



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

ANÁLISIS DE LA CALIDAD DE SERVICIO EN REDES MANET EN BASE A LOS  
PROTOCOLOS DE ENRUTAMIENTO.

Trabajo de Titulación presentado en conformidad con los requisitos establecidos  
para optar por el título de Ingeniero en Redes y Telecomunicaciones

Profesor Guía

MSc. Marcelo Ricardo Filian Narváez.

Autor

Francisco Xavier Cruz Hermosa

Año

2017

## **DECLARACIÓN PROFESOR GUÍA**

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”

---

MSc. Marcelo Ricardo Filian Narváz  
Magister en Conectividad y Redes de Telecomunicaciones  
CI. 0602888638

## **DECLARACIÓN DEL PROFESOR CORRECTOR**

“Declaro haber revisado este trabajo, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

---

Miltón Nepalí Román Cañizares  
MSc. En Gerencia de Redes y Telecomunicaciones  
C.I.: 0502163447

## **DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE**

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

---

Francisco Xavier Cruz Hermosa

CI. 171662025-5

## **AGRADECIMIENTOS**

“A Dios por iluminar cada paso que doy, por darme la oportunidad de vivir, de caminar, de ver, de hablar, de escuchar y de amar, agradezco a Dios por colocar personas sustanciales a mi alrededor y que han sido valiosos para culminar este período importante en mi vida, mi gratitud a Dios por todo lo que he obtenido y por todo lo que soy.”

## **DEDICATORIA**

“Dedico este logro a mis padres por ser el pilar fundamental de equilibrio, constancia, perseverancia y muchos otros valores que diariamente con su ejemplo y amor me han ilustrado para seguir adelante hasta conseguir mis objetivos”

## RESUMEN

Las redes móviles Ad-Hoc son un conjunto de dispositivos conectados mediante la tecnología Wireless, las cuales poseen cierta movilidad independiente y facilidad de desplazarse en cualquier dirección sin la necesidad de infraestructuras preexistentes. El cambio dinámico de sus estaciones móviles permite realizar estudios de las condiciones de enlace entre dispositivos y el comportamiento de sus protocolos de enrutamiento. Estas redes inalámbricas minimizan el uso de puntos de acceso, infraestructuras cableadas y costos de mantenimiento, puesto que cada dispositivo genera simultáneamente comunicación como router y nodo final. La tecnología móvil Ad-Hoc se emplea en; diversas áreas militares, en lugares donde la comunicación por cable sea difícil alcanzar, para el rescate de las personas en caso de catástrofes naturales, o dónde los servicios de las comunicaciones pierdan su equilibrio. Para analizar una red Ad-Hoc se han creado diversos algoritmos de enrutamiento mediante el cual, se pueda generar diversos tipos de dinamismo en los nodos de una red. Se ha seleccionado dos grupos importantes de enrutamiento para su interpretación, estos son: protocolos reactivos (AODV y DSR) y proactivos (DSDV). Mediante las simulaciones creadas en este trabajo, se observará en cada uno de los protocolos cuántos paquetes pueden generar, enviar, perder y recibir dentro de una red Ad-Hoc. Para el desarrollo de las redes se utilizaron los programas de simulación NS-2, XGRAPH y NSG2, con las cuales se realizó diferentes pruebas para verificar el canal eficaz, retardo y pérdida de paquetes, de tal manera, obtener resultados claros, precisos y reales para validar el mejor protocolo y calidad de servicio. Para finalizar el trabajo de titulación, se realizó un manual que ayudará al usuario a realizar las instalaciones de los programas de simulación y su uso adecuado. Se generará una guía para el lector de este proyecto de modo que, ayude a desarrollar temas de investigación, estudio e implementación de nuevas aplicaciones para las redes móviles Ad-Hoc, contribuyendo así al avance tecnológico de estas redes tan importantes.

## **ABSTRACT**

Ad-Hoc mobile networks are a set of devices connected by wireless technology, which have some independent mobility and ease of movement in any direction without the need for pre-existing infrastructures. The dynamic change of their mobile stations allows to study the conditions of link between devices and the behavior of their routing protocols. These wireless networks minimize the use of access points, wired infrastructures and maintenance costs, since each device simultaneously generates communication as a router and final node. Ad-Hoc mobile technology is used in; Various military areas, in places where cable communication is difficult to reach, to rescue people in the event of natural disasters, or where communications services lose their balance. In order to analyze an Ad-Hoc network, several routing algorithms have been created through which different types of dynamism can be generated in the nodes of a network. Two important routing groups have been selected for interpretation, these are: reactive (AODV and DSR) and proactive (DSDV) protocols. Through the simulations created in this work, it will be observed in each of the protocols how many packets they can generate, send, lose and receive within an Ad-Hoc network. For the development of the networks the NS-2, XGRAPH and NSG2 simulation programs were used, with which different tests were carried out to verify the effective channel, delay and loss of packages, in order to obtain clear, accurate and real results To validate the best protocol and quality of service. To finalize the titling work, a manual was developed that will help the user to carry out the installations of the simulation programs and their proper use. A guide will be generated for the reader of this project so that it helps to develop research topics, study and implementation of new applications for Ad-Hoc mobile networks, thus contributing to the technological advancement of these important networks.



# ÍNDICE

INTRODUCCIÓN .....	1
1. Capítulo I. Redes inalámbricas Ad-Hoc o Redes MANET .....	1
1.1.- Arquitectura de una red Ad-Hoc .....	5
1.2.1.- Capa física.....	6
1.2.2.- Capa de enlace.....	6
1.2.3.- Capa de red.....	7
1.2.4.- Capa de transporte .....	7
1.2.5.- Capa de aplicación .....	7
1.3.- Algoritmos de enrutamiento .....	7
1.3.1.- Algoritmo de estado de enlace .....	8
1.3.2.- Algoritmo vector distancia.....	9
1.3.3.- Algoritmo source routing (Enrutamiento de Origen).....	10
1.4.- Métricas de enrutamiento aplicables en redes móviles Ad-Hoc .....	10
1.4.1.- Número de saltos.....	10
1.4.2.- Round Trip time (RTT – Tiempo de retardo de ida y vuelta) por salto .....	11
1.4.3.- ETX (Expected Transmission Count - Recuento de Transmisión Esperada) .....	11
1.4.4.- MIC (Metric of interference and cannel switching- Métrica de enrutamiento para interferencias y canales) .....	11
1.5.- Clasificación de los protocolos de enrutamiento .....	12
1.5.1.- Protocolos de enrutamiento proactivos.....	12

1.5.1.1.- Ventajas de los protocolos de enrutamiento proactivos .....	12
1.5.1.2.- Desventajas de los protocolos de enrutamiento proactivos.....	12
1.5.1.4.- OLSR (Optimized Link Routing- Protocolo de ruta de enlace optimizado) .....	13
1.5.1.5.- Protocolo DSDV (The Destination Sequenced Distance Vector Routing Protocol - Destino secuenciado y enrutamiento de vector distancia) .....	13
1.6.- Protocolos de enrutamiento reactivos.....	15
1.6.1.- Ventajas de los protocolos de enrutamiento reactivos .....	15
1.6.2.- Desventajas de los protocolos de enrutamiento reactivos .....	16
1.6.2.1.- Protocolos de enrutamiento reactivos aplicables para redes Ad-Hoc.....	16
1.6.2.2.- AODV (Ad-Hoc on Demand Distance Vector Routing- Ad-Hoc en demanda de enrutamiento de vector de distancia).....	16
1.6.2.3.- DSR (Dynamic Source Routing-Enrutamiento Dinámico Fuente) .....	17
1.7.- Protocolos de enrutamiento híbridos aplicables en redes Ad-Hoc ....	18
1.7.5.- Parámetros de comparación para los protocolos de enrutamiento en las redes Ad-Hoc.....	20
1.7.5.1.- Métricas de enrutamiento .....	20
1.7.5.2.- Libre de lazos .....	20
1.7.5.3.- Escalabilidad.....	20
1.7.5.4.- Confiabilidad .....	20
1.7.5.5- Balance de carga.....	20
1.7.5.6.- Control de gestión.....	20
1.7.5.7.- Throughput (Desempeño).....	20
1.8.- Calidad de servicio (QOS) en las redes Ad-Hoc .....	23
1.8.1.1.- Pérdida de paquetes .....	23

1.8.1.2.- Retardo o Latencia .....	23
1.8.1.3.- Canal eficaz o throughput.....	23
1.8.2.1.- Señalización de reserva para la calidad de servicio .....	24
1.8.2.2.- Encaminamiento con calidad de servicio.....	24
<b>1.9.- Seguridades en Redes Ad-Hoc .....</b>	<b>24</b>
1.9.1.- Vulnerabilidades en redes Ad-Hoc .....	24
1.9.1.1.- Nodo malicioso .....	24
1.9.1.3.- Nodo comprometido .....	25
1.9.1.3.- Nodo selfish.....	25
1.9.2.- Atributos de seguridad.....	25
1.9.2.1.- Disponibilidad .....	25
1.9.2.2.- Autenticación .....	25
1.9.2.3.- Confidencialidad .....	25
1.9.2.4.- Integridad .....	26
1.9.2.5.- Autorización.....	26
<b>1.10.- Ataques en las redes Ad-Hoc .....</b>	<b>26</b>
1.10.1.- Ataques pasivos .....	26
1.10.2.- Ataques activos .....	26
1.10.3.- Protocolos en encaminamiento seguros.....	26
1.10.3.1.- Ataque al protocolo de encaminamiento.....	27
1.10.3.2.- Desbordamiento de la tabla de encaminamiento.....	27
1.10.3.3.- Envenenamiento de la tabla de encaminamiento .....	27
1.10.3.4.- Replicación de paquetes .....	27
1.10.3.5.- Sleep Deprivation (Privación de sueño).....	27
1.10.3.6.- Divulgación de ubicación .....	27

2. CAPÍTULO II CONSIDERACIONES Y LIMITACIONES PARA EL ANÁLISIS DE LOS PROTOCOLOS DE ENRUTAMIENTO.....	28
2.1.- Parámetros de calidad de funcionamiento de red.....	29
2.2.- Características de los escenarios de simulación con los protocolos de enrutamiento para redes móviles Ad-Hoc.....	31
2.3.- Diseño y funcionamiento del escenario de simulación 1 con protocolo de enrutamiento reactivo AODV. ....	33
2.4.- Diseño y funcionamiento de los escenarios 2, 3, 4 con protocolo de enrutamiento reactivo AODV y 5, 6, 7 con protocolo de enrutamiento proactivo DSDV. ....	39
2.5.- Diseño y funcionamiento del escenario de simulación 8 con protocolo de enrutamiento proactivo DSDV. ....	41
2.6.- Diseño y funcionamiento de los escenarios 9, 10 y 11 con protocolo de enrutamiento reactivo DSR. ....	46
2.7.- Diseño y funcionamiento del escenario de simulación 12 con protocolo de enrutamiento reactivo DSR.....	49
3. CAPÍTULO III ANÁLISIS DE LOS PROTOCOLO DE ENRUTAMIENTO REACTIVOS Y PROACTIVOS.....	55
3.1.- Análisis de resultados obtenidos de los protocolos de enrutamiento AODB, DSR y DSDV.....	56
3.2.- Análisis comparativo de los parámetros de calidad de funcionamiento de red mediante los indicadores ITU-T. ....	60
4. CAPÍTULO IV SIMULADOR DE REDES NS-2.....	64
4.1.- Historia .....	64

4.1.1.- Características del programa de simulación .....	65
4.1.2.- Tipo de herramientas.....	65
4.1.3.- Tipo de archivos .....	66
4.1.4.- Importancia del software NS-2.....	67
4.1.5.- Consideraciones de NS-2.....	67
4.1.6.- Consejos y datos para simulaciones grandes en NS-2 ....	68
4.1.6.1.- Tiempo de arranque.- .....	68
4.1.6.2.- Tiempo de ejecución.....	68
4.1.6.3.- Uso de memoria .....	68
4.1.7.- Ventajas y desventajas de un simulador de red.....	68
4.1.7.1.- Ventajas.....	69
4.1.7.2.- Desventajas .....	69
<b>5. CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>70</b>
5.1.- Conclusiones .....	70
5.2.- Recomendaciones .....	71
<b>REFERENCIAS.....</b>	<b>73</b>
<b>ANEXOS .....</b>	<b>76</b>

## ÍNDICE DE FIGURAS

Figura 1. Redes Ad-Hoc.....	4
Figura 2. Redes Ad-Hoc.....	4
Figura 3. Capas del modelo TCP. Adaptada de Are Carrasco, 2011 .....	6
Figura 4. Clases de algoritmos de enrutamiento. Adaptada de J. Ramón Sánchez,2007.....	8
Figura 5. Algoritmo de estado de enlace. Adaptada de J.Ramón Olague, 2007. ....	9
Figura 6. Algoritmo Vector Distancia. Adaptada de J. Ramón Olague, 2007 ...	10
Figura 7. Algoritmo source routing- Enrutamiento de Origen. ....	10
Figura 8. Comparación de los protocolos de enrutamiento Reactivos, Proactivos e Híbridos aplicables en redes móviles Ad-Hoc. Adaptada de Revista Telem@tica. Vol.13 No. 3, septiembre -diciembre, 2014.....	22
Figura 9. Parámetros de la calidad de funcionamiento en la red. (Nota: La “U” significa “no especificado” o “sin límites”, Tomado de: “Satélite Networking Principles and Protocols .....	30
Figura 10. Características de los escenarios de simulación con protocolos de enrutamiento.....	31
Figura 11. Diseño del escenario 1 con 9 nodos Protocolo de enrutamiento reactivo - AODV.....	33
Figura 12. Actualización de nodos y tablas de rutas .....	34
Figura 13. Nodo 8 dentro de cobertura con protocolo AODV.....	35
Figura 14. Envío de tramas entre los nodos 1-6-8 con protocolo AODV .....	35
Figura 15. Pérdida de las tramas con protocolo reactivo AODV .....	36
Figura 16. Envío de tramas entre los nodos 1-6-4-8 con protocolo reactivo AODV. ....	37
Figura 17. Pérdida de información con protocolo reactivo AODV. ....	37
Figura 18. Finalización del envío y recepción de tramas con protocolo reactivo AODV.....	38
Figura 19. Diseño del escenario 2, 3 y 4 con 22 nodos mediante el Protocolo de enrutamiento reactivo - AODV.....	39

Figura 20. Movilidad de 4 nodos e intercambio de información con los protocolos AODV y DSDV en las simulaciones 2, 3, 5 y 6. ....	40
Figura 21. Diseño del escenario 3 con 9 nodos Protocolo de enrutamiento Proactivo DSDV.....	41
Figura 22. Actualización de las tablas con protocolo proactivo DSDV. ....	42
Figura 23. Pérdida de información entre los nodos origen y destino con protocolo proactivo DSDV. ....	43
Figura 24. Pérdida de información entre los nodos origen y destino con protocolo proactivo DSDV. ....	44
Figura 25. Actualización y renovación de envío de datos entre nodo origen y destino con protocolo proactivo DSDV. ....	45
Figura 26. Escenarios 8, 9 y 10 con protocolo de enrutamiento reactivo DSR	46
Figura 27. Transmisión de datos entre los nodos origen y destino .....	47
Figura 28. Dinamismo entre diversos nodos vecinos para la transmisión de datos entre el nodo origen y destino.....	48
Figura 29. Diseño del escenario 2 con 9 nodos Protocolo de enrutamiento reactivo DSR .....	49
Figura 30. Actualización de las tablas con protocolo reactivo DSR.....	50
Figura 31. Vinculación del Nodo 2 y 4 con protocolo reactivo DSR .....	51
Figura 32. Envío de tramas desde nodo origen a nodo destino con protocolo reactivo DSR. ....	51
Figura 33. Cambio de ruta al nodo 4 con protocolo reactivo DSR.....	52
Figura 34. Mediador de información nodo 3 con protocolo reactivo DSR. ....	53
Figura 35. Envío y recepción de tramas entre nodo origen y destino con protocolo reactivo DSR.....	54
Figura 36. Tabla comparativa de los resultados de 12 escenarios mediante los Protocolos de enrutamiento (AODV, DSR y DSDV). ....	56
Figura 37. Análisis de los parámetros de calidad de funcionamiento de red por medio de los indicadores ITU-T. ....	62

## ÍNDICE DE TABLAS

Tabla 1. Comparación de los protocolos de enrutamiento Reactivos, Proactivos e Híbridos aplicables en redes móviles Ad-Hoc. Adaptada de: Revista Telemática, 2014., 2014 .....	22
Tabla 2. Parámetros de la calidad de funcionamiento en la red. (Nota: La “U” significa “no especificado” o “sin límites”), Tomado de: Satellite Networking Principles and Protocols, .....	30
Tabla 3. Tabla comparativa de los resultados de 12 escenarios mediante los Protocolos de enrutamiento (AODV, DSR y DSDV). 36 Tabla comparativa de los resultados de 12 escenarios mediante los Protocolos de enrutamiento (AODV, DSR y DSDV). .....	56
Tabla 4. Análisis de los parámetros de calidad de funcionamiento de red por medio de los indicadores ITU-T. Análisis de los parámetros de calidad de funcionamiento de red por medio de los indicadores ITU-T. ....	62



## INTRODUCCIÓN

Las redes MANET (Mobile Ad-Hoc network) se han convertido en una solución inalámbrica ya que brindan a los usuarios flexibilidad, autonomía y minimizan el uso de cualquier tipo de infraestructura fija o central para su implementación, por ejemplo, en momentos de emergencia estas redes brindan conectividad de manera sencilla y rápida para que otros dispositivos se puedan unir a la red.

Estas redes presentan grandes ventajas; la configuración, movilidad, costos y tiempos son factores que resultan convenientes al momento de implementar una red de dispositivos móviles, de esta manera, al no necesitar de un nodo central los montajes de red inalámbrica se pueden dar en cualquier lugar y en cualquier momento. Las desventajas que se presentan en estas redes móviles, se deben a su limitación del ancho de banda y sus múltiples desconexiones, que restringe al usuario el uso de las aplicaciones en tiempos reales. Otra desventaja que se presenta en esta tecnología inalámbrica, es la existencia de cambios en su topología de red, las cuales producen pérdida de paquetes, ya que los protocolos de enrutamiento no se dan cuenta de este tipo de cambios repentinos existentes en las red Ad-Hoc.

Los problemas que surgen debido al comportamiento de los nodos minimizan la calidad de servicio de las redes MANET como por ejemplo: la tasa de error se incrementa debido a las interferencias y atenuaciones en la transmisión de paquetes. Otro problema que se presenta es la inseguridad, ya que pueden ser atacadas en cualquier momento por intrusos, de modo que, logran descubrir el tráfico que se intercambia o envían paquetes para inhabilitar la red.

Cada nodo necesita energía (CPU), para aprovechar al máximo la transmisión de datos, pero al no poseer de este recurso, la capacidad de envío y recepción de paquetes baja notablemente debido a una potencia débil de sus dispositivos.

En la actualidad las redes MANET se enfocan cada día más en el avance tecnológico y desarrollo de las comunicaciones vehiculares, sensores de redes,

aplicaciones militares, entornos civiles, situaciones de emergencia, operaciones de rescate y búsqueda, (Karol Casierra, 2016)

## **JUSTIFICACIÓN**

Debido a la evolución de la tecnología y tomando en consideración que el ser humano ha hecho de ella parte de su vida cotidiana y una necesidad para poder comunicarse de forma más rápida y sencilla, en este proyecto se analizará cómo mantener una red Ad-Hoc óptima mediante los análisis de los resultados en diferentes escenarios.

Posteriormente el proyecto es un aporte para el lector dentro de los campos de investigación que realiza el Departamento de la Facultad de Ingeniería y Ciencias Agropecuarias de la Universidad de las Américas ya que tendrá una referencia de esta tesis para solucionar las inquietudes técnicas que se le pueden presentar al momento de implementar múltiples proyectos y desarrollo de aplicaciones para la tecnología de red móvil inalámbrica Ad-Hoc.

## **OBJETIVOS**

### **Objetivo General**

Evaluar los parámetros de calidad en las redes MANET en función de los protocolos de enrutamiento y el dinamismo de la topología.

### **Objetivos Específicos**

Análisis comparativo de la afectación de los parámetros de calidad en base al tipo de protocolo de enrutamiento.

Realizar la simulación a partir de los distintos escenarios donde se ajusten los protocolos de enrutamiento y el dinamismo de topologías de la red.

Realizar un estudio amplio del funcionamiento del programa NS-2 que ha sido seleccionado para realizar las simulaciones de este proyecto.

### **Alcance**

En el desarrollo del presente proyecto se pretende mejorar la calidad de servicio en redes móviles Ad-Hoc mediante evaluaciones de los parámetros de calidad en

la red en base al dinamismo de las topologías y protocolos de enrutamiento que se pueden presentar dentro de estas redes, y por medio de un análisis de los escenarios realizar comparaciones de los resultados en los que se pueden encontrar pérdidas de paquetes, retardo e inestabilidad de comunicación.

Para poder interpretar los resultados deseados de las simulaciones con el software NS-2 se necesitará de los siguientes protocolos:

- **Proactivos:** Los nodos mantienen información de encaminamiento hacia todos los dispositivos de la red, Dentro de este grupo se encuadran aquellos protocolos que basan su funcionamiento en el intercambio periódico de mensajes para actualizar sus tablas de rutas, dentro de este protocolo se analizara el algoritmo DSDV (The Destination Sequenced Distance - Vector Routing Protocol).
- **Reactivos:** En este tipo de protocolos los nodos actualizan las tablas de encaminamiento solamente en caso de necesidad, su característica principal es que, si hay movilidad o cambio de topología, los dispositivos no actualizan sus tablas de encaminamiento a menos que lo requieran, dentro de este protocolo se analizara el algoritmo DSR (Dynamic Source Routing).
- **Híbridos:** Este protocolo divide la red en grupos en los que de forma local se utilice un protocolo proactivo, mientras que para conocer las rutas hacia los demás grupos utilice uno reactivo.

## CAPÍTULO I. REDES INALÁMBRICAS AD-HOC O REDES MANET

Las redes Ad-Hoc son un conjunto de nodos o dispositivos que se comunican de forma inalámbrica, conjuntamente crean una red temporal en la que no se maneja una administración centralizada. Por lo tanto, su sistema de red emigra dinámicamente debido al constante movimiento de los nodos. Cómo se indica en la figura 1, la característica propia de una red Ad-Hoc es que sus terminales son finales, además los dispositivos que conforman esta red, realizan trabajos de retransmisión de paquetes típicamente vinculados con routers, la cual distribuyen los datos para enviar a otras direcciones.



Figura 1. Redes Ad-Hoc

### 1.1.- Arquitectura de una red Ad-Hoc

Los nodos adquieren características del router y host al mismo tiempo, puesto que la arquitectura Ad-Hoc se presenta en cada uno de los nodos bajo el modelo TCP/IP (Protocolo de Control de Transmisión/Protocolo de Internet), que realiza la interconexión lógica de todas las redes físicas y determina el grupo de reglas para realizar la comunicación. De acuerdo a la figura 2, se procederá a analizar los protocolos de enrutamiento aplicables más importantes, para los Reactivos se escogerá: (AODV y DSR), y para los Proactivos: (DSDV), ya que estos protocolos constantemente son los más usados e investigados.

No se considera para su análisis el protocolo Híbrido, porque sus algoritmos son una combinación de los protocolos de enrutamiento Reactivo y Proactivos, además, usar este algoritmo dificulta el avance de este proyecto ya que presenta muchos errores al ejecutar la simulación por lo que, se encuentra aún en desarrollo por los creadores del programa de simulador de redes NS-2.

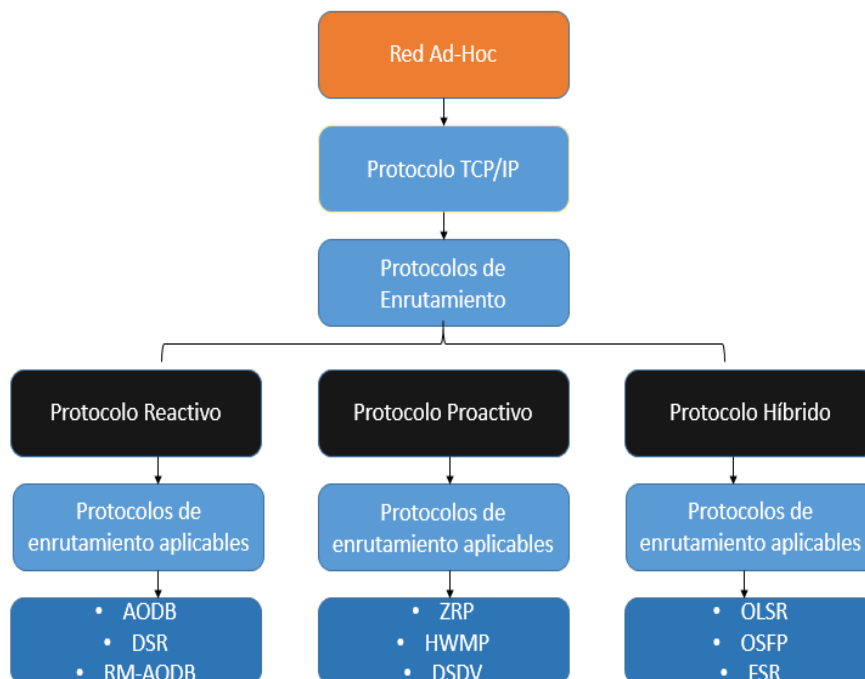


Figura 2. Arquitectura de una red Ad-Hoc

En la figura 2, se realiza una breve explicación de las capas que contiene una red móvil Ad-Hoc.



Figura 3. Capas del modelo TCP. Adaptada de Are Carrasco, 2011

### 1.2.1.- Capa física

El medio de transmisión en las redes Ad-Hoc es el aire, comunicándose mediante canales de RF (radio frecuencia). Los problemas que existen en estas redes inalámbricas, dificultan la presencia de la calidad de servicio cuando la cobertura se distorsiona por varios fenómenos que aparecen en el aire, provocando la atenuación de la señal y el efecto doopler. El estándar que se utiliza es IEEE 802.11X que determina las interfaces para los canales de radio frecuencia en las siguientes bandas: 2.4GHz y 5GHz.

### 1.2.2.- Capa de enlace

La capa de enlace de datos es la encargada de la transferencia fiable de información y responsable de definir el nodo que será próximo para su transmisión. Los protocolos válidos para una red Ad-Hoc es 802.11 a, con una velocidad de 25Mbps a 54 Mbps en ISM (Industrial, Scientific and Medical) con 5GHz para uso no comercial que pertenece a una banda reservada internacionalmente, usan modulación OFDM (Orthogonal Frequency Division Multiplexing) y FEC (Forward Error Division Multiplexing).

El estándar (802.11 b), suministra hasta 11Mbps en la banda ISM con 2,4GHz gestionando técnicas de envío de datos de espectro directo (DSS), y técnicas

de salto de frecuencia (FHSS), con la que obtiene hasta 2Mbps. Así mismo el estándar (802.11 g), trabaja hasta 54 Mbps en la banda 2,4GHz.

El objetivo del nivel de enlace, es asegurar que la información fluya correctamente y libre de errores entre los dispositivos que se encuentren conectados. Esta capa gestiona la dirección y el control de flujo entre nodos para evitar que un nodo más acelerado retire del camino a un nodo más lento.

### **1.2.3.- Capa de red**

La capa de internet o de red, es la encargada de enviar los datos o paquetes a su receptor, además proporciona la selección de la ruta y conectividad entre los diferentes nodos que pueden existir en una red Ad-Hoc.

### **1.2.4.- Capa de transporte**

Esta capa es la que define un destino de una dirección específica. Los protocolos que se usan en esta capa son UDP y TCP. El protocolo TCP supervisa y asegura que todos los datos hayan llegado al destino, estos paquetes son empalmados en forma ordenada para ser entregados a la próxima capa, cuando un paquete presenta error este protocolo espera a que el canal se descongestione para no reducir las tasas de envío de datos, mientras que el protocolo UDP, entrega los datos no orientados a conexión, este sistema no garantiza que la información llegue a su destino ya que los paquetes son encapsulados y en modo de espera para que lleguen al receptor.

### **1.2.5.- Capa de aplicación**

En las redes Ad-Hoc, la capa aplicación no funciona de la misma manera que en las redes cableadas, por lo que el dinamismo de sus nodos no les resulta tan fácil soportar aplicaciones que necesitan los usuarios para transmitir multimedia y aplicaciones interactivas en tiempo real, es por ello que existen propuestas para que estas aplicaciones puedan adaptarse a las redes Ad-Hoc de una manera efectiva.

## **1.3.- Algoritmos de enrutamiento**

Los algoritmos de enrutamiento se emplean porque son responsables de la especificación del enlace en cada paquete, de esta forma, el algoritmo debe encontrar un nodo en la red para enlazar los datos de un punto a otro. Se toma

en cuenta que un enrutador puede realizar dos procesos internos, uno maneja los paquetes conforme va llegando a la red y el otro actualiza las tablas de enrutamiento para decidir cuales rutas va a utilizar al momento de dirigir la información. Este algoritmo utiliza los puertos de salida para transmitir la información organizada, confiada y dirigida por el nivel de red en función de su dirección IP de destino. Cómo se muestra en la figura 4, los algoritmos de enrutamiento más utilizados son:

- Vector Distancia
- Estado de Enlace
- Source Routing.

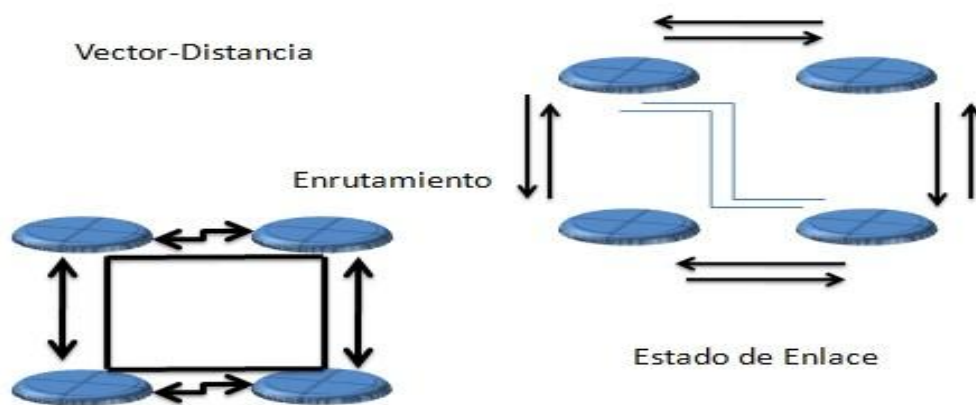


Figura 4. Clases de algoritmos de enrutamiento. Adaptada de J. Ramón Olague Sánchez, 2007.

### 1.3.1.- Algoritmo de estado de enlace

Este algoritmo asigna un valor a cada enlace, en cada uno de ellos existe un mapa completo de la topología de la red que es otorgada por cada nodo, es decir, que cada nodo genera el valor de las conexiones a las cuales está conectado. Mientras que los demás nodos actualizan o restauran la tabla de enrutamiento y el mapa de red, éste algoritmo detecta la ruta o vía más corta, como se indica en la figura 5.



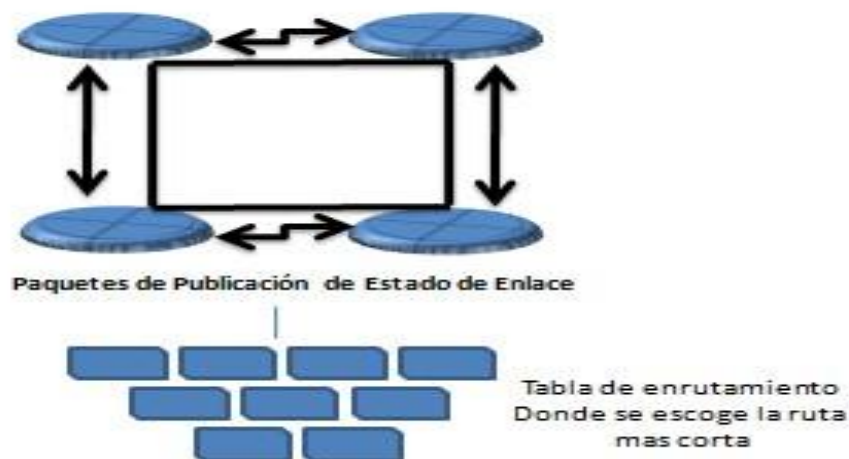


Figura 5. Algoritmo de estado de enlace. Adaptada de J.Ramón Olague, 2007.

### 1.3.2.- Algoritmo vector distancia

El algoritmo vector distancia ya define el costo de los enlaces, ya que cada nodo conoce específicamente la ruta a la que está conectado, a su vez, emite ésta información al nodo que está alrededor. Cada uno de ellos calcula los datos que recibe en su tabla de enrutamiento, es decir, esta operación se basa en el número de saltos que un paquete de datos realiza al pasar por todos los nodos hasta llegar a su destino. Cuando esto sucede el algoritmo determina la ruta óptica mediante el menor número de saltos.

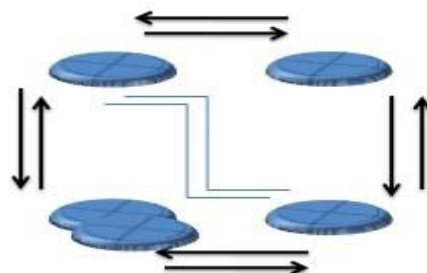


Figura 6. Algoritmo Vector Distancia. Adaptada de J. Ramón Olague, 2007.

### 1.3.3.- Algoritmo source routing (Enrutamiento de Origen).

Los enlaces o las rutas ya están establecidos para los paquetes ya que todas las decisiones relacionadas al enrutamiento de origen vienen procesadas desde las fuentes.

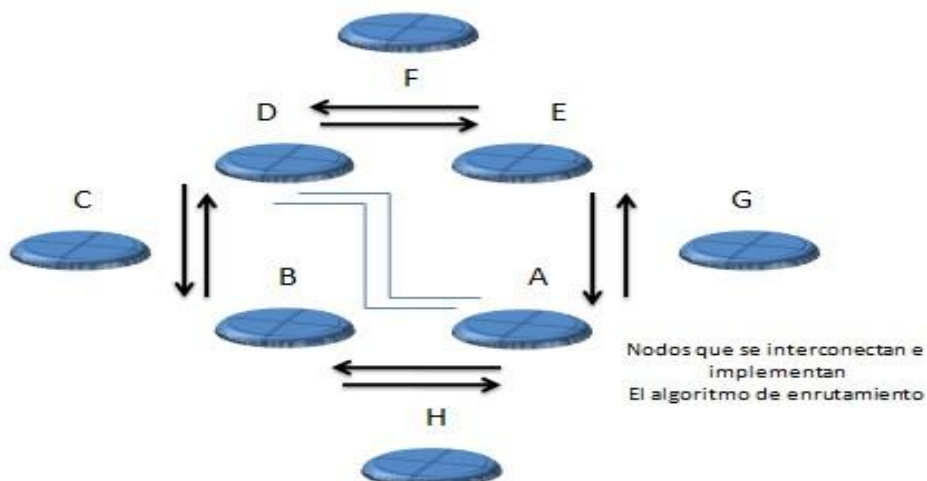


Figura 7. Algoritmo source routing- Enrutamiento de Origen.

### 1.4.- Métricas de enrutamiento aplicables en redes móviles Ad-Hoc

En las redes Ad-Hoc, las métricas son valores que toman los protocolos de enrutamiento para determinar la mejor ruta hacia una red destino. Son usadas para comparar y elegir la ruta que sea más conveniente al momento de transmitir datos, estos parámetros están basados en la topología de la red, a continuación se mencionará algunas de ellas.

#### 1.4.1.- Número de saltos

En este tipo de redes, el número de saltos es una de las métricas más utilizadas porque cada conexión o enlace es tomado en cuenta para la selección de rutas. Cada uno de los nodos que transmiten información conlleva un salto, al generar más saltos, mayor es el tiempo en que tarda la información en llegar a su destino, es por ello que presenta desperdicio de paquetes,

retraso y con alta probabilidad de que su rendimiento no sea óptimo en cada salto que se ejecuta en los nodos.

#### **1.4.2.- Round Trip time (RTT – Tiempo de retardo de ida y vuelta) por salto**

El RTT indica el retardo bidireccional en un enlace. El funcionamiento de esta métrica se da por ciclos, cada nodo que se encuentra alrededor contiene un paquete de medición con valores de fecha y hora (times-tamp). Cuando obtiene la información de los nodos vecinos, éste regresa a su ruta de origen generando retardo por cada salto en la red. El resultado o respuesta indica al nodo emisor que empieza a calcular el valor de ida y vuelta (RTT), el resultado es la suma de todos y cada uno de los valores de RTT evaluados en el camino o enlace de la ruta. Cuando se envían paquetes en la red en este tipo de métricas, siempre existe sobrecarga por el peso de información que se acumula en las colas, lo que implica demoras de transmisión e inestabilidad de enrutamiento en los canales, conocido también como (fenómeno de auto-interferencia).

#### **1.4.3.- ETX (Expected Transmission Count - Recuento de Transmisión Esperada)**

La métrica ETX evalúa el número necesario de transmisiones, de modo que, incluye un grupo de retransmisiones cuando envía paquetes de datos por medio de un enlace. Para poder mejorar su rendimiento, se minimiza el número de las transmisiones, el consumo de energía y las interferencias que se producen al momento de realizar la conexión.

El fundamento a utilizar de este algoritmo, es la suma de los datos de las métricas de cada enlace que integra dicha ruta, la cual envía en forma ordenada cada paquete a su destino.

#### **1.4.4.- MIC (Metric of interference and channel switching- Métrica de enrutamiento para interferencias y canales)**

Esta métrica, controla las interferencias intra-flujo e inter-flujo. Este algoritmo depende de algunos parámetros para mejorar el envío y recepción de paquetes como son el IRU (Interference Aware Resource Usage- Recursos de Interferencia) y (Channel Switching Cost- Costo de conmutación de canales).

La MIC, es una métrica que considera la diversidad de canal para evitar una elevada sobrecarga al momento de calcular el tamaño de paquetes en cada uno de los nodos.

### **1.5.- Clasificación de los protocolos de enrutamiento**

La clasificación de los protocolos de enrutamiento en las redes Ad-Hoc, se distinguen por varias características y parámetros que están basadas en; movilidad y actualización de sus nodos, establecimiento de rutas, descubrimiento de enlaces, mantenimiento de cabeceras, saltos de rutas y dinamismo en sus topologías, estos son:

- Proactivos
- Reactivos
- Híbridos

#### **1.5.1.- Protocolos de enrutamiento proactivos**

En cada lapso de tiempo se emiten paquetes los cuales son usados para descubrir nuevos nodos y el camino hacia ellos. Al momento en que se modifica un enlace, afecta a cualquier nodo de la red. Se emplea varias tablas para conservar actualizadas todas las rutas. Su ventaja principal es la manera de que los nodos vecinos pueden sustraer información de enrutamiento y enlazar una comunicación sencilla.

##### **1.5.1.1.- Ventajas de los protocolos de enrutamiento proactivos**

- Las rutas se buscan periódicamente.
- Cuando la tasa de movilidad es alta ofrece comportamientos accesibles a las solicitudes de las rutas.
- La cantidad de rutas útiles es relativamente mayor frente al total de las posibles.

##### **1.5.1.2.- Desventajas de los protocolos de enrutamiento proactivos**

- Exigen respuestas rápidas.
- Existe sobrecarga en la topología de la red.
- Afectación a cualquier nodo en caso de modificación de una ruta.

##### **1.5.1.3.- Protocolos de enrutamiento proactivos aplicables en redes móviles Ad-Hoc**

Los protocolos de enrutamiento proactivos para redes Ad-Hoc más utilizados son:

- OLSR (Optimized Link Routing- Protocolo de ruta de enlace optimizado).
- Protocolo DSDV (The Destination Sequenced Distance Vector Routing Protocol- Destino secuenciado y enrutamiento de vector distancia).
- CGSR (Cluster-head Gateway Switch Routing)
- BATMAN (Better Approach to Mobile Ad Hoc Networking)
- DREAM (Distance Routing Effect Algorithm for Mobility)

#### **1.5.1.4.- OLSR (Optimized Link Routing- Protocolo de ruta de enlace optimizado)**

En este algoritmo, cada uno de los nodos intercambian en toda la red mensajes para decidir cuál es el mejor enrutamiento de los paquetes, de esta manera minimiza el tamaño de paquetes de control manteniendo las rutas para todos los destinos en la red.

##### **Descubrimiento de rutas**

OLSR usa la técnica de retransmisores multipunto. Cada nodo escoge un conjunto de nodos vecinos que son seleccionados para reenviar todo el tráfico de control que es destinado a publicarse en toda la red. Los nodos de OLSR mantienen las rutas permanentes para poder alcanzar a cualquier otro destino dentro de la red.

##### **Mantenimiento de rutas**

Para poder llevar un registro de mantenimiento rutas, se realiza un control periódico de los mensajes, de ésta manera, cada nodo se va aprendiendo el estado de todos los nodos vecinos y la topología que se encuentra en la red.

#### **1.5.1.5.- Protocolo DSDV (The Destination Sequenced Distance Vector Routing Protocol - Destino secuenciado y enrutamiento de vector distancia)**

Los nodos vecinos en este algoritmo, intercambian periódicamente cada una de sus tablas de encaminamiento. Para calcular la distancia en las que se encuentran los demás nodos no vecinos, DSDV elige siempre el camino más corto para analizar el número de saltos hacia el mejor destino.

### **Descubrimiento de rutas**

Los nodos en cada una de sus tablas tienen la información de todos los vecinos nodos que constan de los datos siguientes:

- La dirección IP del destino
- La cantidad de secuencia de destino.
- El siguiente salto hacia el otro destino por medio de la dirección IP.
- El número de saltos al momento del coste de la ruta al destino.

Cada uno de los nodos contiene el número de secuencia en su tabla de enrutamiento, y cuando intercambian información, los datos de todos los demás nodos se actualizan en sus tablas de enrutamiento. Para descubrir nuevas rutas, este protocolo usa dos tipos de paquetes: full-dump y el paquete incremental que utiliza para disminuir el tráfico en la red.

### **Mantenimiento de rutas**

Cuando la topología de la red cambia o un enlace se cae, cada uno de los nodos reenvía su tabla de enrutamiento. La información que llega actualizada, consta de la dirección IP del destino, el número de saltos y números de secuencia.

Los números de secuencia son utilizados para diferenciar las rutas antiguas de las rutas nuevas, los números de secuencia cambian en cada nodo al momento de presentarse mayor número de saltos y dinamismo en la red, es decir, si existen secuencias con el mismo número se tomará como válido al nodo que mantenga el menor número de saltos posibles.

#### **1.5.1.6.- Protocolo CGSR (Cluster-head Gateway Switch Routing-protocolo de enrutamiento de puerta de enlace de clúster)**

Este protocolo está basado en un sistema de agrupación para cada una de sus tablas, cada tabla contiene un número de nodos que son guiados por un jefe de cabecera. Estos números son registrados y organizados al momento de establecer un nuevo enlace en la red para el envío de paquetes.

#### **1.5.1.7.- Protocolo BATMAN (Better Approach to Mobile Ad Hoc Networking)**

El protocolo BATMAN, divide los paquetes hacia todos los nodos para minimizar la manipulación de datos en las tablas de encaminamiento, esta técnica permite eliminar la necesidad de difundir la información a un solo nodo evitando la congestión de direcciones. Para detectar el mejor camino, cada nodo vecino cuenta los mensajes de origen y destino que fueron recibidos por medio de las diferentes rutas, de esa forma el protocolo BATMAN, avisa a todos sus vecinos de la existencia de nuevos nodos en la red para evitar un mayor número de saltos al momento de realizar la entrega de los datos.

#### **1.5.1.8. - Protocolo DREAM (Distance Routing Effect Algorithm for Mobility)**

Para encontrar un nodo en la red, cada uno de ellos mantiene guardado con una técnica llamada inundación, toda la información de su localización para transmitir a todos los demás nodos, permitiéndoles a los más cercanos la actualización permanente de sus rutas.

### **1.6.- Protocolos de enrutamiento reactivos**

Estos protocolos de enrutamiento permiten buscar una ruta cuando se necesite. Al momento que un nodo requiere encontrar el camino hacia un destino, este nodo inicia el proceso del descubrimiento de nueva ruta, manteniéndose en el destino hasta que este camino ya no sea requerido por el nodo inicial; este proceso limita la cantidad total de rumbos, así contiene la información del enrutamiento para reducir el tamaño de las tablas de rutas. Las desventajas que existen en estos protocolos son: menos calidad en las rutas, latencia al momento de procesar un nuevo descubrimiento y saturación en la red por la técnica de flooding (inundación).

#### **1.6.1.- Ventajas de los protocolos de enrutamiento reactivos**

- La ruta se busca solo cuando se necesita.
- La sobrecarga es mucho menor en cada enlace.

### 1.6.2.- Desventajas de los protocolos de enrutamiento reactivos

- Retrasos de rutas son mayores.
- Para el primer paquete la latencia es alta.
- Alta latencia por la sobrecarga.
- Las rutas individuales viven poco tiempo.
- Existe independencia entre las rutas.
- La cantidad de rutas útiles es menor frente a todas las posibles.

#### **z**

Entre los protocolos de enrutamiento más usados se considera los siguientes:

- AODV (Ad-Hoc on Demand Distance Vector Routing- Ad-Hoc en demanda de enrutamiento de vector de distancia).
- DSR (Dynamic Source Routing-Enrutamiento de origen dinámico).
- TORA (Temporally Ordered Routing Algorithm)
- LQSR (Link Quality Source Routing)

#### 1.6.2.2.- AODV (Ad-Hoc on Demand Distance Vector Routing- Ad-Hoc en demanda de enrutamiento de vector de distancia)

En el protocolo AODV el nodo mantiene la tabla de encaminamiento para cada rumbo conocido mediante el algoritmo vector distancia, al comienzo esta tabla está formada por nodos vecinos, y se añaden nuevos destinos cuando es necesario, es así que cuando el nodo requiere comunicarse con un nodo diferente que no se encuentre en la tabla, empieza desde cero un proceso de descubrimiento de ruta hacia el nodo destino, es decir, se emite un mensaje o varios mensajes de descubrimiento de ruta RREQ que se propagan en la red de nodos parecido al DSR, aquí los nodos desarrollan una tabla inversa de encaminamiento para que se pueda devolver las respuestas RREP de las peticiones de ruta al nodo que fueron enviadas originalmente o al inicio.

#### **Mantenimiento de rutas**

El mantenimiento de rutas se procesa para proteger la ruta encontrada mediante enlaces seguros. Cuando existe mucho movimiento en un nodo origen la topología de red cambia constantemente. A diferencia de esto, cuando un nodo destino o un nodo intermedio se mueve con paquetes dirigidos



a ellos, se presenta un mensaje de error (RERR), que es enviado al nodo inicial por el nodo que encontró el cambio dentro de la topología de la red.

### **Descubrimiento de rutas**

Se realiza por medio de la difusión de algún mensaje tomado como una solicitud de ruta RREQ (Route Request) y se inunda toda la red con varias peticiones hacia los nodos vecinos. Este protocolo soporta envíos broadcast y unicast, enviando peticiones de ruta por medio del broadcast y recibiendo las respuestas por medio del unicast. El proceso empieza verificando la solicitud de ruta en la tabla de enrutamiento para analizar si tiene una ruta específica hacia el destino, en caso de no tener, se envía un mensaje de broadcast de RREQ a los nodos vecinos. mensaje tiene incluido el número de secuencia, el identificador único y la dirección IP, cuando RREQ llega hacia el destino se verifica si la información ya ha sido recibida con el total de número de saltos, si no fue recibido el mensaje, se incrementará el número de saltos al momento de ser reenviada, cuando se identifica la ruta de destino se genera una respuesta RREP( route replay packet- Repetición de paquetes de rutas) como mensaje unicast, cuando la respuesta RREP regresa al sitio origen se desarrolla un camino de ida y en caso de que el nodo obtenga un mensaje igual al regreso este será tomado como rechazo.

#### **1.6.2.3.- DSR (Dynamic Source Routing-Enrutamiento Dinámico Fuente)**

El encaminamiento en este protocolo empieza desde el origen, en cada paquete de datos viene incluida una cabecera que contiene información sobre los nodos fijos que deben atravesar y que a su vez no necesita de mensajes periódicos. De esta forma disminuye la sobrecarga mediante mensajes de control.

### **Descubrimiento de ruta**

La petición de la ruta es por inundación, cada una de las peticiones lleva un identificador que no deja propagar duplicado de rutas. Cada petición registra su camino y si llega a su propio destino, éste contesta de manera automática al origen.

#### **1.6.2.4.- TORA (Temporally Ordered Routing Algorithm – Algoritmo de enrutamiento temporalmente ordenado)**

Es un protocolo de enrutamiento para las redes multihost, que posee atributos para la minimización de sobrecargas al momento de localizar nodos en los cambios de topología de red, estos son; ejecución distribuida, enrutamiento sin bucles, enrutamientos de multipath, mantenimiento de las rutas proactivas y reactivas. Tora es un algoritmo que necesita mantener información de los enrutadores adyacentes, es decir el número de saltos que existe entre nodos.

#### **1.6.2.5.- LQSR (Link Quality Source Routing – Enrutamiento de origen de calidad de enlace)**

El protocolo LQSR se utiliza en la interconexión para redes de malla, mediante el servicio inalámbrico wimax o wifi basado en el enrutamiento de origen dinámico (DSR). Este algoritmo encuentra a todos los nodos de la red y asigna balances relativos de ancho de banda, canal y pérdida para cada enlace posible. La ruta que existe entre dos nodos durante la transmisión de datos, puede consistir en más de un salto, es decir, pueden existir más nodos intermedios entre el origen y destino.

### **1.7.- Protocolos de enrutamiento híbridos aplicables en redes Ad-Hoc**

Los protocolos híbridos unen la técnica de los reactivos y los proactivos formando el enrutamiento interdominio e intradominio. El protocolo reactivo se usa para que se puedan comunicar entre los diversos dominios y los proactivos aportan con la comunicación a nivel de dominio. Los más utilizados por este protocolo son:

- ZRP (Zone Based Routing Protocol)
- ZHLS ( Zone - Based Hierarchical Link State)
- Protocolo CBRP ( Cluster Based Routing Protocol - Protocolo de enrutamiento jerárquico Basado en Clúster)
- DDR (Distributed Dynamic Routing)

#### **1.7.1.- Protocolo ZRP (Zone Based Routing Protocol – Protocolo de enrutamiento por zonas)**

Es un protocolo que aprovecha lo sobresaliente de los reactivos y proactivos. Para generar zonas de enrutamiento cada nodo tiene que detectar a los nodos vecinos que se localizan a una distancia determinada de un salto y a la cual puede llegar directamente. Este proceso de descubrimiento de vecinos se administra por medio del protocolo NDP (Neighbor Discovery Protocol), además brinda informes de los fallos entre los enlaces de la red.

#### **1.7.2.- Protocolo ZHLS (Zone - Based Hierarchical Link State – Estado de enlace jerárquico basado en zonas)**

Todos los nodos en la red construyen dos tablas de enrutamiento; una que va por dentro de la zona de enlaces y otra tabla de enrutamiento que se encuentra entre zonas. Este proceso implica gran sobrecarga de comunicación en la red.

Este protocolo es vulnerable a los ataques de seguridad procedentes de los nodos vecinos maliciosos los cuales generan pérdida de información e interrupción de llegada de datos al destino.

#### **1.7.3.- Protocolo CBRP (Cluster Based Routing Protocol - Protocolo de enrutamiento jerárquico Basado en zonas)**

Este protocolo no necesita realizar descubrimiento de ruta, ya que posee la información de los nodos vecinos dentro del clúster. Este algoritmo soporta enlaces simétricos, selectivos y asimétricos. Su escalabilidad aumenta según el número de nodos vecinos pero, puede acumular demasiado overhead. El protocolo CBRP elige el camino más corto para llegar al nodo destino mediante la optimización del enrutamiento.

#### **1.7.4.- Protocolo DDR (Distributed Dynamic Routing – Enrutamiento Dinámico Distribuido)**

Este algoritmo es eficiente controlando el ancho de banda simple sin bucles. Proporciona mecanismos diferentes para reducir la complejidad de los enrutamientos y mejorar el rendimiento del retardo dentro de la red. Este protocolo no necesita información de la ubicación física de los nodos ya que realiza denominación distribuida de las zonas de forma dinámica, reduciendo notablemente la difusión de información.

### **1.7.5.- Parámetros de comparación para los protocolos de enrutamiento en las redes Ad-Hoc**

La comparativa de los protocolos se realiza para diferenciar las principales características y así escoger la mejor función que se adapte dentro de la red, los parámetros a analizar son:

#### **1.7.5.1.- Métricas de enrutamiento**

Son los parámetros que definen el conjunto de las rutas.

#### **1.7.5.2.- Libre de lazos**

Analiza el protocolo que es capaz de eludir la ocurrencia de lazos dentro de la red.

#### **1.7.5.3.- Escalabilidad**

Define que protocolo determina la inserción de nuevos nodos sin afectar al rendimiento de la red

#### **1.7.5.4.- Confiabilidad**

Determina que protocolo puede garantizar la entrega de paquetes su destino.

#### **1.7.5.5- Balance de carga**

Es el proceso para distribuir el tráfico y la carga parejamente en la red y así evitar las sobrecargas.

#### **1.7.5.6.- Control de gestión**

Mediante este proceso se realiza la asignación de recursos para la red, de tal forma esta pueda operar de una manera aceptable con un nivel de rendimiento alto.

#### **1.7.5.7.- Throughput (Desempeño)**

Es el promedio de la distribución exitosa de los paquetes en un solo canal de comunicación.

#### **1.7.5.8.- Análisis Comparativo de protocolos de enrutamiento para redes MANET.**

En la siguiente figura 8, se puede apreciar un breve análisis de los principales protocolos de enrutamiento que se pueden aplicar en las redes Ad-Hoc.

Mediante los parámetros de comparación, se puede distinguir que protocolos presentan mejores características al momento de ser utilizados. Teóricamente el análisis realizado en esta tesis se establece que el algoritmo ZRP que pertenece al protocolo Híbrido es el mejor al momento de ser generado. La métrica utilizada es mediante saltos por zonas, detectando el camino más corto para llegar a los nodos vecinos. Este protocolo es capaz de evitar golpes internos de lazos dentro de la red. La escalabilidad, confiabilidad, balance de carga y el control de congestión aumentan en base a la poca cantidad de saltos que son generados.

Los protocolos de enrutamiento que mejor se distinguen luego del Algoritmo ZRP son; DSR y AODV que pertenecen a los reactivos. Estos contienen características semejantes al momento de ser ejecutados, ya que usan la misma métrica para encontrar el camino más corto hacia el nodo destino. De la misma manera generan confiabilidad, control de congestión y están libre de lazos internos al momento de la transmisión de información. Su rendimiento decae en la escalabilidad y balance de carga por la movilidad y dinamismo que los nodos realizan en la red. La implementación se puede realizar en Linux (Ubuntu), Windows, NS-2, NS-3, C++, Matlab, entre otros.

Tabla 1.

Comparación de los protocolos de enrutamiento Reactivos, Proactivos e Híbridos aplicables en redes móviles Ad-Hoc. Adaptada de: Revista Telemática, 2014.

Algoritmo	Tipo de protocolo	Métrica Utilizada	Libre de lazos	Escalabilidad	Confiabilidad	Balace de carga	Control de congestión	Throughput (Rendimiento)	Implementación
OLSR	Proactivo	Salto (camino más corto)	Si	No	Si	No	No	Disminuye con la movilidad de los nodos	Linux, Android, Manet manager, NS-3, NS-2
DSDV	Proactivo	Salto (camino más corto)	Si	No	Si	No	No	Disminuye con la movilidad de los nodos	NS-2, NS-3, Linux (Ubuntu)
CGSR	Proactivo	Basados en tablas de enrutamiento	Si	No	No	No	No	Periódica en el intercambio de información de enrutamiento	Linux
BATMAN	Proactivo	Transmisión de calidad de paquetes	Si	No	Si	No	Si	Disminuye con la movilidad de los nodos	Linux (Ubuntu), Android, NS-2, NS-3
DREAM	Proactivo	Basados en tablas de enrutamiento de baja demanda	Si	Si	No	No	No	Mantiene las rutas de forma continua	Linux
AODV	Reactivo	Salto (camino más corto y mas largo)	Si	No	Si	No	Si	Disminuye con la movilidad de los nodos	NS-2, NS-3, Linux (Ubuntu), C++, Matlab
DSR	Reactivo	Salto (camino más corto)	Si	No	Si	No	Si	Disminuye con la movilidad de los nodos	NS-2, NS-3, Linux (Ubuntu), Windows
TORA	Reactivo	Salto (camino más corto y mas largo)	No	Si	Si	No	No	Mejor en comparación con DSDV	Hardware, NS-2
LQSR	Reactivo	Tiempo de transmisión esperado acumulado ponderado (WCETT)	No	Si	No	Si	No	Aumenta cuando no interrumpe los enlaces entre nodos	NS-3, Linux
ZRP	Híbrido	Salto/zonas-camino más corto)	Si	Si	Si	Si	Si	Depende de la pérdida de paquetes	NS-2, Linux
ZHLS	Híbrido	Construcción de dos tablas de enrutamiento	No	No	No	Si	Si	Disminuye por ataques de seguridad	Linux
CBRP	Híbrido	Ruta más corta	No	Si	No	No	No	El tamaño de paquete aumenta en proporción de la longitud de las rutas	NS-3, Linux
DDR	Híbrido	Salto (varios caminos cuando se altera el destino)	Si	No	No	No	Si	Disminuye hasta encontrar el mayor número de rutas válidas.	Linux

## **1.8.- Calidad de servicio (QOS) en las redes Ad-Hoc**

La calidad de servicio en las redes Ad-Hoc, otorga un buen servicio a todos los usuarios que manejan las diferentes aplicaciones en tiempo real, algunas de ellas necesitan más ancho de banda, puesto que ocasionan congestión en la red y por ello pérdida de paquetes de datos.

Para medir la calidad de servicio se toma en cuenta parámetros como: canal eficaz, ancho de banda, pérdida de paquetes y latencia. Si se mide primeramente estos puntos con la calidad de servicio, se puede excluir el tráfico, dando mayor eficiencia al momento de transmitir la información. Por medio de QoS se puede encontrar el camino correcto hacia el destino conjuntamente con los paquetes de datos seguros y confiables, de ésta forma se reduce la pérdida de información y latencia. Debido al dinamismo de su topología estas redes modifican constantemente el ancho de banda, enlaces y el retardo.

### **1.8.1 Parámetros de la calidad de servicio**

Cuando se garantiza la calidad de servicio, el usuario final puede transmitir información de forma adecuada en tiempos apropiados, los parámetros de la calidad de servicio son:

#### **1.8.1.1.- Pérdida de paquetes**

La pérdida de información o paquetes se da al momento que estos no arriban al destino final a causa de la congestión que existe en el camino de comunicación.

#### **1.8.1.2.- Retardo o Latencia**

El retardo se mide en las unidades de tiempo, es decir se mide el tiempo total en que un paquete parte de un nodo origen hasta llegar al destino final.

#### **1.8.1.3.- Canal eficaz o throughput**

Es toda la proporción de datos que fueron recibidos por el destino en tiempos determinados a través de un medio físico o lógico de la red.

### **1.8.2.1.- Señalización de reserva para la calidad de servicio**

La señalización está encargada de procesar la liberación, reserva de los recursos en la red y establecimiento de los flujos del tráfico en la que se encuentra. Estos mecanismos incorporan toda la información de control dentro de los paquetes in-band signalling, y los que procesan mensajes de control out-of band signalling. El protocolo más usado es RSVP, que maneja uno de los mecanismos de señalización más alargados y permite reservar de extremo a extremo los recursos de la red para tráficos multicast y unicast.

### **1.8.2.2.- Encaminamiento con calidad de servicio**

En el encaminamiento de la red, se debe tomar en cuenta los protocolos de enrutamiento como ya se mencionó anteriormente, estos protocolos deben establecer algunas rutas que puedan cumplir con las características de QoS tales como: canal eficaz, minimizar la pérdida de paquetes y que el retardo al momento de entregar la información sea mínima.

## **1.9.- Seguridades en Redes Ad-Hoc**

La seguridad en este tipo de redes es de gran importancia ya que existen muchas posibilidades de ataques internos por el hecho de no poseer un sistema central o fijo de comunicación. Uno de los problemas de seguridad es la gestión y distribución de claves para la autenticación dentro de la red. A continuación se presenta algunos casos de vulnerabilidades existentes:

### **1.9.1.- Vulnerabilidades en redes Ad-Hoc**

Toda red Ad-Hoc es realmente vulnerable a varios ataques por cibernautas, en las cuales se debe distinguir los diferentes tipos de nodos que intentan entrar sin autorización en estas redes:

- Nodo malicioso
- Nodo comprometido
- Nodo selfish

#### **1.9.1.1.- Nodo malicioso**

Es un atacante interno que está comportándose de manera malintencionada, puede ser autenticado por los mismos nodos debido a que no existe información de datos criptográficos.



### **1.9.1.3.- Nodo comprometido**

Es un atacante central, y su comportamiento es malintencionado, puede ser autenticado por los mismos nodos y llegar a ser legítimo, de esta manera, puede adquirir respaldo y confianza de los demás nodos.

### **1.9.1.3.- Nodo selfish**

Este nodo es conocido por las características de egoísmo o acaparador, y puede llegar a prohibir los suministros de los servicios que pertenecen a los otros nodos con el objetivo de conservar los recursos de los dispositivos vecinos.

## **1.9.2.- Atributos de seguridad**

Los atributos de seguridad en las redes Ad-Hoc son los siguientes:

- Disponibilidad
- Autenticación
- Confidencialidad
- Integridad
- Autorización

### **1.9.2.1.- Disponibilidad**

Los servicios brindados por un nodo deben ser independientes de ataques. Los nodos deben permanecer disponibles para la transmisión de datos en cualquier instante.

### **1.9.2.2.- Autenticación**

Se confirma si la información que se está transmitiendo entre nodos es auténtica y no falsa, es decir, cada uno debe indicar de alguna manera que sus credenciales son válidas. Si no existe autenticación cualquier ataque puede pasarse como un nodo de tal forma, éste nodo puede acceder a datos clasificados y sensibles.

### **1.9.2.3.- Confidencialidad**

Es la forma de asegurar que un atacante no ingrese a la red y que no sea capaz de acceder a la información interna al momento que se distribuyen los datos.

#### **1.9.2.4.- Integridad**

Se garantiza que el paquete de información que se está entregando no haya sido modificado mientras se producía la transmisión, y garantizar lo que fue recibido sea realmente lo enviado.

#### **1.9.2.5.- Autorización**

Se establecen reglas para definir la tarea de cada uno de los nodos de la red.

### **1.10.- Ataques en las redes Ad-Hoc**

Existen dos tipos de ataques dentro de las redes Ad-Hoc, una de ellas es la existencia de ataque a los diferentes mecanismos básicos en la red como: el encaminamiento, ataque sobre la administración de claves y el mecanismo de seguridad.

Estos ataques a su vez se separan en:

- Ataques pasivos-
- Ataques activos

#### **1.10.1.- Ataques pasivos**

Estos ataques involucran sólo (eaves-dropping-espionajes) de los datos, como por ejemplo espionaje de tráfico, espionaje de canales secretos y sniffing de las claves.

#### **1.10.2.- Ataques activos**

Lo ataque activos son las acciones que se realizan durante la comunicación de datos entre los nodos, estos ataques pueden modificar, eliminar información y replicar los paquetes. Estas acciones constantemente varían los comportamientos de cada uno de los protocolos, por lo que, ésta información puede ser enviado a los atacantes activos y pasivos.

#### **1.10.3.- Protocolos en encaminamiento seguros**

Los protocolos de encaminamiento brindan seguridad en las redes Ad-Hoc para evitar robo de información, manipulación externa por otros usuarios, actualizaciones de rutas provocadas, consumo de recursos y ancho de banda. Estos diversos tipos de ataque se nombran a continuación:

### **1.10.3.1.- Ataque al protocolo de encaminamiento**

El objetivo de este ataque al encaminamiento es interrumpir toda la operación de la red.

### **1.10.3.2.- Desbordamiento de la tabla de encaminamiento**

Un nodo que esta por atacar, anuncia rutas a otros nodos no existentes autorizados por la red, su objetivo es causar desbordamiento de todas las tablas de encaminamiento evitando la creación de caminos para los nodos autorizados. Los protocolos más vulnerables son los proactivos.

### **1.10.3.3.- Envenenamiento de la tabla de encaminamiento**

El problema se presenta cuando los nodos que se encuentran comprometidos en sus redes empiezan a enviar algunas actualizaciones de muchas rutas ficticias, modificando cada uno de los paquetes de actualización.

### **1.10.3.4.- Replicación de paquetes**

Al momento que un nodo atacante empieza a replicar información se consume muchos recursos como: ancho de banda y la potencia de las baterías.

### **1.10.3.5.- Sleep Deprivation (Privación de sueño)**

Mientras los dispositivos se encuentran en stand-by guardando energía en sus baterías, el atacante aprovecha para consumirlas mediante una solicitud de reenvío de paquetes que son innecesarios o por medio de solicitud de rutas.

### **1.10.3.6.- Divulgación de ubicación**

Esto ocurre cuando el atacante revela información sobre la localización de la estructura o localización de los nodos en la red. Otro tipo de ataque es la denegación de servicios, esta consiste en evitar que un usuario tenga acceso a los recursos de la red para sustituir su identidad, de esta forma, obtienen privilegios de un nodo que está en modo autorizado.

## **CAPÍTULO II CONSIDERACIONES Y LIMITACIONES PARA EL ANÁLISIS DE LOS PROTOCOLOS DE ENRUTAMIENTO.**

Existen diversos protocolos que se aplican en las redes Ad-Hoc como se detalla en el capítulo 1, con el fin de brindar un principio teórico que soporte la investigación desarrollada, para posteriormente escoger específicamente a los principales algoritmos más sobresalientes de cada uno de los protocolos de enrutamiento, estos son: reactivos: (AODV y DSR), y para los proactivos: (DSDV), con el fin de analizar al algoritmo que mejor se desempeña frente a los diversos paradigmas que se puede presentar dentro de una estructura de red móvil. No se considera para la comparación al protocolo híbrido porque sus algoritmos poco usados actualmente son una combinación de los protocolos de enrutamiento reactivos-proactivos, y no se encuentran en el programa de simulación, lo que dificulta realizar un análisis de este protocolo. Seguidamente se expone el resultado alcanzado que dará acceso a un análisis a partir de los datos obtenidos.

Para generar el resultado de la calidad de servicio de los protocolos de enrutamiento aplicados en este proyecto, la ITU-T (Unión Internacional de Telecomunicaciones-Telecomunicaciones), recomienda utilizar sus identificadores que permiten comparar el tiempo de demora, tiempo de espera y la pérdida de paquetes generados en cada escenario. El identificador que se emplea para este caso de red móvil es la clase tipo 4, por lo que, en esta simulación se realizará específicamente transferencia de archivos entre los nodos. Cabe recalcar que existen diversos tipos de identificadores para las diferentes aplicaciones que hay en una red Ad-Hoc como son; videoconferencia, difusión de audio y sonido, señalización, navegación, entre otros. Para el desarrollo de esta tesis se empleó exclusivamente (FTP) ya que se encuentra por default dentro del programa de simulación NS-2 y sus herramientas; Trace Graph 2.02 y NSG-2 facilitan su implementación.

### 2.1.- Parámetros de calidad de funcionamiento de red.

Estos parámetros de funcionamiento son importantes para analizar el comportamiento de las topologías que existen dentro de una red. Nos brindan soporte y apoyo para gestionar problemas que ocurren en la transferencia de información, estos parámetros son: IPER (IP Packet Error Ratio); menciona la tasa de paquetes que contienen errores, IPLR (IP Packet Loss Ratio); hace referencia a la tasa que contiene pérdida de paquetes, IPDV (IP Packet Delay Variation); se refiere al tiempo de espera en recibir o enviar un paquete, IPTD (IP Packet Transfer Delay); menciona el tiempo en que un paquete se demora en pasar de un punto al otro por la red. Los parámetros mencionados sirven para evaluar los tiempos de demora, tiempos de espera y pérdida de paquetes. Pertenecen a las normas Y.1540 y Y.1541, que son otorgados por la ITU-T (Unión Internacional de Telecomunicaciones-Telecomunicaciones).

Las clases de calidad de servicio como se muestra en la figura 9, nos ayudan a identificar en que categoría se encuentran las aplicaciones que estamos empleando. Para poder generar un análisis de QoS, se utilizan diferentes indicadores que ayudan a seleccionar el tipo de clase en la que se encuentra el modelo de aplicación. Como habíamos mencionado anteriormente, al utilizar el medio de transmisión (FTP), estos tres diseños de red se encuentran en el grupo 4, ya que son servicios que soportan tiempos de pérdidas y retardos en la red.

La distribución de las clases se detalla a continuación:

- **Clase 0-1:** son las aplicaciones en tiempos reales, estos sensibles para una interacción alta y sensible al retardo, ejemplo: videoconferencia, difusión de audio y VoIP.
- **Clase 2-3:** son las aplicaciones de datos que ejecutan transacción interactiva como por ejemplo: señalización y navegación.
- **Clase 4:** estas aplicaciones son las que soportan pérdidas pero no hay problemas con el retardo como por ejemplo: transferencia de archivos y videostreaming.

En la figura 9, se observa los parámetros de calidad de funcionamiento y las clases de calidad de servicio:

Tabla 2.

Parámetros de la calidad de funcionamiento en la red. (Nota: La “U” significa “no especificado” o “sin límites”), Tomado de: Satellite Networking Principles and Protocols, 2005.

Parámetro de Calidad de funcionamiento de red.	Clases de QoS					
	Clase 0	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5 no especificada
IPTD	100 ms	400 ms	100 ms	400 ms	1 s	U
IPDV	50 ms	50 ms	U	U	U	U
IPLR	1 x	1 x	1 x	1 x	1 x	U
	$10^{-3}$ (Nota 4)	$10^{-3}$ (Nota 4)	$10^{-3}$	$10^{-3}$	$10^{-3}$	
IPER	$1 \times 10^{-4}$					U

## 2.2.- Características de los escenarios de simulación con los protocolos de enrutamiento para redes móviles Ad-Hoc.

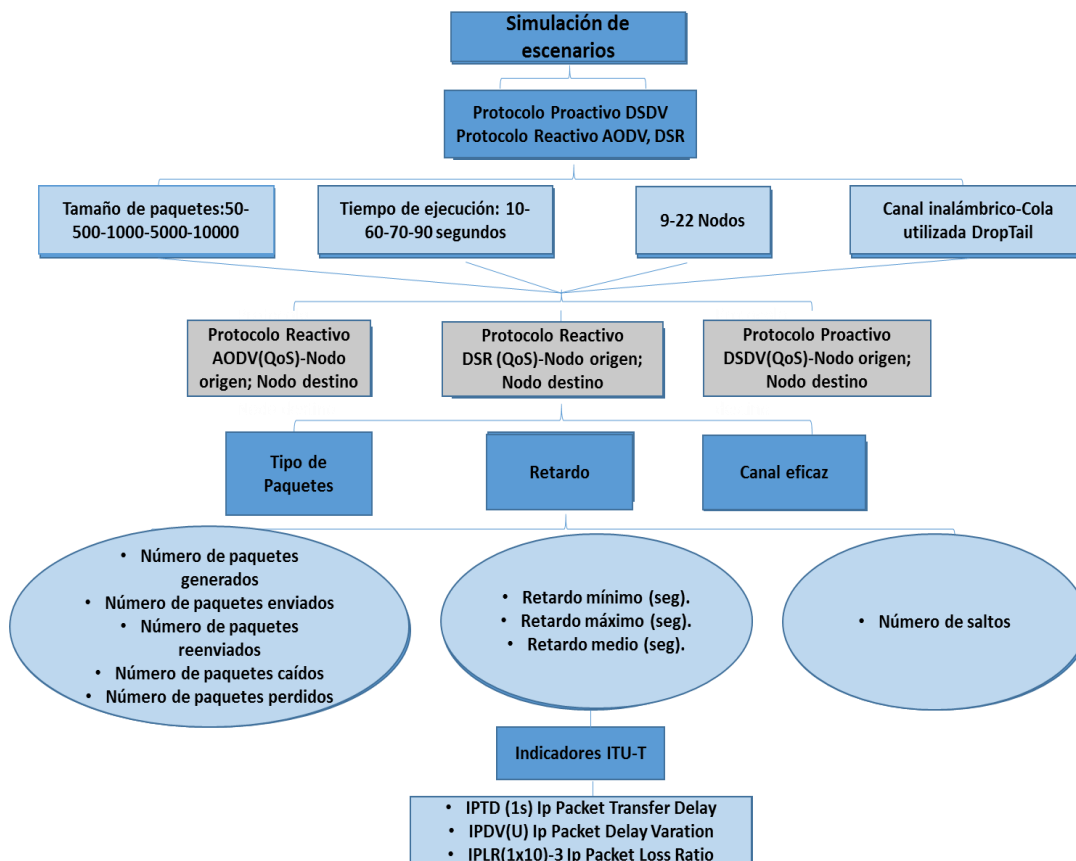


Figura 8. Características de los escenarios de simulación con protocolos de enrutamiento

Cómo se muestra en la figura 8, existen 12 escenarios que estarán encabezados por los protocolos de enrutamiento reactivos (AODV-DSR) y proactivos (DSDV). Los ambientes de simulación están estructurados entre 9 y 22 nodos móviles inalámbricos. El tiempo que duran estos escenarios están entre: 10, 70 y 90 (segundos) de modo que, se pueda observar el dinamismo de los nodos, enlaces, pérdida y envío de paquetes. La comunicación de información será siempre entre el nodo origen-nodo destino, con la ayuda de sus nodos vecinos o directamente entre ellos en caso de no necesitar saltos

de ruta. Ésta transmisión de paquetes será por medio del agente (TCP-Protocolo de Control de Transmisión) y será el encargado de asegurar, administrar que los datos lleguen sin errores al destino. El nodo destino está programado para recibir información por medio del agente (TCP-Sink), el cual se encarga de reconocer y recibir las tramas y a su vez realizar el envío de los paquetes siempre y cuando esté enlazados a algún nodo vecino. En algunos casos la distancia entre los nodos esta fuera del rango de cobertura, por lo tanto no se podrá realizar la comunicación desde el nodo origen al destino. El modelo de las colas utilizado será Droptail (algoritmo de gestión de colas), que vendría a ser una cola FIFO para suprimir todos los paquetes que pueden sobrepasar el tamaño del búffer (espacio de memoria) en cada cola. Para obtener resultados diferentes la limitación de la cola FIFO es: 20, 30, 40, 50 y 70 paquetes, con el fin de efectuar un análisis preciso de los mismos. El tipo de antena es omnidireccional la cual está encargada de ampliar las coberturas de la red inalámbrica 802.11b/g y que a su vez mejora la intensidad de la señal para todas las direcciones.

Posteriormente en el siguiente capítulo, se detallará los resultados obtenidos cómo: tipo de paquetes (generados, enviados, reenviados, caídos, perdidos), el tipo de retardo en segundos (mínimo, máximo, medio), y el canal eficaz (número de saltos).

Para efectuar el análisis final de calidad de servicio, se usarán los indicadores de las normas ITU-T, el cual permitirá definir el mejor protocolo a utilizarse en una red Ad-Hoc. Estos indicadores son: (IPTD-Ip Packet Transfer Delay menor a 1 segundo); (IPDV-Ip Packet Delay Variation no especificado o sin límites) y el indicador (IPLR-Ip Packet Loss Ratio debe ser menor a  $(1 \times 10)^{-3}$ ).



### 2.3.- Diseño y funcionamiento del escenario de simulación 1 con protocolo de enrutamiento reactivo AODV.

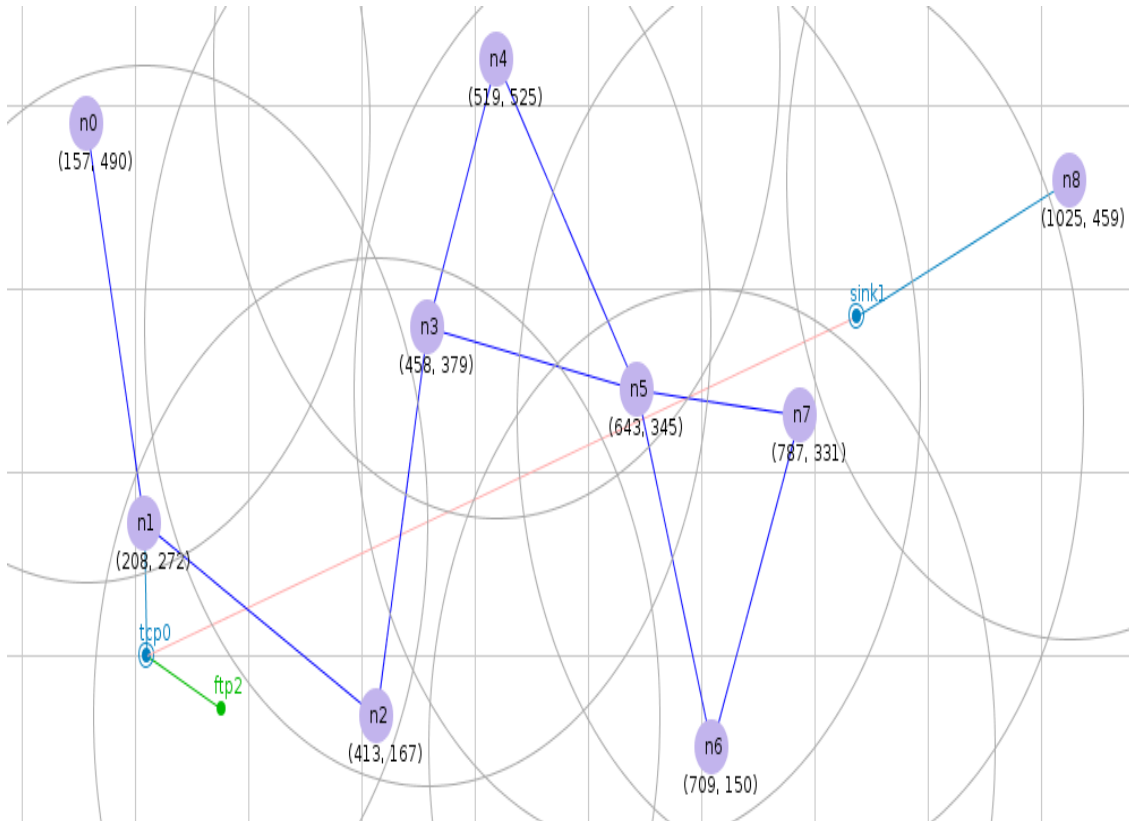


Figura 9. Diseño del escenario 1 con 9 nodos Protocolo de enrutamiento reactivo - AODV.

Cómo se muestra en el diseño de la figura 9, la programación está diseñada para que al inicio de la simulación los nodos principales que son 1 y 8, se puedan comunicar durante toda su ejecución ya sea por nodos vecinos o entre el nodo 1 y 8. La dimensión para éste escenario son los rangos en (x)=1125 y para (y)=640 (metros).

La transmisión de datos en ésta simulación se dará entre el nodo origen 1 y el nodo destino 8; al segundo 0.11800 el nodo 6 y el nodo 3 empiezan a moverse, mientras que al primer segundo todos los nodos actualizan las tablas y a descubrir las nuevas rutas, distancias y número de saltos de los mismos hasta encontrar el nodo destino y así empezar a enviar los datos. El inconveniente

con éste protocolo, es la reacción lenta para descubrir nuevas rutas cuándo se generan distintos cambios dentro de esta topología de red, como se muestra en la figura 10.

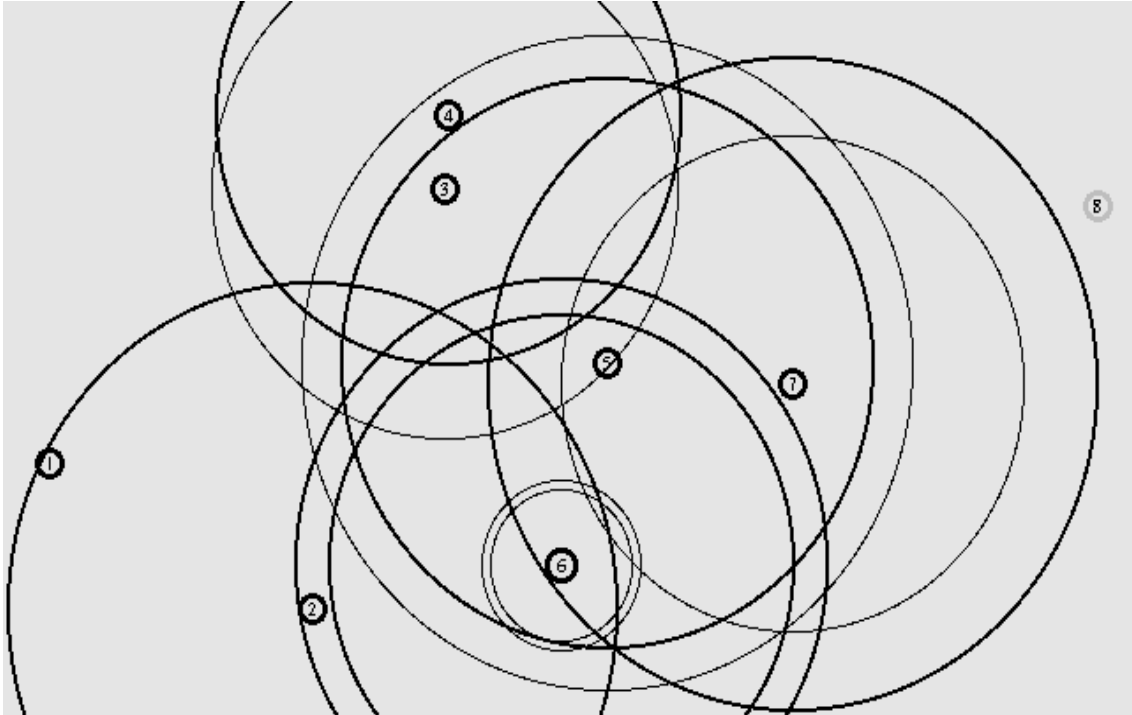


Figura 10. Actualización de nodos y tablas de rutas

Al segundo (3) el nodo 8, empieza acercarse al rango de cobertura de los demás nodos vecinos y a su vez, empieza a recibir señales de las nuevas rutas y número de saltos para crear un nuevo enlace. Cuándo el nodo 8 está dentro de un rango accesible, se realiza nuevamente una actualización de rutas de manera que, éste nodo destino (8), pueda encontrar los caminos más cortos hasta llegar a una distancia que le permita comunicarse con el nodo origen (1). Cómo se muestra en la figura 11.

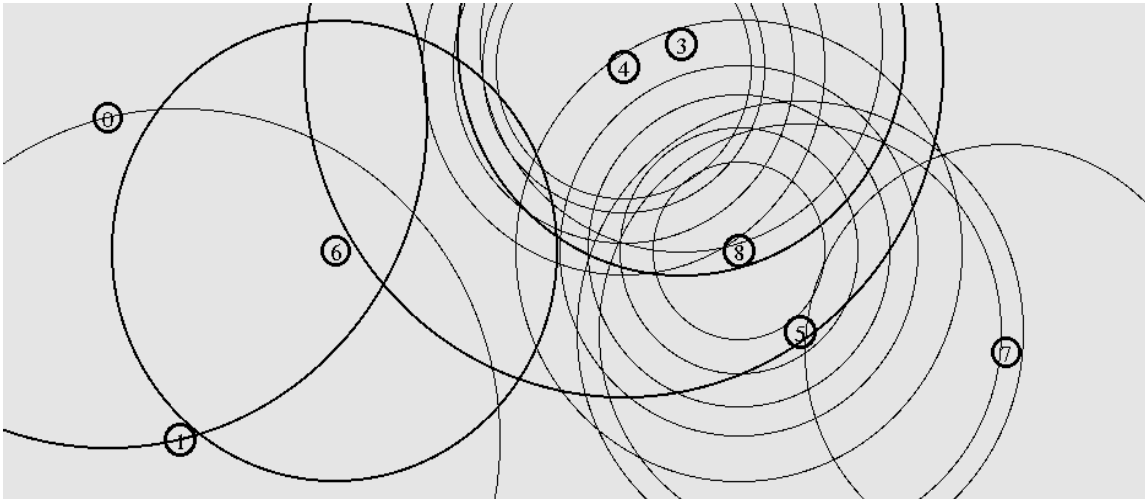


Figura 11. Nodo 8 dentro de cobertura con protocolo AODV.

Cuándo se encuentra actualizada las tablas, el nodo 1 indica el mejor camino para empezar la transmisión. Al segundo 7 se realiza el envío y recepción de las tramas de datos entre el nodo 1 y 8 por medio del nodo vecino más cercano que es el nodo 6, cómo se muestra en la figura 12.

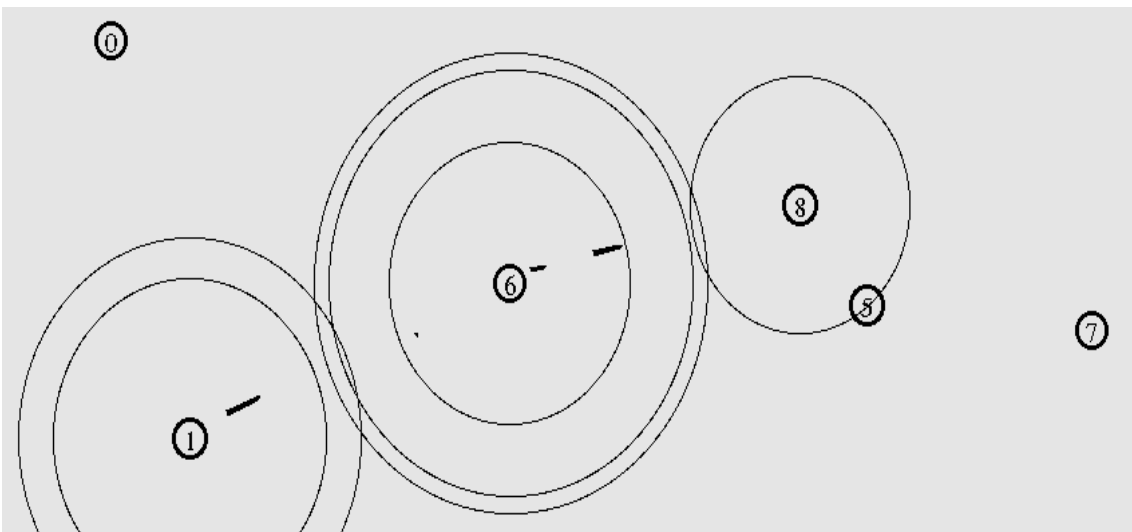


Figura 12. Envío de tramas entre los nodos 1-6-8 con protocolo AODV

El nodo 6 al segundo 7.123906 inicia un nuevo rumbo alejándose del nodo origen y destino. En consecuencia, la pérdida de tramas de datos empieza a incrementarse mientras este nodo toma otra dirección. Durante el tiempo en que, el nodo 8 se aleja del rango de cobertura por el cual se comunicaba con el nodo vecino, éste comienza a realizar nuevas peticiones de búsqueda para

encontrar otros nodos y así, establecer rutas de manera que se pueda reiniciar el envío y recepción de datos hacia el nodo origen, como se muestra en la figura 13.

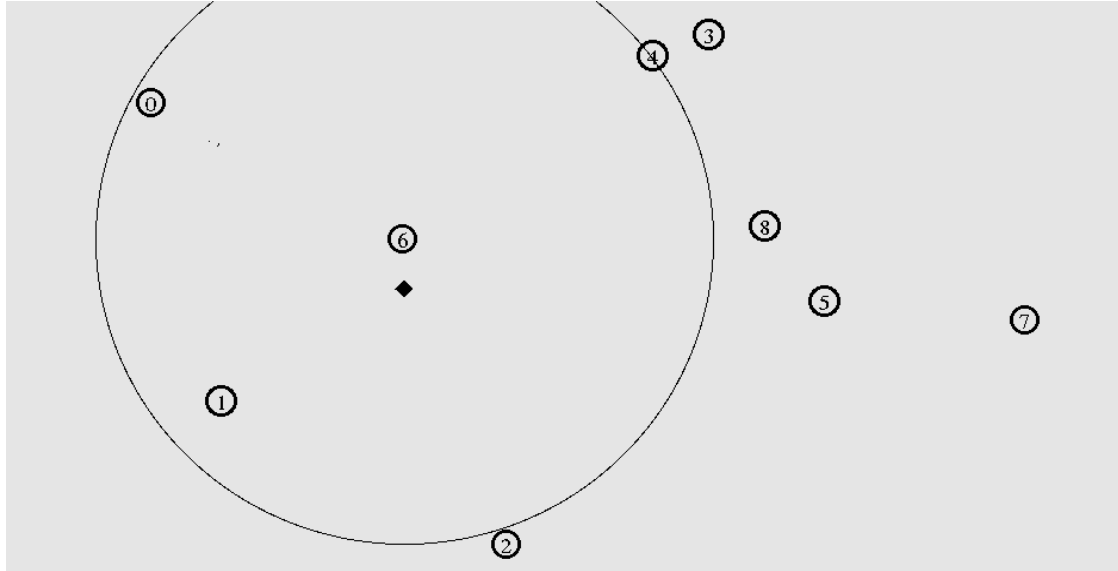


Figura 13. Pérdida de las tramas con protocolo reactivo AODV.

Los nodos que se encontraban conectados anteriormente, toman direcciones diferentes de modo que, rompen los enlaces mediante el cual realizaban la transmisión de paquetes. En ese momento aparecen nuevos nodos para realizar una nueva actualización de las tablas de datos, permitiendo a sus módulos de cabecera renovar y activar la comunicación entre los nodos vecinos, es así; que al segundo 7.640660 el nodo 6 detecta al nodo 4 como el más cercano al nodo destino. Por tanto, el nodo 4 empieza a crear nuevas rutas entre estos nodos intermedios para establecer la transmisión de los datos hacia el nodo origen, cómo se muestra en la figura 14.

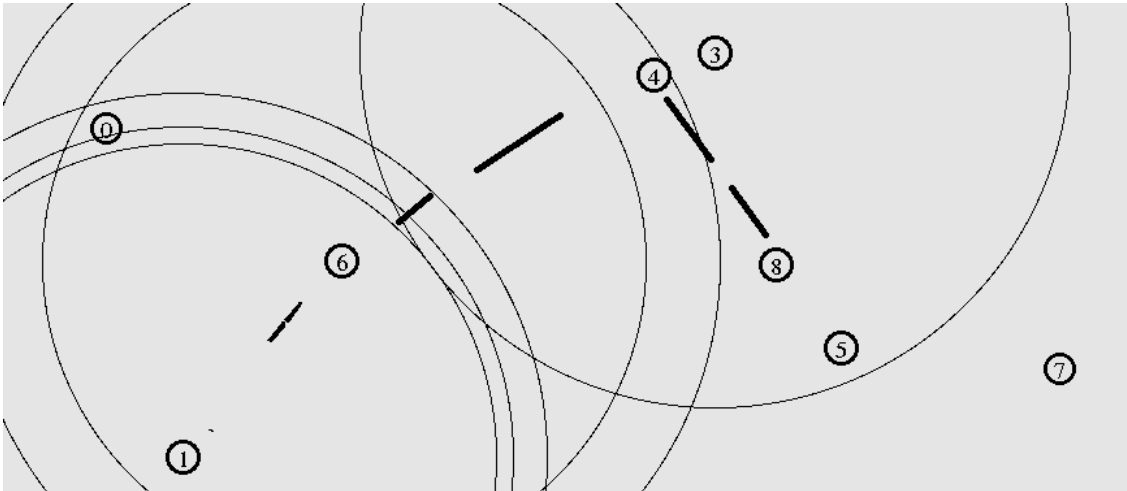


Figura 14. Envío de tramas entre los nodos 1-6-4-8 con protocolo reactivo AODV.

Al segundo 7.908185 el nodo 6, se aísla del nodo 4 mediante el cual compartía comunicación con los demás nodos, éste nodo se aleja debido a su constante cambio de rutas en la red. Por ésta razón, se empiezan a perder las tramas de datos y a dejar de enviar y recibir la información que fue generada por el nodo origen, cómo se muestra en la figura 15.

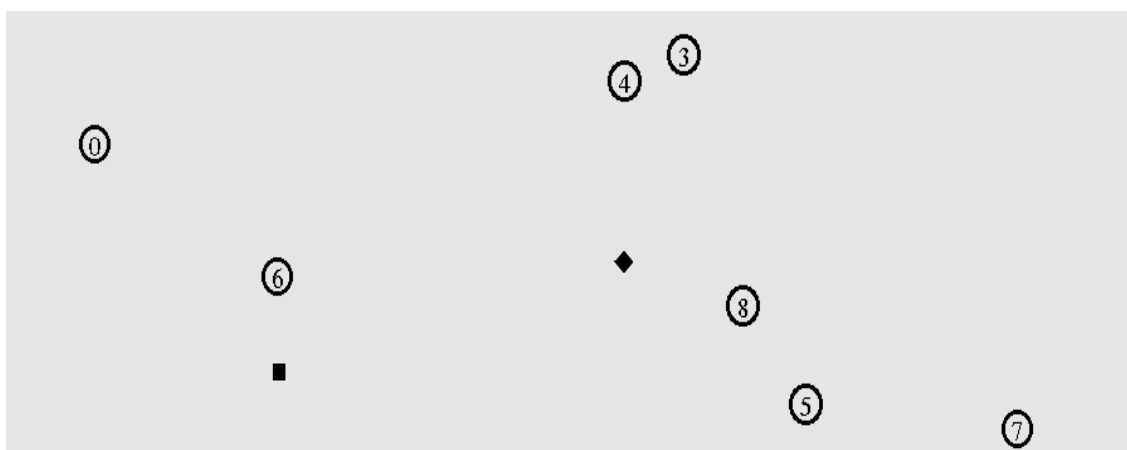


Figura 15. Pérdida de información con protocolo reactivo AODV.

Segundos más tarde debido al dinamismo de la red, aparece un nuevo nodo vecino 2, cómo se muestra en la figura 16. Al segundo 8, éste nodo entra lentamente a la cobertura de los nodos origen y destino en dónde compartían información con los nodos 4, 6, 1 y 8. Cuando el nodo reciente se encuentra a una distancia cercana del origen y destino, se genera la actualización de las tablas, nodos y número de saltos para crear nuevas rutas entre ellos. En consecuencia, al segundo 8.5 se establece el encaminamiento entre el nodo origen 1, nodo vecino 2 y nodo destino 8. De esta forma, finaliza la transmisión de paquetes al segundo 10.

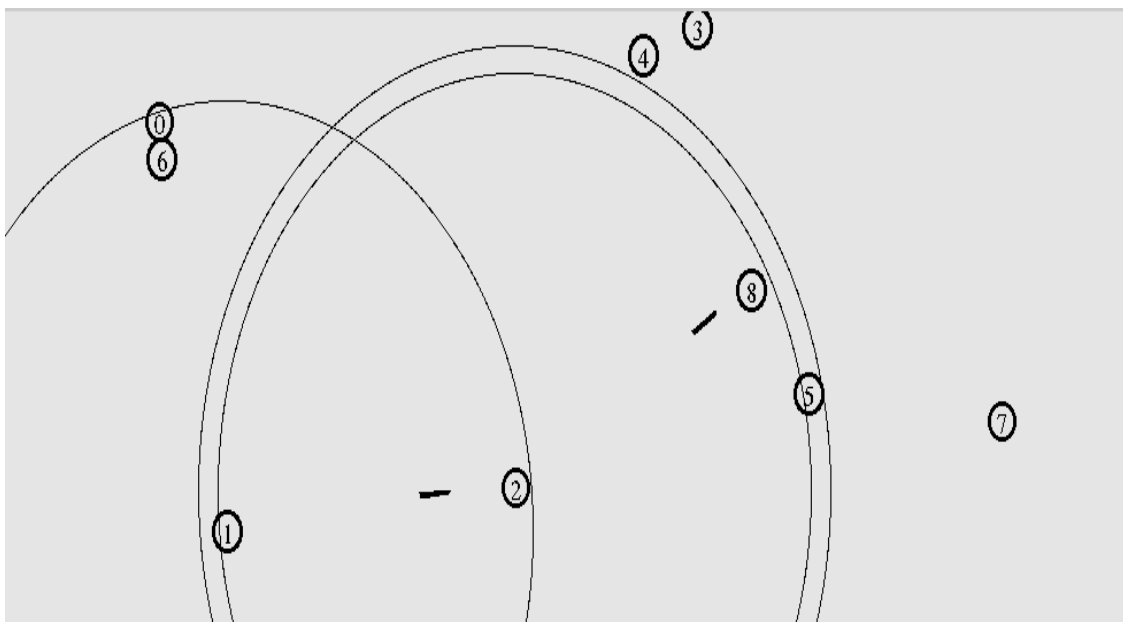


Figura 16. Finalización del envío y recepción de tramas con protocolo reactivo AODV.

**2.4.- Diseño y funcionamiento de los escenarios 2, 3, 4 con protocolo de enrutamiento reactivo AODV y 5, 6, 7 con protocolo de enrutamiento proactivo DSDV.**

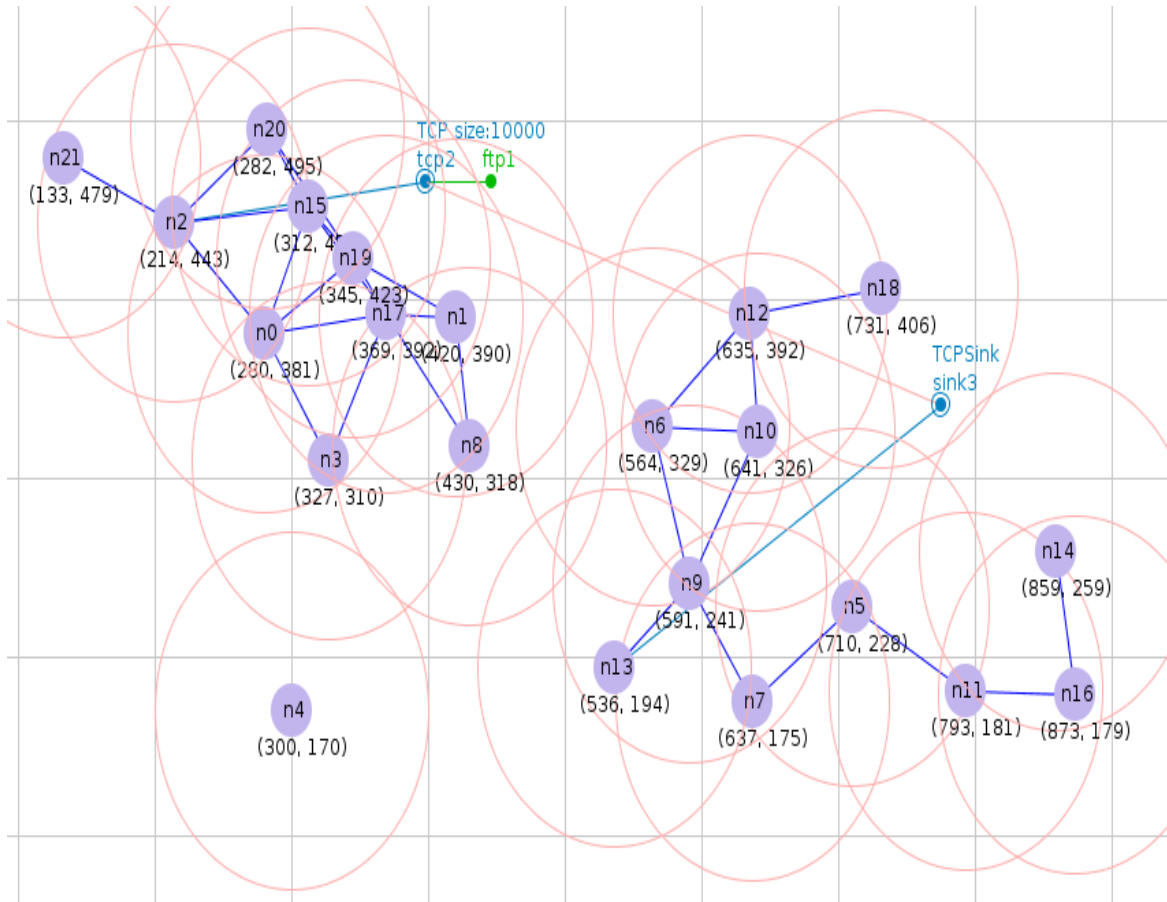


Figura 17. Diseño del escenario 2, 3 y 4 con 22 nodos mediante el Protocolo de enrutamiento reactivo - AODV.

La figura 17 representa la simulación para el escenario 2, 3, 4, 5, 6 y 7, el cual esta estructurado por 22 nodos móviles con diferentes variables en cada uno de ellos como:

- Ancho de banda
- Velocidad de conexión
- Tamaño máximo de celdas
- Tamaño de paquetes
- Tiempos de simulación

La comunicación para las seis simulaciones mencionadas anteriormente siempre será entre el nodo origen 2 y el nodo destino 13.

El comportamiento de los nodos es el mismo en los ambientes con protocolo reactivo AODV (2, 3) y proactivo (5, 6) dado que únicamente cuatro de sus nodos presentan movilidad con diferentes velocidades, mientras que los demás nodos se mantienen estáticos en todo el proceso de simulación. Para mantener comunicación entre el nodo origen y el nodo destino, en las simulaciones 2, 3, 5 y 6, se transmitirán los datos mediante los nodos intermedios 2, 15, 1 y 13 como se muestra en la figura 18.

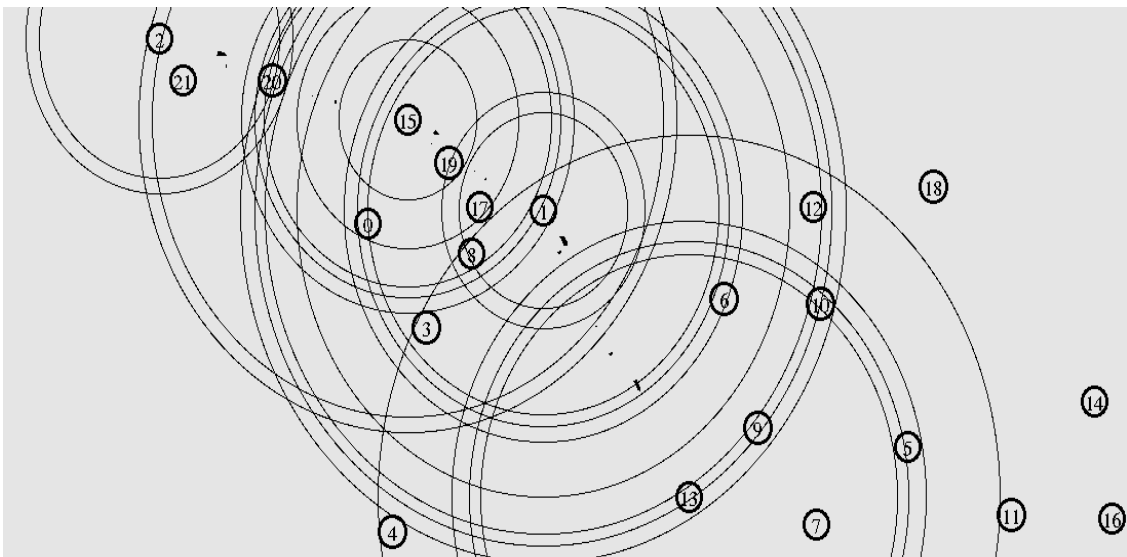


Figura 18. Movilidad de 4 nodos e intercambio de información con los protocolos AODV y DSDV en las simulaciones 2, 3, 5 y 6.



En el escenario 4 y 7, todos los nodos presentan dinamismo en general. Este procedimiento se realizó para distinguir el comportamiento de la topología de la red y obtener diferentes resultados que permitan realizar el análisis de calidad de servicio del mejor protocolo para las redes Ad-Hoc.

Cabe recalcar que la comunicación entre el nodo origen y destino se producirá a través de los nodos vecinos restantes, debido al comportamiento dinámico de los mismos que se genera en esta red.

## 2.5.- Diseño y funcionamiento del escenario de simulación 8 con protocolo de enrutamiento proactivo DSDV.

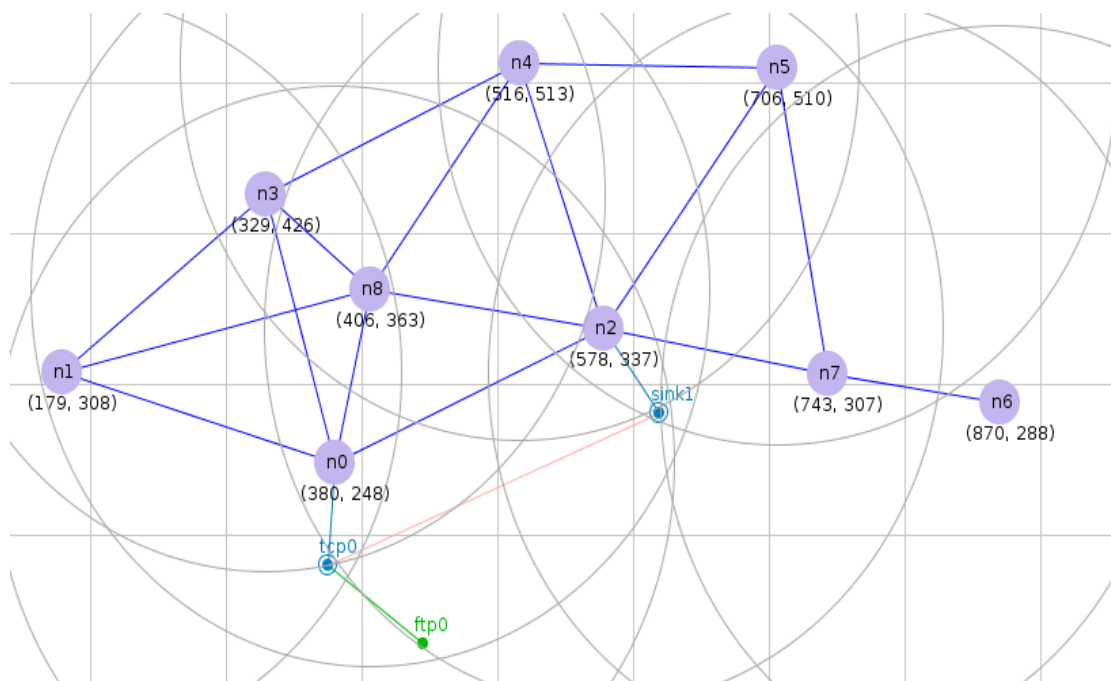


Figura 19. Diseño del escenario 3 con 9 nodos Protocolo de enrutamiento Proactivo DSDV.

Cómo se muestra en la figura 19, ésta red Ad-Hoc presenta 9 nodos en su estructura. Mediante el protocolo proactivo DSDV se generará el intercambio de información entre el nodo origen 0 y nodo destino 2. La dimensión para este escenario son los rangos en (x)=970 y para (y)=623 metros.

En ésta simulación, al segundo 0.1 se realiza una sola actualización de tablas de enrutamiento con el fin de, verificar el mejor camino por el cual el nodo 0 pueda transmitir la información al nodo 2. Al segundo 0.31 los nodos 0, 2 y 7,

presentan dinamismo de forma aleatoria en la red, alejándose cada vez con velocidades lentas del área de cobertura. El nodo 6 empieza realizar un acercamiento rápido a estos nodos, el cual, es reconocido como una posible ruta para establecer la comunicación con algún nodo vecino. Por encontrarse a distancias cercanas entre el nodo 2 con el nodo 0, éstos, no necesitan de nodos intermedios para establecer un enlace de comunicación, por ello, se empiezan a transmitir los paquetes directamente, como se puede observar en la figura 20.

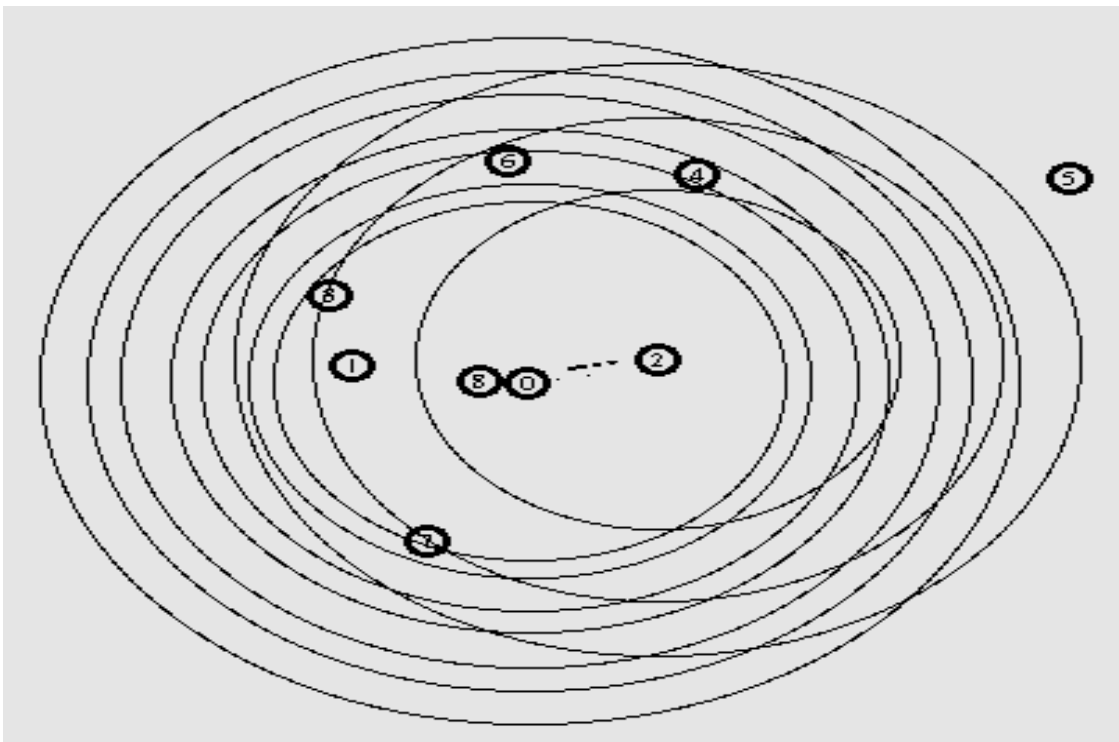


Figura 20. Actualización de las tablas con protocolo proactivo DSDV.

Al segundo 3, el nodo 2 realiza movimientos simétricos hacia delante y hacia atrás del nodo origen provocando pérdida de paquetes por su inestabilidad. Esto se realiza para demostrar, que el protocolo DSDV se demora en buscar nodos vecinos al momento de realizar actualización de sus tablas de enrutamiento. El retraso del reconocimiento de nuevas rutas, genera retardo y caída de paquetes cuando el nodo 2 se aleja más y va quedando fuera del alcance de comunicación, cómo se muestra en la figura 21.

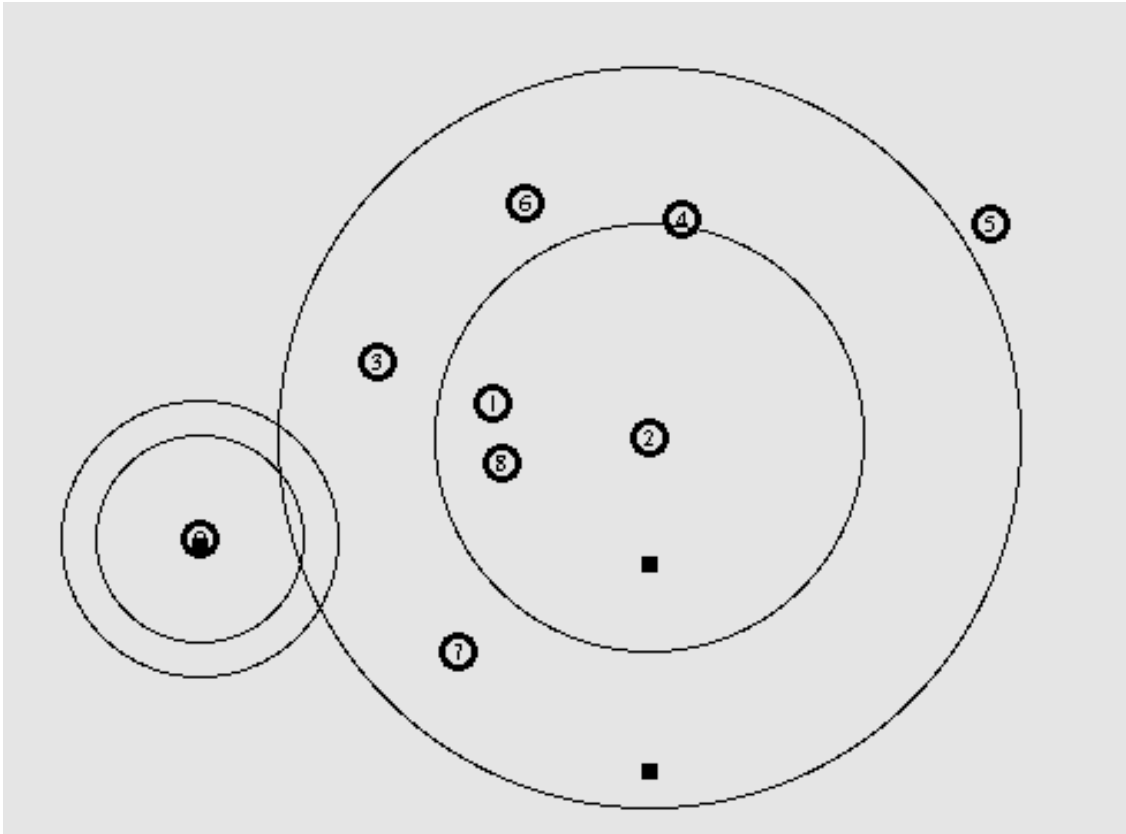


Figura 21. Pérdida de información entre los nodos origen y destino con protocolo proactivo DSDV.

El proceso de transmisión de datos queda pendiente hasta que estos dos nodos vuelvan a estar dentro del rango de señal. Al segundo 3.650000 el nodo 1 empieza a acercarse rápidamente a estos dos nodos que realizaban la transmisión de datos. Esta programación se realizó para verificar que por más cerca que se encuentren los nodos vecinos del nodo origen y del nodo destino; el protocolo DSDV, se demora en reconocer las nuevas rutas generando demasiada pérdida de información, como se muestra en la figura 22.

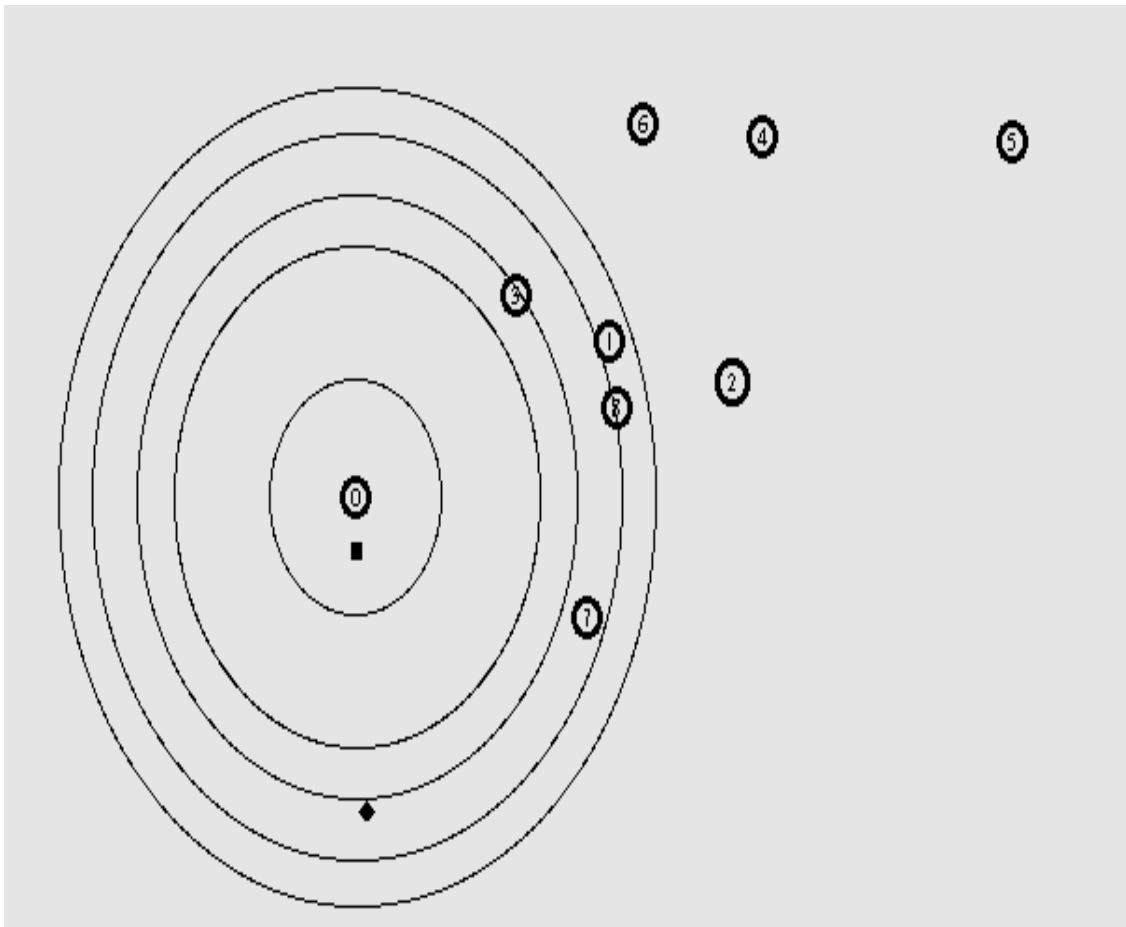


Figura 22. Pérdida de información entre los nodos origen y destino con protocolo proactivo DSDV.

Debido al dinamismo de la topología de esta red, al segundo 5.881031 el nodo destino empieza acercarse al nodo origen. Como en sus cabeceras se guardaron las rutas por la cual compartían anteriormente la información, estos nodos se pueden enlazar automáticamente, por lo tanto, al segundo 6.996309, renuevan la transmisión de datos de forma rápida, como se indica en la figura 23. Ésta simulación de red, finaliza al segundo 10.

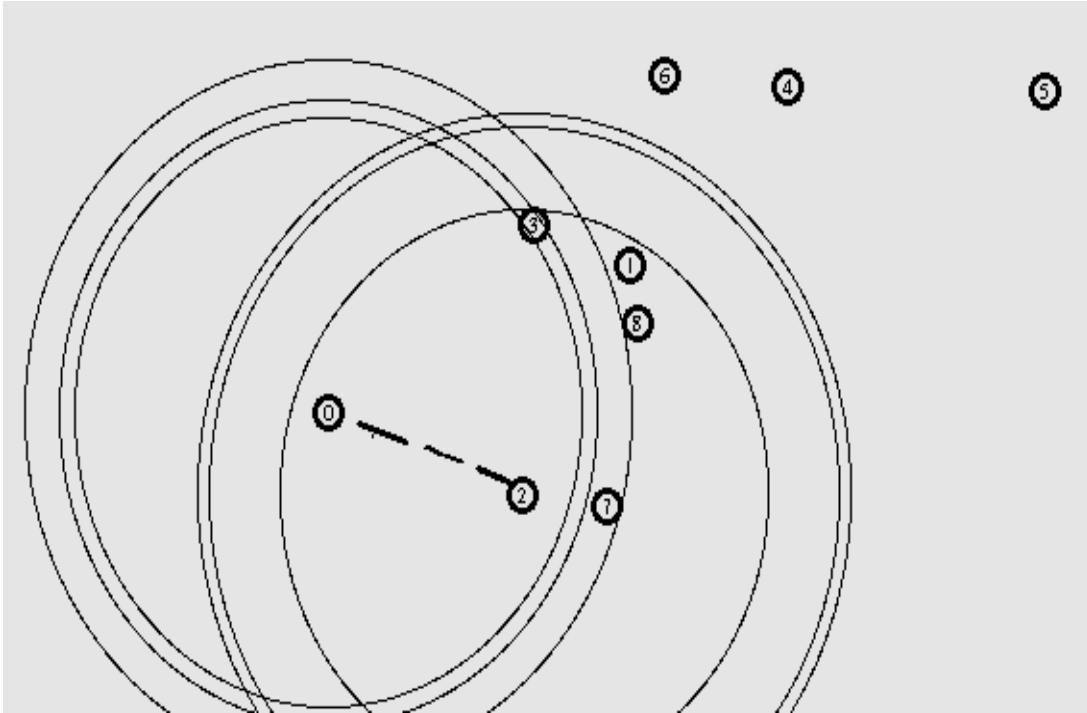


Figura 23. Actualización y renovación de envío de datos entre nodo origen y destino con protocolo proactivo DSDV.

## 2.6.- Diseño y funcionamiento de los escenarios 9, 10 y 11 con protocolo de enrutamiento reactivo DSR.

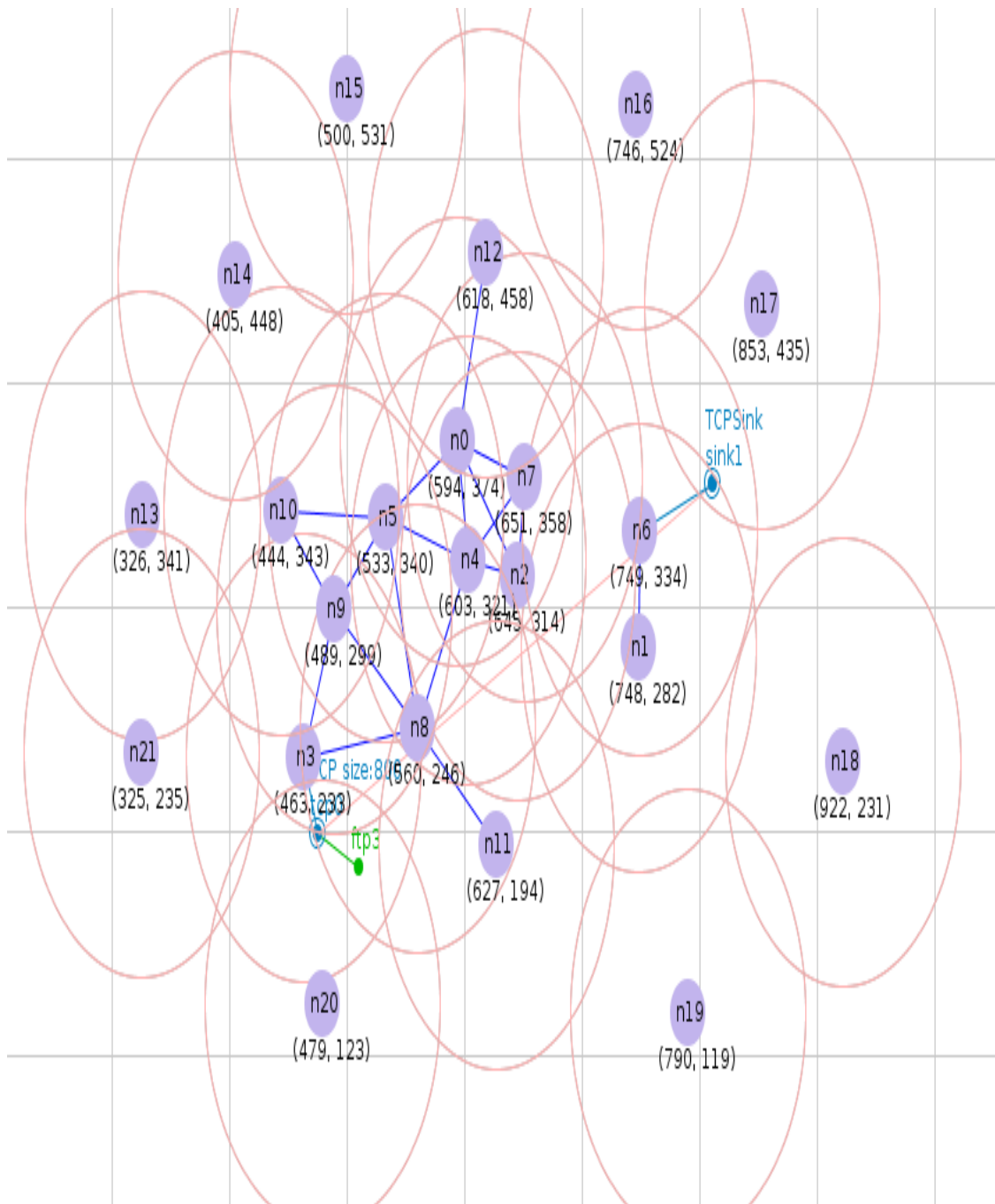


Figura 24. Escenarios 8, 9 y 10 con protocolo de enrutamiento reactivo DSR.

La figura 24 representa la simulación para el escenario 9, 10 y 11 el cual está estructurado por 22 nodos móviles con diferentes variables en cada uno de ellos como:

- Ancho de banda
- Velocidad de conexión
- Tamaño máximo de celdas
- Tamaño de paquetes
- Tiempos de simulación

La comunicación para las tres simulaciones mencionadas anteriormente siempre será entre el nodo origen 3 y el nodo destino 6.

El comportamiento de los nodos en estos ambientes no es el mismo, puesto que en la simulación 9 los nodos permanecen estáticos realizando la comunicación del nodo origen y el nodo destino por medio de los nodos vecinos 3, 8 y 9.

En el escenario 10, la transmisión de paquetes entre el nodo origen y destino se realizará a través de los nodos intermedios 3, 8 y 6, tomando en consideración que los únicos nodos en movimiento son el 8 y 4. Esta simulación presentará movimiento en diferentes velocidades en los dos nodos mencionados anteriormente, los cuales determinarán el comportamiento dentro de la red, como podemos observar en la figura 25.

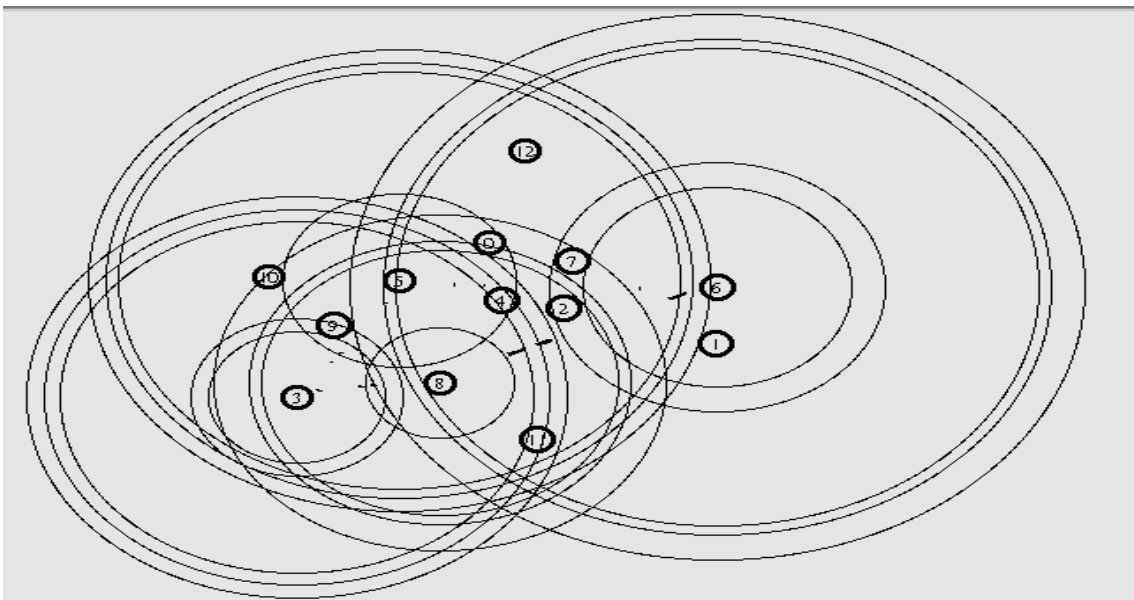


Figura 25. Transmisión de datos entre los nodos origen y destino

En el escenario 11 todos los nodos presentan dinamismo con diferentes comportamientos, con el fin de obtener otros resultados y de este modo poder comparar con el resto de simulaciones.

La comunicación del nodo destino y nodo origen se realizará a través de nodos vecinos, tomando en consideración que éstos se mueven en diversos tiempos, velocidades y distancias como se observa en la figura 26.

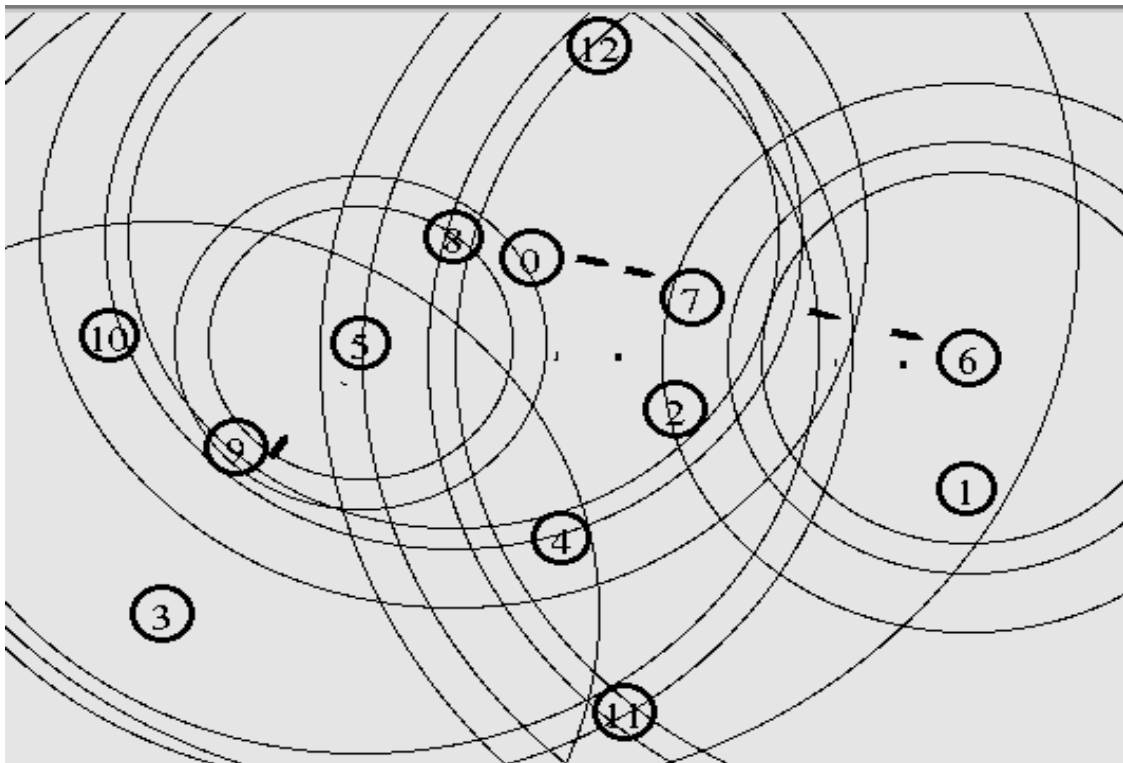


Figura 26. Dinamismo entre diversos nodos vecinos para la transmisión de datos entre el nodo origen y destino.



## 2.7.- Diseño y funcionamiento del escenario de simulación 12 con protocolo de enrutamiento reactivo DSR.

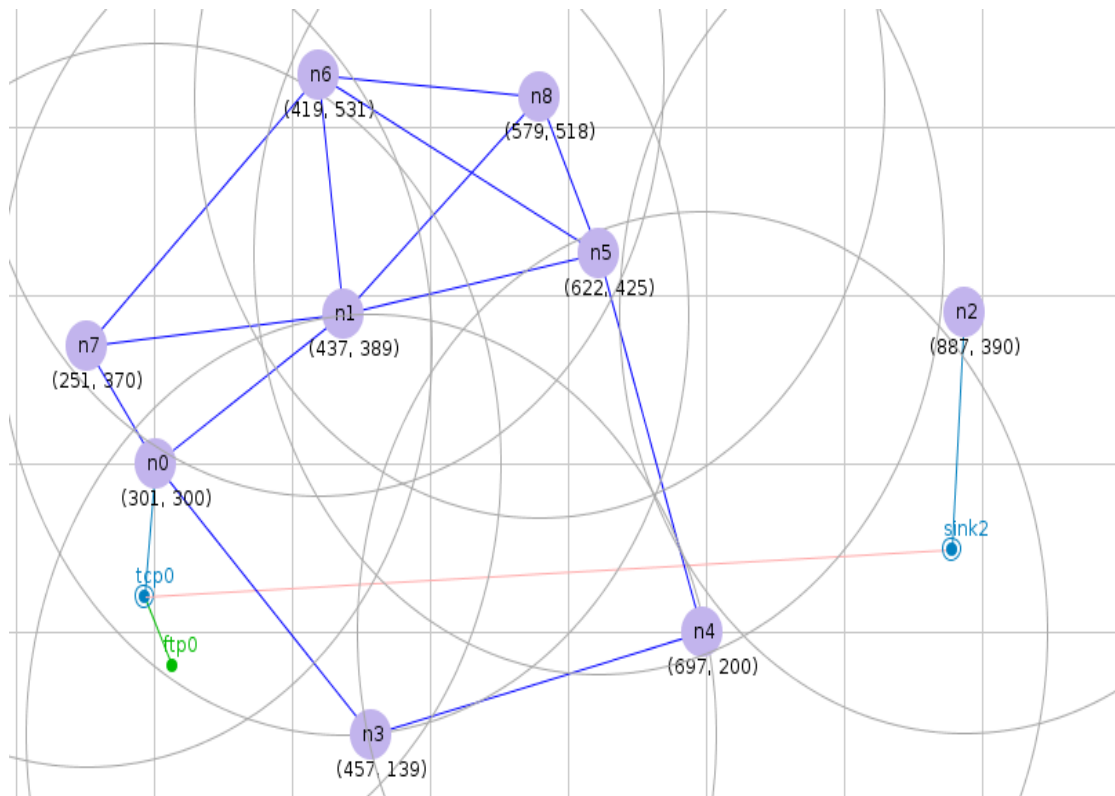


Figura 27. Diseño del escenario 2 con 9 nodos Protocolo de enrutamiento reactivo DSR

Cómo se muestra en la figura 27, ésta red Ad-Hoc presenta 9 nodos en su estructura. Mediante el protocolo reactivo DSR, se generará el intercambio de información entre el nodo origen y destino. Cuando no encuentre la ruta correcta entre ellos, se necesitará de nodos cercanos para el envío y recepción de paquetes. La dimensión para éste escenario son los rangos: (x)=987 y para (y)=631 metros. La simulación en ésta red, se genera cuando el nodo origen inicia el reconocimiento de los nodos vecinos a partir del segundo 1.00000. Para detectar el mejor camino entre el nodo origen y destino, se actualizan todas las tablas de enrutamiento de cada dispositivo, de modo que, éstos nodos puedan transmitir los paquetes por la vía más corta y con menos saltos hacia el destino, cómo se indica en la figura 28.

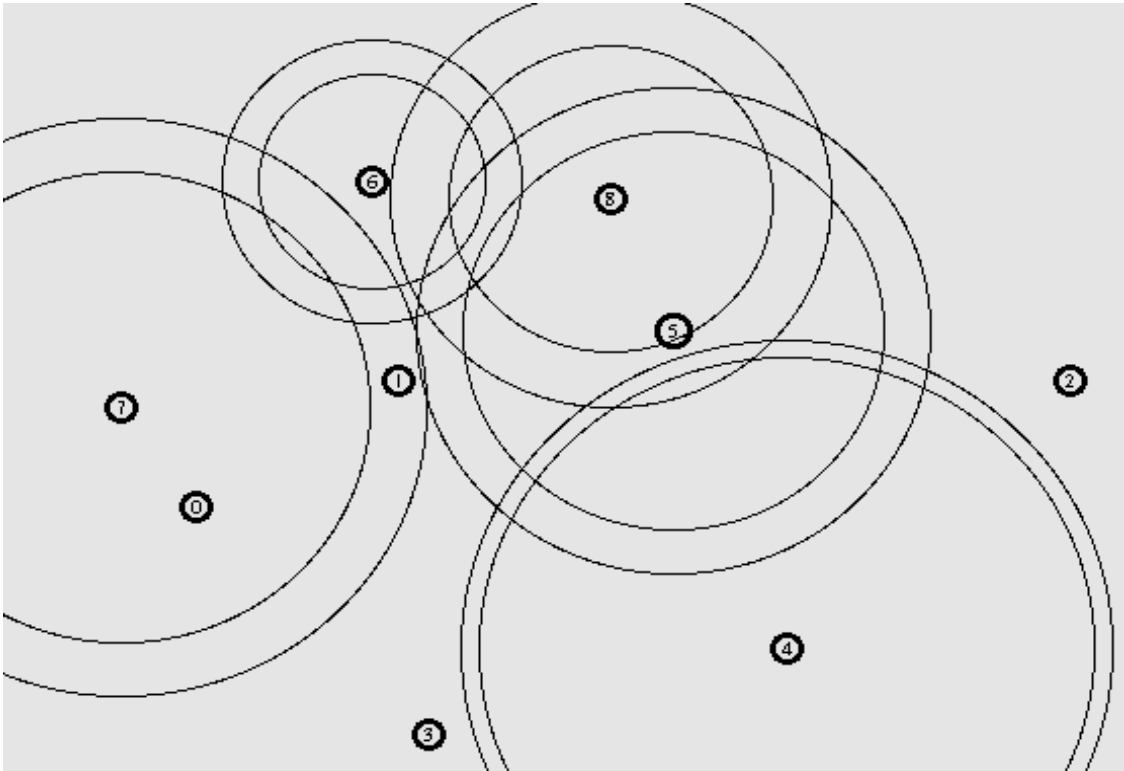


Figura 28. Actualización de las tablas con protocolo reactivo DSR.

Debido a la dinámica que presentan todos los nodos; al segundo 2.00000 el nodo destino 2 y el nodo vecino 4, empiezan a vincularse poco a poco al área de cobertura acortando las distancias entre los nodos existentes, cómo se muestra en la figura 29.

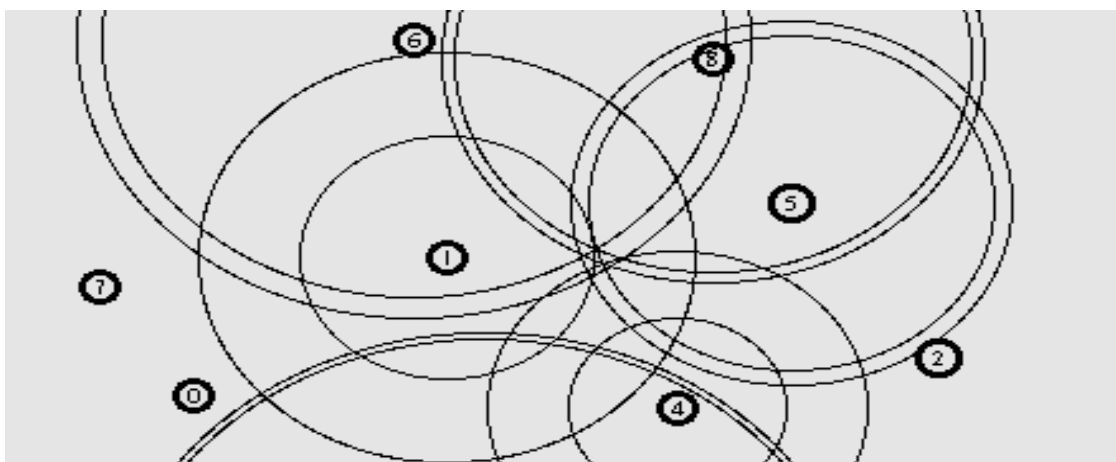


Figura 29. Vinculación del Nodo 2 y 4 con protocolo reactivo DSR

Cuándo se realiza la actualización de las tablas de enrutamiento, cada nodo obtiene la ruta con el menor número de saltos para establecer la comunicación entre el nodo origen y destino. Generada esta información, el nodo origen al segundo 3.2000, empieza a enviar las tramas por medio del enlace creado entre el nodo 7, 6, y 5 hasta llegar al nodo destino 2, cómo se muestra en la figura 30.

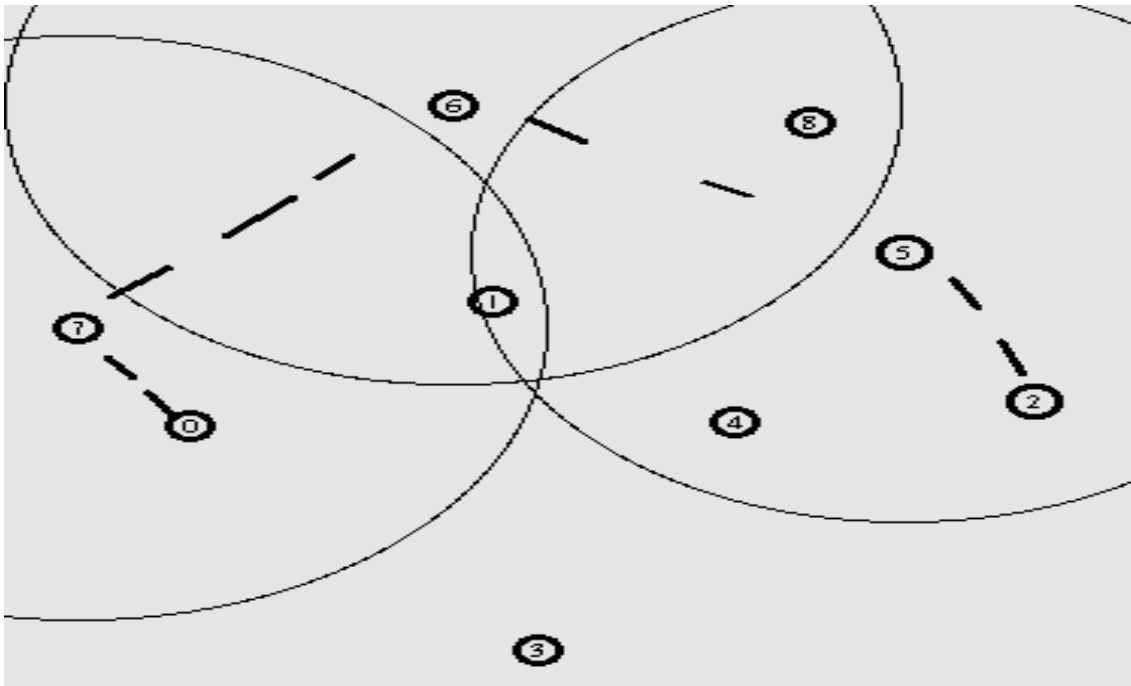


Figura 30. Envío de tramas desde nodo origen a nodo destino con protocolo reactivo DSR.

Segundos después, los nodos anteriores se alejan con rapidez del nodo origen, de modo que, comienza a existir pérdida de paquetes debido a la distancia generada. Cuándo ya no existe un medio de transmisión, el nodo origen presenta retardo en la entrega de paquetes, por lo cual, necesita encontrar una ruta hacia el nodo destino para enviar la información que fue suspendida en el enlace anterior. Cómo se muestra en la figura 31, el nodo 4 se desliza rápidamente entre el nodo origen y destino, en ese instante, se actualizan las tablas de enrutamiento con el fin de crear un nuevo camino para la transmisión de paquetes entre el nodo origen 0, nodo vecino 4 y nodo destino 2.

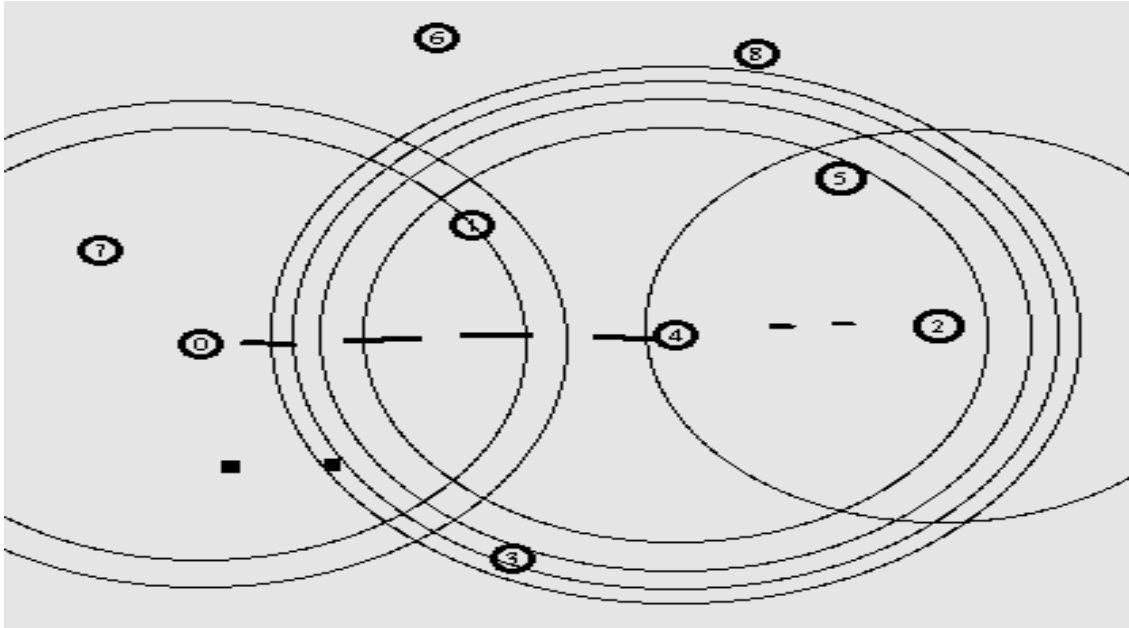


Figura 31. Cambio de ruta al nodo 4 con protocolo reactivo DSR.

Luego de un determinado tiempo, el nodo 4 empieza a alejarse a una velocidad alta del nodo origen y destino, de manera que, se genera caída de paquetes por los diversos movimientos de los nodos en la red. Al segundo 3.291583, el nodo 3 se acerca a esta área de cobertura convirtiéndose en el mediador de la nueva ruta para continuar así, con el envío y recepción de datos, cómo se muestra en la figura 32.

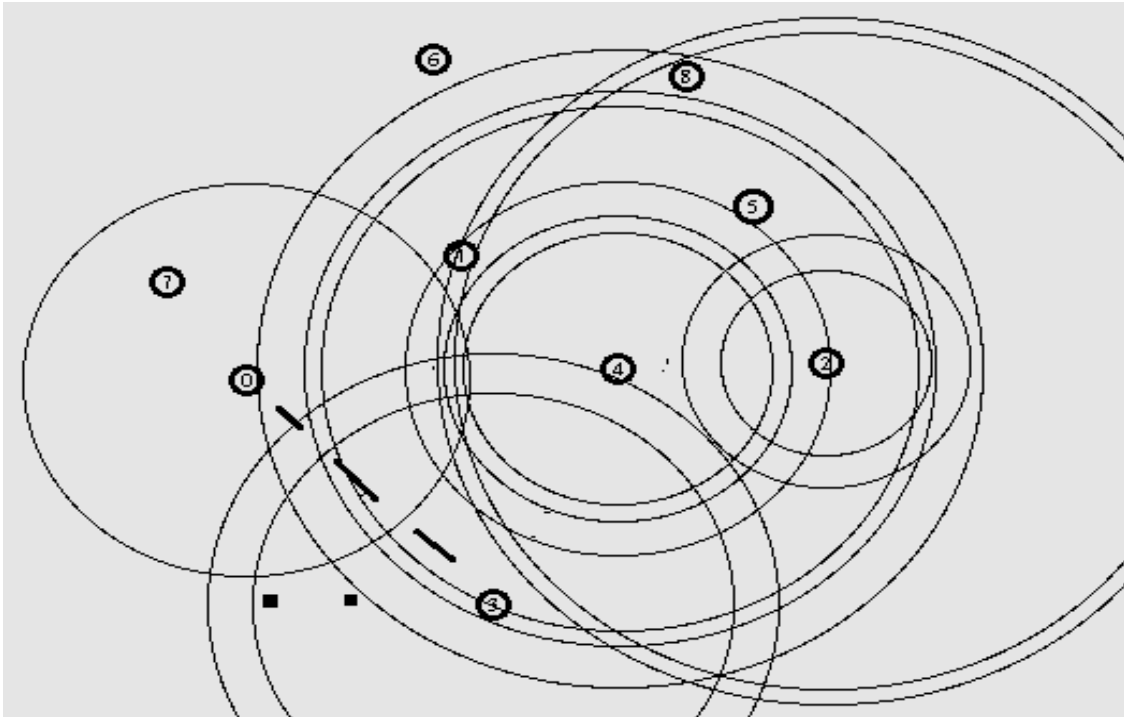


Figura 32. Mediador de información nodo 3 con protocolo reactivo DSR.

De forma lenta, el nodo 3 se retira y toma un nuevo rumbo. Mientras que el nodo 4, reaparece convirtiéndose en nodo intermedio para la transmisión de datos. Al segundo 8 aproximadamente, el nodo 4, se separa del perímetro de cobertura sin generar pérdida de información porque los nodos origen y destino se acercaban durante la comunicación anterior. Por lo tanto; el envío y recepción de las tramas, empieza a realizarse directamente entre nodo origen 0 y nodo destino 2, cómo se muestra en la figura 33.

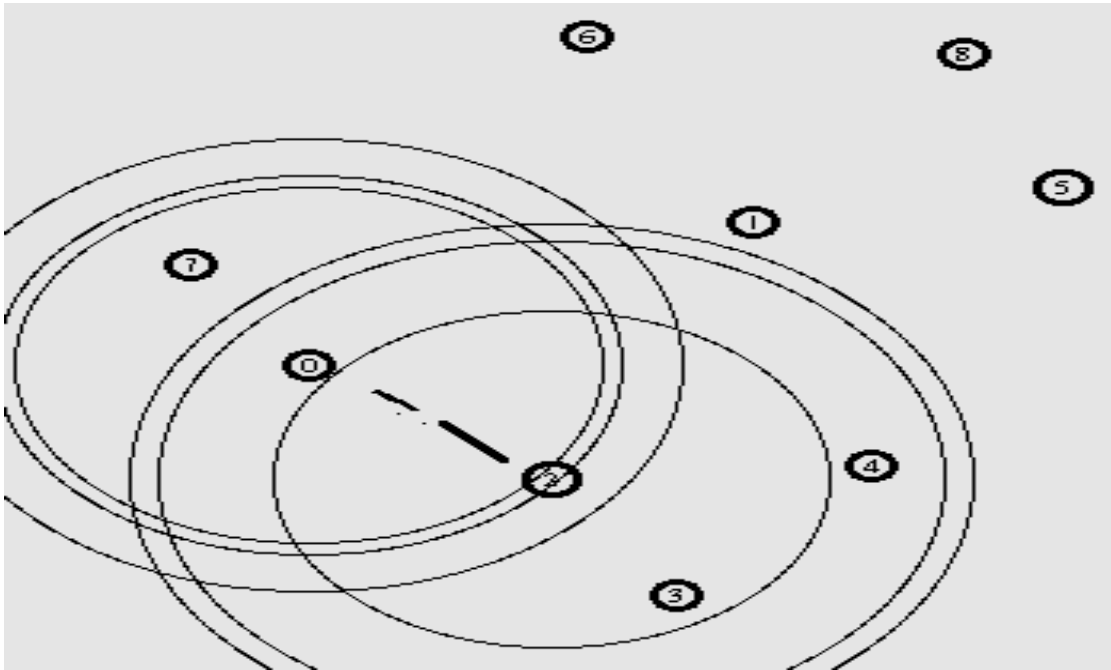


Figura 33. Envío y recepción de tramas entre nodo origen y destino con protocolo reactivo DSR.

### **CAPÍTULO III. ANÁLISIS DE LOS PROTOCOLO DE ENRUTAMIENTO REACTIVOS Y PROACTIVOS**

A continuación se presentará por medio de una tabla comparativa la información de cada uno de los 12 escenarios de simulación, con el fin de analizar el rendimiento, comportamiento y calidad de servicio del protocolo que mejor se acople al dinamismo de la topología de las redes Ad-Hoc. En estas tablas se incluirán los datos utilizados como:

- Ancho de banda (Mb/s)
- Velocidad de conexión (Mb/s)
- Número de nodos
- Tamaño máximo de colas (FIFO)
- Dimensión de los escenarios (metros)
- Movimiento de los nodos
- Tamaño de paquete
- Nodo origen y destino.
- Numero de saltos entre nodos.
- Retardos
  - Mínimo (s)
  - Medio (s)
  - Máximo (s)

Además, con estos datos aplicados en las simulaciones se mostrarán resultados como: número de paquetes generados, enviados, reenviados, caídos y perdidos; el tipo de retardo (segundos), mínimo, máximo y medio; por último el canal eficaz (número de saltos).

### 3.1.- Análisis de resultados obtenidos de los protocolos de enrutamiento AODB, DSR y DSDV

Tabla 3.

Comparativa de los resultados de 12 escenarios mediante los Protocolos de enrutamiento (AODV, DSR y DSDV).

Protocolos de Enrutamiento Reactivos y Proactivos													
Parámetros	Escenario 1 Protocolo AODV	Escenario 2 Protocolo AODV	Escenario 3 Protocolo AODV	Escenario 4 Protocolo AODV	Escenario 5 Protocolo DSDV	Escenario 6 Protocolo DSDV	Escenario 7 Protocolo DSDV	Escenario 8 Protocolo DSDV	Escenario 9 Protocolo DSR	Escenario 10 Protocolo DSR	Escenario 11 Protocolo DSR	Escenario 12 Protocolo DSR	
TIPO DE VARIABLES	ANCHO DE BANDA	10 Mb/s	1Mb/s	15Mb/s	10Mb/s	1Mb/s	15Mb/s	10Mb/s	10Mb/s	2Mb/s	2Mb/s	5Mb/s	10Mb/s
	VELOCIDAD DE CONEXIÓN	11 Mb/s	22Mb/s	22Mb/s	54Mb/s	22Mb/s	22Mb/s	54Mb/s	11Mb/s	11Mb/s	11Mb/s	54Mb/s	11Mb/s
	NÚMERO DE NODOS	9	22	22	22	22	22	22	9	22	22	22	9
	TAMAÑO MÁXIMO DE CELDAS	50 paquetes	20 Paquetes	30 paquetes	70 paquetes	20 paquetes	30 paquetes	70 paquetes	50 paquetes	50 paquetes	40 paquetes	20 paquetes	50 paquetes
	DIMENSIÓN DE ESCENARIO (metros)	(11125x)(640y)	(973x)(609y)	(973x)(604y)	(973x)(854y)	(973x)(609y)	(973x)(604y)	(973x)(854y)	(970x)(623y)	(849x)(558y)	(849x)(558y)	(849x)(558y)	(987x)(631y)
	TIEMPO DE SIMULACIÓN (segundos)	10	70	70	90	70	70	90	10	60	60	60	10
	NODOS EN MOVIMIENTO	2,3,6	2,8,20,13	2,8,20,13	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20	2,8,20,13	2,8,20,13	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20	estáticos	estáticos	8,4	9,12,8,5,7,2,4,6,7	1,2,4
	TAMAÑO DE PAQUETES	1500	10.000	500	10.000	10000	500	10000	1500	800	800	5000	1500
	NÚMERO DE SALTOS	7	3	3	7	4	4	8	2	0	2	8	6
NODOS TX	NODO ORIGEN	N1	N2	N2	N2	N2	N2	N2	N1	N3	N3	N3	N1
	NODO DESTINO	N8	N13	N13	N13	N13	N13	n13	N8	N6	N6	N6	N8
RETARDO	MÍNIMO	0,00415229	0,004970419	0,003085859	0,001243919	0,182650046	0,004105399	0,001247787	0,002004414	0,002684167	0,002644167	0,002575291	0,002125312
	MÁXIMO	6,03142945	6,325727091	0,188681724	0,445137213	0,389729851	0,19637575	3,054723007	0,326286501	0,156950879	0,163360762	0,156536227	2,152256537
	MESEO	0,159649222	0,180569231	0,082867472	0,07331511	0,320157736	0,08537769	0,042014273	0,129444574	0,07041414424	0,070592903	0,071479284	0,14620665
TIPO DE PAQUETES	NÚMERO DE PAQUETES GENERADOS	266	14446	33646	45673	4939	10598	18104	1846	33917	33946	33493	266
	NÚMERO DE PAQUETES ENVIADOS	266	14446	33646	37497	4939	10598	18104	1845	33917	33946	33493	266
	NÚMERO DE PAQUETES REENVIADOS	0	0	0	0	0	0	0	0	0	0	0	0
	NÚMERO DE PAQUETES CAÍDOS	2	113	176	268	7	120	130	57	674	639	597	2
	NÚMERO DE PAQUETES RECIBIDOS	278	14310	33310	45158	1326	10540	17730	1274	33426	33346	32936	278



Como se muestra en la tabla 3, para enviar y recibir archivos ftp en una red pequeña con nodos estáticos, dinamismo y comportamiento lento, es conveniente utilizar el protocolo proactivo DSDV, puesto que este algoritmo entrega un mayor número de paquetes por la rapidez en que realiza la actualización de sus tablas en redes donde no existe tanto movimiento, a diferencia de los protocolos reactivos DSR y AODV que presentaron mayor retardo en esta topología de red.

El retardo que se presenta en la simulación 1, tiene un rango de 0,004152 (s) y 6,031429 (s). En este caso el retardo es alto ya que el nodo origen se encuentra a una distancia lejana. Por esta razón, se presenta una demora en encontrar rutas para llegar al nodo destino. Cuando ya se conoce la ruta para arribar al origen, se genera un total 7 saltos dentro de la red.

Cómo se observa en la tabla 3, el número total de paquetes generados y enviados han sido transmitidos ya que los mensajes se intercambian muy rápido dentro de la topología de esta red. En esta simulación el protocolo DSR no presenta archivos reenviados. El número de paquetes caídos es mayor en el nodo origen ya que al momento de generar datos al destino, éste presenta un encaminamiento por inundación de paquetes, por lo cual, tratan de controlar los tamaños de cada trama recibida para no generar demoras en la comunicación. Cada nodo reenvía la información hasta que la trama sea recibida, lo que origina encolamiento en los nodos utilizados, a su vez pérdida y retardo de datos.

Los paquetes recibidos por el nodo destino es mayor al origen porque con éste protocolo, se realiza frecuentemente la actualización de sus tablas de enrutamiento. El número total de datos recibidos entre los dos nodos origen y destino es bastante efectivo ya que todos los paquetes que se envían por medio de los nodos vecinos reconocen rápidamente la mejor vía con el menor número de saltos.

En la simulación 12 el retardo mínimo, que en este caso son los tiempos en el que un paquete se demora en pasar a través de un nodo a otro es de 2,152256537 (s) y el tiempo que espera un nodo en recibir un paquete es de 0,14620665.

Como se observa en la tabla 3, la simulación 8 indica el tiempo mínimo que se demora un paquete en pasar por un nodo es de 0,326286501 (s) y el retardo máximo es de 0,14620665, debido a la movilidad y distancias cortas que en algunos casos se presentan dentro de la simulación. Estos valores muestran que la transmisión de datos se realiza en ciertos períodos de tiempo, de este modo, el enlace es paralelo lo que genera menor número de saltos entre los nodos origen y destino.

Para generar este análisis se tomó en consideración las tres simulaciones que muestran nueve nodos en sus escenarios con características como: ancho de banda de 10 Mb/s; el número de datos máximo de la cola FIFO, limitado a 50 paquetes; la velocidad de conexión de los nodos es de 11 Mb/s. A pesar de las distancias cortas y largas, el protocolo DSDV presentó menos retardo en reconocer nodos intermedios y rutas nuevas para enlazar el nodo origen y destino en 10 segundos, tiempo el cual duran estas simulaciones. El rendimiento entre estos tres protocolos varía de acuerdo a la movilidad de sus nodos, ancho de banda y la velocidad de transmisión.

Para redes amplias y con un mayor número de nodos (22), el protocolo que funciona de mejor manera es el reactivo DSR bajo las siguientes características como: ancho de banda, este puede estar entre 5 y 10 Mb/s; velocidad de transmisión, 54Mb/s; el número de datos máximo de la cola FIFO; puede estar limitado entre 20 y 70 paquetes y el tamaño de datos se encuentran entre los rangos 5000 y 10000 paquetes. Con las características analizadas anteriormente se puede afirmar que el protocolo DSR es el que mejor presenta rendimiento frente a un alto grado de dinamismo en estas topologías de red, a diferencia del rendimiento de los protocolos DSDV y AODV que en estas simulaciones los datos de envío y recepción fueron menores.

El protocolo DSR sigue presentando un buen desempeño en este tipo de redes con gran cantidad de nodos a pesar de la variación que pueda existir en los parámetros que se detallan a continuación:

- Ancho de banda (1 a 10Mb/s)
- Cola FIFO (20 y 50 paquetes)
- Tamaño de paquetes (500 y 10000)

- Topología (movilidad alta)

Cabe recalcar que los protocolos DSDV y AODV actúan de una manera intermitente por el número de nodos y el dinamismo que se presentan en estas simulaciones.

Otra forma de obtener diferentes resultados fue:

- Aumentando el ancho de banda y disminuyendo el tamaño de paquetes a enviar que se encuentran en rangos de 500- 1000.
- Velocidades entre 11-22 Mb/s.
- La cola FIFO se encuentra limitada en rangos de 30 y 50 paquetes

Con la variedad de parámetros que se generó en las simulaciones con 22 nodos, se afirma que el protocolo DSR Y AODV, se desempeñan del mismo modo en una red con dinamismo y comportamiento intermedio.

Mediante estos análisis se concluye que los tres protocolos de enrutamiento AODV, DSR, DSDV para conseguir mayor número de datos enviados y recibidos en una topología con dinamismo medio y con diferentes movimientos, dependen de la cola FIFO, velocidad de transmisión y ancho de banda con la que está configurada de modo que la pérdida de sus datos sea menor.

Como se muestra en algunos casos de los diferentes escenarios de simulación, si bajamos el ancho de banda en una red Ad-Hoc con características como: velocidad de transmisión 11Mb/s; cola FIFO limitada entre 20 y 50; tamaño de datos entre 1500 y 3000; y el movimiento de los nodos es intermedio; se concluye que el protocolo que mejores resultados obtuvo en este tipo de topologías de red, es el algoritmo DSR, seguido de los protocolos reactivos AODV y por último el proactivo DSDV.

### **3.2.- Análisis comparativo de los parámetros de calidad de funcionamiento de red mediante los indicadores ITU-T.**

Los parámetros de calidad de funcionamiento de red son los indicadores recomendados por las normas ITU-T, para medir la calidad de servicio que presentan los protocolos de enrutamiento. Por ser una red MANET en la que se aplica transferencia de archivos el indicador para esta simulación es la clase tipo 4. Para determinar QoS, los resultados obtenidos deben estar dentro de los rangos de estos indicadores o que su valor sea menor a los que ya están establecidos por estas normas.

Como se muestra en la tabla 4, el análisis comparativo mediante el primer indicador IPTD, para los protocolos de enrutamiento con 9 nodos; AODV, DSR y DSDV tenemos que: AODV y DSR están fuera del rango de 1(s) lo que significa; el tiempo en que se demoran los paquetes en transmitirse de un lado a otro es superior debido a la distancia y tipo de velocidad que presentan los nodos vecinos en la topología de red, de esta manera, se genera mayor número de saltos y aumenta el retardo de paquetes entre el nodo origen y destino. Mientras que DSDV está dentro del rango permitido por el indicador IPTD ya que es menor a 1 segundo. Por lo tanto DSDV presenta un funcionamiento de red óptimo frente a estos dos protocolos, dado que sus nodos origen y destino se encuentran a distancias cercanas y no necesitan realizar tantos saltos a otros nodos vecinos para comunicarse.

El segundo identificador IPDV para las tres simulaciones con los protocolos AODV, DSR y DSDV con 9 nodos, indican que estos algoritmos están dentro de los límites accesibles, puesto que, el tamaño de cada trama está limitado hasta 50 paquetes, es decir, si un nodo envía una trama y no llega al destino, el siguiente paquete no se puede enviar hasta que el nodo reciba la información emitida anteriormente.

Tabla 4.

Análisis de los parámetros de calidad de funcionamiento de red por medio de los indicadores ITU-T.

Clase 4 - QoS en los protocolos de enrutamiento Reactivos y Proactivos												
Parámetros de calidad de funcionamiento de red	Escenario 1 Protocolo AODV	Escenario 2 Protocolo AODV	Escenario 3 Protocolo AODV	Escenario 4 Protocolo AODV	Escenario 5 Protocolo DSDV	Escenario 6 Protocolo DSDV	Escenario 7 Protocolo DSDV	Escenario 8 Protocolo DSDV	Escenario 9 Protocolo DSR	Escenario 10 Protocolo DSR	Escenario 11 Protocolo DSR	Escenario 12 Protocolo DSR
Ip Packet Loss Ratio IPLR (< a 1X10) <sup>-3</sup> Pérdida de paquetes	0,00415229	0,004970419	0,003085859	0,001243919	0,182650046	0,004105399	0,001247787	0,002004414	0,002684167	0,002644167	0,002575291	0,002125312
Ip Packet Transfer Delay IPTD (< a 1 segundo) Tiempo en que se demoran los paquetes	6,03142945	6,325727091	0,188681724	0,445137213	0,389729851	0,19637575	3,054723007	0,326286501	0,156950879	0,163360762	0,156536227	2,152256537
Ip Packet Delay Variation IPDV (no especificado o sin límites ) Tiempo de espera de paquetes	0,159649222	0,180569231	0,082867472	0,07331511	0,320157736	0,08537769	0,042014273	0,129444574	0,07041414424	0,070592903	0,071479284	0,14620665

TPara el tercer identificador IPLR en los tres protocolos AODV, DSR y DSDV el porcentaje que menor indica es el DSDV aunque no se encuentra dentro del rango permitido por el indicador. Éste protocolo de enrutamiento expone un total del 0,002004414 mayor al valor establecido por indicador  $(1 \times 10)^{-3}$ . Este dato representa la pérdida de paquetes frente a las demás simulaciones puesto que, sus nodos origen y destino presentan menos movilidad y en ciertos casos se encuentran a distancias más cercanas de los nodos vecinos, estos paquetes se llegan a perder porque existen situaciones en la simulación en que, los enlaces se caen cuando un nodo queda fuera de la cobertura de su red o se alejan a velocidades altas en plena transmisión de información.

Los escenarios que se muestran a continuación, son las simulaciones que contienen 22 nodos con determinado dinamismo y comportamiento en su topología, estos presentan pérdida de paquetes al momento de realizar la comunicación entre los nodos origen y destino, por este motivo su calidad de servicio es mayor y quedan fuera por el indicador IPLR.

- Protocolo Reactivo AODV: 2.
- Protocolo Proactivo DSDV: 5, 6, 7.
- Protocolo Reactivo DSR: 9, 10, 11 y 12.

De la misma manera el indicador IPTD, separa a los siguientes protocolos por no presentar calidad de servicio al momento de transmitir los datos. Los escenarios con protocolo reactivo 2, 12 y el escenario con protocolo reactivo 7, son los que presentan mayor retraso al entregar la información entre los nodos origen y destino, debido a diferentes velocidades, distancias e interferencias que generan los nodos dentro de la red.

El indicador IPDV no se encuentra especificado para el caso de transmisión de archivos ftp.

Cómo resultado final del análisis de los temas entre: comparación de los parámetros de calidad de funcionamiento de red mediante los indicadores ITU-

T y la interpretación de los resultados entre protocolos AODV, DSR y DSDV mediante la dinámica de los nodos en sus redes; se ha podido demostrar por medio de fundamentos teóricos y por medio de los resultados obtenidos de las 12 simulaciones, que el mejor algoritmo de enrutamiento con dominantes características frente a gran cantidad de nodos y a diversos tipos de dinamismo en la red es el protocolo reactivo DSR, seguido por el protocolo AODV. Para redes pequeñas con menor número de nodos y dinamismo medio en redes Ad-Hoc, se concluyó que el algoritmo que mejor se desempeña frente a estas cualidades es el protocolo proactivo DSDV, seguido por los protocolos reactivos DSR y AODV.

## CAPÍTULO IV. SIMULADOR DE REDES NS-2

A continuación se detalla un estudio del programa NS-2 y sus herramientas con las cuales se realizó las simulaciones de este proyecto.

### 4.1.- Historia

La primera versión de simulador de redes es NS-1, fue la desarrollada en Lawrence Berkeley National Laboratory en el año 1995 por Steve McCanne, Sally Floyd, Kevin Fall. Este software fue conocido como LBNL Network Simulator, derivado de un simulador de red llamado REAL y desarrollado por S.Keshav.

El software NS-2 empezó a desarrollarse en el año 1989 como una herramienta de simulador de red real. Se suministra bajo el núcleo principal C++ y proporciona interfaces de simulación por medio de OTcl que es una variante orientada a objetos de Tcl. Este programa está diseñado para trabajar en sistemas operativos como: FreeBSD, Solaris, Windows y Linux. NS-2 es utilizado ampliamente como una herramienta de investigación y educativa. El usuario puede interactuar con el software bajo el lenguaje OTcl escribiendo (ns) en una línea de comandos.

El programa NS-2 es un software gratuito que sirve para la simulación de redes inalámbricas, redes cableadas y por satélite. La primera versión fue NS.1, la cual no contenía interfaz gráfica por lo que, dificultaba la creación de nodos y manipulación adecuada de sus características.

Luego de estas dos versiones, se desarrolla un nuevo simulador conocido como NS-3. Fue implementado por un equipo liderado por Tom Henderson, Sally Floyd, George Riley y Sumit Roy los cuales tenían apoyo de la Fundación Nacional de Ciencias de Estados Unidos, formando un proyecto de código abierto.

NS-1 ya no se desarrolla ni se mantiene, por lo que sus archivos que fueron generados con esa versión ya no son compatibles con las versiones actuales.



En el año 2010, el avance activo del NS-2 dejó de funcionar, pero existen usuarios que presentan casos de estudio de diferentes protocolos para su desarrollo e investigación.

NS-3 se presenta activo y en desarrollo, el único inconveniente, es al momento de importar o exportar los archivos, ya que no es compatible con sus versiones anteriores.

#### **4.1.1.- Características del programa de simulación**

Las características principales del simulador NS-2 son:

- Soporta sistemas operativos como Linux y Windows
- Acceso mediante línea de comandos
- Creación de topología y dinamismo de las redes inalámbricas.
- Creación de protocolos de enrutamiento reactivos, proactivos e híbridos.
- Creación de tipos de tráfico para una la red.
- Realiza análisis de rendimiento de red, jitter y salto de nodos.

#### **4.1.2.- Tipo de herramientas**

Las herramientas que se pueden instalar para trabajar con el simulador NS-2 son:

- NSG-2, generador de simulación de redes para desarrollar los TCL script.
- El programa Trace Graph 2.02 que se utilizó para generar las gráficas y la información de cada uno de los protocolos de enrutamiento.

Mediante este software, el usuario puede definir en un archivo script la programación del funcionamiento a simular de cada uno de ellos como:

- protocolos de enrutamiento
- número de nodos
- velocidad de nodos
- tamaño de paquetes
- dimensión de la simulación
- agentes (tcp, udp, tcp-sink), entre otros.

NS-2 contiene una interfaz gráfica para poder visualizar las simulaciones creadas ejecutando el comando nam (Animador de red). Esta herramienta dispone de un editor gráfico que facilita la edición de animaciones basadas en TCL. Se puede crear fuentes de tráfico, simular protocolos y crear topologías de red mediante este patrón.

**Con estas herramientas se puede definir:**

- Algoritmos de enrutamiento: DV, LS, PIM, DM, AODV, TORA, DSDV, DSR, OLSR.
- Fuentes de tráfico: ftp, udp, web, telnet, cbr, tcp-sink.
- Fallo de pérdidas de paquetes
- Fallos en la conexión de red
- Calidad de servicio: retardos, número de saltos y la mejor ruta.
- Disciplinas de encolamiento: RED, FQ, SFQ, DRR, Drop-tail

**Con estas herramientas se puede visualizar:**

- Flujo de los paquetes: encolamiento y tipo de paquetes
- Movilidad y dinamismo de nodos.
- Estados de los diferentes protocolos.

**4.1.3.- Tipo de archivos**

El resultado de cada simulación se muestra en unos archivos de registro que se genera por NS-2, todos los datos enviados durante la simulación se van guardando en líneas diferentes dentro de este archivo donde se describe los parámetros del paquete, tiempo de inicio, tiempo de simulación, nodos vecinos, no origen y destino, entre otros. Este programa tiene la función de generar un archivo especial que es interpretado por NAM (animador de red), el cual se puede visualizar de manera gráfica la actividad de nuestra red que fue simulada previamente.

NS-2 genera tres tipos de archivos al momento de ejecutar la simulación:

- **Out.tr:** es un archivo tipo texto en donde guarda los datos que se va obteniendo de la simulación y se lo conoce como traza.
- **Out.nam:** al momento de ejecutar este archivo se puede visualizar la simulación desarrollada.

- **Aadv.tcl:** es el archivo que contiene la programación de cada uno de los nodos.

#### **4.1.4.- Importancia del software NS-2**

Uno de los parámetros más importantes de este software de simulación es la manera en la que su arquitectura interna fue diseñada, la cual está integrada por diferentes tipos de clases que brindan gran potencia a la aplicación, adicionalmente estas clases son de código abierto permitiendo a los usuarios alterar el comportamiento del programa y sus protocolos de enrutamiento que se consideren indispensables para desarrollar cualquier tipo de investigación inalámbrica, cableada o por satélite. Estas clases son:

- Virtual Void.- Posee métodos de comunicación con diferentes módulos y la cola de eventos.
- Event.- Está integrado por el control de eventos, contiene toda la lista de tipos de eventos que se están ejecutando en ese momento.
- MAC.- Se encarga de los trabajos del acceso al medio del nodo y los diferentes protocolos de enrutamiento.

#### **4.1.5.- Consideraciones de NS-2**

Para crear cualquier tipo de simulación se considera ciertos parámetros que se debe configurar para obtener buenos resultados, por ejemplo tamaño de paquetes, número de nodos, ubicación de nodos, estos deben ser similares a una topología real para que puedan ejecutarse.

- El tráfico debe ser igual con el que se va a simular la red física de tal forma que el tipo de paquete, retardo, y el tiempo de comunicación sean similares a los generados por los dispositivos en una red real.
- Los modelos de propagación tiene que ser igual a un escenario real.
- El tiempo de inicio, fin y movilidad de los nodos deben ser similares a una red real, así mismo se deben ubicar en el interior del escenario.
- Las características como ancho de banda, generación de paquetes, retardos, deben ser configurados de tal forma que se parezcan a las propiedades de un dispositivo real.

#### **4.1.6.- Consejos y datos para simulaciones grandes en NS-2**

Existen diversos problemas que se presentan al momento de ejecutar simulaciones grandes en NS-2, sus principales restricciones son el tiempo-CPU y la memoria. Los requisitos para lograr un mejor rendimiento con cifras estadísticas a partir de las simulaciones desarrolladas son:

- Tiempo de inicio o el tiempo antes de que la simulación real pueda comenzar
- Tiempo de ejecución o cuánto tiempo lleva para completar la simulación
- Requisito de memoria para la simulación.

##### **4.1.6.1.- Tiempo de arranque.-**

Este puede definirse como la sobrecarga temporal solicitada antes de que la simulación real pueda comenzar en ns. Para una simulación dada usando enrutamiento estático, el tiempo de inicio consta de los siguientes intervalos de tiempo:

- Tiempos de análisis a través de secuencia de comandos realizados por usuario.
- Tiempos de configuración de enlaces
- Tiempos de configuración de salto de nodos

De este modo podemos decir que los tiempos de inicio consisten básicamente en el tiempo empleado para calcular las rutas y crear tablas de enrutamiento.

##### **4.1.6.2.- Tiempo de ejecución**

Este es el tiempo que tarda la simulación en completarse. Para minimizar los tiempos de inicio de los escenarios se debe desactivar los rastreadores como el trace-all y namtrace-all (todas las trazas que son enviadas al archivo de registro).

##### **4.1.6.3.- Uso de memoria**

Uno de los problemas más frecuentes que afrontan los usuarios mientras se desarrolla grandes simulaciones, es que se queda sin memoria por las diferentes aplicaciones del programa que están corriendo en segundo plano al momento de ser ejecutadas.

#### **4.1.7.- Ventajas y desventajas de un simulador de red**

#### **4.1.7.1.- Ventajas**

- se puede realizar comparaciones de redes con el mundo real.
- Libres de cualquier tipo de factores que imposibiliten la implementación de infraestructuras con altos costos de instalación.
- La licencia es libre y de código abierto.
- Accesibilidad de diseñar redes de gran amplitud y complejidad
- Facilitan la edición de las variables de red.

#### **4.1.7.2.- Desventajas**

- Los resultados de las simulaciones se alejan mucho de las reales por no poseer factores externos como errores de transmisión y tráfico, estos deben ser agregados independientemente.
- Los recursos del sistema disminuyen considerablemente cuando un escenario está diseñado por una topología con bastantes nodos y sus tiempos de movilidad sean extensos.

El presente trabajo de titulación se realizó mediante el programa NS-2 versión 2.34 en el sistema operativo Linux (Ubuntu) versión 16.04.

Para una mejor comprensión del uso e instalación de este simulador y manejo de sus herramientas se puede consultar en el anexo 13 y 15.

## CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

### 5.1.- Conclusiones

La simulación de una red Ad-Hoc ha permitido verificar el comportamiento de los nodos y su eficacia al momento de transmitir paquetes entre ellos, mediante los protocolos de enrutamiento reactivos AODV, DSR y proactivo DSDV, se pudo obtener diferentes resultados de los tipos de retardo que se genera al momento transmitir paquetes de gran tamaño en cada uno de los escenarios.

Mediante el análisis de los resultados en estas doce simulaciones, se ha concluido que, el mejor protocolo para utilizar en las redes Ad-Hoc, es el Protocolo reactivo DSR para redes con mayor número de nodos y el protocolo DSDV para redes que presentan poco dinamismo y menor número de nodos.

Para obtener varios resultados se tomó en consideración ciertos parámetros de dinamismo en la topología de sus redes como: velocidad, variación de ancho de banda, velocidad de transmisión, distancias, tamaño de paquetes, numero de nodos, entre otros.

En los escenarios de simulación que se utilizó el protocolo reactivo AODV, demuestra que los datos generados y enviados contienen menos paquetes mientras el nodo destino se encuentra lejos de la cobertura de los nodos vecinos, cuando más cerca se encuentre el nodo destino con el origen, éste presentará mayor capacidad de trasmisión de las tramas y la pérdida de los datos será menor mientras se encuentre dentro del radio de cobertura de la red.

Los escenarios de simulación en los que se utilizó el protocolo proactivo DSDV demuestran que se tiene mayor número de paquetes perdidos por lo que presenta como desventaja retraso en realizar su actualización de tablas de rutas, cuando más cerca se encuentren los nodos origen con el destino,

el número de paquetes generados y enviados será mayor siempre y cuando no exista una distancia que les deje fuera de la cobertura de la red.

Otro de los protocolos que funciona adecuadamente en este tipo de escenarios de red Ad-Hoc, es el Protocolo reactivo AODV por presentar características y resultados parecidos al protocolo de enrutamiento DSR, y en última instancia el Protocolo Proactivo DSDV, que demostró mejor desempeño en redes con poco número de nodos.

Por medio de los indicadores establecidos por la UIT-T se demuestra que el protocolo DSR desarrolla menor retardo en la transmisión de sus datos, del mismo modo su latencia indica que entrega un paquete activo (constante comunicación entre nodos vecinos) por medio de sus nodos al destino.

Los escenarios de simulación donde se encuentra el algoritmo DSR a pesar de tener varios paquetes caídos debido a las distancias largas, movilidad de sus nodos y cambio de rutas, presenta una red más óptima por lo que; genera, envía y recibe más paquetes dentro de la simulación con el menor retardo y menor número de saltos entre nodo destino y nodo origen.

Este protocolo DSR realizó mantenimientos de rutas mucho más rápidas a comparación de los demás protocolos, cuando la transmisión de los datos se perdía por la caída de los enlaces, este protocolo realizaba actualización de sus tablas de manera rápida y eficiente para encontrar nuevas rutas, mientras que los demás protocolos esperaban a que un nodo este cerca para generar un nuevo encaminamiento de rutas.

## **5.2.- Recomendaciones**

Se recomienda utilizar el protocolo de enrutamiento DSR para este tipo de redes Ad-Hoc (MANET) ya que sus características representadas en estos escenarios de simulación como tipo de paquetes, retardos y número de saltos entre nodos están dentro de los rangos dados por la UIT-T la cual demuestra que la calidad de servicio en el funcionamiento de estas redes es mejor.

El análisis realizado en este proyecto sobre la calidad de servicio, ha demostrado que en este tipo de redes Ad-Hoc, son comúnmente discutidas debido a que estas redes móviles presentan cambios en su topología de red y su dinamismo es constante, es por eso que se debe considerar un caso de estudio más avanzado y permanente sobre estas redes para brindar confiabilidad, escalabilidad y control de congestión en la transmisión de información.

Para obtener menos pérdida de paquetes al momento de enviar archivos, se debe examinar parámetros como las distancias entre dispositivos, usuarios, velocidades de transmisión, movilidad de dispositivos móviles, entre otros.

En un desastre natural al momento de caerse las comunicaciones y servicios de internet, el uso de las tecnologías Ad-Hoc son más prácticas y efectivas por lo que no necesitan instalación de grandes y costosas infraestructuras para poder comunicarse, de esta forma se recomienda utilizar una conexión de este tipo de redes para la interconexión de dispositivos en situaciones de emergencia, como el rescate de personas.

Se recomienda utilizar este tipo de software (NS-2) ya que contiene aplicaciones que permiten desarrollar escenarios de una manera sencilla, facilitando la comprensión de los diversos protocolos de enrutamiento que existen en las redes Ad-Hoc, además genera gráficos de simulación para observar el comportamiento de la red programada por el usuario.

A pesar de que el protocolo proactivo DSDV genera, envía y recibe más paquetes, se recomienda utilizar en redes con menor número de nodos, ya que en estas simulaciones se demostró que los dos nodos tanto de origen como de destino mantuvieron una comunicación casi permanente, dado que las distancias eran cortas y presentaron poco dinamismo en su topología de red.



## REFERENCIAS

Agent, C. t. (2010). The following OTcl code fragment creates a TCP agent and sets it up. Recuperado el 23 de Mayo de 2016 de <http://www.isi.edu/nsnam/ns/doc/node108.html>

Aurilia Torres Ontaneda. (2010). Analisis de calidad de servicio en enrutamiento de las redes moviles Ad Hoc. Recuperado el 12 de Mayo de 2016 de [http://dspace.utpl.edu.ec/bitstream/123456789/1443/3/Utpl\\_Torres\\_Ontaneda\\_Auliriar\\_004x790.%20\(1\).pdf](http://dspace.utpl.edu.ec/bitstream/123456789/1443/3/Utpl_Torres_Ontaneda_Auliriar_004x790.%20(1).pdf)

Carles Gómez i Montenegro,( 2004). REDES AD-HOC: EL PRÓXIMO RETO. Recuperado el 23 de Mayo de 2016 de <https://upcommons.upc.edu/bitstream/handle/2099/9919/Article006.pdf>

CR y SOL. (2006). Simulador ns2. Principios básicos. Recuperado el 12 de Abril de 2016 de <http://crysol.org/es/node/224>

documents.mx. (2015). MANUAL DE INSTALACIÓN TRACEGRAPH. Recuperado el 20 de Mayo de 2016 de <http://documents.mx/documents/manual-de-instalacion-tracegraph-55845bd8b0112.html>

Escuela Politecnica Nacional Sofia Naranjo. (2011). Analisis del comportamiento de la tecnologia Wimax y Wimax con trafico de voz usando simulador NS2. Recuperado el 12 de Abril de 2016 de <http://bibdigital.epn.edu.ec/bitstream/15000/4268/1/CD-3898.pdf>

Encaminamiento en Redes Ad-Hoc. (2013). Departamento de Sistemas Telemáticos y Computación (GSyC). Recuperado el 15 de Abril de 2016 de [https://gsyc.urjc.es/~mortuno/rom/06-encaminamiento\\_adhoc.pdf](https://gsyc.urjc.es/~mortuno/rom/06-encaminamiento_adhoc.pdf)

Francisco Javier Hidalgo Pastor. (2008). Estudio de viabilidad de la utilización de redes inalámbricas Ad -Hoc. Recuperado el 19 de Abril de 2016 de [https://riunet.upv.es/bitstream/handle/10251/13183/TesinaMaster\\_FcoJavierHidalgo.pdf?sequence=1](https://riunet.upv.es/bitstream/handle/10251/13183/TesinaMaster_FcoJavierHidalgo.pdf?sequence=1)

Fernando Rocha. (2010). NS2 Visual Trace Analyzer. Recuperado el 10 de Abril de 2016 de <https://nsvisualtraceanalyzer.files.wordpress.com/2012/07/ns2-visual-trace-analyzer-manual.pdf>

Jhon Jairo Padilla Aguilar. (2008). Manual de Practicas con NS2. Recuperado el 13 de Abril de 2016 de <http://jpadilla.docentes.upbbga.edu.co/programa%20redes/Manual%20de%20Practicas%20con%20NS2.pdf>

Jose Miguel Herrera M. (2004). NS2 - Network Simulator. Recuperado el 14 de Abril de 2016 de <https://www.inf.utfsm.cl/~jherrera/docs/mios/ns.pdf>

Karol Casierra. (2016). Red Ad Hoc Móvil: Características y Aplicaciones Principales. Recuperado el 12 de Abril de 2016 de <http://karolcasierra.blogspot.com/2013/04/red-ad-hoc-movil-caracteristicas-y.html>

Mg. Matías Robles , Ing. Luis Marrone , Lic. Javier D´iaz , Lic. Andrés Barbieri. (2010). Comparación de protocolos de ruteo en redes. Recuperado el 15 de Abril de 2016 de [http://sedici.unlp.edu.ar/bitstream/handle/10915/19343/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/19343/Documento_completo.pdf?sequence=1)

Mano. (2014). Installing Tracegraph2.02 application in Ubuntu. Recuperado el 17 de Abril de 2016 de <https://getch.wordpress.com/tag/tracegraph/>

MathWorks. (2011). BLAS and LAPACK lib locations. Recuperado el 22 de Abril de 2016 de [https://www.mathworks.com/matlabcentral/newsreader/view\\_thread/262144?requestedDomain=www.mathworks.com](https://www.mathworks.com/matlabcentral/newsreader/view_thread/262144?requestedDomain=www.mathworks.com)

Miguel Ángel Tolosa Diosdado . (2005). Sistemas Informáticos. Recuperado el 12 de Junio de 2016 de

[http://eprints.ucm.es/9004/1/Sistema\\_de\\_Autoconfiguracion\\_para\\_Redess\\_Ad\\_Hoc.pdf](http://eprints.ucm.es/9004/1/Sistema_de_Autoconfiguracion_para_Redess_Ad_Hoc.pdf)

NetWord Simulator. (2015). Setting a node to sleep in ns2. Recuperado el 13 de Junio de 2016 de <http://www.ns2blogger.in/>

Network Simulator. (2007). Simulation of data networks / ns2. Recuperado el 23 de Junio de 2016 de [http://www.netlab.tkk.fi/opetus/s383148/s07/luennot/ns2\\_part1.pdf](http://www.netlab.tkk.fi/opetus/s383148/s07/luennot/ns2_part1.pdf)

Nelson R. Rodriguez. (2006). Evaluacion y simulacion del rendimiento del protocolo de ruteo para manet bajo restricciones de QoS. Recuperado el 12 de Junio de 2016 de [http://sedici.unlp.edu.ar/bitstream/handle/10915/19663/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/19663/Documento_completo.pdf?sequence=1)

overflow, s. (2014). How to fix Errors in TCL ns2. Recuperado el 25 de Junio de 2016 de <http://stackoverflow.com/questions/34258069/how-to-fix-errors-in-tcl-ns2/34262246>

Redes Ad Hoc. (2010). Protocolo DSDV (Destination-Sequenced Distance-Vector). Recuperado el 13 de Junio de 2016 de [http://www.gta.ufrj.br/grad/09\\_1/versao-final/adhoc/dsdv.html](http://www.gta.ufrj.br/grad/09_1/versao-final/adhoc/dsdv.html)

Rocabado Moreno Sergio. (2013). caso de estudio de comunicaciones seguras sobre redes moviles. Recuperado el 15 de Junio de 2016 de [http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Redes\\_de\\_Datos/Tesis/Rocabado\\_Moreno\\_Sergio.pdf](http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Redes_de_Datos/Tesis/Rocabado_Moreno_Sergio.pdf)

San Mateo Educacion Superior. (2010). SIMULACIÓN EN NS-2 PARA SISTEMAS DE TRANSPORTE INTELIGENTE (ITS). Recuperado el 20 de Junio de 2016 de <https://sites.google.com/a/funsanmateo.edu.co/semillero-en-nuevas-tecnologias/ns-2/simulacion-en-ns-2-para-sistemas-de-transporte-inteligente-its>

Tamilnadu, E. E. (2010). Quick links. Recuperado el 13 de Julio de 2016 de [http://enggedu.com/source\\_code/ns2/ns2/wireless/Tcl\\_script\\_to\\_make\\_TCP\\_communication\\_between\\_nodes\\_using\\_DSDV\\_routing\\_protocol.php](http://enggedu.com/source_code/ns2/ns2/wireless/Tcl_script_to_make_TCP_communication_between_nodes_using_DSDV_routing_protocol.php)

## **ANEXOS**

## Anexo 1

### Código fuente de la simulación 1 con Protocolo de enrutamiento AODV

```

#=====
#  Simulation parameters setup
#=====
setval(chan) Channel/WirelessChannel  ;# channel type
setval(prop) Propagation/TwoRayGround ;# radio-propagation model
setval(netif) Phy/WirelessPhy        ;# network interface type
setval(mac)  Mac/802_11               ;# MAC type
setval(ifq)  Queue/DropTail/PriQueue  ;# interface queue type
setval(ll)   LL                       ;# link layer type
setval(ant)  Antenna/OmniAntenna      ;# antenna model
setval(ifqlen) 50                     ;# max packet in ifq
setval(nn)    9                       ;# number of mobilenodes
setval(rp)    AODV                    ;# routing protocol
setval(x)     1125                    ;# X dimension of topography
setval(y)     640                     ;# Y dimension of topography
setval(stop)  10.0                   ;# time of simulation end

#=====
#  Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
settopo [new Topography]
$topoload_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file

```

```
settracefile [open out.tr w]
```

```
$ns trace-all $tracefile
```

```
#Open the NAM trace file
```

```
setnamfile [open out.nam w]
```

```
$ns namtrace-all $namfile
```

```
$ns namtrace-all-wireless $namfile $val(x) $val(y)
```

```
setchan [new $val(chan)];#Create wireless channel
```

```
#=====
```

```
# Mobile node parameter setup
```

```
#=====
```

```
$ns node-config -adhocRouting $val(rp) \
```

```
    -llType    $val(ll) \
```

```
    -macType   $val(mac) \
```

```
    -ifqType   $val(ifq) \
```

```
    -ifqLen    $val(ifqlen) \
```

```
    -antType   $val(ant) \
```

```
    -propType  $val(prop) \
```

```
    -phyType   $val(netif) \
```

```
    -channel   $chan \
```

```
    -topoInstance $topo \
```

```
    -agentTrace ON \
```

```
    -routerTrace ON \
```

```
    -macTrace  ON \
```

```
    -movementTrace ON
```

```
#=====
```

```
# Nodes Definition
```

```
#=====
```

```
#Create 9 nodes
```

```
set n0 [$ns node]
```

```
$n0 set X_ 157
```

```
$n0 set Y_ 490
```

```
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 208
$n1 set Y_ 272
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 413
$n2 set Y_ 167
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 458
$n3 set Y_ 379
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 519
$n4 set Y_ 525
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 643
$n5 set Y_ 345
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 709
$n6 set Y_ 150
$n6 set Z_ 0.0
$ns initial_node_pos $n6 20
set n7 [$ns node]
$n7 set X_ 787
```

```

$n7 set Y_ 331
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 1025
$n8 set Y_ 459
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20

```

```

#=====

```

```

#   Generate movement

```

```

#=====

```

```

$ns at 8 " $n2 setdest 430 378 180 "
$ns at 0.1 " $n3 setdest 560 540 120 "
$ns at 0.50 " $n6 setdest 430 350 180 "
$ns at 0.10 " $n6 setdest 380 220 150 "
$ns at 7 " $n6 setdest 159 470 200 "
$ns at 3 " $n8 setdest 600 400 150 "

```

```

#=====

```

```

#   Agents Definition

```

```

#=====

```

```

#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n1 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n8 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500

```

```

#=====

```

```

#   Applications Definition

```

```

#=====

```



```
#Setup a FTP Application over TCP connection
```

```
set ftp2 [new Application/FTP]
```

```
$ftp2 attach-agent $tcp0
```

```
$ns at 1.0 "$ftp2 start"
```

```
$ns at 20.0 "$ftp2 stop"
```

```
#=====
```

```
# Termination
```

```
#=====
```

```
#Define a 'finish' procedure
```

```
proc finish {} {
```

```
global ns tracefilenamfile
```

```
    $ns flush-trace
```

```
close $tracefile
```

```
close $namfile
```

```
execnamout.nam&
```

```
exit 0
```

```
}
```

```
for {set i 0} {$i < $val(nn)} {incr i} {
```

```
    $ns at $val(stop) "\n$i reset"
```

```
}
```

```
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
```

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

```
$ns at $val(stop) "puts \"done\" ; $ns halt"
```

```
$nsrun
```

## Anexo 2

### Código fuente de la simulación 2 con Protocolo de enrutamiento AODV

```

#=====
# Simulation parameters setup
#=====
Phy/WirelessPhy set bandwidth_ 1Mb      ;#Data Rate
Mac/802_11 set dataRate_ 22Mb          ;#Rate for Data Frames
set val(chan) Channel/WirelessChannel  ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy        ;# network interface type
set val(mac) Mac/802_11                ;# MAC type
set val(ifq) Queue/DropTail/PriQueue   ;# interface queue type
set val(ll) LL                          ;# link layer type
set val(ant) Antenna/OmniAntenna       ;# antenna model
set val(ifqlen) 20                      ;# max packet in ifq
set val(nn) 22                          ;# number of mobilenodes
set val(rp) AODV                        ;# routing protocol
set val(x) 973                          ;# X dimension of topography
set val(y) 609                          ;# Y dimension of topography
set val(stop) 70.0                      ;# time of simulation end

#=====
# Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

```

```
create-god $val(nn)
```

```
#Open the NS trace file
```

```
set tracefile [open out.tr w]
```

```
$ns trace-all $tracefile
```

```
#Open the NAM trace file
```

```
set namfile [open out.nam w]
```

```
$ns namtrace-all $namfile
```

```
$ns namtrace-all-wireless $namfile $val(x) $val(y)
```

```
set chan [new $val(chan)];#Create wireless channel
```

```
#=====
```

```
# Mobile node parameter setup
```

```
#=====
```

```
$ns node-config -adhocRouting $val(rp) \
```

```
    -llType    $val(ll) \
```

```
    -macType   $val(mac) \
```

```
    -ifqType   $val(ifq) \
```

```
    -ifqLen    $val(ifqlen) \
```

```
    -antType   $val(ant) \
```

```
    -propType  $val(prop) \
```

```
    -phyType   $val(netif) \
```

```
    -channel   $chan \
```

```
    -topoInstance $topo \
```

```
    -agentTrace ON \
```

```
    -routerTrace ON \
```

```
    -macTrace  ON \
```

```
    -movementTrace ON
```

```
#=====
```

```
# Nodes Definition
```

```
#=====
```

```
#Create 22 nodes
```

```
set n0 [$ns node]
$n0 set X_ 280
$n0 set Y_ 381
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 420
$n1 set Y_ 390
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 214
$n2 set Y_ 443
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 327
$n3 set Y_ 310
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 300
$n4 set Y_ 170
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 710
$n5 set Y_ 228
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 564
$n6 set Y_ 329
$n6 set Z_ 0.0
```

```
$ns initial_node_pos $n6 20
set n7 [$ns node]
$n7 set X_ 637
$n7 set Y_ 175
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 430
$n8 set Y_ 318
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20
set n9 [$ns node]
$n9 set X_ 591
$n9 set Y_ 241
$n9 set Z_ 0.0
$ns initial_node_pos $n9 20
set n10 [$ns node]
$n10 set X_ 641
$n10 set Y_ 326
$n10 set Z_ 0.0
$ns initial_node_pos $n10 20
set n11 [$ns node]
$n11 set X_ 793
$n11 set Y_ 181
$n11 set Z_ 0.0
$ns initial_node_pos $n11 20
set n12 [$ns node]
$n12 set X_ 635
$n12 set Y_ 392
$n12 set Z_ 0.0
$ns initial_node_pos $n12 20
set n13 [$ns node]
$n13 set X_ 536
$n13 set Y_ 194
```

\$n13 set Z\_ 0.0  
\$ns initial\_node\_pos \$n13 20  
set n14 [\$ns node]  
\$n14 set X\_ 859  
\$n14 set Y\_ 259  
\$n14 set Z\_ 0.0  
\$ns initial\_node\_pos \$n14 20  
set n15 [\$ns node]  
\$n15 set X\_ 312  
\$n15 set Y\_ 452  
\$n15 set Z\_ 0.0  
\$ns initial\_node\_pos \$n15 20  
set n16 [\$ns node]  
\$n16 set X\_ 873  
\$n16 set Y\_ 179  
\$n16 set Z\_ 0.0  
\$ns initial\_node\_pos \$n16 20  
set n17 [\$ns node]  
\$n17 set X\_ 369  
\$n17 set Y\_ 392  
\$n17 set Z\_ 0.0  
\$ns initial\_node\_pos \$n17 20  
set n18 [\$ns node]  
\$n18 set X\_ 731  
\$n18 set Y\_ 406  
\$n18 set Z\_ 0.0  
\$ns initial\_node\_pos \$n18 20  
set n19 [\$ns node]  
\$n19 set X\_ 345  
\$n19 set Y\_ 423  
\$n19 set Z\_ 0.0  
\$ns initial\_node\_pos \$n19 20  
set n20 [\$ns node]  
\$n20 set X\_ 282

```

$n20 set Y_ 495
$n20 set Z_ 0.0
$ns initial_node_pos $n20 20
set n21 [$ns node]
$n21 set X_ 133
$n21 set Y_ 479
$n21 set Z_ 0.0
$ns initial_node_pos $n21 20

```

```

#=====

```

```

#   Generate movement

```

```

#=====

```

```

$ns at 14 " $n2 setdest 111 509 50 "
$ns at 15 " $n8 setdest 300 400 100 "
$ns at 14 " $n20 setdest 144 466 100 "

```

```

#=====

```

```

#   Agents Definition

```

```

#=====

```

```

#Setup a TCP connection
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set sink3 [new Agent/TCPSink]
$ns attach-agent $n13 $sink3
$ns connect $tcp2 $sink3
$tcp2 set packetSize_ 10000

```

```

#=====

```

```

#   Applications Definition

```

```

#=====

```

```

#Setup a FTP Application over TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp2

```

\$ns at 1.0 "\$ftp1 start"

\$ns at 70.0 "\$ftp1 stop"

#=====

# Termination

#=====

#Define a 'finish' procedure

proc finish {} {

    global ns tracefile namfile

    \$ns flush-trace

    close \$tracefile

    close \$namfile

    exec nam out.nam &

    exit 0

}

for {set i 0} {\$i < \$val(nn)} {incr i} {

    \$ns at \$val(stop) "\\$n\$i reset"

}

\$ns at \$val(stop) "\$ns nam-end-wireless \$val(stop)"

\$ns at \$val(stop) "finish"

\$ns at \$val(stop) "puts \"done\" ; \$ns halt"

\$ns run



### Anexo 3

#### Código fuente de la simulación 3 con protocolo de enrutamiento AODV

```

#=====
#  Simulation parameters setup
#=====
Phy/WirelessPhy set bandwidth_ 15Mb      ;#Data Rate
Mac/802_11 set dataRate_ 22Mb          ;#Rate for Data Frames
set val(chan) Channel/WirelessChannel  ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy        ;# network interface type
set val(mac) Mac/802_11                ;# MAC type
set val(ifq) Queue/DropTail/PriQueue   ;# interface queue type
set val(ll) LL                          ;# link layer type
set val(ant) Antenna/OmniAntenna       ;# antenna model
set val(ifqlen) 30                      ;# max packet in ifq
set val(nn) 22                          ;# number of mobilenodes
set val(rp) AODV                        ;# routing protocol
set val(x) 973                          ;# X dimension of topography
set val(y) 609                          ;# Y dimension of topography
set val(stop) 70.0                      ;# time of simulation end

#=====
#  Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

```

```
create-god $val(nn)
```

```
#Open the NS trace file
```

```
set tracefile [open out.tr w]
```

```
$ns trace-all $tracefile
```

```
#Open the NAM trace file
```

```
set namfile [open out.nam w]
```

```
$ns namtrace-all $namfile
```

```
$ns namtrace-all-wireless $namfile $val(x) $val(y)
```

```
set chan [new $val(chan)];#Create wireless channel
```

```
#=====
```

```
# Mobile node parameter setup
```

```
#=====
```

```
$ns node-config -adhocRouting $val(rp) \
```

```
    -llType      $val(ll) \
```

```
    -macType     $val(mac) \
```

```
    -ifqType     $val(ifq) \
```

```
    -ifqLen      $val(ifqlen) \
```

```
    -antType     $val(ant) \
```

```
    -propType    $val(prop) \
```

```
    -phyType     $val(netif) \
```

```
    -channel     $chan \
```

```
    -topoInstance $topo \
```

```
    -agentTrace  ON \
```

```
    -routerTrace ON \
```

```
    -macTrace    ON \
```

```
    -movementTrace ON
```

```
#=====
```

```
# Nodes Definition
```

```
#=====
```

```
#Create 22 nodes
```

```
set n0 [$ns node]
$n0 set X_ 280
$n0 set Y_ 381
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 420
$n1 set Y_ 390
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 214
$n2 set Y_ 443
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 327
$n3 set Y_ 310
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 300
$n4 set Y_ 170
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 710
$n5 set Y_ 228
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 564
$n6 set Y_ 329
$n6 set Z_ 0.0
```

```
$ns initial_node_pos $n6 20
set n7 [$ns node]
$n7 set X_ 637
$n7 set Y_ 175
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 430
$n8 set Y_ 318
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20
set n9 [$ns node]
$n9 set X_ 591
$n9 set Y_ 241
$n9 set Z_ 0.0
$ns initial_node_pos $n9 20
set n10 [$ns node]
$n10 set X_ 641
$n10 set Y_ 326
$n10 set Z_ 0.0
$ns initial_node_pos $n10 20
set n11 [$ns node]
$n11 set X_ 793
$n11 set Y_ 181
$n11 set Z_ 0.0
$ns initial_node_pos $n11 20
set n12 [$ns node]
$n12 set X_ 635
$n12 set Y_ 392
$n12 set Z_ 0.0
$ns initial_node_pos $n12 20
set n13 [$ns node]
$n13 set X_ 536
$n13 set Y_ 194
```

\$n13 set Z\_ 0.0  
\$ns initial\_node\_pos \$n13 20  
set n14 [\$ns node]  
\$n14 set X\_ 859  
\$n14 set Y\_ 259  
\$n14 set Z\_ 0.0  
\$ns initial\_node\_pos \$n14 20  
set n15 [\$ns node]  
\$n15 set X\_ 312  
\$n15 set Y\_ 452  
\$n15 set Z\_ 0.0  
\$ns initial\_node\_pos \$n15 20  
set n16 [\$ns node]  
\$n16 set X\_ 873  
\$n16 set Y\_ 179  
\$n16 set Z\_ 0.0  
\$ns initial\_node\_pos \$n16 20  
set n17 [\$ns node]  
\$n17 set X\_ 369  
\$n17 set Y\_ 392  
\$n17 set Z\_ 0.0  
\$ns initial\_node\_pos \$n17 20  
set n18 [\$ns node]  
\$n18 set X\_ 731  
\$n18 set Y\_ 406  
\$n18 set Z\_ 0.0  
\$ns initial\_node\_pos \$n18 20  
set n19 [\$ns node]  
\$n19 set X\_ 345  
\$n19 set Y\_ 423  
\$n19 set Z\_ 0.0  
\$ns initial\_node\_pos \$n19 20  
set n20 [\$ns node]  
\$n20 set X\_ 282

```

$n20 set Y_ 495
$n20 set Z_ 0.0
$ns initial_node_pos $n20 20
set n21 [$ns node]
$n21 set X_ 133
$n21 set Y_ 479
$n21 set Z_ 0.0
$ns initial_node_pos $n21 20

```

```

#=====
#   Generate movement
#=====
$ns at 14 " $n2 setdest 111 509 50 "
$ns at 15 " $n8 setdest 300 400 100 "
$ns at 14 " $n20 setdest 144 466 100 "

```

```

#=====
#   Agents Definition
#=====
#Setup a TCP connection
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set sink3 [new Agent/TCPSink]
$ns attach-agent $n13 $sink3
$ns connect $tcp2 $sink3
$tcp2 set packetSize_ 500

```

```

#=====
#   Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp2

```

\$ns at 1.0 "\$ftp1 start"

\$ns at 70.0 "\$ftp1 stop"

#=====

# Termination

#=====

#Define a 'finish' procedure

proc finish {} {

    global ns tracefile namfile

    \$ns flush-trace

    close \$tracefile

    close \$namfile

    exec nam out.nam &

    exit 0

}

for {set i 0} {\$i < \$val(nn)} {incr i} {

    \$ns at \$val(stop) "\\$n\$i reset"

}

\$ns at \$val(stop) "\$ns nam-end-wireless \$val(stop)"

\$ns at \$val(stop) "finish"

\$ns at \$val(stop) "puts \"done\" ; \$ns halt"

\$ns run

## Anexo 4

### Código fuente de la simulación 4 con protocolo de enrutamiento AODV

```

#=====
# Simulation parameters setup
#=====
Phy/WirelessPhy set bandwidth_ 10Mb      ;#Data Rate
Mac/802_11 set dataRate_ 54Mb          ;#Rate for Data Frames
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy         ;# network interface type
set val(mac) Mac/802_11                ;# MAC type
set val(ifq) Queue/DropTail/PriQueue   ;# interface queue type
set val(ll) LL                          ;# link layer type
set val(ant) Antenna/OmniAntenna       ;# antenna model
set val(ifqlen) 70                      ;# max packet in ifq
set val(nn) 22                          ;# number of mobilenodes
set val(rp) AODV                        ;# routing protocol
set val(x) 973                          ;# X dimension of topography
set val(y) 854                          ;# Y dimension of topography
set val(stop) 90.0                      ;# time of simulation end

#=====
# Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

```



```
create-god $val(nn)
```

```
#Open the NS trace file
```

```
set tracefile [open out.tr w]
```

```
$ns trace-all $tracefile
```

```
#Open the NAM trace file
```

```
set namfile [open out.nam w]
```

```
$ns namtrace-all $namfile
```

```
$ns namtrace-all-wireless $namfile $val(x) $val(y)
```

```
set chan [new $val(chan)];#Create wireless channel
```

```
#=====
```

```
# Mobile node parameter setup
```

```
#=====
```

```
$ns node-config -adhocRouting $val(rp) \
```

```
    -llType    $val(ll) \
```

```
    -macType   $val(mac) \
```

```
    -ifqType   $val(ifq) \
```

```
    -ifqLen    $val(ifqlen) \
```

```
    -antType   $val(ant) \
```

```
    -propType  $val(prop) \
```

```
    -phyType   $val(netif) \
```

```
    -channel   $chan \
```

```
    -topoInstance $topo \
```

```
    -agentTrace ON \
```

```
    -routerTrace ON \
```

```
    -macTrace  ON \
```

```
    -movementTrace ON
```

```
#=====
```

```
# Nodes Definition
```

```
#=====
```

```
#Create 22 nodes
```

```
set n0 [$ns node]
$n0 set X_ 280
$n0 set Y_ 381
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 420
$n1 set Y_ 390
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 214
$n2 set Y_ 443
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 327
$n3 set Y_ 310
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 300
$n4 set Y_ 170
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 710
$n5 set Y_ 228
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 564
$n6 set Y_ 329
$n6 set Z_ 0.0
```

```
$ns initial_node_pos $n6 20
set n7 [$ns node]
$n7 set X_ 637
$n7 set Y_ 175
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 430
$n8 set Y_ 318
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20
set n9 [$ns node]
$n9 set X_ 591
$n9 set Y_ 241
$n9 set Z_ 0.0
$ns initial_node_pos $n9 20
set n10 [$ns node]
$n10 set X_ 641
$n10 set Y_ 326
$n10 set Z_ 0.0
$ns initial_node_pos $n10 20
set n11 [$ns node]
$n11 set X_ 793
$n11 set Y_ 181
$n11 set Z_ 0.0
$ns initial_node_pos $n11 20
set n12 [$ns node]
$n12 set X_ 635
$n12 set Y_ 392
$n12 set Z_ 0.0
$ns initial_node_pos $n12 20
set n13 [$ns node]
$n13 set X_ 536
$n13 set Y_ 194
```

\$n13 set Z\_ 0.0  
\$ns initial\_node\_pos \$n13 20  
set n14 [\$ns node]  
\$n14 set X\_ 859  
\$n14 set Y\_ 259  
\$n14 set Z\_ 0.0  
\$ns initial\_node\_pos \$n14 20  
set n15 [\$ns node]  
\$n15 set X\_ 312  
\$n15 set Y\_ 452  
\$n15 set Z\_ 0.0  
\$ns initial\_node\_pos \$n15 20  
set n16 [\$ns node]  
\$n16 set X\_ 873  
\$n16 set Y\_ 179  
\$n16 set Z\_ 0.0  
\$ns initial\_node\_pos \$n16 20  
set n17 [\$ns node]  
\$n17 set X\_ 369  
\$n17 set Y\_ 392  
\$n17 set Z\_ 0.0  
\$ns initial\_node\_pos \$n17 20  
set n18 [\$ns node]  
\$n18 set X\_ 731  
\$n18 set Y\_ 406  
\$n18 set Z\_ 0.0  
\$ns initial\_node\_pos \$n18 20  
set n19 [\$ns node]  
\$n19 set X\_ 345  
\$n19 set Y\_ 423  
\$n19 set Z\_ 0.0  
\$ns initial\_node\_pos \$n19 20  
set n20 [\$ns node]  
\$n20 set X\_ 282

```

$n20 set Y_ 495
$n20 set Z_ 0.0
$ns initial_node_pos $n20 20
set n21 [$ns node]
$n21 set X_ 133
$n21 set Y_ 479
$n21 set Z_ 0.0
$ns initial_node_pos $n21 20

```

```

#=====

```

```

#   Generate movement

```

```

#=====

```

```

$ns at 16 " $n0 setdest 225 550 90 "
$ns at 13 " $n1 setdest 300 311 58 "
$ns at 14 " $n2 setdest 111 509 50 "
$ns at 30 " $n2 setdest 500 356 60 "
$ns at 62 " $n2 setdest 316 198 60 "
$ns at 30 " $n2 setdest 500 356 60 "
$ns at 6 " $n2 setdest 90 357 78 "
$ns at 20 " $n2 setdest 345 487 50 "
$ns at 30 " $n2 setdest 530 335 56 "
$ns at 10 " $n3 setdest 470 200 70 "
$ns at 13 " $n3 setdest 450 400 40 "
$ns at 18 " $n3 setdest 500 300 70 "
$ns at 5 " $n4 setdest 550 315 50 "
$ns at 47 " $n4 setdest 234 378 30 "
$ns at 55 " $n4 setdest 534 236 60 "
$ns at 5 " $n4 setdest 550 315 50 "
$ns at 39 " $n5 setdest 400 360 40 "
$ns at 5 " $n6 setdest 420 300 20 "
$ns at 11 " $n6 setdest 720 300 50 "
$ns at 48 " $n6 setdest 179 200 100 "
$ns at 5 " $n6 setdest 420 300 20 "
$ns at 15 " $n8 setdest 300 400 100 "

```

```

$ns at 20 " $n8 setdest 500 600 80 "
$ns at 40 " $n8 setdest 320 400 70 "
$ns at 37 " $n9 setdest 548 356 55 "
$ns at 6 " $n10 setdest 525 467 68 "
$ns at 52 " $n11 setdest 639 400 80 "
$ns at 10 " $n12 setdest 599 399 80 "
$ns at 48 " $n13 setdest 245 190 100 "
$ns at 55 " $n13 setdest 600 500 50 "
$ns at 5060 " $n13 setdest 100 380 100 "
$ns at 3 " $n13 setdest 290 200 70 "
$ns at 12 " $n13 setdest 330 632 80 "
$ns at 35 " $n13 setdest 467 754 70 "
$ns at 8 " $n13 setdest 152 342 80 "
$ns at 37 " $n13 setdest 502 337 70 "
$ns at 40 " $n13 setdest 800 200 98 "
$ns at 55 " $n14 setdest 570 360 80 "
$ns at 15 " $n15 setdest 506 354 70 "
$ns at 20 " $n15 setdest 389 534 45 "
$ns at 15 " $n15 setdest 210 550 70 "
$ns at 45 " $n15 setdest 100 300 100 "
$ns at 50 " $n16 setdest 630 320 100 "
$ns at 3 " $n17 setdest 850 300 50 "
$ns at 35 " $n17 setdest 420 345 70 "
$ns at 58 " $n18 setdest 389 600 70 "
$ns at 14 " $n20 setdest 144 466 100 "

```

```
#=====
```

```
# Agents Definition
```

```
#=====
```

```
#Setup a TCP connection
```

```
set tcp2 [new Agent/TCP]
```

```
$ns attach-agent $n2 $tcp2
```

```
set sink3 [new Agent/TCPSink]
```

```
$ns attach-agent $n13 $sink3
```

```

$ns connect $tcp2 $sink3
$tcp2 set packetSize_ 10000

#=====
#   Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp2
$ns at 1.0 "$ftp1 start"
$ns at 90.0 "$ftp1 stop"

#=====
#   Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run

```

## Anexo 5

### Código fuente de la simulación 5 con protocolo de enrutamiento DSDV

```

#=====
#   Simulation parameters setup
#=====
Phy/WirelessPhy set bandwidth_ 1Mb      ;#Data Rate
Mac/802_11 set dataRate_ 22Mb          ;#Rate for Data Frames
set val(chan) Channel/WirelessChannel  ;# channel type
set val(prop) Propagation/TwoRayGround  ;# radio-propagation model
set val(netif) Phy/WirelessPhy         ;# network interface type
set val(mac) Mac/802_11                 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue   ;# interface queue type
set val(ll) LL                           ;# link layer type
set val(ant) Antenna/OmniAntenna       ;# antenna model
set val(ifqlen) 20                       ;# max packet in ifq
set val(nn) 22                            ;# number of mobilenodes
set val(rp) DSDV                          ;# routing protocol
set val(x) 973                             ;# X dimension of topography
set val(y) 609                             ;# Y dimension of topography
set val(stop) 70.0                        ;# time of simulation end

#=====
#   Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]

```



```
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
```

```
#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile
```

```
#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel
```

```
#=====
```

```
# Mobile node parameter setup
```

```
#=====
```

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

```
#=====
```

```
# Nodes Definition
```

```
#=====
```

```
#Create 22 nodes
set n0 [$ns node]
$n0 set X_ 280
$n0 set Y_ 381
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 420
$n1 set Y_ 390
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 214
$n2 set Y_ 443
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 327
$n3 set Y_ 310
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 300
$n4 set Y_ 170
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 710
$n5 set Y_ 228
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 564
$n6 set Y_ 329
```

\$n6 set Z\_ 0.0  
\$ns initial\_node\_pos \$n6 20  
set n7 [\$ns node]  
\$n7 set X\_ 637  
\$n7 set Y\_ 175  
\$n7 set Z\_ 0.0  
\$ns initial\_node\_pos \$n7 20  
set n8 [\$ns node]  
\$n8 set X\_ 430  
\$n8 set Y\_ 318  
\$n8 set Z\_ 0.0  
\$ns initial\_node\_pos \$n8 20  
set n9 [\$ns node]  
\$n9 set X\_ 591  
\$n9 set Y\_ 241  
\$n9 set Z\_ 0.0  
\$ns initial\_node\_pos \$n9 20  
set n10 [\$ns node]  
\$n10 set X\_ 641  
\$n10 set Y\_ 326  
\$n10 set Z\_ 0.0  
\$ns initial\_node\_pos \$n10 20  
set n11 [\$ns node]  
\$n11 set X\_ 793  
\$n11 set Y\_ 181  
\$n11 set Z\_ 0.0  
\$ns initial\_node\_pos \$n11 20  
set n12 [\$ns node]  
\$n12 set X\_ 635  
\$n12 set Y\_ 392  
\$n12 set Z\_ 0.0  
\$ns initial\_node\_pos \$n12 20  
set n13 [\$ns node]  
\$n13 set X\_ 536

\$n13 set Y\_ 194  
\$n13 set Z\_ 0.0  
\$ns initial\_node\_pos \$n13 20  
set n14 [\$ns node]  
\$n14 set X\_ 859  
\$n14 set Y\_ 259  
\$n14 set Z\_ 0.0  
\$ns initial\_node\_pos \$n14 20  
set n15 [\$ns node]  
\$n15 set X\_ 312  
\$n15 set Y\_ 452  
\$n15 set Z\_ 0.0  
\$ns initial\_node\_pos \$n15 20  
set n16 [\$ns node]  
\$n16 set X\_ 873  
\$n16 set Y\_ 179  
\$n16 set Z\_ 0.0  
\$ns initial\_node\_pos \$n16 20  
set n17 [\$ns node]  
\$n17 set X\_ 369  
\$n17 set Y\_ 392  
\$n17 set Z\_ 0.0  
\$ns initial\_node\_pos \$n17 20  
set n18 [\$ns node]  
\$n18 set X\_ 731  
\$n18 set Y\_ 406  
\$n18 set Z\_ 0.0  
\$ns initial\_node\_pos \$n18 20  
set n19 [\$ns node]  
\$n19 set X\_ 345  
\$n19 set Y\_ 423  
\$n19 set Z\_ 0.0  
\$ns initial\_node\_pos \$n19 20  
set n20 [\$ns node]

```

$n20 set X_ 282
$n20 set Y_ 495
$n20 set Z_ 0.0
$ns initial_node_pos $n20 20
set n21 [$ns node]
$n21 set X_ 133
$n21 set Y_ 479
$n21 set Z_ 0.0
$ns initial_node_pos $n21 20

```

```

#=====
#   Generate movement
#=====
$ns at 14 " $n2 setdest 111 509 50 "
$ns at 15 " $n8 setdest 300 400 100 "
$ns at 14 " $n20 setdest 144 466 100 "

```

```

#=====
#   Agents Definition
#=====
#Setup a TCP connection
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set sink3 [new Agent/TCPSink]
$ns attach-agent $n13 $sink3
$ns connect $tcp2 $sink3
$tcp2 set packetSize_ 10000

```

```

#=====
#   Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp1 [new Application/FTP]

```

```
$ftp1 attach-agent $tcp2
```

```
$ns at 1.0 "$ftp1 start"
```

```
$ns at 70.0 "$ftp1 stop"
```

```
#=====
```

```
# Termination
```

```
#=====
```

```
#Define a 'finish' procedure
```

```
proc finish {} {
```

```
    global ns tracefile namfile
```

```
    $ns flush-trace
```

```
    close $tracefile
```

```
    close $namfile
```

```
    exec nam out.nam &
```

```
    exit 0
```

```
}
```

```
for {set i 0} {$i < $val(nn)} {incr i} {
```

```
    $ns at $val(stop) "\$n$i reset"
```

```
}
```

```
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
```

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

```
$ns at $val(stop) "puts \"done\" ; $ns halt"
```

```
$ns run
```

## Anexo 6

### Código fuente de la simulación 6 con protocolo de enrutamiento DSDV

```
#=====
# Simulation parameters setup
#=====
Phy/WirelessPhy set bandwidth_ 15Mb      ;#Data Rate
Mac/802_11 set dataRate_ 22Mb          ;#Rate for Data Frames
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy        ;# network interface type
set val(mac) Mac/802_11                ;# MAC type
set val(ifq) Queue/DropTail/PriQueue   ;# interface queue type
set val(ll) LL                          ;# link layer type
set val(ant) Antenna/OmniAntenna       ;# antenna model
set val(ifqlen) 30                      ;# max packet in ifq
set val(nn) 22                          ;# number of mobilenodes
set val(rp) DSDV                        ;# routing protocol
set val(x) 973                          ;# X dimension of topography
set val(y) 609                          ;# Y dimension of topography
set val(stop) 70.0                      ;# time of simulation end

#=====
# Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
```

```

set topo    [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#=====
#   Mobile node parameter setup
#=====
$ns node-config -adhocRouting $val(rp) \
    -llType    $val(ll) \
    -macType    $val(mac) \
    -ifqType    $val(ifq) \
    -ifqLen    $val(ifqlen) \
    -antType    $val(ant) \
    -propType    $val(prop) \
    -phyType    $val(netif) \
    -channel    $chan \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace    ON \
    -movementTrace ON

#=====
#   Nodes Definition

```



```
#=====
#Create 22 nodes
set n0 [$ns node]
$n0 set X_ 280
$n0 set Y_ 381
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 420
$n1 set Y_ 390
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 214
$n2 set Y_ 443
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 327
$n3 set Y_ 310
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 300
$n4 set Y_ 170
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 710
$n5 set Y_ 228
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 564
```

\$n6 set Y\_ 329  
\$n6 set Z\_ 0.0  
\$ns initial\_node\_pos \$n6 20  
set n7 [\$ns node]  
\$n7 set X\_ 637  
\$n7 set Y\_ 175  
\$n7 set Z\_ 0.0  
\$ns initial\_node\_pos \$n7 20  
set n8 [\$ns node]  
\$n8 set X\_ 430  
\$n8 set Y\_ 318  
\$n8 set Z\_ 0.0  
\$ns initial\_node\_pos \$n8 20  
set n9 [\$ns node]  
\$n9 set X\_ 591  
\$n9 set Y\_ 241  
\$n9 set Z\_ 0.0  
\$ns initial\_node\_pos \$n9 20  
set n10 [\$ns node]  
\$n10 set X\_ 641  
\$n10 set Y\_ 326  
\$n10 set Z\_ 0.0  
\$ns initial\_node\_pos \$n10 20  
set n11 [\$ns node]  
\$n11 set X\_ 793  
\$n11 set Y\_ 181  
\$n11 set Z\_ 0.0  
\$ns initial\_node\_pos \$n11 20  
set n12 [\$ns node]  
\$n12 set X\_ 635  
\$n12 set Y\_ 392  
\$n12 set Z\_ 0.0  
\$ns initial\_node\_pos \$n12 20  
set n13 [\$ns node]

\$n13 set X\_ 536  
\$n13 set Y\_ 194  
\$n13 set Z\_ 0.0  
\$ns initial\_node\_pos \$n13 20  
set n14 [\$ns node]  
\$n14 set X\_ 859  
\$n14 set Y\_ 259  
\$n14 set Z\_ 0.0  
\$ns initial\_node\_pos \$n14 20  
set n15 [\$ns node]  
\$n15 set X\_ 312  
\$n15 set Y\_ 452  
\$n15 set Z\_ 0.0  
\$ns initial\_node\_pos \$n15 20  
set n16 [\$ns node]  
\$n16 set X\_ 873  
\$n16 set Y\_ 179  
\$n16 set Z\_ 0.0  
\$ns initial\_node\_pos \$n16 20  
set n17 [\$ns node]  
\$n17 set X\_ 369  
\$n17 set Y\_ 392  
\$n17 set Z\_ 0.0  
\$ns initial\_node\_pos \$n17 20  
set n18 [\$ns node]  
\$n18 set X\_ 731  
\$n18 set Y\_ 406  
\$n18 set Z\_ 0.0  
\$ns initial\_node\_pos \$n18 20  
set n19 [\$ns node]  
\$n19 set X\_ 345  
\$n19 set Y\_ 423  
\$n19 set Z\_ 0.0  
\$ns initial\_node\_pos \$n19 20

```

set n20 [$ns node]
$n20 set X_ 282
$n20 set Y_ 495
$n20 set Z_ 0.0
$ns initial_node_pos $n20 20
set n21 [$ns node]
$n21 set X_ 133
$n21 set Y_ 479
$n21 set Z_ 0.0
$ns initial_node_pos $n21 20

#=====
#   Generate movement
#=====
$ns at 14 " $n2 setdest 111 509 50 "
$ns at 15 " $n8 setdest 300 400 100 "
$ns at 14 " $n20 setdest 144 466 100 "

#=====
#   Agents Definition
#=====
#Setup a TCP connection
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set sink3 [new Agent/TCPSink]
$ns attach-agent $n13 $sink3
$ns connect $tcp2 $sink3
$tcp2 set packetSize_ 500

#=====
#   Applications Definition
#=====
#Setup a FTP Application over TCP connection

```

```

set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp2
$ns at 1.0 "$ftp1 start"
$ns at 70.0 "$ftp1 stop"

#=====
#   Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run

```

## Anexo 7

### Código fuente de la simulación 7 con protocolo de enrutamiento DSDV

```

#=====
#   Simulation parameters setup
#=====
Phy/WirelessPhy set bandwidth_ 10Mb      ;#Data Rate
Mac/802_11 set dataRate_ 54Mb          ;#Rate for Data Frames
set val(chan) Channel/WirelessChannel  ;# channel type
set val(prop) Propagation/TwoRayGround  ;# radio-propagation model
set val(netif) Phy/WirelessPhy         ;# network interface type
set val(mac) Mac/802_11                 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue   ;# interface queue type
set val(ll) LL                           ;# link layer type
set val(ant) Antenna/OmniAntenna       ;# antenna model
set val(ifqlen) 70                       ;# max packet in ifq
set val(nn) 22                            ;# number of mobilenodes
set val(rp) DSDV                          ;# routing protocol
set val(x) 973                            ;# X dimension of topography
set val(y) 854                            ;# Y dimension of topography
set val(stop) 90.0                       ;# time of simulation end

#=====
#   Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]

```

```
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
```

```
#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile
```

```
#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel
```

```
#=====
```

```
# Mobile node parameter setup
```

```
#=====
```

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

```
#=====
```

```
# Nodes Definition
```

```
#=====
```

```
#Create 22 nodes
set n0 [$ns node]
$n0 set X_ 280
$n0 set Y_ 381
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 420
$n1 set Y_ 390
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 214
$n2 set Y_ 443
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 327
$n3 set Y_ 310
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 300
$n4 set Y_ 170
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 710
$n5 set Y_ 228
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 564
$n6 set Y_ 329
```



\$n6 set Z\_ 0.0  
\$ns initial\_node\_pos \$n6 20  
set n7 [\$ns node]  
\$n7 set X\_ 637  
\$n7 set Y\_ 175  
\$n7 set Z\_ 0.0  
\$ns initial\_node\_pos \$n7 20  
set n8 [\$ns node]  
\$n8 set X\_ 430  
\$n8 set Y\_ 318  
\$n8 set Z\_ 0.0  
\$ns initial\_node\_pos \$n8 20  
set n9 [\$ns node]  
\$n9 set X\_ 591  
\$n9 set Y\_ 241  
\$n9 set Z\_ 0.0  
\$ns initial\_node\_pos \$n9 20  
set n10 [\$ns node]  
\$n10 set X\_ 641  
\$n10 set Y\_ 326  
\$n10 set Z\_ 0.0  
\$ns initial\_node\_pos \$n10 20  
set n11 [\$ns node]  
\$n11 set X\_ 793  
\$n11 set Y\_ 181  
\$n11 set Z\_ 0.0  
\$ns initial\_node\_pos \$n11 20  
set n12 [\$ns node]  
\$n12 set X\_ 635  
\$n12 set Y\_ 392  
\$n12 set Z\_ 0.0  
\$ns initial\_node\_pos \$n12 20  
set n13 [\$ns node]  
\$n13 set X\_ 536

\$n13 set Y\_ 194  
\$n13 set Z\_ 0.0  
\$ns initial\_node\_pos \$n13 20  
set n14 [\$ns node]  
\$n14 set X\_ 859  
\$n14 set Y\_ 259  
\$n14 set Z\_ 0.0  
\$ns initial\_node\_pos \$n14 20  
set n15 [\$ns node]  
\$n15 set X\_ 312  
\$n15 set Y\_ 452  
\$n15 set Z\_ 0.0  
\$ns initial\_node\_pos \$n15 20  
set n16 [\$ns node]  
\$n16 set X\_ 873  
\$n16 set Y\_ 179  
\$n16 set Z\_ 0.0  
\$ns initial\_node\_pos \$n16 20  
set n17 [\$ns node]  
\$n17 set X\_ 369  
\$n17 set Y\_ 392  
\$n17 set Z\_ 0.0  
\$ns initial\_node\_pos \$n17 20  
set n18 [\$ns node]  
\$n18 set X\_ 731  
\$n18 set Y\_ 406  
\$n18 set Z\_ 0.0  
\$ns initial\_node\_pos \$n18 20  
set n19 [\$ns node]  
\$n19 set X\_ 345  
\$n19 set Y\_ 423  
\$n19 set Z\_ 0.0  
\$ns initial\_node\_pos \$n19 20  
set n20 [\$ns node]

```

$n20 set X_ 282
$n20 set Y_ 495
$n20 set Z_ 0.0
$ns initial_node_pos $n20 20
set n21 [$ns node]
$n21 set X_ 133
$n21 set Y_ 479
$n21 set Z_ 0.0
$ns initial_node_pos $n21 20

```

```

#=====

```

```

#   Generate movement

```

```

#=====

```

```

$ns at 16 " $n0 setdest 225 550 90 "
$ns at 13 " $n1 setdest 300 311 58 "
$ns at 14 " $n2 setdest 111 509 50 "
$ns at 30 " $n2 setdest 500 356 60 "
$ns at 62 " $n2 setdest 316 198 60 "
$ns at 30 " $n2 setdest 500 356 60 "
$ns at 6 " $n2 setdest 90 357 78 "
$ns at 20 " $n2 setdest 345 487 50 "
$ns at 30 " $n2 setdest 530 335 56 "
$ns at 10 " $n3 setdest 470 200 70 "
$ns at 13 " $n3 setdest 450 400 40 "
$ns at 18 " $n3 setdest 500 300 70 "
$ns at 5 " $n4 setdest 550 315 50 "
$ns at 47 " $n4 setdest 234 378 30 "
$ns at 55 " $n4 setdest 534 236 60 "
$ns at 5 " $n4 setdest 550 315 50 "
$ns at 39 " $n5 setdest 400 360 40 "
$ns at 5 " $n6 setdest 420 300 20 "
$ns at 11 " $n6 setdest 720 300 50 "
$ns at 48 " $n6 setdest 179 200 100 "
$ns at 5 " $n6 setdest 420 300 20 "

```

```

$ns at 15 " $n8 setdest 300 400 100 "
$ns at 20 " $n8 setdest 500 600 80 "
$ns at 40 " $n8 setdest 320 400 70 "
$ns at 37 " $n9 setdest 548 356 55 "
$ns at 6 " $n10 setdest 525 467 68 "
$ns at 52 " $n11 setdest 639 400 80 "
$ns at 10 " $n12 setdest 599 399 80 "
$ns at 48 " $n13 setdest 245 190 100 "
$ns at 55 " $n13 setdest 600 500 50 "
$ns at 5060 " $n13 setdest 100 380 100 "
$ns at 3 " $n13 setdest 290 200 70 "
$ns at 12 " $n13 setdest 330 632 80 "
$ns at 35 " $n13 setdest 467 754 70 "
$ns at 8 " $n13 setdest 152 342 80 "
$ns at 37 " $n13 setdest 502 337 70 "
$ns at 40 " $n13 setdest 800 200 98 "
$ns at 55 " $n14 setdest 570 360 80 "
$ns at 15 " $n15 setdest 506 354 70 "
$ns at 20 " $n15 setdest 389 534 45 "
$ns at 15 " $n15 setdest 210 550 70 "
$ns at 45 " $n15 setdest 100 300 100 "
$ns at 50 " $n16 setdest 630 320 100 "
$ns at 3 " $n17 setdest 850 300 50 "
$ns at 35 " $n17 setdest 420 345 70 "
$ns at 58 " $n18 setdest 389 600 70 "
$ns at 14 " $n20 setdest 144 466 100 "

```

```
#=====
```

```
# Agents Definition
```

```
#=====
```

```
#Setup a TCP connection
```

```
set tcp2 [new Agent/TCP]
```

```
$ns attach-agent $n2 $tcp2
```

```
set sink3 [new Agent/TCPSink]
```

```

$ns attach-agent $n13 $sink3
$ns connect $tcp2 $sink3
$tcp2 set packetSize_ 10000

```

```

#=====

```

```

#   Applications Definition

```

```

#=====

```

```

#Setup a FTP Application over TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp2
$ns at 1.0 "$ftp1 start"
$ns at 90.0 "$ftp1 stop"

```

```

#=====

```

```

#   Termination

```

```

#=====

```

```

#Define a 'finish' procedure

```

```

proc finish {} {

```

```

    global ns tracefile namfile

```

```

    $ns flush-trace

```

```

    close $tracefile

```

```

    close $namfile

```

```

    exec nam out.nam &

```

```

    exit 0

```

```

}

```

```

for {set i 0} {$i < $val(nn)} {incr i} {

```

```

    $ns at $val(stop) "\$n$i reset"

```

```

}

```

```

$ns at $val(stop) "$ns nam-end-wireless $val(stop)"

```

```

$ns at $val(stop) "finish"

```

```

$ns at $val(stop) "puts \"done\" ; $ns halt"

```

```

$ns run

```



## Anexo 8

### Código fuente de la simulación 8 con protocolo de enrutamiento DSDV

```

#=====
# Simulation parameters setup
#=====
setval(chan) Channel/WirelessChannel ;# channel type
setval(prop) Propagation/TwoRayGround ;# radio-propagation model
setval(netif) Phy/WirelessPhy ;# network interface type
setval(mac) Mac/802_11 ;# MAC type
setval(ifq) Queue/DropTail/PriQueue ;# interface queue type
setval(ll) LL ;# link layer type
setval(ant) Antenna/OmniAntenna ;# antenna model
setval(ifqlen) 50 ;# max packet in ifq
setval(nn) 9 ;# number of mobilenodes
setval(rp) DSDV ;# routing protocol
setval(x) 970 ;# X dimension of topography
setval(y) 623 ;# Y dimension of topography
setval(stop) 10.0 ;# time of simulation end

#=====
# Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
settopo [new Topography]
$topoload_flatgrid $val(x) $val(y)
create-god $val(nn)

```

#Open the NS trace file

settracefile [open out.tr w]

\$ns trace-all \$tracefile

#Open the NAM trace file

setnamfile [open out.nam w]

\$ns namtrace-all \$namfile

\$ns namtrace-all-wireless \$namfile \$val(x) \$val(y)

setchan [new \$val(chan)];#Create wireless channel

#=====

# Mobile node parameter setup

#=====

\$ns node-config -adhocRouting \$val(rp) \

    -IIType \$val(II) \

    -macType \$val(mac) \

    -ifqType \$val(ifq) \

    -ifqLen \$val(ifqlen) \

    -antType \$val(ant) \

    -propType \$val(prop) \

    -phyType \$val(netif) \

    -channel \$chan \

    -topoInstance \$topo \

    -agentTrace ON \

    -routerTrace ON \

    -macTrace ON \

    -movementTrace ON

#=====

# Nodes Definition

#=====

#Create 9 nodes

set n0 [\$ns node]

\$n0 set X\_ 380



\$n0 set Y\_ 248  
\$n0 set Z\_ 0.0  
\$ns initial\_node\_pos \$n0 20  
set n1 [\$ns node]  
\$n1 set X\_ 179  
\$n1 set Y\_ 308  
\$n1 set Z\_ 0.0  
\$ns initial\_node\_pos \$n1 20  
set n2 [\$ns node]  
\$n2 set X\_ 578  
\$n2 set Y\_ 337  
\$n2 set Z\_ 0.0  
\$ns initial\_node\_pos \$n2 20  
set n3 [\$ns node]  
\$n3 set X\_ 329  
\$n3 set Y\_ 426  
\$n3 set Z\_ 0.0  
\$ns initial\_node\_pos \$n3 20  
set n4 [\$ns node]  
\$n4 set X\_ 516  
\$n4 set Y\_ 513  
\$n4 set Z\_ 0.0  
\$ns initial\_node\_pos \$n4 20  
set n5 [\$ns node]  
\$n5 set X\_ 706  
\$n5 set Y\_ 510  
\$n5 set Z\_ 0.0  
\$ns initial\_node\_pos \$n5 20  
set n6 [\$ns node]  
\$n6 set X\_ 870  
\$n6 set Y\_ 288  
\$n6 set Z\_ 0.0  
\$ns initial\_node\_pos \$n6 20  
set n7 [\$ns node]

```

$n7 set X_ 743
$n7 set Y_ 307
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 406
$n8 set Y_ 363
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20

```

```

#=====
#   Generate movement
#=====
$ns at 0.2 " $n0 setdest 600 400 150 "
$ns at 3.2 " $n0 setdest 178 307 200 "
$ns at 3.2 " $n1 setdest 400 400 200 "
$ns at 0.2 " $n2 setdest 310 399 150 "
$ns at 1,5 " $n2 setdest 497 379 150 "
$ns at 5.8 " $n2 setdest 200 169 200 "
$ns at 0.5 " $n6 setdest 420 523 180 "
$ns at 0.2 " $n7 setdest 379 247 160 "

```

```

#=====
#   Agents Definition
#=====
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n2 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500

```

```

#=====
#   Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 20.0 "$ftp0 stop"

#=====
#   Termination
#=====
#Define a 'finish' procedure
proc finish {} {
global ns tracefile namfile
    $ns flush-trace
close $tracefile
close $namfile
execnamout.nam&
exit 0
}
for {set i 0} {$i < $val(nn)} { incre } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$nsrun

```

## Anexo 9

### Código fuente de la simulación 9 con protocolo de enrutamiento DSR

```
#=====
# Simulation parameters setup
#=====
Phy/WirelessPhy set bandwidth_ 2Mb      ;#Data Rate
Mac/802_11 set dataRate_ 11Mb          ;#Rate for Data Frames
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy        ;# network interface type
set val(mac) Mac/802_11                ;# MAC type
set val(ifq) Queue/DropTail/PriQueue   ;# interface queue type
set val(ll) LL                          ;# link layer type
set val(ant) Antenna/OmniAntenna      ;# antenna model
set val(ifqlen) 50                      ;# max packet in ifq
set val(nn) 13                          ;# number of mobilenodes
set val(rp) DSR                         ;# routing protocol
set val(x) 849                          ;# X dimension of topography
set val(y) 558                          ;# Y dimension of topography
set val(stop) 60.0                      ;# time of simulation end

#=====
# Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
```

```

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#=====
#   Mobile node parameter setup
#=====
$ns node-config -adhocRouting $val(rp) \
    -llType    $val(ll) \
    -macType    $val(mac) \
    -ifqType    $val(ifq) \
    -ifqLen    $val(ifqlen) \
    -antType    $val(ant) \
    -propType    $val(prop) \
    -phyType    $val(netif) \
    -channel    $chan \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace    ON \
    -movementTrace ON

#=====
#   Nodes Definition
#=====
#Create 13 nodes
set n0 [$ns node]
$n0 set X_ 594

```

```
$n0 set Y_ 374
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 748
$n1 set Y_ 282
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 645
$n2 set Y_ 314
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 463
$n3 set Y_ 233
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 603
$n4 set Y_ 321
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 533
$n5 set Y_ 340
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 749
$n6 set Y_ 334
$n6 set Z_ 0.0
$ns initial_node_pos $n6 20
set n7 [$ns node]
```

```
$n7 set X_ 651
$n7 set Y_ 358
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 560
$n8 set Y_ 246
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20
set n9 [$ns node]
$n9 set X_ 489
$n9 set Y_ 299
$n9 set Z_ 0.0
$ns initial_node_pos $n9 20
set n10 [$ns node]
$n10 set X_ 444
$n10 set Y_ 343
$n10 set Z_ 0.0
$ns initial_node_pos $n10 20
set n11 [$ns node]
$n11 set X_ 627
$n11 set Y_ 194
$n11 set Z_ 0.0
$ns initial_node_pos $n11 20
set n12 [$ns node]
$n12 set X_ 618
$n12 set Y_ 458
$n12 set Z_ 0.0
$ns initial_node_pos $n12 20
```

```
#=====
```

```
# Agents Definition
```

```
#=====
```

```
#Setup a TCP connection
```

```

set tcp0 [new Agent/TCP]
$ns attach-agent $n3 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n6 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 800

#=====
#   Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp0
$ns at 1.0 "$ftp3 start"
$ns at 60.0 "$ftp3 stop"

#=====
#   Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"

```



\$ns at \$val(stop) "finish"

\$ns at \$val(stop) "puts \"done\" ; \$ns halt"

\$ns run

## Anexo 10

### Código fuente de la simulación 10 con protocolo de enrutamiento DSR

```

#=====
# Simulation parameters setup
#=====
Phy/WirelessPhy set bandwidth_ 2Mb      ;#Data Rate
Mac/802_11 set dataRate_ 11Mb          ;#Rate for Data Frames
set val(chan) Channel/WirelessChannel  ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy        ;# network interface type
set val(mac) Mac/802_11                ;# MAC type
set val(ifq) Queue/DropTail/PriQueue   ;# interface queue type
set val(ll) LL                          ;# link layer type
set val(ant) Antenna/OmniAntenna       ;# antenna model
set val(ifqlen) 40                      ;# max packet in ifq
set val(nn) 13                          ;# number of mobilenodes
set val(rp) DSR                         ;# routing protocol
set val(x) 849                          ;# X dimension of topography
set val(y) 558                          ;# Y dimension of topography
set val(stop) 60.0                      ;# time of simulation end

#=====
# Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

```

```

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#=====
#   Mobile node parameter setup
#=====
$ns node-config -adhocRouting $val(rp) \
    -llType    $val(ll) \
    -macType   $val(mac) \
    -ifqType   $val(ifq) \
    -ifqLen    $val(ifqlen) \
    -antType   $val(ant) \
    -propType  $val(prop) \
    -phyType   $val(netif) \
    -channel   $chan \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace  ON \
    -movementTrace ON

#=====
#   Nodes Definition
#=====
#Create 13 nodes
set n0 [$ns node]
$n0 set X_ 594

```

```
$n0 set Y_ 374
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 748
$n1 set Y_ 282
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 645
$n2 set Y_ 314
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 463
$n3 set Y_ 233
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 603
$n4 set Y_ 321
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 533
$n5 set Y_ 340
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 749
$n6 set Y_ 334
$n6 set Z_ 0.0
$ns initial_node_pos $n6 20
set n7 [$ns node]
```

```

$n7 set X_ 651
$n7 set Y_ 358
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 560
$n8 set Y_ 246
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20
set n9 [$ns node]
$n9 set X_ 489
$n9 set Y_ 299
$n9 set Z_ 0.0
$ns initial_node_pos $n9 20
set n10 [$ns node]
$n10 set X_ 444
$n10 set Y_ 343
$n10 set Z_ 0.0
$ns initial_node_pos $n10 20
set n11 [$ns node]
$n11 set X_ 627
$n11 set Y_ 194
$n11 set Z_ 0.0
$ns initial_node_pos $n11 20
set n12 [$ns node]
$n12 set X_ 618
$n12 set Y_ 458
$n12 set Z_ 0.0
$ns initial_node_pos $n12 20

#=====
#   Generate movement
#=====
$ns at 8 " $n8 setdest 604 262 30 "

```

```
$ns at 13 " $n8 setdest 614 340 80 "
```

```
$ns at 14 " $n8 setdest 554 392 80 "
```

```
$ns at 14 " $n4 setdest 604 263 80 "
```

```
$ns at 28 " $n4 setdest 560 246 80 "
```

```
#=====
```

```
# Agents Definition
```

```
#=====
```

```
#Setup a TCP connection
```

```
set tcp0 [new Agent/TCP]
```

```
$ns attach-agent $n3 $tcp0
```

```
set sink1 [new Agent/TCPSink]
```

```
$ns attach-agent $n6 $sink1
```

```
$ns connect $tcp0 $sink1
```

```
$tcp0 set packetSize_ 800
```

```
#=====
```

```
# Applications Definition
```

```
#=====
```

```
#Setup a FTP Application over TCP connection
```

```
set ftp3 [new Application/FTP]
```

```
$ftp3 attach-agent $tcp0
```

```
$ns at 1.0 "$ftp3 start"
```

```
$ns at 60.0 "$ftp3 stop"
```

```
#=====
```

```
# Termination
```

```
#=====
```

```
#Define a 'finish' procedure
```

```
proc finish {} {
```

```
    global ns tracefile namfile
```

```
$ns flush-trace
close $tracefile
close $namfile
exec nam out.nam &
exit 0
}
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at $val(stop) "\n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

## Anexo 11

### Código fuente de la simulación 11 con protocolo de enrutamiento DSR

```

#=====
#   Simulation parameters setup
#=====
Phy/WirelessPhy set bandwidth_ 5Mb      ;#Data Rate
Mac/802_11 set dataRate_ 54Mb          ;#Rate for Data Frames
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy        ;# network interface type
set val(mac) Mac/802_11                ;# MAC type
set val(ifq) Queue/DropTail/PriQueue  ;# interface queue type
set val(ll) LL                          ;# link layer type
set val(ant) Antenna/OmniAntenna      ;# antenna model
set val(ifqlen) 20                      ;# max packet in ifq
set val(nn) 13                          ;# number of mobilenodes
set val(rp) DSR                          ;# routing protocol
set val(x) 849                          ;# X dimension of topography
set val(y) 558                          ;# Y dimension of topography
set val(stop) 60.0                      ;# time of simulation end

#=====
#   Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

```



```

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#=====
#   Mobile node parameter setup
#=====
$ns node-config -adhocRouting $val(rp) \
    -llType    $val(ll) \
    -macType   $val(mac) \
    -ifqType   $val(ifq) \
    -ifqLen    $val(ifqlen) \
    -antType   $val(ant) \
    -propType  $val(prop) \
    -phyType   $val(netif) \
    -channel   $chan \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace  ON \
    -movementTrace ON

#=====
#   Nodes Definition
#=====
#Create 13 nodes
set n0 [$ns node]
$n0 set X_ 594

```

```
$n0 set Y_ 374
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 748
$n1 set Y_ 282
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 645
$n2 set Y_ 314
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 463
$n3 set Y_ 233
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 603
$n4 set Y_ 321
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 533
$n5 set Y_ 340
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 749
$n6 set Y_ 334
$n6 set Z_ 0.0
$ns initial_node_pos $n6 20
set n7 [$ns node]
```

```

$n7 set X_ 651
$n7 set Y_ 358
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 560
$n8 set Y_ 246
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20
set n9 [$ns node]
$n9 set X_ 489
$n9 set Y_ 299
$n9 set Z_ 0.0
$ns initial_node_pos $n9 20
set n10 [$ns node]
$n10 set X_ 444
$n10 set Y_ 343
$n10 set Z_ 0.0
$ns initial_node_pos $n10 20
set n11 [$ns node]
$n11 set X_ 627
$n11 set Y_ 194
$n11 set Z_ 0.0
$ns initial_node_pos $n11 20
set n12 [$ns node]
$n12 set X_ 618
$n12 set Y_ 458
$n12 set Z_ 0.0
$ns initial_node_pos $n12 20
#=====
#   Generate movement
#=====
$ns at 8 " $n8 setdest 604 262 30 "
$ns at 13 " $n8 setdest 614 340 80 "

```

```
$ns at 14 " $n8 setdest 554 392 80 "
```

```
$ns at 14 " $n4 setdest 604 263 80 "
```

```
$ns at 28 " $n4 setdest 560 246 80 "
```

```
#=====
```

```
#=====
```

```
# Agents Definition
```

```
#=====
```

```
#Setup a TCP connection
```

```
set tcp0 [new Agent/TCP]
```

```
$ns attach-agent $n3 $tcp0
```

```
set sink1 [new Agent/TCPSink]
```

```
$ns attach-agent $n6 $sink1
```

```
$ns connect $tcp0 $sink1
```

```
$tcp0 set packetSize_ 5000
```

```
#=====
```

```
# Applications Definition
```

```
#=====
```

```
#Setup a FTP Application over TCP connection
```

```
set ftp3 [new Application/FTP]
```

```
$ftp3 attach-agent $tcp0
```

```
$ns at 1.0 "$ftp3 start"
```

```
$ns at 60.0 "$ftp3 stop"
```

```
#=====
```

```
# Termination
```

```
#=====
```

```
#Define a 'finish' procedure
```

```
proc finish {} {
```

```
    global ns tracefile namfile
```

```
$ns flush-trace
close $tracefile
close $namfile
exec nam out.nam &
exit 0
}
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at $val(stop) "\n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

## Anexo 12

### Código fuente de la simulación 12 con protocolo de enrutamiento DSR

```

#=====
#  Simulation parameters setup
#=====
setval(chan) Channel/WirelessChannel ;# channel type
setval(prop) Propagation/TwoRayGround ;# radio-propagation model
setval(netif) Phy/WirelessPhy ;# network interface type
setval(mac) Mac/802_11 ;# MAC type
setval(ifq) Queue/DropTail/PriQueue ;# interface queue type
setval(ll) LL ;# link layer type
setval(ant) Antenna/OmniAntenna ;# antenna model
setval(ifqlen) 50 ;# max packet in ifq
setval(nn) 9 ;# number of mobilenodes
setval(rp) DSR ;# routing protocol
setval(x) 987 ;# X dimension of topography
setval(y) 631 ;# Y dimension of topography
setval(stop) 10.0 ;# time of simulation end

#=====
#  Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
settopo [new Topography]
$topoload_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
settracefile [open out.tr w]

```

```
$ns trace-all $tracefile
```

```
#Open the NAM trace file
```

```
setnamfile [open out.nam w]
```

```
$ns namtrace-all $namfile
```

```
$ns namtrace-all-wireless $namfile $val(x) $val(y)
```

```
setchan [new $val(chan)];#Create wireless channel
```

```
#=====
```

```
# Mobile node parameter setup
```

```
#=====
```

```
$ns node-config -adhocRouting $val(rp) \
```

```
    -llType    $val(ll) \
```

```
    -macType   $val(mac) \
```

```
    -ifqType   $val(ifq) \
```

```
    -ifqLen    $val(ifqlen) \
```

```
    -antType   $val(ant) \
```

```
    -propType  $val(prop) \
```

```
    -phyType   $val(netif) \
```

```
    -channel   $chan \
```

```
    -topoInstance $topo \
```

```
    -agentTrace ON \
```

```
    -routerTrace ON \
```

```
    -macTrace  ON \
```

```
    -movementTrace ON
```

```
#=====
```

```
# Nodes Definition
```

```
#=====
```

```
#Create 9 nodes
```

```
set n0 [$ns node]
```

```
$n0 set X_ 301
```

```
$n0 set Y_ 300
```

```
$n0 set Z_ 0.0
```

```
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 437
$n1 set Y_ 389
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 887
$n2 set Y_ 390
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 457
$n3 set Y_ 139
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 697
$n4 set Y_ 200
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 622
$n5 set Y_ 425
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 419
$n6 set Y_ 531
$n6 set Z_ 0.0
$ns initial_node_pos $n6 20
set n7 [$ns node]
$n7 set X_ 251
$n7 set Y_ 370
```



```

$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 579
$n8 set Y_ 518
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20

#=====
#   Generate movement
#=====
$ns at 3.5 " $n1 setdest 490 400 180 "
$ns at 2.0 " $n2 setdest 405 220 180 "
$ns at 2.0 " $n4 setdest 400 400 150 "
$ns at 5.0 " $n4 setdest 540 230 180 "

#=====
#   Agents Definition
#=====
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2
$ns connect $tcp0 $sink2
$tcp0 set packetSize_ 1500

#=====
#   Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0

```

```
$ns at 1.0 "$ftp0 start"
```

```
$ns at 20.0 "$ftp0 stop"
```

```
#=====
```

```
# Termination
```

```
#=====
```

```
#Define a 'finish' procedure
```

```
proc finish {} {
```

```
global ns tracefilenamfile
```

```
    $ns flush-trace
```

```
close $tracefile
```

```
close $namfile
```

```
execnamout.nam&
```

```
exit 0
```

```
}
```

```
for {set i 0} {$i < $val(nn)} { incre } {
```

```
    $ns at $val(stop) "\n$i reset"
```

```
}
```

```
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
```

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

```
$ns at $val(stop) "puts \"done\" ; $ns halt"
```

```
$nsrun
```

## Anexo 13

### MANUAL DE INSTALACIÓN DEL PROGRAMA SIMULADOR DE REDES AD-HOC NS-2

#### Procedimiento

1. Descargar el programa de simulador de redes NS-2 de la siguiente pagina, este programa es de libre uso originalmente creado por la Universidad de Berkeley, este software incluye los paquetes

- Ns (simulador)
- Nam (visualizador de paquetes)
- Bibliotecas del simulador
- Tcl (un lenguaje scripting)
- Otcl ( las extensiones que van orientadas a los objetos)
- TclCl (las bibliotecas de TCL)

2. Extraer el archivo que viene comprimido en (.zip) y copiar en la carpeta home de Linux.

Para realizar la descarga del programa se puede realizar desde el siguiente enlace:

[https://sourceforge.net/projects/nsnam/?source=typ\\_redirect](https://sourceforge.net/projects/nsnam/?source=typ_redirect)

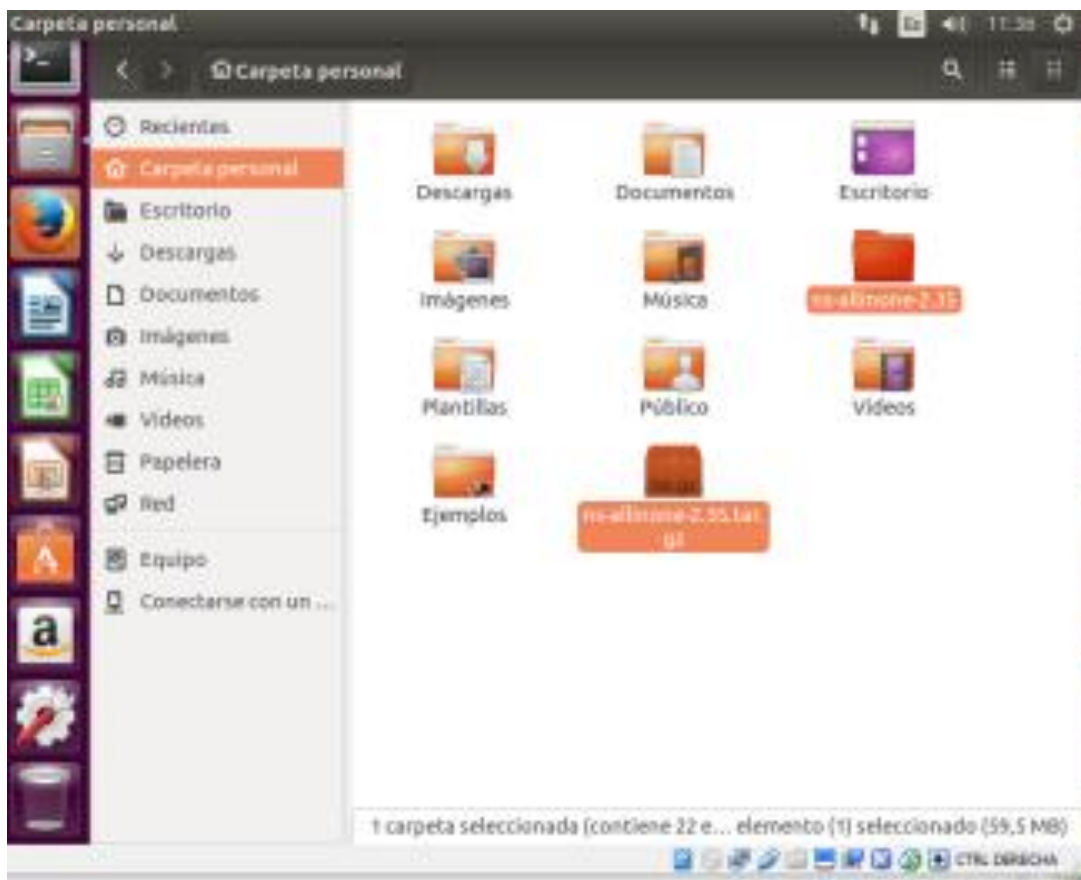


Figura 1. Extracción del archivo descargado

3. Abrir una terminal para realizar una actualización de librerías.
  - Sudo apt-getupdate

```
xavier@xavier-VirtualBox:~$ sudo apt-get update
Des:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Obj:2 http://ec.archive.ubuntu.com/ubuntu xenial InRelease
Des:3 http://ec.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Des:4 http://security.ubuntu.com/ubuntu xenial-security/main i386 DEP-11 Metad
ata [66,9 kB]
Des:5 http://security.ubuntu.com/ubuntu xenial-security/main DEP-11 64x64 Icons
[38,2 kB]
Des:6 http://security.ubuntu.com/ubuntu xenial-security/universe i386 DEP-11 Met
adata [19,4 kB]
Des:7 http://security.ubuntu.com/ubuntu xenial-security/universe DEP-11 64x64 Ic
ons [25,8 kB]
Des:8 http://security.ubuntu.com/ubuntu xenial-security/multiverse i386 DEP-11 M
etadata [212 B]
Des:9 http://ec.archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Des:10 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main i386 Packages [42
0 kB]
Des:11 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main i386 DEP-11 Metad
ata [293 kB]
77% [10 Packages store 0 B] [11 Components-i386 0 B/293 kB OK]
```

Figura 2. Actualización de las librerías

4. Instalamos los paquetes requeridos por NS-2 con el siguiente comando

- Sudo apt-get install build-essential autoconf automake.

```
xavier@xavier-VirtualBox:~$ sudo apt-get install build-essential autoconf automake
Des:11 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main i386 DEP-11 Metad
ata [293 kB]
Des:12 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main DEP-11 64x64 Icon
s [106 kB]
Des:13 http://ec.archive.ubuntu.com/ubuntu xenial-updates/universe i386 Packages
[362 kB]
Des:14 http://ec.archive.ubuntu.com/ubuntu xenial-updates/universe i386 DEP-11 M
etadata [119 kB]
Des:15 http://ec.archive.ubuntu.com/ubuntu xenial-updates/universe DEP-11 64x64
Icons [137 kB]
Des:16 http://ec.archive.ubuntu.com/ubuntu xenial-updates/multiverse i386 DEP-11
Metadata [2.518 B]
Des:17 http://ec.archive.ubuntu.com/ubuntu xenial-backports/main i386 DEP-11 Met
adata [288 B]
Des:18 http://ec.archive.ubuntu.com/ubuntu xenial-backports/universe i386 DEP-11
Metadata [212 B]
Des:19 http://ec.archive.ubuntu.com/ubuntu xenial-backports/multiverse i386 DEP-
11 Metadata [212 B]
Descargados 1.978 kB en 4s (477 kB/s)
AppStream cache update completed, but some metadata was ignored due to errors.
Leyendo lista de paquetes... Hecho
xavier@xavier-VirtualBox:~$ sudo apt-get install build-essential autoconf automa
ke
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
autoconf ya está en su versión más reciente (2.69-9).
fijado autoconf como instalado manualmente.
automake ya está en su versión más reciente (1:1.15-4ubuntu1).
build-essential ya está en su versión más reciente (12.ubuntu2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 440 no actualizados.
xavier@xavier-VirtualBox:~$
```

Figura 3. Instalación del primer paquete de librerías

## 5. Instalar segundo paquete

- Sudo apt-get install tcl8.5-dev tk8.5-dev

```

xavier@xavier-VirtualBox:~$ sudo apt-get install tcl8.5-dev tk8.5-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
fontconfig-config libexpat1 libexpat1-dev libfontconfig1 libfontconfig1-dev
libfreetype6-dev libpng12-dev libtcl8.5 libtk8.5 libxft-dev libxrender-dev
libxss-dev tcl8.5 tk8.5 x11proto-render-dev x11proto-scrnsaver-dev
zlib1g-dev
Paquetes sugeridos:
tcl-tclreadline tcl8.5-doc tk8.5-doc
Se instalarán los siguientes paquetes NUEVOS:
libexpat1-dev libfontconfig1-dev libfreetype6-dev libpng12-dev libtcl8.5
libtk8.5 libxft-dev libxrender-dev libxss-dev tcl8.5 tcl8.5-dev tk8.5
tk8.5-dev x11proto-render-dev x11proto-scrnsaver-dev zlib1g-dev
Se actualizarán los siguientes paquetes:
fontconfig-config libexpat1 libfontconfig1
3 actualizados, 16 nuevos se instalarán, 0 para eliminar y 437 no actualizados.
Se necesita descargar 5.408 kB de archivos.
Se utilizarán 20,5 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [5/n] s
Des:1 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main i386 libexpat1 i386
LibreOffice Impress 0.04.2 [74,2 kB]
Des:2 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main i386 libfontconfig
1 i386 2.11.04-0ubuntu1.1 [140 kB]
4% [2 libfontconfig1 131 kB/140 kB 94%]

```

Figura 4. Instalación del Segundo paquete de librerías

## 6. Instalación del programa Xgraph para visualizar las gráficas de la transmisión entre nodos, utilizar el siguiente comando:

- Sudo apt-get install perlxgraphlibxt-dev libx11-dev libxmu-dev



```
xavier@xavier-VirtualBox:~$ sudo apt-get install perl xgraph libxt-dev libx11-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
libx11-dev ya está en su versión más reciente (2:1.6.3-1ubuntu2).
Fijado libx11-dev como instalado manualmente.
libxmu-dev ya está en su versión más reciente (2:1.1.2-2).
libxt-dev ya está en su versión más reciente (1:1.1.5-0ubuntu1).
Fijado libxt-dev como instalado manualmente.
perl ya está en su versión más reciente (5.22.1-9).
Fijado perl como instalado manualmente.
Paquetes sugeridos:
  xgraph
Se instalarán los siguientes paquetes NUEVOS:
  xgraph
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 437 no actualizados.
Se necesita descargar 94,4 kB de archivos.
Se utilizarán 289 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://ec.archive.ubuntu.com/ubuntu xenial/universe i386 xgraph i386 12.1-17 [94,4 kB]
Descargados 94,4 kB en 1s (71,5 kB/s)
Seleccionando el paquete xgraph previamente no seleccionado.
(Leyendo la base de datos ... 218830 ficheros o directorios instalados actualmen
Ubuntu Software desempaquetar .../xgraph_12.1-17_i386.deb ...
Desempaquetando xgraph (12.1-17) ...
```

Figura 5. Instalación del programa Xgraph

7. Dirigirse a la carpeta donde fue copiado el programa extraído con nombre ns-allinone-2.35 e ingresar a la carpeta ns-2.35.



Figura 6. Ubicación del archivo extraído ns-allinone-2.35





```

ls.N [-/ns-allinone-2.35/ns-2.35/linkstate] - gedit
Abrir [F] Guardar

typedef less<Key> less_key;
typedef map<Key, T, less_key> BaseMap;
LsMap() : BaseMap() {}

// this next typedef of iterator seems extraneous but is required by gcc
typedef typename map<Key, T, less_key> ::iterator iterator;
typedef pair<iterator, bool> pair_iterator_bool;
iterator insert(const Key & key, const T & item) {
    typename BaseMap::value_type v(key, item);
    pair_iterator_bool lb = BaseMap::insert(v);
    return lb.second ? lb.first : BaseMap::end();
}

void eraseAll() { this->erase(BaseMap::begin(), BaseMap::end()); }
T* findPtr(Key key) {
    iterator it = BaseMap::find(key);
    return (it == BaseMap::end()) ? (T *)NULL : &(*it).second;
}
};

/*
LsNodeidList -- A list of int 's. It manages its own memory
*/
class LsNodeidList : public LsList<int> {
public:
    int appendUnique (const LsNodeidList& x);
};

/*-----*/
/*

```

Figura 9. Modificación de la línea 137

10. Para instalar el programa ns-2 debemos navegar en una terminal a la ruta donde fue copiado el software unas veces que fue extraído.

- /home/Xavier/ns-allinone-2.35

11. Instalar el software con el comando:

- ./install

```

xavier@xavier-VirtualBox: ~/ns-allinone-2.35
xavier@xavier-VirtualBox:~$ ls
Descargas  ejemplos.desktop  ns-allinone-2.35      Público
Documentos  Imágenes          ns-allinone-2.35.tar.gz  Videos
Escritorio  Música            Plantillas
xavier@xavier-VirtualBox:~$ cd ns-allinone-2.35/
xavier@xavier-VirtualBox:~/ns-allinone-2.35$ ./install
=====
* Testing for Darwin (OS X) environment
=====
* Testing for Cygwin environment
=====
Cygwin not detected, proceeding with regular install.
=====
* Testing for FreeBSD environment
=====
FreeBSD not detected
=====
* Build xGraph-12.2
=====
configure: error: expected an absolute directory name for --prefix: ..
make all-am
make[1]: se entra en el directorio '/home/xavier/ns-allinone-2.35/xgraph-12.2'
make[1]: No se hace nada para 'all-am'.
make[1]: se sale del directorio '/home/xavier/ns-allinone-2.35/xgraph-12.2'
xgraph has been installed successfully.
=====
* Build CWeb
=====
Making cweb
gcc -g -c -o ctangle.o ctangle.c
common.h:123:8: warning: type defaults to 'int' in declaration of 'history' [-Wt

```

Figura 10. Instalación del software ns-allinone-2.35

12. como último paso realizar una validación con el comando `./valídate` desde la ruta donde se encuentra ubicado el archivo ns-allinone-2.35

Nota: La validación de este parche puede tardar más o menos 40 minutos dependiendo de las características del CPU.

13. Comprobamos la ejecución del programa del simulador de redes escribiendo el comando `(ns)` en una nueva terminal y si el resultado fue el signo `%`, la instalación fue exitosa.

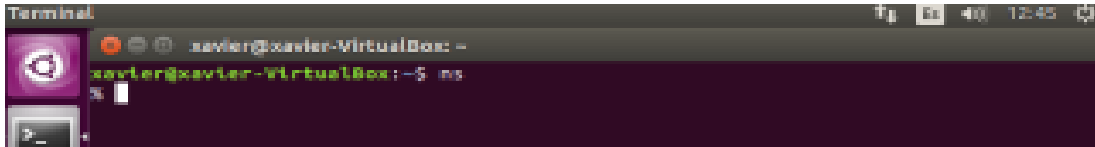


Figura 11. Comprobación del funcionamiento del programa simulador de redes NS-2

14. De igual manera podemos comprobar la interfaz gráfica escribiendo la palabra (nam)

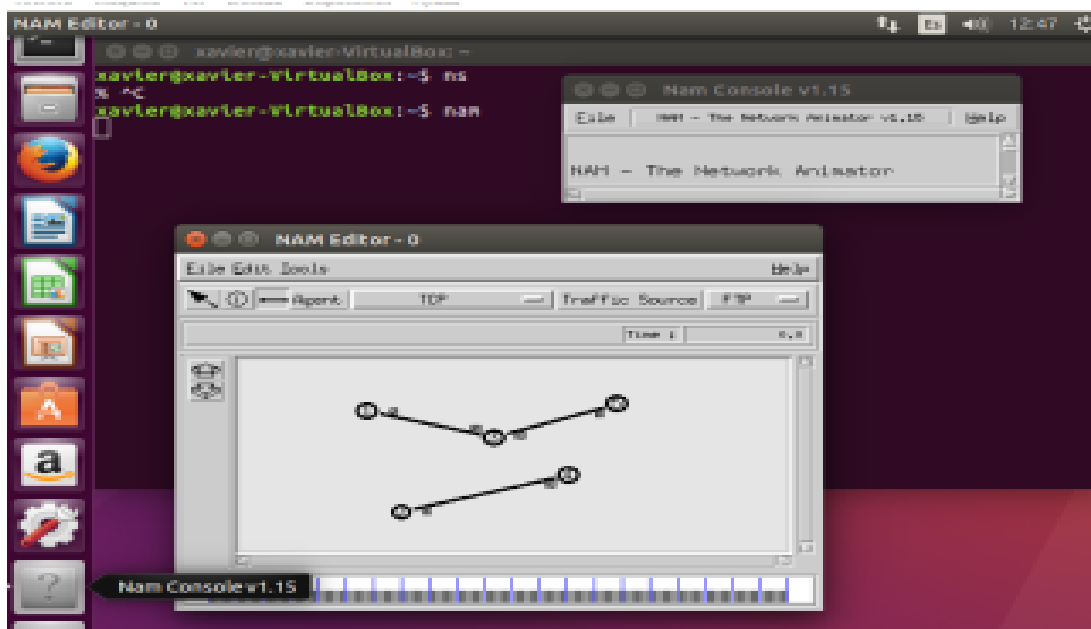


Figura 12. Comprobación de funcionamiento de la aplicación gráfica

- Una guía de instalación adecuada lo puede encontrar en el siguiente link: <https://www.youtube.com/watch?v=5oYJRLB8P1k>

## Anexo 14

### INSTALACIÓN DEL PROGRAMA TRACEGRAPH

#### Procedimiento

El programa TraceGraph puede trabajar tanto en Windows como en Linux, se debe tomar en cuenta que al momento de su instalación puede existir complicaciones por el tipo de librerías con algunas versiones de Linux.

1.- Realizar la descarga del software tracegraph de la página oficial, este paquete contiene las librerías de matlab (mglinstaller).

- [www.tracegraph.com/download.html](http://www.tracegraph.com/download.html)

2.- Descomprimir los archivos en la carpeta (home/usuario) de Linux.

3.- Abrir una terminal para instalar los programas que fueron descomprimidos en la carpeta antes mencionada con el comando:

- ./mglinstaller.
- Una guía de instalación adecuada lo puede encontrar en el siguiente link:

<http://read.pudn.com/downloads155/doc/685571/TRACEGRAPH.pdf>

## Anexo 15

### INSTALACIÓN DEL PROGRAMA NSG2

#### Procedimiento

El software de simulación NSG2 es un generador de los escenarios para NS2 basados en Java, este software genera archivos TCI de tipo inalámbricas y cableadas para ser ejecutadas en NS2, con este sistema se puede crear:

- Nodos inalámbricos y cableados
- Vínculos dúplex y semidúplex
- Agentes Tcp y Udp
- Aplicaciones Ftp y Cbr
- Movimientos entre los nodos

#### Procedimiento

1.- Realizar la descarga del software NSG2 desde el siguiente enlace:

- <https://sites.google.com/site/pengjungwu/nsq>

2.- Descomprimir el archivo descargado y copiar en la ruta Home/usuario del sistema operativo Linux.



Figura 13. Descarga de Software y extracción del Archivo .zip

3.- Abrir una terminal y nos ubicamos en la carpeta donde copiamos el archivo comprimido y ejecutamos el comando:

- Java -jar NSG2.1.jar

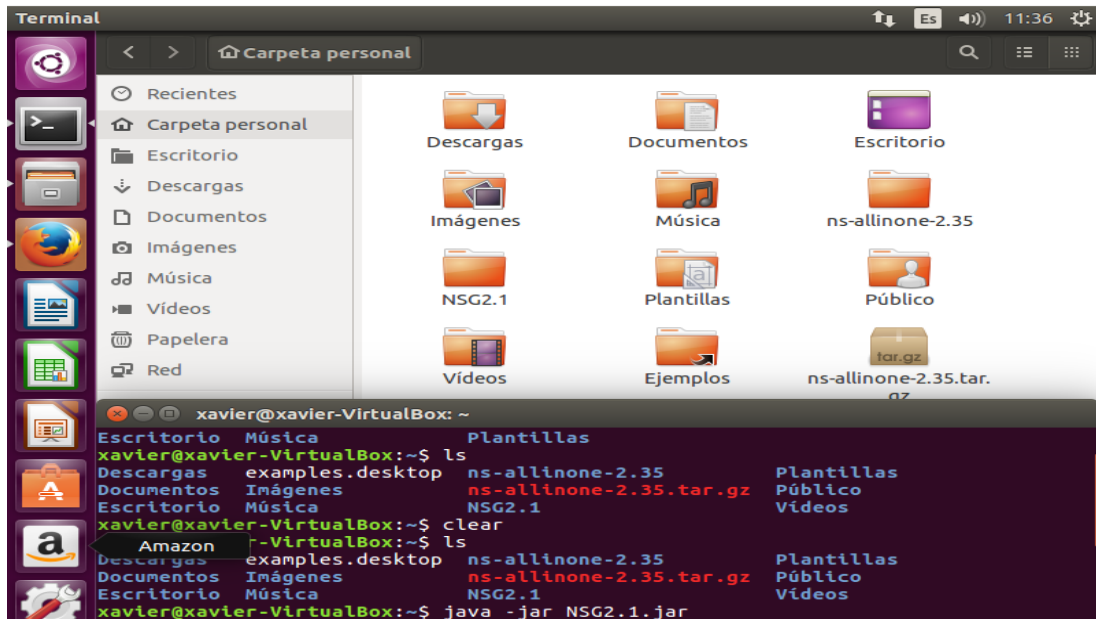


Figura 14. Ejecución del comando Java -jar NSG2.1.jar

- Una guía de instalación adecuada lo puede encontrar en el siguiente link:

<https://www.youtube.com/watch?v=pf8hbcFX1A0>

