

**ANÁLISIS DE TRÁFICO EN TRAYECTORIAS CERRADAS PARA
REDES AD HOC**



**HÉCTOR MEDARDO GÓMEZ RODRÍGUEZ
BRAYAN ALEXANDER ROBAYO GARCÍA
JUAN CAMILO RODRÍGUEZ GÓMEZ**

Los Libertadores
Fundación Universitaria

**FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA ELECTRÓNICA
BOGOTÁ, D. C.
2015**

**ANÁLISIS DE TRÁFICO EN REDES AD HOC PARA TRAYECTORIAS
CERRADAS**

**HÉCTOR MEDARDO GÓMEZ RODRÍGUEZ
BRAYAN ALEXANDER ROBAYO GARCÍA
JUAN CAMILO RODRÍGUEZ GÓMEZ**



Trabajo de grado para optar El Título de Ingeniero Electrónico

Los Libertadores
Fundación Universitaria

**Asesor:
John Petearson Anzola Anzola
Ingeniero Electrónico**

**FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA ELECTRÓNICA
BOGOTÁ, D. C.
2015**

CONTENIDO

	Pág.
1 RESUMEN.....	11
2 INTRODUCCIÓN.....	12
3 OBJETIVOS	14
3.1 OBJETIVO GENERAL.....	14
3.2 OBJETIVOS ESPECÍFICOS.....	14
4 JUSTIFICACIÓN	15
5 ANTECEDENTES.....	16
5.1 Performance Evaluation of AODV and AOMDV with Probabilistic Relay in VANET Environments.....	16
5.2 Power Control and Performance Comparison of AODV and DSR Ad Hoc Routing Protocols.....	18
5.3 Comparison of AODV and DSR on-demand routing protocols in mobile ad hoc networks.....	19
5.4 Comparative performance analysis of GZRP, AOMDV and DSR in MANETs....	21
5.5 Analysis of AOMDV, AODV, DSR, and DSDV Routing Protocols for Wireless Sensor Network.....	23
5.6 Provisioning of Quality of Service in MANETs performance analysis & comparison (AODV and DSR).....	27
5.7 Performance Analysis of AODV, AOMDV, DSR, DSDV Routing Protocols in Vehicular Ad Hoc Network.	30
5.8 Simulation and Comparison of AOMDV, AODV and DSR in MANET.....	33
5.9 Performance Analysis of AODV, DSR, and Swarm Intelligence Routing Protocols In Vehicular Ad hoc Network Environment.	36
5.10 Ad-hoc WSN in Biological Research.	40
5.11 Wireless image-sensor network application for population monitoring of lepidopterous insects pest (moths) in fruit crops.....	40
5.12 Energy Efficiency in the Mobile Ad Hoc Networking Approach to Monitoring Farm Animals.....	41
5.13 Practical MANETs for Pervasive Cattle Monitoring.....	42
5.14 Mobile Ad Hoc Networking Approach to Detecting and Querying Events Related to Farm Animals.....	44
6 MARCO TEÓRICO.....	47
6.1 MÉTRICAS PARA EL ANÁLISIS DE TRÁFICO EN REDES AD HOC.....	47

6.1.1	Retardo de extremo a extremo (DELAY)	47
6.1.2	Jitter.	47
6.1.3	Packet Delivery Ratio (PDR).....	47
6.1.4	Throughput o rendimiento.....	48
6.2	PROTOCOLOS DE ENRUTAMIENTO	48
6.2.1	Ad-Hoc On-Demand Distance Vector (AODV).....	48
6.2.2	Destination-Sequenced Distance-Vector Routing (DSDV).....	49
6.2.3	Ad-Hoc On-Demand Multi Path Distance Vector Routing (AOMDV).....	50
6.2.4	Dynamic Source Routing (DSR).....	51
6.3	CARACTERÍSTICAS PRINCIPALES DE UN CULTIVO DE PAPA	51
6.3.1	Plagas y enfermedades de la papa.....	51
6.3.2	Plagas.	52
6.3.3	La papa en el mundo.....	52
6.3.4	La papa en Colombia.....	54
6.4	CARACTERÍSTICAS PRINCIPALES DE UN CULTIVO DE FRIJOL.	54
6.4.1	Época de Siembra.....	54
7	PLANTEAMIENTO DEL PROBLEMA.....	54
8	METODOLOGÍA.....	55
9	ALCANCES Y LIMITACIONES.....	57
10	ANÁLISIS DE RESULTADOS.....	58
10.1	RESULTADOS DE SIMULACIÓN PARA EL MOVIMIENTO EN GIRASOL.....	58
10.2	RESULTADOS DE SIMULACIÓN PARA EL MOVIMIENTO EN ESTRELLA .	60
10.3	RESULTADOS DE SIMULACIÓN PARA EL MOVIMIENTO EN OCHO.....	62
	CONCLUSIONES.....	64
	BIBLIOGRAFÍA.....	65
	ANEXOS	68
	ANEXO A. CÓDIGO FUENTE PARA GENERACIÓN DE SIMULACIÓN EN NS2.....	68
	ANEXO B. CÓDIGO FUENTE PARA DESCRIPCIÓN DEL TRÁFICO USADO	70
	ANEXO C. CÓDIGOS DE FILTROS AWK PARA SELECCIÓN DE DATOS	73
	FILTRO AWK PARA DELAY.....	73
	FILTRO AWK PARA JITTER	74
	FILTROS AWK PARA PDR	75
	FILTRO AWK PARA THROUGHPUT	76
	ANEXO D. CÓDIGOS PARA GENERACIÓN DE MOVIMIENTOS DE NODOS EN JAVA.....	78

CÓDIGO PRINCIPAL PARA GENERACIÓN DE ESCENARIOS.....	78
CÓDIGO PARA GENERACIÓN DE MOVIMIENTO DE ESTRELLA.....	78
CÓDIGO PARA GENERACIÓN DE MOVIMIENTO DE GIRASOL	82
CÓDIGO PARA GENERACIÓN DE MOVIMIENTO DE OCHO	86



Los Libertadores
Fundación Universitaria

LISTA DE FIGURAS

	Pág.
Figura 1. Resultados obtenidos Delay	17
Figura 2. Resultados obtenidos PDR	17
Figura 3. Datos obtenidos de la simulación.	19
Figura 4. Throughput obtenido.....	20
Figura 5. Delay obtenido.....	20
Figura 6. Relación de entrega de paquetes sobre la movilidad.	22
Figura 7. Delay sobre movilidad.....	22
Figura 8. Control de sobrecarga sobre movilidad.	23
Figura 9. Comparación de PDR con 50 nodos usando TCP y CBR.	24
Figura 10. Comparación de Delay con 50 nodos usando TCP y CBR.....	25
Figura 11. Comparación de LPR con 50 nodos usando TCP y CBR.	25
Figura 12. Comparación de PDR con 100 nodos usando TCP y CBR.	25
Figura 13. Delay con 100 nodos usando TCP.	26
Figura 14. LPR con 100 nodos usando TCP.	26
Figura 15. Comparación de los tipos de entrega de paquetes con TR=250M.	28
Figura 16. Comparaciones de promedio de Delay con TR=250m.	29
Figura 17. Comparaciones promedio de carga de enrutamiento con TR=250m....	29
Figura 19. Delay extremo a extremo Vs velocidad del vehículo, densidad del vehículo.	31
Figura 20. PDR Vs velocidad del vehículo, densidad del vehículo.	32
Figura 21. Pausetime Vs end to end delay.	33
Figura 22. Pausetime Vs routing overhead.....	34
Figura 23. Route discovery Vs pausetime.	34
Figura 24. Packet loss Vs pausetime.....	34
Figura 25. Routing overhead Vs pausetime.....	35
Figura 26. PDR Vs pausetime.....	35
Figura 27. Pausetime Vs throughput.....	36
Figura 28. Troughput Vs velocidad de datos para paquetes de 64 bits.	38
Figura 29. Latencia Vs velocidad de datos para paquetes de 64 bits.	38
Figura 30. Troughput Vs velocidad de datos para paquetes de 512 bits.	38
Figura 31. Latencia Vs velocidad de datos para paquetes de 512 bits.	39
Figura 32. Temperatura censada.....	40
Figura 33. Diseño de la trampa.....	41
Figura 34. Arquitectura del sistema propuesto.....	41
Figura 35. Resultados de simulación.	42
Figura 36. Resultados de simulación.	43
Figura 37. Número de nodos de la red en SR.	45
Figura 38. Trayectoria con forma de estrella.	55

Figura 39. Trayectoria con forma de girasol.	55
Figura 40. Trayectoria con forma de ocho.	56
Figura 41. DELAY Vs Velocidad por Paquete (NARANJA Protocolo AODV, AZUL Protocolo DSK).	58
Figura 42. Figura 42. JITTER Vs Velocidad por Paquete (NARANJA Protocolo AODV, AZUL Protocolo DSK).	58
Figura 43. PDR Vs Velocidad por Paquete (NARANJA Protocolo AODV, AZUL Protocolo DSK).	59
Figura 44. THROUGHPUT Vs Velocidad por Paquete (NARANJA Protocolo AODV, AZUL Protocolo DSK).	59
Figura 45. Retardo extremo a extremo movimiento estrella.	60
Figura 46. Jitter movimiento de estrella.	60
Figura 47. PDR movimiento en estrella.	61
Figura 48. Throughput movimiento estrella.	61
Figura 49. Retardo extremo a extremo movimiento en ocho.	62
Figura 50. Jitter movimiento en ocho.	62
Figura 51. PDR movimiento en ocho.	63
Figura 52. Throughput movimiento en ocho.	63

LISTA DE TABLAS

	Pág.
Tabla 1. Características de simulación paper 1.....	16
Tabla 2. Características de simulación paper 2.....	18
Tabla 3. Características de simulación paper 3.....	19
Tabla 4. Características de simulación paper 4.....	21
Tabla 5. Parámetros de simulación paper 5.....	24
Tabla 6. Parámetros de simulación paper 6.....	27
Tabla 7. Parámetros de simulación paper 7.....	30
Tabla 8. Parámetros de simulación paper 8.....	33
Tabla 9. Parámetros de Simulación paper 9.....	37
Tabla 10. Principales plagas de la papa.....	52
Tabla 11. Parámetros de simulación principal.....	56



Los Libertadores
Fundación Universitaria

GLOSARIO

AD-HOC: es una red específica cuya infraestructura solo tiene sentido en ese instante o situación, es decir su topología es variante en el tiempo.

COSECHA: en agricultura la cosecha se basa en la recolección de los frutos, semillas u hortalizas de los campos en la época del año en que están maduros. La cosecha marca el final del crecimiento de una estación o el final del ciclo de un fruto en particular. En uso general incluye también las acciones posteriores a la recolección del fruto propiamente dicha, tales como la limpieza, clasificación y embalado de lo recolectado hasta su almacenamiento en la granja o su envío al mercado de venta al por mayor.

ENCAMINAMIENTO (o enrutamiento, ruteo): es la función de buscar un camino entre todos los posibles en una red de paquetes cuyas topologías poseen una gran conectividad. Dado que se trata de encontrar la mejor ruta posible, lo primero será definir qué se entiende por mejor ruta y en consecuencia cuál es la métrica que se debe utilizar para medirla.

PROTOCOLO: En informática y telecomunicación, un protocolo de comunicaciones es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitir información por medio de cualquier tipo de variación de una magnitud física. Se trata de las reglas o el estándar que define la sintaxis, semántica y sincronización de la comunicación, así como también los posibles métodos de recuperación de errores. Los protocolos pueden ser implementados por hardware, por software, o por una combinación de ambos.

FLOOD O FLOODING: es un término en inglés que significa literalmente inundación. Se usa en la jerga informática para designar una corrida abusiva de la red de comunicaciones, normalmente por la repetición desmesurada de algún movimiento en un corto espacio de tiempo. Un *flood* consiste en expulsar mucha información en poco tiempo a alguien para intentar que se quede satisfecho. La manera de aprovechar esto en los chats, consiste en enviar muchas peticiones de información a la víctima, de forma que ésta, al contestar, supere el límite del servidor y éste lo saque.

FLUCTUACIÓN: se refiere a la cantidad de variación de retardo que introduce la Red. Una Red con fluctuación cero tarda exactamente el mismo tiempo en transferir cada paquete, mientras que una Red con fluctuación alta tarda mucho más en entregar algunos paquetes que otros.

MANET (*Mobile Ad-hoc Networks*): es un tipo de Red Ad-Hoc que pueden cambiar de ubicación y configurarse sobre la marcha. Debido a que las redes

MANETs son móviles y utilizan conexiones inalámbricas para conectarse a diferentes redes. Esto puede ser una conexión Wi-Fi estándar, u otro medio, como una transmisión celular o satelital.

MALEZA: conjunto de árboles, arbustos y otras plantas que crecen muy juntos entrecruzando y enredando sus ramas de manera que dan lugar a una gran espesura.

QoS o Calidad de Servicio: es el rendimiento promedio de una red de telefonía o de computadoras, particularmente el rendimiento visto por los usuarios de la red. Cuantitativamente mide la calidad de los servicios que son considerados varios aspectos del servicio de red, tales como tasas de errores, ancho de banda, rendimiento, retraso en la transmisión, disponibilidad, etc.

TRANSMISSION CONTROL PROTOCOL (TCP) o Protocolo de Control de Transmisión: es uno de los protocolos fundamentales en Internet. Muchos programas dentro de una red de datos compuesta por redes de computadoras, pueden usar TCP para crear “conexiones” entre sí a través de las cuales puede enviarse un flujo de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto.

VANET: habitualmente referida por su acrónimo en inglés VANET, es un tipo de red de comunicación que utiliza a los vehículos como nodos de la red. Dado el reducido alcance del canal de comunicación (hasta 1 km.), la conectividad se establece de forma esporádica (ad-hoc). Por este motivo, estas redes se consideran un tipo específico de red móvil de comunicación (MANET).

WLAN (*Wireless Local Area Network*): un tipo de red de área local que utiliza ondas de radio de alta frecuencia en lugar de cables para la comunicación entre nodos.

1 RESUMEN

Una de las características de las redes Ad Hoc está dada en la movilidad de sus nodos, lo cual hace más difícil el enrutamiento en este tipo de redes, por la dinámica de su conectividad y topología. En este trabajo se hace un análisis del comportamiento del tráfico para modelos de movilidad que describen trayectorias cerradas, evaluando el rendimiento de los protocolos AOMDV, AODV, DSDV y DSR. El tráfico de datos se puede analizar en un escenario ideal para cultivos de papa y frijol, donde 20 nodos realizan un reconocimiento del área del cultivo desplazándose en trayectorias cerradas que forman figuras como un ocho, un girasol y una estrella. En cada uno de estos desplazamientos se determina y analiza el rendimiento global de la red, mediante las métricas del Delay, Jitter, Throughput, y el PDR.



Los Libertadores
Fundación Universitaria

2 INTRODUCCIÓN

El programa de Ingeniería Electrónica de la Fundación Universitaria Los Libertadores a través del Grupo de Investigación Aplicada en Señales y Sistemas (GUIAS) fomenta y apoya las actividades de los semilleros de investigación, como estrategia que promueve la cultura investigativa en sus estudiantes.

El presente trabajo de grado recopila parte de las actividades y directrices del grupo de investigación, que promueve la divulgación de los trabajos desarrollados al interior del semillero en Señales y Sistemas en temáticas relacionadas con las Redes Auto organizadas.

Los resultados registrados en este trabajo de grado se postularon a los siguientes eventos académicos:

"IV CONGRESO INTERNACIONAL DE INSTRUMENTACIÓN CONTROL Y TELECOMUNICACIONES - CIICT 2015 y el I CONGRESO INTERNACIONAL EN DISEÑO, FABRICACIÓN Y NUEVOS MATERIALES - CIDIFAM 2015" que se realizarán en la ciudad de Tunja, del 21 al 23 de octubre de 2015, bajo la organización de la Universidad Santo Tomás seccional Tunja, con el artículo postulado "*Traffic Analysis in Ad Hoc Networks for Applications in Bean Fields*".

El VII Congreso Iberoamericano de Telemática 2015 que se llevo a cabo en la Ciudad de Popayán en el mes de Junio, participando con el paper titulado "*Traffic Analysis AODV and AOMDV for data transmission in SLAM*", el cual, se encuentra registrado en las respectivas memorias del evento.

El trabajo de grado desarrollado relaciona las redes Ad Hoc, como redes inalámbricas auto organizadas y descentralizadas¹⁻². Entendiendo como Red Ad Hoc, a las redes que no dependen de una infraestructura, como routers (en redes cableadas) o de puntos de accesos en redes inalámbricas. Para ello, cada nodo participa en el encaminamiento mediante el reenvío de datos hacia otros nodos, de modo que la determinación de estos nodos hacia la información se hace dinámicamente sobre la base de conectividad de la red. En adición al encaminamiento clásico, las redes Ad hoc pueden usar *flooding* (inundación de red) para el reenvío de datos³. El trabajo que se presenta a continuación, trata sobre el análisis de tráfico en redes Ad Hoc con un enfoque aplicativo hacia la agricultura, explícitamente los campos de papa y frijol. Se propone analizar las métricas del delay, jitter, throughput y la pérdida de paquetes en redes de sensores distribuidos aleatoriamente en un cultivo, con el fin de transmitir los datos de las condiciones del suelo tales como humedad, temperatura del ambiente que rodea el cultivo, entre otros. Este proceso es llevado a cabo mediante modelos de simulación en NS2, cuyos resultados al llegar a ser implementados, pueden

brindar un mejor tratamiento al suelo en que se cultiva, para así ofrecer mejor calidad y modernidad en los procesos que se conllevan en la agricultura.

Paralelamente en un isomorfismo realizado en este trabajo se trasladó el análisis de tráfico en movimientos cerrados a un espacio plano ideal para aplicaciones de SLAM (*Simultaneous Localization And Mapping*), cuyos resultados derivados de este documento se pudieron publicar en las memorias del evento “VII Congreso Iberoamericano de Telemática 2015”, como se menciona anteriormente.



Los Libertadores
Fundación Universitaria

3 OBJETIVOS

3.1 OBJETIVO GENERAL

Comparar el rendimiento de los protocolos AOMDV, AODV, DSDV y DSR utilizados en redes Ad-Hoc con modelos de movimiento de nodos con trayectorias cerradas.

3.2 OBJETIVOS ESPECÍFICOS

- Analizar los protocolos AOMDV, AODV, DSDV y DSR en un escenario plano de 200 x 200 metros para determinar el impacto de QoS en redes Ad Hoc para trayectorias cerradas.
- Estimar las métricas del Delay, Jitter, PDR y Throughput en un escenario con trayectorias cerradas variando la velocidad de los nodos en redes Ad- Hoc.



4 JUSTIFICACIÓN

En los últimos años la investigación de redes inalámbricas Ad Hoc se ha basado en el análisis de los diferentes protocolos de enrutamiento incluyendo escenarios en los cuales los nodos se mueven libremente. Se han analizado las características de cada protocolo para compararlos entre sí y de esa forma determinar cuál es el más óptimo. En ese contexto, se han propuesto simulaciones generando movimientos para cada nodo de forma aleatoria dentro de un espacio de trabajo definido.

La propuesta científica de este trabajo, consiste en la generación de movimientos cerrados generando Figuras específicas como estrellas, girasoles, ochos, con el fin de abarcar mayor espacio y terreno, como forma moderna y novedosa de analizar las propiedades y características de terrenos destinados a la parte agrícola, siembra de vegetales y elaboración de cultivos.

Actualmente no existen métodos efectivos y eficientes para analizar las características de terreno y suelos, dichos tópicos son analizados a través de un muestreo manual, en el cuál se obtienen datos y se hace una estimación para un posterior análisis químico, a este proceso se le llama análisis de suelo. Dicho análisis es una práctica usual y es aprobado como informativo y como parte primordial de cualquier programa de manejo adecuado.

Este trabajo analiza el impacto del tráfico de datos para movimientos de nodos o dispositivos en escenarios planos, siguiendo trayectorias controladas y cerradas que posteriormente se puede implementar en terrenos destinados a la agricultura y cultivos de vegetales, más precisamente en sembrados de papa y frijol.

Por lo tanto este trabajo contribuye al avance y la adquisición de datos más precisos y exactos favoreciendo el desarrollo de la agricultura y la calidad de los cultivos en el país, aportando conocimiento para lograr la excelencia en los productos nacionales.

5 ANTECEDENTES

En este apartado se destacan el estado del arte en redes Ad Hoc para aplicaciones en agricultura con los siguientes tres enfoques:

- Enfoque de evaluación de métricas para estimar rendimiento.
- Enfoque de comparación de protocolos y escenarios.
- Aplicaciones en el campo Agrícola.

Para el enfoque de evaluación de métricas para estimar rendimiento se destacan los siguientes artículos:

5.1 Performance Evaluation of AODV and AOMDV with Probabilistic Relay in VANET Environments⁴.

En este artículo se realiza una comparación entre dos diferentes protocolos de enrutamiento utilizado en redes VANET (Vehicular Ad-Hoc Network) con el estándar de transmisión 802.11. Para este tipo de redes el mayor problema es la velocidad con la que los vehículos se mueven, ya que se pueden perder las conexiones entre nodos si estos se apartan muy rápido.

Para realizar la comparación de los protocolos de enrutamiento bajo demanda se toma el mapa de una ciudad para que sea el escenario en el cual viajaran los vehículos. Adicionalmente se aplicara un Relé de probabilidades para determinar las posibles conexiones entre los vehículos o nodos, la velocidad con que estos se mueven serán constantes. Los parámetros de simulación se describen en la Tabla 1.

Tabla 1. Características de simulación paper 1.

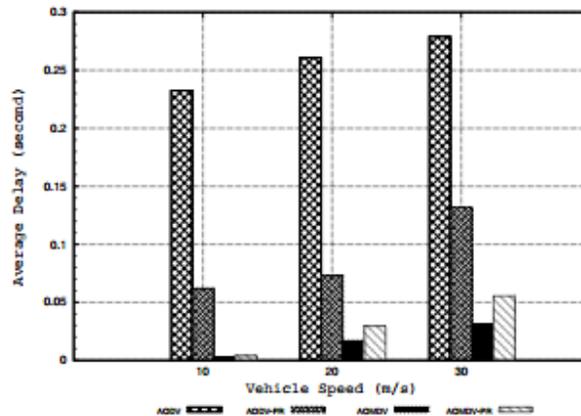
Network Simulator	Ns2.34
Simulation Time	500 seconds
Simulation Area	1000 x 1000 meter
Number of Vehicles	100 vehicles
Total Data Packets Sent	200 packets
Data type	Constant bit rate (CBR)
CBR Interval	1 Second
Data Packet Size	512 bytes
Number of Connections	1 UDP connection
Propagation Model	Nakagamo
MAC protocol	AODV, AODMV
Radio Range	250 meters
Maximun Vehicle Speed	10, 20, 10 m/s (36, 72, 108 km/h)

Fuente: R. Anggoro, T. Kitasuka, R. Nakamura, and M. Aritsugi, "Performance Evaluation of AODV and AOMDV with Probabilistic Relay in VANET Environments," in Networking and Computing (ICNC), 2012 Third International Conference on, 2012, pp. 259–263.

Los resultados obtenidos se muestran en las Figuras 1 y 2. Cabe resaltar que los protocolos a los que se les aplico en Relé probabilístico de denotan como AODV-PR PR (*Ad Hoc On-demand Distance Vector with Probabilistic Relay*) y AOMDV (*Ad Hoc On-demand Multipath Distance Vector with Probabilistic Relay*) y las

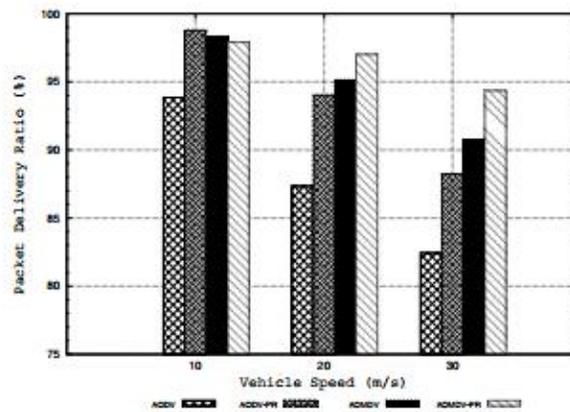
métricas utilizados son el Delay (tiempo que tarda un paquete desde el envío hasta la recepción) y el PDR (diferencia entre el número de paquetes enviados y los recibidos). Para redes Ad Hoc.

Figura 1. Resultados obtenidos Delay



Fuente: R. Anggoro, T. Kitasuka, R. Nakamura, and M. Aritsugi, "Performance Evaluation of AODV and AOMDV with Probabilistic Relay in VANET Environments," in Networking and Computing (ICNC), 2012 Third International Conference on, 2012, pp. 259–263.

Figura 2. Resultados obtenidos PDR



Fuente: R. Anggoro, T. Kitasuka, R. Nakamura, and M. Aritsugi, "Performance Evaluation of AODV and AOMDV with Probabilistic Relay in VANET Environments," in Networking and Computing (ICNC), 2012 Third International Conference on, 2012, pp. 259–263.

Según los resultados obtenidos se muestra una mejora en los protocolos que tienen el relé probabilístico ya que estos tienen una mejor probabilidad de saber su próxima conexión y no se genera una pérdida de paquetes y se disminuye el Delay.

5.2 Power Control and Performance Comparison of AODV and DSR Ad Hoc Routing Protocols⁵.

En este artículo se realiza la comparación entre los protocolos de enrutamiento AODV y DSR (*Dynamic Source Routing*) para redes Ad Hoc esta comparación se realiza teniendo en cuenta la relación a ruido o ruido en el ambiente de acuerdo a la potencia proporcionada a cada protocolo.

Para realizar la comparación de estos protocolos se utilizan métricas ya conocidas para estos tipos de redes, estas métricas son el Delay el Throughput (velocidad con la que es enviada la información). Como se habla de la potencia de cada señal también se utiliza la carga de la red. Esta comparación se genera en un espacio libre donde los nodos tienen un movimiento aleatorio pero no restringido. El modelo de la simulación se describe en la Tabla 2.

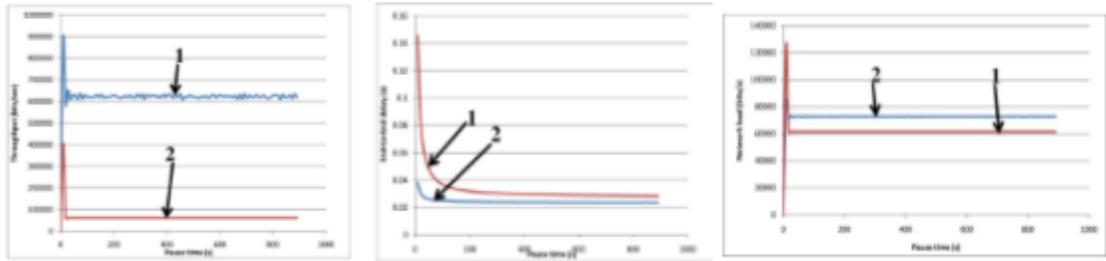
Tabla 2. Características de simulación paper 2.

Channel Type	Wireless Channel
Physical Characteristics	Direct Sequence
Mac Type	802.11b
Data Type	11Mbps
Topologia	700m X 500m
Routing Protocols	AODV, DSR
Number of Nodes	40
Transmit Power	0.005 W
Packet Size	1000 bytes
Mobility Model	Randon Way Point
Simulation Time	900 sec
Traffic Source	CBR
Speed	0-20 m/sec

Fuente: L. Bello, P. Bakalis, S. J. Manam, T. I. Eneh, and K. A. Anang, "Power Control and Performance Comparison of AODV and DSR Ad Hoc Routing Protocols," in *Computer Modelling and Simulation (UKSim)*, 2011 *UkSim 13th International Conference on*, 2011, pp. 457–460

Los resultados obtenidos de se muestran en la Figura 3. Donde el protocolo AODV se ve representado con color azul y señalado con el número 2 y el protocolo DSR se representa con el color rojo y se señala con el número 1. La potencia fue constante y la misma para los dos protocolos.

Figura 3. Datos obtenidos de la simulación.



Fuente: L. Bello, P. Bakalis, S. J. Manam, T. I. Eneh, and K. A. Anang, "Power Control and Performance Comparison of AODV and DSR Ad Hoc Routing Protocols," in Computer Modelling and Simulation (UKSim), 2011 UkSim 13th International Conference on, 2011, pp. 457–460

Los resultados obtenidos muestran que el protocolo AODV tiene un mejor rendimiento en cuanto a lo potencia requerida y el retardo lo cual hace que este protocolo sea el apropiado para cualquier tipo de Red Ad Hoc.

5.3 Comparison of AODV and DSR on-demand routing protocols in mobile ad hoc networks⁶.

En este artículo se realiza la comparación de dos protocolos AODV y DSR, en ambos casos se utilizara el estándar 802.11.

Para realizar la comparación de estos protocolos se utilizan métricas mencionadas anteriormente como el Delay y el Throughput. Esta comparación se genera en un espacio libre, la velocidad de trasmisión ira aumentado de 10 a 80 m/sec, el número de nodos en la primera parte será de 20 y el segunda de 80. Las características de la simulación desarrollada se muestran en la Tabla 3.

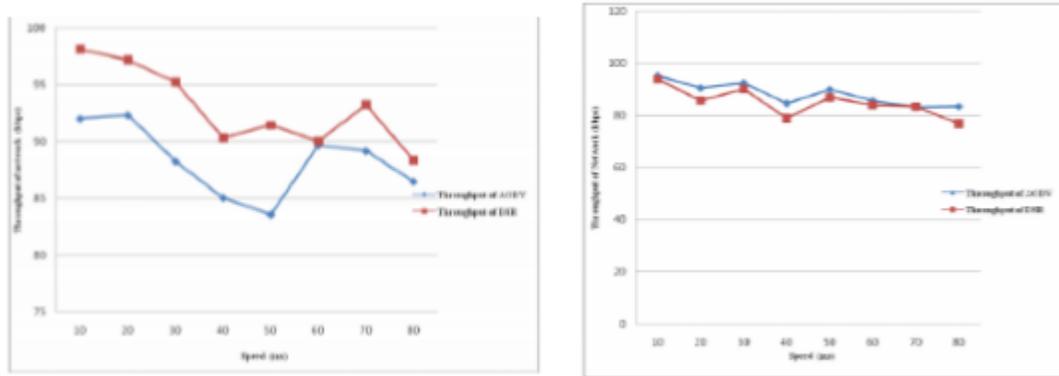
Tabla 3. Características de simulación paper 3.

Simulator	NS-2.35
Channel Type	Channel/Wireless channel
Antenna Type	Antenna/ Omni Antenna
Radio-propagation model	Propagation/ Two Ray Ground
Link layer type	LL
Mac type	Mac/ 802_11
Protocols studied	AODV and DSR
Simulation time	100 sec
Pause time	10 sec
Simulation area	800*800
Transmission range	250m
Trace format	New Wireless format
Node movement model	Random waypoint
Traffic type	CBR(UDP)
Packet size	512 Bytes
Number of nodes	20,80
speed	10 to 80 m/sec

Fuente: A. M. Kanthe, D. Simony, and R. Prasad, "Comparison of AODV and DSR on-demand routing protocols in mobile ad hoc networks," in Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN), 2012 1st International Conference on, 2012, pp. 1–5.

Los resultados obtenidos en la prueba del Throughput se muestran en la Figura 4.

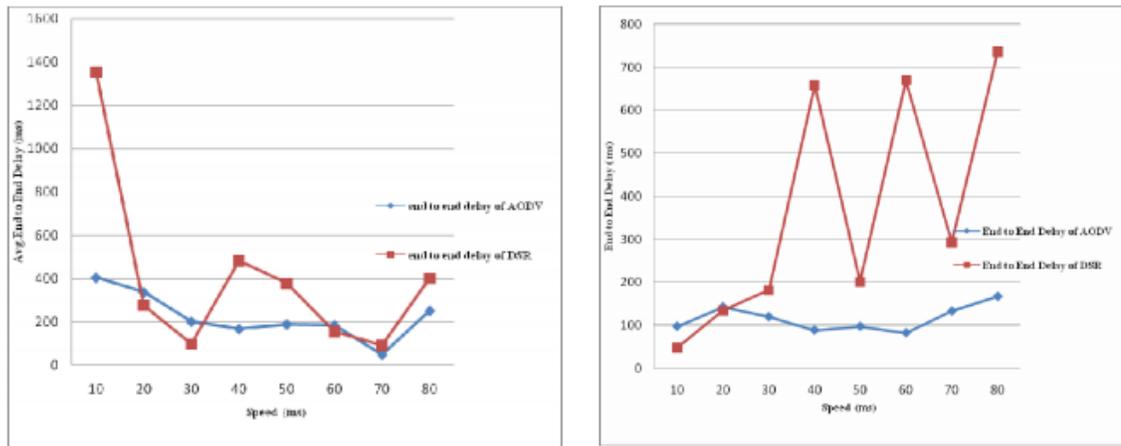
Figura 4. Throughput obtenido.



Fuente: A. M. Kanthe, D. Simony, and R. Prasad, "Comparison of AODV and DSR on-demand routing protocols in mobile ad hoc networks," in Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN), 2012 1st International Conference on, 2012, pp. 1–5.

Los resultados obtenidos en la prueba del Delayse muestran en la Figura 5.

Figura 5. Delay obtenido.



Fuente: A. M. Kanthe, D. Simony, and R. Prasad, "Comparison of AODV and DSR on-demand routing protocols in mobile ad hoc networks," in Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN), 2012 1st International Conference on, 2012, pp. 1–5.

De acuerdo con los resultados obtenidos se determinó que el protocolo AODV tiene un mejor rendimiento frente al DSR pero también se observa que el protocolo AODV tiene un mayor rendimiento al tener un número mayor de nodos.

5.4 Comparative performance analysis of GZRP, AOMDV and DSR in MANETs⁷.

En este trabajo se analizan los rendimientos de los protocolos de enrutamiento AOMDV, GZRP (*Genetic Zone Routing Protocol*) y DSR respecto a la relación de entrega de paquetes, el Delay de extremo a extremo y la sobrecarga de control. Los parámetros utilizados para la simulación para evaluar los protocolos se resumen en la Tabla 4.

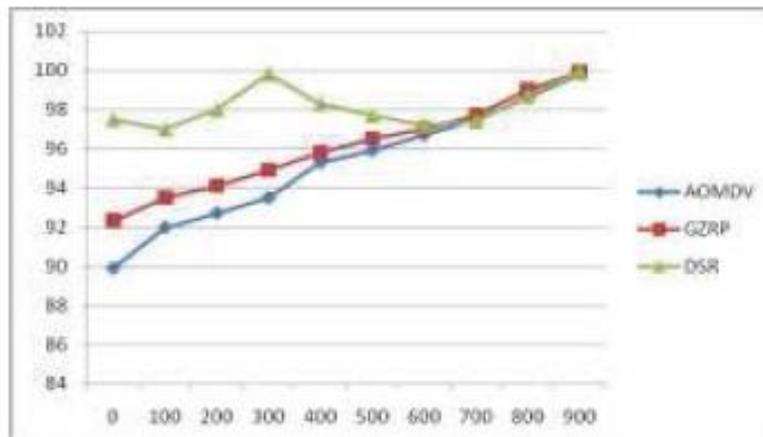
Tabla 4. Características de simulación paper 4.

Parameter	Value
Simulation Time	900 s
Number of experimented trials	6
Network Coverage Area	1500 x 300 m ²
Number of Nodes	50
Mobility Model	Model: Random Way Point Pause Time: 0s, 100s, ..., 900s Minimum Speed: 1 m/s Maximum Speed: 20 m/s Node Placement: Random
MAC	802.11
Bandwidth	2 Mbps
Transmission range	200m
Simulation area	800*800
Matrices	PDR, End to end delay, Control Overhead

Fuente: P. Sarkar and H. Paul, "Comparative performance analysis of GZRP, AOMDV and DSR in MANETs," in Computational Intelligence and Information Technology, 2013. CIIT 2013. Third International Conference on, 2013, pp. 175–178.

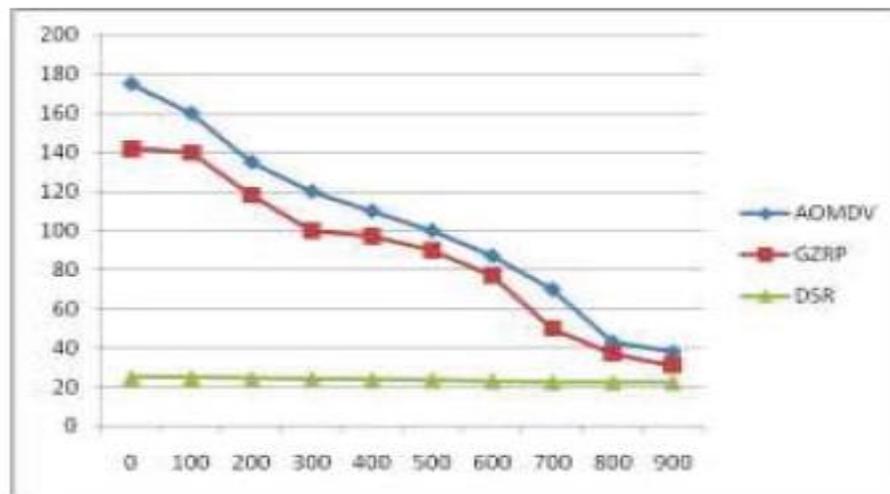
El resultado experimental de la Figura 6 para la relación de la entrega de paquetes infiere que DSR muestra una mejor entrega de paquetes que GZRP y AOMDV. Pero es mejor GZRP que AODV porque GZRP tiene mejor capacidad de equilibrio de carga. La Figura 7 muestra el efecto de retardo medio en la movilidad de los nodos. Indica que DSR muestra el menor retardo medio de movilidad. También muestra que el retraso promedio de GZRP es menor que AOMDV. La Figura 8 muestra el efecto de sobrecarga de control sobre la movilidad de los nodos. Esto demuestra que la sobrecarga de control de GZRP es mínima.

Figura 6. Relación de entrega de paquetes sobre la movilidad.



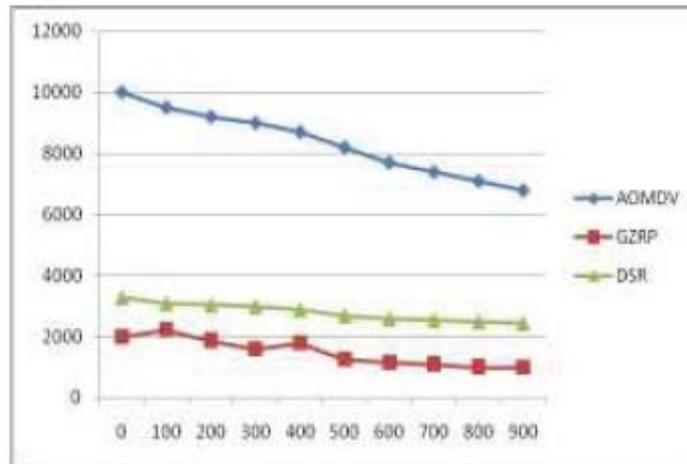
Fuente: P. Sarkar and H. Paul, "Comparative performance analysis of GZRP, AOMDV and DSR in MANETs," in Computational Intelligence and Information Technology, 2013. CIIT 2013. Third International Conference on, 2013, pp. 175–178.

Figura 7. Delay sobre movilidad.



Fuente: P. Sarkar and H. Paul, "Comparative performance analysis of GZRP, AOMDV and DSR in MANETs," in Computational Intelligence and Information Technology, 2013. CIIT 2013. Third International Conference on, 2013, pp. 175–178.

Figura 8. Control de sobrecarga sobre movilidad.



Fuente: P. Sarkar and H. Paul, "Comparative performance analysis of GZRP, AOMDV and DSR in MANETs," in Computational Intelligence and Information Technology, 2013. CIIT 2013. Third International Conference on, 2013, pp. 175–178.

Los resultados experimentales muestran que GZRP utiliza mejor las rutas de fácil acceso para el equilibrio de carga y tolerancia a fallos de la red.

5.5 Analysis of AOMDV, AODV, DSR, and DSDV Routing Protocols for Wireless Sensor Network⁸.

En este artículo los protocolos AODV, AOMDV, DSR y DSDV (*Destination-Sequenced Distance-Vector Routing*) se han analizado mediante la comparación de las diferentes métricas de rendimiento, tales como la relación de entrega de paquetes (PDR), la relación de pérdida de paquetes (LPR), y el retardo promedio, variando el tiempo de pausa y el número de nodo bajo TCP (Transmission Control Protocol) y conexión CBR.

Para la simulación de la red, hay muchos parámetros de rendimiento que están disponibles. Estos se utilizan para evaluar el rendimiento de la red. Algunos ejemplos son Relación de Entrega de Paquetes (PDR), Delay normal de extremo a extremo, la relación de pérdida de paquetes (LPR). En este trabajo se han utilizado tres métricas de rendimiento con fines de simulación de la red.

En la Tabla 5 se observan los parámetros de simulación.

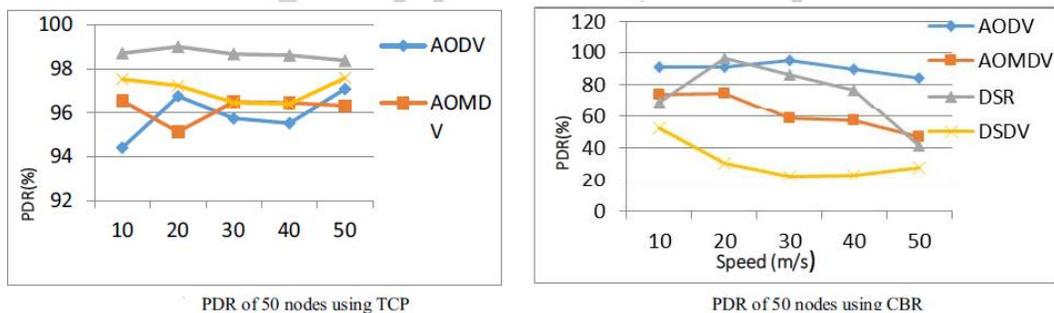
Tabla 5. Parámetros de simulación paper 5.

Parameter	Value
Protocols	AODV, AOMDV, DSR, DSDV
Simulation Area	1200*1200
Packet Size	512
Simulation Time	200s
Pause Time	20s
Speed Time	10, 20, 30, 40, 50
Seed Time	1.0
Number of Nodes	50, 100, 150
Traffic Generation	TCP, CBR
Mobility Model	Random Waypoint
Network Simulator	NS 2.35

Fuente: B. Paul, K. A. Bhuiyan, K. Fatema, and P. P. Das, "Analysis of AOMDV, AODV, DSR, and DSDV Routing Protocols for Wireless Sensor Network," in Computational Intelligence and Communication Networks (CICN), 2014 International Conference on, 2014, pp. 364–369

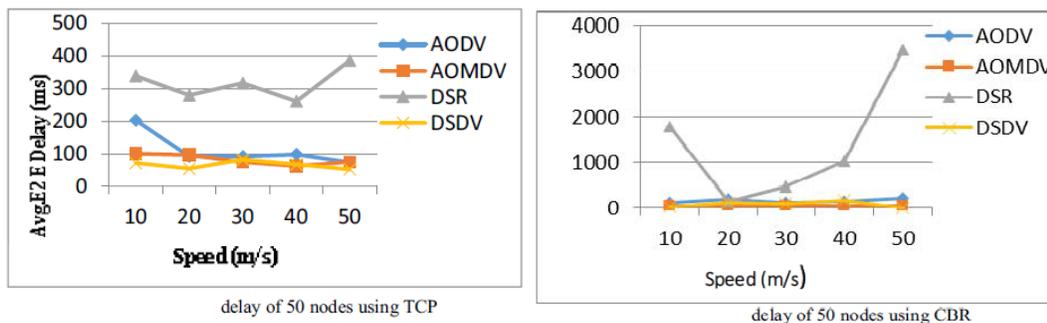
El simulador Network Simulator NS-2.35 fue utilizado para medir el rendimiento. Con el mismo escenario para cada protocolo (AODV, AOMDV, DSR y DSDV), en la Figura 9 se observan los resultados del PDR con 50 nodos usando TCP y usando CBR, en la Figura 10 están plasmado los resultados del Delay con 50 nodos usando TCP y CBR, en la Figura 11 se realiza la comparación del LPR con 50 nodos usando TCP y usando CBR, en la Figura 12 se comparan el PDR con 100 nodos usando TCP y usando CBR, en la Figura 13 se comparan los protocolos analizando el Delay para 100 nodos usando TCP, en la Figura 14 se comparan los protocolos analizando el LPR para 100 nodos usando TCP.

Figura 9. Comparación de PDR con 50 nodos usando TCP y CBR.



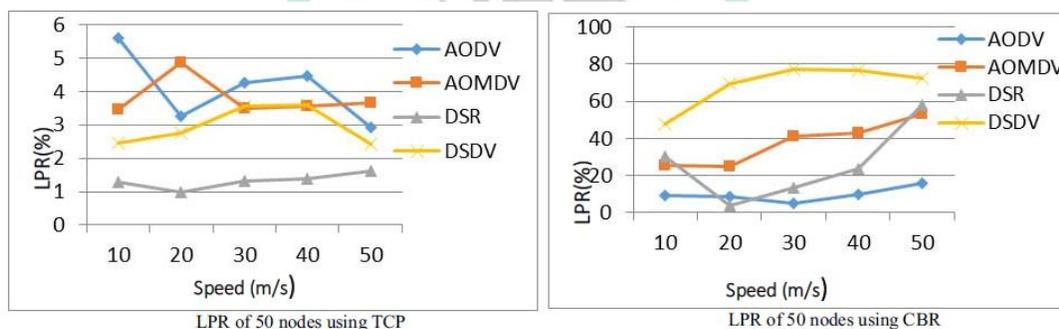
Fuente: B. Paul, K. A. Bhuiyan, K. Fatema, and P. P. Das, "Analysis of AOMDV, AODV, DSR, and DSDV Routing Protocols for Wireless Sensor Network," in Computational Intelligence and Communication Networks (CICN), 2014 International Conference on, 2014, pp. 364–369

Figura 10. Comparación de Delay con 50 nodos usando TCP y CBR.



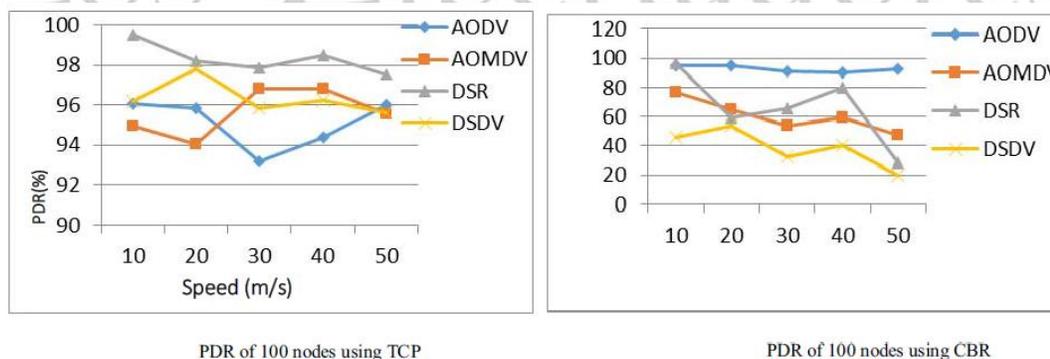
Fuente: B. Paul, K. A. Bhuiyan, K. Fatema, and P. P. Das, "Analysis of AOMDV, AODV, DSR, and DSDV Routing Protocols for Wireless Sensor Network," in Computational Intelligence and Communication Networks (CICN), 2014 International Conference on, 2014, pp. 364–366.

Figura 11. Comparación de LPR con 50 nodos usando TCP y CBR.



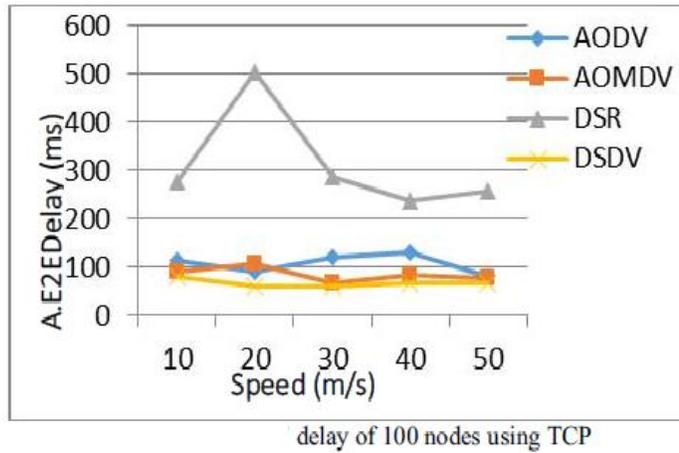
Fuente: B. Paul, K. A. Bhuiyan, K. Fatema, and P. P. Das, "Analysis of AOMDV, AODV, DSR, and DSDV Routing Protocols for Wireless Sensor Network," in Computational Intelligence and Communication Networks (CICN), 2014 International Conference on, 2014, pp. 364–369.

Figura 12. Comparación de PDR con 100 nodos usando TCP y CBR.



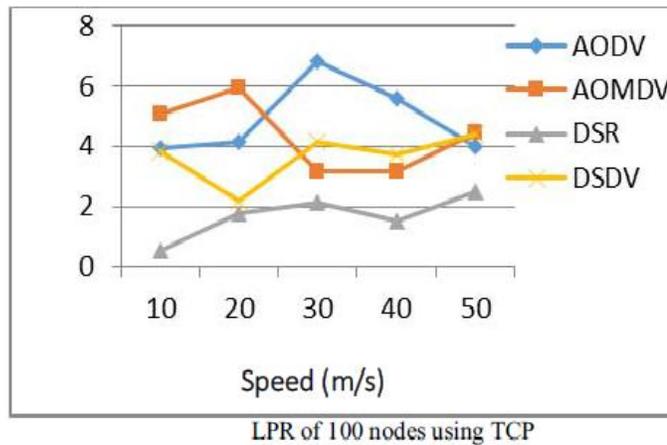
Fuente: B. Paul, K. A. Bhuiyan, K. Fatema, and P. P. Das, "Analysis of AOMDV, AODV, DSR, and DSDV Routing Protocols for Wireless Sensor Network," in Computational Intelligence and Communication Networks (CICN), 2014 International Conference on, 2014, pp. 364–369

Figura 13. Delay con 100 nodos usando TCP.



Fuente: B. Paul, K. A. Bhuiyan, K. Fatema, and P. P. Das, "Analysis of AOMDV, AODV, DSR, and DSDV Routing Protocols for Wireless Sensor Network," in Computational Intelligence and Communication Networks (CICN), 2014 International Conference on, 2014, pp. 364–369

Figura 14. LPR con 100 nodos usando TCP.



Fuente: B. Paul, K. A. Bhuiyan, K. Fatema, and P. P. Das, "Analysis of AOMDV, AODV, DSR, and DSDV Routing Protocols for Wireless Sensor Network," in Computational Intelligence and Communication Networks (CICN), 2014 International Conference on, 2014, pp. 364–369

Para finalmente concluir que el rendimiento de AODV, AOMDV, DSR y DSDV se puede cambiar de acuerdo a diferentes escenarios, y que el rendimiento de DSR es mejor que AODV, AOMDV y DSDV en tipo de conexión TCP. También en tipo de conexión CBR, el rendimiento de AODV es mejor que AOMDV, DSDV y DSR.

5.6 Provisioning of Quality of Service in MANETs performance analysis & comparison (AODV and DSR)⁹.

Este artículo se centra en los dos algoritmos populares AODV y DSR, siendo ambos protocolos de enrutamiento reactivos. Se analizó y comparo su rendimiento a través de la simulación utilizando el simulador NS2.

El modelo de movilidad utiliza el modelo de punto de referencia al azar en un campo rectangular. Las configuraciones de campo utilizadas son: 800 mx 800 m campo con un total de 50 nodos. Aquí, cada paquete inicia su viaje desde una ubicación aleatoria a un destino al azar con una velocidad elegida al azar (uniformemente distribuido entre 1-20 m/s). Una vez alcanzado el destino, otro destino al azar se dirige después de una pausa. El tiempo de pausa, que afecta a las velocidades relativas de los móviles, es variado. Las simulaciones se ejecutan para 100, 200, 300, 400, 500, 600, 700, 800 segundos simulados. Movilidad y tráfico escenarios idénticos se utilizan en los protocolos para recoger resultados justos.

Los parámetros utilizados en la simulación para evaluar los protocolos se resumen en la Tabla 6.

Tabla 6. Parámetros de simulación paper 6.

Parameter	Value
Number of nodes	10 and 45
Simulation Time	100sec
Pause Time	100s
Environment Size	800x800
Transmission Range	250m and 850m
Traffic Size	CBR (Constant Bit Rate)
Packet Size	512 bytes
Packet Rate	5 packets/s
Minium Speed	1 m/s
Maximum Speed	20 m/s
Queue Length	50
Mobility Model	Random Waypoint

Fuente: N. Sharma, S. Rana, and R. M. Sharma, "Provisioning of Quality of Service in MANETs performance analysis #x00026; comparison (AODV and DSR)," in Computer Engineering and Technology (IC CET), 2010 2nd International Conference on, 2010, vol. 7, pp. V7-243-V7-248

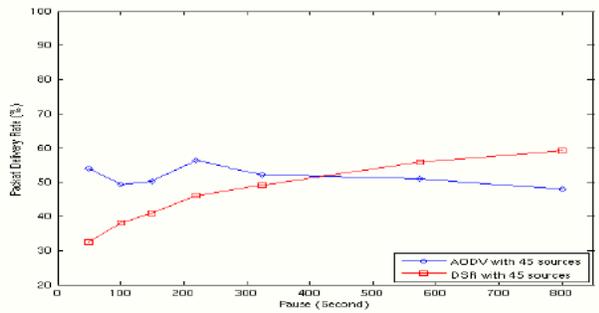
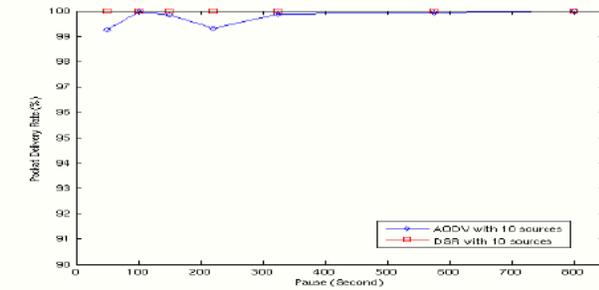
En la Figura 15 se muestran los resultados del experimento de velocidad de administración de paquetes. Se puede ver, con una carga de tráfico bajo como 10 fuentes, DSR superan el AODV, pero cuando las fuentes se hacen más, DSR con la pequeña pausa que significa alta movilidad comienza a hacerse peor que AODV.

Siguiendo la Figura 16 se observa la demora media de AODV y DSR. AODV tiene muchos más paquetes de enrutamiento que DSR, y los paquetes de enrutamiento

consumirán más ancho de banda, AODV entonces tendrá más demora que la DSR.

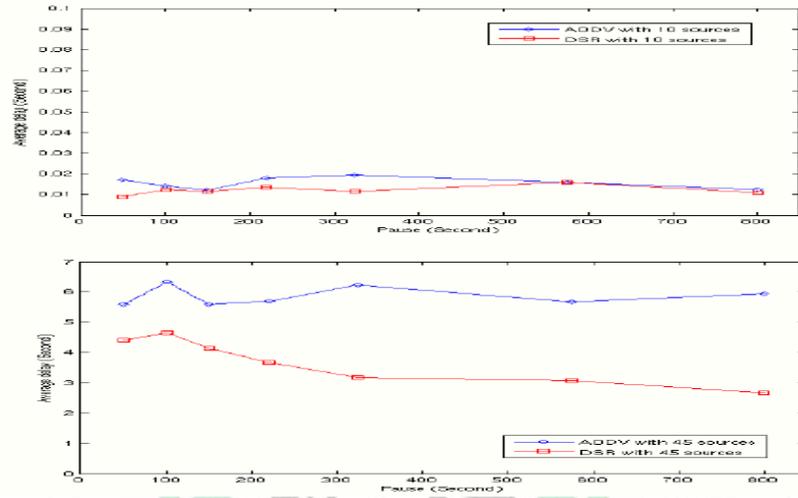
AODV siempre tiene más paquetes de control de enrutamiento que DSR, la sobrecarga de enrutamiento AODV siempre será más alta, incluso en ambiente agotador. Bajo carga pesada, a través de DSR se puede inclinar para elegir la ruta equivocada, sin embargo, en virtud de dicha situación, AODV también generará muchos más paquetes de control que DSR. Este comportamiento se observa en la Figura 17.

Figura 15. Comparación de los tipos de entrega de paquetes con TR=250M.



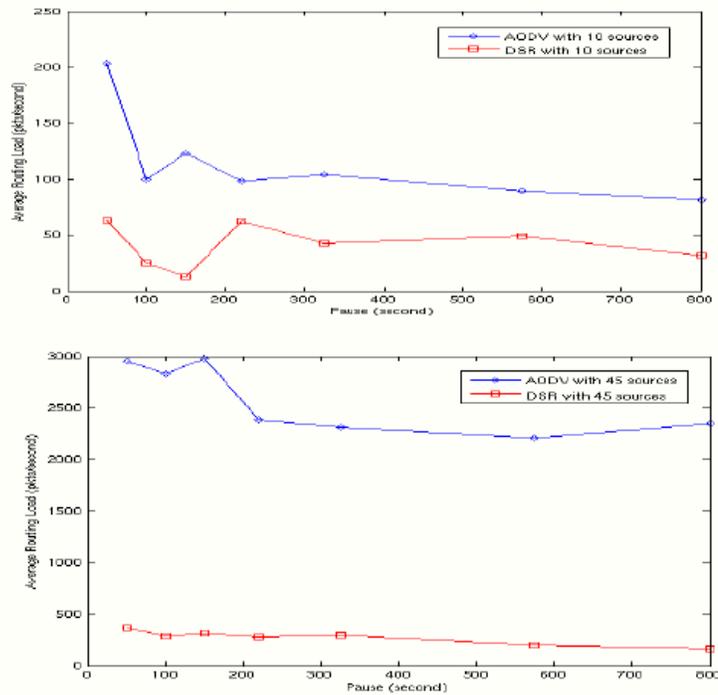
Fuente: N. Sharma, S. Rana, and R. M. Sharma, "Provisioning of Quality of Service in MANETs performance analysis #x00026; comparison (AODV and DSR)," in Computer Engineering and Technology (ICCET), 2010 2nd International Conference on, 2010, vol. 7, pp. V7-243-V7-248

Figura 16. Comparaciones de promedio de Delay con TR=250m.



Fuente: N. Sharma, S. Rana, and R. M. Sharma, "Provisioning of Quality of Service in MANETs performance analysis #x00026; comparison (AODV and DSR)," in Computer Engineering and Technology (ICCET), 2010 2nd International Conference on, 2010, vol. 7, pp. V7-243-V7-248

Figura 17. Comparaciones promedio de carga de enrutamiento con TR=250m.



Fuente: N. Sharma, S. Rana, and R. M. Sharma, "Provisioning of Quality of Service in MANETs performance analysis #x00026; comparison (AODV and DSR)," in Computer Engineering and Technology (ICCET), 2010 2nd International Conference on, 2010, vol. 7, pp. V7-243-V7-248

En donde se concluye que la pérdida de paquetes es muy inferior en caso de AODV inicialmente, pero aumenta sustancialmente cuando el tiempo de simulación aumenta. En el caso de la simulación DSR la pérdida de paquetes es muy alta al principio, pero que disminuye sustancialmente cuando el tiempo de simulación aumenta. Por lo tanto, podemos concluir que si la MANET tiene que ser configurada para una pequeña cantidad de tiempo, AODV se debe elegir debido a la baja pérdida inicial de paquetes y DSR no debería ser tomada en cuenta para configurar una MANET para una pequeña cantidad de tiempo, porque inicialmente la pérdida de paquetes es muy alta.

5.7 Performance Analysis of AODV, AOMDV, DSR, DSDV Routing Protocols in Vehicular Ad Hoc Network¹⁰.

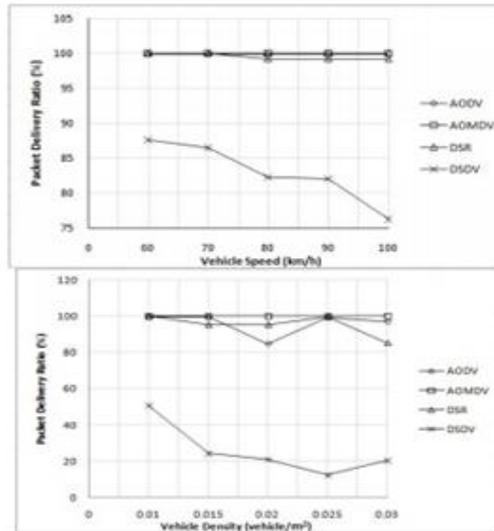
En este paper trata del uso de los protocolos AODV, AOMDV, DSR, DSDV, para determinar el diseño un protocolo de enrutamiento adecuado y eficiente en VANET. La idea es alcanzar la conectividad ubicua para vehículos ya sea a través de vehículo a vehículo o de la comunicación de vehículo a infraestructura que permite el transporte inteligente (ITS). Los protocolos ya mencionados, son explorados para ser comparados en términos de rendimiento de enrutamiento basado en la velocidad del vehículo, la densidad de vehículos, etc.

Tabla 7. Parámetros de simulación paper 7.

SIMULATION PARAMETERS	
PARAMETERS TYPE	VALUE
NETWORK SIMULATOR	NS.2-34
ROUTING PROTOCOL	DSDV, AODV, AOMDV, DSR
SIMULATION TIME	200 s
SIMULATION AREA	10*1000 m
NUMBER OF NODES	100,150,200,250,300
TRAFFIC SOURCE/DESTINATION	DETERMINASTIC
DATA TYPE	CBR
PACKETS GENERATION RATE	5 PACKETS
CBR INTERVAL	0,25 s
PACKET SIZE	100 bytes
MAC PROTOCOL	IEEE802,11p WAVE
MAC RATE	1 Mbps
RTS/CTS	NONE
TRASMISSION RANGE	85 meters
RADIO PROPAGATION MODELS	Two-ray Ground
HELLO DYMO INTERVAL	10000 m
NUMBER OF LANES	3
SPEED OF VEHICLES	60-100 Km/h
SENSING RANGE	85 meters

Fuente: Amirhossein Moravejosharieh, Hero Modares, Rosli Salleh and Ehsan Mostajeran "Performance Analysis of AODV, AOMDV, DSR, DSDV Routing Protocols in Vehicular Ad Hoc Network" Research Journal of Recent Sciences, 2013.

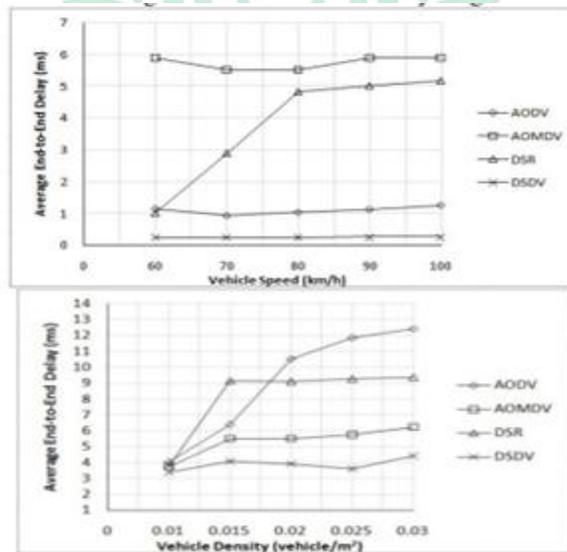
Figura 18. PDR Vs velocidad del vehículo, densidad del vehículo



Packet Loss Ratio Fluctuations Based on Vehicle Velocity and Vehicle Density

Fuente: Amirhossein Moravejosharieh, Hero Modares, Rosli Salleh and Ehsan Mostajeran "Performance Analysis of AODV, AOMDV, DSR, DSDV Routing Protocols in Vehicular Ad Hoc Network" Research Journal of Recent Sciences, 2013.

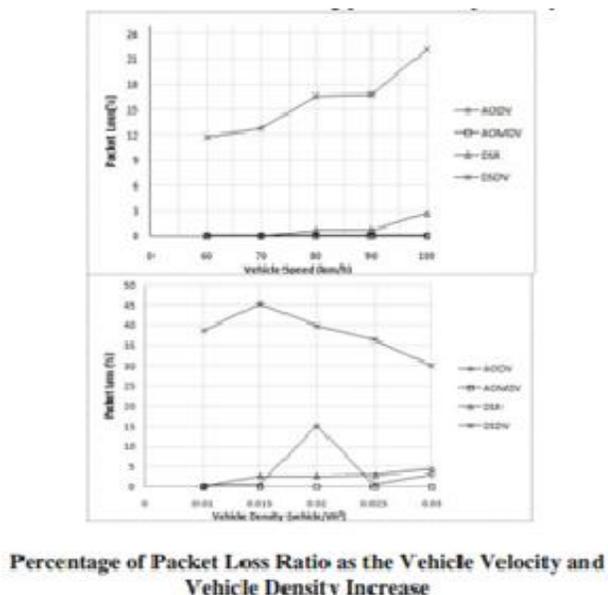
Figura 18. Delay extremo a extremo Vs velocidad del vehículo, densidad del vehículo.



Average End-To-End Diagrams Based on Variety Of Velocity and Density of Vehicles

Fuente: Amirhossein Moravejosharieh, Hero Modares, Rosli Salleh and Ehsan Mostajeran "Performance Analysis of AODV, AOMDV, DSR, DSDV Routing Protocols in Vehicular Ad Hoc Network" Research Journal of Recent Sciences, 2013.

Figura 19. PDR Vs velocidad del vehículo, densidad del vehículo.



Fuente: Amirhossein Moravejosharieh, Hero Modares, Rosli Salleh and Ehsan Mostajeran "Performance Analysis of AODV, AOMDV, DSR, DSDV Routing Protocols in Vehicular Ad Hoc Network" Research Journal of Recent Sciences, 2013.

Figura18: El valor alcanzado especifica la tasa de pérdida de paquetes que se limita el máximo rendimiento de la red. El mejor PDR implica el protocolo de enrutamiento más preciso y adecuado. Considerando el diagrama de la velocidad del vehículo, AODV y AOMDV han mostrado el resultado similar, mientras que el protocolo DSR disminuyó ligeramente a la velocidad de 70. El protocolo de enrutamiento DSDV representa una tendencia descendente significativa de casi el 11%, mientras que la velocidad de los vehículos varía entre 60 y 100 km/h.

Figura19: El tiempo empleado por los paquetes de datos que se entregarán desde el origen al destino que se conoce como un extremo a otro. Por lo tanto, el tiempo en el que el primer paquete de datos es recibido por destino deducirá de la hora a la que el primer paquete transmitido por la fuente. La Figura 6 ilustra el diagrama del retraso promedio que hay de extremo a extremo asociado con los protocolos de enrutamiento mencionados. Aunque el protocolo de enrutamiento DSDV representa el peor, PDR en ambos diagramas de velocidad y densidad, rutas paquetes de datos a un destino con el más bajo promedio de retardo de extremo a extremo en ambos diagramas de densidad y velocidad debido a la intrínseca de los protocolos de enrutamiento proactivas.

Figura20: El número de paquetes perdidos y pérdida de paquetes descartados deliberadamente por los saltos intermedios tales como la sobrecarga de amortiguación o algunos problemas de habilitación de seguridad o debido involuntariamente para evitar la propagación de paquetes dañados. La

Figura representa los diagramas relacionados con tanto evento de pérdida de paquetes y el número de paquetes perdidos asociado con protocolo de enrutamiento mencionado, respectivamente.

5.8 Simulation and Comparison of AOMDV, AODV and DSR in MANET¹¹.

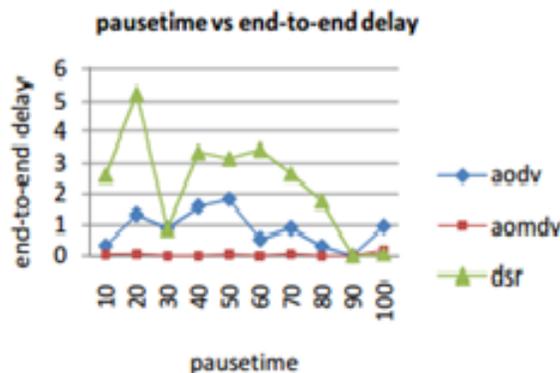
En este paper los tres protocolos de enrutamiento AODV, AODV y DSR se comparan para una red MANET. Los datos que se extraen de este trabajo es el rendimiento de los tres protocolos de enrutamiento en términos de su fracción entrega de paquetes, tiempo de retardo promedio, Routing overhead, Route Discovery Frequency, y Throughput.

Tabla 8. Parámetros de simulación paper 8.

SIMULATION PARAMETERS	
PARAMETERS TYPE	VALUE
NETWORK SIMULATOR	NS.2-34
NUMBER OF NODES	10
SIMULATION TIME	100 s
NODE SPEED	10m/s
DATA RATE	1 Mbps, 10 Mbps
SIMULATION ARES	1000m*1000m
DATA TYPE	CBR
PAUSE TIME	0-100s

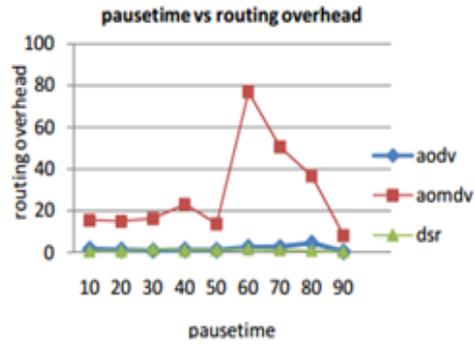
Fuente: P. Jammulaiah and Dr. P. Chenna Reddy, "Simulation and Comparison of AOMDV, AODV and DSR in Manets ", 2013.

Figura 20. Pausetime Vs end to end delay.



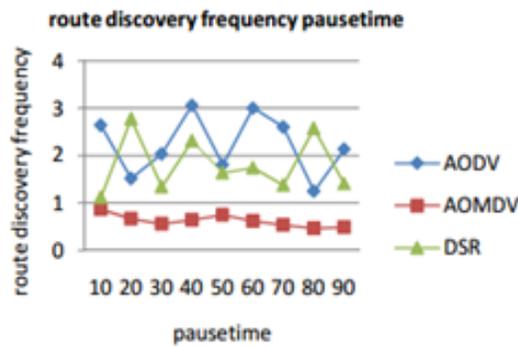
Fuente: P. Jammulaiah and Dr. P. Chenna Reddy, "Simulation and Comparison of AOMDV, AODV and DSR in Manets ", 2013.

Figura 21. Pausetime Vs routing overhead.



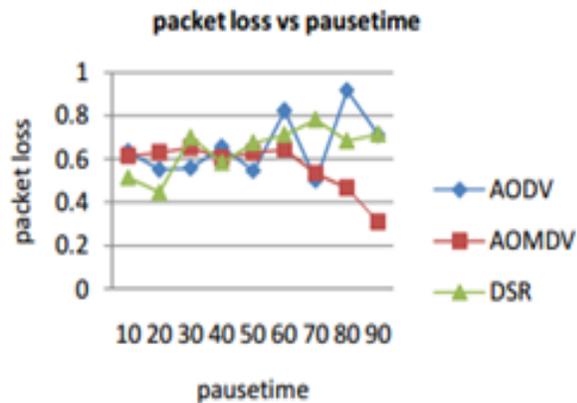
Fuente: P. Jammulaiah and Dr. P. Chenna Reddy, "Simulation and Comparison of AOMDV, AODV and DSR in Manets ", 2013.

Figura 22.Route discovery Vs pausetime.



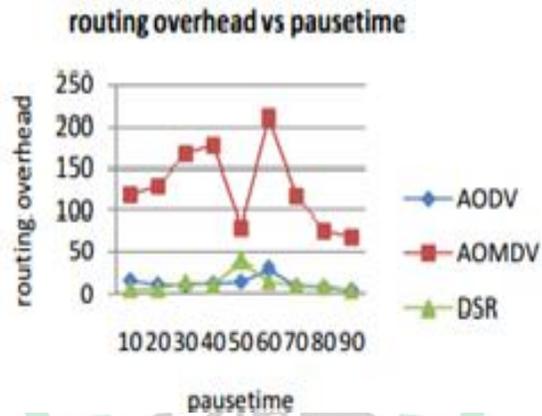
Fuente: P. Jammulaiah and Dr. P. Chenna Reddy, "Simulation and Comparison of AOMDV, AODV and DSR in Manets ", 2013.

Figura 23.Packet loss Vs pausetime.



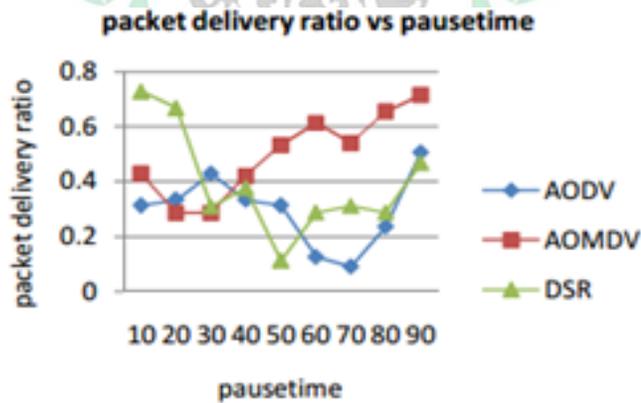
Fuente: P. Jammulaiah and Dr. P. Chenna Reddy, "Simulation and Comparison of AOMDV, AODV and DSR in Manets ", 2013.

Figura 24. Routing overhead Vs pausetime.



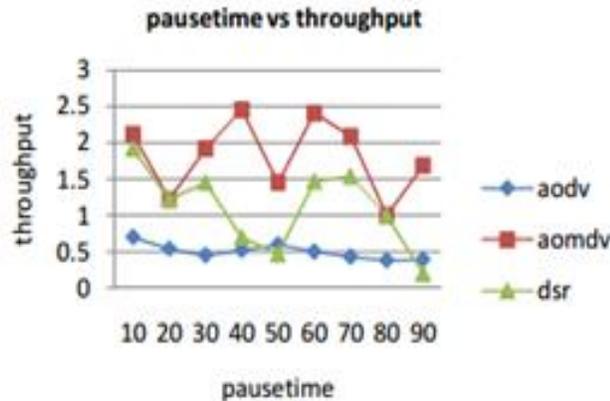
Fuente: P. Jammulaiah and Dr. P. Chenna Reddy, "Simulation and Comparison of AOMDV, AODV and DSR in Manets ", 2013.

Figura 25. PDR Vs pausetime.



Fuente: P. Jammulaiah and Dr. P. Chenna Reddy, "Simulation and Comparison of AOMDV, AODV and DSR in Manets ", 2013.

Figura 26. Pausetime Vs throughput.



Fuente: P. Jammulaiah and Dr. P. Chenna Reddy, "Simulation and Comparison of AOMDV, AODV and DSR in Manets", 2013.

El rendimiento de los tres protocolos de enrutamiento, AOMDV, AODV y DSR se evalúa mediante la comparación de diversos parámetros como la frecuencia de descubrimiento de ruta, la relación de la entrega de paquetes, retardo de extremo a extremo, el gasto de enrutamiento, la pérdida de paquetes, y el rendimiento. De los resultados se puede concluir que la frecuencia de descubrimiento de ruta, el tiempo de retraso entre paquetes de extremo a extremo para AOMDV es muy inferior en comparación con AODV y DSR. Sin embargo, la relación de la entrega de paquetes y rendimiento son relativamente mejor para AOMDV en comparación con AODV y DSR.

5.9 Performance Analysis of AODV, DSR, and Swarm Intelligence Routing Protocols In Vehicular Ad hoc Network Environment¹².

Este paper trata de la comparación y evaluación de los siguientes protocolos de enrutamiento: AODV, DSR, sobre una plataforma VANET. Las VANET se caracterizan por la movilidad, la carga y tamaño de la red. Se evaluaron de la VANET sobre los dos protocolos las siguientes métricas en términos de rendimiento, la latencia, la relación de entrega de datos y los gastos de envío de datos.

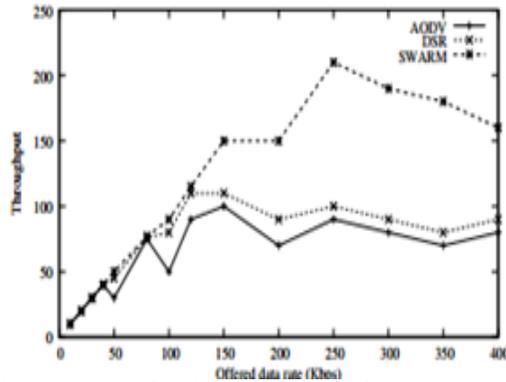
Tabla 9. Parámetros de Simulación paper 9.

SIMULATION PARAMETERS	
PARAMETERS TYPE	VALUE
NETWORK SIMULATOR	NS.2-31
SIMULATION TIME	900 s
SIMULATION AREA	1000m X 1000m
NUMBER OF VEHICLES	30,40,50
TRANSMISSION RANGE	50m, 75m, 100m
SPEED	50 Km/h, 80 Km/h
DATA TYPE	CBR
SOURCE/DESTINATION	RANDOM
DATA PACK SIZE	64 Bytes, 512 Bytes
MAC PROTOCOL	IEEE 802,11 DCF
PROPAGATION MODEL	Two-ray Ground Reflection Model
MOBILITY MODEL	RANDOM WAY POINT
CHANNEL TYPE	WIRELESS CHANNEL
ANTENA MODEL	OMNIDIRECCIONAL
INTERFACE QUEUE TYPE	PRIORITY QUEUE (64 PACKETS MAX)

Fuente: S. S. Manvi, M. S. Kakkasageri, and C. V Mahapurush, "Performance Analysis of AODV, DSR, and Swarm Intelligence Routing Protocols In Vehicular Ad Hoc Network Environment," in Future Computer and Communication, 2009. ICFCC 2009. International Conference on, 2009, pp. 21–25.

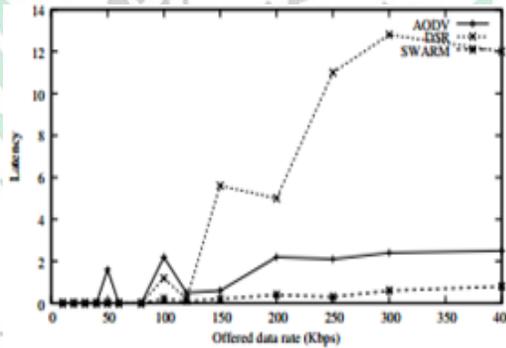
Los Libertadores
Fundación Universitaria

Figura 27. Troughput Vs velocidad de datos para paquetes de 64 bits.



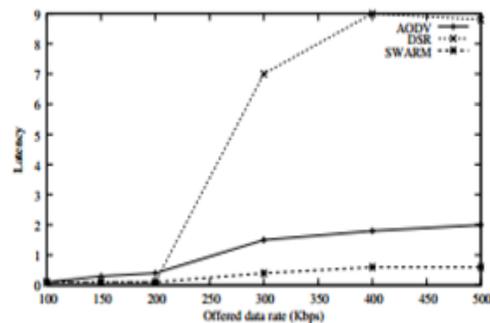
Fuente: S. S. Manvi, M. S. Kakkasageri, and C. V Mahapurush, "Performance Analysis of AODV, DSR, and Swarm Intelligence Routing Protocols In Vehicular Ad Hoc Network Environment," in Future Computer and Communication, 2009. ICFCC 2009. International Conference on, 2009, pp. 21–25.

Figura 28. Latencia Vs velocidad de datos para paquetes de 64 bits.



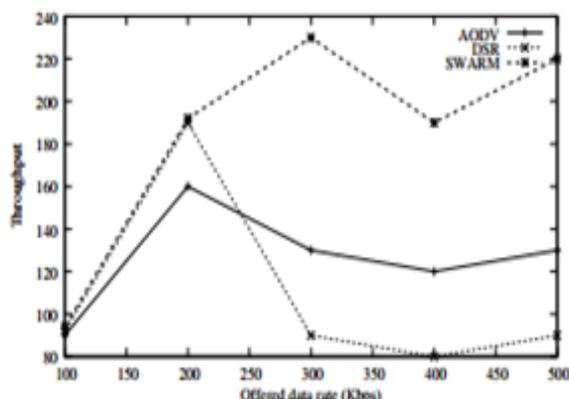
Fuente: S. S. Manvi, M. S. Kakkasageri, and C. V Mahapurush, "Performance Analysis of AODV, DSR, and Swarm Intelligence Routing Protocols In Vehicular Ad Hoc Network Environment," in Future Computer and Communication, 2009. ICFCC 2009. International Conference on, 2009, pp. 21–25.

Figura 29. Troughput Vs velocidad de datos para paquetes de 512 bits.



Fuente: S. S. Manvi, M. S. Kakkasageri, and C. V Mahapurush, "Performance Analysis of AODV, DSR, and Swarm Intelligence Routing Protocols In Vehicular Ad Hoc Network Environment," in Future Computer and Communication, 2009. ICFCC 2009. International Conference on, 2009, pp. 21–25.

Figura 30. Latencia Vs velocidad de datos para paquetes de 512 bits.



Fuente: S. S. Manvi, M. S. Kakkasageri, and C. V Mahapurush, "Performance Analysis of AODV, DSR, and Swarm Intelligence Routing Protocols In Vehicular Ad Hoc Network Environment," in Future Computer and Communication, 2009. ICFCC 2009. International Conference on, 2009, pp. 21–25

El rendimiento por velocidad de datos para diferentes tamaños de paquete se representa en las Figuras 28 y 29. En primer lugar, nos damos cuenta de que el rendimiento de todos los protocolos de enrutamiento aumenta linealmente para baja velocidades de datos. Protocolo de enrutamiento SWARM logra el más alto rendimiento en comparación con el AODV y DSR para altas velocidades de datos.

Algunas de las razones importantes que supera a SWARM que AODV y DSR son los siguientes:

- 1) la ruta de ruta seleccionada garantiza una entrega de datos más grande,
- 2) la tasa de error de paquete en SWARM es mucho más baja que otros,
- 3) SWARM hace uso del mecanismo de selección de ruta adaptativa.

Como se muestra en las Figuras 30 y 31, la latencia del enjambre es la más baja en comparación con AODV, DSR y bajo diferentes tamaños de paquetes. La razón de que SWARM produce una latencia más corta que AODV y DSR es que, la selección de ruta adaptativa en SWARM reenvía paquetes en el camino óptimo. Para AODV y DSR, se requiere más buffering de paquetes, mientras que el reenvío de paquetes; por lo tanto, se aumenta la latencia. A partir de este resultado, podemos concluir que SWARM es aplicable a VANETs con respecto a la latencia, mientras que AODV y DSR son inadecuados.

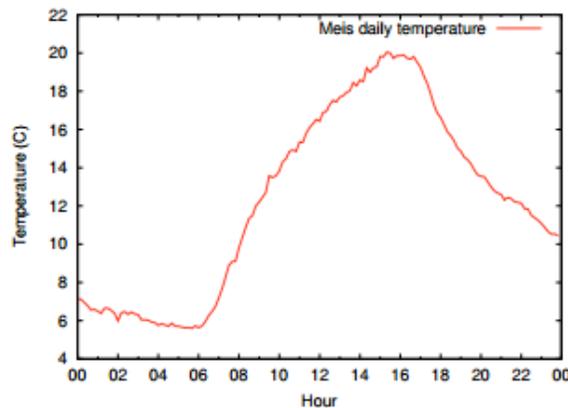
En cuanto a aplicaciones de redes Ad Hoc en el campo agrícola, se han realizado diferentes investigaciones que a continuación se describen:

5.10 Ad-hoc WSN in Biological Research¹³.

En este artículo se realiza una toma de datos por medio de una red ad-hoc para viñedos de España, los datos obtenidos son las diferentes plagas humedad del suelo enfermedades de las plantas. Para realizar la toma de datos se utiliza una red de sensores electrónicos inalámbricos estos sensores se conectan a una estación base la cual contiene un mecanismo el cual utiliza una frecuencia UHF y esta estación se conecta a internet de manera que se pueda realizar una base de datos con la información de los modelos biológicos que se necesitan esta a su vez contiene un radio el cual se comunica con una base de datos generan la cual cada hora se borran los datos para evitar que se llene por completo. Todos estos mecanismos funcionan mediante energía solar para garantizar su funcionamiento continuo.

Uno de los datos más fácil de obtener es la temperatura del terreno ya que se utilizaban sensores de temperatura los resultados se muestran en la Figura 32.

Figura 31. Temperatura censada.

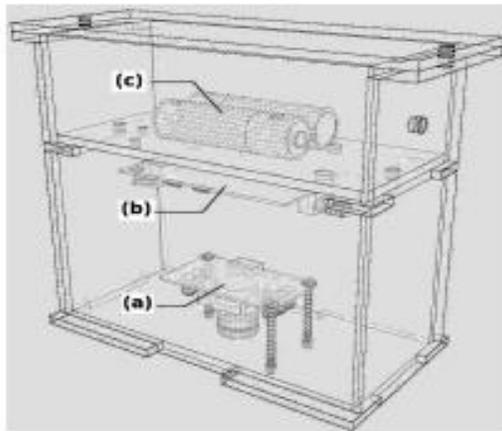


Fuente: P. Marino, F. P. Fontan, M. A. Dominguez, and S. Otero, "Ad-hoc WSN in Biological Research," in Digital System Design, Architectures, Methods and Tools, 2009. DSD '09. 12th Euromicro Conference on, 2009, pp. 841–848

5.11 Wireless image-sensor network application for population monitoring of lepidopterous insects pest (moths) in fruit crops¹⁴.

En este artículo se diseña una trampa para lepidopterous (polillas). Estas trampas son diseñadas para tener control de la plaga y a su vez realizar un procesamiento de las imágenes o de las lepidopterous para realizar a su vez una clasificación según sea la edad o estado de la lepidopterous. La trampa diseñada se muestra en la Figura 33.

Figura 32. Diseño de la trampa.

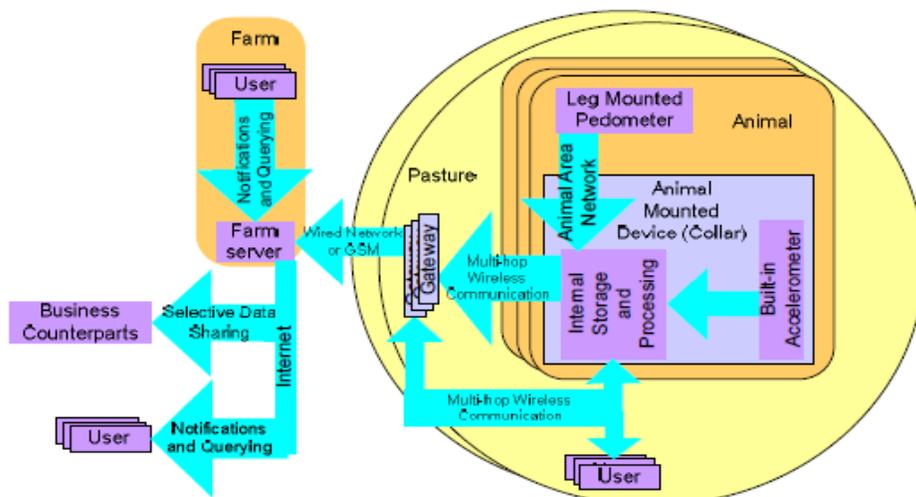


Fuente: M. Gonzalez, J. Schandy, N. Wainstein, L. Barboni, A. Gomez, and C. Croce, "Wireless image-sensor network application for population monitoring of lepidopterous insects pest (moths) in fruit crops," in Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International, 2014, pp. 1394–1398.

En el numeral a de la Figura 33 se muestra la cámara con la cual se captura la imagen, en el numeral b se establece el procesador de imagen y el trasmisor de radio frecuencia y el numeral c muestra las baterías con las cuales se controlara todo el dispositivo. Este dispositivo es solo diseñado pues los recursos económicos y físicos no permitían el montaje.

5.12 Energy Efficiency in the Mobile Ad Hoc Networking Approach to Monitoring Farm Animals¹⁵.

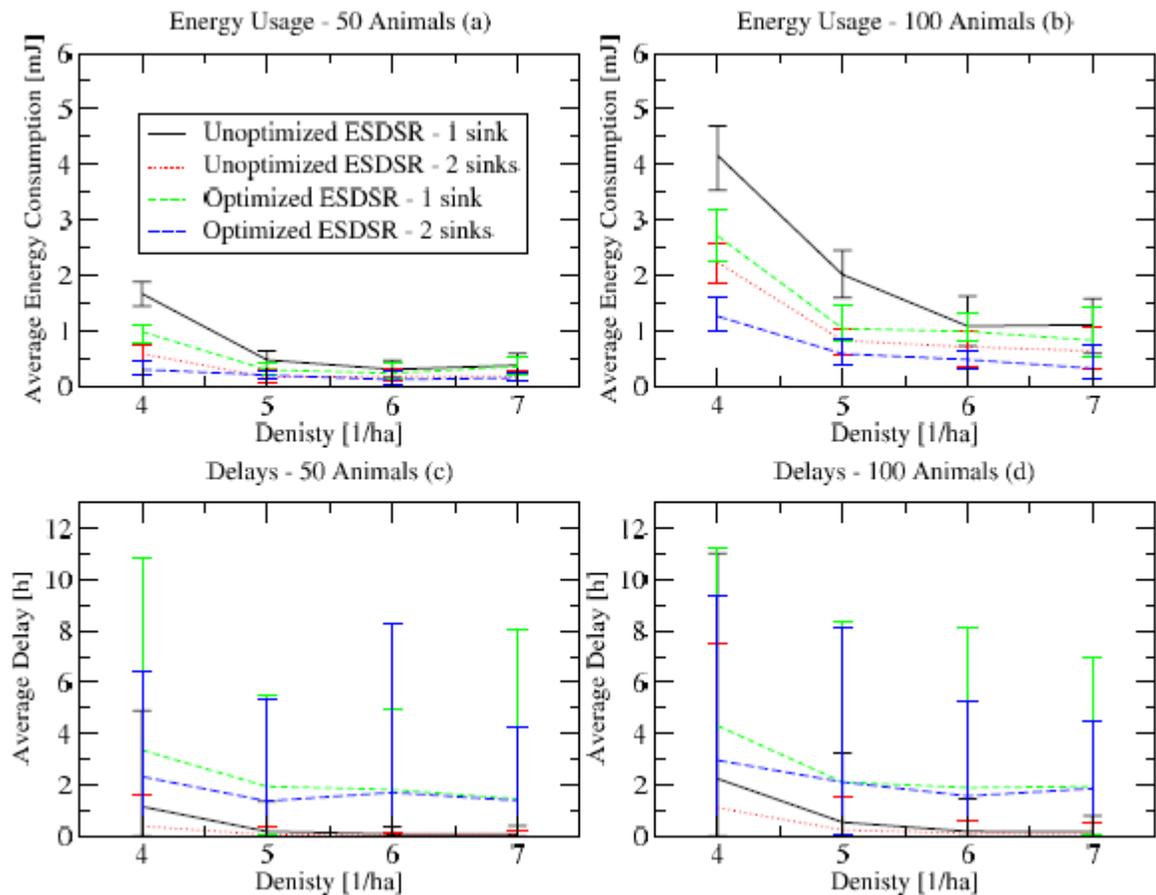
Figura 33. Arquitectura del sistema propuesto.



Fuente: B. Wietrzyk and M. Radenkovic, "Energy Efficiency in the Mobile Ad Hoc Networking Approach to Monitoring Farm Animals," in Networking, 2007. ICN '07. Sixth International Conference on, 2007, p. 1.

En este trabajo se propone un modelo formal para el sistema de monitoreo de ganado. Se formulan por ende el protocolo de enrutamiento MANET y se identifica un protocolo de enrutamiento MANET adecuado que satisface estos requisitos. También se ahorrará energía en las inundaciones basada en el descubrimiento de ruta adecuadas. Finalmente se muestra en la simulación que el enfoque propuesto es factible y las optimizaciones propuestas disminuyen el consumo de energía y aumentar la escalabilidad en términos de densidad y el número de nodos inalámbricos.

Figura 34. Resultados de simulación.



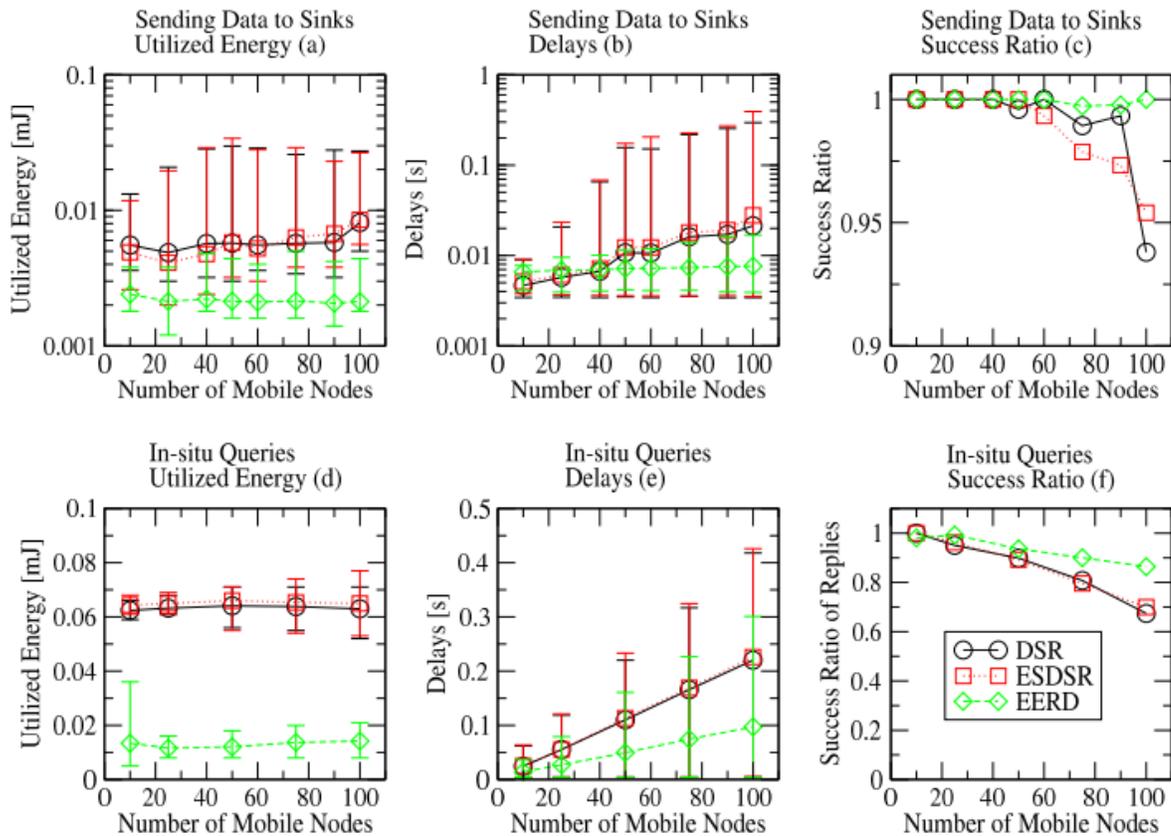
Fuente: B. Wietrzyk and M. Radenkovic, "Energy Efficiency in the Mobile Ad Hoc Networking Approach to Monitoring Farm Animals," in *Networking, 2007. ICN '07. Sixth International Conference on*, 2007, p. 1.

5.13 Practical MANETs for Pervasive Cattle Monitoring¹⁶.

La aplicación de Ad Hoc Re para la vigilancia del ganado tiene el potencial de aumentar la rentabilidad de la producción de ganado y un impacto positivo en el vivo todos los días del personal de la granja. Los principales retos de la investigación son la identificación y el perfeccionamiento de un protocolo de

enrutamiento MANET y el diseño de dicho protocolo. Este paper se enfoca sobre los experimentos de campo que realizaron con el fin de abordar esta cuestión. El enfoque trata sobre redes MANETs energéticamente eficientes para proporcionar monitoreo continuo de parámetros multidimensionales de la movilidad de los animales incluyendo intensidad caminar temporal y espacial y el consumo de alimento con el fin de detectar embarazos, enfermedades de los animales y la reducción de la eficiencia de los pastos.

Figura 35. Resultados de simulación.



Fuente: B. Wietrzyk, M. Radenkovic, and I. Kostadinov, "Practical MANETs for Pervasive Cattle Monitoring," in *Networking, 2008. ICN 2008. Seventh International Conference on*, 2008, pp. 14–23

En cada caso examinado ningún nodo estuvo interrumpido por el agotamiento de su capacidad de la batería. La Figura 36a muestra la energía utilizada por los animales sobre los cuales tenían los nodos montados para enviar datos a la unidad principal. En el protocolo EERE disminuye considerablemente el uso promedio de energía en comparación con DSR y ESDSR (48% -75%). El protocolo propuesto equilibra considerablemente la utilización de energía en comparación con DSR y ESDSR. En particular, los valores máximos de energía utilizada son 68% -87% más bajo en el caso de EERE. Estas mejoras pueden atribuirse a PCDI con optimizaciones propuestas y proponen métricas para la selección de rutas. La Figura 36b muestra retrasos en el envío de datos. Se observa que en el caso de

DSR y ESDSR los retrasos crecen con el número de nodos, mientras que en el caso de EREE los retrasos son casi constantes. En EREE también ofrece retrasos máximos más bajos en el caso de un mayor número de nodos móviles. La Figura 36c muestra el porcentaje de éxito (SR) para la entrega de los datos a los sumideros. Se ve que en todos los casos examinados la SR es muy alta. Fig. 3d muestra la energía utilizada por los nodos sobre los animales. La cantidad de energía utilizada es comparable al caso de la comunicación con el regadero, lo que justifica la optimización de este tipo de comunicación. En la Figura 36e se muestra que los retrasos aumentan con el número de animales. En el caso del protocolo de enrutamiento propuesto este aumento es más baja, lo que significa una mejor escalabilidad.

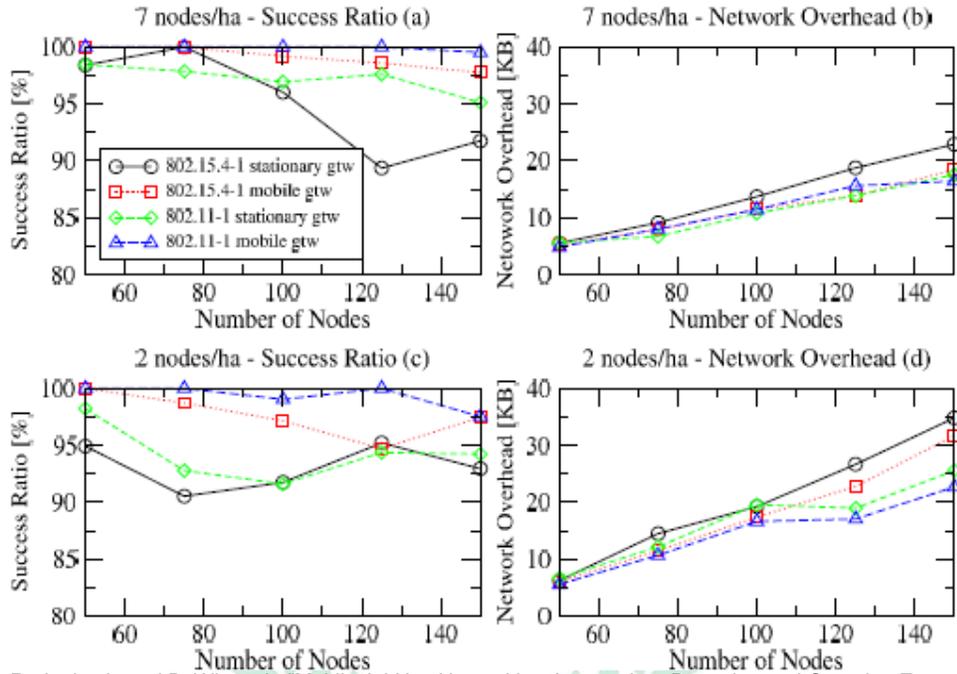
5.14 Mobile Ad Hoc Networking Approach to Detecting and Querying Events Related to Farm Animals¹⁷

En este trabajo se propone un enfoque de redes Ad Hoc de detección y consultade eventos relacionados con los animales de granja, tales como celo, enfermedades de los animales y la disminución de la eficiencia de los pastos. Este problema es esencial porque la falta de atención a la salud o el bienestar de los animales puede llevar a la disminución de la productividad. El enfoque propuesto guiado por los principios identificados utiliza la infraestructura de red disponible, pero también trabaja en unas condiciones especiales. Los datos se adquiere a partir de los dispositivos montados en animales utilizando comunicación inalámbrica confiable. Los datos se pueden también acceder desde un dispositivo portátil que conecta directamente a los dispositivos sensores con comunicación inalámbrica.

Se simularon diferentes números de nodos con la máxima y mínima densidad realista de la media (2 animales / ha y 7 animales / ha) con la duración de 2 horas simuladas, utilizando los simuladores de redes ns-2. Los nodos se mueven con la velocidad de 0,6 m/s (velocidad de movimiento preferida por el ganado) de acuerdo con el Modelo Random waypoint haciendo paradas de cero a un minuto. Su objetivo es enviar (una vez) 16 B de los datos a una pasarela usando el protocolo AODV. Si los nodos no pueden ponerse en contacto con una puerta de entrada, esperan un minuto y vuelven a intentarlo. Se han comparado los enfoques con una entrada fija y una entrada montada.

La puerta de enlace estacionaria se monta en el centro de uno de los lados del área de simulación. Utilizando un protocolo de bajo rango inalámbrico (IEEE 802.15.4) asumiendo gama 10m de transmisores y un protocolo de mayor alcance (IEEE 802.11b) asumiendo 250m de transmisores. Se mide porcentaje de éxito, es decir, el porcentaje de nodos que envió con éxito las mediciones a las puertas de enlace dentro de 2 horas y sobrecarga de la red, es decir la cantidad total de datos transmitidos. En la Figura 37 se plasman los resultados de las simulaciones.

Figura 36. Número de nodos de la red en SR.



Fuente: M. Radenkovic and B. Wietrzyk, "Mobile Ad Hoc Networking Approach to Detecting and Querying Events Related to Farm Animals," in Networking and Services, 2006. ICNS '06. International conference on, 2006, p. 109.

La Figura 37a muestra la influencia del número de nodos de red en el porcentaje de éxito (SR) en condiciones de más alta densidad de población posible. Se puede ver que en el caso de la comunicación de largo alcance (IEEE 802.11b) número de nodos y el tipo de puertas de enlace (fija o móvil, montado es decir animales) tienen poca influencia - el SR se mantiene por encima del 97,7%. En caso de comunicación de corto alcance (IEEE 802.15.4), porcentaje de éxito cae cerca de 90% en el caso de una puerta de entrada estacionaria pero es siempre por encima de 95% para una pasarela móvil. Esto es porque la puerta de enlace móvil cambiar su posición viene a una estrecha proximidad con mayor número de nodos.

La Figura 37b muestra la influencia del número de nodos de sobrecarga de la red (NO) en condiciones de mayor densidad de población posible. En todos los casos examinados, crece casi linealmente con el número de nodos. En caso de comunicación gama baja, utilizando una puerta móvil en lugar de una fija disminuye considerablemente sobrecarga de la red. Esto es porque la puerta de enlace móvil está más cerca del centro de la topología de una puerta de entrada estacionaria colocada en el borde de la zona de simulación de modo que las longitudes de caminos multi-hop son más cortos. La puerta de enlace móvil también viene a la proximidad de un mayor número de nodos que disminuye el número de reintentos.

La Figura 37c muestra la influencia del número de nodos de la red en SR en condiciones de posible baja densidad de población (2 animales / ha). Las diferencias entre las comunicaciones inalámbricas de alto y bajo ancho de banda son más pequeñas que las diferencias entre los tipos de puertas de enlace. Como se puede claramente la puerta de enlace móvil es considerablemente mejor que una estacionario. Esto es debido a la movilidad de la puerta de entrada supera el hecho de que la topología se divide en islas separadas de conectividad. El rendimiento en el peor de los casos, una sola puerta de entrada estacionaria y de bajo ancho de banda de comunicación, es todavía por encima de 90%.

La Figura 37d muestra la influencia del número de nodos de sobrecarga de la red (NO) en condiciones de baja densidad de población posible. NO en caso de comunicación de ancho de banda más bajo es mayor que la topología escasa causa algunos cuellos de botella vulnerables a la congestión. El uso de una puerta de enlace móvil produce menor sobrecarga de la red, ya que está más cerca del centro de la topología de la red y utiliza su movilidad a entrar en la proximidad de mayor número de nodos.

En resumen, en todos los casos al menos 89% de los nodos triunfan y envían sus mediciones a las puertas de enlace dentro de 2 horas, es decir, antes de que se produce el siguiente fragmento de datos examinados. Es un muy buen resultado que en la práctica los nodos tienen mucho más tiempo para transferir sus datos. La observación importante es que el uso de la gama baja y de bajo ancho de banda de la comunicación es suficiente para suministrar datos a los puertas independientemente del número de nodos y la densidad de población. Independientemente de la condición utilizando una puerta móvil mejora considerablemente la fiabilidad del suministro de datos y aumenta ligeramente la sobrecarga de la red.

Los Libertadores
Fundación Universitaria

6 MARCO TEÓRICO

6.1 MÉTRICAS PARA EL ANÁLISIS DE TRÁFICO EN REDES AD HOC

6.1.1 Retardo de extremo a extremo (DELAY).

Esta métrica es el tiempo promedio que transcurre desde que se inicia la transmisión de un paquete desde el nodo fuente al nodo de destino. Esta métrica incluye todos los retrasos posibles presentados en la red, aumentando principalmente por eventos de iniciación, mantenimiento y descubrimiento de rutas, tiempos de procesamiento, tiempos de duración de un paquete en la cola de nodos intermedios, retardos de retransmisión de paquetes para tráfico TCP, tiempo de acceso al canal, sobrecarga de los protocolos, entre otros. El retardo viene dada por:

Ecuación 1. Retardo

$$Retardo = \frac{\sum_{i=1}^{N_r} t_{Ri} - t_{Si}}{N_r}$$

Dónde:

t_{Ri} Es el tiempo de llegada de un paquete en el nodo destino y t_{Si} es el tiempo en el que el paquete fue enviado desde el nodo fuente y N_r es el número total de paquetes recibidos.

6.1.2 Jitter.

La métrica del Jitter es una medida de la variación del retardo de transferencia entre paquetes, puede depender de las rutas de los paquetes y flujo presente en las colas de los nodos intermedios. Hay varias definiciones de fluctuación que intentan capturar la variación del retardo de paquetes. En el desarrollo de este artículo la definición de Jitter tratada en IETF¹⁸, se adopta y se define a continuación:

Ecuación 2. Jitter

$$Jitter = E|T_{i+1} - T_i|$$

Dónde:

T_i es el tiempo de transmisión del paquete i y T_{i+1} representa el momento de la transmisión del siguiente paquete.

6.1.3 Packet Delivery Ratio (PDR).

Es la relación entre el número de paquetes de datos enviado desde el nodo fuente al nodo destino, en términos generales se evalúa el índice de paquetes de llegada, obteniendo la tasa de pérdidas de paquetes. Esta medida muestra la fiabilidad del

protocolo de enrutamiento, en cuanto mayor sea este índice, más fiable es la entrega de paquetes en la red. Esta métrica está dada por:

Ecuación 3. PDR

$$Indice_{PDR} = \frac{T_r}{T_s}$$

Dónde:

T_R es el número total de paquetes recibidos por el nodo de destino y T_S es el número total de paquetes enviados por el nodo fuente.

6.1.4 Throughputo rendimiento.

El Throughput o rendimiento se define como el número de bits recibidos con éxito por el nodo destino dividido por el total de tiempo de transmisión en cuestión de segundos, también se interpreta como la tasa de satisfactoria de transmisiones. A continuación se define como:

Ecuación 4. Throughput

$$Throughput = \frac{\sum T_{PR} \times 8}{T_{Ts} \times 1000}$$

Dónde:

T_{PR} Es el número total de paquetes recibidos por el nodo de destino y T_{Ts} es el tiempo total de la simulación.

6.2 PROTOCOLOS DE ENRUTAMIENTO

6.2.1 Ad-Hoc On-Demand Distance Vector (AODV).

AODV es un protocolo reactivo que descubre rutas en función de las necesidades de la red, mediante un mecanismo de descubrimiento de rutas, basado en tablas de enrutamiento que utiliza como información de entrada la distancia al nodo destino. Sin el uso de la información de enrutamiento del nodo fuente, AODV confía en la información de entrada registrada en su tabla de enrutamiento, esta información es construida por un proceso de inundación previa o de inicio en la red, al propagar un paquete RREP (*Route Reply*) al nodo fuente.

AODV utiliza números de secuencia de mantenimiento al nodo con el fin de actualizar la información de la tabla de enrutamiento y evitar bucles de enrutamiento¹⁹. Todos los paquetes de enrutamiento están marcados con estos números de secuencia. AODV mantiene estados basados en eventos de temporización en cada nodo, para monitorear el enrutamiento en las tablas, por lo

que las entradas que no son utilizadas presentan un tiempo mayor y se procede a eliminar esta posición de la tabla.

Este trabajo se realiza en conjunto con los nodos intermedios que mantienen información que indica al conjunto de nodos vecinos utilizando paquetes notificados con RERR (Route Error) cuando el enlace siguiente salto se destruye. Este paquete se remite por cada nodo predecesor a sus predecesores, borrando así todas las rutas que utilizaban el enlace de la ruta destruida. La propagación de errores de ruta en AODV se puede visualizar conceptualmente como un árbol cuya raíz es el nodo en el punto de ruptura y los nodos que intervinieron en la ruta hacia el nodo fuente son considerados como hojas²⁰.

La ventaja de utilizar AODV es la minimización del espacio de memoria ya que solo mantiene información de las rutas activas aumentando el rendimiento. Por otro lado, una de las desventajas de este protocolo, radica en que no es escalable y en grandes redes no funciona correctamente y no es compatible con enlaces y rutas asimétricas.

6.2.2 Destination-Sequenced Distance-Vector Routing(DSDV).

En DSDV la entrega de paquetes es debido a que utiliza rutas anteriores en caso de haber enlaces rotos. En DSDV la existencia de enlaces anteriores no implica que no sea una ruta válida hasta su destino final. Los paquetes pueden ser enviados a través de otros vecinos que pueden tener las rutas hacia en camino final²¹.

Es un esquema de enrutamiento impulsada por redes móviles ad hoc basado en el algoritmo de Bellman-Ford. Fue desarrollado por C. Perkins y P.Bhagwat en 1994. La principal contribución del algoritmo era resolver el problema de bucle de enrutamiento. Cada entrada en la tabla de enrutamiento contiene un número de secuencia, los números de secuencia son generalmente incluso si un enlace está presente; más, se utiliza un número impar. El número es generado por el destino, y el emisor debe enviar la próxima actualización con este número. Información de enrutamiento se distribuye entre los nodos mediante el envío de copias completas con poca frecuencia y actualizaciones incrementales más pequeñas con más frecuencia²². DSDV define dos métodos de actualización de la información: copias completas e incrementales; el primer modo se utiliza cuando la actualización para ser enviado sólo requiere un NPDU (unidad de red de paquetes de datos); El segundo modo se utiliza cuando se necesitan más NPDU para contener la entrada de actualizar, o cuando un nodo detecta un cambio sustancial a la topología de la red.

Naturalmente, la tabla se describe todas las posibles rutas alcanzables desde el nodo A, además del siguiente salto, número de saltos y el número de secuencia. Cada nodo también utiliza un valor de tiempo que define el tiempo en el que una

nueva ruta aprendió pueden ser validados y reenviado (para evitar bucles) de sedimentación.

6.2.3 Ad-Hoc On-Demand Multi Path Distance Vector Routing (AOMDV).

AOMDV es una extensión del protocolo AODV para encontrar múltiples rutas sin bucles y enlaces disjuntos²³. Las entradas de enrutamiento para cada destino contienen en una tabla la información de los siguientes saltos junto con la correspondencia del número de saltos. Los saltos siguientes deben de tener el mismo número de secuencia, para ayudar a realizar el seguimiento de una ruta. Para cada uno de los destinos, el nodo fuente mantiene el recuento de los saltos, que se define como el número máximo de saltos para todas las rutas. En el proceso de inundación y envío de información se encuentran mensajes duplicados y el anuncio de una ruta recibida por un nodo define una ruta alternativa al destino.

Las rutas libres de lazos están aseguradas para que un nodo acepte rutas alternativas al destino, si se tiene el conteo de un número de saltos menor al número de saltos anunciado para ese destino. Debido a que el número máximo de saltos es utilizado, se anuncia el número de saltos, por tanto, no cambia por el mismo número de secuencia²⁴. Cuando un anuncio de ruta se recibe en un destino con un mayor número de secuencia, la información de registro en la tabla del siguiente salto y el número de saltos anunciados se reinicializa.

AOMDV utiliza el descubrimiento de nodos, enlaces y rutas disjuntas. Para encontrar rutas de nodos disjuntos, cada nodo no rechaza de forma inmediata los paquetes RREQ (*Route Request*) duplicados. Cada RREQ que llegan a través de un nodo vecino diferente al nodo fuente define una trayectoria de nodo disjuntos. Esto se debe a que los nodos no se pueden difundir paquetes RREQ duplicado, por lo que, si dos o más paquetes RREQ lleguen a un nodo intermedio a través de un nodo vecino diferente al nodo fuente, no podría haber atravesado el mismo nodo intermedio. En un intento de conseguir enlaces de conexión disjuntos y múltiples rutas, los nodos destino responden a paquetes duplicados RREQ, donde el nodo destino sólo responde a las RREQ únicos que llegan por sus nodos vecinos. Después del primer salto, los paquetes RREP (*Route Reply*) siguen las rutas inversas o hacia atrás, que corresponde a enlaces de nodos disjuntos. Las trayectorias de cada paquete RREP se pueden interceptar en un nodo intermedio, pero la decisión de tomar una ruta inversa diferente a la del nodo fuente garantiza enlace disjuntos.

La ventaja de usar AOMDV es que permite a los nodos intermedios responder a RREQ, sin dejar de seleccionar caminos disjuntos. Pero, AOMDV aumenta la carga de la red durante el descubrimiento de rutas, debido al aumento de las inundaciones y por estar diseñado para encontrar trayectorias múltiples, al nodo/nodos destino.

6.2.4 Dynamic Source Routing (DSR).

Es un protocolo de enrutamiento para redes malladas inalámbricas. Este protocolo se basa realmente en el enrutamiento de origen lo que mantiene toda la información de enrutamiento en los nodos móviles y continuamente se están actualizando dichos nodos. Sólo tiene dos fases principales, que son la ruta de descubrimiento y ruta de mantenimiento.

También es usado para limitar el ancho de banda consumido por los paquetes de control en redes inalámbricas ad hoc mediante la eliminación de los mensajes mesa de actualización periódicos prescritos en el enfoque basado en tablas.

Este protocolo utiliza un enfoque en el que elimina la necesidad de inundar periódicamente la red con mensajes de actualización de tabla que se requieren en un enfoque basado en tablas. En un enfoque reactivo (*on demand*) como ésta, una ruta se establece sólo cuando es necesario y por lo tanto la necesidad de encontrar las rutas a todos los demás nodos de la red según lo requiera será impulsada por tablas.

Los nodos intermedios también utilizan la información de la caché ruta eficiente para reducir la sobrecarga de control. La desventaja de este protocolo es que el mecanismo de mantenimiento de rutas no repara localmente un enlace roto. La configuración de la conexión de retardo es mayor que en los protocolos de tablas. A pesar de que el protocolo se realiza bien en entornos estáticos y de baja movilidad, el rendimiento se degrada rápidamente con el aumento de la movilidad. Además, una sobrecarga considerable de enrutamiento está involucrada debido al mecanismo de fuente-enrutamiento empleado en DSR. Esta sobrecarga de enrutamiento es directamente proporcional a la longitud del camino²⁵.

6.3 CARACTERÍSTICAS PRINCIPALES DE UN CULTIVO DE PAPA

6.3.1 Plagas y enfermedades de la papa.

Antes de presentar un breve recuento de las principales plagas y enfermedades que afectan la papa en Colombia, es necesario tener en cuenta las siguientes consideraciones:

El establecimiento de un cultivo (población homogénea de plantas), tiene un gran impacto en el ecosistema natural donde existe un equilibrio entre los diferentes organismos (fitopatógenos y benéficos).

La familia solanácea a la cual pertenece la papa, está expuesta a una gran diversidad de agentes causales de daños entre los cuales se distinguen insectos, nematodos, hongos, virus, bacterias, protozoarios y demás organismos que afectan crecimiento, rendimiento y calidad.

El “monocultivo” es un modelo agrícola de altas exigencias en los planes de manejo fitosanitario que no se pueden limitar a la implementación del control químico.

Las diferentes prácticas culturales (remoción de suelo, aplicación de plaguicidas y fertilizantes, uso de riego) generan alteraciones en los recursos suelo, agua y aire, así como en la población de organismos fitopatógenos y benéficos; los primeros, encuentran una mejor condición para su dispersión, establecimiento y daño y, los segundos desventajas para cumplir su función regulatoria.

6.3.2 Plagas.

Las plagas de mayor impacto económico en Colombia en el cultivo de papa, están representadas en su mayoría por insectos:

Tabla 10. Principales plagas de la papa.

SUELO/TUBÉRCULO	FOLLAJE	ALMACENAMIENTO
Gusano Blanco de la papa <i>Premnotrypes vorax</i>	Pulguilla <i>Epitrix cucumeris</i>	Polilla pequeña o Palomilla <i>Phthorimaea operculella</i>
Polilla Guatemalteca de la papa <i>Tecia solanivora</i>	Tostón, mosco o entretelado <i>Lyriomyza quadrata</i> <i>Lyriomyza huidobrensis</i>	Polilla Guatemalteca de la Papa <i>Tecia solanivora</i>
Tiroteador <i>Naupactus</i> sp.	Muques o comedores de follaje <i>Copitarsia consuetata</i> <i>Pedidroma</i> sp.	Afidos <i>Rhopalosiphoninus latysiphon</i>
Polilla Pequeña o Palomilla <i>Phthorimaea operculella</i>	Polilla Pequeña o Palomilla <i>Phthorimaea operculella</i>	Polilla Gigante de la Papa <i>Symmetrischema plaesiosema</i>
Chisa, mojoyoy o morrongo <i>Ancognatha scarabaeoides</i> <i>Phyllophaga obsoleta</i>	Polilla Gigante de la papa <i>Symmetrischema plaesiosema</i>	
Babosa <i>Milax gagates</i>	Trips <i>Frankliniella tuberosi</i> <i>Trips palmi</i>	
	Chupadores Moscas blancas <i>Trialeurodes vaporariorum</i>	
	Trozadores <i>Agrotis ipsilon</i> <i>Feltiasp.</i>	

Fuente: FEDEPAPA y Gobernación de Cundinamarca, 2010. Recopilación de la investigación del sistema productivo papa criolla.

6.3.3 La papa en el mundo.

El agro negocio de la papa se ha expandido en la mayor parte del mundo y su producción y consumo muestran un aumento constante. La producción mundial de papa en fresco ha aumentado en los últimos años; pasó de 308 millones de toneladas a 322 millones entre 1998 y 2007, es decir, una tasa de crecimiento promedio anual de 0,49%.

El principal productor de papa en el año 2007 fue China con más de 70 millones de toneladas cosechadas (22,37%) en 5 millones de hectáreas, seguida por la Federación Rusa que produjo 38 millones de toneladas; le siguieron India, Estados Unidos y Ucrania; curiosamente, ninguno de los principales países productores de papa pertenece a la zona andina, considerado como su centro de diversidad genética.

En países industrializados como Estados Unidos, Japón y en la Unión Europea, la papa se consume especialmente frita (papa en bastones o a la francesa y hojuelas o chips), también horneada, cocinada, gratinada, en sopas, ensaladas, como puré

o croquetas. De otra parte, el consumo en la región andina, se prefiere cocinada, en puré, frita, en sopas, al horno o rellena.

En relación a los procesos industriales se destacan las papas en hojuelas (chips) y las papas pre cocidas, pre fritas congeladas (*french fries*, papa a la francesa o papa en bastones); otros usos industriales de la papa se concentran en la elaboración de harinas (blancas y negras) y almidón (utilizado para tortas, pasteles, helados, embutidos y galletas). También se emplea en las industrias de madera, papel, medicinas, tejidos como goma o aglutinante y en la industria petrolera. De igual manera, un uso reconocido a nivel mundial es la destilación del alcohol de papa para la elaboración de vodka.

6.3.4 La papa en Colombia.

La papa es el producto agrícola más requerido por los estratos socioeconómicos ya que es de fácil adquisición y una gran fuente de carbohidratos casi con un consumo de 62kg/año. Se cultiva durante todo el año en cerca de 133.865 hectáreas calculadas para el año 2012, dispersas en unos 250 municipios de la zona fría y muy fría andina.

La papa en Colombia tiene un rango potencial de adaptación entre 1500 y 4000 msnm; comercialmente se cultiva entre 2000 y 3500 msnm. Alturas inferiores o superiores a éstas, se consideran marginales. La zona óptima de producción se concentra entre 2500 y 3000 msnm.

Alrededor del sistema productivo papa, se vinculan directamente unos 90.000 agricultores, mayoritariamente pequeños productores, e indirectamente dependen de la papa otras 400.000 familias que hacen parte del comercio, semillas, agro insumos, transporte terrestre, comercialización y demás actividades conexas. El cálculo de empleo para 2009 del Consejo Nacional de la Papa se estimó en 77.369 empleos directos y 232.108 empleos indirectos²⁶.

6.4 CARACTERÍSTICAS PRINCIPALES DE UN CULTIVO DE FRIJOL.

6.4.1 Época de Siembra.

La Época de siembra más adecuada para frijol es en la que hay poca precipitación de lluvias ya que se disminuye el riesgo de que el grano se vea afectado por la humedad. En forma general hay tres épocas específicas en las que se siembra el grano las cuales son: a) Siembras de Primavera entre el 15 de mayo y el 15 de Junio, se escoge esta época de tal manera que la madurez del grano coincida con la época seca. b) Siembras segunda o posterior por año agrícola en el país y se realizan en los meses de septiembre y octubre. En esta época se escoge una fecha seca en la cual no haya precipitaciones. C) Siembras de Apante Estas siembras se realizan en el Norte y litoral atlántico, son Estas dos regiones del país y se hacen entre los meses de Diciembre y enero.

7 PLANTEAMIENTO DEL PROBLEMA

Los requerimientos de las actuales aplicaciones multimedia hacen necesario que las redes cuenten con cierto nivel de QoS al usuario. Este problema está prácticamente resuelto en redes fijas, pero las características especiales de las redes móviles ad-hoc hacen necesario una serie de estudios para afrontar este problema. Principalmente, la topología dinámica y los escasos recursos de los nodos hacen que la carga del mecanismo de provisión de QoS sea lo más ligera posible en cuanto a recursos de red (ancho de banda). Se han identificado cuatro tópicos básicos para el soporte de QoS: modelo, señalización, enrutamiento, y acceso al medio. El conjunto de todas estas características determina la capacidad y la calidad del servicio.

Entre los problemas encontrados que se relacionan con dichas características se encuentran: la velocidad de los nodos y las formas y probabilidades de conexión, ya que este tipo de redes son móviles y si los nodos se mueven a velocidades exageradas no realizan una buena conexión con los demás, teniendo en cuenta que cada protocolo realiza dicha tarea de diferente manera, además de los problemas relacionados con el ruido externo y la potencia necesaria para realizar satisfactoriamente la conexión entre nodos. Tratando de darle solución a dichos inconvenientes se han realizado diversas simulaciones variando parámetros, número de nodos, diferentes protocolos y tamaños de escenarios variados.

Los Libertadores
Fundación Universitaria

8 METODOLOGÍA

Para el análisis de la información se emplean trayectorias cerradas con diferentes formas, la primera forma utilizada es una estrella figura 38. La cual tiene 12 saltos para completar la forma de estrella. La trayectoria número dos es un girasol la cual se muestra en la figura 39. Esta trayectoria se compone de 18 saltos antes de finalizar la forma deseada. La trayectoria número tres es un ocho figura 40. Esta trayectoria se compone de 25 saltos antes de finalizar la forma deseada. Cabe resaltar que cada uno de los movimientos diseñados inicia desde un punto aleatorio y la dirección hacia donde corre también es completamente aleatoria.

Figura 37. Trayectoria con forma de estrella.

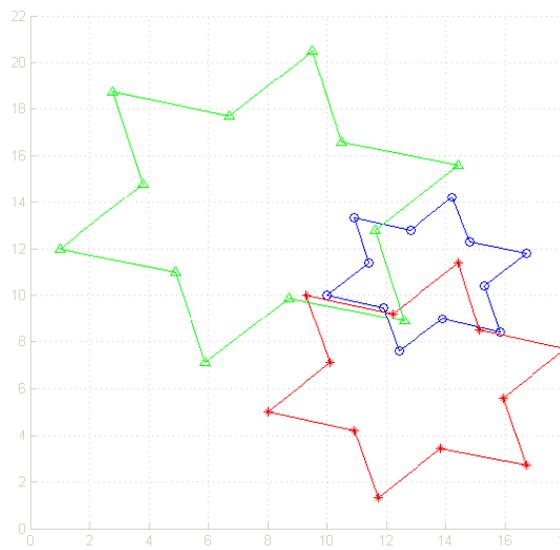


Figura 38. Trayectoria con forma de girasol.

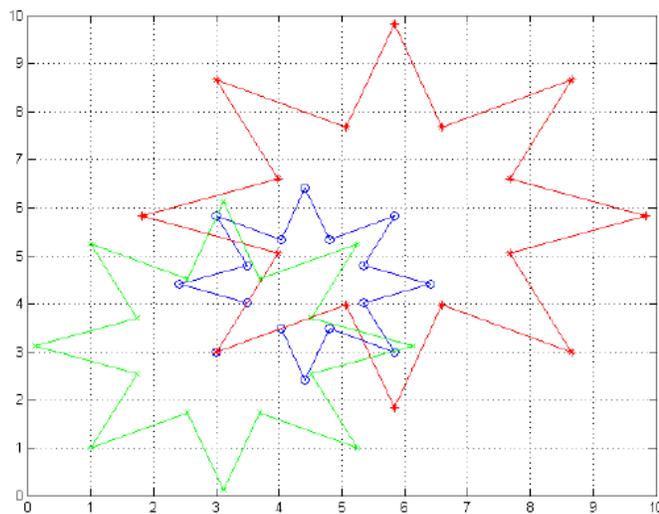
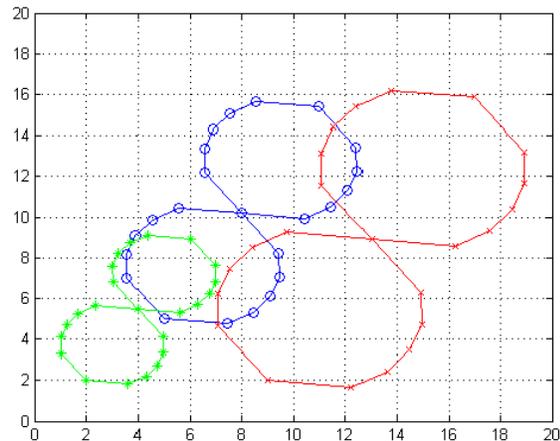


Figura 39. Trayectoria con forma de ocho.



La fuente de tráfico a evaluar es CBR (Constant Bit Rate), entre un nodo fuente y un nodo destino extendidos aleatoriamente en la red. El tamaño de los paquetes utilizados fue de 512 bytes a una tasa de 150 paquetes por segundo. El número de fuentes de tráfico TCP con aplicación FTP utilizado fue aproximadamente equivalente a la variación de la tasa de envío con la mitad de las fuentes de tráfico del total del número de nodos, para un total de diez enlaces TCP. En una área de simulación de 200m x 200m con una velocidad variable entre los 0 Km/h y 20Km/h. con 20 nodos realizando las trayectorias cerradas mencionadas anteriormente.

Tabla 11. Parámetros de simulación principal.

PARAMETROS DE SIMULACION	
TIPO DE PARAMETRO	VALOR
SIMULADOR DE RED	NS.2-31
TIEMPO DE SIMULACION	900 s
AREA DE SIMULACION	200m X 200m
NUMERO DE NODOS	20
RANGO DE TRANSMISION	50m, 75m, 100m
VELOCIDAD	0 Km/h, 20 Km/h
TIPO DE DATOS	CBR
FUENTE/DESTINO	RANDOM
TAMAÑO DE PAQUETES	512 Bytes
PROTOCOLO MAC	IEEE 802,11 DCF
MODELO DE PROPAGACION	Two-ray Ground Reflection Model
MODELO DE MOVILIDAD	PUNTOS ALEATORIOS
TIPO DE CANAL	CANAL INALAMBRICO
MODELO DE ANTENA	OMNIDIRECCIONAL
TIPO DE INTERFAZ COLA	COLA DE PRIORIDAD (64 PAQUETES MAXIMO)

9 ALCANCES Y LIMITACIONES

El proyecto tiene como alcance realizar la simulación de diferentes tipos de movimientos de dispositivos en terrenos agrícolas con el fin de efectuar una adquisición de datos, análisis de suelos, el estado de las plantas y las diferentes plagas que puedan afectar estas mismas, realizando la comparación entre tres diferentes protocolos utilizados en redes Ad Hoc para determinar el rendimiento de estos según sea el movimiento realizado.

Las limitaciones que enfrenta este proyecto es la falta de información sobre la implementación de redes Ad Hoc sobre temas de agricultura, ya que los enfoques de estas redes se destinan para otros campos de investigación; otra limitante es la implementación de los dispositivos en físico, ya que los recursos son escasos y se carece de información, por esta razón sólo se efectuarán simulaciones en el software NS2, así que única y exclusivamente se realiza análisis de tráfico con las topologías propuestas, sin llegar a realizar pruebas en físico, omitiendo cualquier tipo de adquisición de datos de algún dispositivo.

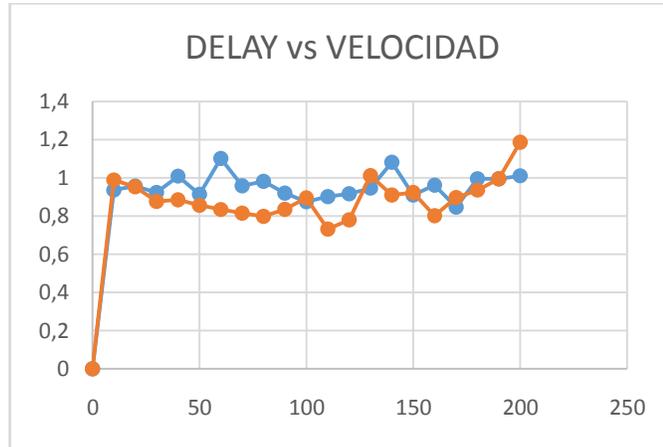


10 ANÁLISIS DE RESULTADOS

10.1 RESULTADOS DE SIMULACIÓN PARA EL MOVIMIENTO EN GIRASOL.

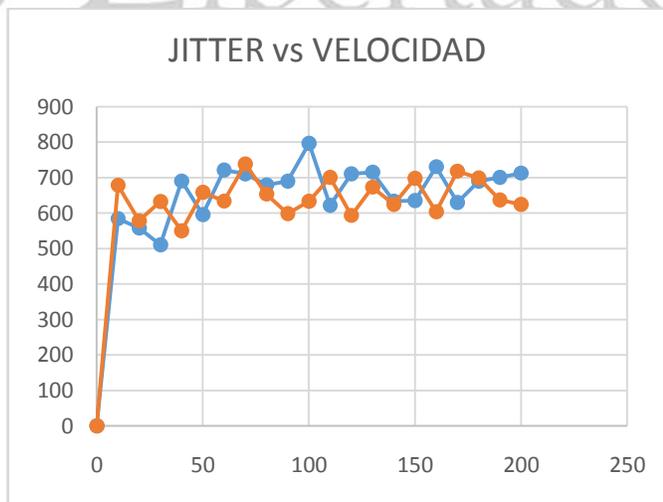
Los primeros datos que se mostrarán a continuación, es el comparativo entre el protocolo AODV y el protocolo DSR para el movimiento GIRASOL.

Figura 40. DELAY Vs Velocidad por Paquete (NARANJA Protocolo AODV, AZUL Protocolo DSK).



En la figura 41 se muestra el resultado del DELAY o retraso que se obtuvo en cada protocolo según la velocidad del paquete. Se muestra que el DELAY no supera 1,2 segundos, lo que conlleva a decir que los dos protocolos son fiables comparándolos con su respuesta en DELAY.

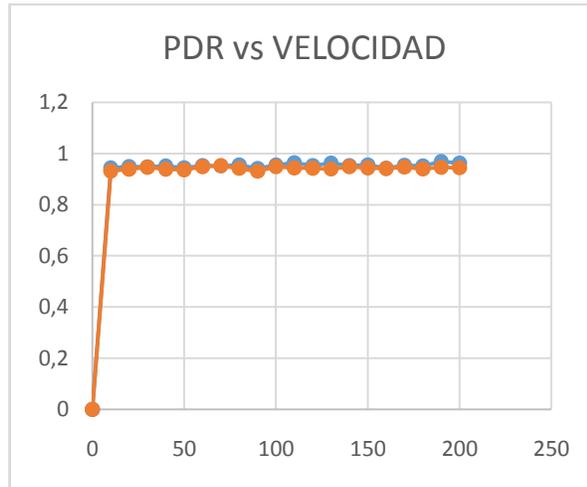
Figura 41. JITTER Vs Velocidad por Paquete (NARANJA Protocolo AODV, AZUL Protocolo DSK).



En la figura 42 se muestra el JITTER obtenido sobre los protocolos AODV y DSR.

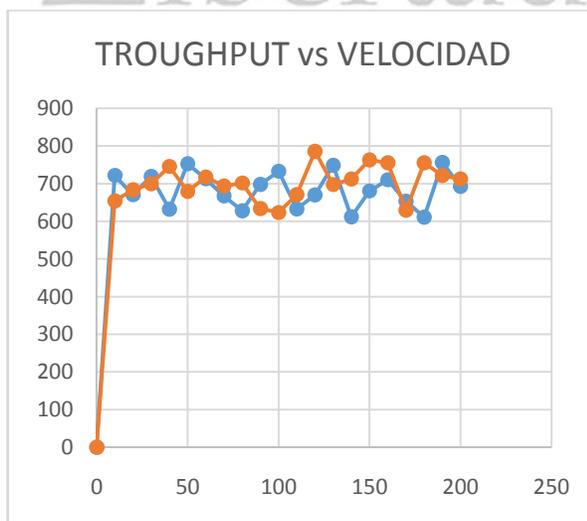
Según la figura anterior, no se demuestra una ventaja clara sobre quien obtiene el mayor JITTER en los protocolos usados. Por lo tanto por el ítem JITTER los protocolos AODV y DSK son buenos utilizando el movimiento girasol.

Figura 42. PDR Vs Velocidad por Paquete (NARANJA Protocolo AODV, AZUL Protocolo DSK).



En la figura 43 se muestra el resultado del PDR por cada protocolo. Mediante la modalidad de movimiento en girasol se pierde aproximadamente 1 paquete por cada 100 paquetes enviados. Visualizando la gráfica, se evidencia que no hay diferencia alguna con el PDR de un protocolo respecto al otro, lo que conlleva que estos protocolos son aceptables para el desarrollo del movimiento girasol.

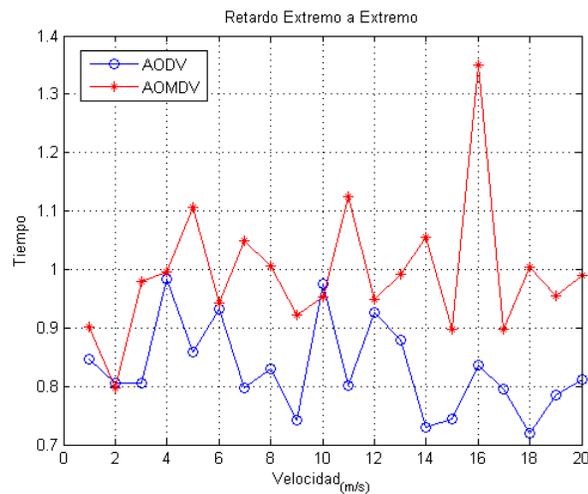
Figura 43. TROUGHPUT Vs Velocidad por Paquete (NARANJA Protocolo AODV, AZUL Protocolo DSK).



En la figura 44 se muestra TROUGHPUT vs velocidad de transmisión. En los dos se muestra un TROUGHPUT o rendimiento en los que se comparan con la figura dada por los resultados de este ítem, se llega a la conclusión de que los dos protocolos se mantienen en un rendimiento o TROUGHPUT utilizando el movimiento girasol.

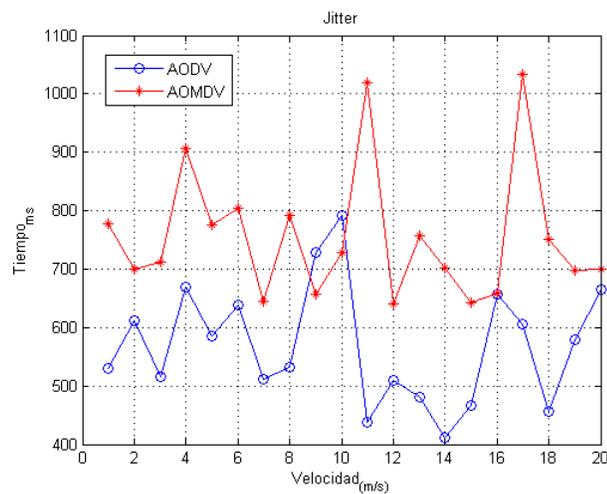
10.2 RESULTADOS DE SIMULACIÓN PARA EL MOVIMIENTO EN ESTRELLA

Figura 44. Retardo extremo a extremo movimiento estrella.



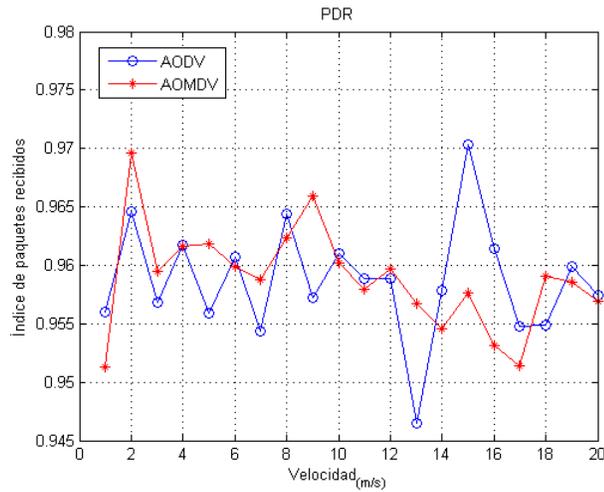
En la figura 45 se muestra el uso del protocolo AODV y AOMDV. Se evidencia un mayor retardo sobre el protocolo AODV, indica que al usar este protocolo para este movimiento lleva a que haya alguna pérdida de paquetes con respecto al protocolo AOMDV ya que registra un bajo retardo de extremo a extremo

Figura 45. Jitter movimiento de estrella.



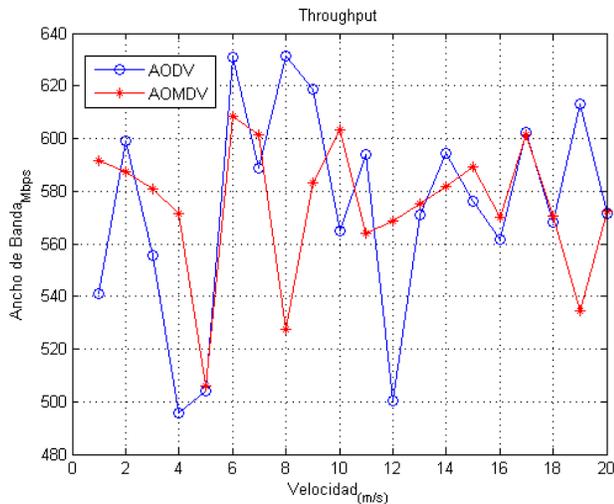
En la figura 46 se muestra el Jitter entre los protocolos ya mencionados. Se detecta según la gráfica mayor Jitter sobre el protocolo AOMDV. El protocolo AODV demuestra una eficiencia sobresaliente con la métrica medida que es el Jitter.

Figura 46. PDR movimiento en estrella.



En la figura 47 se muestra el PDR. No hay mucha diferencia entre los dos protocolos evaluados, pero hay un alto índice de paquetes recibidos lo que refleja que el movimiento en estrella fleja muy buenos resultados respecto a la métrica evaluada.

Figura 47. Throughput movimiento estrella.

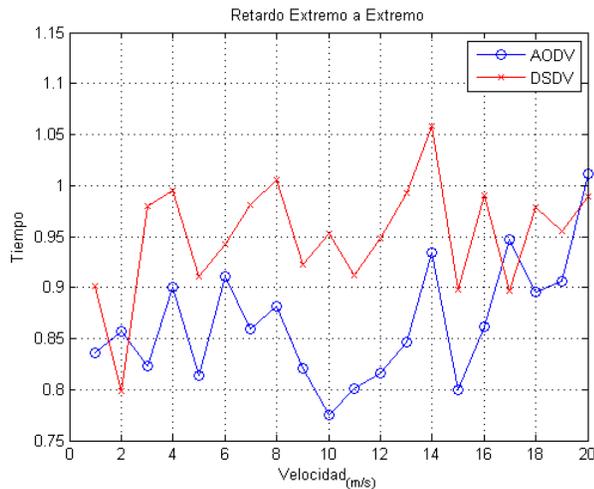


En la figura 48 se muestra la métrica del Throughput. No hay mucha diferencia entre los protocolos evaluados, los protocolos reflejan un rendimiento similar pero

a la vez refleja un rendimiento óptimo y eficiente.

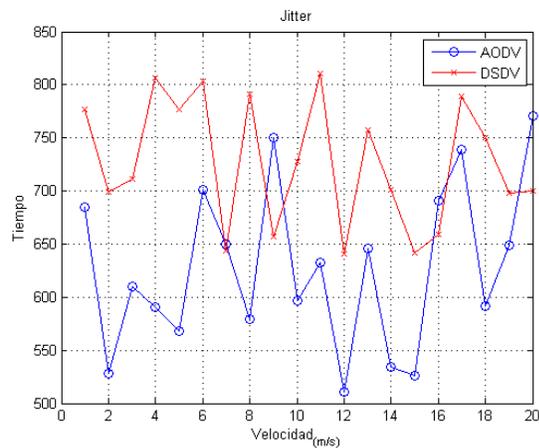
10.3 RESULTADOS DE SIMULACIÓN PARA EL MOVIMIENTO EN OCHO

Figura 48. Retardo extremo a extremo movimiento en ocho.



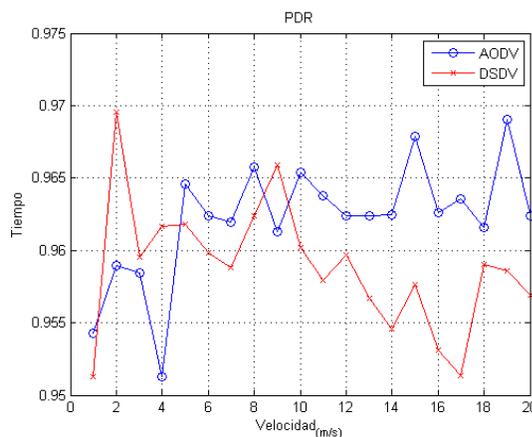
En la figura 49 se muestra el uso de los protocolos AODV y DSDV sobre la métrica del retardo extremo a extremo. Comparando con los anteriores movimientos, es muy bajo el retardo que hay con el uso del protocolo AODV por ende frente a esta métrica este movimiento es muy óptimo y las probabilidades serán mínimas de que se presente perdidas de paquetes.

Figura 49. Jitter movimiento en ocho.



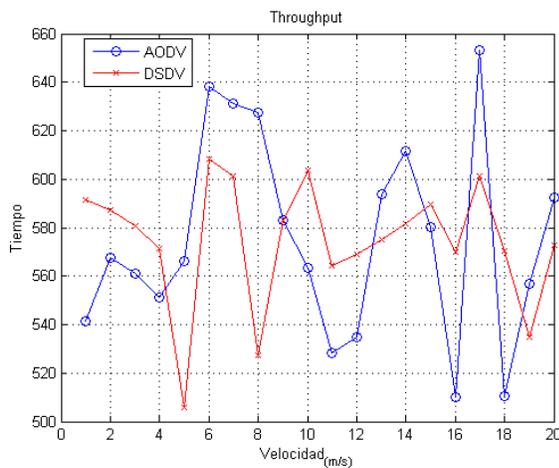
En la figura 50 se muestra el resultado de la métrica de Jitter sobre los protocolos anteriormente mencionados. No hay diferencias amplias con los movimientos de estrella y girasol.

Figura 50. PDR movimiento en ocho.



En la figura 51 se muestra la métrica de PDR, se muestra una leve diferencia con respecto a los anteriores movimientos pero no influye en la fiabilidad que se observa en los demás protocolos.

Figura 51. Throughput movimiento en ocho.



En la figura 52 se muestra la métrica del Throughput. Se aprecia un buen rendimiento sobre los dos protocolos evaluados, hay una leve mejoría usando el movimiento en S, pero comparando el resto de movimientos evaluados, este movimiento sobresale de los demás por su alto rendimiento y eficiencia.

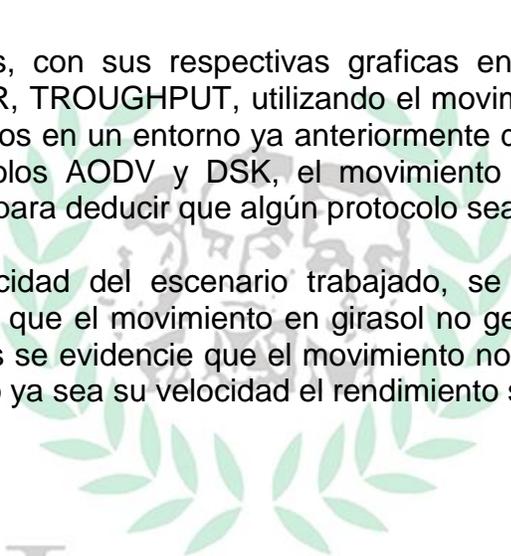
CONCLUSIONES

Para el escenario especificado de 20 nodos móviles moviéndose a una velocidad uniforme y distribuida de forma normal, con un movimiento reconocido por una trayectoria en forma de ocho, se alcanzó una pérdida de paquetes que tiende a permanecer invariable frente a la variación de velocidad.

El ancho de banda se limita al promedio del tráfico background presente en la red, a pesar de realizarse una comunicación punto a punto, la calidad de servicio de la red no se deteriora.

Según los resultados, con sus respectivas graficas en la cuales se midieron, DELAY, JITTER, PDR, TROUGHPUT, utilizando el movimiento de un girasol para el movimiento de nodos en un entorno ya anteriormente descrito, se concluye que utilizando los protocolos AODV y DSK, el movimiento eficiente y no presenta alguna variación alta para deducir que algún protocolo sea mejor que el otro.

Al aumentar la velocidad del escenario trabajado, se evidenció mediante las gráficas y resultados, que el movimiento en girasol no genera cambios bruscos o notorios en los cuales se evidencie que el movimiento no sea útil con velocidades altas, sino al contrario ya sea su velocidad el rendimiento será eficiente.



Los Libertadores
Fundación Universitaria

BIBLIOGRAFÍA

1. Zanjireh, M.M.; Larijani, H.; Popoola, W.O. "Activity-aware clustering algorithm for wireless sensor networks", Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2014 9th International Symposium on, On page(s): 122 – 127.
2. Zanjireh, M.M.; Larijani, H.; Popoola, W.O.; Hahrabi A." Analytical Modelling of ANCH Clustering Algorithm for WSNs", The Thirteenth International Conference on Networks, On page(s): 68 – 73.
3. Zanjireh, M.M. ; Sch. of Eng. & Built Environ., Glasgow Caledonian Univ., Glasgow, UK ; Larijani, H. ; Popoola, W.O. "Activity-aware clustering algorithm for wireless sensor networks" Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2014 9th International Symposium on, On page(s): 122 – 127.
4. R. Anggoro, T. Kitasuka, R. Nakamura, and M. Aritsugi, "Performance Evaluation of AODV and AOMDV with Probabilistic Relay in VANET Environments," in Networking and Computing (ICNC), 2012 Third International Conference on, 2012, pp. 259–263.
5. L. Bello, P. Bakalis, S. J. Manam, T. I. Eneh, and K. A. Anang, "Power Control and Performance Comparison of AODV and DSR Ad Hoc Routing Protocols," in Computer Modelling and Simulation (UKSim), 2011 UkSim 13th International Conference on, 2011, pp. 457–460.
6. A. M. Kanthe, D. Simunic, and R. Prasad, "Comparison of AODV and DSR on-demand routing protocols in mobile ad hoc networks," in Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN), 2012 1st International Conference on, 2012, pp. 1–5.
7. P. Sarkar and H. Paul, "Comparative performance analysis of GZRP, AOMDV and DSR in MANETs," in Computational Intelligence and Information Technology, 2013. CIIT 2013. Third International Conference on, 2013, pp. 175–178.
8. B. Paul, K. A. Bhuiyan, K. Fatema, and P. P. Das, "Analysis of AOMDV, AODV, DSR, and DSDV Routing Protocols for Wireless Sensor Network," in Computational Intelligence and Communication Networks (CICN), 2014 International Conference on, 2014, pp. 364–369.

9. N. Sharma, S. Rana, and R. M. Sharma, "Provisioning of Quality of Service in MANETs performance analysis; comparison (AODV and DSR)," in Computer Engineering and Technology (ICCET), 2010 2nd International Conference on, 2010, vol. 7, pp. V7–243–V7–248.
10. Amirhossein Moravejosharieh, Hero Modares, Rosli Salleh and Ehsan Mostajeran "Performance Analysis of AODV, AOMDV, DSR, DSDV Routing Protocols in Vehicular Ad Hoc Network" Research Journal of Recent Sciences, 2013.
11. P. Jammulaiah and Dr. P. Chenna Reddy, "Simulation and Comparison of AOMDV, AODV and DSR in Manets ", 2013.
12. S. S. Manvi, M. S. Kakkasageri, and C. V Mahapurush, "Performance Analysis of AODV, DSR, and Swarm Intelligence Routing Protocols In Vehicular Ad Hoc Network Environment," in Future Computer and Communication, 2009. ICFCC 2009. International Conference on, 2009, pp. 21–25.
13. P. Marino, F. P. Fontan, M. A. Dominguez, and S. Otero, "Ad-hoc WSN in Biological Research," in Digital System Design, Architectures, Methods and Tools, 2009. DSD '09. 12th Euromicro Conference on, 2009, pp. 841–848.
14. M. Gonzalez, J. Schandy, N. Wainstein, L. Barboni, A. Gomez, and C. Croce, "Wireless image-sensor network application for population monitoring of lepidopterous insects pest (moths) in fruit crops," in Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International, 2014, pp. 1394–1398.
15. B. Wietrzyk and M. Radenkovic, "Energy Efficiency in the Mobile Ad Hoc Networking Approach to Monitoring Farm Animals," in Networking, 2007. ICN '07. Sixth International Conference on, 2007, p. 1.
16. B. Wietrzyk, M. Radenkovic, and I. Kostadinov, "Practical MANETs for Pervasive Cattle Monitoring," in Networking, 2008. ICN 2008. Seventh International Conference on, 2008, pp. 14–23.
17. M. Radenkovic and B. Wietrzyk, "Mobile Ad Hoc Networking Approach to Detecting and Querying Events Related to Farm Animals," in Networking and Services, 2006. ICNS '06. International conference on, 2006, p. 109.
18. T. Clausen, B. Adamson, and C. Dearlove, "Jitter considerations in mobile ad hoc networks (MANETs)," 2008.

19. C. Perkins, E. Belding-Royer, S. Das, and others, "RFC 3561-ad hoc on-demand distance vector (AODV) routing," Internet RFCs, pp. 1–38, 2003.
20. W. Araujo Albano, M. Nogueira, and J. de Souza, "A Taxonomy for Resilience in Vehicular Ad hoc Networks," *Lat. Am. Trans. IEEE (Revista IEEE Am. Lat.)*, vol. 13, no. 1, pp. 228–234, 2015.
21. Khan, K.; Zaman, R.U.; Reddy, K.A.; Reddy, K.A.; Harsha, T.S., "An Efficient DSDV Routing Protocol for Wireless Mobile Ad Hoc Networks and its Performance Comparison," *Computer Modeling and Simulation*, 2008. EMS '08. Second UKSIM European Symposium on , vol., no., pp.506,511, 8-10 Sept. 2008.
22. Perkins, Charles E. and Bhagwat, Pravin (1994). "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers" (PDF). Retrieved 2006-10-20.
23. U. S. Kushwaha and P. K. Gupta, "AOMDV routing algorithm for Wireless Mesh Networks with local repair (AOMDV-LR)," in *Communications and Signal Processing (ICCSP)*, 2014 International Conference on, 2014, pp. 818–822.
24. A. Vidwans, A. K. Shrivastava, and M. Manoria, "QoS Enhancement of AOMDV Routing Protocol Using Queue Length Improvement," in *Communication Systems and Network Technologies (CSNT)*, 2014 Fourth International Conference on, 2014, pp. 275–278.
25. Elizabeth M. Royer and C.-K. Toh. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks, *IEEE Personal Communications Magazine*, April 1999, pp. 46-55.
26. FEDEPAPA y Gobernación de Cundinamarca, 2010. Recopilación de la investigación del sistema productivo papa criolla.
27. El cultivo de frijol, Guía para uso de empresas privadas, consultores individuales y Productores <https://curlacavunah.files.wordpress.com/2010/04/el-cultivodelfrijol.pdf>

ANEXOS

ANEXO A. CÓDIGO FUENTE PARA GENERACIÓN DE SIMULACIÓN EN NS2

```
# Define options
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) 500 ;# max packet in ifq
set val(nn) 20 ;# number of mobilenodes
set val(rp) AOMDV ;# routing protocol
set val(cp) "./trafico.tcl"
set val(sc) "./scenario.tcl"
set val(x) 200 ;# X dimension of topography
set val(y) 200 ;# Y dimension of topography
set val(stop) 40 ;# time of simulation end
set ns [new Simulator]
set tracefd [open out.tr w]
set windowVsTime2 [open win.tr w]
set namtrace [open simwrls.nam w]

$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
# Create nn mobilenodes [$val(nn)] and attach them to the channel.

# configure the nodes
$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) \
```

```

        -channelType $val(chan) \
        -topoInstance $topo \
        -agentTrace ON \
        -routerTrace ON \
        -macTrace OFF \
        -movementTrace ON
    for {set i 0} {$i < $val(nn)} {incr i} {
        set node_($i) [$ns node]
    }
# Posicion Inicial

# Define Escenario

#puts "Loading scenario file..."
source $val(sc)
# Define traffic

#puts "Loading traffic file..."
source $val(cp)

# Define node initial position in nam
for {set i 0} {$i < $val(nn)} {incr i} {
# 30 defines the node size for nam
$ns initial_node_pos $node_($i) 30
}

# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at $val(stop) "$node_($i) reset";
}
# ending nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 40.01 "puts \"end simulation\" ; $ns halt"
proc stop {} {
global ns tracefd namtrace
$ns flush-trace
close $tracefd
close $namtrace
exec nam simwrls.nam &
exit 0

```



Los Libertadores
Fundación Universitaria

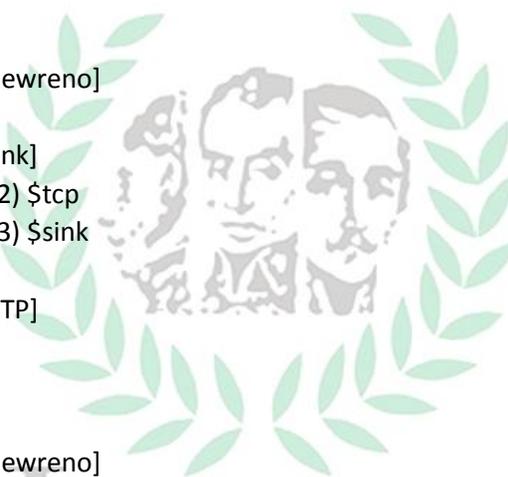
ANEXO B.CÓDIGO FUENTE PARA DESCRIPCIÓN DEL TRÁFICO USADO

```
# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"
```

```
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(2) $tcp
$ns attach-agent $node_(3) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"
```

```
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(4) $tcp
$ns attach-agent $node_(5) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"
```

```
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(6) $tcp
$ns attach-agent $node_(7) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"
```



Los Libertadores
Fundación Universitaria

```
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(8) $tcp
$ns attach-agent $node_(9) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"
```

```
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(10) $tcp
$ns attach-agent $node_(11) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"
```

```
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(12) $tcp
$ns attach-agent $node_(13) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"
```

```
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(14) $tcp
$ns attach-agent $node_(15) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"
```



```
set tcp [new Agent/TCP/Newreno]
$tcp set class_2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(16) $tcp
$ns attach-agent $node_(17) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"
```

```
set tcp [new Agent/TCP/Newreno]
$tcp set class_2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(18) $tcp
$ns attach-agent $node_(19) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 5.0 "$ftp start"
```



Los Libertadores
Fundación Universitaria

ANEXO C.CÓDIGOS DE FILTROS AWK PARA SELECCIÓN DE DATOS

FILTRO AWK PARA DELAY

```
BEGIN {
highest_packet_id = 0;
}
{
action = $1;
time = $2;
level = $4;
packet_id = $6;
if ((action == "s") && (level == "AGT"))
{
start_time[packet_id] = time;
if ( packet_id > highest_packet_id )
highest_packet_id = packet_id;
}
if ((action == "r") && (level == "AGT"))
{
end_time[packet_id] = time;
}
if ( ($1 == "d") && (level == "AGT"))
{
drop_packet++;
end_time[packet_id] = -1;
}
}
END {
for ( packet_id = 0; packet_id <= highest_packet_id; packet_id++ )
{
start = start_time[packet_id];
end = end_time[packet_id];
if (start < end)
{
packet_duration = end - start;
duration_total += packet_duration;
packet_number++;
}
}
delay = duration_total / packet_number;
printf("%f \n", delay);}
```

FILTRO AWK PARA JITTER

```
BEGIN {
    num_rcv=0
}

{

    event = $1
    time = $2
    node_id = $3
    flow_id = $10
    pkt_id = $6
    pkt_size = $8
    flow_t = $45
    level = $4

    # Store packets send time
    if (level == "AGT" && sendTime[pkt_id] == 0 && (event == "+" || event == "s") && pkt_size
    >= 512) {
        sendTime[pkt_id] = time
    }

    # Store packets arrival time
    if (level == "AGT" && event == "r" && pkt_size >= 512) {
        rcvTime[pkt_id] = time
        num_rcv++
    }
}

END {
    # Compute average jitter
    jitter1 = jitter2 = tmp_rcv = 0
    prev_time = delay = prev_delay = processed = 0
    prev_delay = -1
    for (i=0; processed<num_rcv; i++) {
        if(rcvTime[i] != 0) {
            tmp_rcv++
            if(prev_time != 0) {
                delay = rcvTime[i] - prev_time
                e2eDelay = rcvTime[i] - sendTime[i]
            }
        }
    }
}
```

```

if(delay < 0) delay = 0
if(prev_delay != -1) {
jitter1 += abs(e2eDelay - prev_e2eDelay)
jitter2 += abs(delay-prev_delay)
}
prev_delay = delay
prev_e2eDelay = e2eDelay
}
prev_time = recvTime[i]
}
processed++
}
}

END {
printf("Jitter1 = %.2f\n",jitter1*1000/tmp_rcv);
printf("Jitter2 = %.2f\n",jitter2*1000/tmp_rcv);
}

function abs(value) {
if (value < 0) value = 0-value
return value
}

```

FILTROS AWK PARA PDR

```

BEGIN {
    highest_packet_id = 0;
    packet_send = 0;
    packet_received = 0;
}

{
    action = $1;
    time = $2;
    level = $4;
    packet_id = $6;
    if ((action == "s") && (level == "AGT"))
    {
        start_time[packet_id] = time;
        if ( packet_id > highest_packet_id )
            highest_packet_id = packet_id;
    }
}

```

```

}
if ((action == "r") && (level == "AGT"))
{
    end_time[packet_id] = time;
    packet_received++;
}
if(time >= 1500)
    start_time[packet_id] = 0;
}
END {
for ( packet_id = 0; packet_id <= highest_packet_id; packet_id++ )
{
    if(start_time[packet_id] != 0)
        packet_send++;
}
ratio = packet_received / packet_send ;
printf("%f \n", ratio);
}

```

FILTRO AWK PARA THROUGHPUT

```

BEGIN {
    recvdSize = 0
    startTime = 1e6
    stopTime = 0
}
{
    event = $1
    time = $2
    node_id = $5
    flow_id = $10
    pkt_id = $6
    pkt_size = $8
    flow_t = $45
    level = $4

    # Store start time
    if (level == "AGT" && (event == "+" || event == "s") && pkt_size >= 512) {
        if (time < startTime) {

```


ANEXO D. CÓDIGOS PARA GENERACIÓN DE MOVIMIENTOS DE NODOS EN JAVA

CÓDIGO PRINCIPAL PARA GENERACIÓN DE ESCENARIOS

```
publicclass PrincipalGirasol {  
  
    publicstaticvoid main(String[] args) {  
        // TODO Auto-generated method stub  
        final String nombresim = "$node_"; //Prefijo en el simulador  
        final int num = 20; // NumerodeNodos  
        final int areax = 200; //Longitudde X (Informaciondel Area  
deSimulación)  
        final int areay = 200; //Longitudde Y (Informaciondel Area  
deSimulación)  
  
        final double velocidad = 1; //Velocidaddelosnodosen m/s  
        final int tiempo = 100; //Tiempodelasimulacion  
        final int radio = 3;  
        AdminFiles.PrintDato("# Posicion Inicial\n");  
        EjemploGirasol.PosicionInicial(nombresim, num, areax, areay,  
radio);  
        EjemploGirasol.ModeloUniformeSinPausas(nombresim, num, areax,  
areay,radio, velocidad, tiempo);  
        System.out.println("Fin del Proceso!");  
    }  
}
```

CÓDIGO PARA GENERACIÓN DE MOVIMIENTO DE ESTRELLA

```
package ejemplo;  
  
import java.util.Random;  
  
publicclass EjemploUniforme {  
  
    publicstaticdouble[][] nodo;  
  
    /**  
    * Creanodosdeformaaleatoria.  
    * EjemUniforme.CrearNodos("$node_", No. nodos, long en X, long en Y);  
    * Ejemplo: EjemUniforme.CrearNodos("$node_", 10, 200, 100);  
    * @param name Nombrecomosedeclearoen el archivodesimulación  
    * @param n Numerodenodos  
    * @param x Longitudde X (Informaciondel Area deSimulación)  
    * @param y Longitudde Y (Informaciondel Area deSimulación)  
    */  
    publicstaticvoid PosicionInicial(String name,intn, intx, inty, intl){
```

```

        nodo = newdouble[n][10000];
        Random rnd = newRandom();
        for (inti = 0; i<n; i++) {
            nodo[i][0] = (4*1)+((rnd.nextDouble())*(x-(8*1)));//Pos 0.
Sealmacenapositioninicialen x (x1)
            nodo[i][1] = (4*1)+((rnd.nextDouble())*(y-(8*1)));//Pos 1.
Sealmacenapositioninicialen y (y1)
            AdminFiles.PrintDato(name+"("+i+" set X_ "+nodo[i][0]+"\\n");
            AdminFiles.PrintDato(name+"("+i+" set Y_ "+nodo[i][1]+"\\n");
            AdminFiles.PrintDato(name+"("+i+" set Z_ 0.0\\n");
            nodo[i][24]=0;//t0
        }
    }

/**
 * MovimientoconVelocidadConstante sin Pausa.
 * @param name Nombrecomosedeclaroen el archivodesimulacion
 * @param n Numerodenodos
 * @param x Longitudde X (Informaciondel Area deSimulacion)
 * @param y Longitudde Y (Informaciondel Area deSimulacion)
 * @param dx Longituden X delmovimiento
 * @param dy Longituden Y delmovimiento
 * @param v Velocidaddelosnodosen m/s
 * @param t Tiempodelasimulacion
 */
publicstaticvoid ModeloUniformeSinPausas(String name,intn, intx, inty, intl,
doublev, doublet){
    doubleang,tiempo,rad,b;
    booleanestado = false;
    do {
        /**GeneraciondePuntos**/
        for (inti = 0; i<n; i++) {
            rad=0.0174532925;
            ang=Math.random()*360;
            b=ang*rad;
nodo[i][2] = nodo[i][0]+(1*Math.cos(b));//(x2)
            nodo[i][3] = nodo[i][1]+(1*Math.sin(b));//(y2)

nodo[i][4] = nodo[i][0]+(1.73*1*Math.cos((30*rad)+b));//(x3)
            nodo[i][5] = nodo[i][1]+(1.73*1*Math.sin((30*rad)+b));//(y3)

nodo[i][6] = nodo[i][0]+(2*1*Math.cos(b));//(x4)
            nodo[i][7] = nodo[i][1]+(2*1*Math.sin(b));//(y4)

nodo[i][8] = nodo[i][0]+(3*1*Math.cos(b));//(x5)
            nodo[i][9] = nodo[i][1]+(3*1*Math.sin(b));//(y5)

nodo[i][10]= nodo[i][0]+(2.64575*1*Math.cos((-19.1*rad)+b));//(x6)
            nodo[i][11]= nodo[i][1]+(2.64575*1*Math.sin((-19.1*rad)+b));//(y6)

nodo[i][12]= nodo[i][0]+(3.464*1*Math.cos((-30*rad)+b));//(x7)
            nodo[i][13]= nodo[i][1]+(3.464*1*Math.sin((-30*rad)+b));//(y7)

```

```

nodo[i][14]= nodo[i][0]+(2.64575*1*Math.cos((-40.89*rad)+b));//(x8)
nodo[i][15]= nodo[i][1]+(2.64575*1*Math.sin((-40.89*rad)+b));//(y8)

nodo[i][16]= nodo[i][0]+(3*1*Math.cos((-60*rad)+b));//(x9)
nodo[i][17]= nodo[i][1]+(3*1*Math.sin((-60*rad)+b));//(y9)

nodo[i][18]= nodo[i][0]+(2*1*Math.cos((-60*rad)+b));//(x10)
nodo[i][19]= nodo[i][1]+(2*1*Math.sin((-60*rad)+b));//(y10)

nodo[i][20]= nodo[i][0]+(1.732*1*Math.cos((-90*rad)+b));//(x11)
nodo[i][21]= nodo[i][1]+(1.732*1*Math.sin((-90*rad)+b));//(y11)

nodo[i][22]= nodo[i][0]+(1*Math.cos((-60*rad)+b));//(x12)
nodo[i][23]= nodo[i][1]+(1*Math.sin((-60*rad)+b));//(y12)

}

/**CalculodeTiemposenfunciondelaVelocidad**/
for (inti = 0; i<n; i++) {
    AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][2]+" "+nodo[i][3]+" "+v+"\n\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][2]-nodo[i][0],
2)+Math.pow(nodo[i][3]-nodo[i][1], 2))/v;
    nodo[i][24]=nodo[i][24]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][24], t);

    AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][4]+" "+nodo[i][5]+" "+v+"\n\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][4]-nodo[i][2],
2)+Math.pow(nodo[i][5]-nodo[i][3], 2))/v;
    nodo[i][24]=nodo[i][24]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][24], t);

    AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][6]+" "+nodo[i][7]+" "+v+"\n\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][6]-nodo[i][4],
2)+Math.pow(nodo[i][7]-nodo[i][5], 2))/v;
    nodo[i][24]=nodo[i][24]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][24], t);

    AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][8]+" "+nodo[i][9]+" "+v+"\n\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][8]-nodo[i][6],
2)+Math.pow(nodo[i][9]-nodo[i][7], 2))/v;
    nodo[i][24]=nodo[i][24]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][24], t);

    AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][10]+" "+nodo[i][11]+" "+v+"\n\n");

```

```

        tiempo=Math.sqrt(Math.pow(nodo[i][10]-nodo[i][8],
2)+Math.pow(nodo[i][11]-nodo[i][9], 2))/v;
        nodo[i][24]=nodo[i][24]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][24], t);

        AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][12]+" "+nodo[i][13]+" "+v+"\n\n");
        tiempo=Math.sqrt(Math.pow(nodo[i][12]-nodo[i][10],
2)+Math.pow(nodo[i][13]-nodo[i][11], 2))/v;
        nodo[i][24]=nodo[i][24]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][24], t);

        AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][14]+" "+nodo[i][15]+" "+v+"\n\n");
        tiempo=Math.sqrt(Math.pow(nodo[i][14]-nodo[i][12],
2)+Math.pow(nodo[i][15]-nodo[i][13], 2))/v;
        nodo[i][24]=nodo[i][24]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][24], t);

        AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][16]+" "+nodo[i][17]+" "+v+"\n\n");
        tiempo=Math.sqrt(Math.pow(nodo[i][16]-nodo[i][14],
2)+Math.pow(nodo[i][17]-nodo[i][15], 2))/v;
        nodo[i][24]=nodo[i][24]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][24], t);

        AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][18]+" "+nodo[i][19]+" "+v+"\n\n");
        tiempo=Math.sqrt(Math.pow(nodo[i][18]-nodo[i][16],
2)+Math.pow(nodo[i][19]-nodo[i][17], 2))/v;
        nodo[i][24]=nodo[i][24]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][24], t);

        AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][20]+" "+nodo[i][21]+" "+v+"\n\n");
        tiempo=Math.sqrt(Math.pow(nodo[i][20]-nodo[i][18],
2)+Math.pow(nodo[i][21]-nodo[i][19], 2))/v;
        nodo[i][24]=nodo[i][24]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][24], t);

        AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][22]+" "+nodo[i][23]+" "+v+"\n\n");
        tiempo=Math.sqrt(Math.pow(nodo[i][22]-nodo[i][20],
2)+Math.pow(nodo[i][23]-nodo[i][21], 2))/v;
        nodo[i][24]=nodo[i][24]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][24], t);

        AdminFiles.PrintDato("$ns at "+nodo[i][24]+" \""+name+"("+i+")
setdest "+nodo[i][0]+" "+nodo[i][1]+" "+v+"\n\n");
        tiempo=Math.sqrt(Math.pow(nodo[i][0]-nodo[i][22],
2)+Math.pow(nodo[i][1]-nodo[i][23], 2))/v;
        nodo[i][24]=nodo[i][24]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][24], t);

```

```

    }

    } while (!estado);
}

private static boolean TestTiempoSimulacion(double tnodo, double tsimulacion) {
    boolean test=false;
    if(tnodo>tsimulacion){
        test=true;
    }
    return test;
}
}

```

CÓDIGO PARA GENERACIÓN DE MOVIMIENTO DE GIRASOL

```

import java.util.Random;

//import java.util.Random;

public class EjemploGirasol {

    public static double[][] nodo;

    /**
     * Creanodosdeformaaleatoria.
     * EjemUniforme.CrearNodos("$node_", No. nodos, long en X, long en Y);
     * Ejemplo: EjemUniforme.CrearNodos("$node_", 10, 200, 100);
     * @param name Nombrecomosedeclaroen el archivodesimulación
     * @param n Numerodenodos
     * @param x Longitudde X (Informaciondel Area deSimulación)
     * @param y Longitudde Y (Informaciondel Area deSimulación)
     */
    public static void PosicionInicial(String name, int n, int x, int y, int r){
        nodo = new double[n][10000];

        Random rnd = new Random();
        for (inti = 0; i<n; i++) {
            nodo[i][0] = (2.5*r)+((rnd.nextDouble())*(x-(5*r))); //Pos 0.
            Sealmacenaposicioninicialen x (x1)
            nodo[i][1] = (2.5*r)+((rnd.nextDouble())*(y-(5*r))); //Pos 1.
            Sealmacenaposicioninicialen y (y1)
            AdminFiles.PrintDato(name+"("+i+") set X_ "+nodo[i][0]+"\\n");
            AdminFiles.PrintDato(name+"("+i+") set Y_ "+nodo[i][1]+"\\n");
            AdminFiles.PrintDato(name+"("+i+") set Z_ 0.0\\n");
            nodo[i][32]=0; //t0
        }
    }
}

```

```

/**
 * Movimiento con Velocidad Constante sin Pausa.
 * @param name Nombre como se declare en el archivo de simulación
 * @param n Número de nodos
 * @param x Longitud de X (Información del Área de Simulación)
 * @param y Longitud de Y (Información del Área de Simulación)
 * @param dx Longitud de X del movimiento
 * @param dy Longitud de Y del movimiento
 * @param v Velocidad de los nodos en m/s
 * @param t Tiempo de la simulación
 */
public static void ModeloUniformeSinPausas(String name, int n, int x, int y, int r,
double v, double t){
    double ang, tiempo, rad, b;
    boolean estado = false;
    do {
        /** Generación de Puntos */
        for (int i = 0; i < n; i++) {
            rad = 0.0174532925;
            ang = Math.random()*360;
            b = ang*rad;

            nodo[i][2] = nodo[i][0] + (0.571*r*Math.cos((19.57*rad)+b)); // (x2)
            nodo[i][3] = nodo[i][1] + (0.571*r*Math.sin((19.57*rad)+b)); // (y2)

            nodo[i][4] = nodo[i][0] + (0.7653*r*Math.cos((67.5*rad)+b)); // (x3)
            nodo[i][5] = nodo[i][1] + (0.7653*r*Math.sin((67.5*rad)+b)); // (y3)
            // System.out.println(R);
            nodo[i][6] = nodo[i][0] + (0.9312*r*Math.cos((29.7*rad)+b)); // (x4)
            nodo[i][7] = nodo[i][1] + (0.9312*r*Math.sin((29.7*rad)+b)); // (y4)

            nodo[i][8] = nodo[i][0] + (1.4142*r*Math.cos((45*rad)+b)); // (x5)
            nodo[i][9] = nodo[i][1] + (1.4142*r*Math.sin((45*rad)+b)); // (y5)

            nodo[i][10] = nodo[i][0] + (1.2777*r*Math.cos((21.19*rad)+b)); // (x6)
            nodo[i][11] = nodo[i][1] + (1.2777*r*Math.sin((21.19*rad)+b)); // (y6)

            nodo[i][12] = nodo[i][0] + (1.8477*r*Math.cos((22.5*rad)+b)); // (x7)
            nodo[i][13] = nodo[i][1] + (1.8477*r*Math.sin((22.5*rad)+b)); // (y7)

            nodo[i][14] = nodo[i][0] + (1.4744*r*Math.cos((7.45*rad)+b)); // (x8)
            nodo[i][15] = nodo[i][1] + (1.4744*r*Math.sin((7.45*rad)+b)); // (y8)

            nodo[i][16] = nodo[i][0] + (2*r*Math.cos((0*rad)+b)); // (x9)
            nodo[i][17] = nodo[i][1] + (2*r*Math.sin((0*rad)+b)); // (y9)

            nodo[i][18] = nodo[i][0] + (1.4744*r*Math.cos((-7.45*rad)+b)); // (x10)
            nodo[i][19] = nodo[i][1] + (1.4744*r*Math.sin((-7.45*rad)+b)); // (y10)

            nodo[i][20] = nodo[i][0] + (1.8477*r*Math.cos((-22.5*rad)+b)); // (x11)
            nodo[i][21] = nodo[i][1] + (1.8477*r*Math.sin((-22.5*rad)+b)); // (y11)

            nodo[i][22] = nodo[i][0] + (1.2777*r*Math.cos((-21.19*rad)+b)); // (x12)

```

```

nodo[i][23]= nodo[i][1]+(1.2777*r*Math.sin((-21.19*rad)+b));//(y12)

nodo[i][24]= nodo[i][0]+(1.4142*r*Math.cos((-45*rad)+b));//(x13)
nodo[i][25]= nodo[i][1]+(1.4142*r*Math.sin((-45*rad)+b));//(y13)

nodo[i][26]= nodo[i][0]+(0.9312*r*Math.cos((-29.7*rad)+b));//(x14)
nodo[i][27]= nodo[i][1]+(0.9312*r*Math.sin((-29.7*rad)+b));//(y14)

nodo[i][28]= nodo[i][0]+(0.7653*r*Math.cos((-67.5*rad)+b));//(x15)
nodo[i][29]= nodo[i][1]+(0.7653*r*Math.sin((-67.5*rad)+b));//(y15)

nodo[i][30]= nodo[i][0]+(0.571*r*Math.cos((-19.57*rad)+b));//(x16)
nodo[i][31]= nodo[i][1]+(0.571*r*Math.sin((-19.57*rad)+b));//(y16)

}
/**CalculodeTiemposenfunciondelavelocidad**/
for (inti = 0; i<n; i++) {
AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+") setdest
"+nodo[i][2]+" "+nodo[i][3]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][2]-nodo[i][0],
2)+Math.pow(nodo[i][3]-nodo[i][1], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][4]+" "+nodo[i][5]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][4]-nodo[i][2],
2)+Math.pow(nodo[i][5]-nodo[i][3], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][6]+" "+nodo[i][7]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][6]-nodo[i][4],
2)+Math.pow(nodo[i][7]-nodo[i][5], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][8]+" "+nodo[i][9]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][8]-nodo[i][6],
2)+Math.pow(nodo[i][9]-nodo[i][7], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][10]+" "+nodo[i][11]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][10]-nodo[i][8],
2)+Math.pow(nodo[i][11]-nodo[i][9], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

```

```

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][12]+" "+nodo[i][13]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][12]-nodo[i][10],
2)+Math.pow(nodo[i][13]-nodo[i][11], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

```

```

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][14]+" "+nodo[i][15]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][14]-nodo[i][12],
2)+Math.pow(nodo[i][15]-nodo[i][13], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

```

```

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][16]+" "+nodo[i][17]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][16]-nodo[i][14],
2)+Math.pow(nodo[i][17]-nodo[i][15], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

```

```

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][18]+" "+nodo[i][19]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][18]-nodo[i][16],
2)+Math.pow(nodo[i][19]-nodo[i][17], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

```

```

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][20]+" "+nodo[i][21]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][20]-nodo[i][18],
2)+Math.pow(nodo[i][21]-nodo[i][19], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

```

```

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][22]+" "+nodo[i][23]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][22]-nodo[i][20],
2)+Math.pow(nodo[i][23]-nodo[i][21], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

```

```

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][24]+" "+nodo[i][25]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][24]-nodo[i][22],
2)+Math.pow(nodo[i][25]-nodo[i][23], 2))/v;
nodo[i][32]=nodo[i][32]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][32], t);

```

```

AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][26]+" "+nodo[i][27]+" "+v+"\n\n");

```

```

        tiempo=Math.sqrt(Math.pow(nodo[i][26]-nodo[i][24],
2)+Math.pow(nodo[i][27]-nodo[i][25], 2))/v;
        nodo[i][32]=nodo[i][32]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][32], t);

        AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][28]+" "+nodo[i][29]+" "+v+"\n\n");
        tiempo=Math.sqrt(Math.pow(nodo[i][28]-nodo[i][26],
2)+Math.pow(nodo[i][29]-nodo[i][27], 2))/v;
        nodo[i][32]=nodo[i][32]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][32], t);

        AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][30]+" "+nodo[i][31]+" "+v+"\n\n");
        tiempo=Math.sqrt(Math.pow(nodo[i][30]-nodo[i][28],
2)+Math.pow(nodo[i][31]-nodo[i][29], 2))/v;
        nodo[i][32]=nodo[i][32]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][32], t);

        AdminFiles.PrintDato("$ns at "+nodo[i][32]+" \""+name+"("+i+")
setdest "+nodo[i][0]+" "+nodo[i][1]+" "+v+"\n\n");
        tiempo=Math.sqrt(Math.pow(nodo[i][0]-nodo[i][30],
2)+Math.pow(nodo[i][1]-nodo[i][31], 2))/v;
        nodo[i][32]=nodo[i][32]+tiempo;//tiempo
        estado = TestTiempoSimulacion(nodo[i][32], t);
    }
} while (!estado);
}
private static boolean TestTiempoSimulacion(double tnodo, double tsimulacion) {
    boolean test=false;
    if(tnodo>tsimulacion){
        test=true;
    }
    return test;
}
}

```

CÓDIGO PARA GENERACIÓN DE MOVIMIENTO DE OCHO

```

import java.util.Random;

public class EjemploOcho {

    public static double[][] nodo;

```

```

    /**
    * Creanodosdeformaaleaoria.
    * EjemUniforme.CrearNodos("$node_", No. nodos, long en X, long en Y);
    * Ejemplo: EjemUniforme.CrearNodos("$node_", 10, 200, 100);
    * @param name Nombrecomosedeclearoen el archivodesimulación
    * @param n Numerodenodos
    * @param x Longitudde X (Informaciondel Area deSimulación)
    * @param y Longitudde Y (Informaciondel Area deSimulación)
    */
    publicstaticvoid PosicionInicial(String name,intn, intx, inty, intr){
        nodo = newdouble[n][10000];
        Random rnd = newRandom();

        for (inti = 0; i<n; i++) {
            nodo[i][0] = (4*r+1)+((rnd.nextDouble()*(x-((8*r)-1))));//Pos
0. Sealmacenaposicioninicialen x (x1)

            nodo[i][1] = (4*r+1)+((rnd.nextDouble()*(y-((8*r)-1))));//Pos
1. Sealmacenaposicioninicialen y (y1)

            AdminFiles.PrintDato(name+"("+i+") set X_ "+nodo[i][0]+"\\n");
            AdminFiles.PrintDato(name+"("+i+") set Y_ "+nodo[i][1]+"\\n");
            AdminFiles.PrintDato(name+"("+i+") set Z_ 0.0\\n");
            nodo[i][48]=0;//t0
        }
    }

    /**
    * MovimientoconVelocidadConstante sin Pausa.
    * @param name Nombrecomosedeclearoen el archivodesimulación
    * @param n Numerodenodos
    * @param x Longitudde X (Informaciondel Area deSimulación)
    * @param y Longitudde Y (Informaciondel Area deSimulación)
    * @param dx Longituden X delmovimiento
    * @param dy Longituden Y delmovimiento
    * @param v Velocidaddelosnodosen m/s
    * @param t Tiempodelasimulacion
    */
    publicstaticvoid ModeloUniformeSinPausas(String name,intn, intx, inty, intr,
doublev, doublet){

        doublea;
        doubleb;
        doublec;

        a=0.0174532925;

        doubletiempo;
        booleanestado = false;

```

```

do {
  /**GeneraciondePuntos**/
  for (inti = 0; i<n; i++) {

      c=Math.random()*360;
      b=c*a;

nodo[i][2] = nodo[i][0]+(0.8164*r*Math.cos((65.9*a)+b));//(x2)
nodo[i][3] = nodo[i][1]+(0.8164*r*Math.sin((65.9*a)+b));//(y2)

nodo[i][4] = nodo[i][0]+(1.154693*r*Math.cos((54.73*a)+b));//(x3)
nodo[i][5] = nodo[i][1]+(1.154693*r*Math.sin((54.73*a)+b));//(y3)

nodo[i][6] = nodo[i][0]+(1.414213*r*Math.cos((45*a)+b));//(x4)
nodo[i][7] = nodo[i][1]+(1.414213*r*Math.sin((45*a)+b));//(y4)

nodo[i][8] = nodo[i][0]+(1.632987*r*Math.cos((35.26*a)+b));//(x5)
nodo[i][9] = nodo[i][1]+(1.632987*r*Math.sin((35.26*a)+b));//(y5)

nodo[i][10]= nodo[i][0]+(1.825719*r*Math.cos((24.09*a)+b));//(x6)
nodo[i][11]= nodo[i][1]+(1.825719*r*Math.sin((24.09*a)+b));//(y6)

nodo[i][12]= nodo[i][0]+(2*r*Math.cos(b));//(x7)
nodo[i][13]= nodo[i][1]+(2*r*Math.sin(b));//(y7)

////////////////////////////////////
nodo[i][14] = nodo[i][0]+(2.449472*r*Math.cos((-17.71*a)+b));//(x8)
nodo[i][15] = nodo[i][1]+(2.449472*r*Math.sin((-17.71*a)+b));//(y8)

nodo[i][16] = nodo[i][0]+(2.828424*r*Math.cos((-19.47*a)+b));//(x9)
nodo[i][17] = nodo[i][1]+(2.828424*r*Math.sin((-19.47*a)+b));//(y9)

nodo[i][18] = nodo[i][0]+(3.162277*r*Math.cos((-18.43*a)+b));//(x10)
nodo[i][19] = nodo[i][1]+(3.162277*r*Math.sin((-18.43*a)+b));//(y10)

nodo[i][20] = nodo[i][0]+(3.464099*r*Math.cos((-15.79*a)+b));//(x11)
nodo[i][21] = nodo[i][1]+(3.464099*r*Math.sin((-15.79*a)+b));//(y11)

nodo[i][22] = nodo[i][0]+(3.741646*r*Math.cos((-11.48*a)+b));//(x12)
nodo[i][23] = nodo[i][1]+(3.741646*r*Math.sin((-11.48*a)+b));//(y12)

nodo[i][24] = nodo[i][0]+(4*r*Math.cos(+b));//(x13)
nodo[i][25] = nodo[i][1]+(4*r*Math.sin(+b));//(y13)

////////////////////////////////////
nodo[i][26] = nodo[i][0]+(3.741646*r*Math.cos((11.48*a)+b));//(x14)
nodo[i][27] = nodo[i][1]+(3.741646*r*Math.sin((11.48*a)+b));//(y14)

nodo[i][28] = nodo[i][0]+(3.464099*r*Math.cos((15.79*a)+b));//(x15)
nodo[i][29] = nodo[i][1]+(3.464099*r*Math.sin((15.79*a)+b));//(y15)

nodo[i][30] = nodo[i][0]+(3.162277*r*Math.cos((18.43*a)+b));//(x16)

```

```

nodo[i][31] = nodo[i][1]+(3.162277*r*Math.sin((18.43*a)+b));//(y16)

nodo[i][32] = nodo[i][0]+(2.828424*r*Math.cos((19.47*a)+b));//(x17)
nodo[i][33] = nodo[i][1]+(2.828424*r*Math.sin((19.47*a)+b));//(y17)

nodo[i][34] = nodo[i][0]+(2.449472*r*Math.cos((17.71*a)+b));//(x18)
nodo[i][35] = nodo[i][1]+(2.449472*r*Math.sin((17.71*a)+b));//(y18)

nodo[i][36] = nodo[i][0]+(2*r*Math.cos(b));//(x19)
nodo[i][37] = nodo[i][1]+(2*r*Math.sin(b));//(y19)
////////////////////////////////////

nodo[i][38] = nodo[i][0]+(1.825719*r*Math.cos((-24.09*a)+b));//(x20)
nodo[i][39] = nodo[i][1]+(1.825719*r*Math.sin((-24.09*a)+b));//(y20)

nodo[i][40] = nodo[i][0]+(1.632987*r*Math.cos((-35.26*a)+b));//(x21)
nodo[i][41] = nodo[i][1]+(1.632987*r*Math.sin((-35.26*a)+b));//(y21)

nodo[i][42] = nodo[i][0]+(1.414213*r*Math.cos((-45*a)+b));//(x22)
nodo[i][43] = nodo[i][1]+(1.414213*r*Math.sin((-45*a)+b));//(y22)

nodo[i][44] = nodo[i][0]+(1.154693*r*Math.cos((-54.73*a)+b));//(x23)
nodo[i][45] = nodo[i][1]+(1.154693*r*Math.sin((-54.73*a)+b));//(y23)

nodo[i][46] = nodo[i][0]+(0.8164*r*Math.cos((-65.9*a)+b));//(x24)
nodo[i][47] = nodo[i][1]+(0.8164*r*Math.sin((-65.9*a)+b));//(y24)

}

/**CalculodeTiemposenfunciodelaVelocidad**/
for (inti = 0; i<n; i++) {
AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][2]+" "+nodo[i][3]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][2]-nodo[i][0],
2)+Math.pow(nodo[i][3]-nodo[i][1], 2))/v;
nodo[i][48]=nodo[i][48]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][4]+" "+nodo[i][5]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][4]-nodo[i][2],
2)+Math.pow(nodo[i][5]-nodo[i][3], 2))/v;
nodo[i][48]=nodo[i][48]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][6]+" "+nodo[i][7]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][6]-nodo[i][4],
2)+Math.pow(nodo[i][7]-nodo[i][5], 2))/v;
nodo[i][48]=nodo[i][48]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][48], t);

```

```

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][8]+" "+nodo[i][9]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][8]-nodo[i][6],
2)+Math.pow(nodo[i][9]-nodo[i][7], 2))/v;
nodo[i][48]=nodo[i][48]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][10]+" "+nodo[i][11]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][10]-nodo[i][8],
2)+Math.pow(nodo[i][11]-nodo[i][9], 2))/v;
nodo[i][48]=nodo[i][48]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][12]+" "+nodo[i][13]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][12]-nodo[i][10],
2)+Math.pow(nodo[i][13]-nodo[i][11], 2))/v;
nodo[i][48]=nodo[i][48]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][14]+" "+nodo[i][15]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][14]-nodo[i][12],
2)+Math.pow(nodo[i][15]-nodo[i][13], 2))/v;
nodo[i][48]=nodo[i][48]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][16]+" "+nodo[i][17]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][16]-nodo[i][14],
2)+Math.pow(nodo[i][17]-nodo[i][15], 2))/v;
nodo[i][48]=nodo[i][48]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][18]+" "+nodo[i][19]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][18]-nodo[i][16],
2)+Math.pow(nodo[i][19]-nodo[i][17], 2))/v;
nodo[i][48]=nodo[i][48]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][20]+" "+nodo[i][21]+" "+v+"\n\n");
tiempo=Math.sqrt(Math.pow(nodo[i][20]-nodo[i][18],
2)+Math.pow(nodo[i][21]-nodo[i][19], 2))/v;
nodo[i][48]=nodo[i][48]+tiempo;//tiempo
estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][22]+" "+nodo[i][23]+" "+v+"\n\n");

```

```

    tiempo=Math.sqrt(Math.pow(nodo[i][22]-nodo[i][20],
2)+Math.pow(nodo[i][23]-nodo[i][21], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

    AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+")
setdest "+nodo[i][24]+" "+nodo[i][25]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][24]-nodo[i][22],
2)+Math.pow(nodo[i][25]-nodo[i][23], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+") setdest
"+nodo[i][26]+" "+nodo[i][27]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][26]-nodo[i][24],
2)+Math.pow(nodo[i][27]-nodo[i][25], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+") setdest
"+nodo[i][28]+" "+nodo[i][29]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][28]-nodo[i][26],
2)+Math.pow(nodo[i][29]-nodo[i][27], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+") setdest
"+nodo[i][30]+" "+nodo[i][31]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][30]-nodo[i][28],
2)+Math.pow(nodo[i][31]-nodo[i][29], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+") setdest
"+nodo[i][32]+" "+nodo[i][33]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][32]-nodo[i][30],
2)+Math.pow(nodo[i][33]-nodo[i][31], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+") setdest
"+nodo[i][34]+" "+nodo[i][35]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][34]-nodo[i][32],
2)+Math.pow(nodo[i][35]-nodo[i][33], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \""+name+"("+i+") setdest
"+nodo[i][36]+" "+nodo[i][37]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][36]-nodo[i][34],
2)+Math.pow(nodo[i][37]-nodo[i][35], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

```

```

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \"+name+"("+i+") setdest
"+nodo[i][38]+" "+nodo[i][39]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][38]-nodo[i][36],
2)+Math.pow(nodo[i][39]-nodo[i][37], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \"+name+"("+i+") setdest
"+nodo[i][40]+" "+nodo[i][41]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][40]-nodo[i][38],
2)+Math.pow(nodo[i][41]-nodo[i][39], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \"+name+"("+i+") setdest
"+nodo[i][42]+" "+nodo[i][43]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][42]-nodo[i][40],
2)+Math.pow(nodo[i][43]-nodo[i][41], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \"+name+"("+i+") setdest
"+nodo[i][44]+" "+nodo[i][45]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][44]-nodo[i][42],
2)+Math.pow(nodo[i][45]-nodo[i][43], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \"+name+"("+i+") setdest
"+nodo[i][46]+" "+nodo[i][47]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][46]-nodo[i][44],
2)+Math.pow(nodo[i][47]-nodo[i][45], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);

    AdminFiles.PrintDato("$ns at "+nodo[i][48]+" \"+name+"("+i+")
setdest "+nodo[i][0]+" "+nodo[i][1]+" "+v+"\n");
    tiempo=Math.sqrt(Math.pow(nodo[i][0]-nodo[i][46],
2)+Math.pow(nodo[i][1]-nodo[i][47], 2))/v;
    nodo[i][48]=nodo[i][48]+tiempo;//tiempo
    estado = TestTiempoSimulacion(nodo[i][48], t);
}
} while (!estado);
}

private static int Fraction(int i, int j) {
    // TODO Auto-generated method stub
    return 0;
}

private static boolean TestTiempoSimulacion(double t nodo, double t simulacion)
{

```

```
    boolean test=false;
    if(tnodo>tsimulacion){
        test=true;
    }
    return test;
}
```



Los Libertadores
Fundación Universitaria

Anexo

Artículos desarrollados en el
Semillero de Investigación en
Señales y Sistemas

Los Libertadores
Fundación Universitaria

Traffic Analysis AODV and AOMDV for data transmission in SLAM

J. Rodríguez, H. Gómez and J. Anzola

Abstract— This article analyzes data traffic occurs between the protocols AODV and AOMDV transmitting information space recognition, location and simultaneous mapping using UDP. This task is performed by node 20 moving at a constant speed, but distributed entirely normal. The trajectories and displacement of nodes is in the form of closed star to get 360 degrees of information, either view in or out, presenting the end traffic analysis UDP data, evaluating metrics delay, jitter, the performance and RDP. These results allow to estimate that protocol is most appropriate to subsequently deploy it works SLAM.

Keywords— wireless communications, modeling, simulation, neural networks, software tools.

I. INTRODUCCION

UN aspecto tecnológico presente en los sistemas de reconocimiento de terrenos y espacios es el problema de la localización y mapeado simultáneos, el cual permite descubrir si es posible para un robot móvil navegar a través de un entorno desconocido y construir, de manera incremental, un mapa consistente del mismo, mientras que se determina al mismo tiempo su posición dentro de este mapa.

Una de las principales aplicaciones de estudio y desarrollo corresponden a las redes autoorganizadas, las cuales mediante el estándar 802.11x, se plantea como aplicación el suministro cooperativo de información de procesamiento en cuanto a localización y mapeado, transmitiendo esta información sobre tráfico UDP a los nodos vecinos en la red [1].

En este artículo se analiza el uso de trayectorias cerradas en cuanto a movimiento, con referencia al Delay, Jitter, PDR y el Throughput, dado que para los problemas de localización y mapeado simultáneo, se propone el estudio del comportamiento de tráfico UDP en múltiples dispositivos, con el objetivo de construir un escenario desconocido de forma cooperativa.

Lo anterior propone retos tecnológicos dentro de los que se identifica la limitación en el ancho de banda [2] y tiempo de respuesta de procesamiento, en cuanto a enrutamiento con el fin de analizar el rendimiento y proponer soluciones y mejoras en tareas de SLAM (Simultaneous Localization And Mapping). Para ello, se recurre a herramienta de simulación de redes, NS2 (Network Simulator - 2), implementando un escenario plano, donde los nodos se mueven en una trayectoria cerrada formando una estrella, transmitiendo datos UDP que

corresponden a la información del escenario de exploración.

En este artículo no se construye el escenario, únicamente se realiza el análisis de tráfico UDP, con el fin de determinar el impacto de trayectorias cerradas en redes Ad Hoc. Los resultados obtenidos en este artículo servirán de insumo para trabajos cooperativos en SLAM.

Los protocolos que se analizan son: AODV (Ad-hoc On-Demand Distance Vector) y AOMDV (Ad hoc On-Demand Multipath Distance Vector). Se escogen estos protocolos por son algunos de los más representativos y para analizar el tráfico de protocolos especializados en una única ruta y múltiples rutas, respectivamente. Finalmente, se presentará una comparación del rendimiento de estos protocolos.

II. ANTECEDENTES

Los protocolos de enrutamiento On-Demand funcionan bajo el principio de la creación de rutas a la medida, lo que quiere decir que se van creando bajo demanda en la red, por requerimiento de un los nodos fuente y destino, generalmente como respuesta a un cambio topológico en la red.

I. AD-HOC ON-DEMAND DISTANCE VECTOR (AODV)

AODV es un protocolo reactivo que descubre rutas en función de las necesidades de la red, mediante un mecanismo de descubrimiento de rutas, basado en tablas de enrutamiento que utiliza como información de entrada la distancia al nodo destino. Sin el uso de la información de enrutamiento del nodo fuente, AODV confía en la información de entrada registrada en su tabla de enrutamiento, esta información es construida por un proceso de inundación previa o de inicio en la red, al propagar un paquete RREP (Route Reply) al nodo fuente.

AODV utiliza números de secuencia de mantenimiento al nodo con el fin de actualizar la información de la tabla de enrutamiento y evitar bucles de enrutamiento [3]. Todos los paquetes de enrutamiento están marcados con estos números de secuencia. AODV mantiene estados basados en eventos de temporización en cada nodo, para monitorear el enrutamiento en las tablas, por lo que las entradas que no son utilizadas presentan un tiempo mayor y se procede a eliminar esta posición de la tabla.

Este trabajo se realiza en conjunto con los nodos intermedios que mantienen información que indica al conjunto de nodos vecinos utilizando paquetes notificados con RERR (Route Error) cuando el enlace siguiente salto se destruye. Este paquete se remite por cada nodo predecesor a sus predecesores, borrando así todas las rutas que utilizaban el enlace de la ruta destruida. La propagación de errores de ruta en AODV se puede visualizar conceptualmente como un árbol

J. Rodríguez, Fundación Universitaria Los Libertadores, Bogotá D.C., Colombia, jrodriguez1@libertadores.edu.co

H. Gómez, Fundación Universitaria Los Libertadores, Bogotá D.C., Colombia, hmgomezr@libertadores.edu.co

J. Anzola, Fundación Universitaria Los Libertadores, Bogotá D.C., Colombia, jrodriguez1@libertadores.edu.co



cuya raíz es el nodo en el punto de ruptura y los nodos que intervinieron en la ruta hacia el nodo fuente son considerados como hojas [4].

II. AD-HOC ON-DEMAND MULTI PATH DISTANCE VECTOR ROUTING (AOMDV)

AOMDV es una extensión del protocolo AODV para encontrar múltiples rutas sin bucles y enlaces disjuntos [2], [5], [6]. Las entradas de enrutamiento para cada destino contienen en una tabla la información de los siguientes saltos junto con la correspondencia del número de saltos. Los saltos siguientes deben de tener el mismo número de secuencia, para ayudar a realizar el seguimiento de una ruta. Para cada uno de los destinos, el nodo fuente mantiene el recuento de los saltos, que se define como el número máximo de saltos para todas las rutas. En el proceso de inundación y envío de información se encuentran mensajes duplicados y el anuncio de una ruta recibida por un nodo define una ruta alternativa al destino.

Cuando un anuncio de ruta se recibe en un destino con un mayor número de secuencia, la información de registro en la tabla del siguiente salto y el número de saltos anunciados se reinicializa.

AOMDV utiliza el descubrimiento de nodos, enlaces y rutas disjuntas. Para encontrar rutas de nodos disjuntos, cada nodo no rechaza de forma inmediata los paquetes RREQ (Route Request) duplicados [7], [8]. Cada RREQ que llegan a través de un nodo vecino diferente al nodo fuente define una trayectoria de nodo disjuntos. Esto se debe a que los nodos no se pueden difundir paquetes RREQ duplicado, por lo que, si dos o más paquetes RREQ llegan a un nodo intermedio a través de un nodo vecino diferente al nodo fuente, no podría haber atravesado el mismo nodo intermedio. En un intento de conseguir enlaces de conexión disjuntos y múltiples rutas, los nodos destino responden a paquetes duplicados RREQ, donde el nodo destino sólo responde a las RREQ únicos que llegan por sus nodos vecinos. Después del primer salto, los paquetes RREP (Route Reply) siguen las rutas inversas o hacia atrás, que corresponde enlaces de nodos disjuntos. Las trayectorias de cada paquete RREP se pueden intersectar en un nodo intermedio, pero la decisión de tomar una ruta inversa diferente a la del nodo fuente garantiza enlace disjuntos.

III. METODOLOGÍA DE EVALUACIÓN

La implementación física de cada uno de los nodos y la construcción de los escenarios de pruebas realistas, requieren de infraestructura, por lo tanto, como alternativa se utiliza procesos de simulación que aproximan los resultados esperados de la implementación real. Con las dificultades encontradas al reproducir factores, tales como: el comportamiento humano, superficies que no son planas, heterogeneidad de patrones de radiación, radiación externa y otros factores que no se contemplan en la simulación, por lo tanto, los resultados obtenidos en este trabajo hacen parte de un análisis de aproximación cuantitativa y cualitativa.

Metodología

Para comparar los protocolos de enrutamiento se utilizan entornos de simulación idénticos para su evaluación de

desempeño.

El tamaño máximo de paquetes en la cola es de 50 sin ningún tipo de prioridad y con una disciplina de cola FIFO (First In, First Out), basando las simulaciones en un total de en 20 nodos inalámbricos que forman una red Ad Hoc, moviéndose en un área de (200m x 200m) durante un tiempo de 300 segundos. Dentro del área de simulación se eligen movimientos cerrados controlados manteniendo la misma densidad, permitiendo comparar los protocolos de enrutamiento. Como carga de tráfico background o tráfico de red, se generan 10 conexiones de tipo TCP y una conexión UDP donde se evalúa el tráfico punto a punto.

I. MODELO DE MOVIMIENTO

En la simulación, el movimiento de cada nodo hace parte de una distribución normal con parámetros μ igual a la velocidad media de evaluación y σ^2 con varianza igual a uno. Cada nodo móvil comienzan en un punto aleatorio con longitudes de desplazamiento uniforme formando una estrella, la velocidad de cada nodo es uniforme, pero la distribución de las velocidades en toda la red es normal. Esta evaluación se realizó para valores de velocidad desde un metro de desplazamiento por segundo hasta 20 m/s (Ver Figura 1).

II. MODELO DE COMUNICACIÓN

La fuente de tráfico a evaluar es CBR (Constant Bit Rate), entre un nodo fuente y un nodo destino extendidos aleatoriamente en la red. El tamaño de los paquetes utilizados fue de 512 bytes a una tasa de 150 paquetes por segundo. El número de fuentes de tráfico TCP con aplicación FTP utilizado fue aproximadamente equivalente a la variación de la tasa de envío con la mitad de las fuentes de tráfico del total del número de nodos, para un total de diez enlaces TCP.

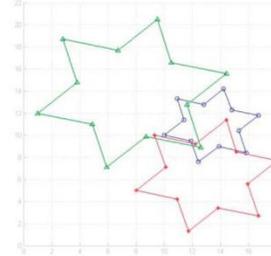


Figura 1. Movimientos realizados por cada uno de los nodos en la red.

IV. MÉTRICAS DE EVALUACIÓN

Las métricas de comparación y evaluación del rendimiento entre los protocolos AODV y AOMDV son:

I. RETARDO DE EXTREMO A EXTREMO (DELAY)

Esta métrica es el tiempo promedio que transcurrido desde que se inicia la transmisión de un paquete desde el nodo fuente al nodo de destino. Esta métrica incluye todos los retrasos posibles presentados en la red, aumentando

principalmente por eventos de iniciación, mantenimiento y descubrimiento de rutas, tiempos de procesamiento, tiempos de duración de un paquete en la cola de nodos intermedios, retardos de retransmisión de paquetes para tráfico TCP, tiempo de acceso al canal, sobrecarga de los protocolos, entre otros. El retardo viene dada por:

$$Retardo = \frac{\sum_{i=1}^{N_r} t_{R_i} - t_{S_i}}{N_r} \quad (1)$$

Donde:

t_{R_i} es el tiempo de llegada de un paquete en el nodo destino y t_{S_i} es el tiempo en el que el paquete fue enviado desde el nodo fuente y N_r es el número total de paquetes recibidos.

II. JITTER

La métrica del Jitter es una medida de la variación del retardo de transferencia entre paquetes, puede depender de las rutas de los paquetes y flujo presente en las colas de los nodos intermedios. Hay varias definiciones de fluctuación que intentan capturar la variación del retardo de paquetes. En el desarrollo de este artículo la definición de Jitter tratada en IETF [9], se adopta y se define a continuación:

$$Jitter = E|T_{i+1} - T_i| \quad (2)$$

Donde:

T_i es el tiempo de transmisión del paquete i y T_{i+1} representa el momento de la transmisión del siguiente paquete.

III. PACKET DELIVERY RATIO (PDR)

Es la relación entre el número de paquetes de datos enviado desde el nodo fuente al nodo destino, en términos generales se evalúa el índice de paquetes de llegada, obteniendo la tasa de pérdidas de paquetes. Esta medida muestra la fiabilidad del protocolo de enrutamiento, en cuanto mayor sea este índice, más fiable es la entrega de paquetes en la red. Esta métrica está dada por:

$$Indice_{PDR} = \frac{T_R}{T_S} \quad (3)$$

Donde: T_R es el número total de paquetes recibidos por el nodo de destino y T_S es el número total de paquetes enviados por el nodo fuente.

IV. THROUGHPUT O RENDIMIENTO

El Throughput o rendimiento se define como el número de bits recibidos con éxito por el nodo destino dividido por el total de tiempo de transmisión en cuestión de segundos, también se interpreta como la tasa de satisfactoria de transmisiones. A continuación se define como:

$$Throughput = \frac{\sum T_{PR} \times 8}{T_{TS} \times 1000} \quad (4)$$

Donde: T_{PR} es el número total de paquetes recibidos por el nodo de destino y T_{TS} es el tiempo total de la simulación.

V. ANÁLISIS DE RESULTADOS

A continuación se muestra los resultados obtenidos:

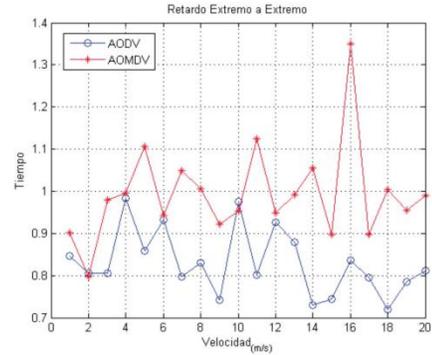


Figura 2. Movimientos realizados por cada uno de los nodos en la red.

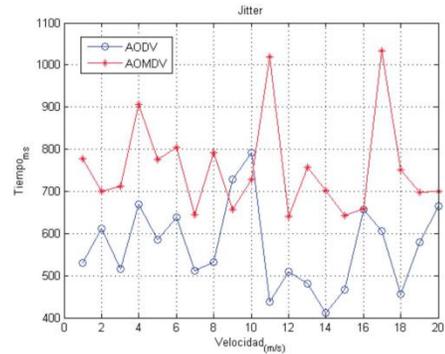


Figura 3. Movimientos realizados por cada uno de los nodos en la red.

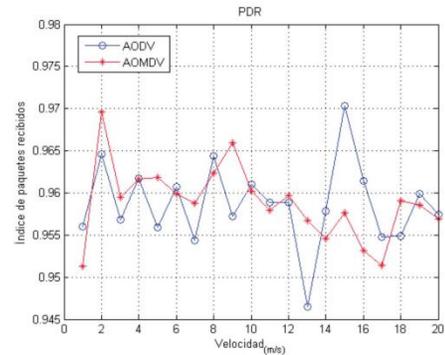


Figura 4. Movimientos realizados por cada uno de los nodos en la red.

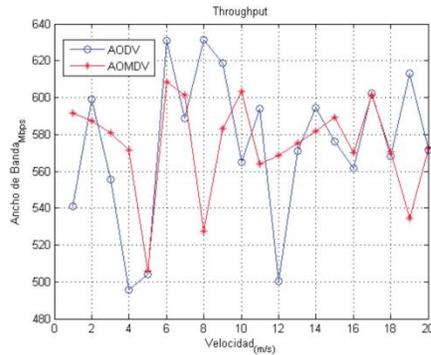


Figura 5. Movimientos realizados por cada uno de los nodos en la red.

Las trayectorias cerradas permitieron tener una tendencia de pérdida de paquetes que nos superaron un índice del 0.93 en relación con la velocidad. En cuanto al retardo de paquetes AODV mantiene el menor retardo con relación a AOMDV debido a carga de paquetes que presentan los protocolos, recordando que AOMDV es multitrayectoria, mientras que AODV es dedicado a una única trayectoria. El ancho de banda consumido debe mantenerse entre una banda mínima de 500 a 650 kbps, no se vio fuertemente afectado por los cambios de velocidad. Para futuros sistemas de transmisión de datos para tareas de SLAM, se recomienda evaluar la localización y mapeado simultáneos mediante trayectorias cerradas, dado que esta permiten una pérdida del 6% de paquetes que sigue siendo elevada.

VI. CONCLUSIONES

El análisis de desempeño de dos protocolos de enrutamiento AODV y AOMDV bajo el enfoque On-demand en la base a las métricas de Retardo, Jitter, PDR y el Throughput se observa que el rendimiento de AODV sobre AOMDV es mejor en todos los aspectos, tanto en velocidades bajas hasta 3m/s como velocidades altas mayores de 15m/s. Esta medición del desempeño ayudará para seleccionar al protocolo AODV como protocolo de comunicación en redes Ad Hoc donde se implemente SLAM.

Para el escenario empleado de 20 nodos móviles moviéndose a una velocidad uniforme y distribuida de forma normal, con un movimiento controlado por una trayectoria en forma de estrella, se obtuvo una pérdida de paquetes que tiende a permanecer constante frente a la variación de velocidad.

El ancho de banda se ha limitado al promedio del tráfico background presente en la red, a pesar de que se realice una comunicación punto a punto, la calidad de servicio de la red no se deteriora. Sin embargo hay trabajos futuros de evaluación en cuanto al rendimiento computacional al implementar los protocolos para una aplicación de SLAM totalmente descentralizada en un escenario Ad Hoc.

REFERENCIAS

- [1] K. Fall and K. Varadhan, "The network simulator NS-2," URL <http://www.isi.edu/nsnam/ns>, 2012.
- [2] N. C. V. N. Pereira and R. M. de Moraes, "Comparative Analysis of AODV Route Recovery Mechanisms in Wireless Ad Hoc Networks," *Lat. Am. Trans. IEEE (Revista IEEE Am. Lat.*, vol. 8, no. 4, pp. 385–393, 2010.
- [3] W. Araujo Albano, M. Nogueira, and J. de Souza, "A Taxonomy for Resilience in Vehicular Ad hoc Networks," *Lat. Am. Trans. IEEE (Revista IEEE Am. Lat.*, vol. 13, no. 1, pp. 228–234, 2015.
- [4] A. Vidwans, A. K. Shrivastava, and M. Manoria, "QoS Enhancement of AOMDV Routing Protocol Using Queue Length Improvement," in *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on*, 2014, pp. 275–278.
- [5] M. Rios, "Variable Route Expiration Time based on a Fixed Probability of Failure for Ad-Hoc Networks Routing Applications," *Lat. Am. Trans. IEEE (Revista IEEE Am. Lat.*, vol. 13, no. 1, pp. 383–389, 2015.
- [6] B. Sharma, S. Chugh, and V. Jain, "Energy Efficient Load Balancing Approach to Improve AOMDV Routing in MANET," in *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on*, 2014, pp. 187–192.
- [7] U. S. Kushwaha and P. K. Gupta, "AOMDV routing algorithm for Wireless Mesh Networks with local repair (AOMDV-LR)," in *Communications and Signal Processing (ICCSP), 2014 International Conference on*, 2014, pp. 818–822.
- [8] C. Perkins, E. Belding-Royer, S. Das, and others, "RFC 3561-ad hoc on-demand distance vector (AODV) routing," *Internet RFCs*, pp. 1–38, 2003.
- [9] T. Clausen, B. Adamson, and C. Dearlove, "Jitter considerations in mobile ad hoc networks (MANETs)," 2008.



Juan Camilo Rodriguez Final year student of Electronic Engineering of the Fundación Universitaria Los Libertadores. Participates as a researcher in the hotbed of Signals and Systems group GUIAS.



Hector Medardo Gomez Final year student of Electronic Engineering of the Fundación Universitaria Los Libertadores. Participates as a researcher in the hotbed of Signals and Systems group GUIAS.



John Anzola Teaching program of Electronic Engineering University Fundación Universitaria Los Libertadores, MSc in Information Sciences and Communications University Distrital Francisco José de Caldas, is currently studying the first year of PhD Engineering of the University Distrital Francisco José de Caldas, Bogotá D.C., Colombia. Integral is Research groups GICOECOL and GUIAS.

Traffic Analysis in Ad Hoc Networks for Applications in Bean Fields

B. Robayo, J. Rodríguez and H. Gomez

Abstract— in this paper, data traffic occurs over a field of agriculture, more exactly in beans, where problems such as moisture, soil conditions, water conditions bean among other presents. This recognition is done by moving nodes 20 at a constant speed. The path of each node is closed and its displacement is in a figure eight, giving you the option to get a 360 degree free at a particular area. The work Presented in this article is Analyzed data traffic metrics delay, jitter, throughput and PDR.

Keywords— Wireless Communications, bean, AODV, DSR, Ad Hoc.

I. INTRODUCCION

UN aspecto tecnológico presente en los sistemas de reconocimiento de terrenos y espacios es el problema de la localización y mapeado simultáneos, el cual permite descubrir si es posible para un robot móvil navegar a través de un entorno desconocido y construir de manera incremental, un mapa consistente del mismo, mientras que se determina al mismo tiempo su posición dentro de este mapa.

El frijol es una de las principales actividades de la economía campesina en varias regiones del país, ya que genera ingresos, empleo rural y como producto básico en la dieta alimenticia de la población por su alto contenido de proteínas y de elementos minerales esenciales.

Los principales inconvenientes que se tienen para la producción de frijol en Colombia, están relacionados con la alta incidencia de enfermedades y plagas agravan el crecimiento de la semilla, lo cual exige un alto uso de plaguicidas para su manejo con consecuencias negativas como la alta exposición y riesgo de los trabajadores a intoxicaciones, la contaminación del medio. [1].

Para minimizar este riesgo, y a su vez mejorar la producción de este elemento se realizara una toma de datos como las plagas, humedad y todos los factores mencionados anteriormente por medio de robots los cuales tendrán un determinado movimiento y estos realizaran el envío y recepción de esta información por medio de redes Ad Hoc, para llevar un control específico de las mejores condiciones para la producción del frijol por esta razón. En este artículo se analiza el uso de trayectorias cerradas en cuanto a movimiento, con referencia al Delay, Jitter, PDR y el Throughput, dado que para los problemas de localización y mapeado simultáneo.

B. Robayo, Fundación Universitaria Los Libertadores, Bogotá D.C., Colombia, barobayog@libertadores.edu.co

J. Rodríguez, Fundación Universitaria Los Libertadores, Bogotá D.C., Colombia, hmgomezr@libertadores.edu.co

H. Gómez, Fundación Universitaria Los Libertadores, Bogotá D.C., Colombia, hmgomezr@libertadores.edu.co

Se propone el estudio del comportamiento de tráfico UDP en múltiples dispositivos, con el objetivo de construir un escenario desconocido de forma cooperativa.

Lo anterior propone retos tecnológicos dentro de los que se identifica la limitación en el ancho de banda [2] y tiempo de respuesta de procesamiento y proponer soluciones y mejoras para los cultivos de frijol que actualmente siembran, los campesinos colombianos. Para ello, se recurre a herramienta de simulación de redes, NS2 (Network Simulator - 2) [3], implementando un escenario plano, donde los nodos se mueven en una trayectoria cerrada formando un girasol, transmitiendo datos UDP que corresponden a la información del escenario de exploración.

En este artículo no se construye el escenario, únicamente se realiza el análisis de tráfico UDP, con el fin de determinar el impacto de trayectorias cerradas en redes Ad Hoc y se hace a bajas velocidades las posibles que se pueden llegar alcanzar por medio de robots con orugas y las condiciones del terreno.

Los protocolos que se analizan son: AODV (Ad-hoc On-Demand Distance Vector) [4] y DSDV (Destination-Sequenced Distance-Vector Routing) [5]. Se escogen estos protocolos por algunos de los más representativos y para analizar el tráfico de protocolos especializados en una única ruta y múltiples rutas, respectivamente. Finalmente, se presentará una comparación del rendimiento de estos protocolos.

II. ANTECEDENTES

Los protocolos de enrutamiento On-Demand funcionan bajo el principio de la creación de rutas a la medida, lo que quiere decir que se van creando bajo demanda en la red, por requerimiento de un los nodos fuente y destino, generalmente como respuesta a un cambio topológico en la red.

I. AD-HOC ON-DEMAND DISTANCE VECTOR (AODV)

AODV es un protocolo reactivo que descubre rutas en función de las necesidades de la red, mediante un mecanismo de descubrimiento de rutas, basado en tablas de enrutamiento que utiliza como información de entrada la distancia al nodo destino. Sin el uso de la información de enrutamiento del nodo fuente, AODV confía en la información de entrada registrada en su tabla de enrutamiento, esta información es construida por un proceso de inundación previa o de inicio en la red, al propagar un paquete RREP (Route Reply) al nodo fuente.

AODV utiliza números de secuencia de mantenimiento al nodo con el fin de actualizar la información de la tabla de

enrutamiento y evitar bucles de enrutamiento [6]. Todos los paquetes de enrutamiento están marcados con estos números de secuencia. AODV mantiene estados basados en eventos de temporización en cada nodo, para monitorear el enrutamiento en las tablas, por lo que las entradas que no son utilizadas presentan un tiempo mayor y se procede a eliminar esta posición de la tabla.

Este trabajo se realiza en conjunto con los nodos intermedios que mantienen información que indica al conjunto de nodos vecinos utilizando paquetes notificados con RERR (Route Error) cuando el enlace siguiente salto se destruye. Este paquete se remite por cada nodo predecesor a sus predecesores, borrando así todas las rutas que utilizaban el enlace de la ruta destruida. La propagación de errores de ruta en AODV se puede visualizar conceptualmente como un árbol cuya raíz es el nodo en el punto de ruptura y los nodos que intervinieron en la ruta hacia el nodo fuente son considerados como hojas [1].

La ventaja de utilizar AODV es la minimización del espacio de memoria ya que solo mantiene información de las rutas activas aumentando el rendimiento. Por otro lado, una de las desventajas de este protocolo, radica en que no es escalable y en grandes redes no funciona correctamente y no es compatible con enlaces y rutas asimétricas.

II. DSDV (DESTINATION-SEQUENCED DISTANCE-VECTOR ROUTING)

En DSDV la entrega de paquetes es debido a que utiliza rutas anteriores en caso de haber enlaces rotos. En DSDV la extinción de enlaces anteriores no implica que no sea una ruta válida hasta su destino final. Los paquetes pueden ser enviados a través de otros vecinos que pueden tener las rutas hacia el destino final. [7].

Es un esquema de enrutamiento impulsado por redes móviles ad hoc basado en el algoritmo de Bellman-Ford. Fue desarrollado por C. Perkins y P. Bhagwat en 1994. La principal contribución del algoritmo era resolver el problema de bucle de enrutamiento. Cada entrada en la tabla de enrutamiento contiene un número de secuencia, los números de secuencia son generalmente incluso si un enlace está presente; más, se utiliza un número impar. El número es generado por el destino, y el emisor debe enviar la próxima actualización con este número. Información de enrutamiento se distribuye entre los nodos mediante el envío de copias completas con poca frecuencia y actualizaciones incrementales más pequeñas con más frecuencia. [8]. DSDV define dos métodos de actualización de la información: copias completas e incrementales; el primer modo se utiliza cuando la actualización para ser enviado solo requiere un NPDU (unidad de red de paquetes de datos); El segundo modo se utiliza cuando se necesitan más NPDU para contener la entrada de actualizar, o cuando un nodo detecta un cambio sustancial a la topología de la red.

Naturalmente, la tabla se describen todas las posibles rutas alcanzables desde el nodo A, además de la siguiente salto, número de saltos y el número de secuencia. Cada nodo también utiliza un valor de tiempo que define el tiempo en el

que una nueva ruta aprendió pueden ser validados y reenviado (para evitar bucles) de sedimentación.

III. METODOLOGÍA DE EVALUACIÓN

La implementación física de cada uno de los nodos y la construcción de los escenarios de pruebas realistas, requieren de infraestructura, por lo tanto, como alternativa se utiliza procesos de simulación que aproximan los resultados esperados de la implementación real. Con las dificultades encontradas al reproducir factores, tales como: el comportamiento humano, superficies que no son planas, heterogeneidad de patrones de radiación, radiación externa y otros factores que no se contemplan en la simulación, por lo tanto, los resultados obtenidos en este trabajo hacen parte de un análisis de aproximación cuantitativa y cualitativa.

Metodología

Para comparar los protocolos de enrutamiento se utilizan entornos de simulación idénticos para su evaluación de desempeño.

El tamaño máximo de paquetes en la cola es de 50 sin ningún tipo de prioridad y con una disciplina de cola FIFO (First In, First Out), basando las simulaciones en un total de 20 nodos inalámbricos que forman una red Ad Hoc, moviéndose en un área de (200m x 200m) durante un tiempo de 300 segundos. Dentro del área de simulación se eligen movimientos cerrados controlados manteniendo la misma densidad, permitiendo comparar los protocolos de enrutamiento. Como carga de tráfico background o tráfico de red, se generan 10 conexiones de tipo TCP y una conexión UDP donde se evalúa el tráfico punto a punto.

I. MODELO DE MOVIMIENTO

En la simulación, el movimiento de cada nodo hace parte de una distribución normal con parámetros μ igual a la velocidad media de evaluación y θ^2 con varianza igual a uno. Cada nodo móvil comienzan en un punto aleatorio con longitudes de desplazamiento uniforme formando una estrella, la velocidad de cada nodo es uniforme, pero la distribución de las velocidades en toda la red es normal. Esta evaluación se realizó para valores de velocidad desde un metro de desplazamiento por segundo hasta 20 m/s (Ver Figura 1).

II. MODELO DE COMUNICACIÓN

La fuente de tráfico a evaluar es CBR (Constant Bit Rate), entre un nodo fuente y un nodo destino extendidos aleatoriamente en la red. El tamaño de los paquetes utilizados fue de 512 bytes a una tasa de 150 paquetes por segundo. El número de fuentes de tráfico TCP con aplicación FTP utilizado fue aproximadamente equivalente a la variación de la tasa de envío con la mitad de las fuentes de tráfico del total del número de nodos, para un total de diez enlaces TCP.

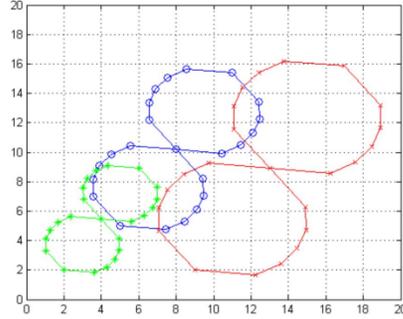


Figura 1. Movimientos realizados por cada uno de los nodos en la red.

IV. MÉTRICAS DE EVALUACIÓN

Las métricas de comparación y evaluación del rendimiento entre los protocolos AODV y DSDV son:

I. RETARDO DE EXTREMO A EXTREMO (DELAY)

Esta métrica es el tiempo promedio que transcurre desde que se inicia la transmisión de un paquete desde el nodo fuente al nodo de destino. Esta métrica incluye todos los retrasos posibles presentados en la red, aumentando principalmente por eventos de iniciación, mantenimiento y descubrimiento de rutas, tiempos de procesamiento, tiempos de duración de un paquete en la cola de nodos intermedios, retardos de retransmisión de paquetes para tráfico TCP, tiempo de acceso al canal, sobrecarga de los protocolos, entre otros. El retardo viene dada por:

$$Retardo = \frac{\sum_{i=1}^{N_r} t_{R_i} - t_{S_i}}{N_r} \quad (1)$$

Donde:

t_{R_i} es el tiempo de llegada de un paquete en el nodo destino y t_{S_i} es el tiempo en el que el paquete fue enviado desde el nodo fuente y N_r es el número total de paquetes recibidos.

II. JITTER

La métrica del Jitter es una medida de la variación del retardo de transferencia entre paquetes, puede depender de las rutas de los paquetes y flujo presente en las colas de los nodos intermedios. Hay varias definiciones de fluctuación que intentan capturar la variación del retardo de paquetes. En el desarrollo de este artículo la definición de Jitter tratada en IETF [9], se adopta y se define a continuación:

$$Jitter = E T_{i+1} - T_i \quad (2)$$

Donde:

T_i es el tiempo de transmisión del paquete i y T_{i+1} representa el momento de la transmisión del siguiente paquete.

III. PACKET DELIVERY RATIO (PDR)

Es la relación entre el número de paquetes de datos enviado desde el nodo fuente al nodo destino, en términos generales se evalúa el índice de paquetes de llegada, obteniendo la tasa de pérdidas de paquetes. Esta medida muestra la fiabilidad del protocolo de enrutamiento, en cuanto mayor sea este índice, más fiable es la entrega de paquetes en la red. Esta métrica está dada por:

$$Indice_{PDR} = \frac{T_R}{T_S} \quad (3)$$

Donde:

T_R es el número total de paquetes recibidos por el nodo de destino y T_S es el número total de paquetes enviados por el nodo fuente.

IV. THROUGHPUT O RENDIMIENTO

El Throughput o rendimiento se define como el número de bits recibidos con éxito por el nodo destino dividido por el total de tiempo de transmisión en cuestión de segundos, también se interpreta como la tasa de satisfactoria de transmisiones. A continuación se define como:

$$Throughput = \frac{T_{PR} \times 8}{T_{TS} \times 1000} \quad (4)$$

Donde:

T_{PR} es el número total de paquetes recibidos por el nodo de destino y T_{TS} es el tiempo total de la simulación.

V. ANÁLISIS DE RESULTADOS

A continuación se muestra los resultados obtenidos:

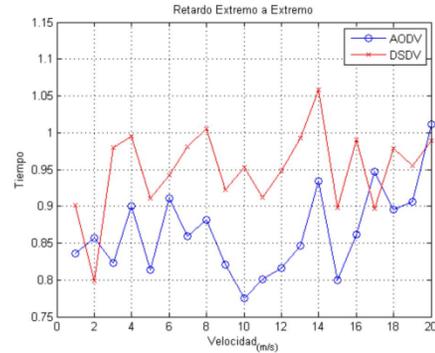


Figura 2. Movimientos realizados por cada uno de los nodos en la red.

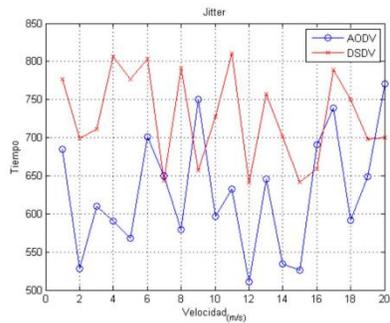


Figura 3. Movimientos realizados por cada uno de los nodos en la red.

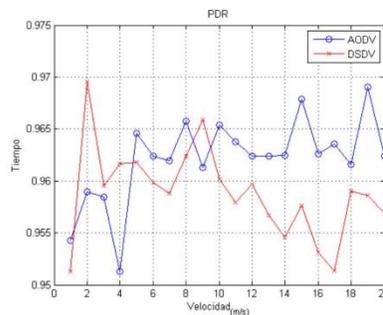


Figura 4. Movimientos realizados por cada uno de los nodos en la red.

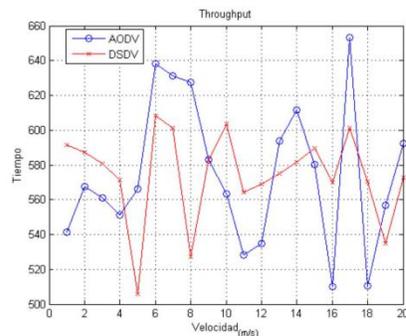


Figura 5. Movimientos realizados por cada uno de los nodos en la red.

VI. CONCLUSIONES

El análisis de desempeño de dos protocolos de enrutamiento AODV y DSDV bajo el enfoque On-demand en la base a las métricas de Retardo, Jitter, PDR y el Throughput se observa que el rendimiento de AODV sobre DSDV es mejor en todos los aspectos, tanto en velocidades bajas hasta 3m/s como velocidades altas mayores de 15m/s. Esta medición del desempeño ayudará para seleccionar al protocolo AODV como protocolo de comunicación en redes Ad Hoc para la implementación en cultivos de frijol.

Para el escenario especificado de 20 nodos móviles moviéndose a una velocidad uniforme y distribuida de forma normal, con un movimiento reconocido por una trayectoria en forma de ocho, se alcanzó una pérdida de paquetes que tiende a permanecer invariable frente a la variación de velocidad.

El ancho de banda se limita al promedio del tráfico background presente en la red, a pesar de realizarse una comunicación punto a punto, la calidad de servicio de la red no se deteriora.

REFERENCIAS

- [1] J. Arias Restrepo, T. Rengifo Martinez, and M. Jaramillo, "Manual Técnico en La Producción de Frijol Voluble" URL <http://www.fao.org/co/manualfrijol.pdf>
- [2] M. Rios, "Variable Route Expiration Time based on a Fixed Probability of Failure for Ad-Hoc Networks Routing Applications," *Lat. Am. Trans. IEEE (Revista IEEE Am. Lat.*, vol. 13, no. 1, pp. 383-389, 2015.
- [3] K. Fall and K. Varadhan, "The network simulator NS-2," URL <http://www.isi.edu/nsnam/ns>, 2012.
- [4] N. C. V. N. Pereira and R. M. de Moraes, "Comparative Analysis of AODV Route Recovery Mechanisms in Wireless Ad Hoc Networks," *Lat. Am. Trans. IEEE (Revista IEEE Am. Lat.*, vol. 8, no. 4, pp. 385-393, 2010.
- [5] M. Manjunath, "Spatial DSDV (S-DSDV) routing algorithm for mobile ad hoc network," *Contemporary Computing and Informatics (IC3I), 2014 International Conference on*, vol. no., pp.625,629, 27-29 Nov. 2014
- [6] C. Perkins, E. Belding-Royer, S. Das, and others, "RFC 3561-ad hoc on-demand distance vector (AODV) routing," *Internet RFCs*, pp. 1-38, 2003.
- [7] Khan, K.; Zaman, R.U.; Reddy, K.A.; Reddy, K.A.; Harsha, T.S., "An Efficient DSDV Routing Protocol for Wireless Mobile Ad Hoc Networks and its Performance Comparison," *Computer Modeling and Simulation, 2008. EMS '08. Second UKSIM European Symposium on*, vol., no., pp.506,511, 8-10 Sept. 2008.
- [8] Perkins, Charles E. and Bhagwat, Pravin (1994). "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers" (PDF). Retrieved 2006-10-20.
- [9] T. Clausen, B. Adamson, and C. Dearlove, "Jitter considerations in mobile ad hoc networks (MANETs)," 2008.



Brayan Alexander Robayo. Final year student of Electronic Engineering of the Fundación Universitaria Los Libertadores. Participates as a researcher in the hotbed of Signals and Systems group GUAS.



Juan Camilo Rodriguez. Final year student of Electronic Engineering of the Fundación Universitaria Los Libertadores. Participates as a researcher in the hotbed of Signals and Systems group GUAS.



Hector Medardo Gomez Final year student of Electronic Engineering of the Fundación Universitaria Los Libertadores. Participates as a researcher in the hobbed of Signals and Systems group GUAS.