

Análisis de los protocolos de ruteo OLSR y AODV en redes Ad-Hoc IBSS

Trabajo final de cátedra – año 2011

Autores:

- Gerardo Roman Piermattei Monney – gpiermattei@gmail.com
- Bruno Eduardo Garcia Ferrari – bruno2112@gmail.com

Carrera:

- Ingeniería en Telecomunicaciones

Cátedra:

- Redes de Información – Facultad de Ingeniería – UNRC

Docentes a cargo:

- Ing. Magnago Héctor – hem@uti.unrc.edu.ar
- Ing. Cruz Ivana – icruz@ing.unrc.edu.ar
- Ing. Solivellas Pablo – psolivellas@ing.unrc.edu.ar

Análisis de los protocolos de ruteo OLSR y AODV en redes Ad-Hoc IBSS

Resumen. En una red inalámbrica mallada existen múltiples caminos para llegar a un destino, se puede utilizar un protocolo para optimizar la forma en que los nodos eligen el camino para llegar al destino. En este informe se compara el funcionamiento de dos protocolos de ruteo disponibles para redes inalámbricas malladas que permiten movilidad para sus clientes dentro de la red. Del tipo de los protocolos pro-activos se analiza OLSR y AODV del tipo de los reactivos. Los parámetros a analizar son: uso de hardware, uso de red, velocidad de respuesta.

Palabras claves. OLSR, AODV, Ad-Hoc IBSS, MANET, WMN, Mesh Network.

1 Introducción

Una “Wireless Mesh Network” (WMN) es una red de comunicaciones hecha por nodos inalámbricos organizados en una topología de malla. En estas redes se diferencian: Cliente mesh, Router mesh y Gateways.

Los Clientes mesh suelen ser computadoras tanto fijas como móviles, celulares y cualquier otro dispositivo inalámbricos. Los Router mesh, son los encargados de reenviar el tráfico generado por los clientes hacia los Gateways, estos últimos son los que proveen de conexión a Internet a la red mesh.

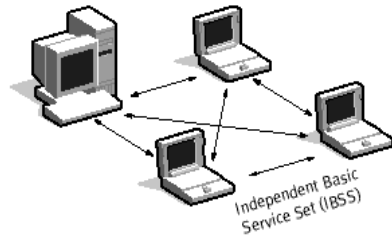
A diferencia de las redes inalámbricas punto-a-multipunto las cuales se calculan con un radio de cobertura que se ve afectado principalmente por la geografía del terreno o por la distribución de edificios, en una red mesh no existe un radio de cobertura como tal y lo que se tiene en realidad es una “nube mesh” formada por los Router mesh donde se les puede brindar servicios a los Clientes mesh. Esta nube por las características de la red mesh puede sobrepasar obstáculos con gran facilidad sin necesidad de usar grandes torres o altas potencias de transmisión.

Una WMN ofrece redundancia en cuanto a los caminos disponibles para llegar a destino, esto es gracias a que un nodo tiene contacto con varios nodos y en caso de que uno falle puede llegar a destino a través de otro.

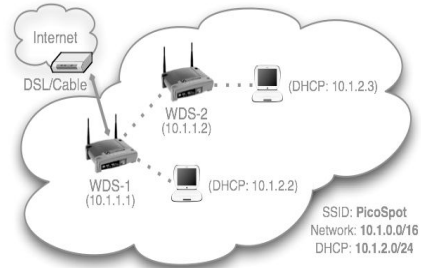
La forma en que los nodos descubren las rutas para alcanzar al destino es el objetivo de investigación de este trabajo.

2 Arquitectura de una WMN

Se distinguen dos arquitecturas bien marcadas, cada una tiene sus ventajas y desventajas:



Ad-Hoc IBSS



Infrastructure ESS

Se mencionan brevemente sus principales características:

<i>Ad-Hoc IBSS</i>	<i>Infrastructure ESS</i>
<p>Red plana, todos los nodos se configuran en modo Ad-Hoc.</p> <p>El mismo <i>Cliente Mesh</i> también actúa como <i>Router Mesh</i>.</p> <p>Se realiza ruteo por IP (capa 3).</p> <p>Costos de instalación nulos.</p> <p>Necesitan de la participación de todos los nodos.</p> <p>El área de servicio depende la ubicación de los clientes, por lo tanto necesita muchos clientes para cubrir un área amplia.</p> <p>El administrador de la red dispone de menos control sobre la misma.</p> <p>Permite movilidad, es transparente para el usuario.</p>	<p>Backbone de APs usando WDS.</p> <p>Los clientes se asocian a un AP en particular.</p> <p>Se realiza ruteo por MAC (capa 2).</p> <p>Costos de instalación medio.</p> <p>Necesitan de una infraestructura de red.</p> <p>Se puede garantizar un área de servicio.</p> <p>El administrador de la red dispone de completo control sobre la red.</p> <p>Permite movilidad.</p>

Según la tabla anterior podemos resumir:

- Ad-Hoc IBSS son ideales para: *Aplicaciones comunitarias*, se basan en la cooperación entre todos los nodos de la red, costos nulos.
- Infrastructure ESS son ideales para: *Aplicaciones comerciales*, bajos costos, rápida implementación y control completo de la red.

En este informe centramos nuestro interés en las redes Ad-Hoc IBSS y principalmente analizaremos dos de los protocolos que controlan estas redes.

3 OLSR: Optimized Link State Routing

Es un protocolo del tipo pro-activo, es decir que cada nodo mantiene actualizada ,*en todo momento*, su tabla de ruteo incluyendo a todos los posibles destinos en la red mesh.

La desventaja de un protocolo pro-activo es que impone una carga adicional en la red debido a la transmisión permanente de mensajes de control. Como ventaja cuando un nodo debe enviar un paquete a un destino siempre estará disponible la información necesaria en su tabla de ruteo para poder hacerlo por lo tanto no introduce demoras para iniciar la comunicación.

La permanente actualización del estado de la red necesita un intercambio de mensajes de control entre los nodos, esto implica un aumento de tráfico y contribuye a la congestión de la red. Para optimizar la difusión de mensajes, OLSR utiliza la técnica de “Retransmisores Multipunto”; cada nodo selecciona un conjunto de nodos vecinos como “Retransmisores Multipunto” (MPR, Multipoint Relays). Estos nodos seleccionados son los encargados de reenviar el tráfico de control destinado a difundir en toda la red.

Cada nodo OLSR mantiene permanentemente rutas para alcanzar a cualquier destino en la red, por esa característica este protocolo se ve favorecido por los patrones de tráfico donde un gran número de nodos se comunica con otro gran número de nodos y donde el par [fuente, destino] cambian en el tiempo.

OLSR está diseñado para trabajar en una red completamente distribuida sin depender de un equipo central. El protocolo no necesita transmisión fiable de los mensajes de control por lo tanto, se transportan sobre UDP/IP utilizando el puerto 698 asignado por la IANA.

Tipos de mensajes:

- Mensajes HELLO: realizan la tarea de sensar el estado del enlace, detección de vecinos y señalización de MPR.
- Mensajes TC: realizan la tarea de declarar la topología de la red publicando estado de los enlaces.
- Mensajes MID: realizan la tarea de declarar la presencia de nodos con múltiples interfaces.

4 AODV: Ad hoc On-Demand Distance Vector

Es un protocolo reactivo, por lo tanto solo genera su tabla de ruteo cuando tiene la necesidad de transmitir un mensaje a un destino particular.

Esto implica que introducen demoras iniciales al momento de calcular la ruta para alcanzar al destino, pero no transmiten constantemente mensajes a la red, disminuyendo los mensajes de control enviados a la red y también el consumo de energía. Recordemos que estos protocolos se aplican a dispositivos móviles donde el consumo energético es importante para la duración de la batería.

Este protocolo está diseñado para redes que contienen desde decenas hasta miles de nodos móviles. AODV puede manejar bajas, medias y relativamente altas tasas de movilidad. Este algoritmo está diseñado para ser utilizado en redes donde los nodos pueden confiar entre cada uno de ellos ya sea porque se han configurado claves o se sabe que no existen nodos maliciosos o intrusos. Ha sido diseñado para reducir los

mensajes de control de tráfico y eliminar el overhead en los paquetes de datos para poder de esta forma mejorar la escalabilidad y aumentar la performance de la red.

Los mensajes definidos por AODV son: pedido de ruta (RREQs), respuesta de ruta (RREP), error de ruta (RERR). Estos mensajes se envían mediante UDP con una cabecera IP normal. Los mensajes de broadcast no inundan por completo la red, sino que solo se dispersan hasta donde el campo TTL de la cabecera IP permite.

Al ser un protocolo reactivo, siempre y cuando cada nodo dentro de una comunicación tenga una ruta valida hacia el otro, AODV no cumple ninguna función. Cuando se necesita una nueva ruta, el nodo envía mensajes RREQ mediante broadcast para encontrar una ruta. La ruta se determina cuando los RREQs llegan al destino o cuando alguno de los nodos ya tenía una ruta hacia dicho destino. De encontrarse un nodo con una ruta activa, esta ruta será válida si el número de secuencia asociado es al menos tan grande como el contenido en el RREQ. La ruta se establece mediante un mensaje RREP en unicast hacia el nodo que origino el RREQ. Cada nodo que recibe el mensaje RREQ guarda una ruta hacia el nodo que lo origino para luego poder retransmitir el RREP.

Cada nodo monitorea el estado de las rutas activas de los saltos próximos. Cuando ocurre un cambio en la topología y un enlace deja de estar activo o está fuera de alcance, la ruta ya no es válida por lo que el nodo que lo detecta debe transmitir un mensaje RERR a los demás nodos involucrados en esa ruta comunicando que la ruta ya no es válida. El mensaje RERR indica los destinos que ya no son alcanzables por el enlace roto. Para poder enviar estos mensajes, cada nodo mantiene una lista de “nodos precursores” conteniendo esta lista cada dirección IP de los nodos que pueden utilizarlo como siguiente salto para alcanzar un destino. Esta lista se crea durante el proceso de creación de la ruta con los RREQ para poder devolver los RREP.

AODV es un protocolo de ruteo, y como tal maneja tablas de ruteo. La información de estas tablas se debe mantener un cierto periodo de tiempo, incluso para rutas de corta duración como son las creadas para retornar los RREPs. En las tablas se guarda la siguiente información:

- Dirección IP de destino
- Numero de secuencia de destino
- Flag de numero de secuencia de destino valida
- Otras flags de estado de ruta (valida, invalida reparable, en reparación, etc)
- Interfaz de red
- Cantidad de saltos (cantidad de saltos hasta llegar al destino)
- Próximo salto
- Lista de precursores
- Tiempo de vida de la ruta

5 Implementación

Los protocolos analizados están pensados para funcionar en muchas clases de hardware y principalmente en dispositivos móviles. Para nuestro análisis se instalaron en computadoras con GNU/Linux “Ubuntu” en las versiones 9.10 de 32 bits y 10.04 de 64 bits con kernel 2.6 o superior.

Para su instalación en Ubuntu se necesitan los siguientes paquetes: make, bison, linux, linux-image, linux-headers, linux-source, build essentials.

Instalación de OLSR

Descargamos el código de fuente:

```
http://www.olsr.org/releases/0.6/olsrd-0.6.0.tar.gz
```

Estando en el directorio “olsrd-0.6.0”, compilamos y luego instalamos

```
make  
make install
```

Después de su instalación es necesario modificar el archivo de configuración:

```
nano /etc/olsrd/olsrd.conf
```

En el archivo solo hay que indicar la red a la que pertenece el host, interfaz que formara parte de la red ah-hoc y implementará OLSR y el modo de “debug” para ver información del protocolo.

Antes de iniciar el protocolo, es necesario unirse a la red Ad-hoc creada sobre la cual actuará OLSR, para esto hacemos:

```
ifconfig wlan0 down  
iwconfig wlan0 mode ad-hoc channel 2 essid “mesh”  
ifconfig wlan0 192.168.10.3 up
```

Es bueno deshabilitar cualquier software de gestor de red que se utilice en la PC para que no nos re-configura la interfaz wlan0.

Luego verificamos la conexión mediante el comando:

```
iwconfig wlan0
```

Todos los nodos en la red deben tener el mismo numero de celda.

Para iniciar el protocolo se utiliza el comando:

```
olsrd -f /etc/olsrd/olsrd.conf
```

El parámetro -f indica la ruta al archivo de configuración

Luego de iniciar el protocolo, si configuramos el “debug” para que muestre la información que maneja el protocolo, vamos a ver en pantalla todo lo aprendido por OLSR.

El resultado final de OLSR se muestra en las tablas de ruteo del sistema operativo:

```
route -n
```

Instalación de AODV

Utilizamos la implementación de la Universidad de Uppsala (Suecia), se puede descargar el código de fuente desde la pagina:

```
http://sourceforge.net/projects/aodvuu/
```

Estando en el directorio “aodv-uu-0.9.6”, compilamos y luego instalamos

```
make  
make install
```

Antes de iniciar el protocolo, es necesario unirse a la red Ad-hoc creada sobre la cual actuará AODV, para esto hacemos:

```
ifconfig wlan0 down  
iwconfig wlan0 mode ad-hoc channel 2 essid “mesh”  
ifconfig wlan0 192.168.10.3 up
```

Es bueno deshabilitar cualquier software de gestor de red que se utilice en la PC para que no nos re-configura la interfaz wlan0.

Luego verificamos la conexión mediante el comando:

```
iwconfig wlan0
```

Todos los nodos en la red deben tener el mismo numero de celda.

Para ejecutar el protocolo utilizamos el comando

```
aodvd -i wlan0 -r /var/log/aodv_route.log -l /var/log/aodv.log
```

Parámetros:

- -i: indica la interfaz
- -r: indica donde se guarda un archivo log con las rutas aprendidas (parámetro opcional)
- -l: indica donde se guarda un archivo log la información aprendida por AODV (parámetro opcional)

El resultado final de AODV se muestra en las tablas de ruteo del sistema operativo:

```
route -n
```

6 Escenario de prueba 1: 6 nodos alineados

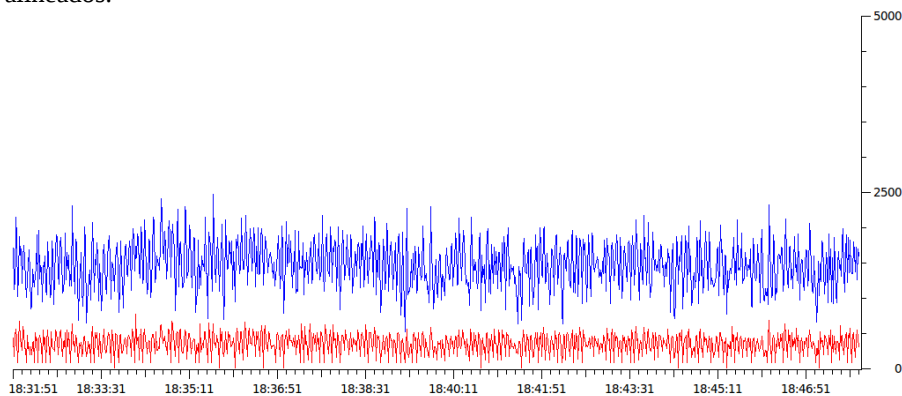
Comportamiento de OLSR:

OLSR envía permanente paquetes a la red. Esto implica uso permanente de la red, mayor uso de CPU, mayor uso de memoria RAM.

En una computadora, el uso de CPU y memoria son inapreciables pero pueden llegar a ser significativos si se corre el protocolo en un dispositivo portátil de hardware más limitado.

Respecto al uso de red, también puede ser despreciado, pero en caso de ser una red donde se tarife según el tráfico inyectado o la red tenga una capacidad baja ya no podría ser despreciado y debería utilizarse una configuración mas “lenta” de OLSR para disminuir la carga de la red.

En la siguiente figura se muestra el tráfico generado por el protocolo con la configuración por defecto que trae el archivo OLSR y con una red de 6 nodos alineados.

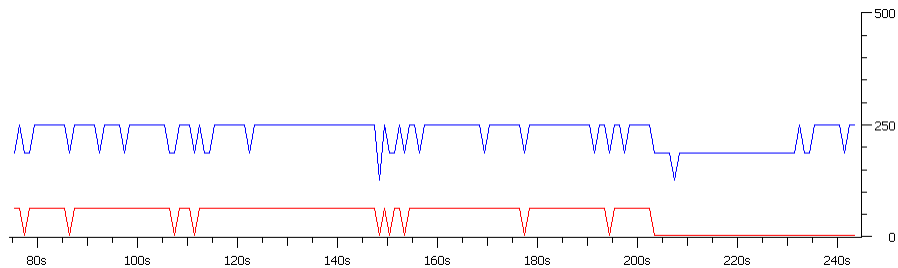


La gráfica indica *Bytes* vs. *Tiempo*, solo incluye los datos que usa OLSR para mantener el estado de la red. En color rojo se muestra el tráfico enviado por el nodo a la red y en color azul se muestra el tráfico total en la red.

Comportamiento de AODV:

Como se describió anteriormente, AODV es un protocolos reactivo, por lo tanto no le

interesa el estado de los nodos en la red hasta que tiene la necesidad de enviar un paquete a alguno de ellos. Este comportamiento le permite lograr un menor uso de la red, menor uso de hardware, en particular menos memoria RAM respecto a OLSR. La siguiente figura muestra el tráfico generado por AODV para mantener la conexión a la red.



La gráfica indica **Bytes vs. Tiempo**, solo incluye los datos que usa AODV para mantener el estado de la red. En color rojo se muestra el tráfico enviado por el nodo a la red y en color azul se muestra el tráfico total en la red.

La desventaja de AODV es que introduce una demora inicial al momento de establecer la comunicación con el destino porque el nodo debe esperar a que AODV termine de descubrir la ruta al destino.

Por ejemplo, para el escenario de 6 nodos alineados, AODV no conoce la ruta para llegar a cualquier otro nodo que no tenga directamente conectado, por lo tanto, para analizar esa demora inicial podemos utilizar el comando “ping” y ver las estadísticas

Desde el nodo 1 ejecutamos el comando:

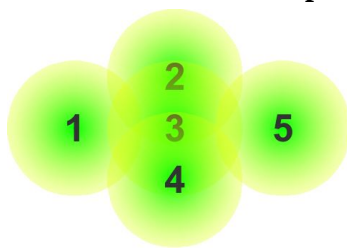
```
ping 192.168.100.6 -i 0.1 -c 10
```

El resultado final del comando es el siguiente:

```
--- 192.168.100.6 ping statistics ---
10 packets transmitted, 5 received, 50% packet loss, time 936ms
rtt min/avg/max/mdev = 21.962/29.061/46.738/9.183 ms
```

Esto muestra la demora inicial en la comunicación hasta que AODV resuelve la ruta para llegar al último de los 6 nodos en línea.

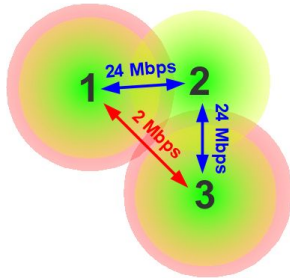
7 Escenario de prueba 2: Múltiples caminos



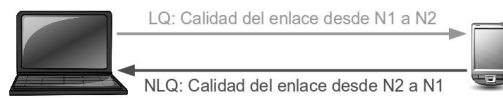
Como primer análisis, lo que se pretende es que el nodo 1 se comunique con el nodo 5 eligiendo el mejor camino de los 3 caminos disponibles.

La segunda parte del análisis es dar de baja el nodo intermedio que el nodo 1 eligió para comunicarse con el nodo 5 y analizar cuánto tiempo tarda en encontrar un nuevo camino para llegar a destino.

Comportamiento de OLSR para seleccionar la ruta:



Un nodo podría alcanzar a su destino a través de 2 saltos con buena intensidad de señal, pero también, podría alcanzar a su destino con 1 salto pero con una intensidad de señal muy mala. Entonces si OLSR elije la menor cantidad de saltos enviaría sus paquetes a la ruta de menor intensidad de señal con lo cual estaría tomando una decisión incorrecta. Para solucionar este problema se incluye información acerca de la calidad del enlace entre los nodos tanto directa como inversa, esto es:



En función a los valores LQ (Link Quality.) y NLQ (Neighbor Link Quality), OLSR calcula un parámetro para cuantificar los valores anteriores y tenerlo en cuenta a la hora de calcular el costo de la ruta, este parámetro es llamado ETX (Expected Transmission Count) estima la cantidad de retransmisiones que un paquete tendría al pasar por esa ruta.

$$ETX = 1 / (LQ * NLQ)$$

Luego para calcular el costo total de una ruta, OLSR suma los ETX de las rutas intermedias hasta llegar a destino y elije la ruta de menor costo total.



Información aprendida por OLSR en el Nodo 5 (Resumida)

```

--- 19:49:41.459522-----LINKS
IP address      hyst  LQ/NLQ      ETX
192.168.100.2  0.000  0.929/0.886  1.214
192.168.100.3  0.000  0.929/0.906  1.188
192.168.100.4  0.000  0.929/0.929  1.158

--- 19:49:41.459596 ----- TWO-HOP NEIGHBORS
IP addr (2-hop) IP addr (1-hop) Total cost
192.168.100.2   192.168.100.3   2.402
                  192.168.100.4   2.376
192.168.100.1   192.168.100.2   2.406
                  192.168.100.3   2.402
                  192.168.100.4   2.376

192.168.100.3   192.168.100.4   2.376
                  192.168.100.2   2.406
192.168.100.4   192.168.100.2   2.456
                  192.168.100.3   2.402

--- 19:49:41.459791 ----- TOPOLOGY
    
```

Source IP addr	Dest IP addr	LQ/NLQ	ETX
192.168.100.1	192.168.100.2	0.929/0.906	1.188
...
192.168.100.3	192.168.100.5	0.886/0.906	1.246
192.168.100.4	192.168.100.1	0.906/0.906	1.219
...
192.168.100.4	192.168.100.3	0.906/0.906	1.219

En la información aprendida por OLSR en el nodo 5 vemos que para llegar al nodo 1 la ruta de menor costo es pasando por el nodo 4 que es justamente el nodo de mayor disponibilidad, podemos verificar esta condición con el comando “traceroute”.

```
traceroute to 192.168.100.1, 30 hops max, 60 bytepackets
 1 (192.168.100.4) 4.692 ms 5.237 ms 6.577 ms
 2 (192.168.100.1) 27.789 ms 33.417 ms 35.497 ms
```

La segunda parte del análisis es dar de baja el nodo 4, mientras hay comunicación entre el nodo 5 y el nodo 1, entonces forzamos a OLSR a encontrar una nueva ruta para llegar al nodo 1 y medir cuando tiempo tarda en encontrar dicha ruta.

Para hacer esta medición, vamos a utilizar el comando “ping” con algunos parámetros adicionales para limitar la cantidad de peticiones enviadas y el tiempo entre las peticiones.

Ejecutamos en el nodo 5 el siguiente comando:

```
ping 192.168.100.1 -i 0.1 -c 200
```

Damos de baja el nodo 4 mientras que los nodos 1 y 5 están simulando una comunicación y el resultado final del comando es:

```
--- 192.168.100.1 ping statistics ---
200 packets transmitted, 86 received, 57% packet loss, time 20808ms
rtt min/avg/max/mdev = 2.820/13.851/145.371/20.093 ms, pipe 2
```

Se realizaron varias mediciones y todas resultaron dentro del mismo orden, OLSR con la configuración por defecto pierde aproximadamente el 50% de los paquetes enviados cada 0,1 segundos.

Lógicamente, se podría configurar los parámetros de OLSR para que responda mas rápido ante cambio en la red, pero esto implicaría mayor uso de la red por intercambio de paquetes de OLSR y mayores usos de recursos de hardware.

Comportamiento de AODV:

Se procede igual que el caso de OLSR, dando de baja el nodo al que el protocolo elige como mejor salto para llegar al destino.

Ejecutamos en el nodo 5 el siguiente comando:

```
ping 192.168.100.1 -i 0.1 -c 200
```

Damos de baja el nodo 4 mientras que los nodos 1 y 5 están simulando una comunicación y el resultado final del comando es:

```
--- 192.168.100.1 ping statistics ---
200 packets transmitted, 173 received, 13% packet loss, time 20198ms
rtt min/avg/max/mdev = 2.627/21.266/116.308/18.466 ms, pipe 2
```

Se realizaron varias mediciones y todas resultaron dentro del mismo orden, AODV con la configuración por defecto pierde aproximadamente el 15% de los paquetes enviados cada 0,1 segundos.

8 Conclusiones

En base a las pruebas realizadas logramos contrastar el comportamiento de AODV y OLSR. Demostramos los puntos fuertes y débiles de ambos protocolos. A continuación resumimos estos puntos.

Como se analizo anteriormente, AODV introduce una demora inicial al momento de establecer la comunicación y en OLSR esta demora inicial no existe, pero AODV es mas rápido frente a OLSR en encontrar una nueva ruta a su destino, por lo tanto AODV logra menor pérdida de paquetes que OLSR durante una comunicación ya establecida. Lógicamente que se pueden configurar los protocolos para que respondan mas rápido a cambios en la red, pero estas configuraciones serán propias para cada aplicación en función de las necesidades. En este trabajo se analizaron las configuraciones por defecto de ambos protocolos.

Otro punto a destacar es el uso de la red de los protocolos para mantenerse conectados. AODV hace una mínima transmisión de paquetes mientras que en OLSR la carga de tráfico a la red puede ser significativa. Por el comportamiento de OLSR, este uso de red, crecería a valores altos si configuramos al protocolo para que detecte mas rápido cambios en la red.

Respecto a los requisitos de hardware, no pudimos medir uso de CPU porque al implementar los protocolos sobre computadoras el uso de CPU era inapreciables en todo el tiempo para ambos protocolos. Pero por el comportamiento de estos protocolos se supone que AODV lograría menor uso de CPU frente a OLSR porque el nodo no tiene que procesar toda la información sobre el estado completo de la red.

El parámetro que pudimos medir respecto a los requisitos de hardware fue el uso de memoria RAM, para los mismos escenarios OLSR consume de 150 a 200 KBytes mientras que AODV no supera los 7 KBytes. Para una computadora estos valores son mínimos, pero pueden ser notables para dispositivos móviles.

Los resultados indican que AODV tiene mejores prestaciones frente a OLSR, pero OLSR tiene un punto a favor en cuando a compatibilidad de hardware. Este protocolo esta disponible para instalar en numerosas plataformas, tanto fijas como móviles y su instalación es sencilla. OLSR esta implementado en varios firmware para routers de baja gama, esto nos permite crear redes mesh con dispositivos de costos muy bajos de una manera muy rápida y fácil. Esto hace que OLSR sea un protocolo muy utilizado en redes mesh.

9 Referencias

Sitio oficial de los desarrolladores de OLSR: <http://www.olsr.org>

Información general sobre AODV: <http://moment.cs.ucsb.edu/AODV/aodv.html>

Implementación de AODV por la Universidad de Uppsala: <http://www.uu.se>

RFC de OLSR: <http://www.ietf.org/rfc/rfc3626.txt>

RFC de AODV: <http://www.ietf.org/rfc/rfc3561.txt>

Desarrolladores de firmware para varios routers de gama baja con OLSR ya instalado <http://openwrt.org>, <http://www.dd-wrt.com>