  
module: qt4 (enabled)  
    No dependencies!  
module: python-dev (enabled)  
    No dependencies!  
module: pygraphviz (enabled)  
    No dependencies!  
module: pygoocanvas (enabled)  
    No dependencies!  
module: netanim-3.106 (enabled)  
    depends on:  
        qt4 (optional:False)  
        g++ (optional:False)  
module: pyviz-prerequisites (enabled)  
    depends on:  
        python-dev (optional:True)  
        pygraphviz (optional:True)  
        pygoocanvas (optional:True)  
module: ns-3.23 (enabled)  
    depends on:  
        netanim-3.106 (optional:True)  
        pybindgen-0.17.0.886 (optional:True)  
        pyviz-prerequisites (optional:True)  
  
-- System Dependencies --  
> g++ - OK  
> pygoocanvas - Missing  
    >> The pygoocanvas is not installed, try to install it.  
    >> Try: "sudo apt-get -y install python-pygoocanvas", if you have sudo rights.  
> pygraphviz - Missing  
    >> The pygraphviz is not installed, try to install it.  
    >> Try: "sudo apt-get -y install python-pygraphviz", if you have sudo rights.  
> python-dev - Missing  
    >> The python-dev is not installed, try to install it.  
    >> Try: "sudo apt-get -y install python-dev", if you have sudo rights.  
> qt4 - OK  
tesis@tesis:~$
```

Figura 42: Módulos específicos para la configuración incumplidos.

En la figura 43 se muestra el comando necesario para instalar los módulos de configuración incumplidos, que es **\$sudo apt-get install python-pygoocanvas python-graphviz python-dev**.



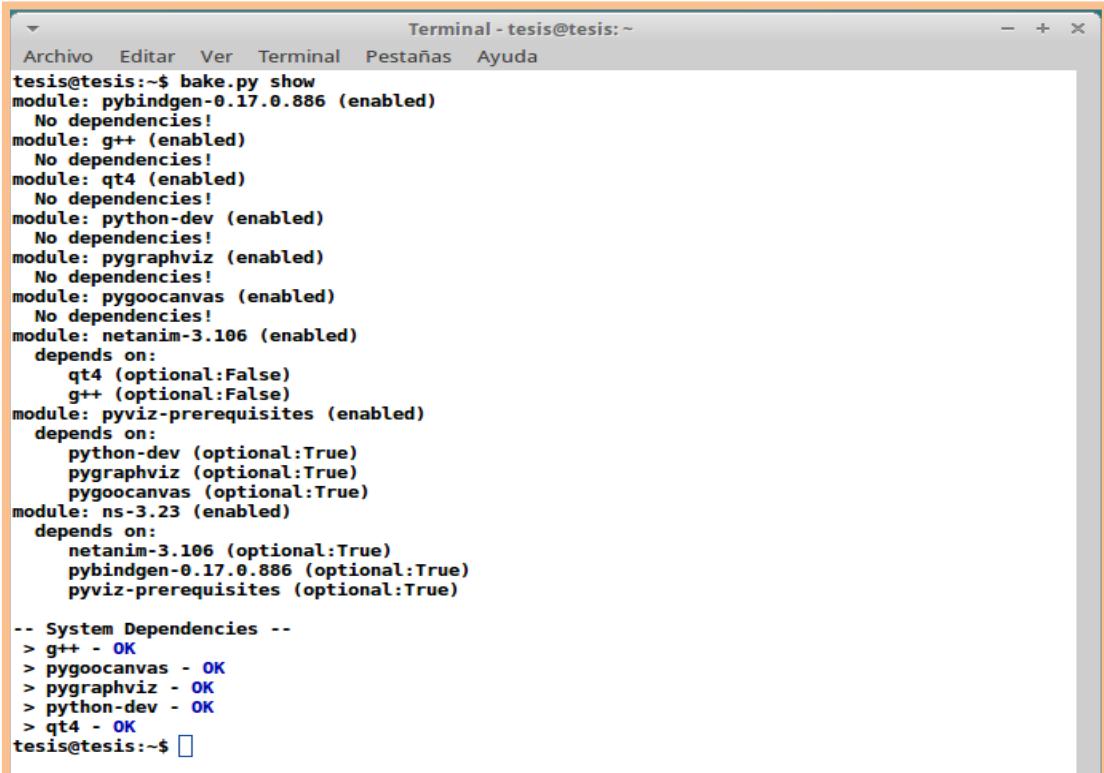
```

Terminal - tesis@tesis:~
Archivo Editar Ver Terminal Pestañas Ayuda
tesis@tesis:~$ sudo apt-get -y install python-pyoocanvas python-pygraphviz python-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  graphviz libcdt5 libcgraph6 libexpat1-dev libgoocanvas-common libgoocanvas3
  libgvc6 libgvpr2 libpathplan4 libpython-dev libpython2.7-dev python-support
  python2.7-dev
Paquetes sugeridos:
  graphviz-doc
Se instalarán los siguientes paquetes NUEVOS:
  graphviz libcdt5 libcgraph6 libexpat1-dev libgoocanvas-common libgoocanvas3
  libgvc6 libgvpr2 libpathplan4 libpython-dev libpython2.7-dev python-dev
  python-pyoocanvas python-pygraphviz python-support python2.7-dev
0 actualizados, 16 se instalarán, 0 para eliminar y 3 no actualizados.
Necesito descargar 24,1 MB de archivos.
Se utilizarán 41,6 MB de espacio de disco adicional después de esta operación.

```

Figura 43: Instalación de módulos específicos para la configuración.

De igual manera se puede volver a visualizar con el comando **bake.py show** y nos mostrara que todos los módulos específicos para la configuración se encuentran instalados como se observa en la figura 44.



```

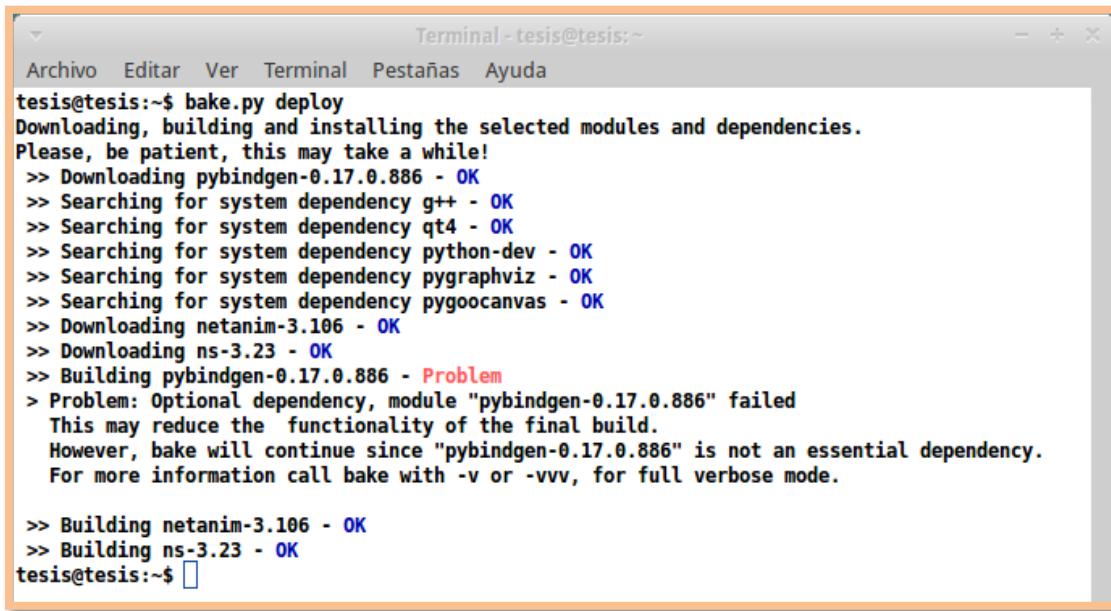
Terminal - tesis@tesis:~
Archivo Editar Ver Terminal Pestañas Ayuda
tesis@tesis:~$ bake.py show
module: pybindgen-0.17.0.886 (enabled)
  No dependencies!
module: g++ (enabled)
  No dependencies!
module: qt4 (enabled)
  No dependencies!
module: python-dev (enabled)
  No dependencies!
module: pygraphviz (enabled)
  No dependencies!
module: pyoocanvas (enabled)
  No dependencies!
module: netanim-3.106 (enabled)
  depends on:
    qt4 (optional:False)
    g++ (optional:False)
module: pyviz-prerequisites (enabled)
  depends on:
    python-dev (optional:True)
    pygraphviz (optional:True)
    pyoocanvas (optional:True)
module: ns-3.23 (enabled)
  depends on:
    netanim-3.106 (optional:True)
    pybindgen-0.17.0.886 (optional:True)
    pyviz-prerequisites (optional:True)

-- System Dependencies --
> g++ - OK
> pyoocanvas - OK
> pygraphviz - OK
> python-dev - OK
> qt4 - OK
tesis@tesis:~$ 

```

Figura 44: Visualización de módulos específicos instalados.

Ahora para descargar, construir e instalar los módulos para la instalación de Ns3 utilizamos el comando **\$bake.py deploy**, esto descarga los módulos seleccionados, todas sus dependencias, además de construir los módulos independientes de Ns3 como se puede ver en la figura 45.



```
Terminal - tesis@tesis:~
```

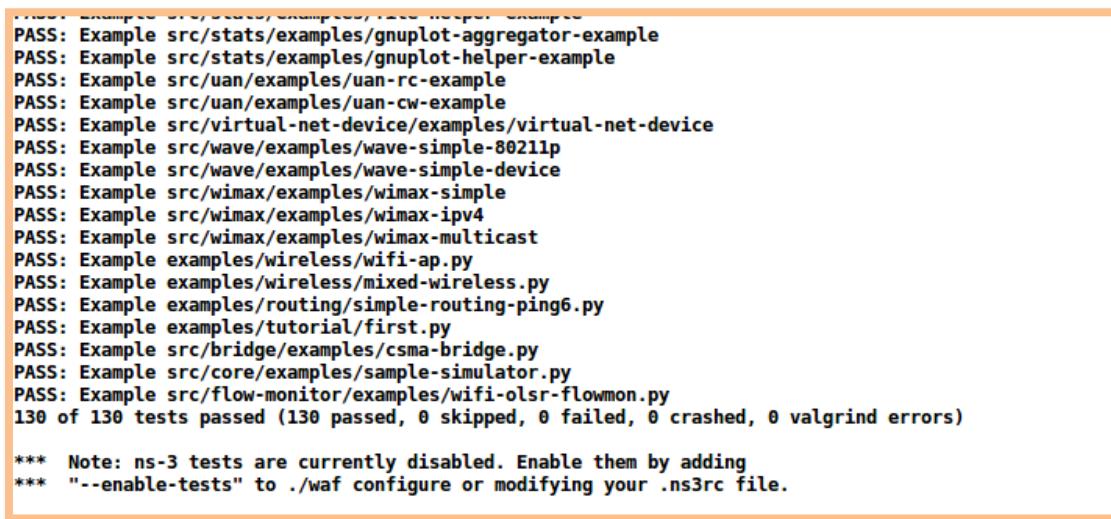
Archivo Editar Ver Terminal Pestañas Ayuda

```
tesis@tesis:~$ bake.py deploy
Downloading, building and installing the selected modules and dependencies.
Please, be patient, this may take a while!
>> Downloading pybindgen-0.17.0.886 - OK
>> Searching for system dependency g++ - OK
>> Searching for system dependency qt4 - OK
>> Searching for system dependency python-dev - OK
>> Searching for system dependency pygraphviz - OK
>> Searching for system dependency pygoocanvas - OK
>> Downloading netanim-3.106 - OK
>> Downloading ns-3.23 - OK
>> Building pybindgen-0.17.0.886 - Problem
> Problem: Optional dependency, module "pybindgen-0.17.0.886" failed
  This may reduce the functionality of the final build.
  However, bake will continue since "pybindgen-0.17.0.886" is not an essential dependency.
  For more information call bake with -v or -vvv, for full verbose mode.

>> Building netanim-3.106 - OK
>> Building ns-3.23 - OK
tesis@tesis:~$
```

Figura 45: Descarga y construcción de módulos.

Luego de este paso ya podemos comprobar realizando un test para verificar si pasa todas las pruebas de instalación deberán aparecernos múltiples mensajes con la palabra **PASS**. Lo que conseguimos con el comando **./test.py** como se observa en la figura 46.

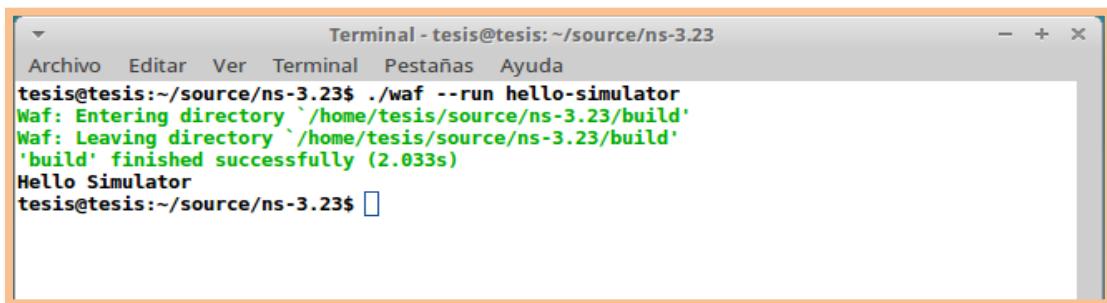


```
./test.py
PASS: Example src/stats/examples/gnuplot-aggregator-example
PASS: Example src/stats/examples/gnuplot-helper-example
PASS: Example src/uan/examples/uau-rc-example
PASS: Example src/uau/examples/uau-cw-example
PASS: Example src/virtual-net-device/examples/virtual-net-device
PASS: Example src/wave/examples/wave-simple-80211p
PASS: Example src/wave/examples/wave-simple-device
PASS: Example src/wimax/examples/wimax-simple
PASS: Example src/wimax/examples/wimax-ipv4
PASS: Example src/wimax/examples/wimax-multicast
PASS: Example examples/wireless/wifi-ap.py
PASS: Example examples/wireless/mixed-wireless.py
PASS: Example examples/routing/simple-routing-ping6.py
PASS: Example examples/tutorial/first.py
PASS: Example src/bridge/examples/csma-bridge.py
PASS: Example src/core/examples/sample-simulator.py
PASS: Example src/flow-monitor/examples/wifi-olsr-flowmon.py
130 of 130 tests passed (130 passed, 0 skipped, 0 failed, 0 crashed, 0 valgrind errors)

*** Note: ns-3 tests are currently disabled. Enable them by adding
*** "--enable-tests" to ./waf configure or modifying your .ns3rc file.
```

Figura 46: Prueba para verificar la construcción de Ns3.

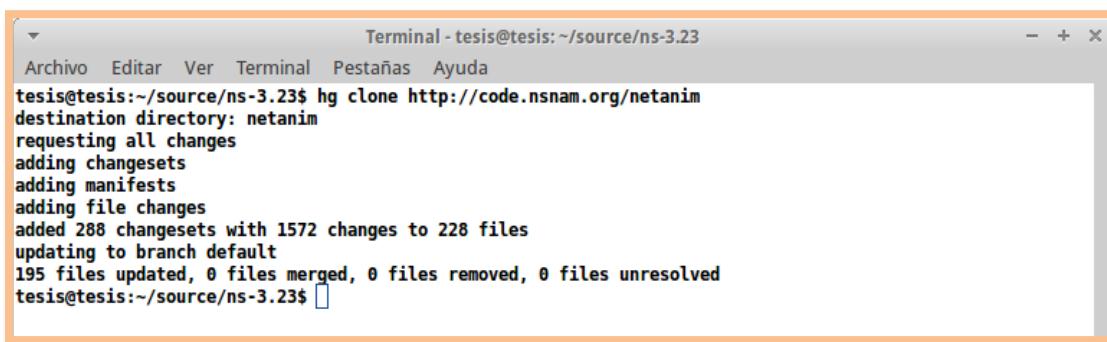
A continuación procedemos a verificar la instalación de Ns3 con el comando que permite correr los ejemplos en Ns3, desde el directorio de Ns-3.23 con ***./waf –run hello-simulator*** el mensaje para mostrar que el simulador se instaló correctamente es **Hello Simulator** como se puede ver en la figura 47.



```
Terminal - tesis@tesis: ~/source/ns-3.23
Archivo Editar Ver Terminal Pestañas Ayuda
tesis@tesis:~/source/ns-3.23$ ./waf --run hello-simulator
Waf: Entering directory `/home/tesis/source/ns-3.23/build'
Waf: Leaving directory `/home/tesis/source/ns-3.23/build'
'build' finished successfully (2.033s)
Hello Simulator
thesis@tesis:~/source/ns-3.23$
```

Figura 47: Prueba para verificar la instalación de Ns3.

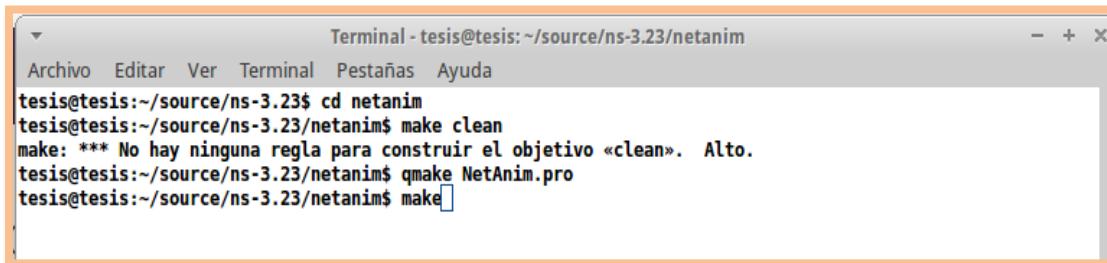
Para poder visualizar los nodos en la red lo que vamos hacer es utilizar NetAnim instalándolo de la siguiente manera primero con el comando ***hg clone http://code.nsnam.org/netanim*** como lo indica la figura 48.



```
Terminal - tesis@tesis: ~/source/ns-3.23
Archivo Editar Ver Terminal Pestañas Ayuda
tesis@tesis:~/source/ns-3.23$ hg clone http://code.nsnam.org/netanim
destination directory: netanim
requesting all changes
adding changesets
adding manifests
adding file changes
added 288 changesets with 1572 changes to 228 files
updating to branch default
195 files updated, 0 files merged, 0 files removed, 0 files unresolved
thesis@tesis:~/source/ns-3.23$
```

Figura 48: Descarga de NetAnim.

Y luego pasamos a la construcción de NetAnim como se muestra en la figura 49.



```
Terminal - tesis@tesis: ~/source/ns-3.23/netanim
Archivo Editar Ver Terminal Pestañas Ayuda
tesis@tesis:~/source/ns-3.23$ cd netanim
tesis@tesis:~/source/ns-3.23/netanim$ make clean
make: *** No hay ninguna regla para construir el objetivo «clean». Alto.
tesis@tesis:~/source/ns-3.23/netanim$ qmake NetAnim.pro
tesis@tesis:~/source/ns-3.23/netanim$ make
```

Figura 49: Construcción de NetAnim.

Para utilizar NetAnim se realiza desde el directorio de NetAnim con el siguiente comando **./NetAnim** lo que nos abrirá una interfaz como la que se muestra en la figura 50.

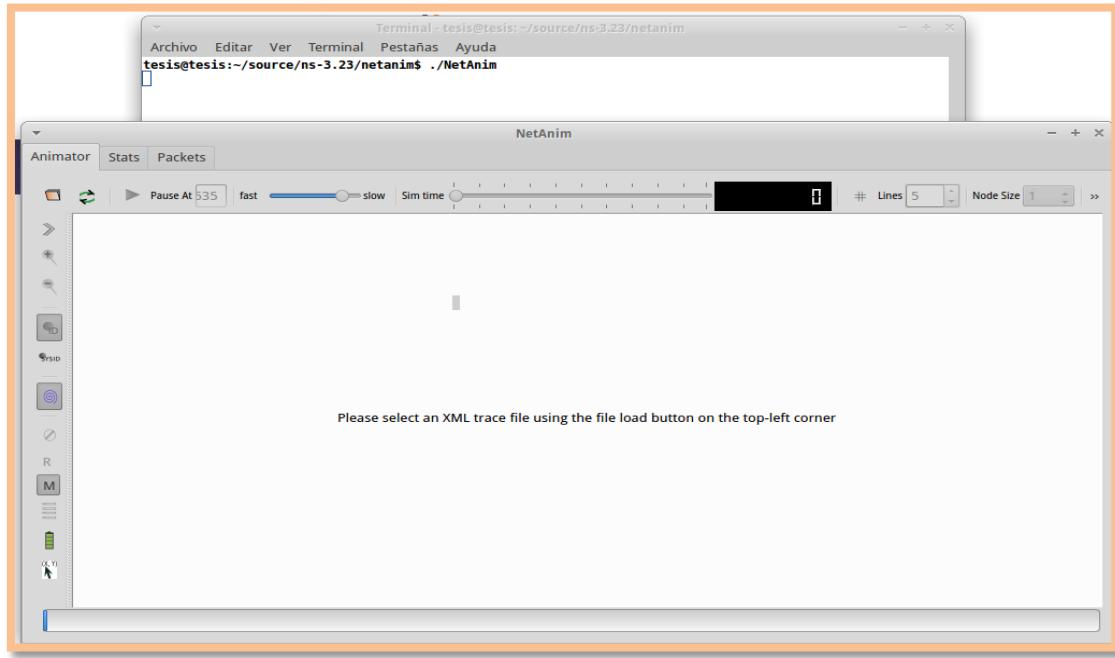


Figura 50: Interfaz de NetAnim.

Ahora procedemos a abrir el documento en xml que resulta luego de realizar estos pasos anteriormente mencionados. Y nos muestra los nodos que conforman la red tal como en la figura 51.

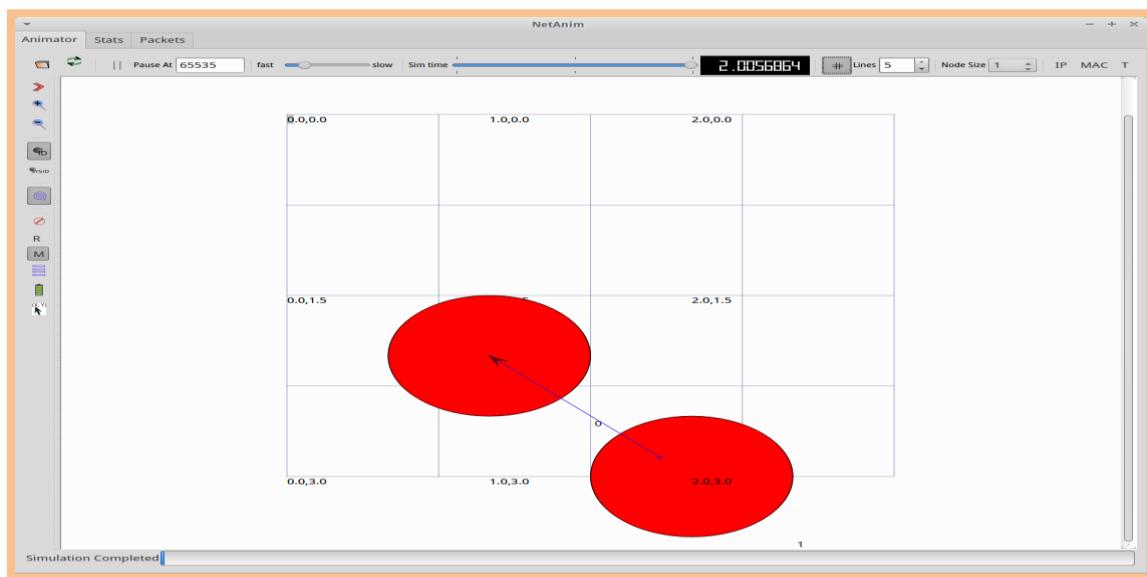


Figura 51: Interfaz de nodos en NetAnim.

3.11 Codificación de la Topología

Dentro de la codificación se realizan ajustes mediante código a los elementos de la topología, a sus características y escenario en donde se desenvuelven, como lo haríamos en una configuración sobre estructura en un laboratorio o en un entorno real, en las redes Ad-Hoc se va estructurando capa por capa del modelo OSI[27].

- **MÓDULOS DE INCLUSIÓN**

Como primera parte del código se estructura dentro del mismo los módulos incluidos dentro de la simulación, en la figura 52 se detallan los módulos utilizados.

```
#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/olsr-module.h"
#include "ns3/dsdv-module.h"
#include "ns3/dsr-module.h"
#include "ns3/applications-module.h"
#include "ns3/netanim-module.h"
#include "ns3/flow-monitor-module.h"
```

Figura 52: Módulos incluidos en la simulación.

- **DEFINICION DE NOMBRES**

La figura 53 muestra los nombres utilizados dentro del código, para facilitar el manejo de variables e integrar funciones.

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

Figura 53: Definición de nombres Ns3 y DSR.

- **REGISTRO DE NOTAS**

En la simulación se utilizaron las notas que se observan en la figura 54, para manejo de registros en la ejecución.

```
NS_LOG_COMPONENT_DEFINE ("manet-uno");

NS_LOG_COMPONENT_DEFINE ("manet-dos");

NS_LOG_UNCOND (PrintReceivedPacket (socket, packet));

NS_FATAL_ERROR ("No such protocol:" << m_protocol);

NS_LOG_INFO ("assigning ip address");

NS_LOG_INFO ("Run Simulation.");
```

Figura 54: Registro de notas utilizados en la simulación.

- **FUNCION PRINCIPAL**

La función principal del programa realiza la simulación y guarda los valores establecidos en un Csv, para su posterior análisis como lo muestra la figura 55.

```
int
main (int argc, char *argv[])
{
    RoutingExperiment experiment;
    std::string CSVfileName = experiment.CommandSetup
(argc,argv);

    //el archivo en blanco empieza a escribir los encabezados
    std::ofstream out (CSVfileName.c_str ());
    // Se almacena los resultados en un csv
    out << "SimulationSecond," <<
    "ReceiveRate," <<
    "PacketsReceived," <<
    "NumberOfSinks," <<
    "RoutingProtocol," <<
    "TransmissionPower" <<
    std::endl;
    out.close ();

    int nSinks = 10;
    double txp = 7.5;

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

Figura 55: Función principal de la Simulación.

- **CLASES PARA EL DISEÑO DE LA TOPOLOGIA**

Las clases que se utilizaron para el diseño de la topología, como para declaración de variables, la comunicación que se establece, etc. Y son las que se detallan en la figuras siguientes figuras.

La figura 56 realiza la creación de nodos en forma Ad-Hoc, y los agrega a un contenedor.

```
NodeContainer adhocNodes;  
adhocNodes.Create (nWifis);
```

Figura 56: Creación de Nodos (capa física).

La figura 57 muestra el tipo de comunicación que van a utilizar los protocolos, en este caso es utilizando el estándar 802.11b WIFI.

```
// Establecer canal wifi utilizando helpers  
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 ());
```

Figura 57: Fijación de canal Wifi (Capa de Enlace).

Con esta clase se instala la tecnología a utilizar en cada nodo, en este caso es la forma Ad-Hoc, como lo muestra la figura 58.

```
NetDeviceContainer adhocDevices = wifi.Install  
(wifiPhy, wifiMac, adhocNodes);
```

Figura 58: Instalación de modo Ad-Hoc.

En la figura 59 se detalla en código la forma en que se utilizó la clase **InternetStackHelper** para otorgarle las reglas de cada protocolo, en su forma de enrutamiento.

```

InternetStackHelper internet;

// Asignar Protocolo a ejecución
switch (m_protocol)
{
    case 1:
        list.Add (olsr, 100);
        m_protocolName = "OLSR";
        break;
    case 2:
        list.Add (AODV, 100);
        m_protocolName = "AODV";
        break;
    case 3:
        list.Add (dsdv, 100);
        m_protocolName = "DSDV";
        break;
    case 4:
        m_protocolName = "DSR";
        break;
    default:
        NS_FATAL_ERROR ("No such protocol:" << m_protocol);
}

if (m_protocol < 4)
{
    internet.SetRoutingHelper (list);
    internet.Install (adhocNodes);
}
else if (m_protocol == 4)
{
    internet.Install (adhocNodes);
    dsrMain.Install (dsr, adhocNodes);
}

```

Figura 59: Segmento de comparación de protocolo.

En la figura 60 se define las direcciones Ip en versión cuatro, las que se detalla en código y se utilizaran en todos los nodos.

```

Ipv4AddressHelper addressAdhoc;
addressAdhoc.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer adhocInterfaces;
adhocInterfaces = addressAdhoc.Assign (adhocDevices);

```

Figura 60: Asignación de direcciones Ip (Capa de Red).

- **APLICACIONES**

De igual manera se utilizan ciertas clases para definir las aplicaciones que soporta la red en Ns3 se definen con la clase **OnOffHelper** que se utiliza para asignación de sockets y direcciones ip a la red, como lo muestra la figura 61.

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

Figura 61: Uso de clase de Aplicaciones (Capa de Aplicación).

- **SIMULADOR**

Y finalmente el inicio y final de la simulación con las siguientes líneas de código como muestra la figura 62.

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

Figura 62: Inicio y fin de la simulación.

3.12 Obtención de métricas y animación de la Red

Además de esto dentro del código se utilizaron dos módulos adicionales detallados en las figuras a continuación.

- **ANIMACION**

Para lo que es la observación de la animación en Ns3 se utilizó el módulo **NetAnim**, el que se codifica de la siguiente manera y se observa en la figura 63.

```
AnimationInterface anim ("animacionDos.xml");  
anim.SetMobilityPollInterval(Seconds(1));  
anim.EnablePacketMetadata(true);
```

Figura 63: Líneas de código para uso de NetAnim.

El que nos muestra la animación del primer escenario que consta de 25 nodos, con características definidas anteriormente utilizando el protocolo **OLSR** observado en la figura 64.

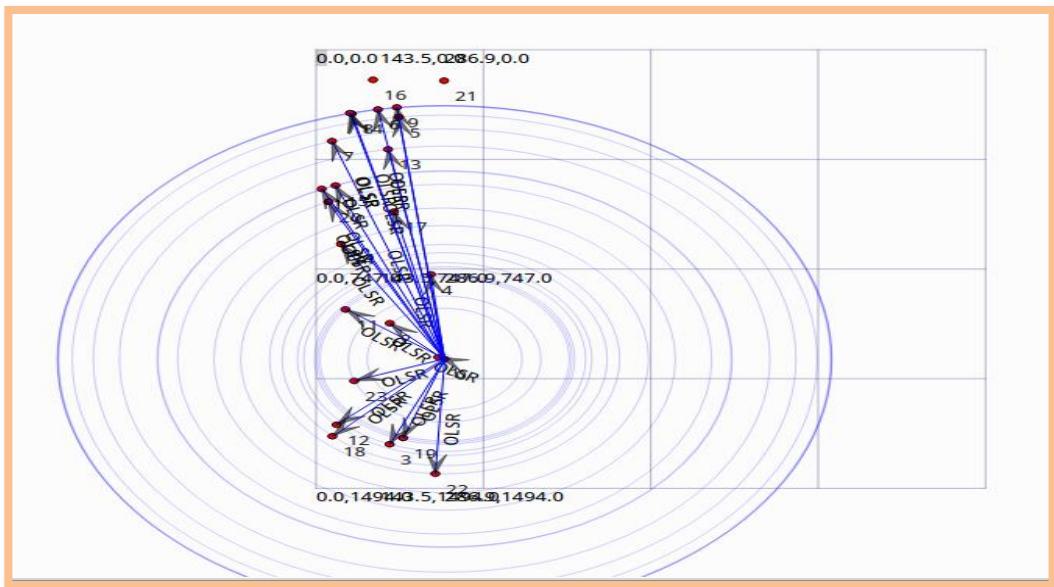


Figura 64: Envío de broadcast con OLSR en escenario uno.

Así mismo para el escenario de prueba dos que consta de 50 nodos, con características definidas de igual manera anteriormente, utilizando de la misma manera el protocolo **OLSR**, y se observa en la figura 65.

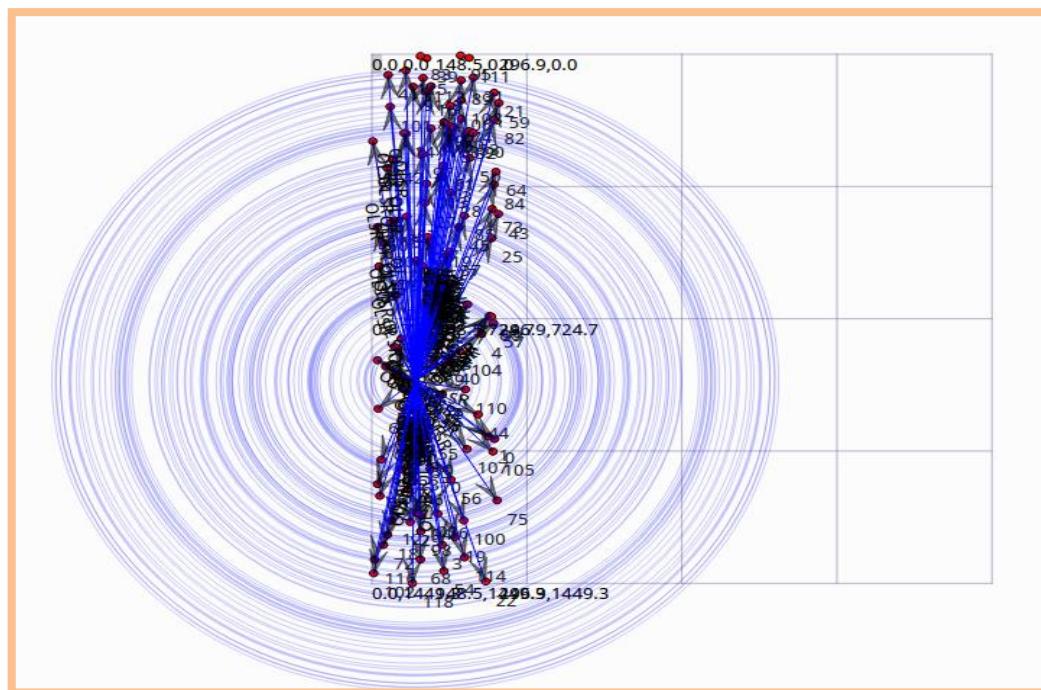


Figura 65: Envío de broadcast con OLSR en escenario dos.

A continuación tenemos el resultado grafico en los dos escenarios con el protocolo **OADV**. En la figura 66 se puede apreciar que al igual que en OLSR, AODV realiza inundaciones en toda la red, tratando así de establecer la mejor ruta.

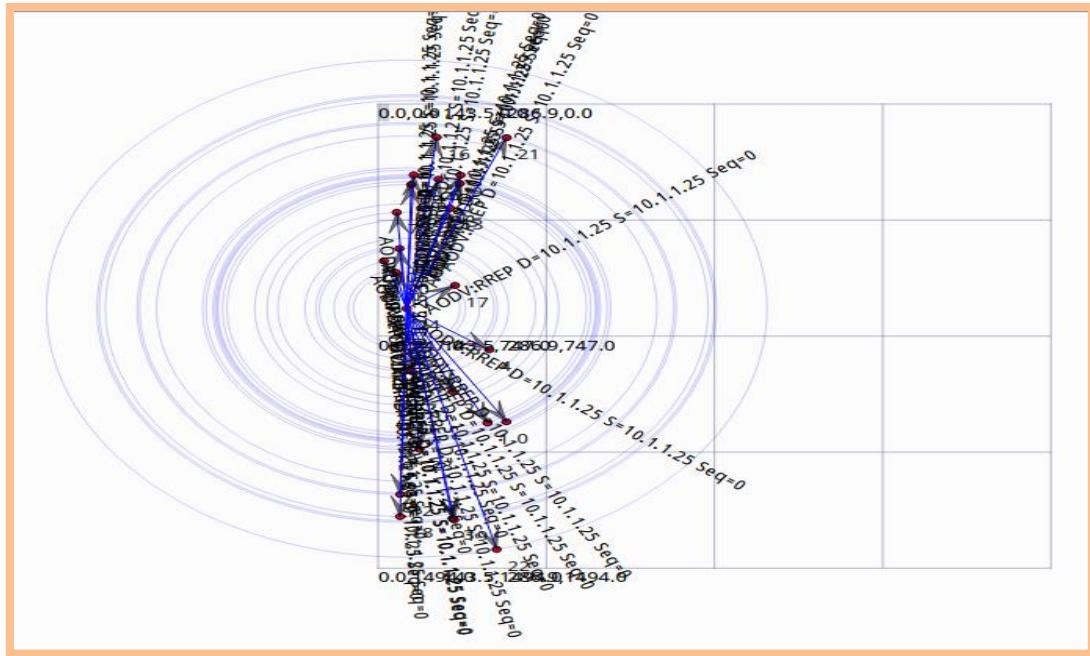


Figura 66: Envío de broadcast con AODV en escenario uno.

Pero además de esto sectoriza las inundaciones para tal como lo muestra la figura 67, esto con el fin de reducir la latencia por la demora en el descubrimiento de rutas.

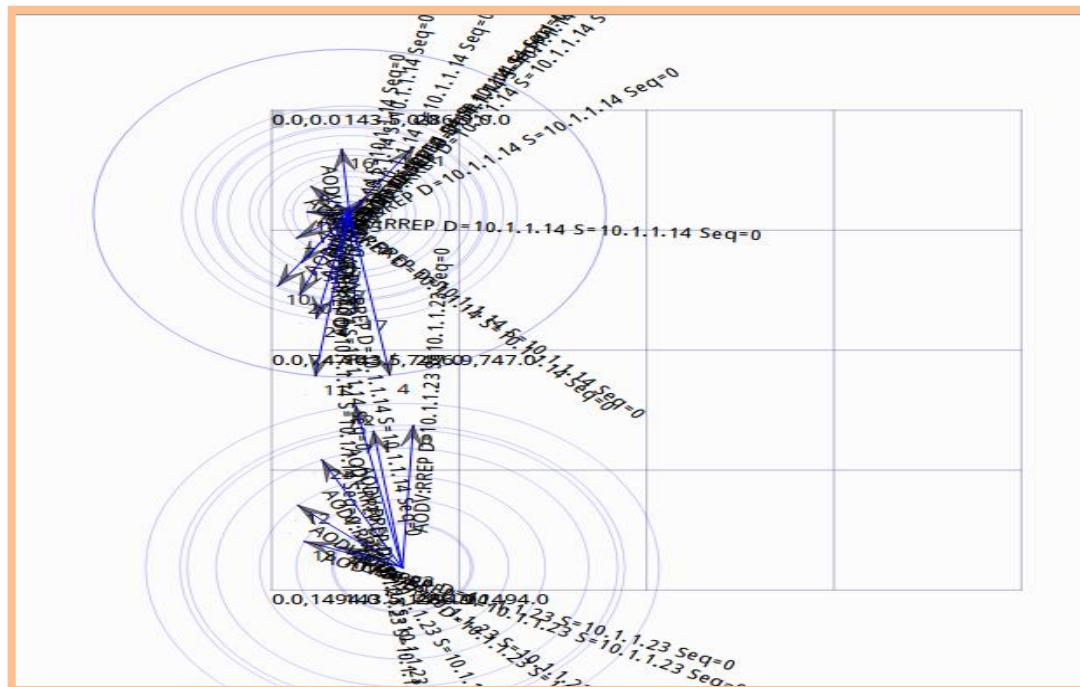


Figura 67: Sectorización de envío de broadcast con AODV en escenario uno.

De igual manera en el escenario de prueba dos se pueden visualizar la sectorización de las inundaciones para mejora en el tiempo de latencia, apreciado en las figuras 68 y 69.

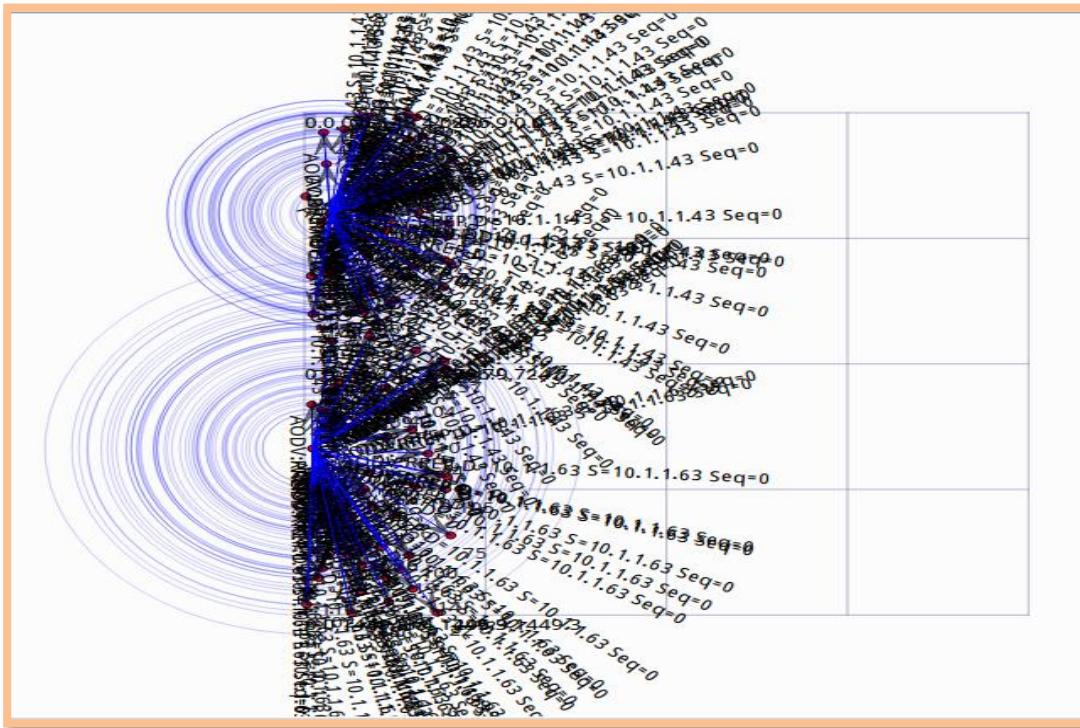


Figura 68: Sectorización de envío de broadcast con AODV en escenario dos.

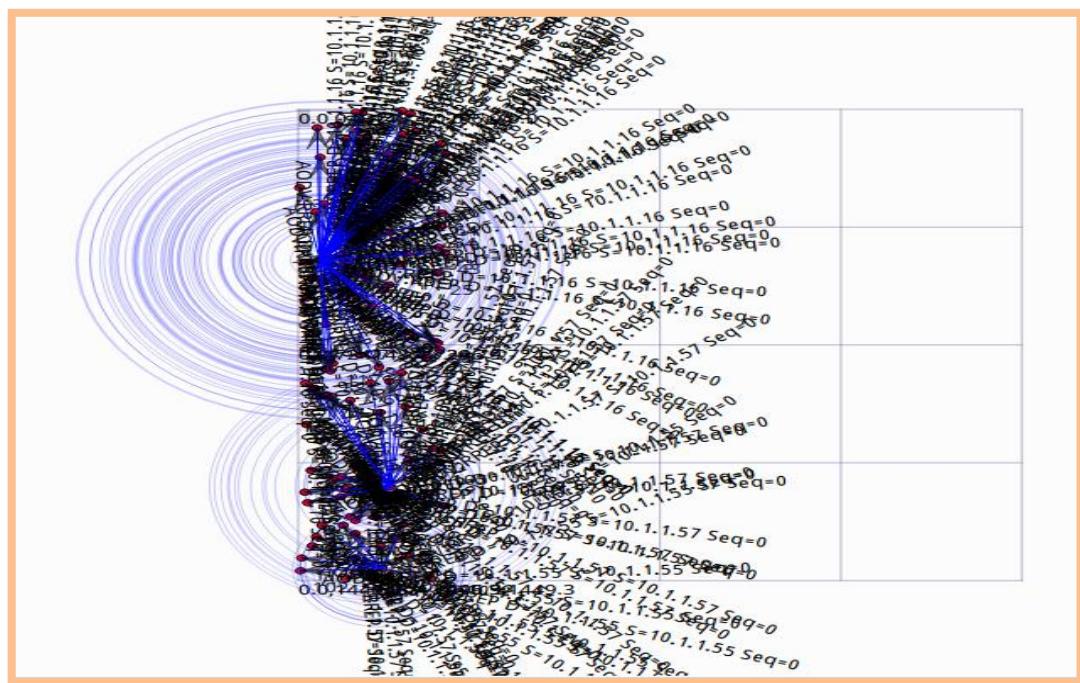


Figura 69: Mayor sectorización de envío de broadcast con AODV en escenario dos.

La forma gráfica del tercer protocolo elegido **DSDV**, se aprecia en las gráficas siguientes.

En este caso utilizando el escenario uno, en cambio se aprecia tiempos en los que hay movimiento de nodos, pero sin descubrimiento de rutas, la figura 70 muestra el tiempo en que no se aprecia esta etapa.

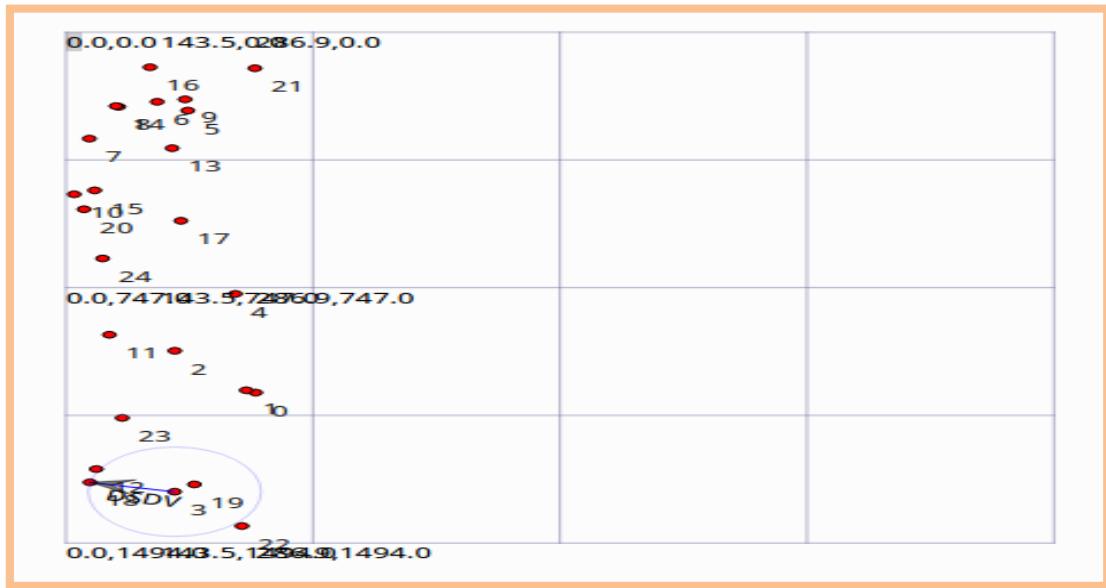


Figura 70: Bajo envío de broadcast con DSDV en escenario uno.

Tomando ciertos momentos para realizar el descubrimiento de las rutas como se observa en la figura 71.

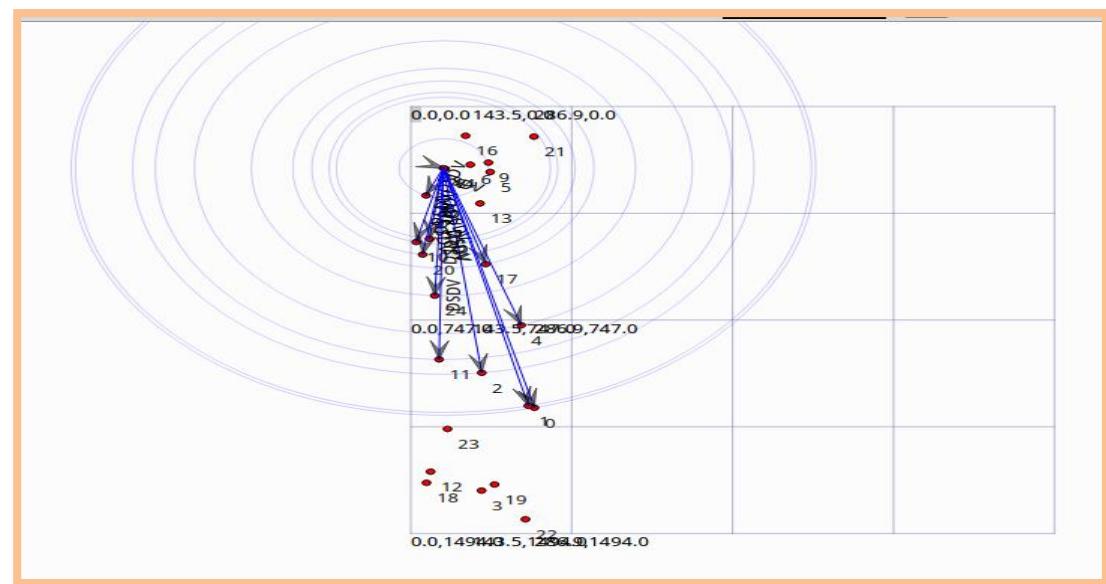


Figura 71: Envío de broadcast con DSDV en escenario uno.

Al simular el escenario dos resulta de la misma manera, existen lapsos de tiempo en los que no existe descubrimiento de rutas, esto debido a que cada nodo tiene conocimiento sectorizado local. En la figura 72 muestra el escenario de prueba dos utilizando el protocolo DSDV, en un lapso de tiempo sin descubrimiento de rutas.

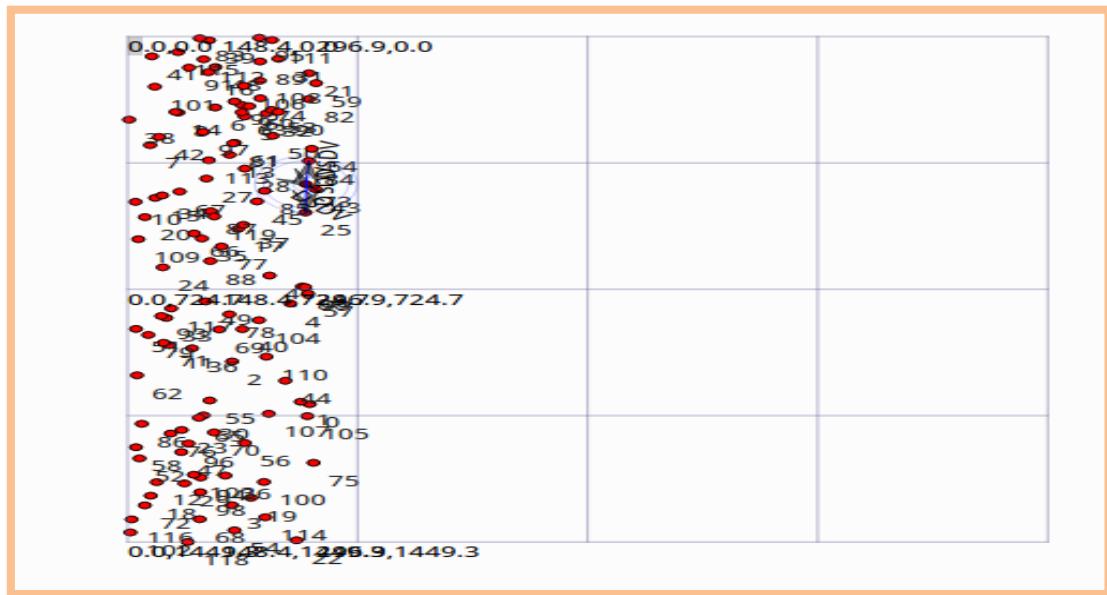


Figura 72: Escenario dos sin envío de broadcast usando DSDV.

Así mismo los momentos en que si existe descubrimiento de rutas apreciado en la figura 73.

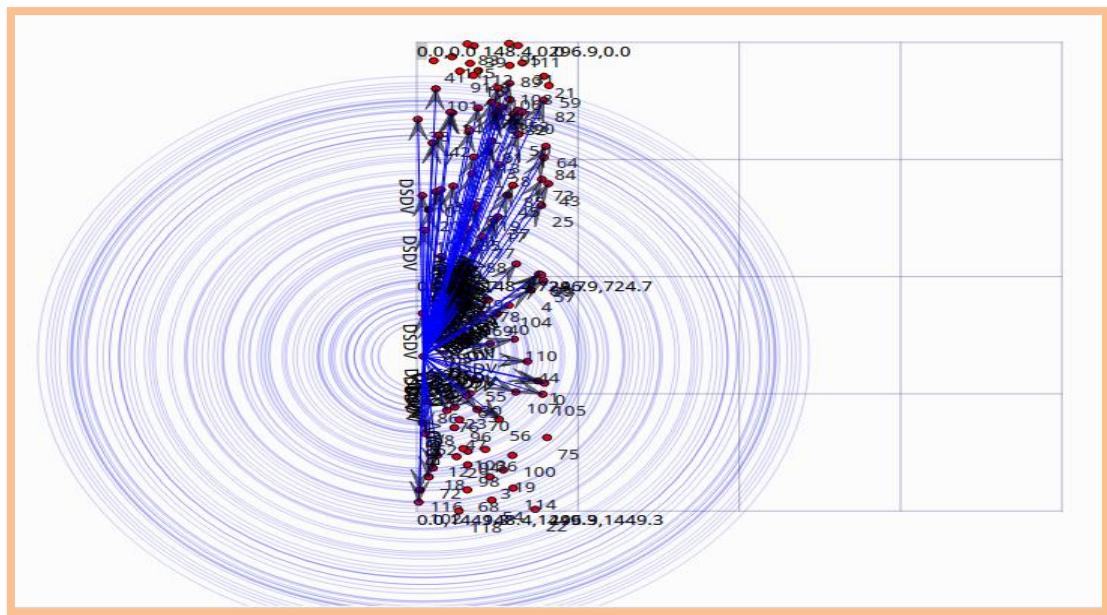


Figura 73: Escenario dos con envío de broadcast usando DSDV.

Y finalmente el protocolo **DSR** simulado en Netanim de Ns3, en el que se puede apreciar más movilidad en los nodos y en muy poca medida el tema de descubrimiento de rutas, estableciendo una secuencia para poder apreciar este tránsito en las gráficas 74,75 y 76.

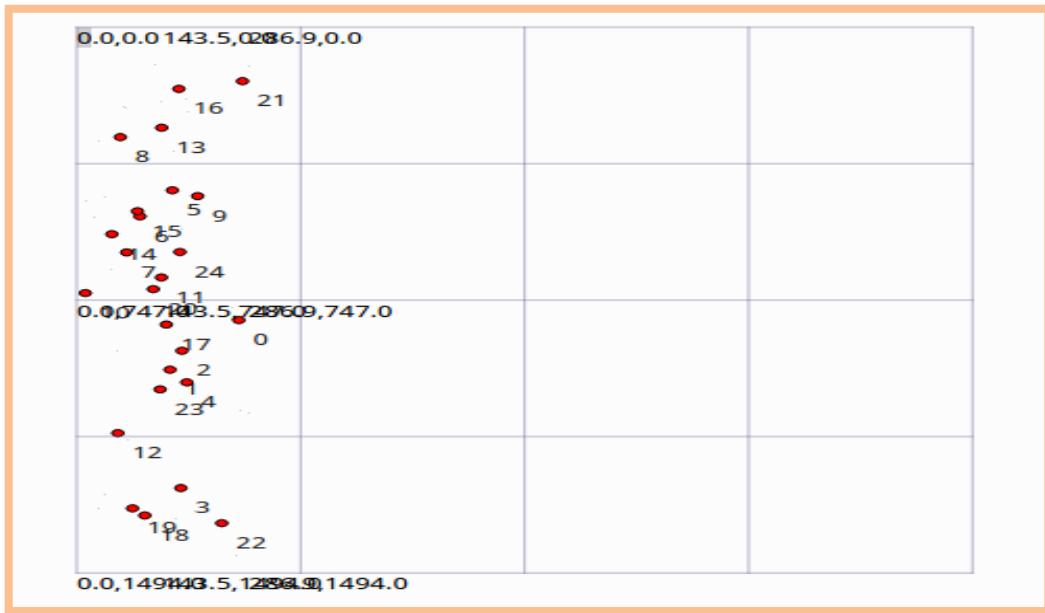


Figura 74: Movilidad inicial escenario uno y sin broadcast usando DSR.

Se puede apreciar mayor movilidad en esta figura si tomamos en cuenta la ubicación de nodos 4, 5, 8, 9, 12, 17, 18, 19 que son los que mayormente se observa variación de su ubicación.

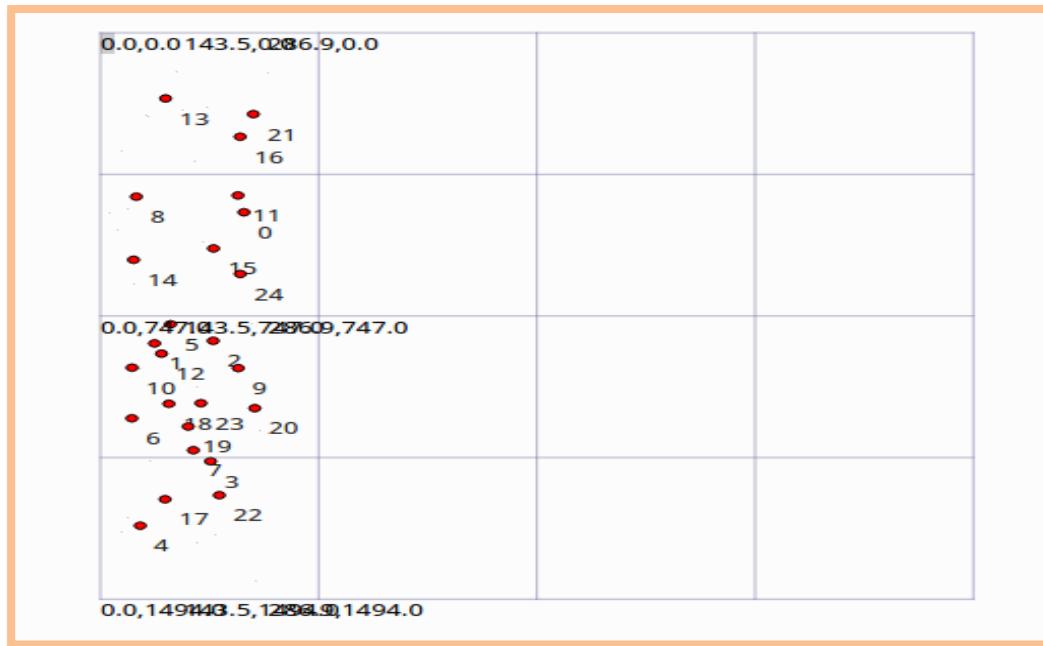


Figura 75: Movilidad de los nodos escenario uno y sin broadcast usando DSR.

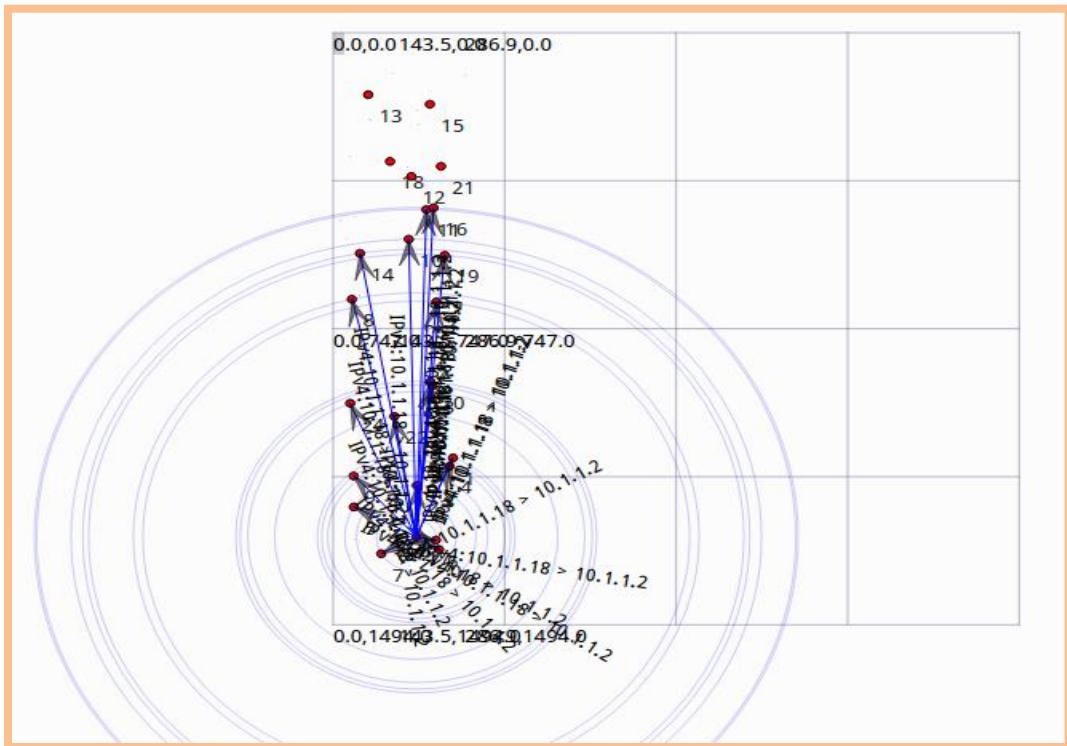


Figura 76: Envío de broadcast en escenario uno usando DSR.

De igual manera sucede en el escenario dos, esto se aprecia en las gráficas 77, 78 y 79. Lo que varía en tiempo de ejecución sobretodo en este escenario, sea cualquiera el protocolo que se utilice.

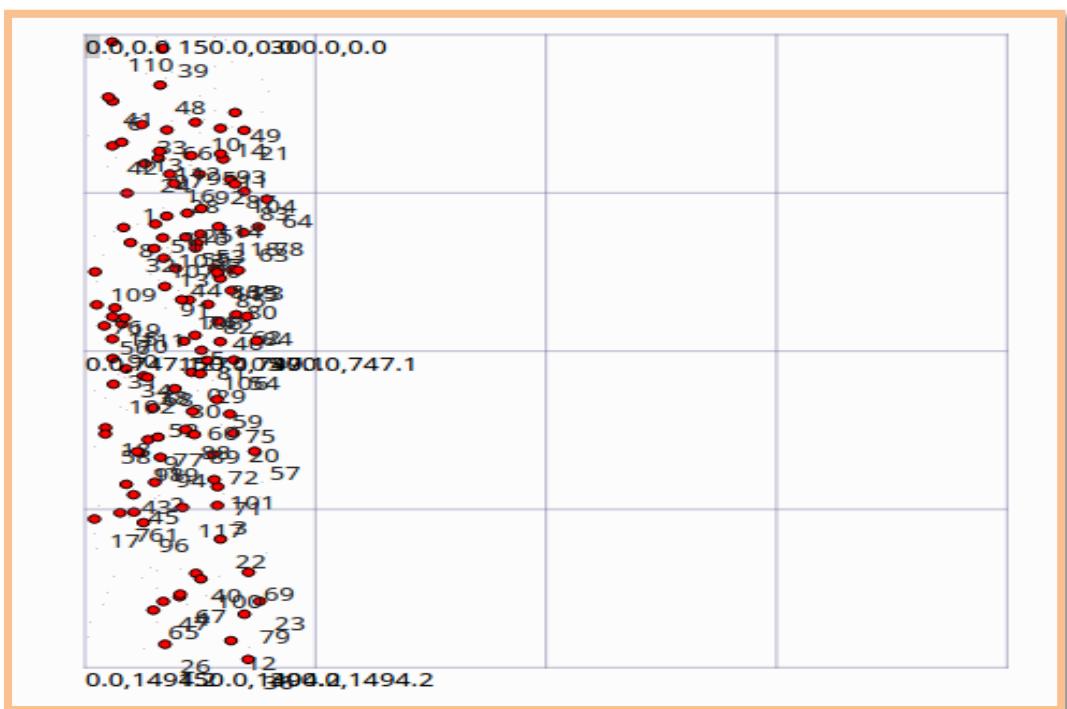


Figura 77: Movilidad inicial escenario dos y sin broadcast usando DSR.

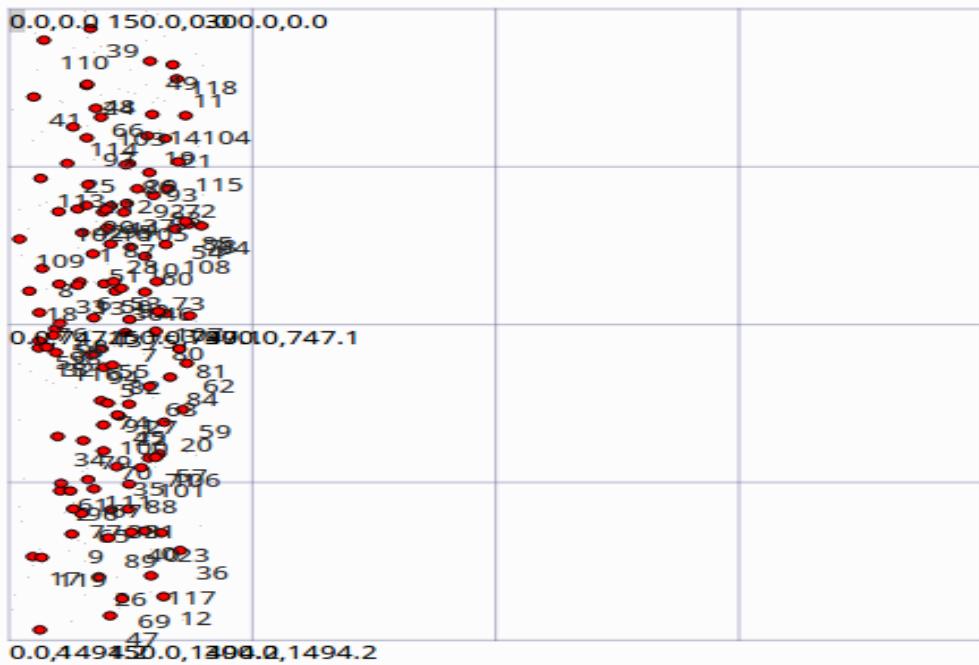


Figura 78: Movilidad de los nodos escenario dos y sin broadcast usando DSR.

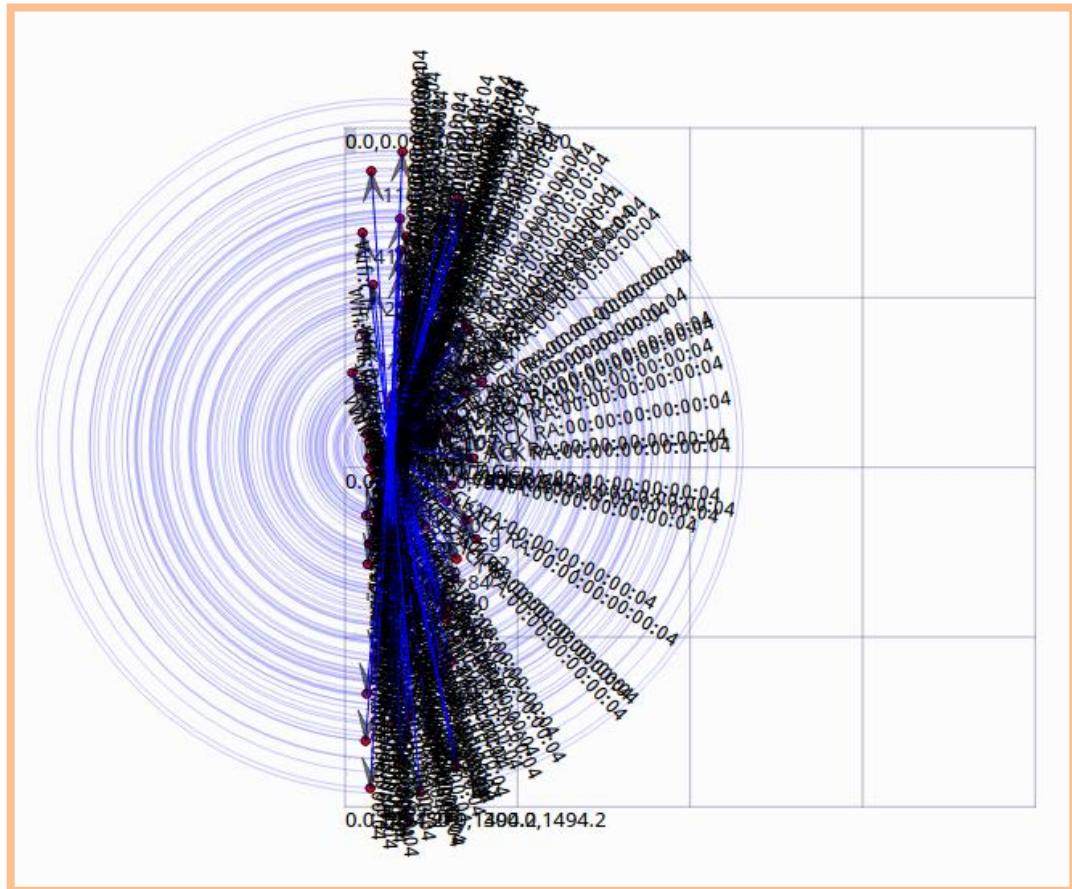


Figura 79: Envío de broadcast en escenario dos usando DSR.

- **ANÁLISIS**

Para el tema de análisis, métricas y estadísticas se utilizó el módulo **FlowMonitor** el que se puede observar las líneas para su codificación en la figura 80[28].

```
Ptr<FlowMonitor> flowmon;
FlowMonitorHelper flowmonHelper;
flowmon = flowmonHelper.InstallAll();
```

Figura 80: Líneas de código para uso de FlowMonitor.

Esta forma de captura de métricas obtiene archivos **.pcap** los que pueden ser analizados en cualquier capturador de paquetes, en este caso se ha elegido **wireshark**.

En la figura 81 se muestra los paquetes **.pcap** que se generan al correr la simulación, dependiendo del número de nodos serán también el número de archivos pcap[29].

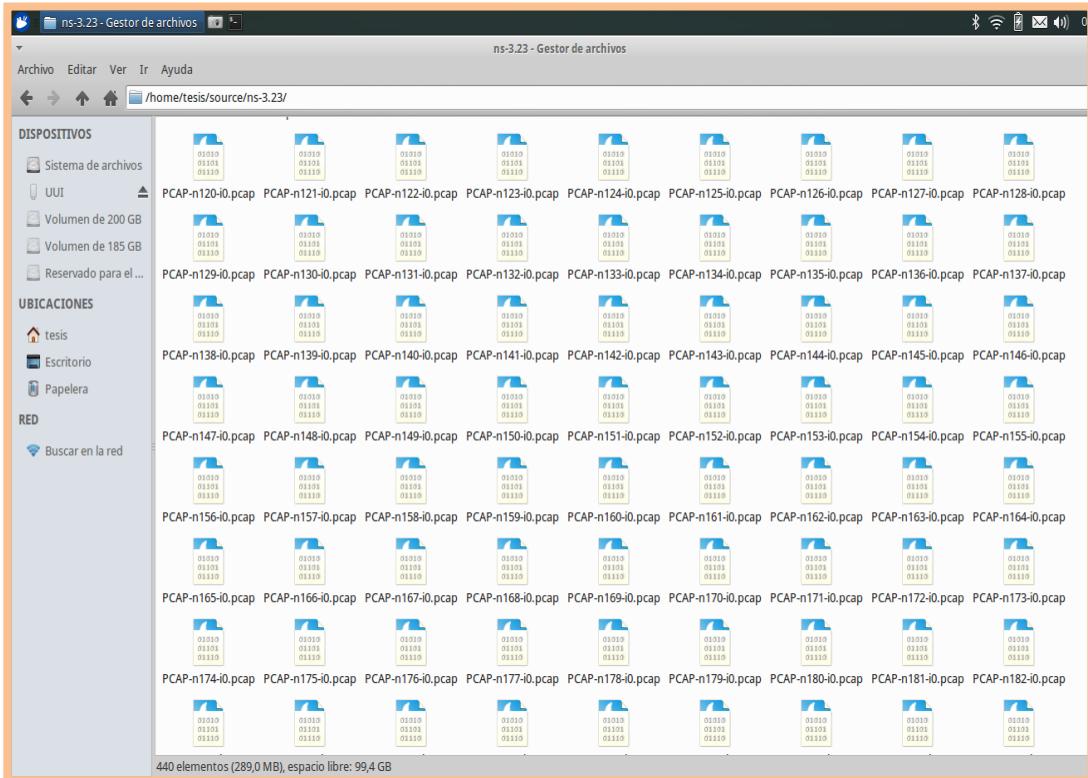


Figura 81: Archivos generados .pcap.

Estos archivos se los puede analizar con **wireshark** la figura 82 muestra la captura de paquetes generados con los archivos .pcap

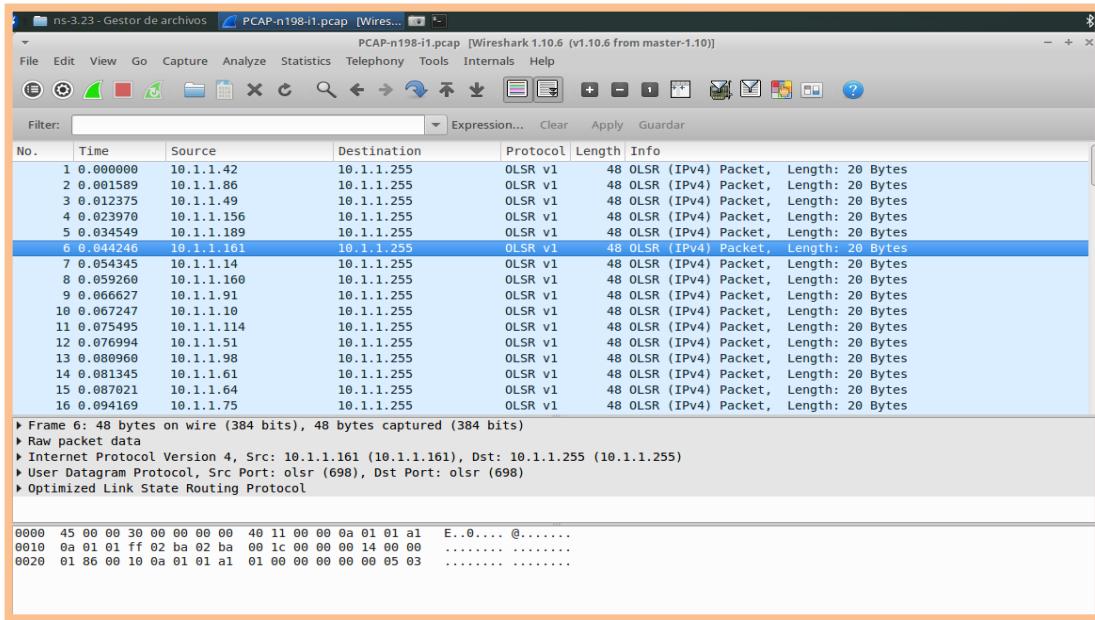


Figura 82: Análisis de archivos .pcap en wireshark.

Otra manera para el análisis es los archivos generados con flowmonitor que tienen las métricas que por defecto se presentan en la figura 83 se puede observar brevemente la estructura de estos archivos .xml

```
<FlowStats>
<Flow timesForwarded="230" lostPackets="14" rxPackets="368" txPackets="399" rxBytes="33856" txBytes="36708" lastDelay="+2238251.0ns"
      jitterSum="+6317756016.0ns" delaySum="+7781200454.0ns" timeLastRxPacket="+199812262598.0ns" timeLastTxPacket="+199810024347.0ns"
      timeFirstRxPacket="+100321049206.0ns" timeFirstTxPacket="+100310024347.0ns" flowId="1"></Flow>
<Flow timesForwarded="150" lostPackets="0" rxPackets="399" txPackets="398" rxBytes="36616" txBytes="36616" lastDelay="+286892.0ns" jitterSum="+2563358265.0ns"
      delaySum="+1605622509.0ns" timeLastRxPacket="+199808113306.0ns" timeLastTxPacket="+199807826414.0ns" timeFirstRxPacket="+100581094043.0ns"
      timeFirstTxPacket="+100557826414.0ns" flowId="2"></Flow>
<Flow timesForwarded="93" lostPackets="6" rxPackets="370" txPackets="399" rxBytes="34040" txBytes="36616" lastDelay="+3205161.0ns" jitterSum="+376772109.0ns"
      delaySum="+447600837.0ns" timeLastRxPacket="+198084228710.0ns" timeLastTxPacket="+199831023549.0ns" timeFirstRxPacket="+100594168395.0ns"
      timeFirstTxPacket="+100581023549.0ns" flowId="3"></Flow>
-<Flow timesForwarded="607" lostPackets="62" rxPackets="282" txPackets="347" rxBytes="25944" txBytes="31924" lastDelay="+937306.0ns"
      jitterSum="+40770398041.0ns" delaySum="+27078860445.0ns" timeLastRxPacket="+199844789969.0ns" timeLastTxPacket="+199843852663.0ns"
      timeFirstRxPacket="+100640013138.0ns" timeFirstTxPacket="+100593852663.0ns" flowId="4">
  <packetsDropped number="0" reasonCode="0"/>
  <packetsDropped number="8" reasonCode="1"/>
  <bytesDropped reasonCode="0" bytes="0"/>
  <bytesDropped reasonCode="1" bytes="736"/>

```

Figura 83: Archivo .xml generados en la simulación.

3.13 Análisis de las métricas obtenidas

Haciendo uso de los archivos obtenidos durante la simulación, tanto en **.xml**, **.flowmon** y **.pcap**, realizando un análisis de trazas con **netamin** propio para Ns3, y **wireshark** en el caso de **DSR**, ya que se considera que DSR es un protocolo más utilizado para redes de tipo TCP/IP y Netanim no logra recolectar ningún tipo de datos de este tipo, se logró obtener las estadísticas que se detallan a continuación:

Para el análisis de algunas de las métricas recolectadas, en el caso de DSR que es diferente a los demás se ha podido obtener algunas estadísticas propias de este protocolo, y con respecto a AODV, DSDV y OLSR se obtuvieron todas las métricas ya que estos funcionan con el módulo propio para monitoreo **FlowMonitor**, de Ns-3.

Primeramente hago referencia en las figuras a continuación los archivos que se obtuvieron mediante Netanim.

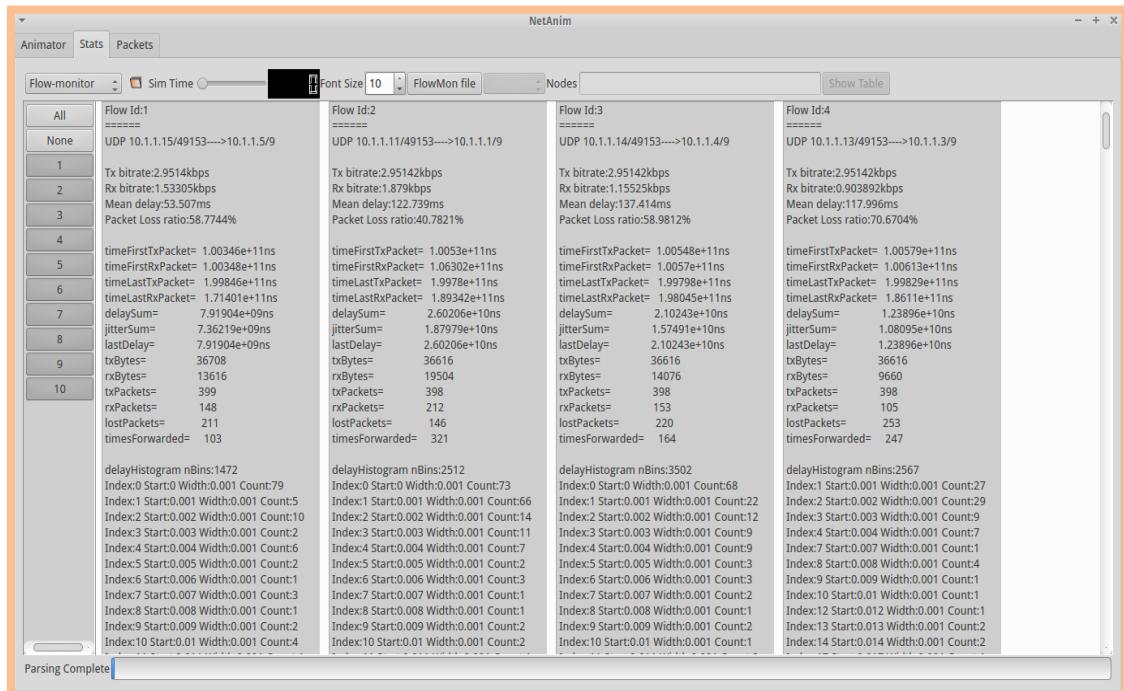


Figura 84: Resultados de las métricas obtenidas en Netanim.

En la figura 84 se observa el resultado que se obtiene al leer el archivo .xml generado luego de la simulación, este nos muestra un conjunto de datos relacionados con las métricas obtenidas al correr la simulación, este tiene un archivo por cada corrida, es decir se analizó ocho archivos .xml que resultan de monitorear la red o la simulación con flowmonitor.

En el caso de AODV y DSR se generó 20 síntesis, AODV analizado mediante Flowmonitor y Drs con Wireshark como se muestra en la figura 85. Y en el caso de OLSR y DSDV se generaron 10 síntesis generadas y leídas con Flowmonitor.

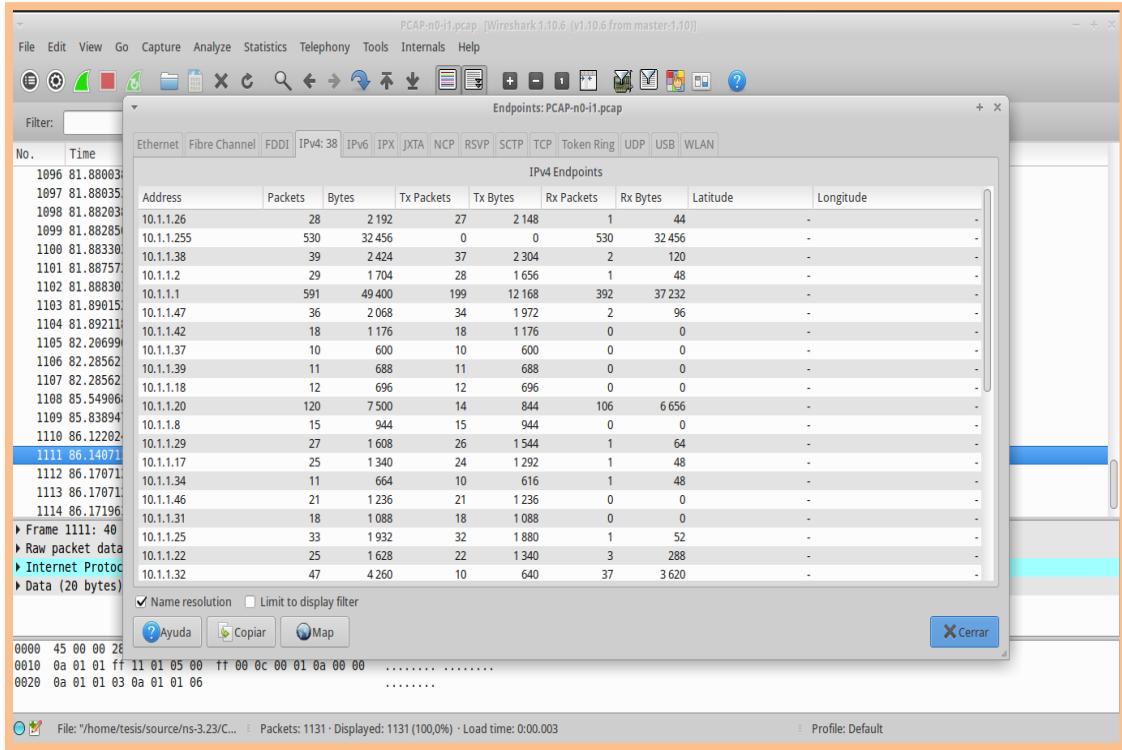


Figura 85: Resultados de las métricas obtenidas en Netanim.

3.13.1 Escenario de prueba uno

3.13.1.1 Time First Tx Packets

Los datos recogidos para las estadísticas entre los diferentes protocolos de acuerdo al escenario de prueba uno, están detallados en la **TABLA XV**, en este caso DSR es el protocolo que no puede obtener la métrica de **TimeFirstTx Packets** que se refiere a el tiempo en que el primer paquete fue transmitido. En este escenario se aprecia según la tabla que, tanto AODV, OLSR y DSDV empiezan sus transmisiones al mismo tiempo, luego de esto AODV es el más estable en cuanto a división de tiempo de transmisión, OLSR y DSDV tienen variación en los tiempos.

TABLA XV. ANÁLISIS DE LA MÉTRICA TIME FIRST TX PACKETS ESCENARIO I

AODV	DSR	OLSR	DSDV
1,00289	N/A	1,00289	1,00289
1,00295	N/A	1,00417	1,00417
1,00295	N/A	1,00454	1,00454
1,00295	N/A	1,0053	1,00463
1,00295	N/A	1,00666	1,00483
1,00306	N/A	1,00733	1,0053
1,00306	N/A	1,0111	1,00666
1,00306	N/A	1,01218	1,00733
1,00306	N/A	1,05483	1,0111
1,0031	N/A	1,05713	1,02128

Las medidas obtenidas durante diez recogidas de datos con flowmonitor, se representan en la figura 86, graficada a partir de la **TABLA XV**, la que confirma que AODV es el protocolo para este escenario que se mantiene estable y que termina de transmitir primero sus paquetes, seguido de DSDV y OLSR como última opción.

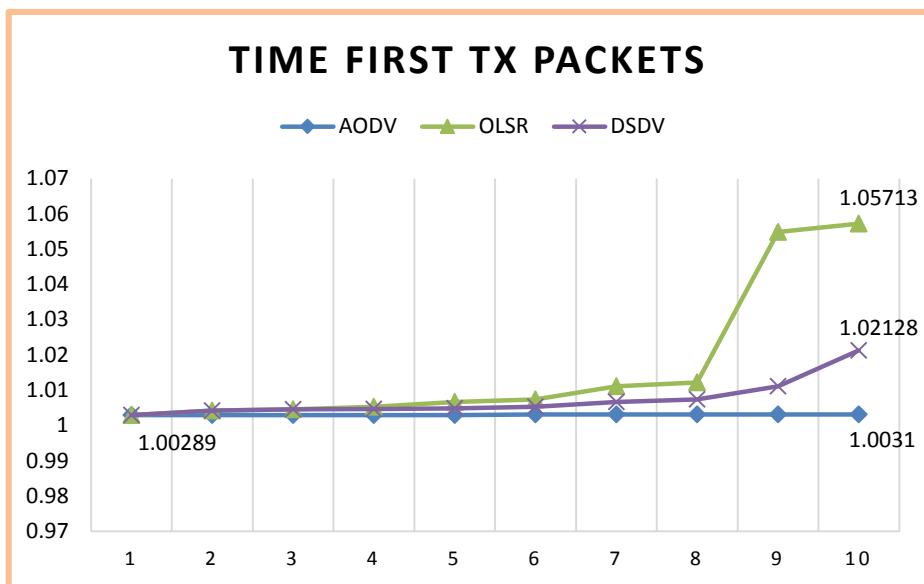


Figura 86: Línea estadística de la métrica TxPackets escenario I.

Otra manera de comprobar es mediante el cálculo de promedio entre tiempos de transmisión de paquetes, así mismo se debe tomar en cuenta el protocolo que menor tiempo utiliza para la transmisión, ya que esto puede deberse a entrega más rápida de los paquetes por lo que los enlaces se desocupan y están listos para una nueva transmisión, este análisis se lo puede observar en la figura 87.

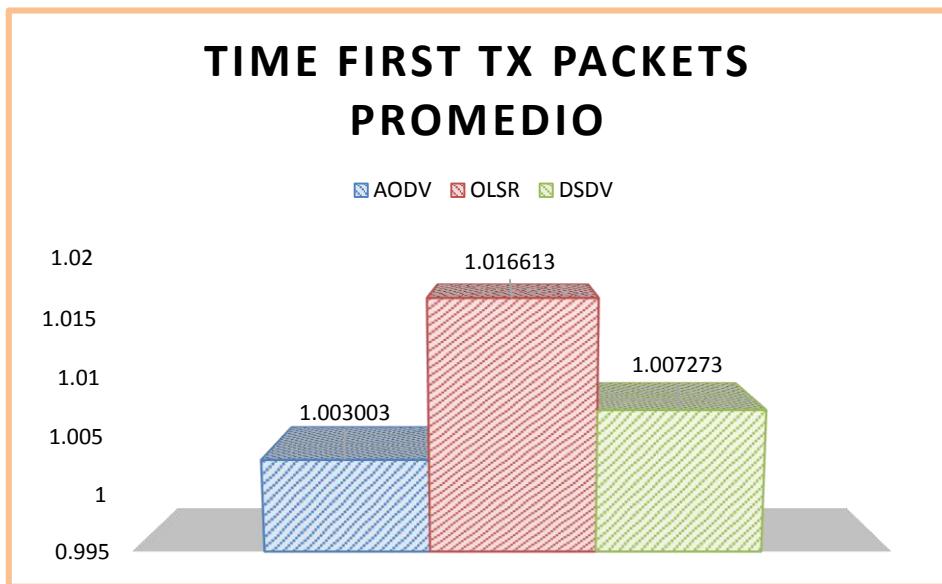


Figura 87: Promedio de la métrica Time First TxPackets escenario I.

3.13.1.2 Time First Rx Packets

Esta métrica en cambio define el tiempo en el que llego al destino el primer paquete enviado, lo que nos ayuda a establecer en cambio que tan efectivas son las rutas que elige cada protocolo, y están definidas en la **TABLA XVI**, en donde se puede apreciar de la misma manera mayor estabilidad en el protocolo AODV, y variación con DSDV y OLSR, con DSR no es posible obtener esta métrica.

TABLA XVI. ANÁLISIS DE LA MÉTRICA TIME FIRST RX PACKETS ESCENARIO I

AODV	DSR	OLSR	DSDV
1,00289	N/A	1,00289	1,00289
1,00295	N/A	1,00417	1,00417
1,00295	N/A	1,00454	1,00454
1,00295	N/A	1,0053	1,00463
1,00295	N/A	1,00666	1,00483
1,00306	N/A	1,00733	1,0053
1,00306	N/A	1,0111	1,00666
1,00306	N/A	1,01218	1,00733
1,00306	N/A	1,05483	1,0111
1,0031	N/A	1,05713	1,02128

Basados en la **TABLA XVI**, se puede obtener la variación de cada protocolo en cuanto a la medida **TimeFirstRxPackets** y se la puede apreciar en la figura 88, en donde se puede observar de mejor manera la variación del tiempo que tiene cada protocolo a lo largo de la recepción de paquetes.

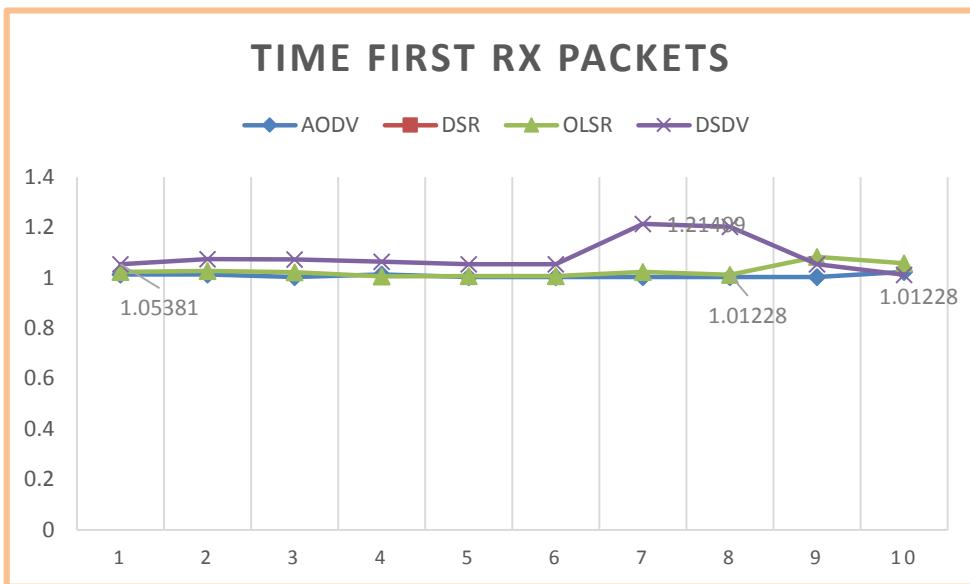


Figura 88: Línea estadística de la métrica Time First RxPackets escenario I.

Y finalmente el promedio de tiempo de recepción de paquetes por protocolo, en este se aprecia que AODV lidera el tiempo más rápido de recepción, seguido de OLSR y en tercer lugar DSDV, apreciados en la figura 89.

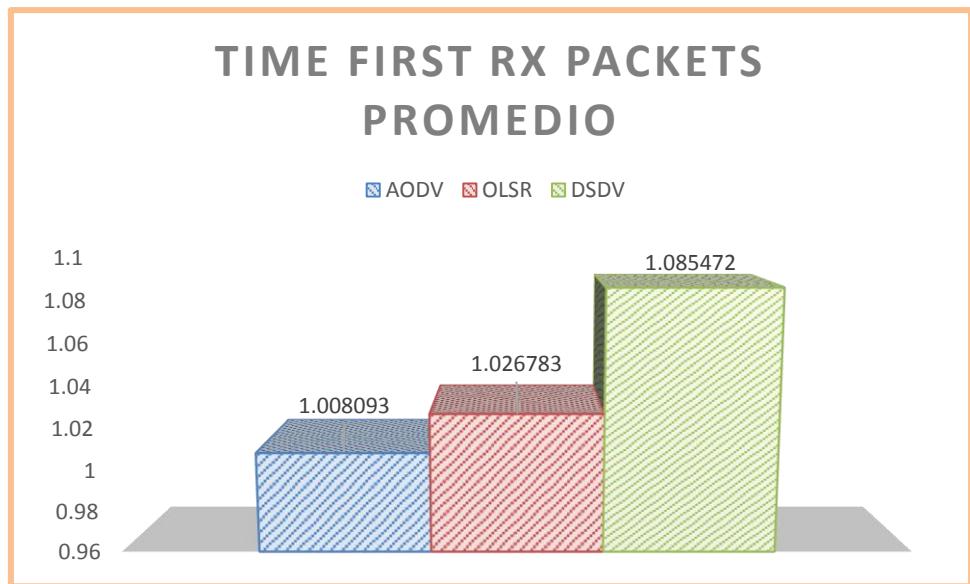


Figura 89: Promedio de la métrica Time First RxPackets escenario I.

3.13.1.3 Time Last Tx Packets

Esta métrica en cambio ayuda a medir el tiempo en que el último paquete ha sido transmitido, en la **TABLA XVII** se puede apreciar hay un poco más de estabilidad con OLSR y DSDV, mientras que AODV es más variante y con DSR no se puede obtener estadísticas.

TABLA XVII. ANÁLISIS DE LA MÉTRICA TIME LAST TX PACKETS ESCENARIO I

AODV	DSR	OLSR	DSDV
1,99789	N/A	1,64303	1,99789
1,03826	N/A	1,81207	1,99917
1,99314	N/A	1,99956	1,99954
1,06036	N/A	1,81299	1,99963
1,82219	N/A	1,86166	1,99983
1,96785	N/A	1,99983	1,9978
1,49189	N/A	1,9186	1,99916
1,1171	N/A	1,99968	1,99983
1,67592	N/A	1,91983	1,9986
1,18746	N/A	1,90213	1,99968

Así mismo de la **TABLA XVII** se puede obtener el grafico para poder observar la variación de la métrica entre protocolos, como vemos AODV tiene variaciones grandes, y el más estable es DSDV apreciado en la figura 90.

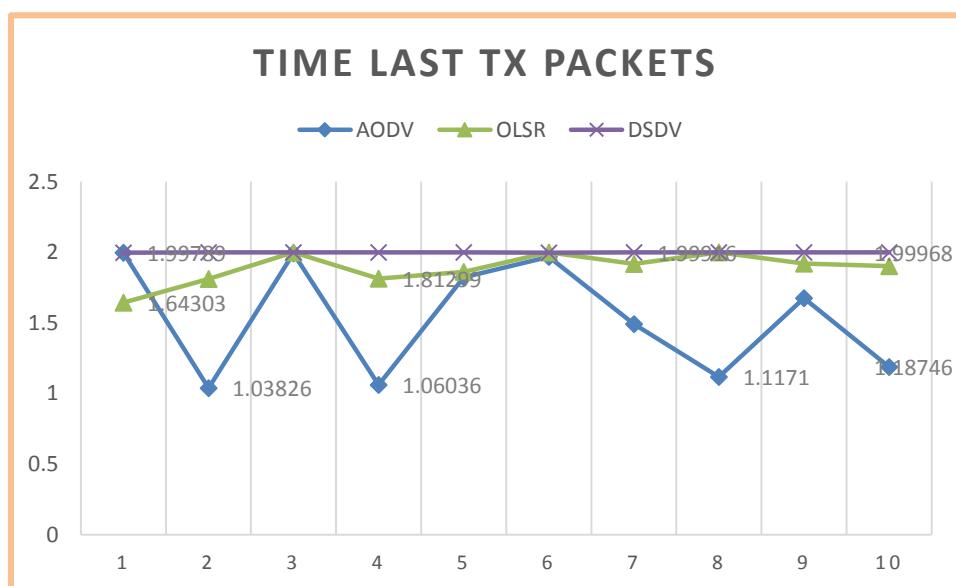


Figura 90: Línea estadística de la métrica Time Last TxPackets escenario I,

Ya en términos de promedio se puede definir que el mejor tiempo de envío del último paquete, según las estadísticas recogidas en diferentes tiempos es la de AODV a pesar

de sus cambios bruscos, seguido de OLSR y en tercer lugar DSDV, que se lo observa en la figura 91.

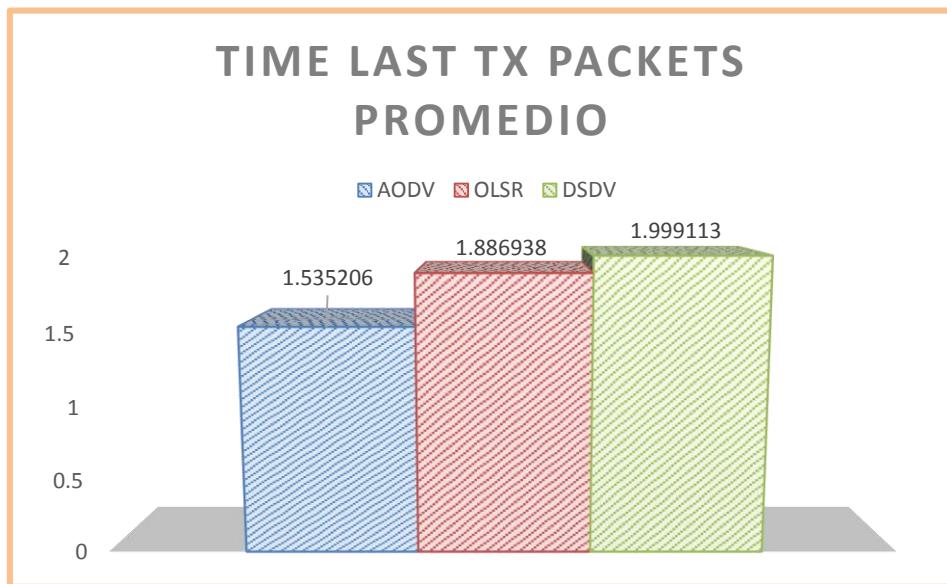


Figura 91: Promedio de la métrica Time Last TxPackets escenario I.

3.13.1.4 Time Last Rx Packets

Ahora seguimos con la métrica que nos ayuda al cálculo del mejor tiempo de entrega del último paquete, en la **TABLA XVIII** se establecen diez medidas tomadas en diferentes tiempos, según cada protocolo, sin tomar en cuenta la medida de DSR que no se la puede obtener.

TABLA XVIII. ANÁLISIS DE LA MÉTRICA TIME LAST RX PACKETS ESCENARIO I

AODV	DSR	OLSR	DSDV
1,99791	N/A	1,64303	1,94541
1,01301	N/A	1,81207	1,60453
1,99314	N/A	1,99956	1,99956
1,04058	N/A	1,81299	1,80339
1,8221	N/A	1,80919	1,95342
1,96785	N/A	1,99984	1,99785
1,49189	N/A	1,70615	1,76188
1,1171	N/A	1,99968	1,66341
1,67596	N/A	1,9199	1,05361
1,11737	N/A	1,90215	1,01228

A partir de la **TABLA XII** se obtuvo la figura 92, en la cual se puede observar que AODV tienen de nuevo el mayor número de cambios bruscos, en cuanto a DSDV tiene de ir de

tiempos altos a tiempos bajos, y finalmente OLSR que al contrario va de tiempos bajos a tiempos altos de entrega de paquetes.

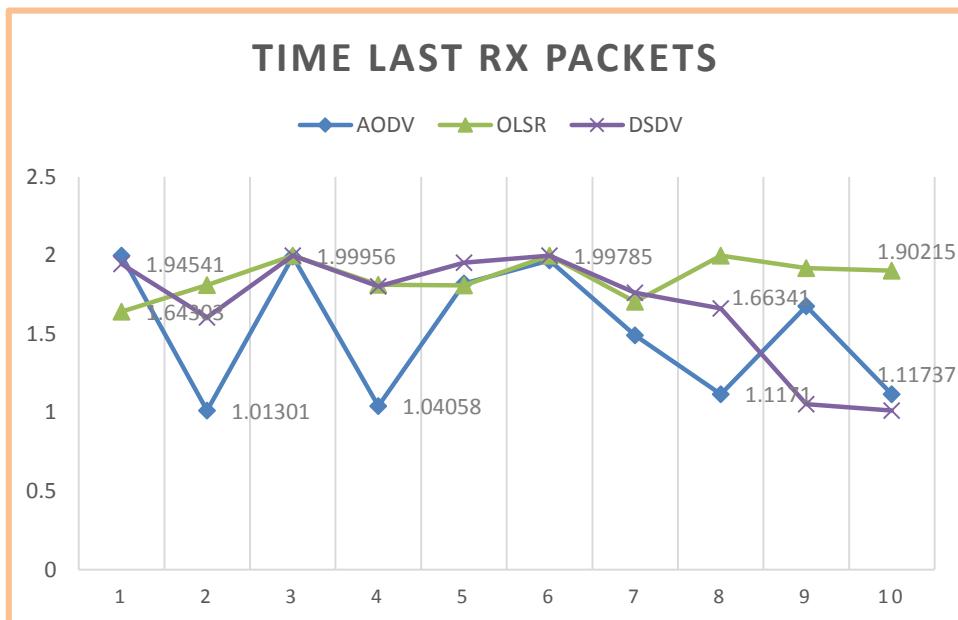


Figura 92: Línea estadística de la métrica Time Last RxPackets escenario I.

En cuanto a promedios AODV sigue liderando en el tiempo más corto de envío del último paquete, seguido de DSDV y en tercer lugar OLSR, con DSR no se ha podido obtener esta métrica, la figura 93 muestra las estadísticas.

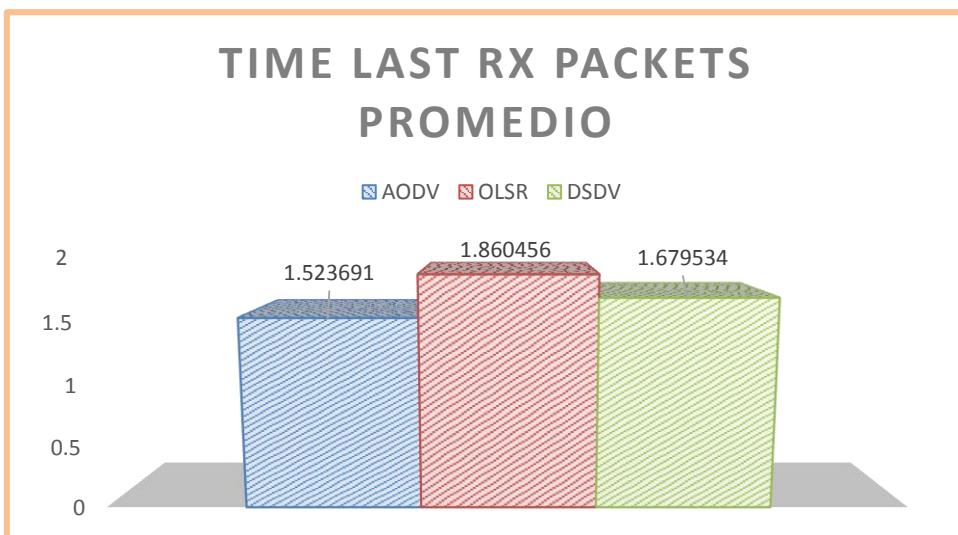


Figura 93: Promedio de la métrica Time Last RxPackets escenario I.

3.13.1.5 Delay

Delay es una de las métricas principales a ser tomadas en cuenta dentro del rendimiento de una red, y establece el retraso de extremo a extremo es decir **source/destination**, según la **TABLA XIX** en AODV puede llegar el delay a un 6,30808ns, OLSR hasta 8,21575ns y DSDV hasta un 9,39598ns.

TABLA XIX. ANÁLISIS DE LA MÉTRICA DELAY ESCENARIO I

AODV	DSR	OLSR	DSDV
6,30808	N/A	4,52217	3,95117
2,69001	N/A	8,21575	7,63899
1,04381	N/A	4,02744	9,39598
2,69831	N/A	1,23347	8,42636
1,65599	N/A	1,86272	2,55194
3,61285	N/A	7,97546	1,53442
1,57865	N/A	2,22502	7,72174
1,77607	N/A	1,4991	1,80656
1,29559	N/A	1,35225	7,06168
1,05165	N/A	5,11716	3,16036

En la figura 94 se puede apreciar las tendencias de cada recogida de datos, para los tres protocolos, observándose grandes cambios bruscos en todos, así mismo los topes máximos que alcanzan cada uno de ellos.

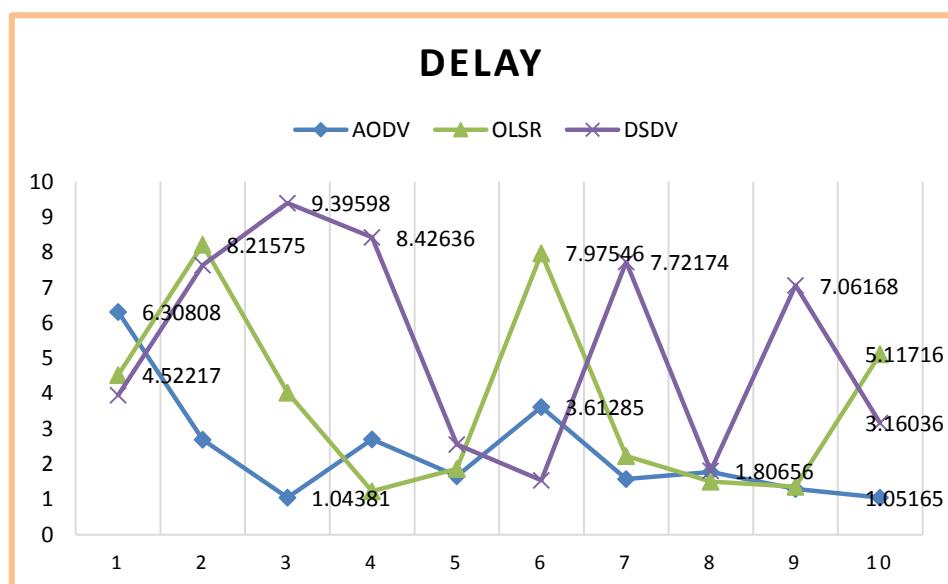


Figura 94: Línea estadística de la métrica Delay escenario I.

También se realizó un promedio de retraso obtenido en varios tiempos, lo que nos ayuda a definir un tiempo de retraso en cada protocolo como lo muestra la figura 95, AODV lidera con menor tiempo de retraso, seguido de OLSR y en tercer lugar DSDV.

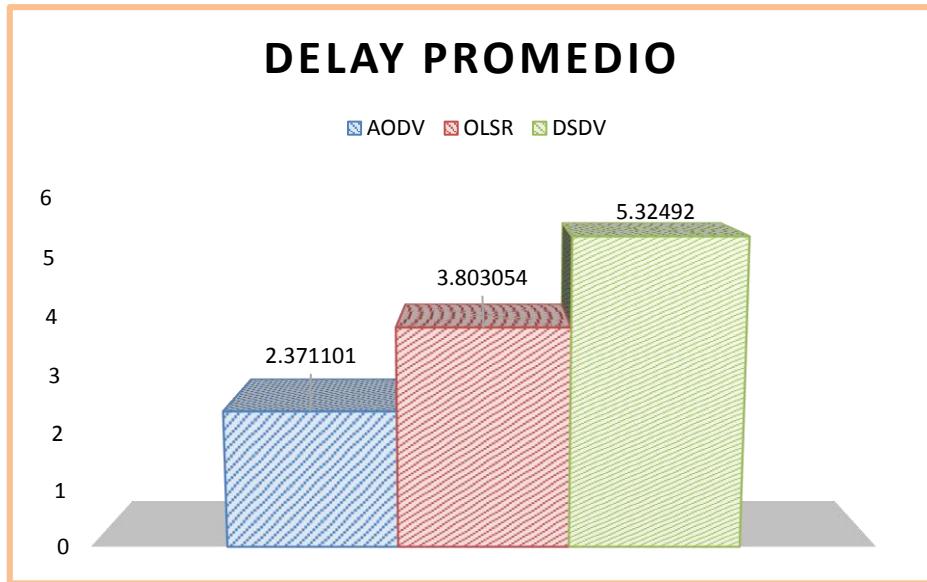


Figura 95: Promedio de la métrica Delay escenario I.

3.13.1.6 Jitter

Es la fluctuación o los cambios que sufre el retraso (Delay), en la **TABLA XX** se aprecia los valores que ha tomado esta métrica en diez capturas de monitoreo.

TABLA XX. ANÁLISIS DE LA MÉTRICA JITTER ESCENARIO I

AODV	DSR	OLSR	DSDV
2,73973	N/A	4,25566	6,78479
1,60524	N/A	7,94375	6,57903
1,0966	N/A	3,89412	6,33946
1,01778	N/A	1,09261	7,94228
1,75669	N/A	2,98102	3,14108
3,63034	N/A	6,45498	1,37225
1,25457	N/A	2,40902	7,10135
1,68459	N/A	5,14597	1,65423
1,85765	N/A	9,72326	6,70978
1,02261	N/A	5,04431	3,9055

De la misma manera el Jitter tiene su figura tomando en cuenta la **TABLA XX**, esto se representa en la figura 96, y se puede ver gráficamente la variación del retraso sus valores más altos como sus valores más bajos. Siendo tomado en cuenta los límites de

la gráfica, OLSR el que presenta mayor fluctuación, seguido de DSDV y en tercer lugar OLSR.

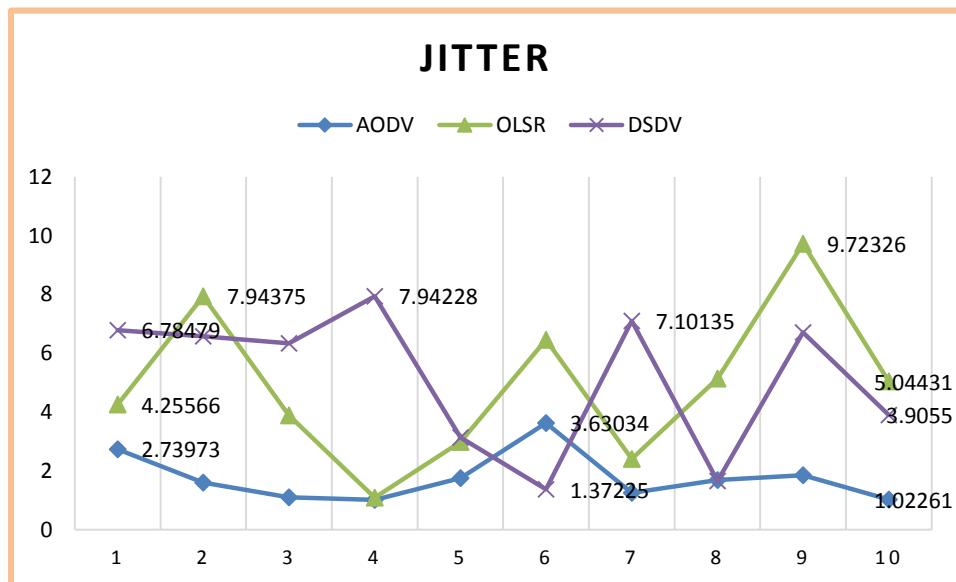


Figura 96: Línea estadística de la métrica Jitter escenario I.

Con respecto a valores tomados como promedios en la fluctuación de la simulación total, la figura 97 muestra el menor número de fluctuación en AODV, seguido de OLSR y por ultimo DSDV.

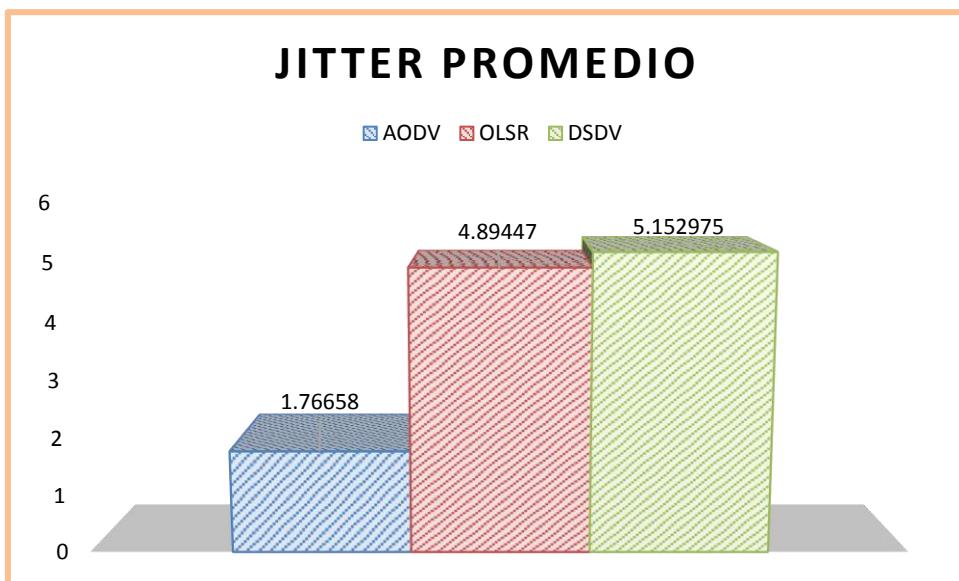


Figura 97: Promedio de la métrica Jitter escenario I.

3.13.1.7 Tx Bytes

Luego se toma en cuenta la métrica del número de bytes enviados, en la **TABLA XXI**, se aprecia el número de bytes por cada monitoreo realizado, esta métrica ya es posible obtener datos de DSR.

TABLA XXI. ANÁLISIS DE LA MÉTRICA TX BYTES ESCENARIO I

AODV	DRS	OLSR	DSDV
36708	8388	13708	36708
336	13332	19044	36708
500	4160	28796	36708
312	35400	22632	36708
876	169772	16468	36708
1754	10720	34868	36616
754	79480	21804	36616
144	12444	36432	36616
432	124	25484	36616
186	57484	30636	36432

Y en la gráfica basada en la **TABLA XXI** se aprecia el número y el cambio que tiene cada protocolo con respecto a la tasa de bytes enviados en cada tiempo monitoreado siendo el mayor número de bytes enviados tomando en cuenta los límites altos, el de DSR, seguido de DSDV, luego OLSR y por ultimo AODV, se muestra en la figura 98.

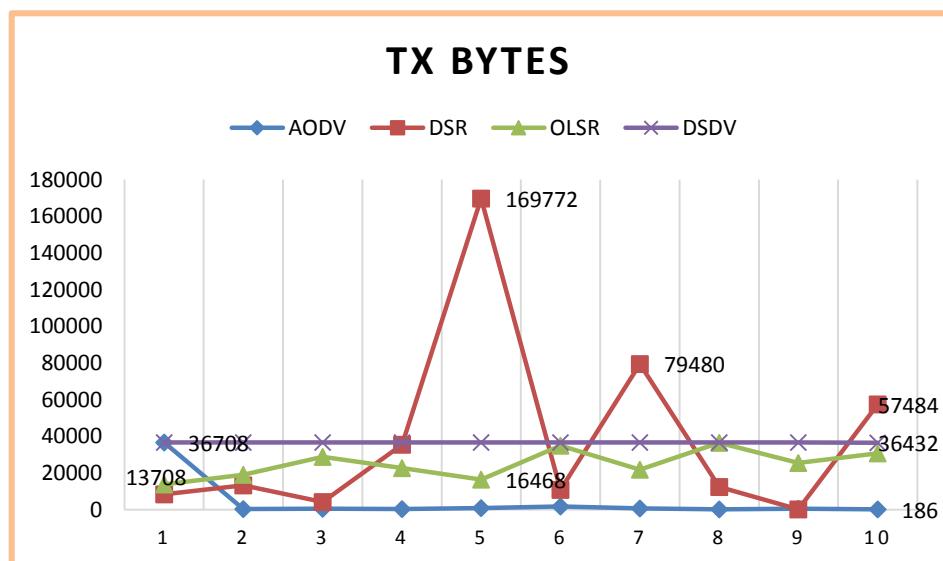


Figura 98: Línea estadística de la métrica TxBytes escenario I.

Con respecto al promedio de los datos tomados en cuenta en envío de Bytes, se puede observar que la mayor tasa de bytes enviados recae en DSR, seguido de DSDV, luego

OLSR y finalmente AODV, razón por la cual se puede justificar en cierta parte el éxito en cuanto a tiempos de entrega más cortos, tomando en cuenta la figura 99.

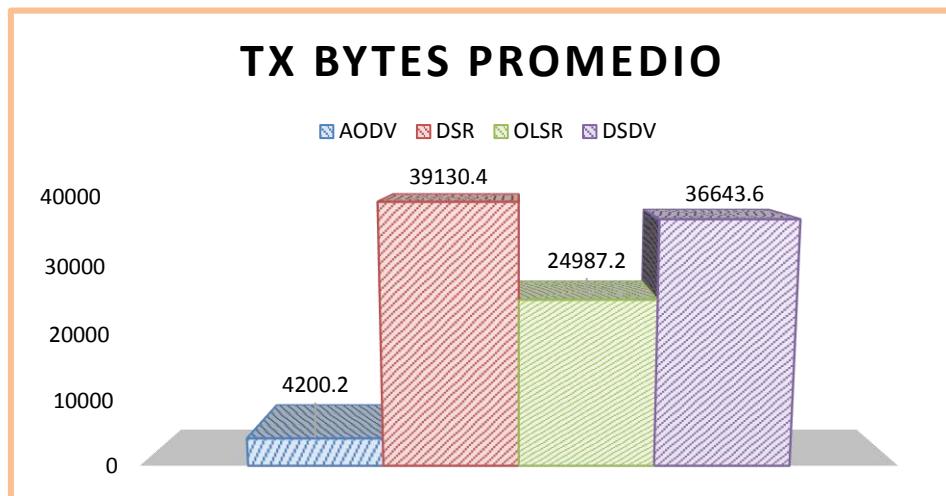


Figura 99: Promedio de la métrica TxBytes escenario I.

3.13.1.8 Rx Bytes

Ahora cabe analizar de estos bytes enviados cuantos llegan con éxito que es la métrica tomada en cuenta en la **TABLA XXII**, recogida así mismo en diferentes tiempos de simulación, para los cuatro protocolos.

TABLA XXII. ANÁLISIS DE LA MÉTRICA RX BYTES ESCENARIO I

AODV	DSR	OLSR	DSDV
15088	5216	2116	4692
144	0	11408	7268
500	3280	22632	20240
272	220	13064	17296
846	62120	7452	13800
1532	5984	24748	11868
706	31788	17296	4692
144	620	36432	7544
432	124	16652	13432
138	192	24932	36432

En la figura 100 se aprecia la variación en los diferentes tiempos del número de bytes recibidos por el destino, apreciándose que el mayor número en cuanto a límites de bytes recibidos se da en DSR, seguido OLSR, DSDV y finalmente AODV con menor número de bytes recibidos.

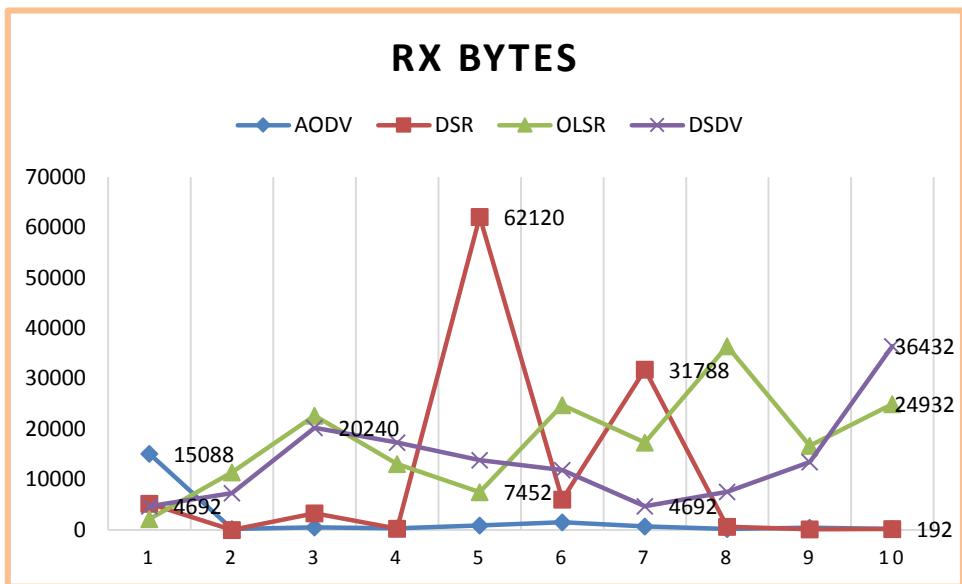


Figura 100: Línea estadística de la métrica RxBytes escenario I.

Ahora con respecto a promedios que nos da una medida más exacta del valor en el tiempo total de simulación, nos da una entrega de bytes mayor al destino por parte de OLSR, seguido de DSDV, luego DSR y por ultimo AODV, apreciado en la figura 101.

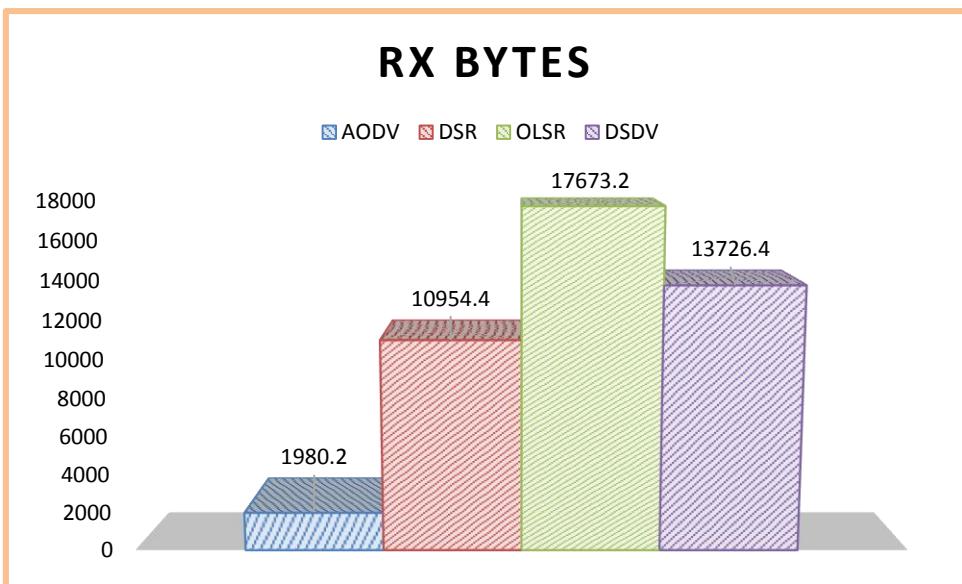


Figura 101: Promedio de la métrica RxBytes escenario I.

3.13.1.9 Tx Packets

El número de paquetes enviados por cada protocolo también es analizado en la **TABLA XXIII**, siendo el más estable en cuanto a número DSDV, luego AODV, seguido de los protocolos con valores variantes DSR y OLSR.

TABLA XXIII. ANÁLISIS DE LA MÉTRICA TX PACKETS ESCENARIO I

AODV	DSR	OLSR	DSDV
399	94	149	399
7	233	207	399
11	45	313	399
7	296	246	399
19	1587	179	399
38	108	379	398
17	690	237	398
3	117	296	398
9	2	277	396
5	465	333	396

En la figura 102, se puede apreciar también la variación y estabilidad en cuanto a número de paquetes enviados, respecto de cada protocolo, se alcanza a observar DSR y OLSR con variaciones y acercándose a la estabilidad a AODV y DSDV.

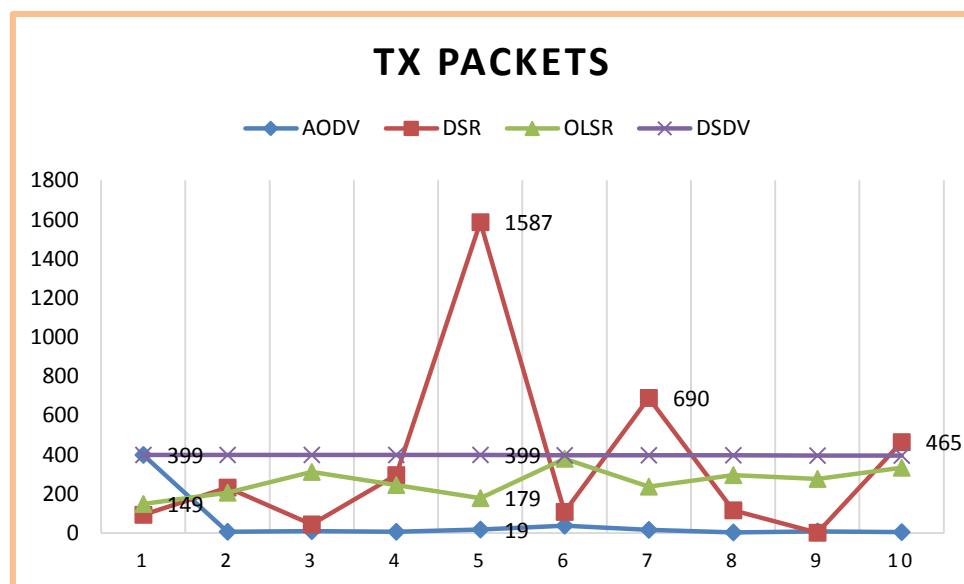


Figura 102: Línea estadística de la métrica TxPackets escenario I.

Ahora obteniendo una métrica en general, en la figura 103 en cuanto a número de paquetes enviados por cada protocolo, tenemos que DSDV es el protocolo que tiene

una tasa más alta de paquetes enviados, seguido de DSR, a continuación OLSR y finalmente AODV.

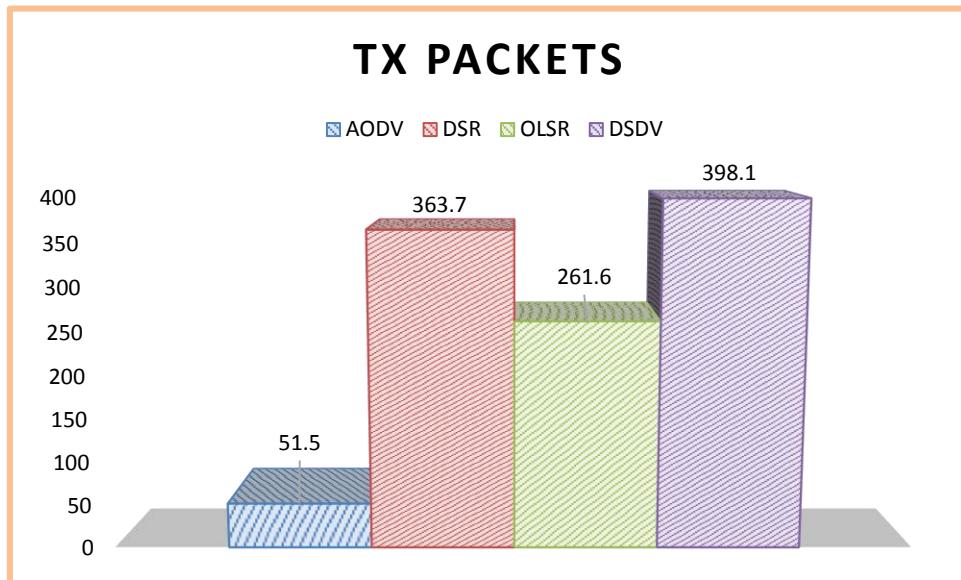


Figura 103: Promedio de la métrica TxPackets escenario I.

3.13.1.10 Rx Packets

Esta métrica en cambio define cuantos paquetes del total de paquetes enviado se entregaron con éxito, lo que nos ayuda a establecer en cambio que tan confiables son las rutas que elige cada protocolo, y están definidas en la **TABLA XXIV**.

TABLA XXIV. ANÁLISIS DE LA MÉTRICA RX PACKETS ESCENARIO I

AODV	DSR	OLSR	DSDV
164	47	23	51
3	0	124	79
11	37	246	220
6	4	142	188
18	681	2,98	150
33	67	6,45	129
16	291	2,41	51
3	13	5,45	82
9	2	9,72	146
4	3	5,04	396

El número de paquetes entregados con éxitos tiene variaciones dependiendo del protocolo y del momento en que fue monitoreado, en cuanto a límite máximo DSR alcanza el mayor número de paquetes entregados con éxito, seguido de OLSR, luego DSDV y por último AODV, que se puede apreciar en la figura 104.

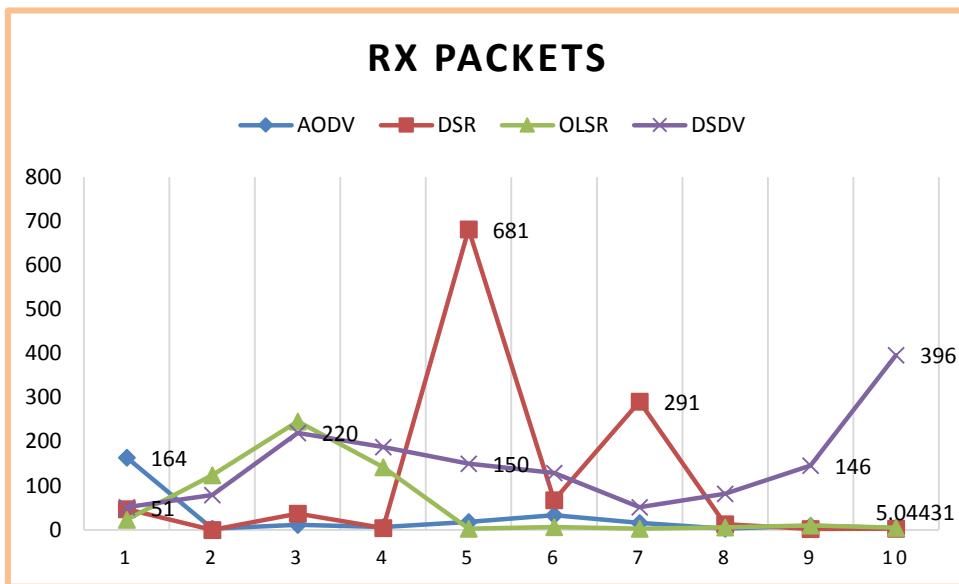


Figura 104: Línea estadística de la métrica RxPackets escenario I.

Ahora un valor establecido para cada protocolo mediante el promedio, estos valores se puede observar en la figura 105, y nos dan como resultado que el protocolo que ofrece una mayor entrega de paquetes seguro es DSDV, seguido de DSR, luego OLSR y finalmente AODV.

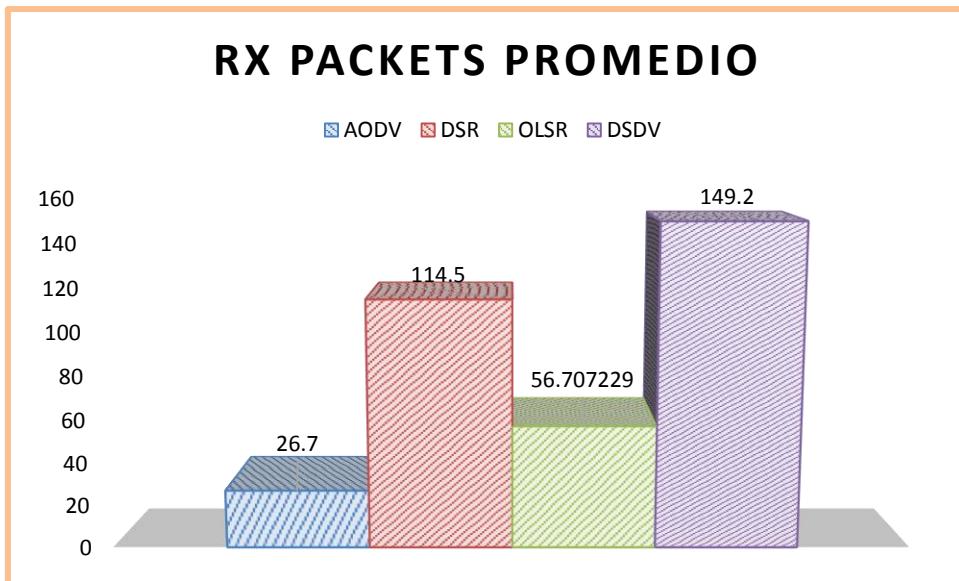


Figura 105: Promedio de la métrica RxPackets escenario I.

3.13.1.11 Lost Packets

Otra métrica a ser evaluada es el número de paquetes perdidos, mientras menor sea el número de paquetes más satisfactoria es la experiencia con el paquete. Las métricas tomadas en diferentes tiempos de monitoreo se pueden apreciar en la **TABLA XXV**.

TABLA XXV. ANÁLISIS DE LA MÉTRICA LOST PACKETS ESCENARIO I

AODV	DRS	OLSR	DSDV
216	47	126	315
4	233	83	280
0	8	67	179
1	292	104	171
1	906	98	223
5	41	110	238
1	399	49	307
0	104	0	276
0	0	94	210
1	462	62	0

En la figura 106 se puede apreciar el número de paquetes perdidos en diferentes tiempos y tomados en cuenta límites máximos se aprecia que DSR está en primer lugar con mayor número de paquetes perdidos, seguido de DSDV, luego OLSR y finalmente AODV.

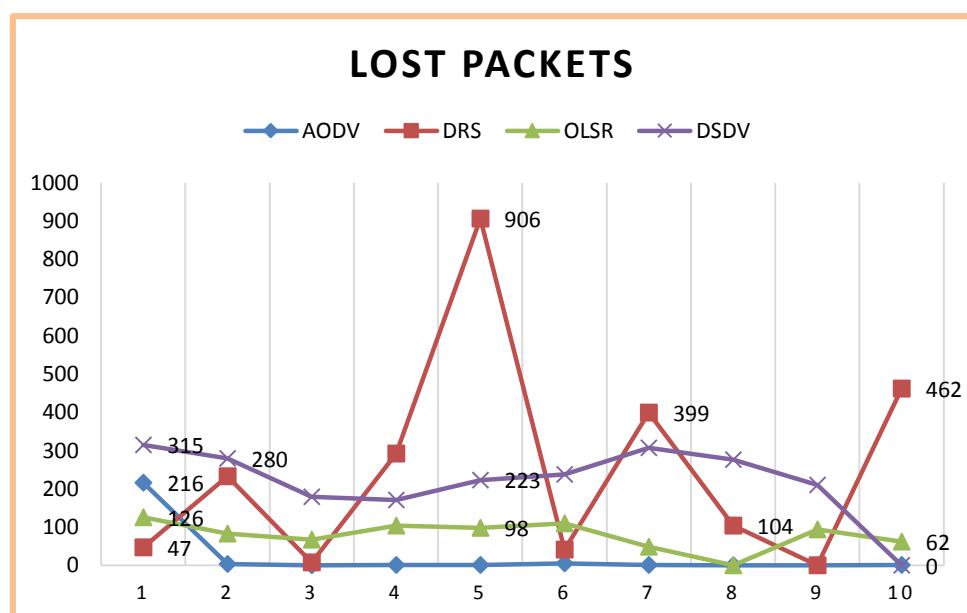


Figura 106: Línea estadística de la métrica LostPackets escenario I.

Ahora en la figura 107, se aprecia un promedio de paquetes perdidos siendo DSR el protocolo que más paquetes perdidos tiene, seguido de DSDV, luego OLSR y AODV.

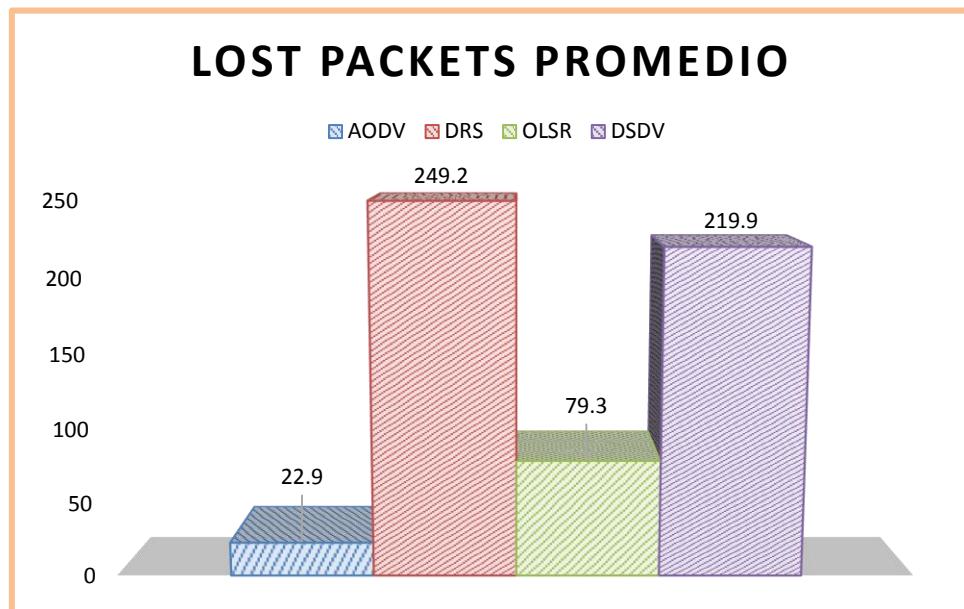


Figura 107: Promedio de la métrica LostPackets escenario I.

3.13.1.12 Cálculo de Troughput

Otro indicador importante para definir la excelencia en cuanto a uso de protocolos en el escenario de prueba uno es el Troughput, que es la tasa promedio de éxito en la entrega de un mensaje sobre un canal es comunicación en este caso es un enlace lógico. En la **TABLA XXVI** se observa los valores obtenidos para cada protocolo[30].

TABLA XXVI. CÁLCULO DE TROUGHPUT ESCENARIO I

AODV	DSR	OLSR	DSDV
9,901	54,772	88,366	68,632

Así mismo en la figura 108 se observa las estadísticas en grafico de barras, el cual nos muestra mayor throughput para OLSR, seguido de DSDV, DSR y por ultimo AODV.

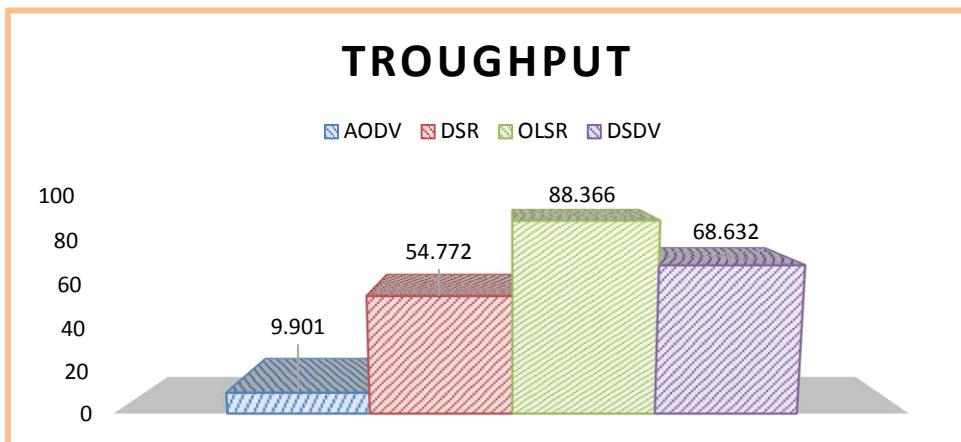


Figura 108: Cálculo de Troughput escenario I.

3.13.1.13 Cálculo de Tasa de Entrega

Como último indicador tenemos la tasa de entrega, en la **TABLA XXVII** se reflejan los resultados obtenidos para esta métrica, siendo AODV el que tiene mayor tasa de entrega, le sigue DSDV, DSR y OLSR[30].

TABLA XXVII. CÁLCULO DE TASA DE ENTREGA ESCENARIO I

AODV	DSR	OLSR	DSDV
51,845	31,482	21,677	37,478

En la figura 109 se aprecia los rangos de la métrica de tasa de entrega para todos los protocolos, en donde se manifiesta de forma gráfica los resultados obtenidos.

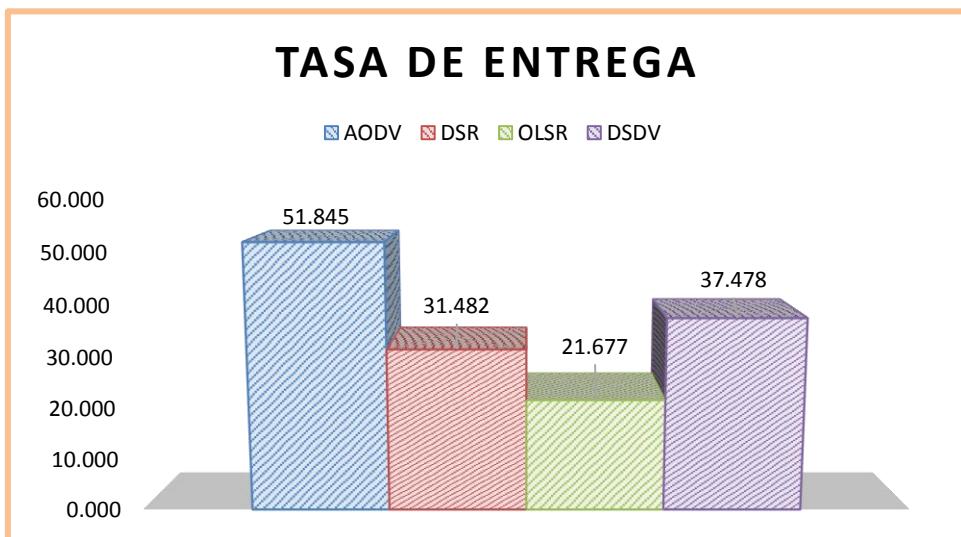


Figura 109: Cálculo de Tasa de entrega escenario I.

3.13.2 Escenario de prueba dos

Ahora empezamos con el escenario de prueba dos, en el que existen 50 nodos lo que aumenta el tráfico en la red, y por ende se podrá obtener nuevas métricas, y comportamientos de cada protocolo según las nuevas exigencias.

3.13.2.1 Time First Tx Packets

Igual que en el escenario anterior analizamos el menor tiempo a la hora de enviar el primer paquete, estos valores tomados en diferentes tiempos y reflejados en la **TABLA XXVIII**, así mismo con DSR no se puede obtener estas métricas ya sea con Wireshark o con Flowmonitor.

TABLA XXVIII. ANÁLISIS DE LA MÉTRICA TIMEFIRSTTXPACKETS ESCENARIO II

AODV	DSR	OLSR	DSDV
1,00346	N/A	1,00346	1,00346
1,0053	N/A	1,00548	1,0053
1,00533	N/A	1,00579	1,00548
1,00533	N/A	1,00584	1,00579
1,00533	N/A	1,00597	1,00584
1,00533	N/A	1,00885	1,00597
1,00533	N/A	1,01021	1,00767
1,00533	N/A	1,01175	1,00885
1,00542	N/A	1,0553	1,01021
1,00542	N/A	1,10767	1,01175

En base a la tabla anterior, se puede obtener la gráfica representada en la figura 110 el cual nos muestra un mejor rendimiento con AODV, seguido de DSDV y OLSR.

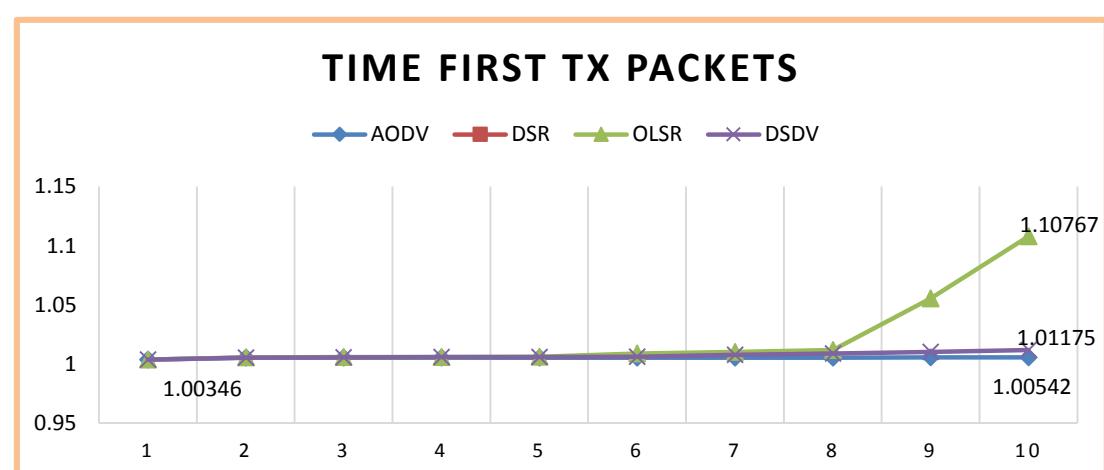


Figura 110: Línea estadística de la métrica Time First TxPackets escenario II.

Para poder medir de mejor manera esta métrica entre los tiempos recogidos en cada monitoreo, se realizó un promedio entre estos resultando con mejor tiempo AODV, seguido de DSDV por muy poco, y finalmente OLSR, lo que se observa en la figura 111.

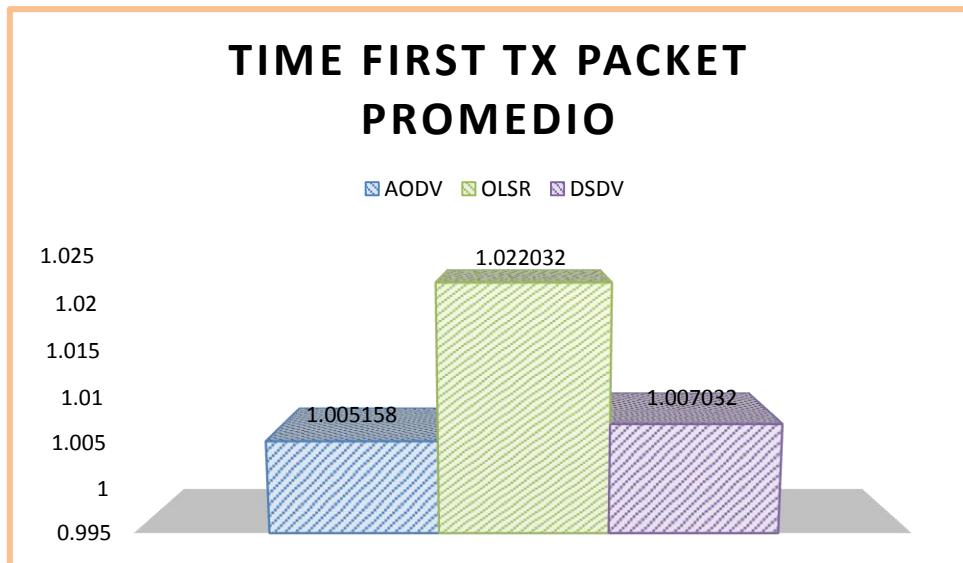


Figura 111: Promedio de la métrica Time First TxPackets escenario II.

3.13.2.2 Time First Rx Packets

Luego de medir el mejor tiempo de envío del primer paquete, ahora se mide el menor tiempo en el que este paquete se recibe por el extremo, estos valores obtenidos se ordenaron en la **TABLA XXIX** para analizarlo de mejor manera.

TABLA XXIX. ANÁLISIS DE LA MÉTRICA TIMEFIRSTRXPACKETS ESCENARIO II

AODV	DSR	OLSR	DSDV
1,00338	N/A	1,00348	1,00348
1,00707	N/A	1,00568	1,06302
1,00545	N/A	1,05348	1,0057
1,00559	N/A	1,02351	1,00613
1,00552	N/A	1,04363	1,3761
1,00536	N/A	1,0091	1,06048
1,01536	N/A	1,01025	1,06067
0,00000	N/A	1,01177	1,06026
1,01557	N/A	1,06046	1,01025
1,0055	N/A	1,12573	1,01182

La gráfica reflejada en la figura 112, está basada en la **TABLA XXIX** en esta se puede apreciar de mejor manera las variaciones que tiene cada protocolo con respecto a la

métrica Time First Rx Packets, notando que AODV, y DSDV pueden sufrir cambios más bruscos en ciertos momentos, mientras que OLSR es más estable en sus valores.

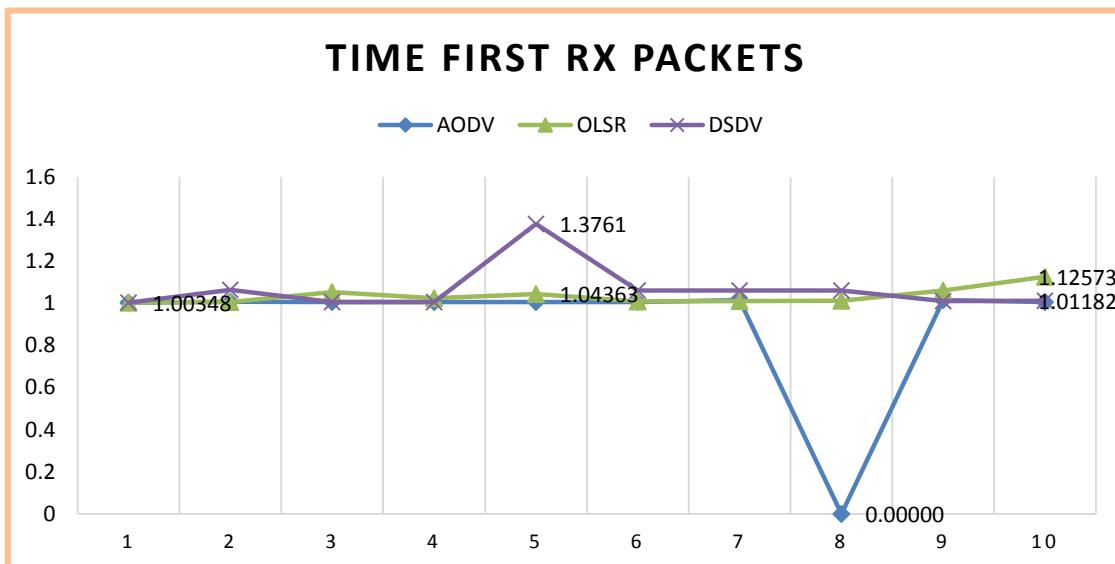


Figura 112: Línea estadista de la métrica Time First RxPackets escenario II.

Guiaido de los valores de la **TABLA XXIX** también se ha calculado un promedio entre estos valores, para manejar un solo valor durante toda la simulación y poder compararlos, quedando según la figura 113, con menor tiempo de entrega del primer paquete AODV seguido de OLSR y finalmente DSDV.

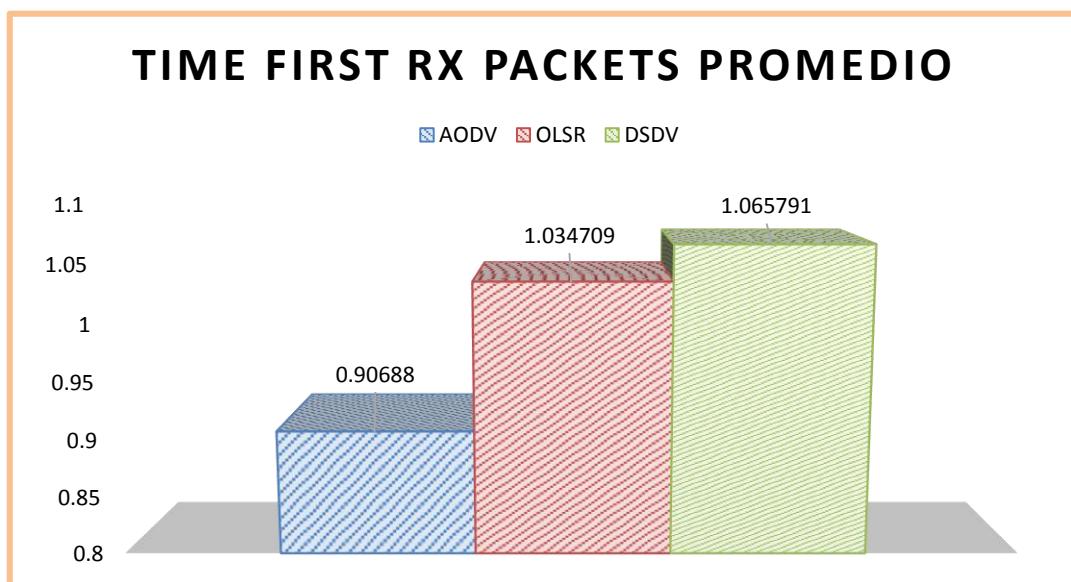


Figura 113: Promedio de la métrica Time First RxPackets escenario II.

3.13.2.3 Time Last Tx Packets

La siguiente métrica a evaluar es el tiempo en el que se envía el último paquete, tomando en cuenta distintos tiempos en la simulación se ha recogido los datos de la **TABLA XXX**, los que nos ayudaran a graficar ciertas tendencias a continuación.

TABLA XXX. ANÁLISIS DE LA MÉTRICA TIMELASTTXPACKETS ESCENARIO II

AODV	DSR	OLSR	DSDV
1,99846	N/A	1,96096	1,99846
1,9978	N/A	1,99798	1,9978
1,10526	N/A	1,99829	1,99798
1,99332	N/A	1,99834	1,99829
1,83082	N/A	1,99847	1,99834
1,50379	N/A	1,99885	1,99834
1,10526	N/A	1,9971	1,99847
1,95374	N/A	1,99925	1,99767
1,93342	N/A	1,9978	1,99885
1,6816	N/A	1,9978	1,99771
1,28117	N/A	1,99767	1,99925

La gráfica obtenida con los datos de la **TABLA XXX**, se aprecian en la figura 114 la que nos muestra los valores de DSDV y OLSR tan cercanos que no se logra apreciar OLSR, solo en su primer tiempo que tiene un mínimo de variación además de tener valores estables, no siendo así AODV ya que refleja grandes cambios en cada instante.

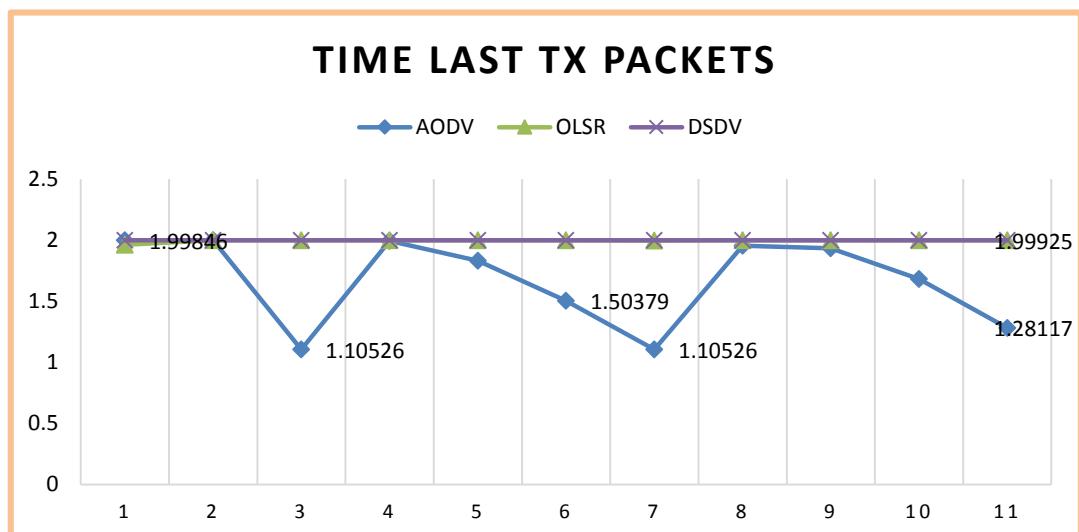


Figura 114: Línea estadística de la métrica Time Last TxPackets escenario II.

Y finalmente graficando el promedio entre los datos obtenidos, para apreciar un valor o una tendencia durante toda la simulación siendo AODV el que tiene menor tiempo en la entrega del último paquete, seguido de OLSR y finalmente DSDV, pudiéndose apreciar en la figura 115.

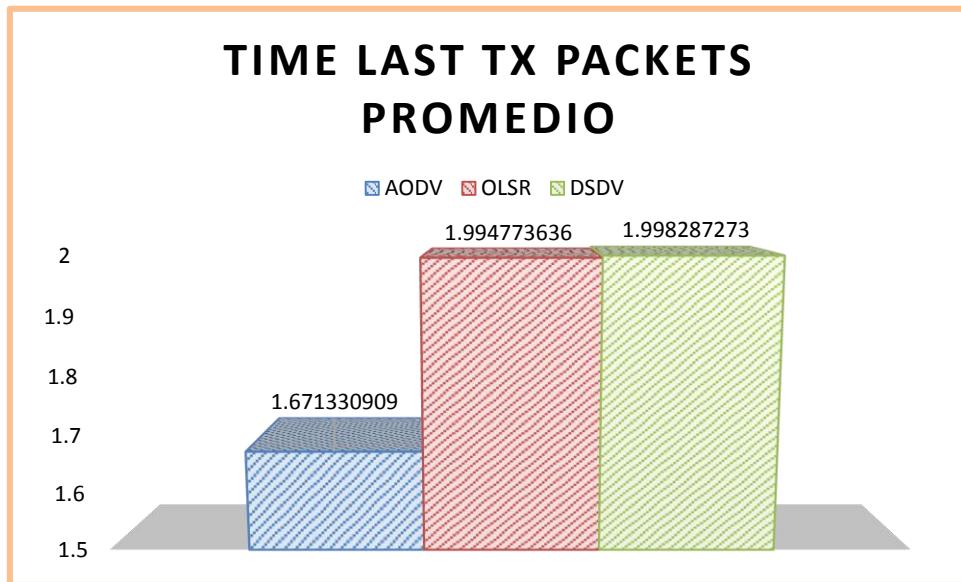


Figura 115: Promedio de la métrica Time Last TxPackets escenario II.

3.13.2.4 Time Last Rx Packets

Se trata de obtener seguidamente el tiempo en el que se recibió el paquete que se envió al final, en la **TABLA XXXI** se observa los datos recogidos en diez monitoreas realizados en la simulación.

TABLA XXXI. ANÁLISIS DE LA MÉTRICA TIMELASTRXPACKETS ESCENARIO II

AODV	DSR	OLSR	DSDV
1,99854	N/A	1,95405	1,71401
1,99788	N/A	1,996	1,89342
1,10539	N/A	1,99851	1,98045
1,99333	N/A	1,99841	1,8611
1,83135	N/A	1,99847	1,96391
1,50412	N/A	1,99886	1,99847
1,01536	N/A	1,99774	1,93518
0,0000	N/A	1,99925	1,73172
1,93344	N/A	1,98306	1,86042
1,59329	N/A	1,99768	1,99925

Luego de esto se puede obtener así mismo los gráficos para analizar de mejor manera estos datos, en la figura 116 se puede ver mayor estabilidad con OLSR, seguido de DSDV y finalmente AODV con más variaciones a lo largo de la simulación.

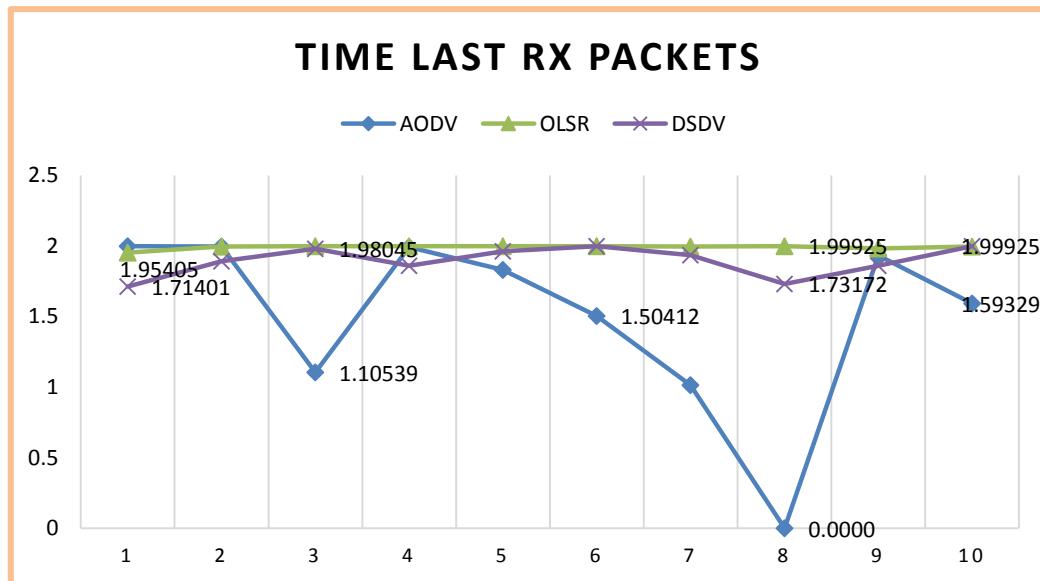


Figura 116: Línea estadística de la métrica Time Last RxPackets escenario II.

El promedio obtenido en la figura 117, nos muestra un solo valor para establecer durante toda la simulación, siendo el protocolo con menor tiempo en la entrega del último paquete AODV, luego DSDV y finalmente OLSR.

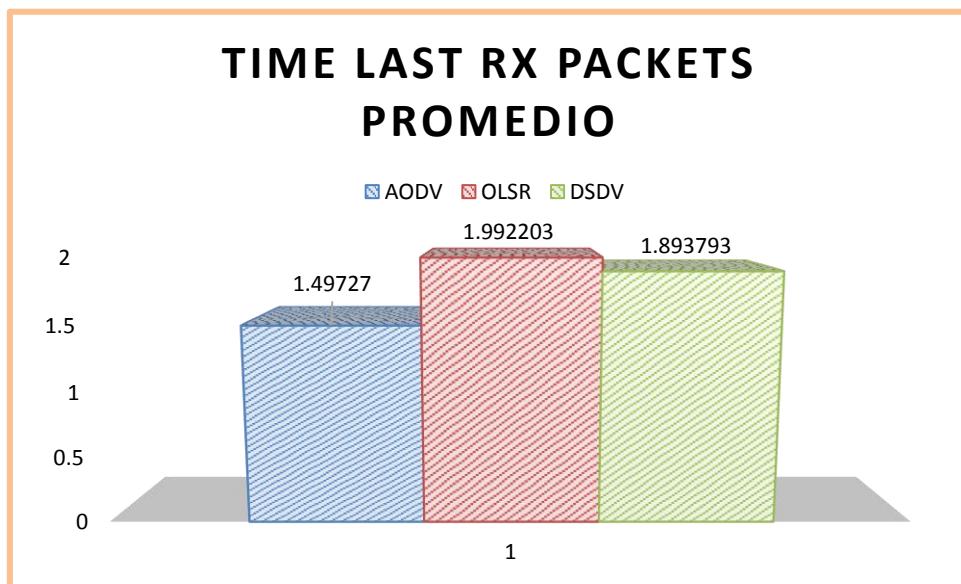


Figura 117: Promedio de la métrica Time Last RxPackets escenario II.

3.13.2.5 Delay

El retraso que resulta en cada envío-recepción de paquetes es conocido como Delay, estos retrasos se establecen en la **TABLA XXXII** para su posterior análisis.

TABLA XXXII. ANÁLISIS DE LA MÉTRICA DELAY ESCENARIO II

AODV	DSR	OLSR	DSDV
1,17986	N/A	3,3738	7,91904
2,1632	N/A	2,57906	2,60206
2,46304	N/A	1,0845	2,10243
1,82327	N/A	3,48179	1,23896
1,35154	N/A	9,79945	2,01861
9,35095	N/A	5,90149	1,49655
1,00291	N/A	1,39946	1,29317
0,0000	N/A	4,61027	1,09096
1,25829	N/A	1,01984	1,4593
2,58626	N/A	2,2205	4,71895

El Delay para este escenario se puede observar en la figura 118, en esta se muestra más estabilidad en los valores del protocolo DSDV, AODV y OLSR con más variaciones.

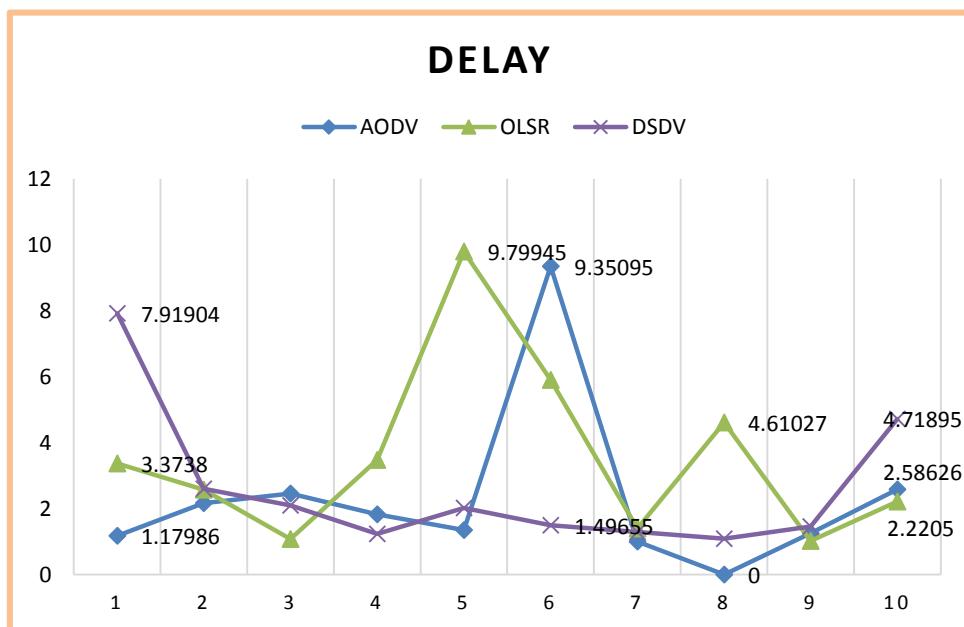


Figura 118: Línea estadística de la métrica Delay escenario II.

Así mismo se puede obtener un retraso promedio, siendo el tiempo promedio de menor retraso el de AODV, seguido de DSDV y finalmente OLSR, como se aprecia en la figura 119.

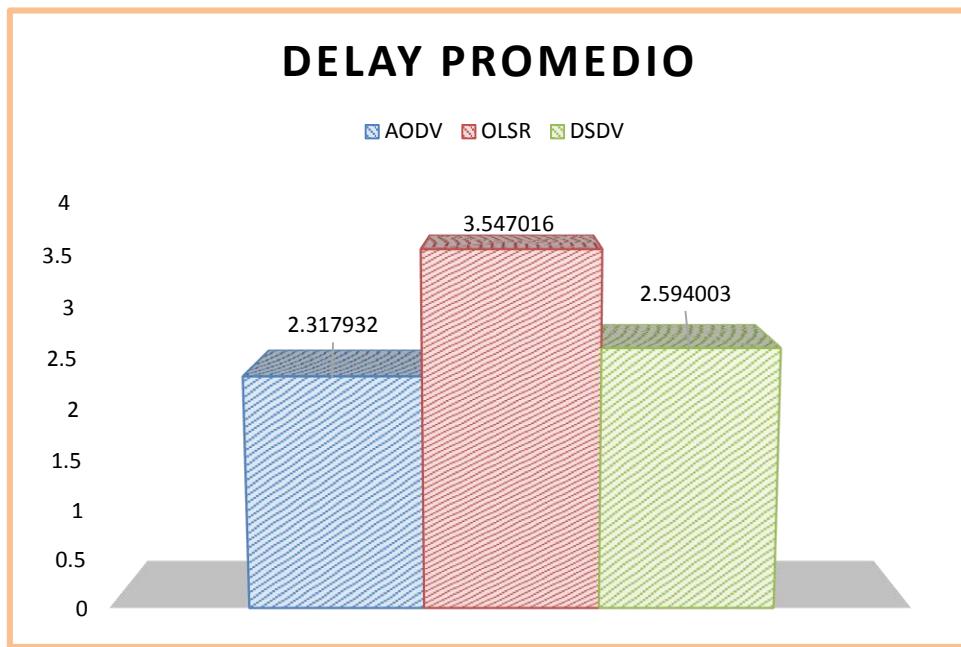


Figura 119: Promedio de la métrica Delay escenario II.

3.13.2.6 Jitter

La variación de este retraso se observa en la **TABLA XXXIII**, lo que es analizado a continuación mediante gráficas, de análisis estadístico.

TABLA XXXIII. ANÁLISIS DE LA MÉTRICA JITTER ESCENARIO II

AODV	DSR	OLSR	DSDV
1,40813	N/A	3,0016	7,36219
2,14201	N/A	1,87225	1,87979
2,2816	N/A	8,80004	1,57491
2,19173	N/A	3,81324	1,08095
1,48245	N/A	8,98731	1,42471
1,45049	N/A	6,71604	1,20213
0,00000	N/A	1,11638	9,45443
0,00000	N/A	4,10327	7,87388
1,27947	N/A	9,29601	1,52225
1,21016	N/A	2,12279	5,80645

Esta variación del retraso se observa en la figura 120, medidas que se pueden obtener de AODV, OLSR y DSDV, siendo imposible el recoger valores para DSR igual que en el escenario de prueba uno.

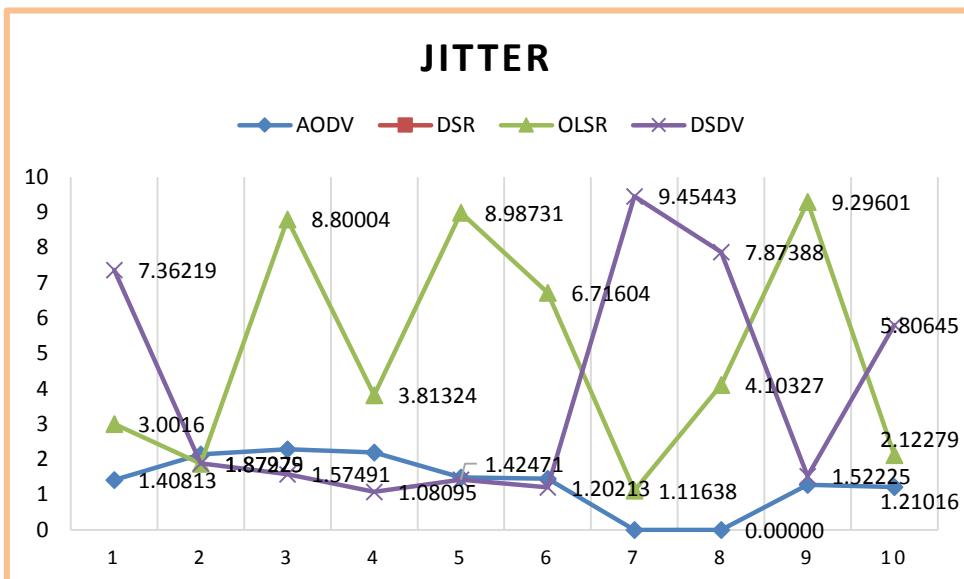


Figura 120: Línea estadística de la métrica Jitter escenario II.

Para obtener una medida global durante todo el tiempo de simulación se puede observar en la figura 121, en esta se establece la menor variación del retraso el de AODV, seguido de DSDV y a continuación OLSR.

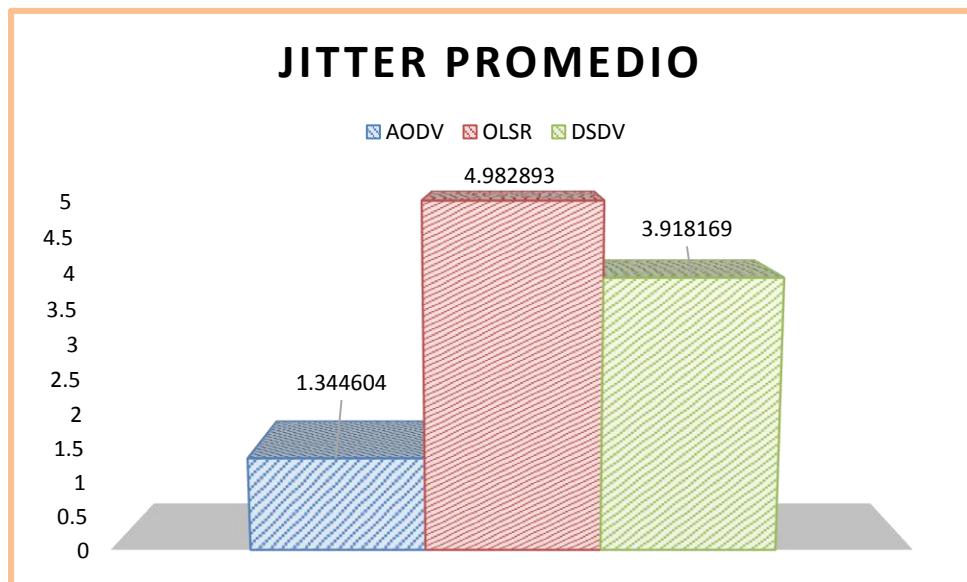


Figura 121: Promedio de la métrica Jitter escenario II.

3.13.2.7 TxBytes

La métrica a continuación es la cantidad de bytes que se transmiten en cada envío, de igual manera se recogieron algunos datos en cada seguimiento con Flowmonitor, estos datos se encuentran en la **TABLA XXXIV**, que sirven para su análisis estadístico. Destacando que esta métrica ya ha sido posible recoger.

TABLA XXXIV. ANÁLISIS DE LA MÉTRICA TX BYTES ESCENARIO II

AODV	DSR	OLSR	DSDV
36708	2192	28980	36708
36616	32456	34776	36616
126	2424	27232	36616
1192	1704	18400	36616
854	49400	36616	36616
986	2068	34776	36616
494	1176	29440	36524
192	600	36432	36524
876	688	32844	36432
486	696	30912	36432

La figura 122 muestra gráficamente la variación en los valores de bytes en cada envío, mostrando estabilidad en los datos que lleva el protocolo DSDV seguido de OLSR, y DSR seguido de AODV con valores más variantes y aproximados a cero.

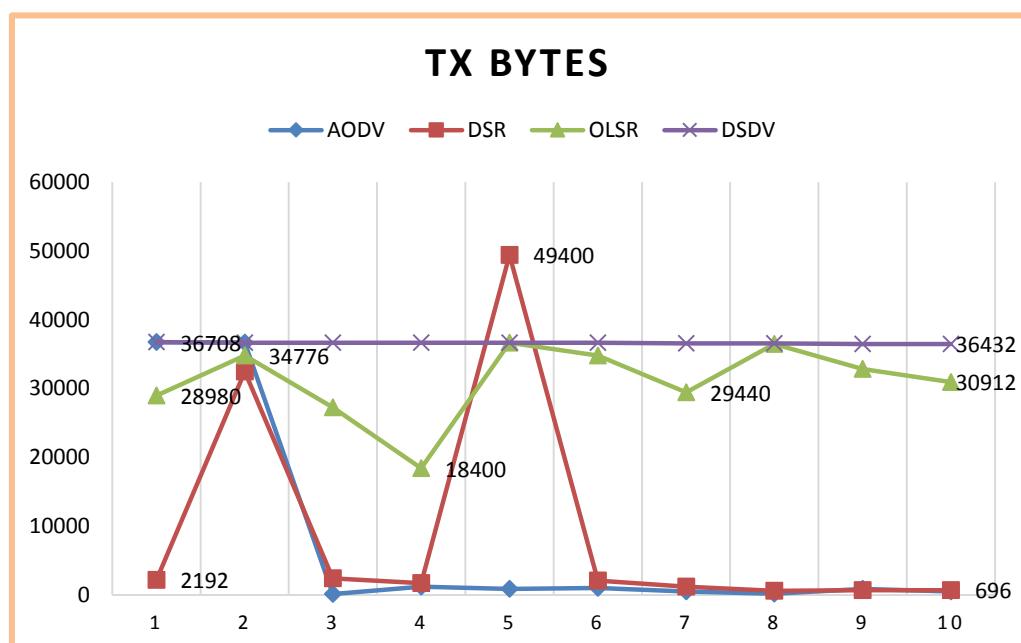


Figura 122: Línea estadística de la métrica TxBytes escenario II.

Los valores promedio obtenidos en esta métrica se reflejan en la figura 123, dejando ver que el protocolo que menor bytes lleva en cada envío es AODV, seguido de DSR, luego OLSR y finalmente DSDV. Pudiendo ser algo a favor o en contra al momento de evaluar en el rendimiento de la red.

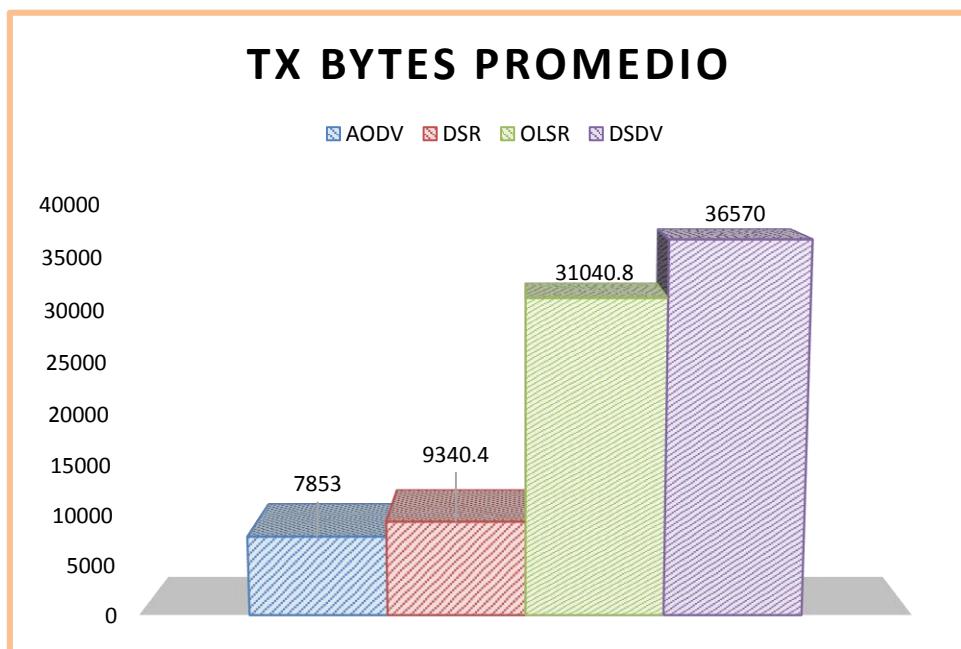


Figura 123: Promedio de la métrica TxBytes escenario II.

3.13.2.8 RxBytes

De la misma manera se debe establecer el valor de la entrega exitosa de bytes, durante diez seguimientos con flowmonitor que están ubicados en la **TABLA XXXV**, y se analizan a continuación.

TABLA XXXV. ANÁLISIS DE LA MÉTRICA RX BYTES ESCENARIO II

AODV	DSR	OLSR	DSDV
20700	2148	20516	13616
15640	0,0	25484	19504
126	2304	16744	14076
1192	1656	9844	9660
1932	12168	32476	12604
1470	1972	27232	29348
48	1176	19320	23276
0,0	600	36156	16100
854	688	21712	10948
834	696	26128	28520

Se observa las variaciones que sufren los protocolos en algunos momentos de la simulación, se puede definir en la figura 124, que tienen mayores variaciones OLSR y DSDV, y menores variaciones y tendencia a cero por parte de AODV y DSR.

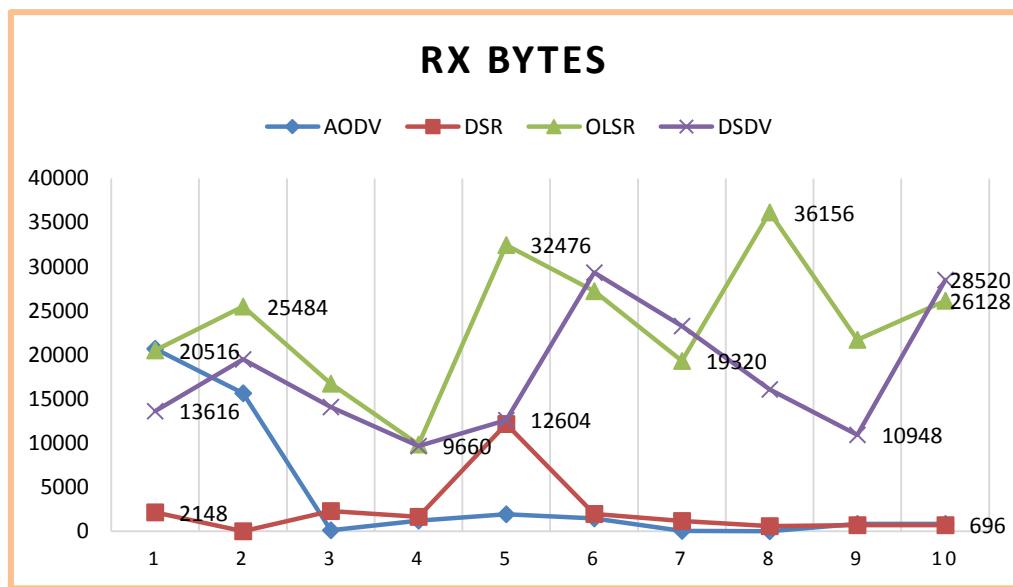


Figura 124: Línea estadística de la métrica RxBytes escenario II.

La figura 125 muestra el promedio durante toda la simulación, de la recepción de mayor número de bytes por envío siendo OLSR el que recepta mayor número de bytes, seguido de DSDV, AODV y finalmente DSR.

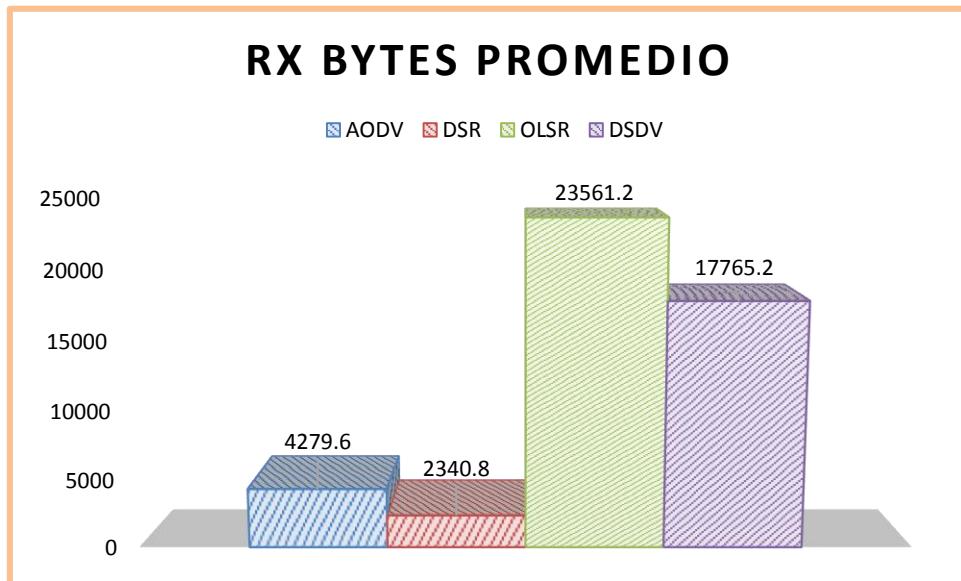


Figura 125: Promedio de la métrica RxBytes escenario II.

3.13.2.9 TxPackets

El número de paquetes enviados en cada momento se establece en la **TABLA XXXVI**, estos valores se analizan a continuación.

TABLA XXXVI. ANÁLISIS DE LA MÉTRICA TX PACKETS ESCENARIO II

AODV	DSR	OLSR	DSDV
399	28	315	399
398	530	378	398
3	39	296	398
25	29	200	398
42	591	398	398
31	36	378	398
2	18	320	397
21	10	396	397
19	11	357	396
22	12	336	396

Así mismo se puede obtener gráficas estadísticas de líneas que pueden mostrar la estabilidad o variabilidad del envío de paquetes dependiendo del protocolo, en la figura 126 se observa mayor estabilidad en los datos recogidos del protocolo DSDV y OLSR, mientras que AODV y DSR sufren mayores variaciones y tendencias a cero.

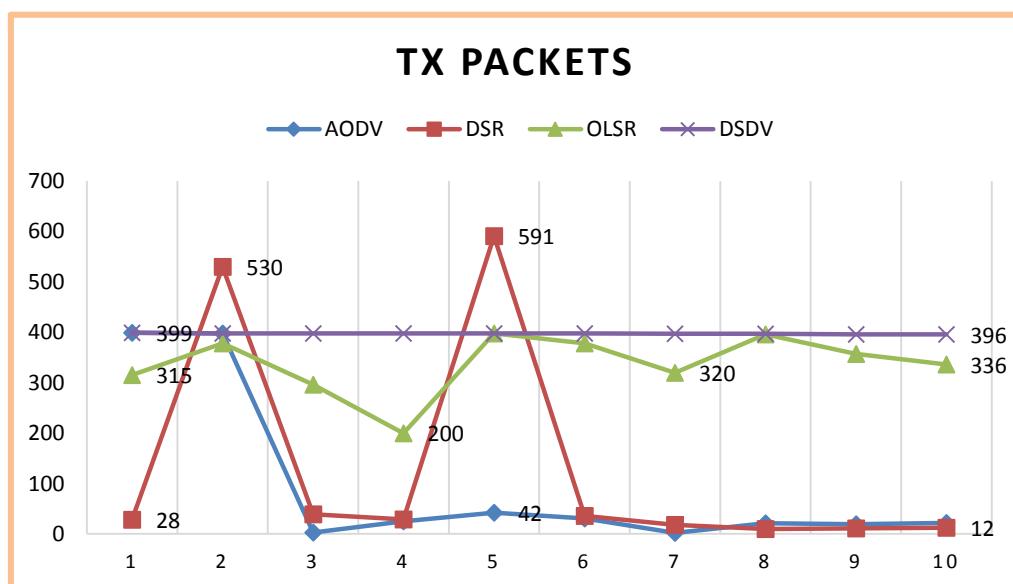


Figura 126: Línea estadística de la métrica TxPackets escenario II.

El promedio de paquetes enviados por cada protocolo se establece en la figura 127, esta nos ayuda a definir que el protocolo con mayor número de paquetes enviados es DSDV, seguido de OLSR, luego DSR y finalmente AODV.

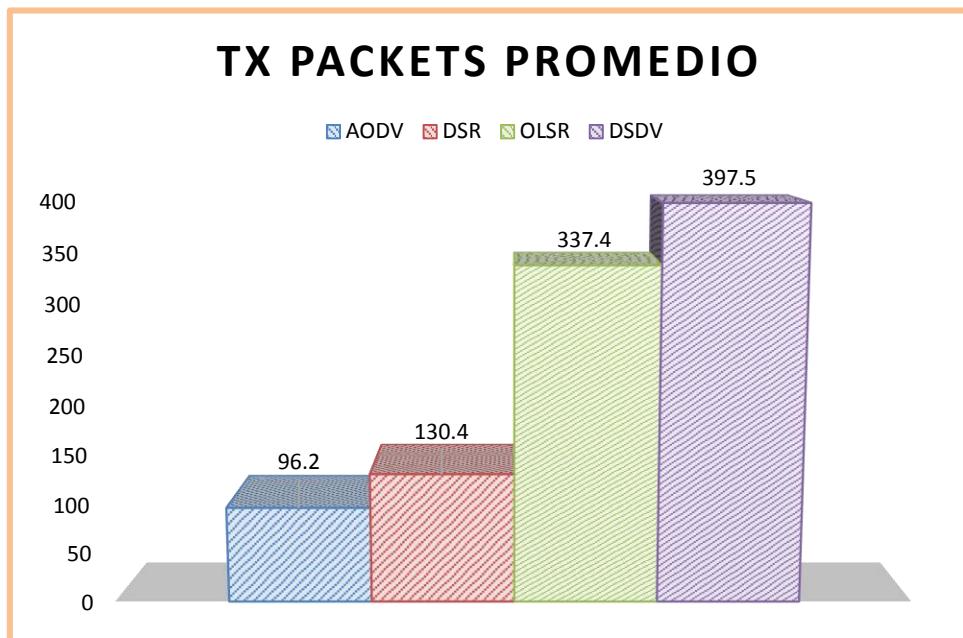


Figura 127: Promedio de la métrica TxPackets escenario II.

3.13.2.10 RxPackets

Ahora de esos paquetes enviados se debe analizar cuantos llegan a su destino, este análisis también se obtiene mediante recogida de datos con flowmonitor, estos datos se establecen en la **TABLA XXXVII**.

TABLA XXXVII. ANÁLISIS DE LA MÉTRICA RX PACKETS

AODV	DSR	OLSR	DSDV
225	27	223	148
170	0	277	212
3	37	182	153
25	28	107	105
42	199	353	137
31	34	296	319
1	18	210	253
0	10	393	175
19	11	236	119
19	12	284	310

De la misma manera se observa en la figura 128 las variaciones de cada protocolo con respecto a esta métrica, siendo más variantes durante la simulación DSDV y OLSR, mientras que se puede observar una mayor estabilidad con AODV y DSR.

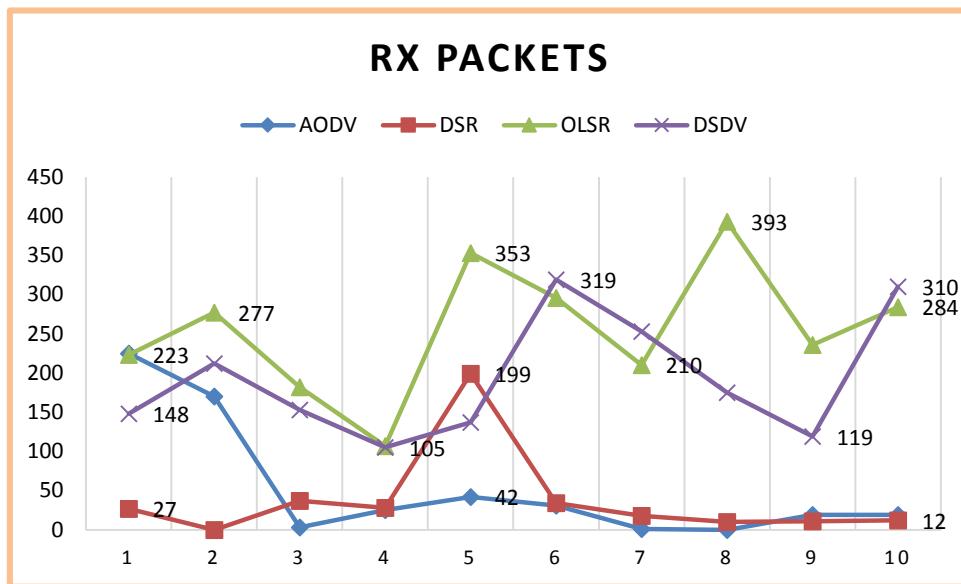


Figura 128: Línea estadística de la métrica RxPackets escenario II.

El número de paquetes recibidos por el destino en promedio por cada protocolo, se observa en la figura 129, la que nos ofrece los datos del protocolo con mayor número de paquetes recibidos por envío, en primer lugar OLSR, seguido de DSDV, luego AODV y finalmente DSR.

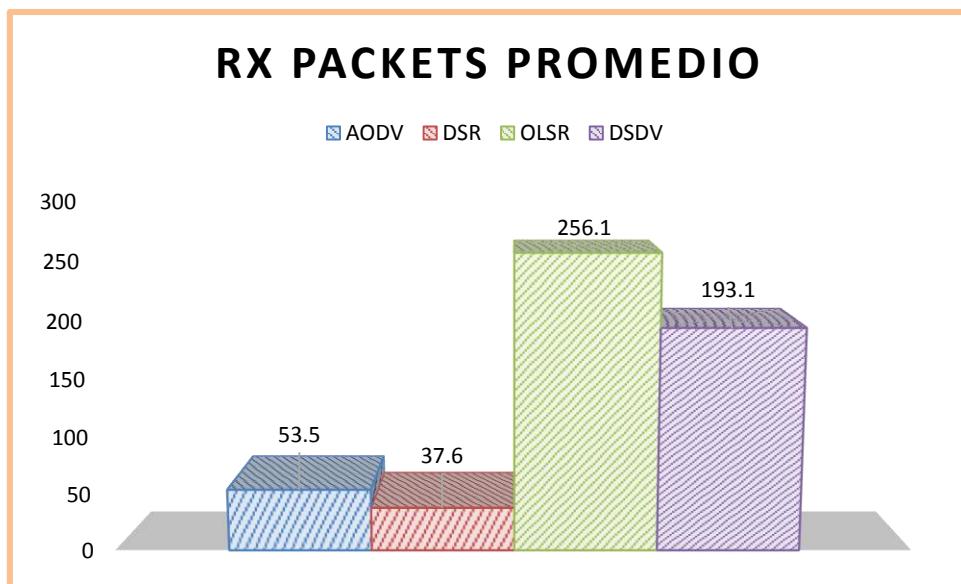


Figura 129: Promedio de la métrica RxPackets escenario II.

3.13.2.11 Lost Packets

El número de paquetes perdidos es sumamente importante, durante el tiempo de simulación se ha recogido varios valores para su posterior análisis, estos datos reflejados en la **TABLA XXXIII**.

TABLA XXXIII. ANÁLISIS DE LA MÉTRICA LOST PACKETS ESCENARIO II

AODV	DSR	OLSR	DSDV
171	1	89	211
208	530	93	146
0	2	107	220
0	1	78	253
0	392	45	239
0	2	78	79
1	0	97	117
16	0	3	182
0	0	88	237
3	0	47	86

La línea estadística mostrada en la figura 130 expresa, en sus límites máximos valores altos con respecto a número de paquetes perdidos a DSR, mientras que DSDV es un poco menos propenso a la perdida de paquetes seguido de OLSR y finalmente con menor número de paquetes perdidos a AODV.

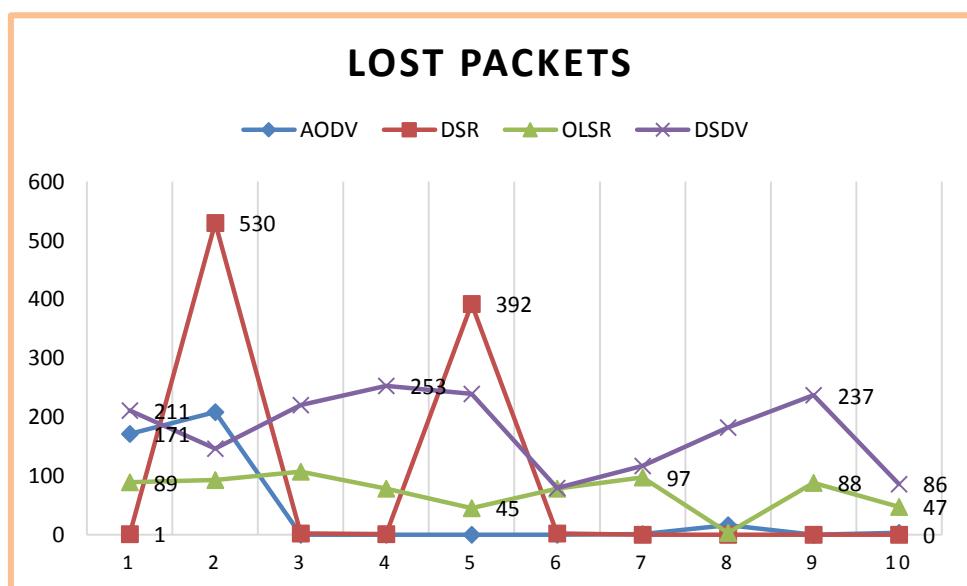


Figura 130: Línea estadística de la métrica LostPackets escenario II.

Así mismo se maneja un promedio para poder reflejar un valor global durante la simulación, obteniendo el protocolo con menor número de paquetes perdidos siendo este AODV, seguido de OLSR, luego DSR y finalmente DSDV con mayor número de paquetes perdidos, tal como se observa en la figura 131.

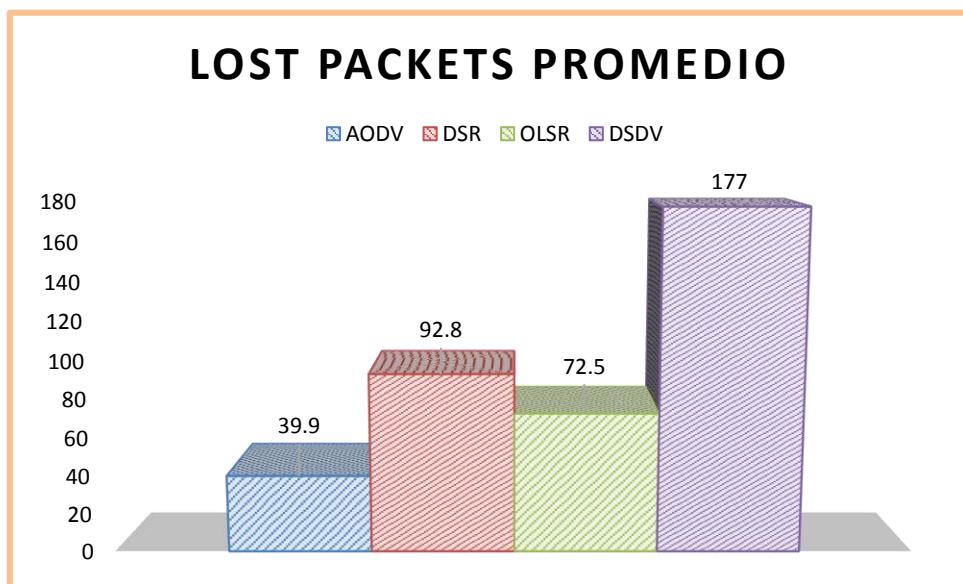


Figura 131: Promedio de la métrica LostPackets escenario II.

3.13.2.12 Cálculo de Troughput

El cálculo del Troughput, es decir el éxito de entrega de mensajes utilizando un medio resaltando en este caso que el medio es lógico, y que los mensajes que se envían son de tipo broadcast tanto en el primer escenario, como en este escenario de prueba dos, este cálculo obtenido mediante formula se refleja en la **TABLA XXXIX**.

TABLA XXXIX. CÁLCULO DE TROUGHPUT ESCENARIO II

AODV	DSR	OLSR	DSDV
21,398	11,704	117,806	88,826

El throughput calculado y observado en la **TABLA XXXIX**, se grafica estadísticamente en barras y se aprecia en la figura 132, la que nos muestra a DSR, seguido de AODV, luego DSDV y finalmente OLSR de menor a mayor éxito en la tasa de envío de mensajes sobre el canal.

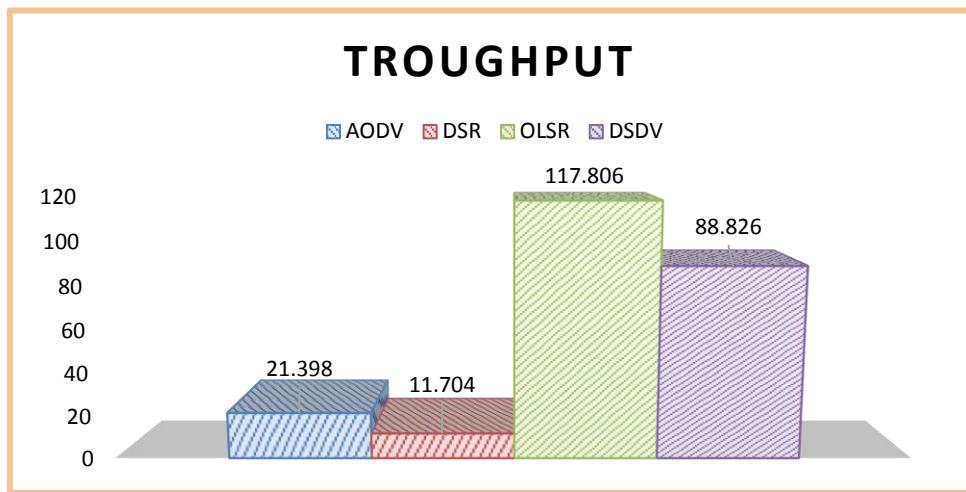


Figura 132: Cálculo de Troughput escenario II.

3.13.2.13 Cálculo de Tasa de Entrega

Además se calcula otra métrica importante que es la tasa de entrega durante la simulación, estos valores obtenidos se establecen en la **TABLA XXXX**, observando mediante valores el rendimiento de cada protocolo.

TABLA XXXX. CÁLCULO DE TASA DE ENTREGA ESCENARIO II

AODV	DSR	OLSR	DSDV
55,613	28,834	75,904	48,579

En la figura 133 se logra apreciar de mejor manera los valores obtenidos en cada protocolo, siendo el que mayor tasa de entrega ofrece OLSR, seguido de AODV, DSDV y finalmente DSR.

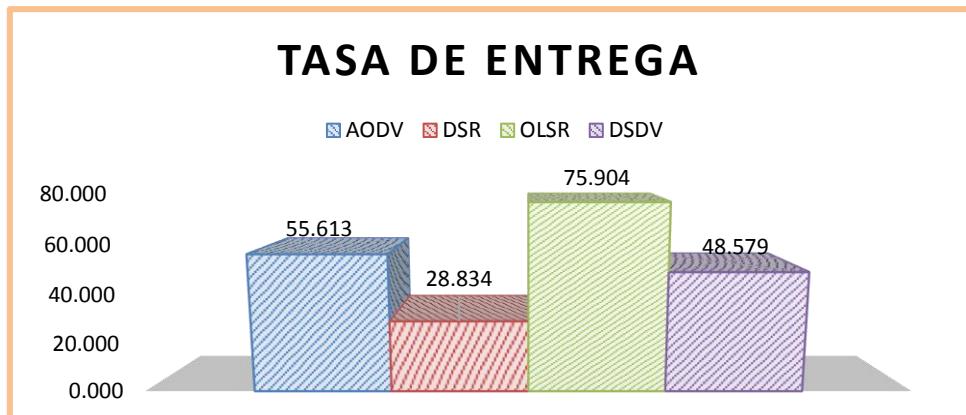


Figura 133: Cálculo de Tasa de entrega escenario II.

3.13.1 Propuesta del mejor protocolo para el primer escenario

En la **TABLA XXXXI** se puede observar una comparación de resultados obtenidos para las diferentes métricas en este tipo de escenario, con menor número de nodos Ad-Hoc, en este caso 25 se puede observar que mediante el número de métricas recolectadas ***TimeFirstTxPackets***, ***TimeFirstRxPackets***, ***TimeLastTxPackets*** y ***TimeLastRxPackets*** el mejor protocolo a utilizar es **AODV** ya que es preciso en tiempos de envío- entrega, y con respecto a ***Delay*** y ***Jitter*** es el que menor retraso y variaciones del mismo sufre, el número de bytes y de paquetes que envía y entrega ***TxBytes***, ***RxBytes***, ***TxPackets***, y ***RxPackets*** puede ser algo a favor o en contra según como lo veamos, ya que en este caso es el protocolo que lleva en cada envío la menor cantidad en comparación con los demás, pero se puede resaltar que es el que menor número de paquetes pierde durante la simulación, lo que se puede interpretar como la división de paquetes de forma equitativa de tal manera de llevar de a poco la cantidad total en menor número de tiempo y con menor perdida de paquetes, y con respecto a las dos métricas principales ***Troughput*** y ***Tasa de entrega***, que son predominantes a la hora de elegir se puede apreciar que **AODV** tiene el mayor envío de mensajes sobre el canal lo que es un factor determinante en la ayuda para el problema de nodos ocultos ya que siempre está actualizando y descubriendo nuevas rutas para el envío de paquetes, pero puede desembocar en un sobrecarga de mensajes en la red, y finalmente la tasa de entrega que resalta entre los demás al ser el que mayor éxito tiene.

TABLA XXXXI. COMPARACIÓN DE MÉTRICAS ENTRE PROTOCOLOS
ESCENARIO DE PRUEBA UNO.

METRICAS	PROTOCOLOS			
	AODV	DSR	OLSR	DSDV
TimeFirsTxPackets	1º en enviar primer paquete	Sin datos	3º en enviar primer paquete	2º en enviar primer paquete
TimeFirstRxPacket	1º en recibir primer paquete	Sin datos	2º en recibir primer paquete	3º en recibir primer paquete
TimeLastTxPackets	1º enviar último paquete	Sin datos	2º en enviar último paquete	3º en enviar último paquete
TimeLastRxPackets	1º en recibir último paquete	Sin datos	3º en recibir último paquete	2º en recibir último paquete
Delay	1º con menor retardo	4º con menor retardo	2º con menor retardo	3º con menor retardo
Jitter	1º con menor variación del retardo	4º con menor variación del retardo	2º con menor variación del retardo	3º con menor variación del retardo

TxBytes	4º en cantidad de bytes enviados	1º en cantidad de bytes enviados	3º en cantidad de bytes enviados	2º en cantidad de bytes enviados
RxBytes	4º en cantidad de bytes recibidos	3º en cantidad de bytes recibidos	1º en cantidad de bytes recibidos	2º en cantidad de bytes recibidos
TxPackets	4º en cantidad de paquetes enviados	2º en cantidad de paquetes enviados	3º en cantidad de paquetes enviados	1º en cantidad de paquetes enviados
RxPackets	4º en cantidad de paquetes recibidos	2º en cantidad de paquetes recibidos	3º en cantidad de paquetes recibidos	1º en cantidad de paquetes recibidos
LostPackets	1º en menor cantidad de paquetes perdidos	4º en menor cantidad de paquetes perdidos	2º en menor cantidad de paquetes perdidos	3º en menor cantidad de paquetes perdidos
Troughput	4º en uso de mensajes sobre el canal	3º en uso de mensajes sobre el canal	1º en uso de mensajes sobre el canal	2º en uso de mensajes sobre el canal
Tasa de entrega	1º en mayor tasa de entrega	3º en mayor tasa de entrega	4º en mayor tasa de entrega	2º mayor en tasa de entrega

3.13.2 Propuesta del mejor protocolo para el segundo escenario

En la **TABLA XXXII** se puede observar una comparación de resultados obtenidos para el escenario dos que consta del doble en número de nodos Ad-Hoc, en este caso 50 se obtuvo que mediante el número de métricas recolectadas ***TimeFirstTxPackets***, ***TimeFirstRxPackets***, ***TimeLastTxPackets*** y ***TimeLastRxPackets*** el mejor protocolo a utilizar es **AODV** que emplea menores tiempos de envío- entrega, lo que puede deberse a la sectorización que maneja y con respecto a ***Delay*** y ***Jitter*** también es este escenario es el que menor retraso y variaciones del mismo sufre, el número de bytes y de paquetes que envía y entrega ***TxBytes***, ***RxBytes***, ***TxPackets***, y ***RxPackets*** juega a favor en este caso mejorando tiempos, ya que en este caso es el protocolo que lleva en cada envío la menor cantidad en comparación con los demás, pero se puede resaltar que es el que menor número de paquetes pierde durante la simulación, lo que se puede interpretar como la división de paquetes de forma equitativa de tal manera de llevar de a poco la cantidad total en menor número de tiempo y con menor perdida de paquetes, y con respecto a las dos métricas principales ***Troughput*** y ***Tasa de entrega***, que son predominantes a la hora de elegir se puede apreciar que **AODV** tiene el uno de los mayores envío de mensajes sobre el canal lo que es un factor determinante en la ayuda para el problema de nodos ocultos ya que siempre está actualizando y descubriendo

nuevas rutas para el envío de paquetes, pero puede desembocar en un sobrecarga de mensajes en la red, además se puede apreciar también dentro de esta métrica que la sectorización que hace en el envío de mensajes broadcast le ayuda a mejorar rutas y tiempos, ya que se hace pequeños submapas de toda la red con el fin de tener precisión y realizar más rápidamente el enrutamiento y finalmente la tasa de entrega que es buena entre los demás al ser uno de los que mayor éxito tiene, a excepción de OLSR.

TABLA XXXII. COMPARACIÓN DE MÉTRICAS ENTRE PROTOCOLOS
ESCENARIO DE PRUEBA DOS.

METRICAS	PROTOCOLOS			
	AODV	DSR	OLSR	DSDV
TimeFirstTxPackets	1º en envío del primer paquete	Sin datos	3º en envío del primer paquete	2º en envío del primer paquete
TimeFirstRxPacket	1º en recibir el primer paquete	Sin datos	2º en recibir el primer paquete	3º en recibir el primer paquete
TimeLastTxPackets	1º en enviar el último paquete	Sin datos	2º en enviar el último paquete	3º en enviar el último paquete
TimeLastRxPackets	1º en recibir el último paquete	Sin datos	3º en recibir el último paquete	2º en recibir el último paquete
Delay	1º con menor retardo	4º con menor retardo	3º con menor retardo	2º con menor retardo
Jitter	1º con menor variación del retardo	4º con menor variación del retardo	3º con menor variación del retardo	2º con menor variación del retardo
TxBytes	4º en número de bytes por envío	3º en número de bytes por envío	2º en número de bytes por envío	1º en número de bytes por envío
RxBytes	4º en número de bytes recibidos	3º en número de bytes recibidos	1º en número de bytes recibidos	2º en número de bytes recibidos
TxPackets	4º en número de paquetes enviados	3º en número de paquetes enviados	2º en número de paquetes enviados	1º en número de paquetes enviados
RxPackets	3º en número de paquetes recibidos	4º en número de paquetes recibidos	1º en número de paquetes recibidos	2º en número de paquetes recibidos
LostPackets	1º en menor número de paquetes perdidos	3º en menor número de paquetes perdidos	2º en menor número de paquetes perdidos	4º en menor número de paquetes perdidos
Troughput	3º en uso de mensajes sobre el canal	4º en uso de mensajes sobre el canal	1º en uso de mensajes sobre el canal	2º en uso de mensajes sobre el canal
Tasa de entrega	2º en mayor tasa de entrega	4º en mayor tasa de entrega	1º en mayor tasa de entrega	3º en mayor tasa de entrega

g. Discusión

El presente proyecto de titulación, muestra a la comunidad universitaria y comunidad en general las pautas obtenidas mediante la experimentación en un medio simulado, para lograr la estabilidad en redes que no se basan en infraestructura predefinida, en este caso las redes Ad-Hoc, así mismo mediante métricas obtiene el mejor rendimiento de la red de acuerdo a un escenario definido, lo que sienta las bases para el estudio de las redes de acuerdo a su rendimiento, logrando mejores tiempos de envío-recepción, menor perdida de paquetes, menor retraso y variación en el mismo, mayor éxito en la actualización de rutas y tasa de entrega. Todo lo que se puede lograr haciendo uso de los protocolos adecuados.

1. Desarrollo de la Propuesta Alternativa

Objetivo 1: Analizar las características y funcionamiento de las redes Ad-Hoc, y sus protocolos proactivos y reactivos.

Para su cumplimiento se llevó a cabo la recolección de información en fuentes confiables, otorgando respuestas a muchas dudas sobre este tipo de redes, su funcionamiento, protocolos, arquitectura, características (Revisión de Literatura), así como realizar una vista rápida a su crecimiento y uso (Sección 1.1 de la Fase 1 Análisis de las Redes Ad-Hoc), principales problemas (Sección 1.2 de la Fase 1 Análisis de las Redes Ad-Hoc), entender el desenvolvimiento de la red, así como de cada protocolo, y de su aplicación en la herramienta().

Objetivo 2: Realizar el modelo de la topología de red, y simular distintos escenarios, utilizando los protocolos.

Este objetivo se cumplió, realizando la especificación de cada escenario o de cada topología (Sección 2.1 de la Fase 2 Modelado de la Topología), el funcionamiento de cada protocolo (Sección 2.2 de la Fase 2 Modelado de la Topología), la justificación de porque se utilizó Ns-3 que es una de las mejores herramientas para la simulación de redes basadas en el estándar 802.11 (Sección 2.3 de la Fase 2 Modelado de la Topología), la manera de aplicarlo mediante código escrito en el lenguaje C++, en la herramienta Ns-3 (Sección 3.8 de la Fase 3 Simulación y Evaluación), Además este objetivo contempla la instalación y configuración de las herramientas a utilizar(Sección 3.10 de la Fase 3 Simulación y Evaluación).

Objetivo 3: Evaluar el desempeño y proponer el protocolo más adecuado para cada escenario, orientado a la estabilización de las redes Ad-Hoc.

Finalmente la manera de alcanzar este objetivo, luego de la fase de simulación se pudo establecer algunas herramientas para la recogida de datos (Sección 3.12 de la Fase 3 Simulación y Evaluación), las que nos entregaron algunas métricas (Sección 3.13 de la Fase 3 Simulación y Evaluación), que después de su análisis tanto en líneas estadísticas, diagramas de barras y promedios para poder concluir valores globales (Escenario de Prueba Uno y Escenario de Prueba Dos de la Fase 3 Simulación Y Evaluación), se obtuvo una tabla general de comparación de métricas(Sección 3.13.1-3.13.2 de la Fase 3 Simulación y Evaluación), por cada escenario haciendo una explicación final de porque se escogió el protocolo en mención, y como ayuda este a solucionar los problemas más comunes de las redes Ad-Hoc y por ende a la estabilización de la red en general.

2. Valoración Técnica Económica Ambiental

El desarrollo del presente proyecto de titulación, exigió una inversión económica, con la que se logró el resultado final, que fue alcanzar los objetivos planteados, evidenciando en cada fase los resultados obtenidos, en la **TABLA XXXIII** se detalla la inversión económica del proyecto:

TABLA XXXIII. VALORACIÓN ECONÓMICA

PRESUPUESTO				
RECURSOS HUMANOS				
Descripción	Cantidad	Núm. de Horas	Valor Unitario	Valor Total
Gabriela Rodríguez	1	789	10	7890,00
Director de Tesis	1	100	0	0,00

Fotocopias	500	—	0,02	10,00
Anillados	3	—	2,00	6,00
CD's	3	—	1,00	3,00
SUBTOTAL				49,50
SERVICIOS BÁSICOS				
Transporte	----	—	0,30	100,00
Comunicación	45	—	1,00	45,00
Internet	----	---	0,00	00,00
SUBTOTAL				145,00
RECURSOS TÉCNICOS TECNOLÓGICOS				
HARDWARE				
Portátil Satellite	1	789	1040,00	1040,00
Flash Memory	1	—	10,00	10,00
Impresora	1	—	180,00	180,00
SUBTOTAL:				1230,00
SOFTWARE				
Xubuntu 14.04.02	1	789	0,00	0 , 00
Ns-3.23	1	700	0,00	0 , 00
Gantt Project	1	30	0,00	0 , 00
Libre Office	1	789	0,00	0 , 00
Wireshark 1.10.6	1	300	0,00	0 , 00
Netanim 3.106	1	300	0,00	0 , 00
SUBTOTAL				0 , 00
TOTAL				9.314,50

En proyecto en mención se desarrolló, utilizando herramientas de software con licencia libre, reduciendo notablemente el costo en recurso materiales, en su totalidad el costo fue asumido por la autora del proyecto de titulación, excepto el costo de recurso humano del director del proyecto, ya que este valor es adjudicado por la Universidad.

h. Conclusiones

En base a las experimentaciones realizadas durante el proyecto de titulación, se puede establecer algunas conclusiones, que servirán para las personas interesadas en el tema.

- Las Redes Ad-Hoc trabajan según el estándar 802.11 definido para redes inalámbricas, por lo que cualquier tipo de red inalámbrica sin infraestructura, puede ser tratada como una red Ad-Hoc, pudiendo diferenciarse con respecto al escenario, por las variaciones en el número de nodos y el protocolo que se utilice.
- El protocolo que a nivel general, dio como resultado mejores métricas y manejo de manera eficiente los problemas de sobrecarga y latencia es AODV, elegido en ambos escenarios como la mejor opción, pero cabe resaltar que a pesar de ser líder en ambos escenarios, va perdiendo características mientras se aumenta el número de nodos, y lo que ocurre de manera inversa con DSR, que podría ser otra opción elegida para el segundo escenario.
- La herramienta de simulación Ns-3, es apta para la investigación de redes inalámbricas, con el uso de artículos relacionados, se pudo verificar que es ampliamente flexible en cuanto a su arquitectura, y a su vez siendo Open Source, permite la contribución de terceros en nuevos modelos, con lo que se experimenta en un ambiente de continuo crecimiento, que permite la investigación de modelos de movilidad inalámbricos y protocolos de ruteo.
- Los módulos de Ns-3, toman en cuenta los protocolos manejados en los escenarios (AODV, OLSR, DSR, DSDV), y para los demás protocolos proactivos y reactivos, aún no existe módulos desarrollados, y para protocolos híbridos no tiene aplicación. Los protocolos que tienen soporte actualmente AODV, OLSR Y DSDV se manejan con facilidad, mientras que DSR tiene que implementarse de manera diferente, ya que Ns-3 no maneja directamente este protocolo como los demás.
- La comparativa de métricas de rendimiento, realizada entre protocolos proactivos y reactivos, dieron la pauta para resaltar que los protocolos reactivos, son mejores que los proactivos, y en el caso de envío de mensajes broadcast los proactivos inundan la red periódicamente, lo que causa problemas de sobrecarga o latencia y perdida de paquetes, que desemboca directamente en su inestabilidad.

- En el caso de los protocolos Híbridos, en teoría, al manejar características proactivas y reactivas, hacen que el enrutamiento posea mejores características, lo que se puede concluir teóricamente y que al momento de ponerlo en práctica y comparar los tres tipos (Proactivos, Reactivos e Híbridos) se podría determinar si son la mejor opción ya que trabajan con enrutamiento reactivo por sectores y enrutamiento proactivo al momento de unir los diferentes sectores como es el caso de ZRP, lo que elimina el problema de latencia y sobrecarga.
- El integrar a la simulación realizada por Ns-3, herramientas de monitoreo y análisis de paquetes, optimiza el uso de tiempo a la hora de analizar el comportamiento de varios escenarios, esto gracias a que las herramientas presentan las métricas de manera resumida y lista para la interpretación.

i. Recomendaciones

A continuación se detallan algunas recomendaciones para el mejor desarrollo del proyecto:

- Antes de realizar la simulación, se debe tomar en cuenta las métricas propias de cada protocolo, ya que las métricas obtenidas varían de protocolo a protocolo, a excepción de las métricas como Troughput y Tasa de entrega, que son aplicables a nivel general.
- Durante el proceso de simulación se debe hacer uso de registro de notas mediante mensajes Tipo LOG, que sirven de ayuda al momento de ejecutar la simulación, con los que se podrá apreciar de manera precisa y acertada, los errores y la manipulación de datos en la ejecución del código.
- Para la simulación de un escenario, con más de 60 nodos, disponer de altas capacidades de procesamiento, en cuanto a memoria Ram y velocidad del procesador, en el equipo utilizado para la simulación, la herramienta NetAnim ocupó todo el recurso de memoria Ram (12Gb), sin ofrecer resultados.
- En el análisis de las métricas obtenidas de la simulación, utilizar, además de herramientas para animación, monitoreo y captura de datos, una herramienta para gráfica de estadísticas, lo que ahorra tiempo en la interpretación de datos.
- Antes de empezar con la experimentación en Ns-3, tomar en cuenta la documentación, la API y manuales; brindados por el sitio oficial de Ns-3, para de esta manera adoptar y trabajar adecuadamente, durante la integración de los diferentes métodos que se emplean a lo largo de la simulación.
- Para la comparación de métricas entre escenarios, identificar adecuadamente las métricas y su significado (valoración y peso), para llegar a una conclusión e interpretación acertada de los datos obtenidos en la simulación.

j. Bibliografía

- [1] F. R. M. Solera, "*Introducción a las Redes Manet*", 2014. [En línea]. Disponible en: http://repositorio.cedia.org.ec/bitstream/123456789/960/1/T1_Introduction_MANET_vf.pdf.
- [2] J. C. Pallares, "*Redes Ad Hoc entre vehículos*", 2010. [En línea]. Disponible en: <http://upcommons.upc.edu/pfc/handle/2099.1/4690>.
- [3] V. Zuñiga, "*Redes De Transmisión De Datos*", 2005. [En línea]. Disponible en: http://www.uaeh.edu.mx/docencia/Tesis/icbi/licenciatura/documentos/redes_de_transmision_de_datos.pdf.
- [4] J. C. Miranda y F. G. Montenegro, "*Banco de pruebas para redes ad-hoc*", *Potencia*, 2010. [En línea]. Disponible en: <http://upcommons.upc.edu/pfc/handle/2099.1/13796>.
- [5] J. P. Aspas, "*Redes ad-hoc, el próximo reto*", *Wireless Networks*, 2004. [En línea]. Disponible en: <http://upcommons.upc.edu/revistes/bitstream/2099/9919/1/Article006.pdf>.
- [6] A. Mercado, R. Berrios F., y P. Chan Ye, "*Redes inalámbricas ad hoc*", 2003. [En línea]. Disponible en: <http://facultad.bayamon.inter.edu/cgonzalezr/ELEN4618/Adhoc.pdf>.
- [7] E. M. Reyes, "*Prestaciones De Las Redes Ad Hoc Inalambricas: Teoría a través de Capas*", *Time*, 2009. [En línea]. Disponible en: https://ecienzia.urjc.es/bitstream/handle/10115/2425/TesisDoctoral_EduardoMorgado.pdf?sequence=1.
- [8] P. Roosevelt, "*Redes móviles ad hoc*" [En línea]. Disponible en: <http://es.scribd.com/doc/246766943/Manet#scribd>.
- [9] D. R. Cañas, "*Protocolo de encaminamiento Aco Híbrido para Redes Móviles Ad Hoc*", 2009. [En línea]. Disponible en: <http://eprints.ucm.es/9890/1/MasterDelfin09.pdf>.

- [10] M. Ángel, T. Diosdado, P. Marta, y L. Fernández, "Sistema de Autoreconfiguración para Redes Ad Hoc", 2005. [En línea]. Disponible en: http://eprints.ucm.es/9004/1/Sistema_de_Autoconfiguracion_para_Redes_Ad_Hoc.pdf.
- [11] C. E. C. Arellano, "Análisis de protocolos de encaminamiento para redes inalámbricas tipo malla en modo infraestructura", 2011. [En línea]. Disponible en: <http://148.206.53.84/tesiuami/UAMI15536.pdf>.
- [12] F. J. García Romero, "El núcleo de red en redes de nueva generación", Vasa, 2012. [En línea]. Disponible en: http://oa.upm.es/13951/1/PFC_FRANCISCO_JAVIER_GARCIA_ROMERO.pdf.
- [13] N. N. Simulator, "Elección simulador de red", 2012. [En línea]. Disponible en: http://iie.fing.edu.uy/investigacion/grupos/artes/csic_mac_2012/Elecci%C3%B3n_NS3.pdf.
- [14] J. R. MorenoCadena, L. Román, y O. Gonzalo, "Evaluación Del Desempeño De La Herramienta ns-3 En Ambientes Inalámbricos Bajo El estándar IEEE 802.11." [En línea]. Disponible en: <http://repositorio.espe.edu.ec/bitstream/21000/6193/1/AC-TEL-ESPE-034950.pdf>.
- [15] G. N. U. Texinfo y F. S. Foundation GNU, "ns-3 Reference Manual", 2010. [En línea]. Disponible en: <https://www.nsnam.org/docs/release/3.9/manual.pdf>.
- [16] Inec, "Ecuador en Cifras Tic 2013", 2013 8th Computing Colombian Conference (8CCC), 2013. [En línea]. Disponible en: http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/Resultados_principales_140515.Tic.pdf.
- [17] V. V. Mancha, L. A. Zárate, y C. Verikoukis, "Diseño de protocolos MAC para redes ad-hoc utilizando antenas direccionales inteligentes.", 2005. [En línea]. Disponible en: <http://upcommons.upc.edu/pfc/bitstream/2099.1/3549/2/40798-2.pdf>.
- [18] T. C. A, C. P. E, y A. Q. A, "IMPLEMENTACIÓN DE PROTOCOLOS EN EL NETWORK SIMULATOR (NS-2)." [En línea]. Disponible en: <http://e-serv.uned.es/fez/eserv/taee:congreso-2006-1022/S1G01.pdf>.

- [19] C. Profentzas, "Studying Routing Issues in VANETs by Using NS-3", 2012. [En línea]. Disponible en: http://eureka.lib.teithe.gr:8080/bitstream/handle/10184/4515/christos_profentzas.pdf?sequence=1.
- [20] N. N. Simulator, "Ns-3 Manual Network Simulator", 2015. [En línea]. Disponible en: <https://www.nsnam.org/docs/manual/ns-3-manual.pdf>.
- [21] D. Rabinovich, C. Osimani, y G. Olocco, "Dimensionamiento de redes inalámbricas de banda ancha utilizando NS-3 mediante síntesis de tráfico autosimilar." [En línea]. Disponible en: <http://conaiisi.frc.utn.edu.ar/PDFsParaPublicar/1/schedConfs/8/65-455-1-DR.pdf>.
- [22] F. R. M. Solera, "Primeros pasos con ns-3.", 2014. [En línea]. Disponible en: http://repositorio.cedia.org.ec/bitstream/123456789/960/8/T8_ejemplos_sencillos_vf.pdf.
- [23] E. de la F. Rubio, "Redes Inalámbricas." [En línea]. Disponible en: <https://profesores.ing.unab.cl/~delaf/archivos/cursos/comunicaciones-inalambricas/laboratorios/Laboratorio+02+Modos.pdf>.
- [24] D. G. Robles, "Evaluación de algoritmos de propagación de mensajes en redes oportunistas", 2008. [En línea]. Disponible en: <http://upcommons.upc.edu/pfc/bitstream/2099.1/20050/4/memoria.pdf>.
- [25] E. K. Çetinkaya y J. P. G. Sterbenz, "Mobility Models in ns-3, Introduction to Network Simulation with ns-3", 2011. [En línea]. Disponible en: <http://www.ittc.ku.edu/~jpgs/courses/mwnets/lecture-lab-mobile-ns3-display.pdf>.
- [26] F. R. M. Solera, "INSTALACIÓN DE NS-3", 2006. [En línea]. Disponible en: http://repositorio.cedia.org.ec/bitstream/123456789/960/7/T7_Introduccion_Instalacion_vf.pdf.
- [27] J. Gavidia y R. Lara, "Análisis del desempeño de una red Ad-Hoc inalámbrica móvil MANET, empleando el software NS 2.32 bajo el estándar IEEE 802.15." [En línea]. Disponible en: http://wicom.espe.edu.ec/documentos/ANALISIS DEL DESEMPEÑO DE UNA RED AD-HOC INALAMBRICA MOVIL MANET EMPLEANDO EL SOFTWARE NS 2_32 BAJO EL ESTANDAR IEEE 802_15.pdf.

- [28] J. C. Ugas, "*Estudio y análisis de prestaciones de redes móviles Ad Hoc mediante simulaciones NS-2 para validar modelos analíticos*", Recercat.Net, 2009. [En línea]. Disponible en: https://upcommons.upc.edu/pfc/bitstream/2099.1/8374/1/PFC_Jordi_Chalmeta.pdf.
- [29] Aurilia Torres Ontaneda, "*Análisis de la calidad de Servicio en el enrutamiento de las Redes Móviles Ad Hoc*", 2010. [En línea]. Disponible en: [http://dspace.utpl.edu.ec/bitstream/123456789/1443/3/Utpl_Torres_Ontaneda_Auliriar_004x790. \(1\).pdf](http://dspace.utpl.edu.ec/bitstream/123456789/1443/3/Utpl_Torres_Ontaneda_Auliriar_004x790. (1).pdf).
- [30] D. D. La Cruz, R. Lara, y G. Olmedo, "*Análisis del Throughput Normalizado en una red Ad-Hoc Móvil*", 2007. [En línea]. Disponible en: <http://wicom.espe.edu.ec/documentos/ANALISIS DEL THROUGHPUT NORMALIZADO EN UNA RED AD-HOC MOVIL.pdf>.

k. Anexos

Anexo 1: Certificación de la traducción hecha al Resumen del Informe Final

Prof. Joan Morales
DOCENTE DE FINE-TUNED ENGLISH



CERTIFICA:

Que el documento aquí compuesto es fiel traducción del idioma español al idioma inglés del resumen de la tesis titulada: "ANALISIS Y SIMULACION DE PROTOCOLOS DE ENRUTAMIENTO ADECUADOS EN DIFERENTES ESCENARIOS PARA REDES AD HOC, MEDIANTE LA HERRAMIENTA Ns-3" de la autora Gabriela Maribel Rodríguez Pineda, egresada de la carrera de Ingeniería en Sistemas del Área de la Energía, las Industrias y los Recursos naturales No renovables de la Universidad Nacional de Loja.

Lo certifica en honor a la verdad y autoriza a la interesada hacer uso del presente en lo que a sus intereses convenga.

Loja, 25 de Junio de 2015


Prof. Joan Morales
DOCENTE DE FINE-TUNED ENGLISH



Anexo 2: Descripción de siglas

TABLA XXXIV. TABLA DE SIGNIFICADOS

SIGLAS	DESCRIPCIÓN	SIGNIFICADO
ADHOC	Mobile Ad-Hoc Network	Red Inalámbrica Móvil
AODV	Ad-hoc On Demand Vector	Ad-hoc en vector demanda
DSDV	Destination Sequenced Distance Vector	Destino Secuenciado con vector distancia
DSR	Dynamic Source Routing	Enrutamiento dinámico fuente
IP	Internet Protocol	Protocolo de Internet
IPV4	Internet Protocol Version 4	Protocolo de Internet versión 4
IPV6	Internet Protocol Version 6	Protocolo de Internet versión 6
LLC	Logical Link	Control lógico de enlace
MAC	Media Access Control	Control de acceso al medio
NS3	Network Simulator 3	Simulador de Redes versión 3
OLSR	Optimized link State Routing Protocol	Protocolo de enrutamiento de estado de enlace optimizado
PCAP	Packet capture data	Paquete de captura de datos
TCP	Transmission control protocol	Protocolo de control de transmisión
TORA	Temporally Ordered Routing ALgorithm	Algoritmo de enrutamiento de ordenamiento temporal
XML	Extensible Markup Languaje	Lenguaje de marcas extenso
ZRP	Zone Routing Protocol	Protocolo de zona de enrutamiento

Anexo 3: Licencia Creative Commons

Figura 134: Licencia Creative Commons

