

Comparación de protocolos de ruteo en redes ad-hoc

Mg. Matías Robles , Ing. Luis Marrone , Lic. Javier Díaz , Lic. Andrés Barbieri

Laboratorio en Nuevas Tecnologías Informáticas (LINTI), Facultad de Informática,
Universidad Nacional de La Plata
{jdiaz,barbieri}@cespi.unlp.edu.ar
{lmarrone}@linti.info.unlp.edu.ar
{mrobles}@info.unlp.edu.ar
<http://www.linti.info.unlp.edu.ar>

Resumen *Una red ad-hoc es una colección de nodos wireless que forman de manera dinámica una red temporaria sin la necesidad de tener una administración centralizada. Una de sus particularidades es que los nodos que la componen pueden entrar y salir de la red en cualquier momento sin previo aviso pero, además, deben cumplir con funciones de ruteo. Como los protocolos de ruteo tradicionales no pueden adaptarse correctamente a esta clase de redes se desarrollaron protocolos de ruteo específicos. En este trabajo se compara el funcionamiento y la performance de tres de los protocolos más utilizados: DSDV, DSR y AODV. Estas comparaciones se realizan utilizando el simulador NS-2.*

Keywords: Ruteo, Ad-Hoc, NS-2 DSDV, AODV, DSR

1. Introducción

Una de las principales ventajas de las redes wireless es la de permitir la movilidad de los usuarios mientras siguen conectados a la red. Sin embargo, en las redes infraestructura, esa movilidad está limitada por el área de cobertura de la estación base o access point. En determinadas situaciones, esta área puede ser muy pequeña o, simplemente, podría no existir una estación base, lo cual, no permitirá tener comunicación entre los nodos. Además, existen distintos escenarios donde podría ser necesario desplegar una red en forma rápida, como pueden ser cuando existen catástrofes naturales, operaciones de rescate, guerras. etc. Las redes ad-hoc [10] son muy atractivas para resolver estas situaciones porque se pueden formar dinámicamente mediante la colaboración de todos los nodos que la integran. No hay necesidad de que exista un acuerdo previo en lo que respecta al rol específico que cada nodo debe asumir, es descentralizada. Toda la actividad de la red, incluyendo el descubrimiento de la topología de la misma y el envío de mensajes, es realizada por los nodos.

Este tipo de redes tienen una topología dinámica y arbitraria que puede cambiar rápidamente y de manera impredecible debido a que los nodos son libres de

moverse aleatoriamente. La principal consecuencia de esta movilidad es que los links pueden formarse y romperse con mucha frecuencia, lo que implica que la red debe ser auto-configurable y auto-organizada. Obviamente, el camino entre un origen y un determinado destino que atraviesa varios nodos intermedios puede, repentinamente, modificarse. Cuando esto sucede, la red debe ser capaz de encontrar un nuevo camino en el menor tiempo posible y todos los nodos que la componen deben cooperar para lograrlo. Esta funcionalidad requiere que cada uno de los nodos sea capaz de reenviar datos en nombre de otros nodos, es decir, que funcionen como un router.

En una red wireless del tipo infraestructura, la comunicación entre los nodos que la componen es single-hop. Todos los nodos se comunican, directamente, con la estación base. Pero, en una red ad-hoc, puede suceder que dos nodos cualquiera no se puedan conectar, directamente, entre ellos, aunque si lo pueden hacer utilizando nodos intermedios, es decir, la comunicación entre los nodos es multi-hop. Es por esta característica que se requiere la presencia de un protocolo de ruteo que permita encontrar las posibles rutas entre un origen y un destino. Este protocolo debe ser ejecutado en todos los nodos que conforman la red. Debido a los limitados recursos con que cuenta una red ad-hoc, el diseño de protocolos de ruteo eficientes y confiables se ha transformado en un gran desafío.

En este trabajo se compara el funcionamiento de 3 (tres) de los más importantes protocolos de ruteo que existen en la actualidad y que han sido estudiados entre otros en [9][11]. Realizando distintas simulaciones se analizará como es el desempeño de cada uno de ellos con el fin de descubrir sus ventajas y desventajas. Los protocolos elegidos son: Destination Sequence Distance Vector(DSDV) [1], Dynamic Source Routing(DSR) [6][2] y Ad-hoc On-demand Distance Vector(AODV) [4].

2. Protocolos de ruteo ad-hoc

2.1. Clasificación de los protocolos de ruteo

Como es sabido, en las redes “estáticas” tradicionales, los algoritmos de ruteo se pueden clasificar en dos grupos: link-state y distance-vector. Estos tienen un correcto desempeño en ese tipo de redes, pero no pueden escalar para grandes redes de tipo ad-hoc y no son capaces de adecuarse a los constantes cambios topológicos que existen en estas. Para solucionar estos inconvenientes se les han realizado adaptaciones, como también se han desarrollado nuevos protocolos específicos para esta clase de redes.

La gran cantidad de protocolos de ruteo para redes ad-hoc (más de 30 protocolos [7]) existentes se los puede clasificar de la siguiente forma: planos (flat), jerárquicos (hierarchical) y de posición geográficamente asistidos (Geographic position assisted)[3][5]. En el ruteo plano, o ruteo uniforme, todos los nodos son

idénticos en lo que a su rol y responsabilidad, dentro de la red, se refiere. En cambio, en el ruteo jerárquico, existen nodos que tienen una responsabilidad diferente en el funcionamiento del protocolo de ruteo. Por último, en los protocolos de posición geográfica asistida, los nodos pueden ser ayudados por algún dispositivo especial, como puede ser un dispositivo GPS.

Dentro del ruteo plano, los protocolos se pueden dividir en tres grupos según la forma en que los nodos obtienen y mantienen la información de ruteo: proactivos (table-driven), reactivos (on-demand) e híbridos. Los protocolos proactivos intentan mantener la información de ruteo a todos los demás nodos de la red consistente y actualizada, por lo tanto, si un nodo necesita una ruta la obtiene inmediatamente. Como mantienen la información en tablas, también se los conoce como table-driven. Estas tablas son enviadas periódicamente a todos sus vecinos. Aun cuando no sean utilizadas, las rutas a todos los destinos posibles se obtienen y mantienen en cada uno de los nodos. Por su parte, en un protocolo reactivo, las rutas son buscadas, únicamente, cuando se necesitan. El proceso de descubrimiento de ruta es iniciado por el origen y, si se encuentra una ruta hacia el destino deseado, ésta se mantiene hasta que el destino deja de ser accesible. Los protocolos proactivos tienen la ventaja de un menor retardo end-to-end porque las rutas deberán estar disponibles al momento de querer establecer una conexión pero, tienen como desventaja, que generan una gran cantidad de tráfico para mantener las tablas actualizadas, por lo que, pueden presentar problemas de escalabilidad en redes grandes. Por último, los protocolos híbridos combinan los méritos de estos dos tipos de protocolos y, además, solucionan sus limitaciones.

DSDV - Destination Sequenced Distance Vector DSDV es un protocolo de ruteo proactivo, table-driven, basado en el algoritmo de Bellman-Ford. Fue desarrollado en 1994. Su principal ventaja es que soluciona el problema de los loops de ruteo al agregar, a cada entrada en la tabla de ruteo, un valor conocido como "destination sequence number". Este valor es utilizado para distinguir las rutas viejas de las nuevas y así evitar la formación de loops. Cada nodo en la red mantiene una tabla de ruteo en la cual están registrados todos los posibles destinos dentro de la red y la cantidad de saltos para alcanzar cada uno de esos destinos.

Las actualizaciones de las rutas se pueden hacer de dos maneras: time-driven o event-driven. Un nodo, cada cierto tiempo, transmite las actualizaciones, que contienen toda su tabla de ruteo, a sus vecinos. También, se envían actualizaciones cada vez que se produce un cambio significativo en su tabla de ruteo. DSDV puede enviar las actualizaciones de dos formas diferentes: "full-dump", se envía la tabla de ruteo completa, o "incremental", únicamente se envían las rutas cuya métrica ha cambiado desde que se envió la última actualización. Todas las actualizaciones son de tipo broadcast sin importar el método empleado.

Comparación de protocolos de ruteo en redes ad-hoc

Los paquetes de datos son enviados utilizando las tablas de ruteo existentes en cada nodo. Estas tablas contienen la lista de todos los destinos posibles y la cantidad de saltos a cada uno de ellos. Además, cada entrada tiene asociado el correspondiente “destination sequence number”, el cual es generado por el nodo destino.

Un nodo puede recibir distintas actualizaciones desde diferentes vecinos. Las rutas recibidas las debe comparar con las que tiene en su tabla de ruteo. Si alguna ruta recibida no existe en su tabla de ruteo, la debe agregar. Pero, si existe, el valor recibido del campo “destination sequence number”, que está asociado a un destino en particular, se debe comparar con el existente en la tabla de ruteo para el mismo destino. La ruta con el valor más nuevo será utilizada y las que tienen un valor más viejo son descartadas. Si ese valor es el mismo, la ruta con mejor métrica será la elegida. Las métricas de las rutas recibidas y que son elegidas para agregar a la tabla de ruteo serán incrementadas en uno.

AODV - Ad-Hoc On-Demand Distance Vector La RFC 3561 [4] define este protocolo. Su funcionamiento es bajo demanda, es decir, los nodos, solamente, tienen las rutas que están usando o que han usado recientemente. Provee rutas libres de loops, aun mientras se están reparando links rotos, al utilizar un contador llamado “destination sequence number”. Este contador está asociado a cada entrada en la tabla de ruteo y está determinado por el nodo destino. La métrica utilizada para comparar dos rutas es el “Hop Count” (Cuenta de Saltos).

Para encontrar una ruta entre dos nodos cualquiera de la red, AODV utiliza un proceso de descubrimiento de ruta broadcast. Define tres tipos de mensajes: Route Request (RREQ), Route Reply (RREP) y Route Error (REER), que son enviados utilizando el protocolo UDP. Cuando un nodo quiere establecer una conexión con un destino en particular, primero debe verificar que no tiene una ruta válida a ese destino. Si existe una, la utiliza y AODV no hace nada. En caso contrario, debe iniciar el proceso de “Path Discovery”. Para esto, envía un mensaje broadcast RREQ a sus vecinos. Cada uno de los vecinos que reciba este mensaje puede tomar una de las dos acciones: si tiene una ruta válida hacia ese destino, le contesta al origen con un RREP, y, si no, debe reenviar el RREQ, también en forma broadcast, después de incrementar el Hop Count. Para saber si una ruta hacia un destino es válida (o “fresca”), el nodo debe verificar que el valor del campo “destination sequence number”, asociado a la entrada en la tabla de ruteo, es mayor o igual al valor de ese campo en el RREQ. La respuesta, sea de parte del nodo destino o de un nodo intermedio, es mediante un mensaje RREP de tipo unicast.

Cada nodo que reenvía un RREQ debe almacenar, entre otra información, la dirección del nodo del cual recibió el pedido. Al hacer esto, automáticamente, se va estableciendo el reverse-path hacia el origen. De esta forma, cuando un nodo recibe el correspondiente RREP para esa ruta, sabe a que vecino se lo debe

enviar en forma unicast. Además del reverse-path, se establece el forward-path. Mientras el RREP viaja de regreso al origen, cada nodo por donde pasa este mensaje, debe almacenar la dirección del vecino del cual lo recibió. Un nodo puede recibir un mismo RREP más de una vez. Sólo debe retransmitir el primero que recibió. Los demás los debe descartar, salvo que el “destination sequence number” del RREP recibido sea mayor que el enviado anteriormente o, si son iguales, que tengo una mejor métrica. El origen puede comenzar a transmitir los datos tan pronto como reciba el primer RREP y puede, luego, actualizar su información de ruteo si aprende una mejor ruta.

Existen varios timers que se utilizan para ir limpiando las entradas del reverse-path o del forward-path. También existen timers para eliminar de la tabla de ruteo todas aquellas rutas que no hayan sido utilizadas en el último periodo de tiempo. La ruta se considera inválida y se elimina. Si un nodo detecta que un link de una ruta activa se rompe, envía un mensaje REER a todos los vecinos que utilizan ese link en forma activa.

DSR - Dynamic Source Routing DSR es un protocolo de ruteo reactivo que, a diferencia de AODV, utiliza source-routing. Para poder realizar este tipo de ruteo, un nodo debe ser capaz de obtener la dirección de todos los dispositivos intermedios entre el origen y el destino. Cada vez que un nodo envía un paquete, el listado ordenado de todos los nodos intermedios por los que debe pasar el paquete es incluido en la cabecera de dicho paquete. Esto produce un gran overhead en el rendimiento de la red pero, tiene como ventaja, que no se producen loops. Además, otra ventaja es que los nodos intermedios no necesitan mantener las tablas de ruteo actualizadas porque toda la información de ruteo ya se encuentra en el nodo origen.

El protocolo consta de dos mecanismos que trabajan en conjunto para descubrir y mantener las rutas: Route Discovery y Route Maintenance. Route Discovery es el proceso encargado de obtener todos los nodos intermedios por donde deben pasar los paquetes y Route Maintenance es el mecanismo, por el cual, el nodo origen detecta que la ruta hacia un determinado destino no es más utilizable debido a cambios topológicos de la red. Ambos procesos operan, enteramente, bajo demanda. DSR no utiliza ningún tipo de mensajes periódicos ni detección del estado del enlace. De acuerdo a la terminología utilizada en DSR, el nodo origen se llama *initiator* y el nodo destino, *target*.

Cuando el initiator quiere enviar un mensaje a un target particular, primero busca en su “route cache” si tiene alguna ruta, previamente aprendida, hacia ese destino. Si la tiene, la utiliza. En caso contrario, debe ejecutar el proceso Router Discovery para encontrar una ruta. Para iniciar este proceso, el initiator transmite un mensaje Route Request (RREQ) de tipo broadcast, el cual es recibido por todos los vecinos dentro del área de cobertura de su señal wireless. Cada RREQ identifica el initiator y el target de la ruta que se intenta encontrar y

contiene un identificador único del pedido, *request ID*, establecido por el initiator. Además, incluye la lista de las direcciones de todos los nodos por los cuales esta copia, en particular, ha ido pasando. Esta lista es inicializada como una lista vacía por el initiator. El request ID es usado por los nodos para determinar si un determinado mensaje, del mismo initiator, es recibido más de una vez.

Si el nodo al que arriba el RREQ no es el target y es la primera vez que recibe esa solicitud, debe agregar su dirección a la lista de nodos por los que pasó el mensaje y retransmitirlo. En caso de que el mismo RREQ sea recibido más de una vez, sólo se aceptará el primer mensaje y todos los demás se descartarán. Un pedido, también, será descartado si la dirección del nodo que recibió el mensaje se encuentra en la lista de nodos por los que ya ha pasado el RREQ (es decir, el nodo ya recibió y reenvió ese mismo RREQ).

Al arribar RREQ al target, éste le contestará al initiator enviándole un Route Reply (RREP). En este mensaje, el target incluirá una copia de la lista de nodos por los que pasó el pedido. Cuando el initiator recibe la contestación, copia la lista de nodos en su cache para utilizarla en el envío de los siguientes mensajes al target. Obviamente, si el target no tiene una ruta en su cache al initiator para enviarle el RREP, no inicia su propio proceso de Route Discovery, lo que podría conducir a un lazo infinito, sino que utiliza la secuencia de nodos, invertida, que copió del RREQ y está informando al initiator.

Si un paquete es retransmitido una cierta cantidad de veces por un nodo y no se recibe ninguna confirmación, el nodo debe retornar un mensaje de error al initiator, indicando el link sobre el cual el mensaje no pudo ser enviado. Si el initiator tiene otra ruta en su cache hacia ese target, la debe usar. En caso contrario, debe ejecutar el Route Discovery nuevamente.

Una ventaja que presenta este protocolo es que permite múltiples rutas a un destino lo que posibilita que cada nodo pueda controlar las rutas que utiliza para rutear los paquetes, por ejemplo para realizar un balanceo de carga. Además, es capaz de trabajar con link unidireccionales. La desventaja que presenta es que el source routing podría resultar en un gran overhead si el path entre dos nodos es muy grande o si se utilizan direcciones del tipo IPv6.

3. Comparación de los protocolos de ruteo

3.1. Diseño de la simulación

Para realizar este trabajo se utilizó el simulador NS-2.34 [8]. La configuración básica consiste en un cuadrado de 900 x 900 metros, en los cuales se disponen, aleatoriamente, un total de 90 nodos. El tráfico que generan los nodos es del tipo CBR (Constant Bit Rate), que corre sobre UDP, con un tamaño de paquete de 512 bytes. Cada nodo envía 4 paquetes por segundo. Los nodos se pueden mover

a una velocidad de hasta 20 metros por segundo con distintos tiempos de pausa (pause time). Los tiempos seleccionados son los siguientes: 0 (los nodos están continuamente en movimiento), 25, 50, 100, 250, 500 y 900 (los nodos no tiene movimiento). En total se realizaron 4 simulaciones distintas donde se varían la cantidad de conexiones establecidas entre los nodos: 3, 15, 25 y 40. Todas las simulaciones se ejecutaron por un total de 900 segundos.

Con el fin de comparar el rendimiento de los 3 protocolos en cada una de las situaciones diagramadas, se evaluaron las siguientes métricas:

Packet Delivery Rate: es el ratio (razón) entre el número de paquetes CBR enviados y el número de paquetes CBR recibidos. Describe el porcentaje de paquetes que alcanzaron el destino. Cuanto mayor sea este porcentaje, mejor es el rendimiento de la red.

Average Delay: es el tiempo, promedio, que tardan los paquetes de datos en llegar desde el origen al destino teniendo en cuenta todos los posibles retardos que sufre el paquete en su camino, como pueden ser: buffering durante el proceso de descubrimiento de ruta, encolamiento en las interfaces, retardos de retransmisión en la capa MAC, propagación y tiempos de transferencia.

Routing Overhead: es el ratio (razón) entre el número de paquetes de ruteo transmitidos y el número de paquetes de datos recibidos en el destino. Cada paquete de ruteo retransmitido por un nodo se cuenta como válido. Permite analizar el número promedio de paquetes de ruteo enviados por cada paquete de datos que llega a destino.

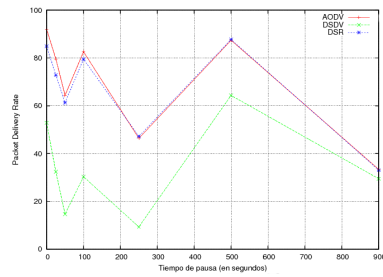
3.2. Packet Delivery Rate

Como se puede observar en los gráficos de la Figura 1, DSDV es el protocolo con peor rendimiento al ser el protocolo en el que se pierden la mayor cantidad de paquetes. De los otros dos protocolos, DSR es el que tiene mejor rendimiento en todo los casos, aunque la diferencia con AODV es muy pequeña en alguna simulaciones. Esto se produce a pesar de que DSR es el protocolo que genera la mayor cantidad de paquetes de ruteo en casi todas las simulaciones. El problema que presenta DSDV es el tiempo que tarda en encontrar las rutas entre el origen y el destino. Una situación interesante se produce en las simulaciones con 3 y 15 conexiones. En estas, el rendimiento de la red es peor cuando menor movilidad presentan los nodos. Por ejemplo, de la 3 conexiones especificadas, se establece una sola. Las dos restantes no pueden encontrar las rutas correspondientes. En las demás simulaciones, al moverse los nodos se establecen las rutas.

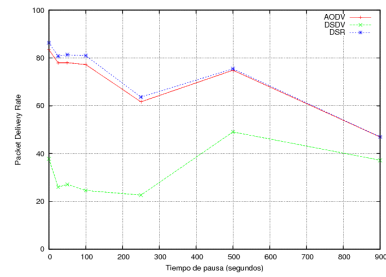
3.3. Average Delay

En los gráficos de la Figura 2, se puede observar el retardo promedio de los paquetes de datos, desde que son enviados por el origen hasta que son recibidos en el destino. DSR es el protocolo que mayor retardo promedio presenta. Esto se debe, principalmente, a la gran cantidad de paquetes de ruteo que necesita

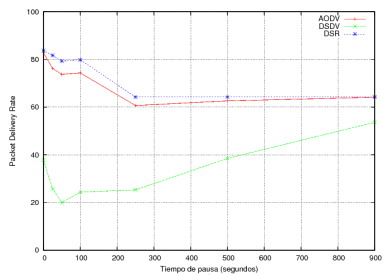
Comparación de protocolos de ruteo en redes ad-hoc



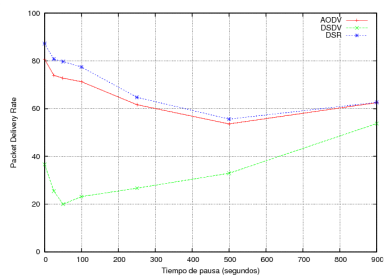
(a) 3 conexiones



(b) 15 conexiones



(c) 25 conexiones



(d) 40 conexiones

Figura 1: Packet Delivery Rate

enviar para encontrar las rutas, lo que se produce con mayor intensidad en los tiempos de pausa 25 y 50. DSDV, al ser el que menor cantidad de paquetes de ruteo genera, es el que mejor retardo promedio presenta. A medida que los nodos presentan menor movilidad, el retardo de los distintos protocolos convergen.

3.4. Routing Overhead

A excepción de la simulación con 3 conexiones, DSDV es el protocolo que mejor relación presenta entre la cantidad de paquetes de ruteo enviados y la cantidad de paquetes de datos recibidos, correctamente, por los destinos según lo que se puede observar en la Figura 3. Esto se debe a que, sin importar la cantidad de conexiones establecidas, la cantidad de paquetes de ruteo no se modifica sustancialmente, algo que si sucede con los otros dos protocolos que al funcionar bajo demanda (on-demanda) deben generar paquetes de ruteo por cada nueva conexión, si es que no tienen la ruta al destino.

4. Conclusiones y Trabajos Relacionados

A pesar de mostrar un menor Average Delay y un mejor Routing Overhead, DSDV es el que peor rendimiento presenta debido al tiempo que tarda en formar

Comparación de protocolos de ruteo en redes ad-hoc

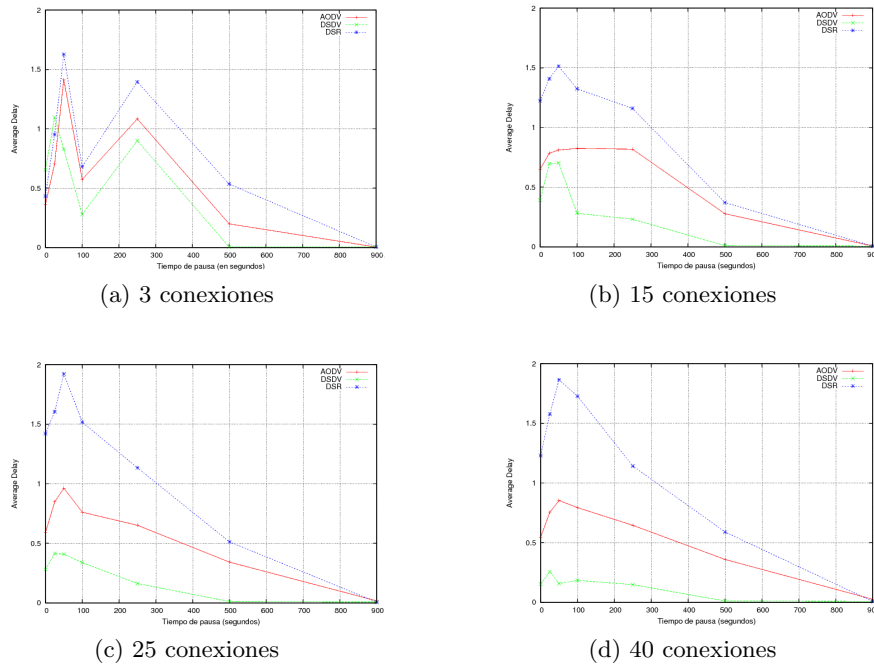


Figura 2: Average Delay

las rutas. Por el contrario, DSR es el que tiene el mejor rendimiento a pesar de ser el que mayor cantidad de paquetes de ruteo envía. Por su parte, AODV presenta un rendimiento más parejo, prácticamente, en todas las simulaciones. La ventaja que presenta DSR con respecto a los otros dos protocolos es que puede tener rutas alternativas en su cache hacia un destino en particular.

Por lo explicado anteriormente, se puede inferir que los dos protocolos de ruteo ad-hoc reactivos estudiados presentan un mejor rendimiento que el protocolo de ruteo proactivo. Sin embargo, presentan un Average Delay más grande que para determinados tipos de tráfico, como voz y/o vídeo, puede ser muy contraproducente. El estudio de como mejorar este punto es crucial para que este tipo de redes tengan una mayor aceptación.

Esta investigación está siendo realizada en el marco del Proyecto TRIComFor, Tecnologías en Redes Inalámbricas para Comunicación y Formación, aprobado y financiado por la CyTED, en conjunto con integrantes del Departamento de Informática de la Universidad de Minho, Portugal.

Comparación de protocolos de ruteo en redes ad-hoc

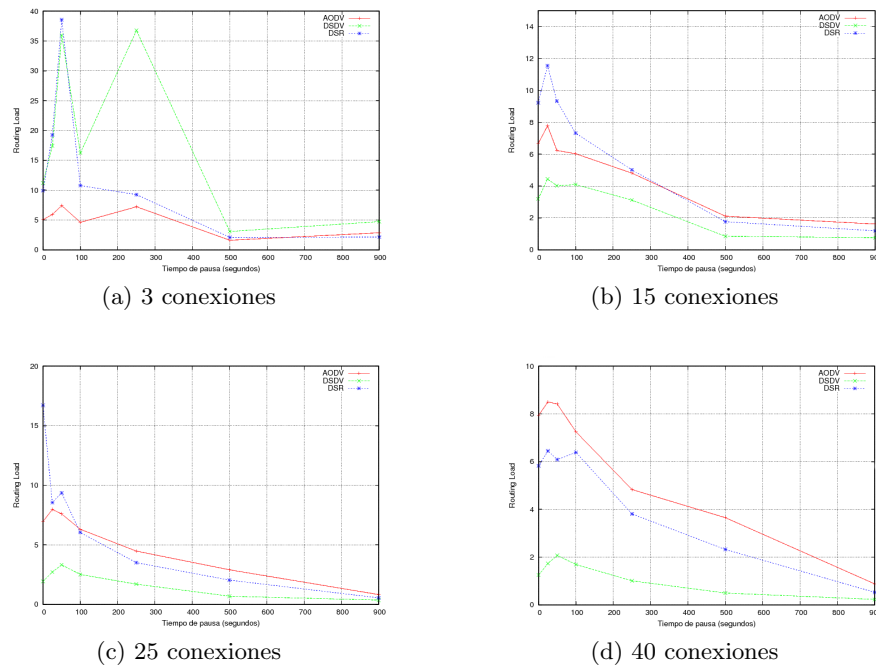


Figura 3: Routing Overhead

Referencias

1. Perkins, Bhagwat : Highly Dynamic Destination-Sequenced Distance-Vector Routing for Mobile Computer. <http://www.cs.virginia.edu/~c17v/cs851-papers/dsdv-sigcomm94.pdf> (1994).
2. Johnson, Maltz, Broch : DSR - The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad-Hoc Networks.
3. Petteri Kuosmanen : Classification of Ad Hoc Routing Protocols <http://keskus.hut.fi/opetus/s38030/k02/Papers/12-Petteri.pdf>.
4. Perkins, Belding-Royer : Ad-hoc On-Demand Distance Vector (AODV) Routing. (2003).
5. Ad-Hoc Networks - Technologies and protocols. Springer (2005).
6. Johnson, Maltz : The Dynamic Source Routing Protocol (DSR) for Mobile Ad-Hoc Networks for IPv4 (2007).
7. Faculté de Science de la Technologie et de la Communication, [http://wiki.unil.secan-lab/Ad-Hoc+Protocols+\(\\$28\)Classification\(\\$29\).html](http://wiki.unil.secan-lab/Ad-Hoc+Protocols+($28)Classification($29).html).
8. The Network Simulator - NS-2 <http://www.isi.edu/nsnam/ns/>.
9. Shah, Khandre, Shirole, Bhole : Performance Evaluation of Ad Hoc Routing Protocols using NS2 Simulation, Mobile and Pervasive Computing, 167–171 (2008).
10. Sarkar, Basavaraju, Puttamadappa : Aurbach - Ad-hoc mobile wireless networks - Principles, protocols and applications. Auerbach Publications (2008).
11. Bikash Rath : Implementing and comparing DSR and DSDV routing protocols for mobile ad-hoc networking, <http://ethesis.nitrkl.ac.in/1131/1/bikas.pdf> (2009).