

PROTOTIPO DE UNA RED MESH CON PROTOCOLO DE ENRUTAMIENTO OLSR PARA LA UNIVERSIDAD PONTIFICIA BOLIVARIANA SECCIONAL BUCARAMANGA

Oscar Jovanni Maestre Sanmiguel¹, Javier Eduardo García Prada²

Universidad Pontificia Bolivariana
Semillero SIRESI
Bucaramanga, Colombia

Resumen

Este artículo describe la implementación del prototipo de una red mesh con protocolo de enrutamiento OLSR y que tiene como objetivo evaluar la cobertura y el desempeño de una red inalámbrica descentralizada analizando las ventajas que brinda este tipo de arquitectura, también se indican los pasos para la configuración de la red con diferentes topologías desplegadas en el campus de la Universidad Pontificia Bolivariana Seccional Bucaramanga.

Palabras claves

GNU/Linux, IEEE 802.11, OLSR, Redes Mesh.

Abstract

This paper describes the deployment of a prototype mesh network with OLSR routing protocol that aims to assess the coverage and performance of a wireless decentralized network analyzing the advantages offered by this type of architecture, it also provides the steps for configuration with different network topologies deployed on the campus of the Universidad Pontificia Bolivariana – Bucaramanga Branch.

Keywords:

GNU/Linux, IEEE 802.11, OLSR, Mesh Network.

¹ Oscar Jovanni Maestre Sanmiguel, Estudiante Ingeniería Informática. Email: oscar.maestre@upb.edu.co

² Javier Eduardo García Prada, Estudiante Ingeniería Informática. Email: javier.garcia@upb.edu.co

1. Introducción

El aumento de dispositivos móviles tales como computadoras portátiles, tablets y smartphones, han marcado en la actualidad una tendencia en la demanda de conectividad entre los usuarios. Las redes inalámbricas IEEE 802.11 más conocidas como “Wi-Fi” se han convertido en la solución perfecta para suplir las necesidades de los usuarios porque posee características tales como movilidad, portabilidad, facilidad de uso e implementación, dando éxito a esta tecnología y llegando a tomar fuerza en entornos urbanos, empresariales, educativos entre otros.

Una de las más recientes innovaciones en esta tecnología son las redes inalámbricas mesh (en Inglés, Wireless Mesh Network WMN), las cuales son redes descentralizadas, compuestas de nodos dispuestos en una topología en malla, en la que cada nodo se encuentra conectado a otros [1] mediante enlaces punto a punto (Ad-hoc), creando redundancia de caminos para el envío y recepción de los datos [11], su cobertura crece con cada uno de estos enlaces, es decir la red se expande a medida que el número de nodos sea mayor.

Para su funcionamiento las redes mesh deben usar protocolos de enrutamiento dinámico, y uno de los más populares es el Optimized Link State Routing OLSR. Este se caracteriza por ser proactivo, es decir mantiene las tablas de enrutamiento de cada nodo actualizadas en todo momento, mediante la técnica conocida como Multipoint Relaying (MPR), la cual detecta los puntos muertos ya sea por carencia de señal o desconexión repentina de algún nodo y hace que estas redes se reconstruyan continuamente [2].

El presente artículo muestra un caso real de aplicación de las redes mesh con el protocolo OLSR sobre parte del campus de la Universidad Pontificia Bolivariana Seccional Bucaramanga, allí se dan características de topografía diversa, edificaciones con puntos de acceso (AP) a la red, y con una alta demanda de dispositivos móviles que no toleran una eventual caída del AP o que la señal se vea obstruida. La aplicación realizada sobre el entorno anteriormente descrito puede resultar de interés para otras universidades con requerimientos de conectividad en condiciones similares.

El documento inicia con la descripción de las redes inalámbricas IEEE 802.11 y su modo de operación. En la Sección 3, se explica el origen de las redes mesh, su funcionamiento y ventajas. En la sección 4 se hace la descripción del protocolo de estado de enlace OLSR. En la sección 5 se muestra la implementación de la red mesh, el proceso de configuración de las interfaces de red e instalación de OLSR y finaliza la sección con pruebas de conexión. En la Sección 6, se analizan los resultados de cobertura y redundancia de la red mesh implementada. Por último, se concluye sobre las ventajas en el entorno descrito.

2. Redes Inalámbricas IEEE 802.11

Las redes inalámbricas son una de las tecnologías más prometedoras y discutidas en la actualidad ya que permite la interconexión entre dos o más puntos por medio de ondas electromagnéticas que usan el aire como medio de transmisión, añadiendo así flexibilidad y movilidad.

Esta tecnología opera en el estándar de la IEEE 802.11 también llamado Wi-Fi [1], desarrollado por la IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) donde se definen la forma en que

deben operar los dispositivos de red inalámbricos para establecer una conexión y además garantizar calidad del servicio, estos procesos se llevan a cabo en los dos niveles inferiores de modelo OSI (Capa física y enlace de datos).

Existen muchos protocolos en la familia 802.11 siendo los más representativos para la construcción de redes inalámbricas de bajo costo, los estándares 802.11a, 802.11b, 802.11g y 802.11n, ver Tabla 1. Estos protocolos trabajan en las bandas de radio frecuencia de 2.4GHz y 5GHz asignadas por la Unión Internacional de telecomunicaciones (UIT) como bandas de uso libre sin licencia [15].

Tabla 1. Valores Característicos, Protocolos de Radio de la Familia IEEE 802.11

	802.11^a	802.11b	802.11g	802.11n
Velocidad Estimada	54Mbps	11Mbps	54Mbps	300Mbps
Frecuencia	5GHz	2.4GHz	2.4GHz	2.4GHz y 5GHz
Distancia (metros)	20m	100m	150m	250m
Canales sin solapamiento	12	3	3	12

2.1 Funcionamiento

Para realizar una conexión entre dos o más puntos es necesario que los dispositivos de red inalámbricos estén operando en la misma porción del espectro de radio, esto significa que los radios 802.11a se comunican con los de la misma frecuencia de 5GHz por ejemplo con otro radio 802.11a o 802.11n, sin embargo un dispositivo 802.11b con frecuencia de 2.4GHz no puede comunicarse con un 802.11a, es decir las tarjetas inalámbricas deben coincidir en el mismo canal para realizar una comunicación; este canal es una fracción del espectro donde se evita solapamientos entre ellos realizando una canalización que depende de la frecuencia de radio y regulación propia de cada país.

2.2 Modos de operación

La IEEE 802.11 especifica modos de operación de las interfaces de redes inalámbricas, influenciadas en gran medida por su topología [11].

- *Modo Infraestructura (o Maestro):* Dispone como mínimo un Punto de Acceso (AP), donde las estaciones no pueden realizar una conexión directa, todos los datos deben pasar a través del AP de manera centralizada, ver Fig. 1.
- *Modo Administrador (o Cliente):* Estas tarjetas solo pueden unirse a una red creada por una tarjeta en modo infraestructura, aparentando ser un cliente del maestro.

- *Modo Ad-Hoc*: Se caracteriza por no tener AP y ser descentralizada, las estaciones se comunican directamente entre sí, con conexión punto a punto, cada nodo debe estar dentro de la cobertura de los otros para comunicarse, ver Fig. 2.

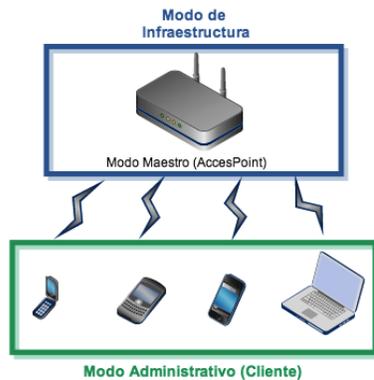


Figura 1: Modo de operación infraestructura y administrativo

Fuente: Autores a partir de [11]



Figura 2: Modo de operación Ad-hoc (punto a punto)

Fuente: Autores a partir de [11]

3. Red Mesh

Es una red inalámbrica, descentralizada, punto a punto, compuesta de nodos independientes que se comunican entre sí, siendo estos nodos computadores de escritorio, portátiles, dispositivos móviles como tabletas, Smartphone u otras unidades que soportan el estándar IEEE 802.11 [1]. La movilidad de estos nodos la ubican dentro la categoría MANET (Mobile ad hoc network) [4]. En este tipo de redes cada nodo propaga la señal haciendo que aumente la cobertura de la red, formando así un cloud mesh.

3.1 Origen

Las redes mesh nacieron de antiguas técnicas de comunicación militar solucionando problemas de conectividad para la transmisión a larga distancia de manera inalámbrica, donde el equipo de radio de cada soldado contribuía como nodo en la formación de una red cuyo radio de cobertura aumentaba o disminuía dependiendo de la ubicación de cada equipo portado por los soldados[3]. Todo esto evolucionó gracias a que se fueron desarrollando varias formas de brindar servicio y

tráfico al incursionar en ambientes civiles [4]. En la Fig. 3 se observa una red mesh compuesta de nodos parcialmente conectados.

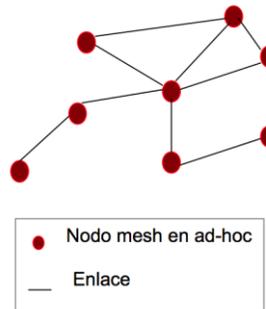


Figura 3: Ejemplo de una red mesh parcial
Fuente: Autores

Estas redes se diferencian de otras porque los puntos que la forman se pueden conectar entre sí por medio de múltiples saltos lo cual otorga una redundancia de enlaces a medida que estos sean mayores.

3.2 Funcionamiento

Básicamente cada nodo o punto de una red mesh posee una interfaz de comunicación inalámbrica compatible con el estándar IEEE 802.11 y configurada en modo ad-hoc lo cual es una ventaja ya que puede operar como una estación cliente o como un enrutador para permitir saltar hacia otro punto de la red. Para lograr que la tabla de rutas se mantenga actualizada, se hace necesario el uso de un protocolo de enrutamiento que verifique el estado de los caminos y la ruta más conveniente. Todos los nodos de una red responden al mismo Service Set Identifier (SSID) y operan dentro el mismo canal de frecuencia, como se describió en la Sección I.[11]

4. OLSR

El Optimized Link State Routing (OLSR) [7], es un protocolo de enrutamiento por estado de enlace para redes ad-hoc, definido por el RFC3626 que utiliza la capa tres del modelo OSI para encontrar las rutas a través de toda la malla, donde cada nodo envía un mensaje de saludo (Hello) en intervalos establecidos. Los nodos adyacentes reciben este mensaje, comparan el mapa de red con el mensaje para detectar un cambio en la ruta, si dicho cambio existe se procede a transmitir a los vecinos un mensaje TC (cambio de topología). Cada vez que llega un mensaje TC se debe volver a calcular toda la topología de la red; así cada nodo sabrá enrutar cada paquete usando su nueva tabla de enrutamiento actualizada, ver Fig. 4. OLSR se basa en el algoritmo de Dijkstra usando una métrica ETX (Expected Transmission Count) [8] para establecer dichos caminos; luego, todos los nodos emiten esos cambios de la estructura de la red a sus vecinos manteniendo actualizada la tabla de rutas [6]. En la tabla 2 se listan los tipos de mensajes usados por el protocolo OLSR.

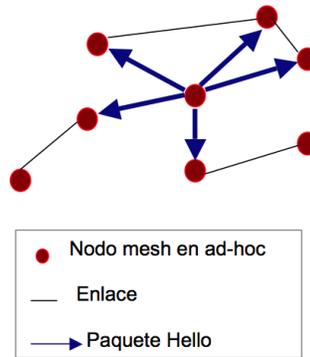


Figura 4: Propagación del paquete *hello* generado por OLSR
Fuente: Autores

Tabla 2. Tipos de Mensajes OLSR

Tipo	Descripción
HELLO	Recuenta el estado de enlace con sus vecinos.
TC	Actualiza la topología de la red y la publica a sus vecinos.
MID	Declara la presencia de nodos con múltiples interfaces.
HNA	Indica una conexión a otras redes.

Fuente: Autores a partir de [5]

OLSR está diseñado para operar en modo distribuido, por lo que no depende de una entidad central, donde la carga del tráfico se reparte en varios nodos inalámbricos y no solo en uno [5].

Debido a que OLSR es un protocolo proactivo, utiliza permanentemente la red para el envío de los mensajes HELLO, TC, MID y HNA; a medida que aumenta la cantidad de nodos cambia la topología, y el tráfico de mensajes puede congestionar la red, ocasionando pérdida de paquetes. Así mismo, OLSR en cada nodo hace el cálculo completo de la ruta hacia otros nodos dado que conoce la topología de la red, esto presenta la ventaja de que pueda funcionar de manera descentralizada pero tiene alto costo de procesamiento en los nodos. Para mejorar la calidad de servicio, OLSR brinda un variado número de plug-ins que permiten la optimización del tráfico en servicios específicos como HTTP, streaming, VoIP, visualización de la topología y herramientas de administración de la red, que ayudan a la detección, análisis y solución de estos problemas. Esta variedad de plug-ins se encuentran disponibles gracias a la licencia BSD que pone a disposición el código fuente sin limitación alguna. [9].

5. Implementación Red Mesh

Para proceder con las pruebas de cobertura y redundancia de una red mesh se hace necesario seleccionar el hardware y software adecuado [1]: para el primero se ha de tener en cuenta como mínimo una interfaz de red inalámbrica que soporte el modo ad-hoc en sus controladores; en cuanto al software, se debe tener en cuenta el sistema operativo y las herramientas para modificar parámetros tales como el modo de operación de la interfaz de red, la frecuencia de los canales,

la potencia de transmisión, ESSID, Cell-ID entre otros. Estas herramientas pueden estar ya instaladas en el sistema operativo o por el contrario se deben implementar de forma manual. Las pruebas aquí realizadas se hicieron bajo el sistema operativo GNU/Linux el cual se integra con la Wireless Extensions (WE), API genérica que permite a un controlador obtener un espacio de configuración y almacenar estadísticas de la interfaz de red; también se utiliza la Wireless Tools (WT) que son un conjunto de herramientas para manipular las WE por medio de una interfaz por línea de comandos [10]. La Tabla 3 muestra las condiciones generales de las pruebas desarrolladas. La Tabla 4 muestra los intervalos de tiempo usados en los paquetes por el protocolo OLSR, y la Tabla 5 muestra el direccionamiento lógico de la red mesh en cada uno de sus cuatro nodos. Las pruebas fueron realizadas con un prototipo en las instalaciones de la Universidad Pontificia Bolivariana Seccional Bucaramanga:

Tabla 3. Condiciones Generales

Cantidad de nodos mesh	4
Estándar inalámbrico	802.11b/g/n
Potencia de transmisión por nodo	10dBm
Sistema operativo	OpenWRT (backfire), Ubuntu 10.04
Canal inalámbrico	11
Implementación de OLSR	Olsrd v0.6.1
Línea de vista entre nodos	sí
ESSID	MESH-UPB
Cell-ID	00:11:22:33:44:55
Tipo de antena	Omnidireccional
Ganancia de las antenas	5 dB

Tabla 4. Intervalos de los Mensajes OLSR

Tipo de paquete	Intervalo de tiempo
HELLO	5 segundos
TC	0.5 segundos
HNA	5 segundos
MID	5 segundos

Tabla 5. Direccionamiento de los Nodos

Identificación del nodo	Dirección IP/Prefijo
Nodo1 (n1)	10.1.1.2/32
Nodo2 (n2)	10.1.1.3/32
Nodo3 (n3)	10.1.1.4/32
Nodo4 (n4)	10.1.1.5/32

4.1 Configuración de las interfaces inalámbricas y OLSR

En primer lugar se debe identificar la tarjeta de red inalámbrica que se desea utilizar, para ello se accede a la consola de GNU/Linux ejecutando el comando (1):

```
#iwconfig (1)
```

```
lo          no wireless extensions.
eth1       no wireless extensions.
wlan0      IEEE 802.11bg  ESSID:off/any
           Mode:Managed Access Point: Not-Associated Tx-Power=20 dBm
           Retry long limit:7   RTS thr:off   Fragment thr:off
           Power Management:on
eth0       no wireless extensions.
```

Figura 5. Resultado del comando iwconfig

Fuente: Autores

La Fig 5, muestra el resultado de la ejecución del comando. Allí se puede reconocer la tarjeta de red inalámbrica con el alias wlan0, es importante recordar que todos los nodos mesh deben operar con un mismo nombre de red (ESSID), igual canal y número de celda (Cell-ID) para que la comunicación se pueda establecer entre ellos [11].

El comando (2) ha de ser ejecutado en cada uno de los nodos mesh:

```
#iwconfig wlan0 essid MESH-UPB mode ad-hoc channel 11 txpower 10 ap
00:11:22:33:44:55 (2)
```

Luego, el comando (1) es ejecutado nuevamente y los resultados muestran que los cambios fueron aplicados a la interfaz de red. Ver Fig 6.

```
wlan0      IEEE 802.11bg  ESSID:"MESH-UPB"
           Mode:Ad-Hoc  Frequency:2.462 GHz  Cell: 00:11:22:33:44:55
           Tx-Power=10 dBm
           Retry long limit:7   RTS thr:off   Fragment thr:off
           Power Management:on
```

Figura 6. Resultado de los cambios efectuados en la interfaz wlan0

Fuente: Autores

De esta forma se configuran los nodos con un ESSID llamado MESH-UPB, modo ad-hoc en el canal 11, una potencia de salida de 10 dBm y un número de celda (Cell-ID) de 00:11:22:33:44:55. Cabe mencionar que existen otros parámetros adicionales y no menos importantes como lo son el rts y frag que aunque no se implementaron en el prototipo deben ser tenidos en cuenta, el primero también llamado Request to Send o petición de envío sirve para controlar el volumen de tráfico que pasa sobre un dispositivo inalámbrico que actúa como puente o punto de acceso, y es de bastante utilidad en la disminución de las colisiones que se presentan cuando todos los nodos operan en la misma frecuencia, y se envían paquetes de forma simultánea. El segundo parámetro es el de fragmentación que consiste en dividir cada paquete en pequeños pedazos cuando la red presenta problemas de pérdidas considerables cuando se envían o reciben datos [12].

Cada nodo mesh requiere de una dirección lógica única para establecer una comunicación exitosa, el comando *ifconfig* hace posible asociar una dirección IP con la interfaz de red inalámbrica. Los comandos (3) a (6) muestran el uso de dicho comando para asignar una dirección IP y máscara de subred a la interfaz de red inalámbrica wlan0 a cada uno de los nodos:

Nodo 1:

```
#ifconfig wlan0 10.1.1.2 netmask 255.255.255.255 (3)
```

Nodo 2:

```
#ifconfig wlan0 10.1.1.3 netmask 255.255.255.255 (4)
```

Nodo 3:

```
#ifconfig wlan0 10.1.1.4 netmask 255.255.255.255 (5)
```

Nodo 4:

```
#ifconfig wlan0 10.1.1.5 netmask 255.255.255.255 (6)
```

Para llevar a cabo un enrutamiento dinámico se requiere de la implementación de olsrd, denominado “Demonio de Enrutamiento de Estado de Enlace Optimizado”, es una aplicación desarrollada para el enrutamiento de redes inalámbricas [11] y se puede descargar de forma gratuita desde la página web <http://olsr.org>; una vez instalada se puede editar el archivo de configuración ubicado en la ruta `/etc/olsrd.conf` con el fin de ajustar los intervalos de tiempo de los paquetes como se presentó en la tabla 4. La Fig 7. Muestra los parámetros y sus nuevos valores.

```
#####
### OLSRd Interfaces configuration ###
#####
# multiple interfaces can be specified for a single configuration block
# multiple configuration blocks can be specified

# WARNING, don't forget to insert your interface names here !
Interface "<OLSRd-Interface1>" "<OLSRd-Interface2>"
{
  # Emission intervals in seconds.
  # If not defined, Freifunk network defaults are used
  # (default is 2.0/20.0 for Hello and 5.0/300.0 for Tc/Mid/Hna)

  HelloInterval      5.0
  HelloValidityTime  6.0
  TcInterval          0.5
  TcValidityTime     30.0
  MidInterval         5.0
  MidValidityTime    30.0
  HnaInterval        5.0
  HnaValidityTime    30.0
}
```

Figura 7. Archivo de Configuración OLSR

Fuente: Autores

Así se asegura que los cambios de la red sean reportados en lapsos de tiempo cortos para verificar adyacencias de nodos, recalculando la topología de la red, propagando interfaces y redes remotas. Como cada nodo se convierte en un dispositivo de enrutamiento y de pasarela hacia otros equipos o redes, el kernel de Linux necesita tener activo el reenvío de paquetes, este se habilita con el comando (7)

```
#echo "1" > /proc/sys/net/ipv4/ip_forward (7)
```

Finalmente se ejecuta el comando `olsrd` junto con el archivo de configuración indicando la interfaz de red inalámbrica en todos los nodos mesh. Ver comando (8).

```
#olsrd -i wlan0 -cf /etc/olsrd.conf (8)
```

El comando (9) ejecutado para cada nodo, permite verificar que `olsrd` está funcionando correctamente.

```
#netstat -aunp | grep olsrd (9)
```

El resultado indica que el `olsrd` se está ejecutando en el puerto 698/UDP, en todas las direcciones IP que disponga el enrutador o computadora. La comunicación por este puerto proporciona el intercambio de los paquetes HELLO, TC, MID, y la propagación de otra información relevante como las zonas HNA. Ver Fig. 8.

```
udp        0      0 0.0.0.0:698      0.0.0.0:*        15915/olsrd
```

Figura 8. Resultado del comando `netstat` filtrando el proceso `olsrd`

Fuente: Aurores

6. Resultados

La estructura del prototipo de red mesh que se implementó tiene dos variaciones, la primera consiste en posicionar los nodos mesh en topología de línea inalámbrica (ver Fig. 9), formando una conexión punto a punto, permitiendo extenderse en longitud máxima, donde `n1`, `n2`, `n3` y `n4` representan cada uno de los puntos nodales y `d` la distancia que separa un nodo del otro.

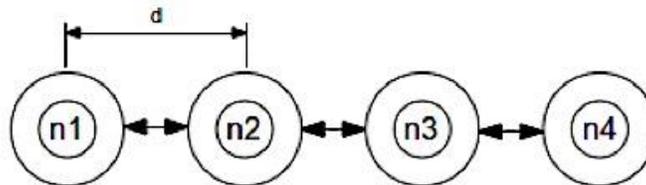


Figura 9: Red mesh dispuesta como línea inalámbrica

Fuente: Los autores

Para este caso es evidente que no existe una redundancia de enlaces pero sí una prolongación de la cobertura al aumentar la distancia `d` de acuerdo al número de nodos existente. Ahora si se desea analizar de manera práctica cómo `n1` lograría comunicarse con `n4` y viceversa, es posible llevar a cabo pruebas con herramientas de rastreo de paquetes, una de las más populares es `traceroute` utilizada en plataformas UNIX y también implementada en Windows; para usarla basta con ingresar el comando `traceroute` seguido de la dirección IP de destino [14]. Ver comando (10).

```
#traceroute 10.1.1.5 (10)
```

La figura 10 muestra los resultados de ejecutar el comando. Se verifica que `n1` pudo comunicarse con `n4` cuya dirección es "10.1.1.5" haciendo tres saltos (10.1.1.3, 10.1.1.4 y 10.1.1.5), donde se expande el área de cobertura de la señal formando un backbone al realizar una concatenación de los demás nodos y es esto debido a que `olsrd` usó MPR para propagar las rutas de forma dinámica y se alteraron las tablas de enrutamiento de todos los miembros de la red mesh para lograr la convergencia de la red.

```
tracert to 10.1.1.5 (10.1.1.5), 30 hops max, 38 byte packets
 1 10.1.1.3 (10.1.1.3) 1026.769 ms 19.462 ms 36.048 ms
 2 10.1.1.4 (10.1.1.4) 19.843 ms 31.047 ms 17.611 ms
 3 10.1.1.5 (10.1.1.5) 38.196 ms 19.286 ms 81.530 ms
```

Figura 10. Resultado del comando traceroute

Fuente: Autores

Por otro lado la segunda configuración que se implementó es la de malla completa donde todas las cuatro estaciones tienen comunicación directa entre sí. Para definir el número de enlaces de la malla completa se aplicó la fórmula de la teoría de grafos $n(n-1)/2$ [14], reemplazando n por el número de nodos (4), y de esta forma se obtiene un número total de seis (6) enlaces como se aprecia en la Fig. 10.

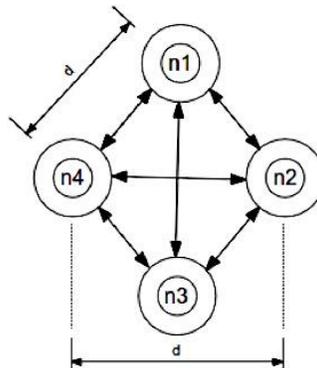


Figura 10. Red mesh usando topología malla completa

Fuente: Autores

La Fig.10 refleja que efectivamente existe una comunicación directa y una redundancia de enlaces; para comprobarlo se hace uso del comando route, ver comando (11)

```
#route -n (11)
```

Donde n indica que no se visualizan nombres de los host. Como ejemplo, el anterior comando aplicado sobre el nodo 4 mostrará los resultados de la Fig. 11. Ellos reflejan que dicho nodo tiene acceso a los otros tres nodos de forma directa (puerta de enlace 0.0.0.0/32 para cada destino), además se corrobora que está utilizando la interfaz de red wlan0 para comunicarse con las direcciones 10.1.1.2, 10.1.1.3 y 10.1.1.4

Tabla de rutas IP del núcleo						
Destino	Pasarela	Genmask	Indic	Métric	Ref	Uso Interfaz
10.1.1.4	0.0.0.0	255.255.255.255	UH	2	0	0 wlan0
10.1.1.3	0.0.0.0	255.255.255.255	UH	2	0	0 wlan0
10.1.1.2	0.0.0.0	255.255.255.255	UH	2	0	0 wlan0

Figura 11. Tabla de enrutamiento del nodo 4

Fuente: Autores

7. Conclusiones

Ante la necesidad de conectividad y el auge de dispositivos móviles que demandan acceso a redes inalámbricas, las redes mesh constituyen una alternativa importante que permite mejorar la cobertura de la señal y redundancia de los enlaces de las redes Wi-Fi.

La implementación de una red mesh haciendo uso de software Open Source representa una gran ventaja porque son de libre acceso y en la mayoría de los casos son gratuitas.

Gracias al uso de OLSR se pudo verificar los cambios en la rutas de forma inmediata, que fueron transmitidas por medio de la inundación de paquetes usando MPR, aunque se debe tener en cuenta que con el aumento de los nodos la cantidad de tráfico generado por OLSR va a aumentar.

Se pudo observar el comportamiento de la red mesh en disposición de topología de línea inalámbrica y de malla completa, siguiendo cada paquete que transita a través de ella.

8. Referencias

- [1] IEEE Std 802.11, 2012 Edición (ISO/IEC 8802-11: 2012) IEEE Standard for Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Network - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
- [2] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR), <http://www.ietf.org/rfc/rfc3626.txt>, Octubre 2003.
- [3] D. Sosa, G. Sosa and R. Blengio. "Monitoreo Redes Mesh", Facultad de Ingeniería Universidad de la República Mayo 2008
- [4] R. Karrer, A. Sabharwal, and E. Knightly, "Enabling large-scale wireless broadband: the case for TAPs, in Proceedings of HotNets-II", Cambridge, MA, Noviembre 2003.
- [5] J. Campos Berga. Anonimato en redes ad-hoc mediante integración de los protocolos HIP y OLSR. Julio 2009
- [6] A. Benjamin. "The grid roofnet: a rooftop ad hoc wireless network. Master's thesis", Massachusetts Institute of Technology, Mayo 2002.
- [7] A. Batiste Troyano. "Protocolos de encaminamiento en redes inalámbricas mesh: un estudio teórico y experimental". Universidad de Sevilla, junio 2011. 71pp
- [8] Implementation and Performance Measurement and Analysis of OLSR Protocol- Hassan Hussein Sinky
- [9] OLSR (2013, Junio), Mesh networking daemon. [En línea] Disponible: <http://www.olsr.org>
- [10] Flickenger Rob: "Building Wireless Community Networks". O'Reilly. First Edition January 2002. ISBN: 0-596-00204-1, 138 pages

- [11] Redes Inalámbricas en los Países en Desarrollo Tercera Edición
- [12] Cachea Johnny, Liu. Vicent: "Hacking Exposed Wireless: Wireless Security Secrets & Solutions". McGraw-Hill/Osborne 2007 (416 pages) ISBN:9780072262582
- [13] Schroder Carla. Linux Networking CookbookTM. O'Reilly Media, 2008
- [14] G. Chartrand, P. Zhang: "Introduction to Graph Theory". McGraw-Hill, 2005
- [15] L. Chamorro, E. Pietrosemoli – Redes Inalámbricas para el desarrollo en América Latina y el Caribe – Asociación para el progreso de las comunicaciones - ISBN 92-85049-66-7 – Diciembre 2008

Sobre los autores



Oscar Maestre Sanmiguel: Estudiante de Ingeniería Informática de la Universidad Pontificia Bolivariana Seccional Bucaramanga, Tecnólogo en Telecomunicaciones, Tecnólogo en Administración de Redes, certificado en CISCO CCNA y PANDUIT, coautor de 1 libro y 1 artículo basado en investigaciones con NETFPGA, miembro activo del semillero SIRESI, actualmente se desempeña en el campo desarrollo de software, Redes de Datos y software libre.



Javier Eduardo García Prada: Estudiante de Ingeniería Informática de la Universidad Pontificia Bolivariana Seccional Bucaramanga, ha participado en semilleros de investigación alrededor de temas en estudios de programación, redes de datos y seguridad informática, miembro activo del semillero SIRESI.