

COMPORTAMIENTO DE LOS PROTOCOLOS DE ENCAMINAMIENTO EN REDES MANET (MOBILE AD HOC NETWORK)

ROUTING PROTOCOLS' BEHAVIOR IN MANET NETWORKS (MOBILE Ad HOC NETWORK)

Recibido: 20 de abril del 2010

Aprobado: 25 de mayo del 2010

FOUAD ZIAD OTHMAN*
ALI MOHAMAD KHALED**

Universidad Central "Marta Abreu de las Villas"
fouad@uclv.edu.cu

Resumen

Las redes móviles Ad-Hoc o redes de múltiple salto están formadas por enlaces de radio entre nodos, sin la necesidad de una administración central. Además de funcionar como terminal final, cada nodo realiza funciones de retransmisión de paquetes (enrutadores) para extender la comunicación hasta destinos que se encuentran fuera del área de cobertura. Para lograr tales fines, se requiere de un protocolo de encaminamiento que permita crear rutas hacia los destinos deseados, los cuales permiten que dicha red sea autoconfigurada y autorganizada eliminando la necesidad de una infraestructura de red predefinida. Se han desarrollado principalmente dos tipos de protocolos: reactivos y proactivos. En este artículo se hace un estudio mediante técnicas de simulación de eventos discretos usando el simulador de redes NS-2, para evaluar el comportamiento de diferentes protocolos de encaminamiento relativos al throughput, entrega y pérdida de paquetes.

Palabras clave: MANET, encaminamiento, Ad Hoc, simulación, NS-2.

Abstract

The mobile Ad-Hoc networks or multiple jump networks are formed by radio links between nodes, without the need of central administration. In addition of working as final terminal, every node performs functions of bundles' retransmission (routers) to extend communication up to destinations that are out of the coverage area. To achieve such goals, it requires a routing protocol that allows creating routes towards the destinations expected; these allow such network to be auto-configured and auto-organized eliminating the need of a predefined network infrastructure. Principally two types of protocols have been developed: reactive and proactive. In this article we study through simulation of discreet events using the networks simulator NS-2, to evaluate the behavior of different routing protocols, relative to the throughput, delivery and loss of bundles.

Keywords: MANET, routing, Ad Hoc, simulation, NS-2.

• Cómo citar este artículo: Fouad Ziad Othman y Ali Mohamad Khaled (2010) "Comportamiento de los protocolos de encaminamiento en redes MANET (Mobile Ad Hoc Network)", en *Revista Ingeniería Solidaria*, vol. 6, núm. 10, pp. 45-52.

* Ingeniero en Telecomunicaciones y Electrónica. Aspirante a Magíster en Telemática de la Universidad Central de las Villas (Cuba). Convenio Académico Universidad Cooperativa de Colombia, seccional Bogotá, y Universidad Central "Marta Abreu" de las Villas (Cuba). Correo electrónico: fouad@uclv.edu.cu

** Ingeniero en Informática. Aspirante a Magíster de la Universidad Central de las Villas (Cuba). Correo electrónico: khaled@uclv.edu.cu

Introducción

MANET es una red autoorganizada que se forma por los nodos móviles, existente sin la necesidad de una infraestructura previa o una administración central como una estación base (Abolhasan, Wysocki y Dutkiewicz, 2004, p. 154), como muestra la figura 1. Los nodos deben ser capaces de retransmitir los paquetes hacia otros destinos que se encuentran fuera del área de cobertura. El encaminamiento es el proceso mediante el cual se trata de encontrar una ruta o camino entre dos puntos de la red. El algoritmo, encargado de decidir la línea de salida por la que se transmitirá un paquete de entrada (Goitia), tiene como fin encontrar la mejor ruta (con menor retardo, menos salto, menos congestión, etcétera). La tarea vital de los protocolos de encaminamiento es gestionar la información de control de la manera más eficiente para mejorar la estabilidad, permitiendo que las redes Ad Hoc sean autoconfigurables y autoorganizadas, y eliminando la necesidad de la existencia de una infraestructura predefinida.

El encaminamiento en redes Ad Hoc, en comparación con otras, presenta complejidades adicionales debido a que los nodos tienen la libertad de moverse de manera aleatoria dentro de la red (Frodigh y Johansson, 2000, p. 220). Existe una gran cantidad de protocolos de encaminamiento diseñados para redes cableadas, los cuales no son compatibles con redes móviles inalámbricas por sus características propias, como su topología dinámica y la limitación del ancho de banda (Solórzano, 2008, p. 246). Por tales razones, se han desarrollado varios tipos de protocolos de encaminamiento para dichas redes.

Los protocolos de encaminamiento se pueden clasificar según sus características; la forma más conocida se basa en cómo el nodo móvil procesa la información de control. De allí surgen dos tipos principales: los protocolos proactivos, que son básicamente una adaptación de los protocolos vector distancia y estado de enlace de las redes cableadas, y los protocolos reactivos, también llamados bajo demanda, que fueron desarrollados principalmente para las redes MANET. Existe otro tipo de protocolos que se denominan híbridos, en los que se mantiene una filosofía proactiva en ámbito local y reactivo a nivel más global.

El encaminamiento en redes Ad Hoc se debe tratar de manera distinta debido a la movilidad de los nodos, que hace que los fallos y las recuperaciones de los enlaces tengan características diferentes a los de otras redes, siendo el índice de errores proporcional a la velocidad de los nodos (Hernández, 2006, p. 200). El objetivo de este trabajo es evaluar mediante técnicas de simulación de eventos discretos el comportamiento de los diferentes protocolos de encaminamiento usando el Network Simulator ns-2. El artículo está organizado de la siguiente manera: primero se ofrece una base teórica de los protocolos de encaminamiento y una introducción general del ns-2, posteriormente se hace una descripción de los escenarios desarrollados, para luego hacer el análisis y discusión de resultados.

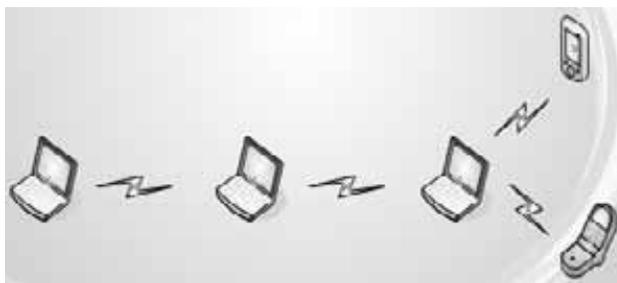


Figura 1. Red Ad Hoc

Fuente: los autores

Protocolos de encaminamiento

Encaminamiento proactivo

En el encaminamiento proactivo se actualizan periódicamente las tablas de ruteo para poder optimizar eficientemente la ruta adecuada de acuerdo con su costo; así, cuando se genera tráfico, éste será enviado por la mejor ruta (no necesariamente la más corta) y no se incurrirá en retardos por descubrimientos de rumbos. La creación y mantenimiento de rutas se da por actualizaciones periódicas o desencadenadas por eventos, como adición o pérdida de conexión con algún nodo. Estas actualizaciones pueden hacer que la carga de la red se incremente limitando así el ancho de banda y el consumo energético (Pérez, 2006, p. 145). Estos tipos de protocolos fueron modificados de los existentes para redes cableadas (Urra, 2006) como RIP (Routing Information Protocol) y OSPF (Open Shortest Path First).

DSDV (Destination-Sequenced Distance Vector)

El protocolo DSDV es de vector de distancia, que, al igual que RIP, está basado en el algoritmo de Bellman-Ford (Azadm, Rahman y Anwar, 2007, p. 89) para el cálculo de sus rutas, teniendo en su tabla de encaminamiento el conocimiento total de la red. Los nodos vecinos intercambian periódicamente (proactivo) sus tablas de encaminamiento para estimar la distancia a la que se encuentran los demás nodos. Las modificaciones introducidas por DSDV proporcionan básicamente la obtención de rutas sin bucle mediante la introducción del número de secuencia para la determinación de los caminos más nuevos. Aunque DSDV sólo proporciona una ruta para cada destino (singlepath) (Ovando, 2008), siempre elige la vía más corta basándose en el número de saltos hacia éste. DSDV utiliza dos tipos de mensajes de actualización, uno más grande (full-dump) y otro más pequeño (incremental). Los mensajes de este último tipo pueden utilizarse para actualizaciones intermedias entre envíos periódicos (full-dump) de la tabla entera de encaminamiento. Además, se realizan estimaciones de los tiempos de establecimiento de rutas que retrasaran el envío de mensajes incrementales. Cada nodo mantiene una tabla con la siguiente información para cada destino:

- Dirección IP del destino.
- Número de secuencia del destino.
- Próximo salto hacia el destino (dirección IP).
- Coste de la ruta hacia el destino (número de saltos).
- Tiempo de instalación para eliminar rutas antiguas.

Cada nodo envía periódicamente en modo broadcast la tabla actualizada a sus vecinos:

- Cada nodo añade su número de secuencia cuando envía su tabla de encaminamiento.
- Cuando los demás nodos reciben dicha información, actualizan sus tablas de encaminamiento.

Encaminamiento reactivo

El encaminamiento reactivo (bajo demanda) fue diseñado principalmente con el objetivo de reducir la carga de los protocolos proactivos, ya que sólo busca las rutas y las mantiene cuando sea necesario.

La principal ventaja que presenta es la baja cantidad de información de encaminamiento que intercambia, lo cual es ideal en el caso de las redes inalámbricas debido a que los recursos son limitados con respecto a las cableadas. La desventaja es la demora que se introduce a la hora de buscar las rutas solicitadas por los nodos vecinos, ya que en estos protocolos se calculan bajo demanda.

DSR (Dynamic Source Routing)

Este protocolo, que es propio de los Redes Ad Hoc (Alsaadi, 2006, p. 187), se basa en el encaminamiento en el origen, con la ventaja de que los nodos intermedios no tienen por qué mantener actualizada la información para reenviar los paquetes ya que estos traen la decisión de la ruta a tomar. Los paquetes de datos incluyen una cabecera de información de la trayectoria que debe atravesar para llegar a su destino. No inundan la red con información de encaminamiento ya que no la sobrecargan con datos de control (León y Barrera, 2007, p. 320; Chiu, 2006, p. 230), y la inunda cuando solicite una ruta para establecer comunicaciones. En cada nodo se mantiene una memoria caché para almacenar las ruta ya establecidas. En cuanto al proceso de descubrimiento de éstas (figura 2), dicho protocolo genera tres tipos de mensajes, route request, replay y error, que son mensajes de solicitud, respuesta y error respectivamente.

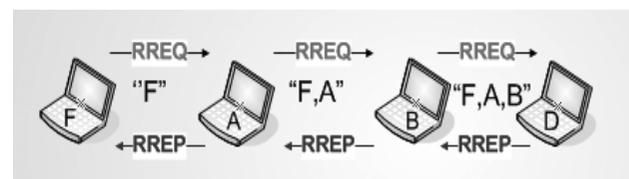


Figura 2. Proceso de descubrimiento de rutas

Fuente: los autores

AODV (Ad Hoc On-Demand Routing Protocol)

AODV es un protocolo bajo demanda que fue diseñado con el objetivo de mejorar el algoritmo DSDV (Perkins, Belding-Royer y Das, 2003, p. 321; Wang y Lu, 2004, p. 234). Emplea el algoritmo vector distancia, que mantiene una tabla de encaminamiento de los nodos activos; es decir, los que no participan en la ruta seleccionada no mantienen la tabla y tam-

poco participan en el intercambio de información. Esta tabla está formada por los vecinos y se le añaden nuevos cuando es necesario. El protocolo AODV usa dos técnicas: descubrimiento y mantenimiento de ruta. El primer proceso ocurre cuando un nodo necesita comunicarse con otro para el cual no hay ruta válida, y el segundo se lleva a cabo cuando un enlace cae. De modo igual al DSR, para iniciar una ruta nueva el protocolo propaga una RREQ (Route Request-Solicitud de ruta) para los nodos vecinos; en cambio, en este protocolo se va generando una tabla de encaminamiento inversa para poder responder las contestaciones RREP (Route Replay-Respuesta de ruta) a las solicitudes de rutas que pueden ser generadas por el destino o por un nodo intermedio que tiene una ruta válida en su tabla (Kurosawa y Nakayama, 2007, p. 456).

AODV usa los números de secuencia para evitar los problemas de cuentas al infinito (Kumar, Reddy y Hiremath, 2008, p. 367). Cuando se recibe una solicitud de ruta en los nodos intermedios, primero se chequea si está duplicada o no, y se descartan las que lo están. Si no está duplicada, el nodo chequea si en su tabla de encaminamiento hay una ruta válida hacia el destino indicado en la solicitud. En caso que no haya, o el número de secuencia Dst_Seq de la ruta válida sea menor que el número de secuencia de la solicitud, entonces el nodo reenvía la solicitud e incrementa el número de salto en ella. Al recibir una solicitud de ruta, el nodo intermedio crea o actualiza la entrada en su tabla de ruteo que corresponde a la fuente del solicitante. Si la entrada existe, actualiza el número de secuencia en caso de que éste sea menor que el número de secuencia de la solicitud, o si los dos son iguales pero el número de salto de la nueva ruta indicado en el solicitud es menor. Cuando se descubra una ruptura en un enlace que no puede ser reparada, se envía RERR para notificarla a los precursores y luego invalidar la ruta en la tabla (Ramírez, 2006). En cambio, si el nodo origen de un envío es el que se mueve y altera la topología de la red, debe reiniciar un nuevo descubrimiento de ruta hacia el destino.

Network Simulator (ns-2)

Es un simulador de eventos discretos disponible para Linux y Windows (la versión de Windows

no es recomendada). ns-2 fue ideado inicialmente para redes fijas y más tarde fue incorporado al modelo inalámbrico por el grupo Monarch (Mobile Networking Architectures) (Fall, 2008). Esta herramienta de simulación es muy potente, ya que cuenta con dos lenguajes de programación, C++ y OTCL, que interactúan para lograr un mejor funcionamiento (Bonastre y Salvador, 2004, p. 267). El lenguaje OTCL (Object Tool Command Language) es usado para la configuración del simulador, especificando los distintos elementos de la red y sus comportamientos mediante el uso de scripts (código de programación), que requiere de un alto nivel de detalle en los escenarios a simular. Por otro lado, C++ es usado para la implementación de los protocolos, que requiere de un lenguaje de programación capaz de manipular bytes y encabezados de paquetes eficientemente. La distribución posee Licencia Pública General (GPL), la cual es creada por la fundación de software gratuito (Free Software Foundation) encargada de proteger su libre distribución, modificación y uso, siendo ésta la principal razón para su éxito. Existen muchos simuladores para aplicaciones de redes móviles: ns-2, QualNet, Opnet, Matlab, etcétera. Una investigación reciente (Kurkowski, Camp y Colagrosso, 2005) demostró que el resultado del uso del simulador ns-2 durante los últimos años fue del 44,4% comparado con el resto.

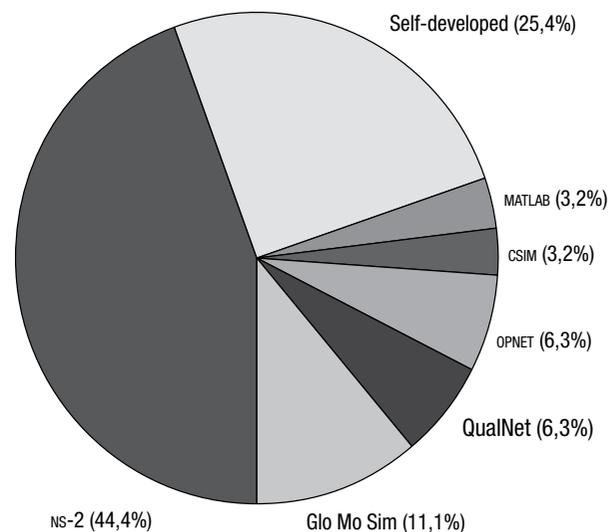


Figura 3. Comparación simuladore

Fuente: Kurkowski, Camp y Colagrosso, 2005

El ns-2 se divide en dos partes principales, la primera es el simulador ns en sí y la segunda es el NAM (Network Animator), que permite visualizar el comportamiento de la red durante la simulación y mediante su interfaz gráfica se pueden crear y editar los modelos de simulación sin tener que recorrer a los scripts TCL para correr las animaciones. NS-2 cuenta también con la herramienta Xgraph usada para generar gráficos bidimensionales y análisis de los resultados. La visualización de los resultados se ha hecho mediante el uso de GNUPLOT (Broker, 2004).

Simulaciones

La simulación de los escenarios se ejecutará en el ns-2 en su versión sq33. Se hará un análisis de dos diferentes escenarios con los diferentes protocolos de encaminamiento implementados en el ns-2 mencionado anteriormente, para obtener el comportamiento de cada uno en cuanto al *throughput* (eficiencia) y otros parámetros tales como la pérdida y entrega de paquetes.

Primer escenario

En el primer escenario, como se muestra en la figura 3, la red está compuesta por 4 nodos: el (0) establecerá comunicación FTP (File Transfer Protocol) sobre TCP con el (1) primeramente, a través de los nodos intermedios (3 saltos); luego, con el movimiento del nodo destino a una velocidad de 2 m/s, usando el comando *setdest* del ns, se establecerá una comunicación de dos saltos y finalmente se logre establecer una comunicación directa. La figura 4 muestra la cantidad de paquetes recibidos mediante la ejecución de los comandos *grep*, *cat* y *wc* de la siguiente manera:

```
cat adhoc.tr | grep ^r | grep AGT | grep tcp | wc -l
```

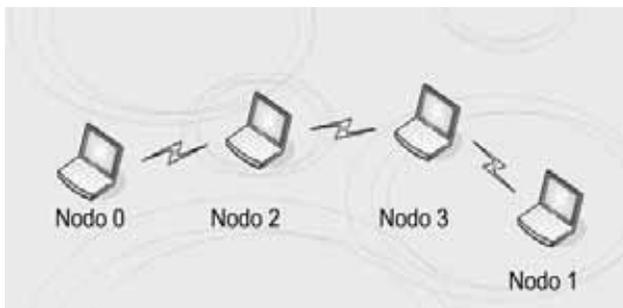


Figura 4. Primer escenario

Fuente: los autores

Esta información se puede extraer similarmente a través del uso del comando *grep* solamente:

```
grep "^r.*\-Nl AGT.*\-It tcp.*" adhoc.tr
```

El fichero *adhoc.tr* es el de traza que se genera por el ns.

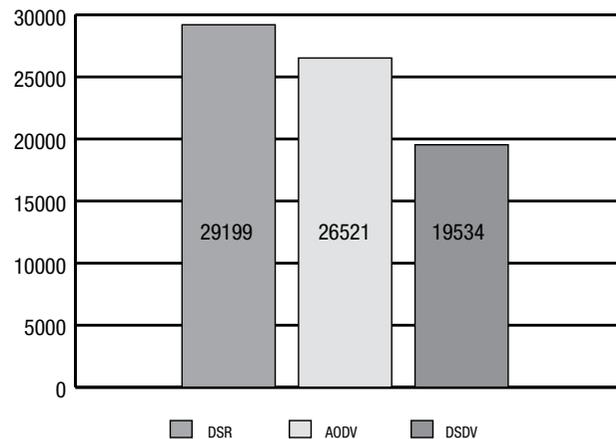


Figura 5. Número de paquetes recibidos

Fuente: los autores

Los mejores resultados son ofrecidos por el protocolo DSR ya que, como veremos posteriormente en el análisis del *throughput*, logra establecer una comunicación directa más rápida debido al uso de la ruta más corta basándose en el número de salto.

El protocolo DSDV entrega los peores resultados comparados con los protocolos reactivos, debido a que se realiza una estimación de tiempo de establecimiento de ruta que demora el envío de los mensajes de actualización.

El *throughput* se puede definir como la velocidad de transmisión de los datos (la cantidad de información útil recibida por el destino final). Es uno de los parámetros que mide la eficiencia de un protocolo. La figura 5 muestra el *throughput* (Mb/s) del escenario 1 para los diferentes protocolos de encaminamientos. Como se puede apreciar, los protocolos reactivos entregan mejores resultados en comparación con el encaminamiento proactivo. Esto se puede notar en el tramo en que los protocolos logran una comunicación de dos saltos, en la que el proactivo alcanza una baja velocidad de transmisión (1,2Mb/s) debido a que los nodos intercambian sus tablas de encaminamiento e inun-

dan la red con paquetes de control. Dentro de los reactivos, el protocolo DSR se destaca ya que, como se mencionó anteriormente, establece la comunicación directa más rápido mediante el uso del modo promiscuo de la capa física para hacer escucha en la red, y almacena esta información para luego tomar la ruta más eficiente (sin usar los mensajes de detección de vecinos activos). El protocolo AODV bajo demanda busca la ruta únicamente cuando es solicitada. DSDV realiza estimación de los tiempos de establecimiento de rutas, retrasando así el envío de mensajes incrementales; es además afectado por la distorsión que se manifiesta en la figura 5 debido a la aparición y desaparición del nodo 3.

Segundo escenario

En el segundo escenario, mostrado en la figura 6, hay una red Ad Hoc formada por 5 nodos, en la cual el 0 (nodo fuente) es el único que está en movimiento, alterando la topología de la red, con una velocidad de 5m/s. Primero, establece una comunicación directa, que luego pierde, para luego, a través de los nodos intermedios, establecer comunicación de 2 y 3 saltos consecutivamente.

En el análisis del *throughput* del segundo escenario mostrado en la figura 7, los nodos logran una comunicación directa al mismo tiempo, ya que los protocolos de encaminamiento no influyen en el intercambio de información (1 salto). DSDV no logra establecer comunicación debido al movimiento del nodo 1 y la demora de los mensajes de actualización de rutas.

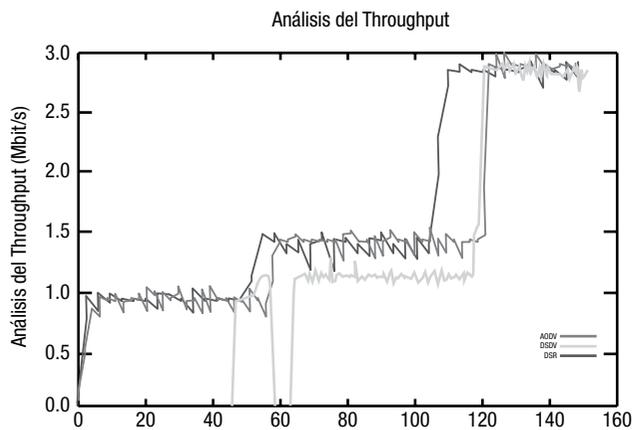


Figura 6. Throughput (Mb/s)

Fuente: los autores

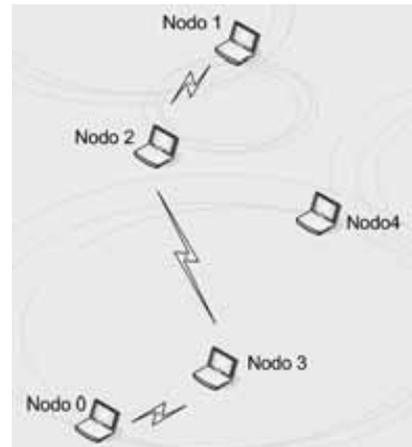


Figura 7. Segundo escenario

Fuente: los autores

La pérdida de paquetes en redes Ad-Hoc puede ser causada por razones de encaminamiento (no existir una ruta válida hacia un destino) o cuando el tamaño del buffer de la cola está lleno y no puede recibir más paquetes. Mediante el uso del lenguaje de programación *mawk* se ha hecho un análisis para determinar la cantidad total de paquetes caídos y la razón que causa tal evento. Usando dicho lenguaje sería fácil la manipulación de datos de los archivos de trazas generados por el simulador NS-2.

Fórmula para el cálculo de la cantidad de paquetes perdidos:

```
$ mawk '($1=="d" && $19=="RTR") {printf ("%s\n", a++)}' adhoc.tr
$ mawk '($1=="d" && $19=="IFQ") {printf ("%s\n", a++)}' adhoc.tr
```

Esto significa que si el valor del campo 1 (\$1) es d (drop) y el valor del campo 19 (\$19) es RTR o IFQ (tipo de pérdida), aumenta el contador en uno. Esto se aplicará a todas las líneas de los ficheros de trazas *adhoc.tr* y *adhocd.tr* para AODV y DSDV respectivamente.

La figura 8 muestra la pérdida de paquetes del protocolo DSDV ya sea por razón de encaminamiento o por la cola (IFQ) y el instante de ocurrencia de éste.

AODV: Paquetes perdidos tipo RTR: 28

Paquetes perdidos tipo IFQ: 18

DSDV: Paquetes perdidos tipo RTR: 30

Paquetes perdidos tipo IFQ: 28

La pérdida de paquetes del protocolo AODV por causa del encaminamiento RTR ocurre cuando los nodos pierden la conexión de 3 saltos y 2 saltos respectivamente e inundan la red con paquetes de control para establecer comunicación. La pérdida de paquetes por razones de la cola (IFQ) ocurre en el instante que termine la simulación (todos los paquetes que estén guardados en el buffer de salida de la cola son descartados al finalizar la simulación). Para el protocolo DSDV la pérdida (Lost_RTR_DSDV), mostrada en la figura 8 ocurre debido a un cambio de topología de 3 saltos buscando el camino más corto. Luego, en los instantes en que el nodo 3 sale de la red completamente y vuelve a entrar en la área de cobertura debido al movimiento del nodo 2, causa esas pérdidas de información de control que se intercambian periódicamente entre los nodos vecinos. Las pérdidas causadas por razones de IFQ en DSDV (Lost_IFQ_DSDV) son mayores que en el caso del protocolo AODV ya que aquí existe una mayor carga en la red, como la del ARP (Address Resolution Protocol). La última tiene que buscar la dirección MAC del destino antes de enviar paquetes; en caso de que el ARP esté lleno, el paquete es descartado. Otras pérdidas son debidas a que el tamaño del buffer está lleno (IFQ full), que normalmente ocurre cuando existe sobrecarga en la red y finalmente las pérdidas de fin de simulación explicada anteriormente.

Análisis del Throughput

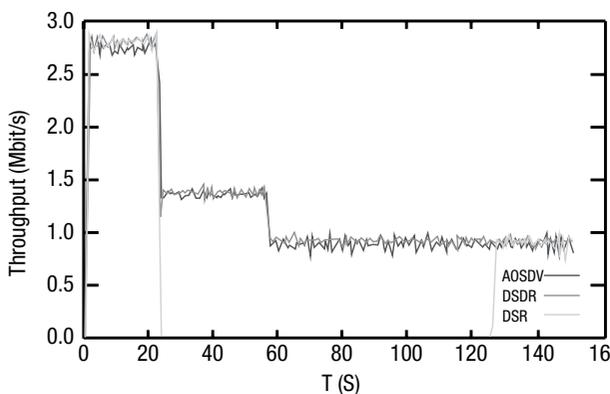


Figura 8. Throughput (Mb/s)

Fuente: los autores

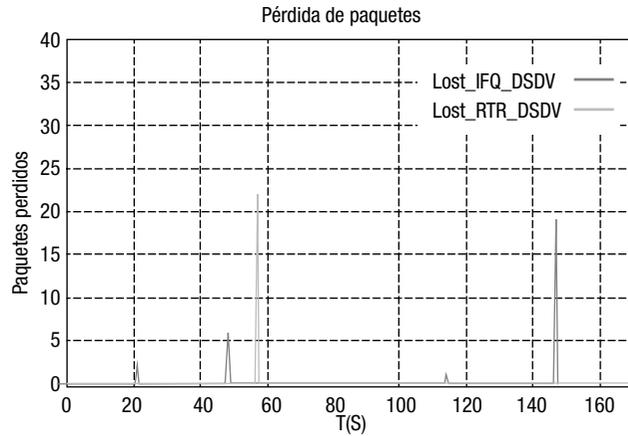


Figura 9. Pérdida de paquetes (dsv)

Fuente: los autores

Conclusiones

El protocolo DSDV ofrece los peores resultados ya que el tiempo de establecimiento de ruta es muy grande y se ve afectado por la conexión y desconexión de los nodos no activos, ya que este cambio se debe anunciar por la red para actualizar su tabla de encaminamiento. Dicho protocolo no es eficiente en escenarios en los que haya mucha movilidad. La sobrecarga introducida en la red por las actualizaciones de ruta limita el ancho de banda, perturbando así la eficiencia del protocolo y aumenta las pérdidas de paquetes.

Los resultados obtenidos en cuanto al *throughput*, entrega y pérdidas de paquetes, reafirma que los protocolos reactivos son más eficientes en este tipo de red y, en específico, el protocolo DSR, que entrega mejores resultados dentro de los protocolos reactivos.

Referencias

- Abolhasan M., Wysocki, T. A. y Dutkiewicz, E. (2004, enero), "A review of routing protocols for mobile ad hoc networks. Ad Hoc Networks", en Elsevier, vol. 2, núm. 1, pp. 1-22.
- Alsaadi, M. (2006), *Simulation Study for a Secure Routing Protocol for Wireless Mobile Ad Hoc Network*, [tesis de maestría], Mayagües, University of Puerto Rico, Maestría en Ingeniería Computacional.

- Azadm, S., Rahman, A. y Anwar, F. (2007), "A Performance of Proactive and Reactive Routing Protocols of Mobile Ad-hoc Network (MANET)". *J. Eng. Applied Science*, vol. 2, núm. 5, pp. 891-896.
- Bonastre, O. y Salvador, C. (2004), *Introducción a la programación de protocolos de comunicaciones con Network Simulator 2*, San Vicente, Editorial Club Universitario.
- Chiu, C. (2006), *Wireless Traceback in Dynamic Source Routing*, [tesis de maestría], China, National Chiao Tung University, Maestría en Ingeniería Computacional.
- Fall, K. y Varadhan, K. (Sep. 2008), "The ns Manual".
- Frodigh, M. y Johansson, P. (2000), "Formación de redes inalámbricas ad hoc: el arte de la formación de redes sin red", *Ericsson Review*, núm. 4, pp. 248-263.
- Goitia, M. J. (2004), *Protocolos de enrutamiento para la capa de red en arquitecturas de redes de datos*, Argentina, Universidad Nacional de Nordeste, Departamento Informática.
- Hernández, G. (2006), *Evaluación de criterios de elección de pasarelas en MANET híbridadas*, Proyecto, Málaga, Universidad de Málaga, Escuela Técnica Superior de Ingeniería de Telecomunicación.
- Kumar, A., Reddy, L. y Hiremath, P. (2008), "Performance Comparison of Wireless Mobile Ad-Hoc Network Routing Protocols", en *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, núm. 6, pp. 337-343.
- Kurkowski, S., Camp, T. y Colagrosso, M. (2005), "MANET Simulation Studies: The Current State and New Simulation Tools", Technical Report CSM-MCS-05-02, Colorado, Department of Math and Computer Sciences, Colorado School of Mines, Golden.
- Kurosawa, S. y Nakayama, H. (2007), "Detecting Blackhole Attack on AODV-based Mobile Ad Hoc Networks by Dynamic Learning Method", *International Journal of Network Security*, vol. 5, núm. 3, pp. 338-346.
- León, M. y Barrera, H. (2007), "Análisis del desempeño del protocolo de enrutamiento DSR bajo diferentes modelos de movilidad", *Épsilon*, núm. 8, pp. 111-123.
- Ovando, M. (2008), *Arquitectura de descubrimiento de servicios en MANET basada en dispositivos de capacidades superiores liderando clusters* [tesis doctoral], Murcia, Universidad de Murcia, Doctorado en Ingeniería de la Información y las Comunicaciones.
- Pérez, M. (2006), *Protocolos de encaminamiento en origen con identificadores no únicos para redes Ad-Hoc de dispositivos con recursos limitados* [tesis doctoral], Madrid, Universidad Rey Juan Carlos, Doctorado en Ingeniería Telemática y Tecnología Electrónica.
- Perkins C., Belding-Royer, E. y Das, S. (julio 2003), "Ad Hoc On-Demand Distance Vector (AODV) Routing", *Internet Engineering Task Force (IETF)*.
- Ramírez, J. (junio 2006), "Optimización de los parámetros del protocolo de encaminamiento AODV", en *ETSI de Telecomunicación*, Málaga, Campus de Teatinos.
- Solórzano, F. (2008), *Análisis de sistemas de videoconferencia en tiempo real sobre redes inalámbricas móviles Ad Hoc 802.11*, Proyecto, Quito, Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica.
- Urra, D. (2006), *Comparativa de implementación de protocolos reactivos de encaminamiento en redes Ad-Hoc*, Proyecto, Cataluña, Universidad Politécnica de Cataluña.
- Wang, W. y Lu, Y. (2004), "On Vulnerability and Protection of Ad Hoc On-Demand Distance Vector Protocol" [conferencia], 10th IEEE International Conference on Telecommunication (ICT).