

**ESTUDIO Y COMPARACIÓN DE PROTOCOLOS DE ENRUTAMIENTO
AVANZADOS PARA REDES INALÁMBRICAS**

CHRISTIAN JOSE GELVES TORRADO

SONIA JULIANA MORA NIÑO

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA DE SISTEMAS
GRUPO DE TECNOLOGÍAS DE INFORMACIÓN (GTI)
TELEMÁTICA
BUCARAMANGA**

2014

**ESTUDIO Y COMPARACIÓN DE PROTOCOLOS DE ENRUTAMIENTO
AVANZADOS PARA REDES INALÁMBRICAS**

**CHRISTIAN JOSE GELVES TORRADO
SONIA JULIANA MORA NIÑO**

Trabajo de grado presentado para optar al título de Ingeniero de Sistemas

**Director
Magister Roberto Carvajal Salamanca**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA DE SISTEMAS
GRUPO DE TECNOLOGÍAS DE INFORMACIÓN (GTI)
TELEMÁTICA
BUCARAMANGA
2014**

Nota de aceptación:

Director

Evaluador 1

Evaluador 2

Bucaramanga, 14 de julio de 2014

CONTENIDO

	pág.
INTRODUCCIÓN	14
1. PROTOCOLOS DE ENRUTAMIENTO AVANZADOS PARA REDES INALÁMBRICAS	16
1.1 INTRODUCCIÓN	16
1.2 CLASIFICACIÓN DE LOS PROTOCOLOS DE ENRUTAMIENTO	17
1.3 PROTOCOLOS DE ENRUTAMIENTO ANALIZADOS	18
1.3.1 Protocolo OLSR (Optimized Link State Routing)	18
1.3.2 Protocolo DSR (Dynamic Source Routing Protocol)	21
1.3.3 Protocolo DSDV (Destination Sequenced Distance Vector)	23
1.3.4 Protocolo TORA (Temporally Ordered Routing Algorithm)	25
1.3.5 Protocolo AODV (Ad-Hoc On Demand Distance Vector)	29
1.3.5.1 Información de enrutamiento	30

1.3.5.2 Descubrimiento de Rutas	31
1.3.5.3 Mantenimiento de rutas	34
1.3.6 Protocolo B.A.T.M.A.N. (Better Approach To Mobile Adhoc Networking)	35
1.3.6.1 Características	36
1.3.7 Protocolo BABEL	38
1.3.7.1 Funcionamiento	38
1.3.7.2 Búsqueda de rutas	39
1.3.8 Protocolo PWRP (Predictive Wireless Routing Protocol)	40
2. DISEÑO DE UN MODELO DE COMPARACIÓN PARA EVALUACIÓN DE PROTOCOLOS DE ENRUTAMIENTO AVANZADOS PARA REDES INALÁMBRICAS	42
2.1 MODELOS DE COMPARACIÓN DE HERRAMIENTAS DE SOFTWARE LIBRE	42
2.2 DISEÑO DEL MODELO DE COMPARACIÓN	43
2.2.1 Durabilidad intrínseca	43
2.2.2 Solución industrializada	45

2.2.3 Adaptabilidad técnica	46
2.2.4 Estrategia	46
2.2.5 Prestación de servicios	47
2.2.6 Aspectos técnicos	48
3. APLICACIÓN DEL MODELO DE COMPARACIÓN PARA PROTOCOLOS DE ENRUTAMIENTO AVANZADOS PARA REDES INALÁMBRICAS	65
3.1 PRINCIPALES CARACTERÍSTICAS DE LOS PROTOCOLOS A EVALUAR	65
3.1.1 Protocolo OLSR	66
3.1.2 Protocolo DSR	66
3.1.3 Protocolo DSDV	66
3.1.4 Protocolo AODV	67
3.1.5 Protocolo BATMAN	67
3.1.6 Protocolo BABEL	68
3.2 APLICACIÓN DEL MODELO DE EVALUACIÓN	69

4. DISEÑO E IMPLEMENTACIÓN DE UNA RED AD-HOC QUE PERMITE EMULAR EL FUNCIONAMIENTO DEL PROTOCOLO OLSR	82
5. CONCLUSIONES	90
BIBLIOGRAFÍA	92

LISTA DE TABLAS

	pág.
Tabla 1. Características, sub-características y atributos del modelo de comparación diseñado	50
Tabla 2. Evaluación de los seis protocolos pre-seleccionados	70
Tabla 3. Valores Totales de la evaluación de los seis protocolos de enrutamiento avanzados para redes inalámbricas, según el modelo de comparación.	78
Tabla 4. Puntuaciones de los protocolos evaluados según la prioridad.	80
Tabla 5. Porcentaje total del cumplimiento de los protocolos evaluados.	80

LISTA DE FIGURAS

	pág.
Figura 1. Ejemplo de broadcast.	19
Figura 2. Establecimiento de rutas en DSR.	22
Figura 3. Establecimiento de rutas en DSDV.	25
Figura 4. Creación de rutas indicando la dirección del enlace en TORA	27
Figura 5. Mantenimiento de rutas en TORA.	28
Figura 6. Descubriendo rutas con mensajes RREQ	32
Figura 7. Descubriendo camino de vuelta.....	33
Figura 8. Descubrimiento de ruta. Envío del RREP e inicio del envío de Datos	34
Figura 9. Mantenimiento de ruta. Envío del RERR.	35
Figura 10. Proceso de descubrimiento y mantenimiento de ruta.	40
Figura 11. Proceso de eliminación de rutas inválidas.	41
Figura 12. Porcentaje total obtenido por los seis protocolos evaluados, según el modelo diseñado.....	81

Figura 13. Escenario emulación.....	82
Figura 14. Configuración Red Ad-Hoc	84
Figura 15. Conexión a la red ad-hoc “pruebafinal”	85
Figura 16. Conectividad del PC1 con el PC2.....	86
Figura 17. Conectividad del PC1 con el PC3.....	86
Figura 18. Conectividad del PC1 con el PC4.....	87
Figura 19. Lanzamiento del protocolo OLSR	88
Figura 20. Conectividad de la red – Tabla de ruteo.	89

GLOSARIO

AD-HOC: tipo de red inalámbrica descentralizada

AODV: protocolo de enrutamiento Ad -hoc sobre demanda de vector distancia

B.A.T.M.A.N.: protocolo de mejor aproximación a una red ad-hoc móvil

BF: algoritmo vector-distancia de Bellman-Ford

DAG: gráfico a cíclico dirigido

DSDV: protocolo de vector distancia de destino secuencial

DSR: protocolo de enrutamiento de origen dinámico

ID: identificador único

IPC: comunicación entre procesos

IPV6: protocolo de enrutamiento versión 6

MANET: redes moviles Ad-Hoc

MPR: retransmisores multipunto

NDPU: unidad de datos de protocolo de red

NETT: tiempo neto de recorrido

OLSR: protocolo de estado de enlace optimizado

OPENBRR: modelo de calificación disposición de negocios abierto

OSMM: modelo de madurez de código abierto

PDT: tiempo de descubrimiento de la trayectoria

PWRP: protocolo de enrutamiento inalámbrico predictivo

QSOS: calificación y selección de software de código abierto

RERR: error de ruta

RREP: respuesta de ruta

RREQ: solicitud de ruta

SSID: identificador de conjunto de servicios

TORA: algoritmo de enrutamiento temporal ordenada

TTL: tiempo de vida

RESUMEN

Este trabajo muestra los resultados alcanzados durante el estudio de los diferentes protocolos de enrutamiento avanzados para redes inalámbricas determinando sus características específicas, ventajas y debilidades que brindan en diferentes ambientes de trabajo para una mejor optimización de la comunicación.

Por medio de una metodología de evaluación propuesta de herramientas de software libre, se logró comparar los diferentes protocolos de enrutamiento para redes inalámbricas, enfocados en el monitoreo de hardware y servicios, con el fin de presentar el protocolo con mayor grado de cumplimiento a las características; Seguido a esto, se realizara la emulación de dicho protocolo de enrutamiento concebido como el mejor.

Palabras Claves: Redes inalámbricas, Ad-Hoc, QSOS, OLSR y Software Libre.

INTRODUCCIÓN

Las redes inalámbricas desde que aparecieron en el año 1970, han sido objeto de un progresivo interés por parte de la comunidad científica así como también de la comunidad en general, ya que permiten transmitir datos desde un punto a otro sin la necesidad de un medio físico, ajustándose más al estilo y ritmo de vida de los usuarios. Gracias a los últimos avances y progresos en las tecnologías de las comunicaciones se puede esperar que en un futuro cercano se produzca una mayor propagación de estas tecnologías.

Uno de los inconvenientes que padecen las redes inalámbricas es el enrutamiento de mensajes por medio de una infraestructura dinámica, debido a que sus nodos se encuentran en constante movimiento. Una red inalámbrica cuenta con múltiples caminos para llegar a su destino, pero es de gran importancia manejar el mejor protocolo de enrutamiento, el cual debe establecer una ruta óptima para la comunicación desde el nodo origen hasta el nodo final.

En las redes inalámbricas el enrutamiento es un aspecto crítico y significativo, por esto se hace importante realizar un estudio y comparación de los diferentes protocolos de enrutamiento, determinando sus características funcionales y aspectos como: uso de hardware, uso de red, velocidad de respuesta, entre otros, con el fin de establecer el mejor protocolo que permite dar una eficaz movilidad para los clientes dentro de una red optimizando la comunicación en diferentes ambientes de trabajo.

Para iniciar con el estudio y comparación de los protocolos de enrutamiento avanzados para redes inalámbricas; en este proyecto de grado se propuso como objetivo general lograr identificar los protocolos de enrutamiento avanzados para redes inalámbricas, con el propósito de realizar una comparación que permita determinar su uso y aplicación en ambientes de trabajo con requerimientos específicos, utilizando posteriormente un prototipo para realizar una emulación de al menos dos protocolos.

Éste objetivo se puede llevar a cabo mediante el cumplimiento de ciertos requerimientos específicos como la búsqueda y realización de un estado del arte de los protocolos de enrutamiento avanzados para redes inalámbricas que facilite una mejor comprensión de los conceptos, características y aplicaciones de cada uno de ellos. El diseño de un modelo de comparación que permita evaluar los protocolos de enrutamiento avanzados para redes inalámbricas. La realización de un cuadro comparativo de los protocolos de enrutamiento avanzados para redes inalámbricas, que facilite determinar su uso y aplicación en diferentes ambientes de trabajo. Y por último pero no menos importante, es poder diseñar e implementar un prototipo de red inalámbrica que permita emular el funcionamiento de al menos 2 protocolos de enrutamiento avanzados para este tipo de redes; logrando así con el cumplimiento del objetivo general de este proyecto de grado.

1. PROTOCOLOS DE ENRUTAMIENTO AVANZADOS PARA REDES INALÁMBRICAS

1.1 INTRODUCCIÓN

La búsqueda de un protocolo de enrutamiento en redes Ad-Hoc se ha convertido en un importante desafío debido a la complejidad de este tipo de redes. A diferencia de las redes clásicas cableadas, que presuponen que la topología de la red es poco cambiante, en una MANET se utiliza como medio de transmisión el aire y está en constante cambio por la movilidad de sus estaciones. Por ello no se pueden utilizar los algoritmos ya existentes y se deben buscar otros nuevos que soporten estas condiciones, teniendo en cuenta las limitaciones del ancho de banda, la memoria reducida y la saturación por el denso tráfico que han de soportar.

La función principal de los protocolos de enrutamiento es seleccionar el camino entre el nodo fuente y destino de una manera rápida y fiable. Las redes inalámbricas pueden utilizar los protocolos de enrutamiento de otras redes ya existentes, pero para que en las redes se dé un correcto funcionamiento los protocolos deben modificarse, asegurando las principales características que son el número de saltos, el rendimiento, la tolerancia a fallos, el equilibrado de carga, la escalabilidad y el soporte adaptivo.

Para que un protocolo ofrezca un enrutamiento eficiente el algoritmo debería tener las siguientes características: señalización mínima, mínimo tiempo de procesamiento, que no se produzcan bucles, que sea distribuido, que soporte una

topología dinámica, que soporte enlaces unidireccionales¹.

1.2 CLASIFICACIÓN DE LOS PROTOCOLOS DE ENRUTAMIENTO

Los protocolos de enrutamiento inalámbricos están basados en algoritmos capaces de generar constantemente mensajes de control, asumiendo rutas bidireccionales con una gran calidad. Estos protocolos se pueden clasificar por tres criterios diferentes que son:

- Centralizados o distribuidos: En un protocolo centralizado, una estación central es la encargada de tomar todas las decisiones. En cambio con un protocolo distribuido, las decisiones de enrutamiento se comparten entre todas las estaciones.
- Adaptativo o estáticos: Un protocolo adaptativo intercambia su comportamiento de acuerdo al comportamiento de la red, en cambio un protocolo de enrutamiento estático no cambia su comportamiento.
- Reactivos, proactivos o híbridos: Se denomina un protocolo reactivo cuando éste encuentra rutas cuando lo considere necesario. Ej: AODV, DSR, TORA. En cambio un protocolo de enrutamiento proactivo descubre las rutas antes de necesitarlas. Estos protocolos mantienen tablas de enrutamientos las cuales se actualizan constantemente. Ej: DSDV, OSPF, OLSR. Los protocolos híbridos es una combinación de los protocolos proactivos y reactivos, con el fin de lograr una solución equilibrada.

¹ HIDALGO, Francisco J. Estudio de viabilidad de la utilización de redes inalámbricas Ad-Hoc en edificios departamentales, publicado en Diciembre de 2008. Recuperado el 19 de Agosto de 2013, de riunet: http://riunet.upv.es/bitstream/handle/10251/13183/TesinaMaster_FcoJavierHidalgo.pdf?sequence=1

1.3 PROTOCOLOS DE ENRUTAMIENTO ANALIZADOS

Cada protocolo de enrutamiento posee características propias, objetivos y usa distintas técnicas de enrutamiento. En este capítulo se analizan diferentes protocolos que se especifican a continuación.

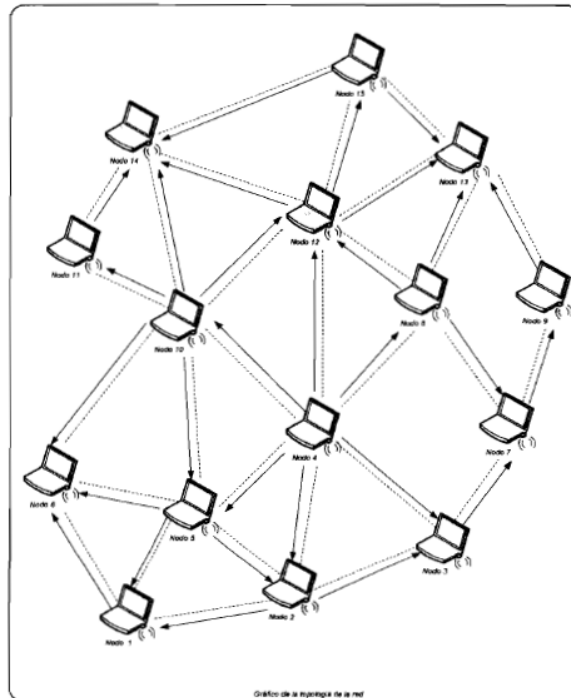
1.3.1 Protocolo OLSR (Optimized Link State Routing) Es un protocolo proactivo que se basa en el estado de los enlaces. Utiliza una técnica eficiente de transmisión de paquetes de estado de enlace MPR (Multipoint Relaying) que consiste en elegir un conjunto de nodos vecinos que cubran el acceso de nodos distantes a 2 saltos o más. La optimización por medio de esta técnica se realiza de dos maneras:

- Al reducir el tamaño de los paquetes de control.
- Al reducir el número de enlaces que son usados para transmitir los paquetes de estado de enlace.

La reducción de los paquetes de estado de enlace, se hace al declarar solamente a un subconjunto de enlaces, dentro de las actualizaciones de estado de enlace. Este subconjunto de vecinos que están designados para las actualizaciones de estado de enlace, y son asignados con la posibilidad de la transmisión de paquetes, son llamados multipuntos de difusión (multipoints relays). La optimización con el uso de multipuntos de difusión facilita las actualizaciones de estado de enlaces periódicas. El mecanismo de actualización de estado de enlace, no genera ningún otro paquete de control cuando un enlace se cae, o cuando un

nuevo enlace se crea. La actualización de estado de enlace alcanza eficiencia operando en redes de gran densidad².

Figura 1. Ejemplo de broadcast.



Fuente: FERNANDEZ, Carlos Arturo. Análisis de los protocolos de enrutamiento y estudio de los parámetros y esquemas para proveer calidad de servicio (QoS) en redes móviles AD HOC, su operación y aplicaciones en los sistemas de telecomunicaciones. Trabajo de grado ingeniería electrónica y telecomunicaciones. Quito: Escuela Politécnica Nacional, 2004.pag 59.

La figura 1 muestra el número de mensajes transmitidos que se requieren, utilizando el modelo de transmisión convencional de broadcast. En este caso el

² FERNANDEZ, Carlos Arturo. Análisis de los protocolos de enrutamiento y estudio de los parámetros y esquemas para proveer calidad de servicio (QoS) en redes móviles AD HOC, su operación y aplicaciones en los sistemas de telecomunicaciones. Trabajo de grado ingeniería electrónica y telecomunicaciones. Quito: Escuela Politécnica Nacional, 2004.pag 59.

número de mensajes transmitidos es aproximadamente igual al número de nodos que forman la red.

Para saber el estado de un enlace se envían mensajes de HELLO, los cuales son pequeños paquetes, que contienen una lista de vecinos con los cuales el nodo tiene enlaces bidireccionales y la lista de vecinos cuyas transmisiones fueron recibidas en un pasado reciente pero con los que no se ha confirmado la presencia de un enlace bidireccional. Cuando un nodo detecte la aparición de un nuevo vecino se debe incluir una nueva entrada a la tabla de enrutamiento e incluir el estado del enlace. Igualmente si se detecta una variación en el estado de un enlace, se debe corroborar en la tabla de enrutamiento que el cambio ha sido reflejado. Cada entrada tiene asociado un valor de tiempo de espera; sí durante ese tiempo determinado no se recibe información de un enlace, se elimina de la tabla de enrutamiento el enlace y el vecino correspondiente. Para calcular las rutas cada nodo contiene una tabla de enrutamiento con el estado de enlace y el nodo, este estado de enlace se mantiene por el intercambio de mensajes periódicos. Los nodos vecinos pueden estar en uno de tres posibles estados de enlace, los cuales son:

- Unidireccional
- Bidireccional
- Multipunto de difusión

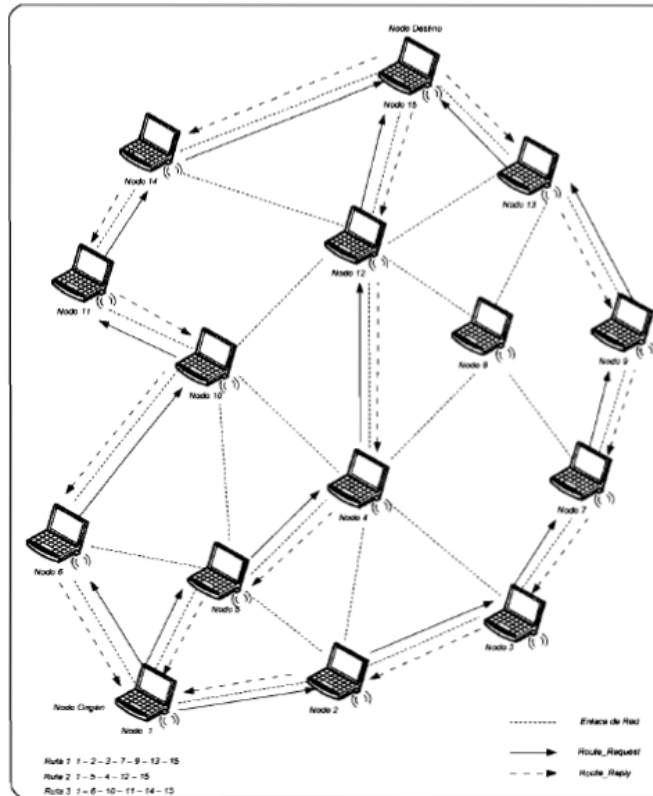
El protocolo OLSR posee varias ventajas que lo hacen mejor que otros protocolos basados en tablas, reduce la sobrecarga de paquetes de control asociados y reduce el número de broadcast de la red.

1.3.2 Protocolo DSR (Dynamic Source Routing Protocol) El protocolo de enrutamiento de origen dinámico, es un protocolo sobre demanda; está diseñado para restringir el consumo de ancho de banda producido por los paquetes de control en redes Ad Hoc, al eliminar los mensajes periódicos de actualización de tablas requeridos por los modelos basados en tablas. Lo que diferencia en cuanto a la demanda, al protocolo DSR de otros protocolos, es que éste no usa transmisiones periódicas de paquetes, las cuales suelen emplearse para informar a un nodo de la presencia de un vecino.

En el protocolo DSR el enrutamiento se estructura desde el origen, se incluye en la cabecera de los datos un campo de información sobre los nodos exactos que debe atravesar, de modo que no se necesita de mensajes periódicos y se disminuye la sobrecarga de mensajes de control. Además ofrece la posibilidad de obtener con una solicitud de una ruta, múltiples caminos hacia el destino. Cada nodo dispone de una memoria caché de rutas donde almacena las rutas ya descubiertas. Cuando un nodo quiere transmitir lo primero que hace es consultar su tabla de enrutamiento para saber si hay una ruta hacia ese destino. Si no tiene la ruta comienza el descubrimiento mediante broadcast de mensajes³.

³ RODRIGUEZ, E., DECO, C., BURZACCA, L., & PETINARI, M. (2000). *Setrem*. Recuperado el 17 de 02 de 2014, de Redes Inalámbricas de uso comunitario: Un análisis comparativo de protocolos: <http://sites.setrem.com.br/stin/2012/anais/Mauro.pdf>

Figura 2. Establecimiento de rutas en DSR.



Fuente: RODRIGUEZ, E., DECO, C., BURZACCA, L., & PETINARI, M. (2000). *Setrem*. Recuperado el 17 de 02 de 2014, de Redes Inalámbricas de uso comunitario: Un análisis comparativo de protocolos: <http://sites.setrem.com.br/stin/2012/anais/Mauro.pdf>

El protocolo DSR elimina la necesidad de envío de paquetes periódicos a la red con mensajes de actualización de tablas que requieren los protocolos basados en tablas, utilizando un modelo reactivo. Para reducir eficientemente los paquetes de control, los nodos intermedios utilizan un caché con información de rutas.

Una desventaja que presenta el protocolo DSR, es que el mecanismo de mantenimiento de rutas no repara localmente una falla en el enlace; además un caché que contiene rutas con información anterior puede llevar a inconsistencias durante la fase de reconstrucción de rutas.

Pese a que el protocolo tiene un buen desempeño en ambientes estáticos o de baja movilidad, el desempeño disminuye rápidamente si se incrementa la movilidad de los nodos, además de que debido al mecanismo de descubrimiento de rutas se halla gran cantidad de paquetes de control, estos paquetes son directamente proporcionales al largo de la ruta.

1.3.3 Protocolo DSDV (Destination Sequenced Distance Vector) El protocolo de Vector Distancia de Destino Secuencial, fue uno de los primeros protocolos propuestos para redes móviles Ad Hoc. Es un protocolo unicast proactivo, su funcionamiento se basa en el algoritmo vector-distancia de Bellman-Ford (BF), donde cada nodo mantiene una tabla que contiene la distancia más corta y el primer nodo en la ruta hacia otro nodo en la red. Incorpora actualizaciones de la tabla con números de secuencia que se aumentan para prevenir el problema de cuenta hacia el infinito, lazos y para una mayor convergencia.

El protocolo DSDV utiliza números de secuencia y al mismo tiempo mantiene la simplicidad del algoritmo de BF. El intercambio de información de enrutamiento y el cálculo de rutas de comunicación se realiza de forma periódica, de esta manera, cada nodo obtiene y procesa información actualizada y confiable de la estructuración de la red. El funcionamiento distribuido de este protocolo demanda que cada uno de los nodos que componen la red distribuya su tabla de enrutamiento con sus nodos vecinos. Solo de esta forma, cada nodo podrá calcular, para cada destino dentro de la red, al nodo vecino que facilite la mejor ruta de comunicaciones.

Debido a que es un protocolo de enrutamiento basado en tablas, todos los destinos se encuentran disponibles para cada nodo todo el tiempo. Las tablas que se intercambian entre vecinos en intervalos regulares de tiempo lo hacen con el fin de tener la topología de la red actualizada. Sí un nodo observa un cambio

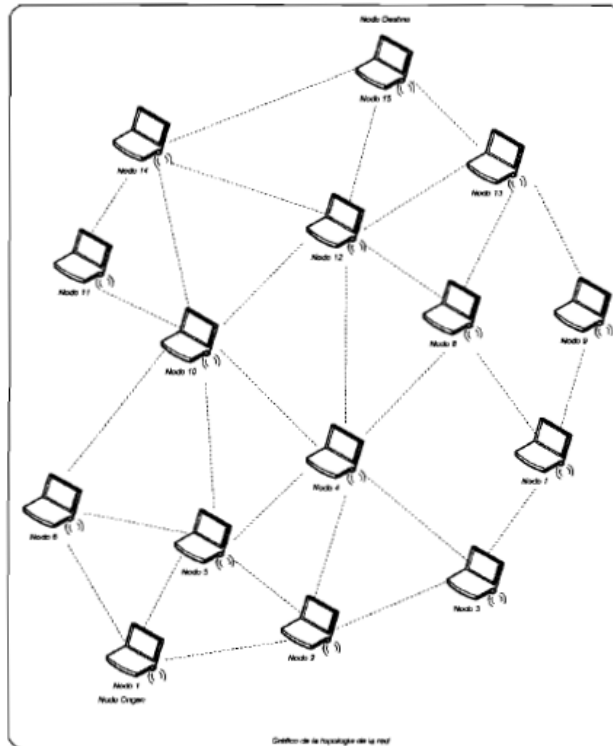
significativo en la topología local, las tablas también se transmiten. Estas tablas de actualización son de dos tipos:

- Tablas de actualización incremental: Una actualización incremental hace referencia a que toma un solo paquete de datos de red (NDPU) y se usa cuando un nodo no observa cambios significativos en la topología local.
- Tabla de actualización completa: Una tabla de actualización completa utiliza varios NDPU's, y se usa cuando el nodo observa cambios significativos en la topología o cuando una actualización incremental requiere más de un NDPU.

Las tablas de actualización son iniciadas por un destino, con un nuevo número de secuencia que será mayor siempre que el anterior. Después de recibir una tabla de actualización, un nodo determina si con la información recibida actualiza sus tablas o mantiene su información anterior por un tiempo hasta seleccionar el menor número de saltos, es decir, la mejor métrica recibida de varias versiones de la misma actualización desde distintos nodos vecinos.

El nodo puede retransmitir o rechazar la tabla de actualización, basado en el número de secuencia de la tabla.

Figura 3. Establecimiento de rutas en DSDV.



Fuente: RODRIGUEZ, E., DECO, C., BURZACCA, L., & PETINARI, M. (2000). *Setrem*. Recuperado el 17 de 02 de 2014, de Redes Inalámbricas de uso comunitario: Un análisis comparativo de protocolos: <http://sites.setrem.com.br/stin/2012/anais/Mauro.pdf>

Dos desventajas del protocolo DSDV son, que no es recomendable para redes con gran número de nodos debido al limitado ancho de banda y que el nodo tiene que esperar un mensaje de actualización iniciado por el mismo destino, sí un nodo desea obtener información de un destino en particular. Lo anterior provoca retardos y la presencia de rutas obsoletas en los nodos.

1.3.4 Protocolo TORA (Temporally Ordered Routing Algorithm) El protocolo de enrutamiento de ordenamiento temporal, es un protocolo altamente adaptable que utiliza un algoritmo distribuido basado en el concepto de inversión de los

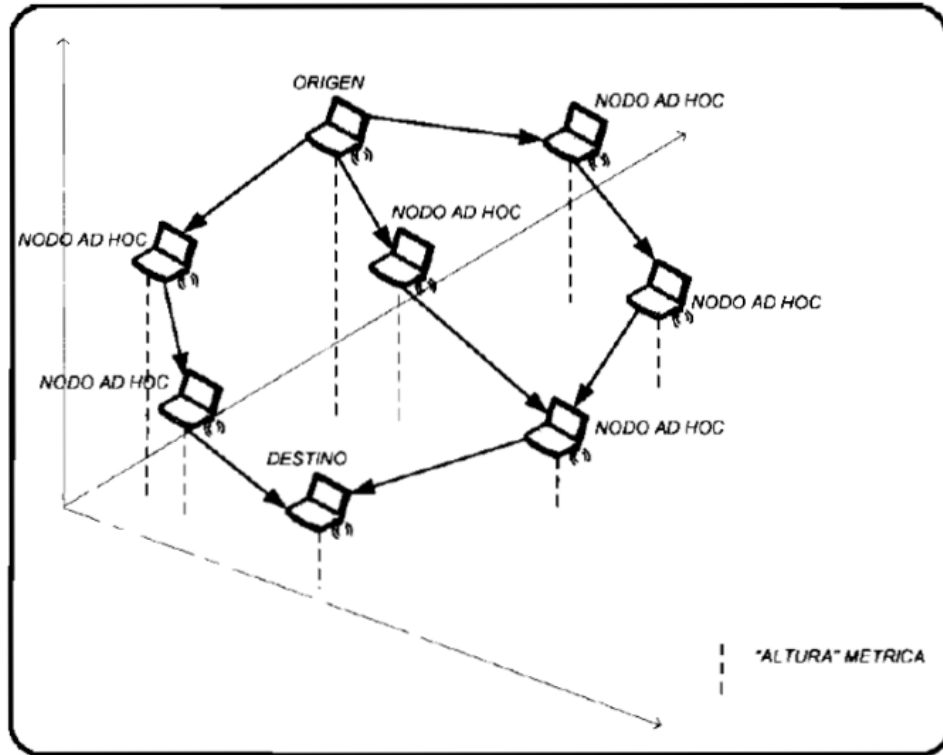
enlaces para proporcionar rutas libres de bucles, este protocolo puede operar en un entorno inalámbrico de alta movilidad, es iniciado por el origen y brinda múltiples rutas para cualquier destino deseado. La clave del diseño del protocolo TORA está en el mantenimiento de los mensajes de control, en un conjunto muy pequeño de nodos cercanos en donde ocurre un cambio en la topología. Pero para lograr esto, los nodos necesitan mantener la información de enrutamiento acerca de nodos adyacentes.

El protocolo TORA consta de tres funciones básicas:

- Creación de rutas
- Almacenamiento de rutas
- Eliminación de rutas

Durante las etapas de creación y mantenimiento de rutas, los nodos utilizan una métrica llamada altura para establecer un gráfico acíclico dirigido (DAG), este gráfico muestra cómo el nodo mira la red sin lazos de enrutamiento, con la raíz en el destino. Luego, los enlaces se asignan en una dirección, ya sea, de subida o bajada, basada en la altura relativa de los nodos vecinos. Lo que dará como resultado un árbol cuya raíz será el nodo destino.

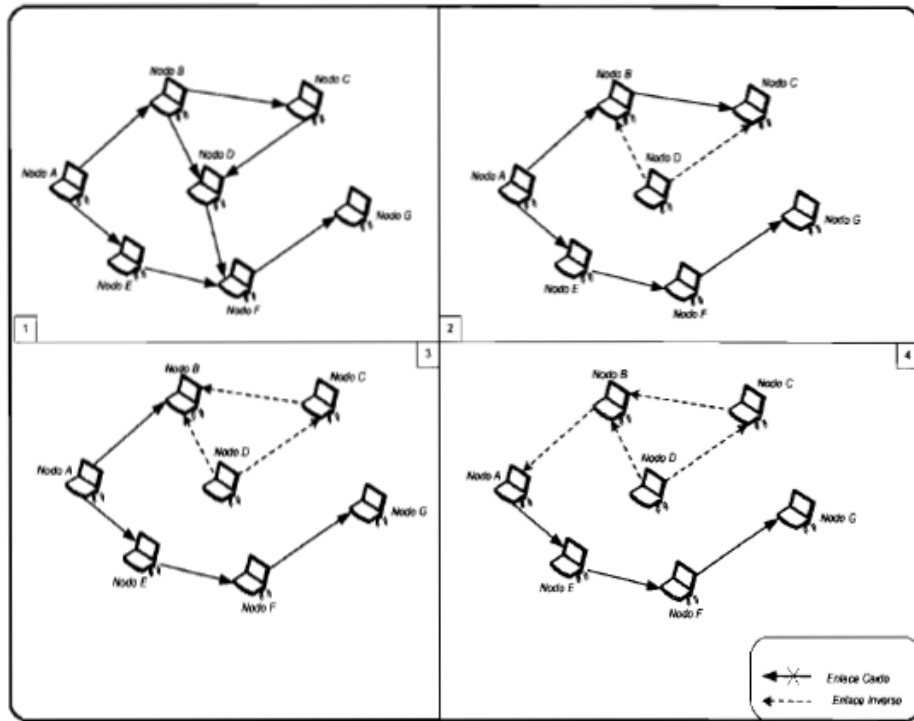
Figura 4. Creación de rutas indicando la dirección del enlace en TORA



Fuente: FERNANDEZ, Carlos Arturo. Análisis de los protocolos de enrutamiento y estudio de los parámetros y esquemas para proveer calidad de servicio (QoS) en redes móviles AD HOC, su operación y aplicaciones en los sistemas de telecomunicaciones. Trabajo de grado ingeniería electrónica y telecomunicaciones. Quito: Escuela Politécnica Nacional, 2004.pag 86.

Si algún nodo se desplaza se puede romper la correspondiente rama del árbol, por lo que sería necesario la ejecución de un procedimiento de mantenimiento de rutas para restablecerla.

Figura 5. Mantenimiento de rutas en TORA.



Fuente: FERNANDEZ, Carlos Arturo. Análisis de los protocolos de enrutamiento y estudio de los parámetros y esquemas para proveer calidad de servicio (QoS) en redes móviles AD HOC, su operación y aplicaciones en los sistemas de telecomunicaciones. Trabajo de grado ingeniería electrónica y telecomunicaciones. Quito: Escuela Politécnica Nacional, 2004.pag 86.

El tiempo es un factor importante para el protocolo TORA, debido a que la métrica de la altura depende del tiempo lógico de la falla del enlace; TORA asume que todos los nodos tienen relojes sincronizados. La métrica de TORA está compuesta de cinco elementos que son:

- Tiempo lógico de falla del enlace
- Identificador único (ID) del nodo que define el nuevo nivel de referencia
- Un bit indicador de reflejo

- Un parámetro de propagación ordenada
- Una identificación (ID) única para el nodo

Los tres primeros elementos colectivamente representan el nivel de referencia. Un nuevo nivel de referencia es definido cada vez que un nodo pierde su último enlace de bajada debido a una falla en el mismo. La etapa de eliminación de rutas en TORA, básicamente implica el broadcast de un paquete que permita borrar todas las rutas inválidas.

En TORA existe gran posibilidad de que ocurran oscilaciones, especialmente cuando conjuntos múltiples de nodos están en el mismo tiempo detectando particiones, eliminando rutas y construyendo nuevas rutas basados en otros nodos. Dado que TORA usa una coordinación intermodal, su problema de inestabilidad es similar a la cuenta de infinito, que ocurre en los protocolos de enrutamiento de vector-distancia, excepto que en el caso de TORA estas oscilaciones son temporales y la convergencia del protocolo se da luego de un determinado tiempo⁴.

1.3.5 Protocolo AODV (Ad-Hoc On Demand Distance Vector) El protocolo AODV permite el enrutamiento dinámico, autoconfigurable y multisalto entre nodos. Es un protocolo de enrutamiento IP que permite descubrir y sostener rutas hacia estaciones de la red. Es un protocolo reactivo ya que las rutas se crean sólo cuando se necesitan (cuando la estación origen quiere transmitir datos a un destino). Las decisiones de enrutamiento se hacen usando vectores distancia, por ej.: distancias medidas en saltos a todos los routers disponibles. Éste protocolo soporta unicast y broadcast.

⁴ FERNANDEZ, Carlos Arturo. Análisis de los protocolos de enrutamiento y estudio de los parámetros y esquemas para proveer calidad de servicio (QoS) en redes móviles AD HOC, su operación y aplicaciones en los sistemas de telecomunicaciones. Trabajo de grado ingeniería electrónica y telecomunicaciones. Quito: Escuela Politécnica Nacional, 2004.pag 86.

Cada estación mantiene un número de secuencia que guarda un timestamp, y una tabla de enrutamiento que contiene rutas hacia los destinos. Los números de secuencia se usan para determinar si una ruta es actual (cuanto mayor sea el número, más actualizada está la ruta; el más antiguo puede descartarse) y con ellos el protocolo se asegura de que no hay bucles. Cada entrada de la tabla contiene la dirección del siguiente salto (siguiente estación hacia el destino), un contador de saltos (número de saltos hacia el destino) y un destination sequence number. Ya que es un esquema de vector-distancia on-demand, los routers mantienen las distancias de aquellos destinos con los que necesitan contactar o transmitir información. Cada ruta activa se asocia con un tiempo de vida almacenado en la tabla; cuando este tiempo finaliza, la ruta se marca como inválida y después se borra de la tabla para no sobrecargarla. AODV usa dos procedimientos principales, el descubrimiento de rutas y el mantenimiento de rutas⁵.

1.3.5.1 Información de enrutamiento El protocolo AODV almacena la información de enrutamiento en forma de tablas de rutas. Cada uno de los nodos de la red mantiene su propia tabla, que tiene tantas entradas como destinos conoce el nodo. Una entrada consta típicamente de los siguientes campos:

- Dirección IP del destino.
- Número de secuencia del nodo destino. Número de secuencia asociado al nodo destino, cuyo valor se obtiene de los mensajes de control.

⁵ HIDALGO, Francisco J. Estudio de viabilidad de la utilización de redes inalámbricas Ad-Hoc en edificios departamentales, publicado en Diciembre de 2008. Recuperado el 19 de Agosto de 2013, de riunet: http://riunet.upv.es/bitstream/handle/10251/13183/TesinaMaster_FcoJavierHidalgo.pdf?sequence=1

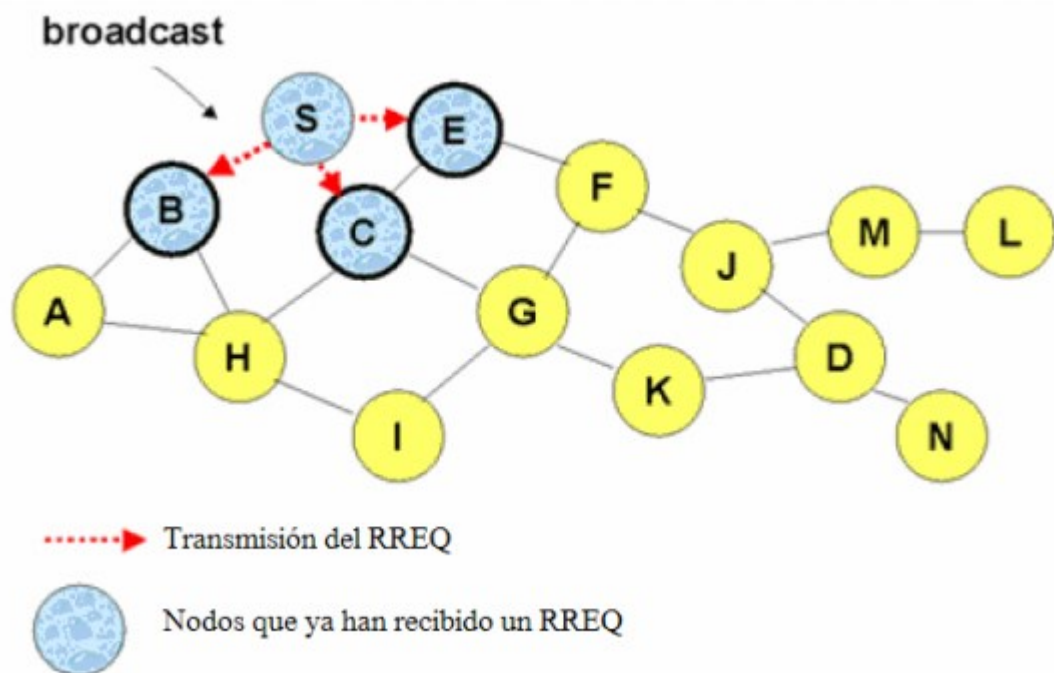
- Indicador de validez del número de secuencia del nodo destino. Si se pretende alcanzar un nodo destino y ha fallado uno de los enlaces implicados, o la ruta ha expirado, el número de secuencia asociado a ese nodo destino se marca como inválido.
- Otros indicadores sobre estado y rutas. Por ejemplo, indicadores acerca de la validez o no de la ruta y en este último caso si es reparable, no es reparable y se debe buscar un camino alternativo, o bien, si está siendo reparada.
- Interfaz de red.
- Número de saltos. Número de saltos necesarios para alcanzar el destino desde este nodo.
- Siguiente salto. Nodo adyacente al que se debe enviar el paquete para llegar al destino deseado.
- Lista de precursores. Lista de nodos que forman el camino resultante del proceso de descubrimiento de rutas.
- Tiempo de vida de la ruta. Tiempo en el que la ruta caduca o debe ser borrada⁶.

1.3.5.2 Descubrimiento de Rutas Cuando la estación fuente (sender, que envía datos) necesita una ruta a un destino, envía por broadcast un mensaje ROUTE

⁶ GIL Jimenez, M. E. ciencia. 2009. Recuperado el 10 de 02 de 2014, de Universidad Rey Juan Carlos: http://ciencia.urjc.es/jspui/bitstream/10115/2546/1/PFC_MariaElenaGilJimenez.pdf

REQUEST (RREQ). Las estaciones guardan un identificador de broadcast. Este identificador y la dirección IP del origen, identifica a un RREQ. Cuando el emisor emite un RREQ su identificador de broadcast y el número de secuencia aumentan en uno. El RREQ se almacena durante un tiempo PATH DISCOVERY TIME (PDT), con esto no se vuelve a procesar cuando vecino se lo envía de vuelta. El emisor espera durante un tiempo NET TRAVERSAL TIME (NETT) hasta que llegue un mensaje ROUTE REPLY (RREP). Si no se recibe este mensaje (RREP) durante este tiempo, el emisor vuelve a enviar por broadcast otro RREQ hasta un número de veces RREQ TRIES. El tiempo de espera (NETT) se duplica con cada intento adicional.

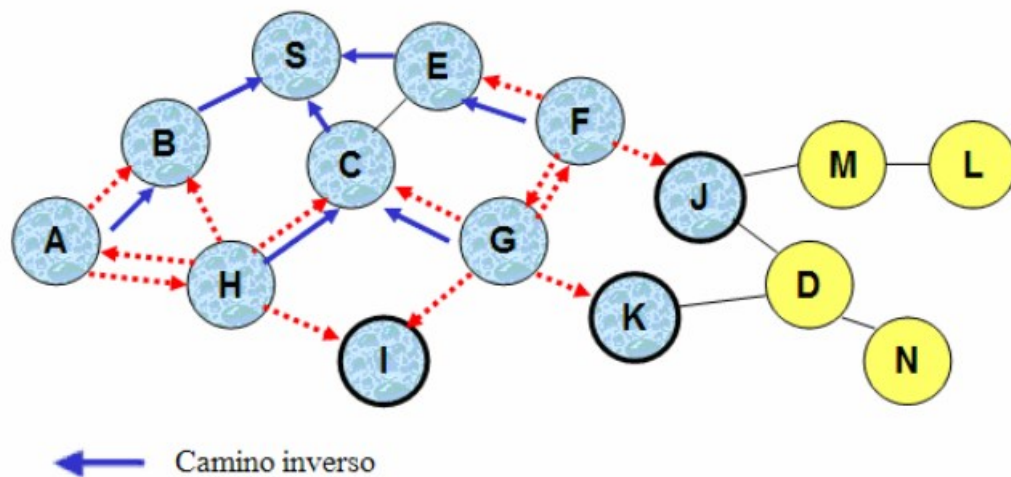
Figura 6. Descubriendo rutas con mensajes RREQ



Fuente: GIL Jimenez, M. E. ciencia. 2009. Recuperado el 10 de 02 de 2014, de Universidad Rey Juan Carlos: http://ciencia.urjc.es/jspui/bitstream/10115/2546/1/PFC_MariaElenaGilJimenez.pdf

Cuando la estación tiene una ruta a un destino responde enviando un mensaje RREP a la estación a la que le llegó el mensaje de petición por unicast. El RREP se transmite de vuelta a la estación fuente, las estaciones conforman punteros hacia el destino. El RREP llega a la estación fuente cuando la ruta se ha creado y los paquetes de datos pueden dirigirse al destino.

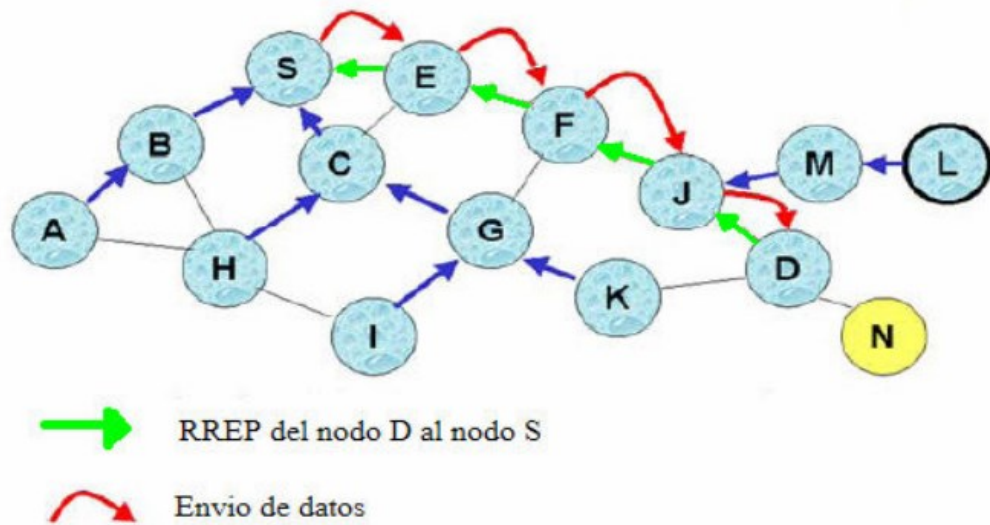
Figura 7. Descubriendo camino de vuelta



Fuente: GIL Jimenez, M. E. ciencia. 2009. Recuperado el 10 de 02 de 2014, de Universidad Rey Juan Carlos: http://ciencia.urjc.es/jspui/bitstream/10115/2546/1/PFC_MariaElenaGilJimenez.pdf

En la figura 7 se ve como el nodo C vuelve a recibir el paquete en modo broadcast de los nodos H y G, pero detecta que el mensaje lo había recibido anteriormente y lo descarta sin reenviar nuevamente. Cuando el mensaje llega al nodo destino, este responde al RREQ enviando de forma unicast un mensaje RREP (Route Reply). El mensaje RREP contiene la ruta hacia el origen invirtiendo el camino del RREQ.

Figura 8. Descubrimiento de ruta. Envío del RREP e inicio del envío de Datos



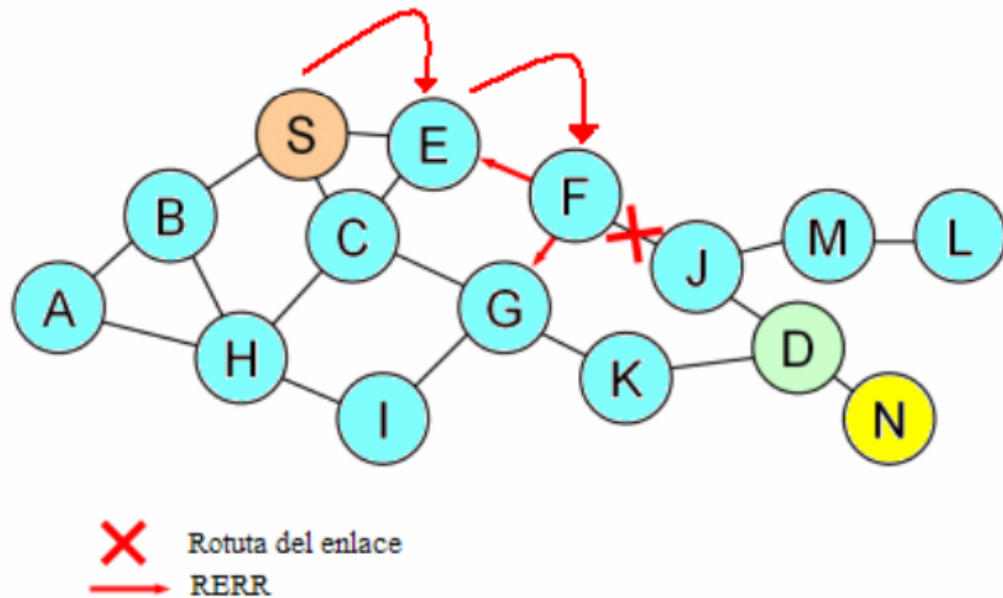
Fuente: GIL Jimenez, M. E. ciencia. 2009. Recuperado el 10 de 02 de 2014, de Universidad Rey Juan Carlos: http://ciencia.urjc.es/jspui/bitstream/10115/2546/1/PFC_MariaElenaGilJimenez.pdf

En la figura 8 el RREP sabe el camino hasta el nodo S al invertir la secuencia del RREQ de llegada. Una vez seleccionado el camino, ya se inicia el envío de datos.

1.3.5.3 Mantenimiento de rutas Una ruta puede dejar de funcionar cuando unas de las estaciones de la red se mueven, esta estación deberá informar a los vecinos los cuales reenviarán el mensaje a todas las estaciones, para posteriormente borrar esta ruta. La estación origen es la encargada de crear una nueva ruta.

El mantenimiento de rutas consiste en brindar un feedback al sender en caso de que un router o enlace deje de funcionar, así la ruta podrá modificarse o redescubrirse.

Figura 9. Mantenimiento de ruta. Envío del RERR.



Fuente: GIL Jimenez, M. E. ciencia. 2009. Recuperado el 10 de 02 de 2014, de Universidad Rey Juan Carlos: http://ciencia.urjc.es/jspui/bitstream/10115/2546/1/PFC_MariaElenaGilJimenez.pdf

Usando mensajes HELLO una estación es capaz de guardar información de sus vecinos los cuales se envían en un tiempo determinado para detectar fallos en los enlaces. Si hay un enlace roto se envía por broadcast un mensaje ROUTE ERROR (RERR) a la red. Cualquier estación que recibe este mensaje, invalida la ruta y reenvía por broadcast los mensajes de error con el destino inalcanzable a todas las estaciones de la red.

1.3.6 Protocolo B.A.T.M.A.N. (Better Approach To Mobile Adhoc Networking)

B.A.T.M.A.N. es un protocolo de enrutamiento dinámico y proactivo para redes malladas ad-hoc que utiliza las tablas de enrutamiento para las decisiones de enrutamiento. B.A.T.M.A.N. no selecciona una ruta completa entre los nodos origen y destino, sino que escoge un nodo de salto para usarlo como gateway

hacia el destino. Este protocolo no calcula rutas completas entre un nodo origen y destino sino que selecciona un nodo de salto para utilizarlo como gateway hacia el destino.

Una de las principales características de B.A.T.M.A.N. es hallar otros nodos B.A.T.M.A.N. y seleccionar el mejor vecino para llegar a ellos. Realiza un seguimiento de los nuevos nodos y comunica a sus vecinos de su existencia.

Es decir cada nodo transfiere mensajes broadcast para señalar a los nodos vecinos de su existencia. Los vecinos lo retransmiten a sus vecinos, y así sucesivamente. De esta manera el mensaje llega a todos los nodos de la red. Los campos que contienen estos mensajes son la dirección de origen, la dirección del nodo que retransmite, el TTL y el número de secuencia. El tamaño típico de estos mensajes es de 52 bytes.

1.3.6.1 Características Las principales características de este protocolo son:

- Soporte en múltiples interfaces: se puede utilizar en más de una tarjeta inalámbrica o Ethernet.

- Soporte de interfaces alias: lo que permite ejecutar otros protocolos en paralelo.

- Interfaz IPC (Inter-Process Communication): para conectarse al demonio batmand. Algunas de las acciones que permite son:
 - Información debugging
 - Consulta de nodos vecinos
 - Consulta de nodos gateway
 - Configuración del gateway en tiempo de ejecución

- Mensajes HNA
- Modificación de interfaces

- Policy Routing: A partir de la versión B.A.T.M.A.N. 0.3 se soporta el policy routing y se pueden utilizar las funciones especiales de enrutamiento proporcionadas por el kernel de Linux. Con el comando “route” se puede consultar las 4 tablas de enrutamiento B.A.T.M.A.N.:

- Tabla “redes”: Contiene las entradas de las redes anunciadas (HNA).
- Tabla “hosts”: Contiene todos los nodos alcanzables.
- Tabla “unreachable”: Contiene los nodos que no son alcanzables en la red.
- Tabla “tunnel”: Contiene la ruta por defecto que se utiliza si existe un gateway disponible.

- Selección de gateway: El protocolo B.A.T.M.A.N. permite anunciar una conexión a internet, e informa del ancho de banda disponible, la velocidad de bajada y de subida. Se pueden buscar anuncios de gateway y decidir a cuál se conecta basándose en los siguientes modos de comportamiento:

- Fast Internet Connection: Considera la calidad del enlace y la clase de gateway. Mantiene la conexión hasta que ya no sea posible.
- Stable Internet Connection: Elige la conexión más estable hacia la gateway y la mantiene hasta que sea posible.
- Fast-switching: Elige la conexión más estable hacia la gateway pero la cambia cuando encuentra otra mejor.
- Late-switching: Elige la conexión más estable hacia la gateway pero la cambia cuando encuentra otra conexión con X veces más calidad. (X parámetro configurable).

- Servidor de visualización: Ya que el protocolo B.A.T.M.A.N. no contiene la topología de la red, cada nodo envía la vista local de sus vecinos al servidor de visualización y éste con todos los envíos de cada nodo hace una recopilación y los incluye en el formato que utilizan las herramientas de visualización de red⁷.

1.3.7 Protocolo BABEL Es un protocolo nuevo, fundamentado en el algoritmo vector de distancias y originado para ser robusto y eficiente en redes inalámbricas malladas y redes cableadas. Se origina sobre las ideas de Destination-Sequenced Distance Vector routing (DSDV) y Ad hoc On-Demand Distance Vector Routing (AODV). Es proactivo, pero con características adaptativas (reactivo).

Diseñado para redes inalámbricas Ad-hoc, es un protocolo robusto para nodos móviles. Babel impide la formación de bucles sin fin y ofrece una rápida convergencia.

1.3.7.1 Funcionamiento El funcionamiento está dado por las siguientes características:

- Nodos e interfaces: Babel distingue entre interfaces y nodos. Un nodo puede tener más de una interfaz formando parte de la red.
- Métrica: Babel se basa en la calidad de los enlaces entre nodos para calcular la métrica de una ruta. Para determinar estas métricas se utilizan dos tipos de paquetes: Hello y IHU (I Heard yoU).

⁷ TRUJILLO, María Benito. Evaluación experimental de redes malladas basadas en el Protocolo B.A.T.M.A.N. Trabajo de grado ingeniería técnica de telecomunicación. Madrid: Universidad Carlos III de Madrid. Departamento Ingeniería Técnica de telecomunicación, 2010. 187 p.

- Algoritmo de Bellman-Ford: Calcula la ruta más corta entre dos nodos en un grafo dirigido y ponderado. Babel se basa en él para el cálculo de rutas.
- Condición de viabilidad: Cuando el enlace entre dos nodos vecinos se rompe, pueden crearse bucles en la actualización de rutas. Para evitar esto, se define la condición de viabilidad como aquella que permite a un nodo rechazar un anuncio de ruta por parte de otro nodo, si dicho anuncio puede crear un bucle sin fin.
- Números de secuencia: Babel utiliza números de secuencia para las rutas, una solución introducida por DSDV y AODV. Cada ruta transporta un número de secuencia, un número entero no decreciente que se propaga inalterado por toda la red.

1.3.7.2 Búsqueda de rutas Con el fin de advertir de su presencia al conjunto de sus vecinos, un nodo envía periódicamente paquetes Hello por todas sus interfaces, en modo broadcast.

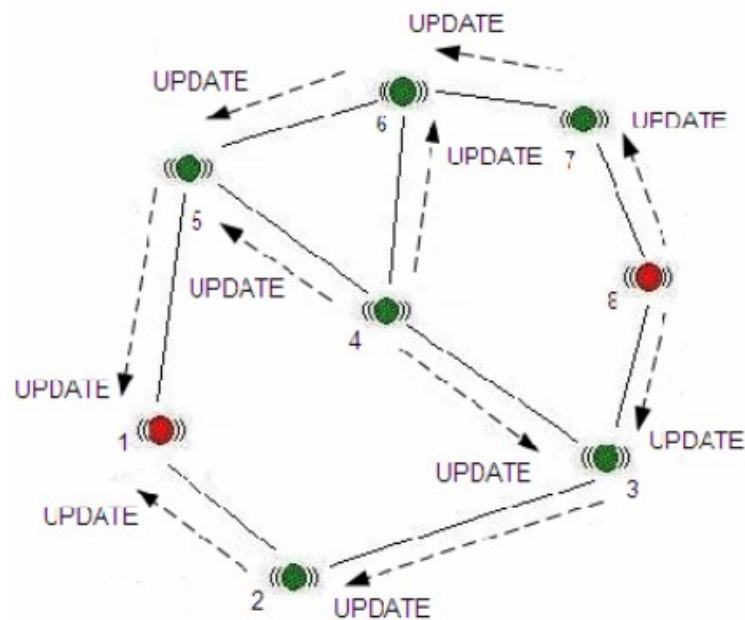
La estructura de un paquete Hello se muestra a continuación:

- Discriminador: Indica que se trata de un paquete Hello.
- Longitud: Longitud del cuerpo del mensaje.
- Número de secuencia: Valor del número de secuencia del paquete para la interfaz que lo envía.
- Intervalo: Límite máximo de tiempo en el que se volverá a enviar otro paquete Hello⁸.

⁸ TROYANO, Albert B. Protocolos de encaminamiento en redes inalámbricas mesh: un estudio teórico y experimental, publicado en Junio de 2011. Recuperado el 20 de Agosto de 2013, de openaccess: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/8164/1/abatistet_TFM_0611.pdf

1.3.8 Protocolo PWRP (Predictive Wireless Routing Protocol) Es el único protocolo de enrutamiento proactivo desarrollado específicamente para redes inalámbricas. Valida el estado del enlace y la calidad del enlace. Este protocolo recalcula 4 veces por segundo el estado de los enlaces de toda la red mallada y selecciona la ruta en función del estado de la red en cada momento. Este funcionamiento soporta redes dinámicas y proporciona la ruta más óptima en cada momento.

Figura 10. Proceso de descubrimiento y mantenimiento de ruta.

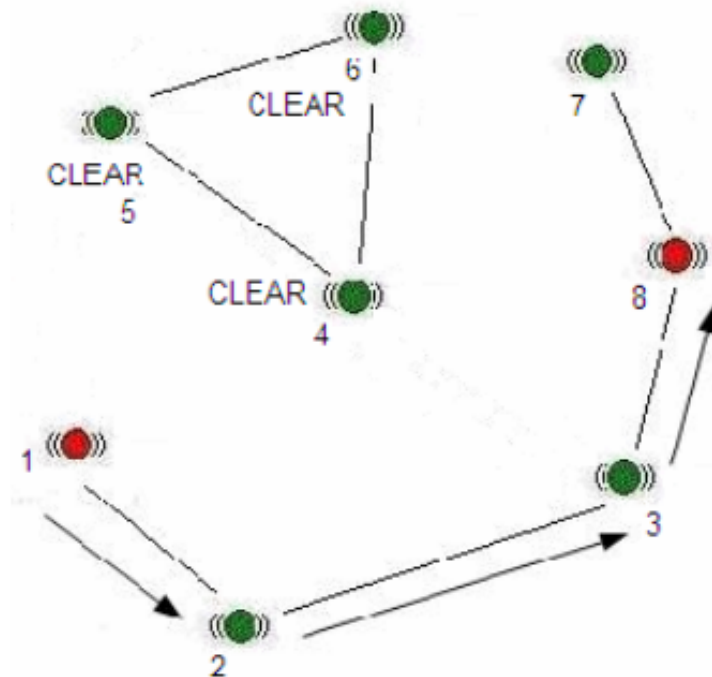


Fuente: TROYANO, Albert B. Protocolos de encaminamiento en redes inalámbricas mesh: un estudio teórico y experimental, publicado en Junio de 2011. Recuperado el 20 de Agosto de 2013, de [openaccess: Http://openaccess.uoc.edu/webapps/o2/bitstream/10609/8164/1/abatistet_TFM_0611.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/8164/1/abatistet_TFM_0611.pdf)

PWRP optimiza dinámicamente las trayectorias de los datos del servidor al cliente y puede escalar a las redes grandes que atraviesan ciudades, provincias, y regiones enteras manteniendo alto rendimiento. PWRP, selecciona dinámicamente

la trayectoria punto a punto basada en la tasa de error de paquete, proporcionando la máxima eficacia espectral, así los enlaces requeridos se encuentren en diferentes bandas de frecuencia, PWRP anula los efectos de la interferencia de RF, reduciendo al mínimo las retransmisiones totales, mejorando la capacidad del suscriptor⁹.

Figura 11. Proceso de eliminación de rutas inválidas.



Fuente: TROYANO, Albert B. Protocolos de encaminamiento en redes inalámbricas mesh: un estudio teórico y experimental, publicado en Junio de 2011. Recuperado el 20 de Agosto de 2013, de [openaccess: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/8164/1/abatistet_TFM_0611.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/8164/1/abatistet_TFM_0611.pdf)

⁹ RODRIGUEZ, Eduardo. DECO, Claudia BURZACCA, Luciana y PETINARI, Mauro. Redes inalámbricas de uso comunitario: un análisis comparativo de protocolos. En: Google Scholar.

2. DISEÑO DE UN MODELO DE COMPARACIÓN PARA EVALUACIÓN DE PROTOCOLOS DE ENRUTAMIENTO AVANZADOS PARA REDES INALÁMBRICAS

Una buena alternativa para las empresas o instituciones que no tienen un gran presupuesto para obtener un protocolo de software propietario, es conseguir uno que sea Software Libre. Por consiguiente se diseñó un modelo de comparación que permite evaluar y comparar protocolos de enrutamiento avanzados para redes inalámbricas, basados en software libre fundamentado en una metodología de comparación de herramientas de software libre, con el fin de dar cumplimiento al segundo objetivo específico.

A continuación se mencionarán algunos de los distintos modelos existentes para la comparación de herramientas de software libre.

2.1 MODELOS DE COMPARACIÓN DE HERRAMIENTAS DE SOFTWARE LIBRE

Existen diferentes metodologías que son usadas con el objetivo de hacer una evaluación de aplicaciones en el ámbito del software libre, éstas pueden estar enfocadas para el producto software final y para su proceso de desarrollo. Algunas de estas metodologías son:

- OPEN SOURCE MATURITY MODEL – CAPGEMINI (OSMM)
- OPEN SOURCE MATURITY MODEL – NAVICA (OSMM)
- OPEN BUSINESS READINESS RATING (OPENBRR)
- OPEN SOURCE MATURITY MODEL – QUALIPSO (OMM)

- QUALIFICATION AND SELECTION OF OPEN SOURCE SOFTWARE (QSOS)

Según el estudio de comparación de las metodologías de evaluación de herramientas de Software Libre realizado en el proyecto de grado “*Estudio y Comparación de Herramientas de Software Libre para la Monitorización de Servicios y Hardware existentes en una Red*” realizado por Carlos Mauricio Guzmán Díaz de la Universidad Autónoma de Bucaramanga, permitió determinar que la metodología Qualification and Selection of Open Source Software (QSOS), es la más apropiada para aplicarse al estudio de nuestro proyecto.

Por lo tanto, el modelo de comparación diseñado se fundamenta en la metodología QSOS.

2.2 DISEÑO DEL MODELO DE COMPARACIÓN

A continuación se hará una breve descripción de las características y sub-características que se utilizaron para evaluar los protocolos de enrutamiento avanzados para redes inalámbricas basados en software libre. Algunas de estas características fueron tomadas de la metodología QSOS para evaluar herramientas de Software Libre y otras fueron propuestas por los autores de este proyecto.

2.2.1 Durabilidad intrínseca

Característica: Durabilidad Intrínseca	
Sub-características	Atributos
Madurez	<ul style="list-style-type: none"> • Edad • Estabilidad • Historia • Probabilidad de bifurcación

Característica: Durabilidad Intrínseca	
Sub-características	Atributos
Adopción	<ul style="list-style-type: none"> • Popularidad • Referencias • Contribución a la comunidad • Libros
Liderazgo en desarrollo	<ul style="list-style-type: none"> • Equipo líder • Estilo de gestión
Actividad	<ul style="list-style-type: none"> • Desarrolladores, identificadores, rendimiento • Actividad en errores • Actividad en funcionalidades • Actividad de liberación
Independencia de desarrollo	

- **Madurez:** hace referencia a la antigüedad, estabilidad y el pasado del software; en esta sub-característica se evalúan los atributos 1) Edad, 2) Estabilidad, 3) Historia y 4) Probabilidad de bifurcación.
- **Adopción:** Indica qué tan conocido es el software y qué tan fuerte es su comunicación con el público y su documentación. Son evaluados los atributos 1) Popularidad, 2) Referencias, 3) Contribución a la comunidad y 4) Libros.
- **Liderazgo en desarrollo:** tiene que ver con el equipo de desarrollo del software y su forma de continuar el desarrollo; los atributos evaluados son: 1) Equipo líder y 2) Estilo de Gestión.
- **Actividad:** se enfoca en la organización del equipo de desarrollo y sus avances en cuanto a nuevas funcionalidades en las liberaciones del software y la corrección de errores. Los atributos evaluados son: 1) Desarrolladores, identificadores, rendimiento, 2) Actividad en errores, 3) Actividad en funcionalidades y 4) Actividad de liberación.

- Independencia de desarrollo: indica si la iniciativa de desarrollo pertenece a un solo grupo o se ven involucradas personas externas.

2.2.2 Solución industrializada

Característica: Solución industrializada	
Sub-características	Atributos
Servicios	<ul style="list-style-type: none"> • Entrenamiento • Soporte • Consultoría
Documentación	
Aseguramiento de la Calidad	
Empaquetado	
Explotabilidad	

- Servicios: informa sobre qué tipo de servicios ofrece la empresa o el grupo desarrollador del software: Se evalúan atributos como 1) Entrenamiento, 2) Soporte y 3) Consultoría.
- Documentación: qué tanta documentación existe referente al software.
- Aseguramiento de la Calidad: indica sobre la forma y las herramientas usadas en el proceso de aseguramiento de la calidad seguido por la empresa o equipo de desarrollo.
- Empaquetado: hace referencia a los diferentes sistemas operativos en los que se puede instalar el software.
- Explotabilidad: da una idea de qué tan fácil de usar es el software y si tiene funciones de administración y control.

2.2.3 Adaptabilidad técnica

Característica: Adaptabilidad técnica	
Sub-características	
• Modularidad	
• Productos	

- Modularidad: se refiere a la posibilidad de expansión de funcionalidades del software mediante el uso de módulos.
- Productos: hace énfasis al cómo se hacen las modificaciones o extensiones del código fuente y si éstas requieren una recopilación completa o parcial del código.

2.2.4 Estrategia

Característica: Estrategia	
Sub-características	Atributos
Licencia	<ul style="list-style-type: none">• Permisividad• Protección contra bifurcaciones propietarias
Derechos de autor	
Modificación del código fuente	
Estado actual del desarrollo del software (roadmap)	
Patrocinador	
Independencia Estratégica	

- Licencia: con base en el tipo de licencia se evalúan los atributos 1) Permisividad y 2) Protección contra bifurcaciones propietarias.
- Derechos de autor: indica si los derechos del código pertenecen a una sola empresa o equipo desarrollador o están distribuidos en varias empresas o equipos.

- **Modificación del código fuente:** describe la forma cómo se hacen las modificaciones al código fuente; indica si se requiere de alguna herramienta para tal gestión.
- **Estado actual del desarrollo del software (roadmap):** con esta sub-característica se indica si el equipo de desarrollo tiene establecido un plan de trabajo para las nuevas liberaciones y correcciones de errores.
- **Patrocinador:** se refiere a si el equipo de desarrollo tiene o no un patrocinador que pague su trabajo.
- **Independencia Estratégica:** indica si el equipo de desarrollo depende o no de alguna empresa que dé un enfoque a los nuevos desarrollos.

2.2.5 Prestación de servicios

Característica: Prestación de servicios
Sub-características
<ul style="list-style-type: none"> • Mantenimiento

- **Mantenimiento:** hace una evaluación basándose en el estado del código, en cuanto a su calidad, el uso de uno o varios lenguajes de programación, la simplicidad pero eficacia del código y si existe o no una buena documentación sobre éste.

2.2.6 Aspectos técnicos

Característica: Aspectos técnicos
Sub-características
<ul style="list-style-type: none">• Planificación• Modelo de información de estado• Modelo de transmisión• Direccionamiento• Modelo de comunicación• Modos de actualización• Número de caminos buscados• Escalabilidad• Estructura• Adecuación para redes altamente dinámicas• IPv6

- Planificación: hace referencia a si el protocolo es proactivo, reactivo o híbrido.
- Modelo de información de estado: Indica de qué forma el protocolo transmite la información, si es basado en topología o basado en destino.
- Modelo de transmisión: hace referencia a la forma de comunicación entre nodos y el consumo que implica, es decir, si es broadcast, multicast ó unicast.
- Direccionamiento: se refiere a la forma en cómo está diseñada la estructura de los nodos, si es un protocolo jerárquico o plano.
- Modelo de comunicación: Indica si el protocolo se comunica por uno o varios canales, es decir, si es un protocolo monocanal o multicanal.
- Modos de actualización: hace referencia a la frecuencia con que se realizan las actualizaciones, es decir, si son actualizaciones periódicas, dirigidas por eventos o por lo contrario son actualizaciones híbridas.

- Número de caminos buscados: se refiere al número de caminos para transmitir la información, es decir, si lo hacen por único camino o por múltiples caminos.
- Escalabilidad: Indica la propiedad que el protocolo tiene para adaptarse al crecimiento de la red.
- Estructura: hace referencia a las funciones ejercidas por los nodos, si son funciones uniformes o no uniformes.
- Adecuación para redes altamente dinámicas: como su nombre lo dice, indica qué tan bueno es la adecuación de los protocolos para redes altamente dinámicas.
- IPv6: evalúa si el protocolo tiene soporte para IPv6 y qué tan estable es el soporte.

En la Tabla 1 se muestran las características, sub-características y atributos que se evalúan en el modelo diseñado, junto con el puntaje de evaluación, el cual se encuentra determinado por un rango de 0 a 2, siendo 0 la peor calificación y 2 la mejor calificación.

Tabla 1. Características, sub-características y atributos del modelo de comparación diseñado

Durabilidad intrínseca		Puntuación		
		0	1	2
Madurez	Edad	Menos de 3 meses.	Entre 3 meses y 3 años.	Más de 3 años.
	Estabilidad	Software inestable con numerosos lanzamientos o parches que generan efectos secundarios.	Versión de producto existente estable pero antiguo. Dificultades para estabilizar las versiones siguientes.	Software estable. Liberaciones siguientes presentan correcciones de errores, pero enfocadas a funcionalidades nuevas.
	Historia, conocimiento de problemas	Se tiene conocimiento de varios problemas del software el cual puede ser prohibitivo	No se conoce de problemas principales o crisis.	Se tiene historia de la buena gestión de situaciones de crisis.
	Probabilidad de bifurcación	El software es muy probable que se bifurque en el futuro.	El software viene de una bifurcación, pero tiene muy pocas posibilidades que se bifurque en el futuro.	El software tiene muy pocas posibilidades de que se bifurque. No procede de una bifurcación.

Tabla 1. (Continuación)

Durabilidad intrínseca		Puntuación		
		0	1	2
Adopción	Popularidad (relacionada a: público general, nicho, entre otros)	Muy pocos usuarios identificados	Uso detectable en Internet (SourceForge, Freshmeat, Google, entre otros)	Numerosos usuarios, numerosas referencias
	Referencias	Ninguna	Pocas referencias, los usos no críticos	Frecuentemente implementada para aplicaciones críticas.
	Contribución a la comunidad	Ninguna comunidad o sin actividad real (foro, lista de correo, entre otros)	Comunidad existente con una actividad notable	Comunidad fuerte: gran actividad en foros, numerosos contribuyentes y defensores
	Libros	No tiene libros acerca del software.	Menos de 5 libros acerca del software disponibles.	Más de 5 libros acerca del software disponibles, en varios lenguajes.
Liderazgo en desarrollo	Equipo líder	1 a 2 personas involucradas, no claramente identificadas	Entre 2 y 5 personas independientes	Más de 5 personas
	Estilo de Gestión	Dictadura completa	Despotismo ilustrado	Consejo de Arquitectos con un líder identificado (Ejemplo: ASF, entre otros)

Tabla 1. (Continuación)

Durabilidad intrínseca		Puntuación		
		0	1	2
Actividad	Desarrolladores, identificadores, rendimiento.	Menos de 3 desarrolladores, no identificados.	Entre 4 y 7 desarrolladores, o más desarrolladores no identificados con importante rendimiento.	Más de 7 desarrolladores claramente identificados, un equipo muy estable.
	Actividad en errores	Lenta reacción en el foro o en lista de correo, o nada sobre solución de errores en notas de la versión	Detectables pero sin proceso claramente expuestas, larga reacción /tiempo de solución.	Reacción fuerte, basada en roles y asignación de tareas
	Actividad en funcionalidades	Ninguna o pocas nuevas funcionalidades	Evolución del producto impulsado por el equipo central o por petición de usuarios sin ningún tipo de claridad	Herramienta(s) para gestionar las peticiones de características, fuerte interacción con el plan de trabajo.
	Actividad de liberación	Actividad muy débil en tanto la producción y liberaciones de nuevas versiones	Actividad en la producción y liberación de nuevas versiones. Frecuente liberaciones pequeñas correspondientes a soluciones de errores.	Importante actividad, con frecuentes lanzamientos de menor importancia para solución de errores y los lanzamientos previstos de versiones principales relativos a la predicción del plan de trabajo

Tabla 1. (Continuación)

Durabilidad intrínseca		Puntuación		
		0	1	2
Independencia de desarrollo	Independencia de desarrollo	Desarrollo realizado de 100% por empleados de una sola compañía	60% Máximo	20% Máximo
Solución Industrializada		Puntuación		
		0	1	2
Servicios	Entrenamiento	No existe oferta de entrenamiento identificadas	Existe oferta, pero se limita geográficamente y a una lengua o es proporcionado por un solo fuente	Variada oferta de diferentes contratistas, en varios idiomas y divididos en módulos de diferentes niveles.
	Soporte	Ninguna oferta de soporte excepto a través de foros públicos y lista de correos	Existe oferta, pero es proporcionada por un solo contratista sin fuerte compromiso de calidad de servicio	Varios proveedores de servicios con un fuerte compromiso (por ejemplo: el tiempo de resolución garantizada)
	Consultoría	Ninguna oferta de servicios de consultoría.	Oferta existe, pero se limita geográficamente y con una lengua o es proporcionado por un solo contratista.	Los servicios de consultoría prestados por contratistas son en diferentes idiomas.

Tabla 1. (Continuación)

Solución Industrializada		Puntuación		
		0	1	2
Documentación	Documentación	No hay documentación	Documentación existe pero se limita a un idioma o no está bien detallado	Documentación siempre actualizada, traducido y posiblemente adaptada a diferentes lectores (usuario final, administración de sistemas, administrador)
Aseguramiento de Calidad	Aseguramiento de Calidad	No tiene ningún proceso AC	Tiene identificado AC pero no está formalizado y sin herramienta.	Proceso de pruebas automáticas de código incluido en la vida de ciclo con la publicación de los resultados
	Herramientas	No hay una herramienta de gestión de solicitud de funciones.	Herramientas estándar proporcionado (por ejemplo: hosting forge), pero mal usados.	Muy activo en el uso de herramientas para manejar las funciones y la asignación de tareas y monitoreando el progreso.
Empaquetado	Código fuente	El software no se puede instalar desde el código fuente, sin mucho trabajo.	La instalación de los fuentes es limitada y depende de condiciones muy estrictas (OS, arco, lib, entre otros)	La instalación desde el código fuente es sencilla.
	Debian	El software no está empaquetado para Debian.	Existe un paquete de Debian, pero tiene problemas importantes y no tiene un soporte oficial.	El software está empaquetado para la distribución.

Tabla 1. (Continuación)

Solución Industrializada		Puntuación		
		0	1	2
Empaquetado	FreeBSD	El software no está empaquetado para FreeBSD.	Existe un paquete, pero tiene problemas importantes y no tiene un soporte oficial.	Existe un paquete oficial para FreeBSD.
	HP-UX	El software no está empaquetado para HP-UX.	Existe un paquete, pero tiene problemas importantes y no tiene un soporte oficial.	Existe un paquete oficial para HP-UX.
	MacOSX	El software no está empaquetado para MacOSX.	Existe un paquete, pero tiene problemas importantes y no tiene un soporte oficial.	Existe un paquete oficial para la distribución.
	Mandriva	El software no está empaquetado para Mandriva.	Existe un paquete, pero tiene problemas importantes y no tiene un soporte oficial.	Existe un paquete oficial para la distribución.
	NetBSD	El software no está empaquetado para NetBSD.	Existe un paquete, pero tiene problemas importantes y no tiene un soporte oficial.	Existe un paquete oficial para NetBSD.

Tabla 1. (Continuación)

Solución Industrializada		Puntuación		
		0	1	2
Empaquetado	OpenBSD	El software no está empaquetado para OpenBSD.	Existe un paquete, pero tiene problemas importantes y no tiene un soporte oficial.	Existe un paquete oficial para OpenBSD.
	RedHat / Fedora	El software no está empaquetado para RedHat / Fedora.	Existe un paquete, pero tiene problemas importantes y no tiene un soporte oficial.	Existe un paquete oficial para la distribución.
	Solaris	El software no está empaquetado para Solaris.	Existe un paquete, pero tiene problemas importantes y no tiene un soporte oficial.	Existe un paquete oficial para Solaris de Sun.
	SuSE	El software no está empaquetado para SuSE.	Existe un paquete, pero tiene problemas importantes y no tiene un soporte oficial.	El software es empaquetado para la distribución.

Tabla 1. (Continuación)

Solución Industrializada		Puntuación		
		0	1	2
Empaquetado	Windows	El proyecto no puede ser instalado en Windows.	Existe un paquete pero limitado o con importantes publicaciones o con cubrimientos específicos para las liberaciones de Windows.	Existe un instalador oficial para Windows.
Explotabilidad	Facilidad de uso, ergonomía.	Difícil de usar, requiere un conocimiento en profundidad de la funcionalidad del software.	Austera y muy ergonómico en términos técnicos.	GUI incluye ayudas de funciones elaboradas (Ejemplo: skins/themes management).
	Administración / Control.	No tiene funcionalidades de administración o monitoreo.	Existen funcionalidades pero incompletas y requiere de mejoramiento.	Funcionalidades de administración y monitoreo completas y de fácil uso. Posible integración con herramientas externas. (Ejemplo: vía SNMP, entre otros)

Tabla 1. (Continuación)

Adaptabilidad técnica.		Puntuación		
		0	1	2
Modularidad	Modularidad	Software monolítico	Presencia de módulos de alto nivel que permite un primer nivel de adaptación de software	Concepción modular, lo que permite una fácil adaptación del software mediante la selección de módulos o incluso desarrollando nuevos
Productos	Modificación del Código	Todo a mano.	Posible recopilación, aunque compleja, sin necesidad de herramientas o documentación.	Recopilación por medio de y siempre con documentación.
	Extensión del Código	Cualquier modificación requiere recompilar el código	Arquitectura diseñada para la extensión estática, pero requiere recopilación	Principio de extensión, arquitectura diseñada para la extensión dinámica sin recopilación

Tabla 1. (Continuación)

Estrategia		Puntuación		
		0	1	2
Licencia	Permisividad (que se ponderarán solo si el usuario quiere convertirse en propietario de código)	Licencia muy estricta, como GPL	Licencia moderadamente permisiva entre los dos extremos (GPL y BSD), doble licenciamiento dependiendo del tipo de usuario (persona, compañía) o sus actividades.	Licencia muy permisiva como las BSD o Apache
	Protección contra bifurcaciones propietarias	Muy permisiva como las licencias BSD o Apache	Licencia moderadamente permisiva entre los dos extremos (GPL y BSD), doble licenciamiento dependiendo del tipo de usuario (persona, compañía) o sus actividades.	Licencia muy estricta, como GPL
Derechos de autor	Derechos de autor	Derechos de autor en manos de unos pocos individuos o entidades, por lo que es más fácil de cambiar la licencia	Derechos autor en manos de muchas personas que poseen el código de una manera homogénea, lo que hace muy difícil su cambio de licencia.	Derechos autor en poder de una persona jurídica en quien confía en la comunidad (por ejemplo, la FSF o ASF)

Tabla 1. (Continuación)

Estrategia		Puntuación		
		0	1	2
Modificación del código fuente	Modificación del código fuente	No hay forma práctica de proponer modificación del código.	Herramientas proporcionadas para acceder y modificar el código (como CVS o SVN), pero no se ha usado para desarrollar el software.	El proceso de modificación del código está bien definido, expuesto y respetado, basado en la asignación de roles.
Estado actual de desarrollo del software (roadmap)	Estado actual de desarrollo del software (roadmap)	No tiene publicado estado actual de software (roadmap)	Plan de trabajo existente no tiene ninguna planificación.	Estado actual de desarrollo del software (roadmap) con planificación y medición de retrasos.
Patrocinador	Patrocinador	Software no tiene un patrocinador, el equipo central no se paga.	El software tiene un patrocinador único que puede determinar su estrategia.	El software es patrocinado por la industria.
Independencia Estratégica	Independencia Estratégica	Ninguna estrategia detectables o una fuerte dependencia de un actor único (persona, empresa, patrocinador, entre otros)	Visión estratégica compartida con varios u otros proyectos de código libre y abierto, pero sin un fuerte compromiso de los propietarios de los derechos de autor	Una fuerte independencia del equipo de trabajo, los derechos de autor pertenecen a una persona jurídica, tenencia, y una fuerte participación en el proceso de normalización.

Tabla 1. (Continuación)

Estrategia		Puntuación		
		0	1	2
Modificación del código fuente	Modificación del código fuente	No hay forma práctica de proponer modificación del código.	Herramientas proporcionadas para acceder y modificar el código (como CVS o SVN), pero no se ha usado para desarrollar el software.	El proceso de modificación del código está bien definido, expuesto y respetado, basado en la asignación de roles.
Estado actual de desarrollo del software (roadmap)	Estado actual de desarrollo del software (roadmap)	No tiene publicado estado actual de software (roadmap)	Plan de trabajo existente no tiene ninguna planificación.	Estado actual de desarrollo del software (roadmap) con planificación y medición de retrasos.
Patrocinador	Patrocinador	Software no tiene un patrocinador, el equipo central no se paga.	El software tiene un patrocinador único que puede determinar su estrategia.	El software es patrocinado por la industria.
Independencia Estratégica	Independencia Estratégica	Ninguna estrategia detectables o una fuerte dependencia de un actor único (persona, empresa, patrocinador, entre otros)	Visión estratégica compartida con varios u otros proyectos de código libre y abierto, pero sin un fuerte compromiso de los propietarios de los derechos de autor	Una fuerte independencia del equipo de trabajo, los derechos de autor pertenecen a una persona jurídica, tenencia, y una fuerte participación en el proceso de normalización.

Tabla 1. (Continuación)

Prestación de Servicios		Puntuación		
		0	1	2
Mantenimiento	Documentación Técnica.	No hay documentación (guía de desarrollo o generado automáticamente como javadoc)	Documentación incompleta o desactualizada, sin concepción y consideraciones de la arquitectura.	Documentación detallada y actualizada, incluye la concepción, diseño y consideraciones de la arquitectura.
Dominio del código	Directo	Ninguna experticia directa del código fuente.	Dominio de código, pero limitada a una sola persona o sólo una parte del código fuente	Dominio de código por varios individuos que cubren en conjunto la totalidad del código fuente.
	Indirecto	Ninguna experticia indirecta del código fuente.	Fuerte dominio a través de expertos externos proporcionados por los socios	Asociación con el dueño de los derechos de autor y / o el equipo básico

Tabla 1. (Continuación)

Aspectos Técnicos	Puntuación		
	0	1	2
Planificación	Protocolo proactivo: Mantiene información consistente y actualizada desde cada nodo hacia todos los demás.	Protocolo reactivo: Solo obtiene información de enrutamiento cuando es necesario.	Protocolo híbrido: Proactividad en las cercanías del nodo considerado pero buscando las rutas bajo demanda para los nodos más alejados.
Modelo de información de estado	No tiene modelo de información de estado	Basados en topología: La información de enrutamiento se transmite en forma de paquetes de estado de enlaces, puede producir sobrecarga.	Basados en destino: Cada nodo mantiene una estructura de datos llamada vector de distancia en la que se incluye cierta información.
Modelo de transmisión	Broadcast: Consume un ancho de banda excesivo y en muchas ocasiones de forma innecesaria.	Multicast: Envía información idéntica a varios destinos, debe mantener información adicional sobre los nodos que se adhieren. Genera inundación.	Unicast: Comunicación uno a uno, es decir, de un nodo origen a un único nodo destino.
Direccionamiento	No aplica	Jerárquico: Existen nodos de niveles superiores y nodos de niveles inferiores. Si uno de estos nodos falla, perjudica todos sus nodos subordinados.	Plano: No existe jerarquía en la red, todos los nodos están al mismo nivel. Si un nodo falla, no afecta a todos los demás.
Modelo de comunicación	No aplica	Multicanal: Protocolos de bajo nivel que, además de funcionalidad de enrutamiento, incorporan mecanismos para la asignación de canal.	Monocanal: Los nodos se comunican sobre un único canal lógico inalámbrico
Modos de actualización	Actualizaciones periódicas: Los nodos distribuyen la información de enrutamiento cada cierto periodo fijo de tiempo.	Dirigidas por eventos: Nodo detecta un evento, como el fallo de un enlace o la aparición de un nuevo enlace, envía un paquete de actualización para difundir el nuevo estado de la red.	Actualizaciones híbridas: nodo difunde su vector de distancias periódicamente y si descubre que un enlace ha fallado distribuye un mensaje de manera inmediata.

Tabla 1. (Continuación)

Aspectos Técnicos	Puntuación		
	0	1	2
Número de caminos buscados	No aplica	Único camino: Permiten un ahorro en el almacenamiento de la información y la simplificación del algoritmo de búsqueda.	Múltiples caminos: Son más robustos y confiables, ya que cuentan a su favor con una rápida recuperación ante fallos de enlaces. Realiza reparto de carga.
Escalabilidad	No tiene propiedad para adaptarse al crecimiento de la red.	Su propiedad para adaptarse al crecimiento de la red es regular.	Su propiedad para adaptarse al crecimiento de la red es muy buena.
Estructura	No aplica	No uniforme: Algunos nodos ejercen papeles específicos en cuanto al tratamiento de la información de enrutamiento.	Uniforme: Todos los nodos cumplen idénticas funciones, enviando mensajes de control y respondiendo a los mismos de igual modo.
Adecuación para redes altamente dinámicas	Su adecuación para redes altamente dinámicas es mala.	Su adecuación para redes dinámicas es regular.	Su adecuación para redes altamente dinámicas es buena.
IPv6	Protocolo sin soporte para IPv6.	Soporte IPv6 estable pero antiguo con problemas importantes y no tiene un soporte oficial.	Soporte IPv6 estable y oficial.

Fuente: Autores del proyecto

3. APLICACIÓN DEL MODELO DE COMPARACIÓN PARA PROTOCOLOS DE ENRUTAMIENTO AVANZADOS PARA REDES INALÁMBRICAS

En este capítulo se aplica el modelo de comparación para protocolos de enrutamiento avanzados para redes inalámbricas, basados en software libre y diseñado en el capítulo anterior, que facilite determinar su uso y aplicación en diferentes ambientes de trabajo, permitiendo además seleccionar los dos mejores protocolos.

Los protocolos de enrutamiento avanzados para redes inalámbricas que van a ser evaluados, son aquellos basados en software libre y son los siguientes:

- OLSR
- DSR
- DSDV
- AODV
- BATMAN
- BABEL

3.1 PRINCIPALES CARACTERÍSTICAS DE LOS PROTOCOLOS A EVALUAR

Con base en la información presentada en el Capítulo 1 de este documento, se presenta a continuación, un listado de las principales características de los protocolos de enrutamiento avanzados para redes inalámbricas basados en software libre, anteriormente mencionados.

3.1.1 Protocolo OLSR Sus características son:

- Protocolo unicast y proactivo.
- La métrica de ruteo utilizada es el conteo de saltos.
- Utiliza técnica de transmisión de paquetes de estado de enlace MPR.
- Se adapta bien en redes con gran número de nodos y alta movilidad.
- Para saber el estado de un enlace, se envían mensajes de HELLO.
- Para calcular las rutas cada nodo contiene una tabla de enrutamiento con el estado de enlace y el nodo, el estado de enlace se mantiene por el intercambio de mensajes periódicos.
- Reduce la sobrecarga de paquetes de control asociados y el número de broadcast de la red.

3.1.2 Protocolo DSR Sus características son:

- Protocolo unicast y reactivo.
- La métrica de ruteo utilizada es el conteo de saltos.
- Protocolo de enrutamiento de origen dinámico y sobre demanda.
- El enrutamiento se estructura desde el origen, se incluye en la cabecera de los datos un campo de información sobre los nodos exactos que debe atravesar.
- No necesita de mensajes periódicos.
- Disminuye la sobrecarga de mensajes de control.
- Obtiene con una solicitud de una ruta, múltiples caminos hacia el destino.
- Para transmitir lo primero que hace es consultar la tabla de enrutamiento para saber si hay una ruta hacia ese destino y si no tiene la ruta comienza el descubrimiento mediante broadcast de mensajes.

3.1.3 Protocolo DSDV Sus características son:

- Protocolo unicast y proactivo.

- La métrica de ruteo utilizada es el conteo de saltos.
- Su funcionamiento se basa en el algoritmo vector-distancia de Bellman-Ford, donde cada nodo mantiene una tabla que contiene la distancia más corta y el primer nodo en la ruta hacia otro nodo en la red.
- El intercambio de información de enrutamiento y el cálculo de rutas de comunicación se realiza de forma periódica.
- Protocolo de enrutamiento basado en tablas, por tanto, todos los destinos se encuentran disponibles para cada nodo todo el tiempo.

3.1.4 Protocolo AODV Sus características son:

- Reactivo
- Las decisiones de encaminamiento se hacen usando vectores distancia (VD).
- Soporta Unicast y Broadcast.
- Dos procedimientos principales, el descubrimiento de rutas y el mantenimiento de rutas.
- Elimina rutas inválidas de la tabla para no sobrecargarla y notifica a la fuente.
- Si la estación fuente necesita una ruta a un destino, envía por broadcast un mensaje ROUTE REQUEST (RREQ).
- El destino envía al origen un mensaje ROUTE REPLY (RREP) por unicast, estableciendo punteros hacía el origen.
- Mensajes HELLO son enviados periódicamente para detectar fallos en los enlaces.
- Su modo de actualización es dirigido por eventos.
- Caminos buscados: Múltiples.
- Puede manejar bajas, medias y relativamente altas tasas de movilidad.

3.1.5 Protocolo BATMAN Sus características son:

- Proactivo
- Aprende el mejor salto próximo para cada destino. No es necesario encontrar o calcular la ruta completa.
- Altas probabilidades de fallos en conexiones, pérdidas de paquetes.
- Los paquetes del protocolo B.A.T.M.A.N. contienen solo una cantidad limitada de información y son por ende muy pequeños (52 bytes).
- No hay entidad central que conozca todas las posibles vías a través de la red
- Todo nodo solo determina el dato para elegir el próximo salto.
- Protocolo muy liviano y rápidamente adaptable a fluctuaciones de la topología de red.
- Cada nodo transmite en difusión mensajes llamados OMG (Mensajes de Originador) por broadcast para informar a sus nodos vecinos sobre su existencia.
- Mantiene el rastro de los nuevos originadores e informa a sus vecinos sobre su existencia.
- Usa mensajes HNA para anunciar una puerta de enlace a una red o a un host.

3.1.6 Protocolo BABEL Sus características son:

- Proactivo.
- Basado en el algoritmo vector de distancias.
- Emplea varias técnicas para asegurar la ausencia de patologías de ruteo tales como bucles y ofrece una rápida convergencia.
- Tiene múltiples estrategias para el cálculo de costos de los enlaces y métricas de ruteo.
- Protocolo muy robusto en presencia de nodos móviles.
- Un nodo puede tener más de una interfaz formando parte de la red.
- Babel se basa en la calidad de los enlaces entre nodos para calcular la métrica de una ruta.

- Para determinar métricas utiliza dos tipos de paquetes: Hello y IHU (I Heard yoU).
- Consta con un algoritmo que calcula la ruta más corta entre dos nodos en un grafo dirigido y ponderado.

3.2 APLICACIÓN DEL MODELO DE EVALUACIÓN

A continuación en la tabla 2 se presenta la evaluación de las características, sub-características y atributos establecidos en el modelo diseñado para los protocolos de enrutamiento avanzados para redes inalámbricas mencionados anteriormente. La puntuación de todos los aspectos fue de la siguiente manera: 0 = no cubierto, 1 = parcialmente cubierto y 2 = cubierto en su totalidad. La información fue obtenida por medio de consultas en los sitios web de algunos protocolos y en artículos técnicos y científicos de dichos protocolos.

Tabla 2. Evaluación de los seis protocolos pre-seleccionados

Durabilidad intrínseca		OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Madurez	Edad	2	2	2	2	2	2
	Estabilidad	2	2	1	2	2	1
	Historia, conocimiento de problemas	2	1	0	2	2	0
	Probabilidad de bifurcación	2	1	1	2	0	1

Tabla 2. (Continuación)

Durabilidad intrínseca		OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Adopción	Popularidad (relacionada a: público general, nicho, ...)	2	1	1	2	2	1
Adopción	Referencias	2	1	1	2	2	1
	Contribución a la comunidad	1	0	0	0	1	0
	Libros	0	0	0	0	0	0
Liderazgo en desarrollo	Equipo líder	1	1	1	1	1	2
	Estilo de Gestión	Desconocido	Desconocido	Desconocido	Desconocido	Desconocido	Desconocido
Actividad	Desarrolladores, identificadores, rendimiento.	1	0	0	1	1	0
	Actividad en errores	1	0	0	1	1	0

Tabla 2. (Continuación)

Durabilidad intrínseca		OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Actividad	Actividad en funcionalidades	1	1	0	1	1	0
	Actividad de liberación	1	1	0	1	1	0
Independencia de desarrollo	Independencia de desarrollo	2	2	2	2	2	2
Solución Industrializada		OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Servicios	Entrenamiento	1	0	0	1	1	0
	Soporte	0	0	0	1	0	0
	Consultoría	0	0	0	0	0	0
Documentación	Documentación	2	2	1	2	2	1
Aseguramiento de Calidad	Aseguramiento de Calidad	Desconocido	Desconocido	Desconocido	Desconocido	Desconocido	Desconocido
	Herramientas	Desconocido	Desconocido	Desconocido	2	2	Desconocido
Empaquetado	Código fuente	2	2	1	2	2	1
	Debian	0	2	0	2	2	0
	FreeBSD	2	0	0	0	0	0
	HP-UX	1	0	0	1	1	0

Tabla 2. (Continuación)

Adaptabilidad técnica.		OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Productos	Modificación del Código	1	1	0	1	1	1
	Extensión del Código	1	0	0	1	0	0
Estrategia		OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Licencia	Permisividad (que se ponderarán solo si el usuario quiere convertirse en propietario de código)	2	2	2	2	2	2
Licencia	Protección contra bifurcaciones propietarias	0	0	0	0	0	0
Derechos de autor	Derechos de autor	1	1	1	1	1	1

Tabla 2. (Continuación)

Estrategia		OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Modificación del código fuente	Modificación del código fuente	1	0	0	1	0	0
Estado actual de desarrollo del software (roadmap)	Estado actual de desarrollo del software (roadmap)	1	1	1	1	1	1
Patrocinador	Patrocinador	0	0	0	0	0	0
Independencia Estratégica	Independencia Estratégica	1	0	0	0	0	0

Tabla 2. (Continuación)

Prestación servicios		OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Mantenimiento	Complejidad intrínseca	Desconocido	Desconocido	Desconocido	Desconocido	Desconocido	Desconocido
	Documentación Técnica.	1	0	0	1	0	0
Dominio del código	Directo	2	1	0	1	1	0
	Indirecto	1	1	1	1	1	1
Aspectos Técnicos		OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Planificación		1	1	0	0	0	0
Modelo de información de estado		2	1	2	1	1	2
Modelo de transmisión		1	2	0	2	0	0
Direccionamiento		2	2	2	2	1	1
Modelo de comunicación		2	2	2	2	2	2
Modos de actualización		2	1	2	1	0	0
Número de caminos buscados		2	2	1	2	1	2

Tabla 2. (Continuación)

Aspectos Técnicos	OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Escalabilidad	2	1	1	2	1	1
Estructura	1	2	2	2	2	2
Adecuación para redes altamente dinámicas	2	0	0	2	2	2
IPv6	2	2	0	1	0	2

Fuente: Autores del proyecto

Al realizar esta evaluación siguiendo el modelo de comparación, se obtuvieron los siguientes valores totales:

Tabla 3. Valores Totales de la evaluación de los seis protocolos de enrutamiento avanzados para redes inalámbricas, según el modelo de comparación.

Totales	OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Durabilidad intrínseca	20	13	9	19	18	10
Solución Industrializada	19	12	8	18	17	7
Adaptabilidad técnica.	2	1	0	2	1	1
Estrategia	6	4	4	5	4	4
Prestación servicios	4	2	1	3	2	1
Aspectos Técnicos	19	16	12	17	10	14

Fuente: Autores del proyecto

Al obtener esta información, el objetivo era hacer un filtrado con el fin de presentar estadísticas que fueron tomadas en cuenta para la selección del mejor protocolo.

Se definió un orden de prioridades para las características evaluadas establecidas por la metodología QSOS con la finalidad de seleccionar el mejor protocolo. Se tomaron en cuenta los aspectos considerados más importantes, siguiendo por los relevantes, y dejando de último los de menor importancia para el proyecto. Dicho orden con su respectivo peso es el siguiente:

1. Solución Industrializada 30%
2. Aspectos Técnicos 25%

3.	Durabilidad Intrínseca	15%
4.	Adaptabilidad Técnica	12%
5.	Estrategia	10%
6.	Prestación Servicios	8%

De acuerdo a las necesidades del proyecto, se escogió como característica más importante la solución industrializada, ya que en ella se evalúan aspectos como el soporte, documentación del software, aseguramiento de la calidad y facilidad de instalación; aspectos que brindaron la información necesaria para llevar de manera correcta la respectiva emulación. Seguido de esta característica se considera también importante los aspectos técnicos, ya que allí se evalúan todos los aspectos que caracterizan a los protocolos de enrutamiento avanzados para redes inalámbricas, que permitieron diferenciar los protocolos entre sí. Y posterior a ésta le siguen las demás características que influyeron también a la hora de seleccionar el mejor protocolo.

Teniendo en cuenta este orden establecido de las características con su respectivo peso y la tabla de valores totales de la evaluación (Tabla 3), para evaluar el correcto cumplimiento de los protocolos a las características presentadas por el modelo diseñado en el capítulo anterior y poder seleccionar el mejor protocolo, lo que se hizo fue multiplicar el peso de cada prioridad por los valores establecidos en la Tabla 3. Estos resultados se reflejan en la siguiente tabla:

Tabla 4. Puntuaciones de los protocolos evaluados según la prioridad.

Totales	OLSR	DSR	DSDV	AODV	BATMAN	BABEL
Solución Industrializada	5.7	3.6	2.4	5.4	5.1	2.1
Aspectos Técnicos	4.75	4	3	4.25	2.5	3.5
Durabilidad intrínseca	3	1.95	1.35	2.85	2.7	1.5
Adaptabilidad técnica.	0.24	0.12	0	0.24	0.12	0.12
Estrategia	0.6	0.4	0.4	0.5	0.4	0.4
Prestación servicios	0.32	0.16	0.08	0.24	0.16	0.08

Fuente: Autores del proyecto

Y luego de obtener la Tabla 4, se sumaron todas las puntuaciones obtenidas de cada protocolo como se muestra a continuación:

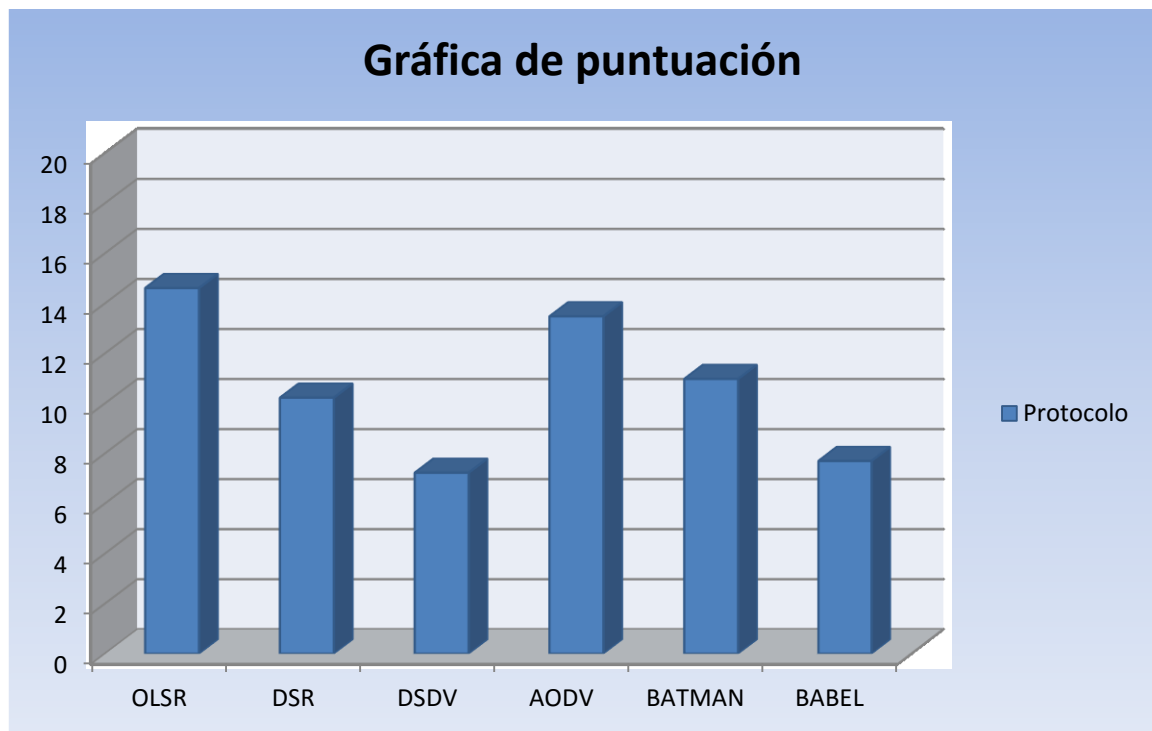
Tabla 5. Porcentaje total del cumplimiento de los protocolos evaluados.

PROTOCOLO	OLSR	DSR	DSDV	AODV	BATMAN	BABEL
TOTAL	14.61	10.23	7.23	13.48	10.98	7.7

Fuente: Autores del proyecto

Para mostrar de manera gráfica los resultados conseguidos en la Tabla 7, se pudo lograr la siguiente figura que muestra la puntuación total del cumplimiento de cada protocolo a las características presentadas por el modelo diseñado en el capítulo anterior:

Figura 12. Porcentaje total obtenido por los seis protocolos evaluados, según el modelo diseñado.



Fuente: Autores del proyecto

Teniendo en cuenta la ficha de evaluación y la tabla de porcentajes totales (Tabla 5) generada anteriormente, se logró establecer que OLSR es el protocolo con mayor grado de cumplimiento, lo que permitió concluir que es el protocolo de mejor desempeño y soporte para redes empresariales.

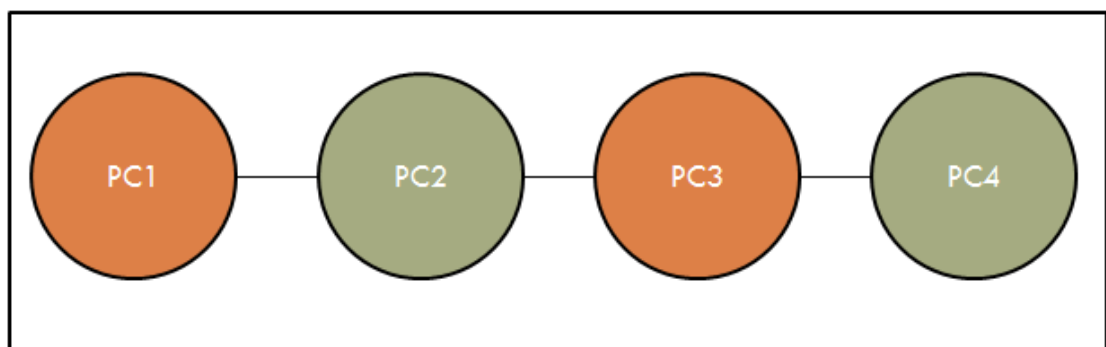
Por lo anterior, OLSR es el protocolo de enrutamiento que se implementará mediante un prototipo para la emulación de su funcionamiento.

4. DISEÑO E IMPLEMENTACIÓN DE UNA RED AD-HOC QUE PERMITE EMULAR EL FUNCIONAMIENTO DEL PROTOCOLO OLSR

En este capítulo se explica el proceso de instalación del protocolo OLSR, el escenario de la red Ad-Hoc y todo lo correspondiente al hardware para la respectiva emulación.

Para la emulación se utilizaron 2 computadores portátiles y 2 computadores de escritorio, a los cuales se les instaló una tarjeta de red inalámbrica LINKSYS y sus respectivos controladores de red. Una vez conseguido el hardware necesario se optó por elegir el Sistema Operativo Ubuntu 14.04 para Linux por su facilidad de instalación, potencia y estabilidad. Posterior a esto se creó la topología que se iba a implementar. Se utilizó un escenario con configuración lineal en el que los nodos de la red solo tienen cobertura con sus nodos vecinos como se muestra a continuación:

Figura 13. Escenario emulación



Fuente: Autores del proyecto

Posterior a esto se implementó OLSR en su versión más estable 6.6.1; el código fuente del proyecto se descargó de la página oficial de OLSR (www.olsr.org). Este protocolo funciona en las siguientes plataformas: Windows (XP y Vista), Linux, OS X, NetBSD, FreeBSD, OpenBSD.

Para su correcta instalación en los computadores fue necesario realizar los siguientes pasos:

- Instalar Bison (generador de analizadores sintácticos) y Flex (un corrector sintáctico) por medio del Gestor de paquetes synaptic de Ubuntu.
- Extraer el software del protocolo con el siguiente comando:
 - # tar -xzf olsrd-0.6.6.1.tar.gz
- Estando dentro de la carpeta olsrd-0.6.6.1 se encuentra el archivo README donde se encuentran los requerimientos y especificaciones para realizar la correcta instalación, el cual puede ser ejecutado por medio del siguiente comando:
 - # cd README
- Se ingresa al modo súper usuario y se digitan los siguientes comandos:
 - # make
 - # make install
- Se edita el fichero /etc/olsrd.conf; aquí residen todos los parámetros configurables del daemon olsrd. En éste se debe modificar la versión del

protocolo IP que se requiere utilizar (en este caso IPv4) y también se especifica el nombre de la interfaz a utilizar (wlan0).

- Cada computador debe estar en el mismo rango de direccionamiento para poder intercambiar información. Por lo tanto lo primero es crear la red ad-hoc asociando los cuatro computadores a un mismo SSID (en este caso pruebafinal), al mismo canal (11) y numero de celda (Cell-ID). También se puede configurar una potencia de salida por medio del comando txpower la cual fue de 10 dBm.

Esto se configura por medio del siguiente comando:

- iwconfig wlan0 essid "pruebafinal" mode ad-hoc channel 11 txpower 10 ap 00:11:22:33:44:55

Se puede verificar la anterior configuración por medio del comando:

```
# iwconfig wlan0
```

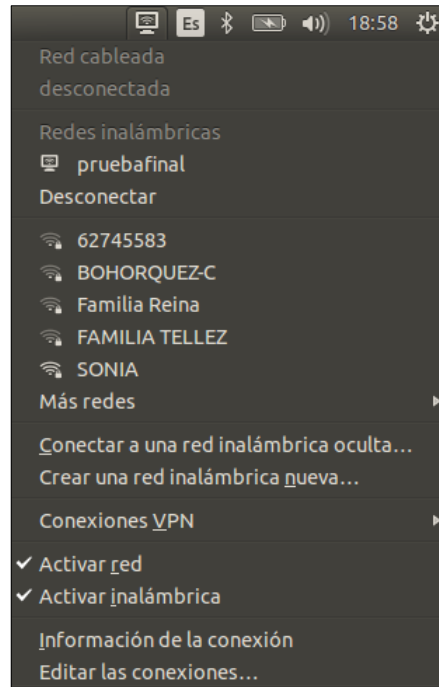
Como se muestra en las siguientes figuras:

Figura 14. Configuración Red Ad-Hoc

```
root@unab-laptop:/home/unab# iwconfig wlan0
wlan0 IEEE 802.11abgn ESSID:"pruebafinal"
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
Encryption key:off
Power Management:off
root@unab-laptop:/home/unab#
```

Fuente: Autores del proyecto

Figura 15. Conexión a la red ad-hoc “pruebafinal”



Fuente: Autores del proyecto

- Cada nodo o computador 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. Se le asignó una dirección IP y máscara de subred a la interfaz de red inalámbrica wlan0 de acuerdo al siguiente esquema de direccionamiento:
 - PC1
 - # ifconfig wlan0 10.1.1.1 netmask 255.255.255.0
 - PC2
 - # ifconfig wlan0 10.1.1.2 netmask 255.255.255.0
 - PC3
 - # ifconfig wlan0 10.1.1.3 netmask 255.255.255.0
 - PC4

- # ifconfig wlan0 10.1.1.4 netmask 255.255.255.0
- Luego se hacen las respectivas pruebas de conectividad por medio del comando ping entre los nodos de la red. Como se muestra en las siguientes figuras:

Figura 16. Conectividad del PC1 con el PC2.

```
root@tesis-PC:/home/tesis# ping 10.1.1.2
PING 10.1.1.2 (10.1.1.2) 56(84) bytes of data.
64 bytes from 10.1.1.2: icmp_seq=1 ttl=64 time=8.71 ms
64 bytes from 10.1.1.2: icmp_seq=2 ttl=64 time=4.84 ms
64 bytes from 10.1.1.2: icmp_seq=3 ttl=64 time=7.18 ms
64 bytes from 10.1.1.2: icmp_seq=4 ttl=64 time=8.22 ms
64 bytes from 10.1.1.2: icmp_seq=5 ttl=64 time=5.83 ms
64 bytes from 10.1.1.2: icmp_seq=6 ttl=64 time=5.26 ms
64 bytes from 10.1.1.2: icmp_seq=7 ttl=64 time=3.24 ms
64 bytes from 10.1.1.2: icmp_seq=8 ttl=64 time=2.91 ms
^C
--- 10.1.1.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7012ms
rtt min/avg/max/mdev = 2.912/5.777/8.719/2.013 ms
root@tesis-PC:/home/tesis#
```

Fuente: Autores del proyecto

Figura 17. Conectividad del PC1 con el PC3.

```
root@tesis-PC:/home/tesis# ping 10.1.1.3
PING 10.1.1.3 (10.1.1.3) 56(84) bytes of data.
64 bytes from 10.1.1.3: icmp_seq=1 ttl=64 time=6.84 ms
64 bytes from 10.1.1.3: icmp_seq=2 ttl=64 time=2.20 ms
64 bytes from 10.1.1.3: icmp_seq=3 ttl=64 time=4.01 ms
64 bytes from 10.1.1.3: icmp_seq=4 ttl=64 time=2.00 ms
64 bytes from 10.1.1.3: icmp_seq=5 ttl=64 time=13.1 ms
64 bytes from 10.1.1.3: icmp_seq=6 ttl=64 time=2.02 ms
64 bytes from 10.1.1.3: icmp_seq=7 ttl=64 time=2.20 ms
64 bytes from 10.1.1.3: icmp_seq=8 ttl=64 time=2.17 ms
64 bytes from 10.1.1.3: icmp_seq=9 ttl=64 time=2.16 ms
^C
--- 10.1.1.3 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8010ms
rtt min/avg/max/mdev = 2.003/4.082/13.105/3.527 ms
root@tesis-PC:/home/tesis#
```

Fuente: Autores del proyecto

Figura 18. Conectividad del PC1 con el PC4.

```
root@tesis-PC: /home/tesis
64 bytes from 10.1.1.3: icmp_seq=9 ttl=64 time=2.16 ms
^C
--- 10.1.1.3 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8010ms
rtt min/avg/max/mdev = 2.003/4.082/13.105/3.527 ms
root@tesis-PC:/home/tesis# ping 10.1.1.4
PING 10.1.1.4 (10.1.1.4) 56(84) bytes of data.
64 bytes from 10.1.1.4: icmp_seq=1 ttl=64 time=29.2 ms
64 bytes from 10.1.1.4: icmp_seq=2 ttl=64 time=3.38 ms
64 bytes from 10.1.1.4: icmp_seq=3 ttl=64 time=3.42 ms
64 bytes from 10.1.1.4: icmp_seq=4 ttl=64 time=3.36 ms
64 bytes from 10.1.1.4: icmp_seq=5 ttl=64 time=3.52 ms
64 bytes from 10.1.1.4: icmp_seq=6 ttl=64 time=3.41 ms
64 bytes from 10.1.1.4: icmp_seq=7 ttl=64 time=3.78 ms
64 bytes from 10.1.1.4: icmp_seq=8 ttl=64 time=4.22 ms
^C
--- 10.1.1.4 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7011ms
rtt min/avg/max/mdev = 3.362/6.799/29.281/8.502 ms
```

Fuente: Autores del proyecto

- Cada nodo se convierte en un dispositivo de enrutamiento y de pasarela hacia otros equipos o redes, lo cual hace necesario activar el reenvío de paquetes del kernel de Linux, este se habilita por medio del siguiente comando:
 - # echo "1" > /proc/sys/net/ipv4/ip_forward
- Teniendo en cuenta estas consideraciones, solo queda ejecutar el comando que arranca el protocolo por medio del siguiente comando:
 - # olsrd -i wlan0

Figura 19. Lanzamiento del protocolo OLSR

```
Archivo Editar Ver Terminal Ayuda
--- 15:35:54.699230 ----- LINKS
IP address      hyst      LQ      ETX
10.1.1.1        0.000    0.886/0.940  1.198
10.1.1.4        0.000    0.607/1.000  1.645
10.1.1.3        0.000    0.497/1.000  2.008

--- 15:35:54.69 ----- NEIGHBORS
      IP address  LQ      NLQ      SYM      MPR      MPRS  will
10.1.1.3        0.000  YES      NO       NO       3
10.1.1.1        0.000  YES      NO       NO       3
10.1.1.4        0.000  YES      NO       NO       3

--- 15:35:54.699385 ----- TWO-HOP NEIGHBORS
IP addr (2-hop) IP addr (1-hop) Total cost
10.1.1.3        10.1.1.1      3.331
                  10.1.1.4      2.772
10.1.1.1        10.1.1.4      3.289
                  10.1.1.3      4.029
10.1.1.4        10.1.1.3      3.210
                  10.1.1.1      2.843
-
```

Fuente: Autores del proyecto

- En este momento las estaciones empiezan a calcular la topología enviándose mensajes entre sus estaciones vecinas.
- También se puede verificar la conectividad de la red ejecutando el siguiente comando, el cual muestra las tablas de ruteo del sistema operativo:
 - # route -n

Como se puede ver en la siguiente figura:

Figura 20. Conectividad de la red – Tabla de ruteo.

```
root@tesis-PC:/home/tesis# route -n
Tabla de rutas IP del núcleo
Destino          Pasarela          Genmask          Indic Métric Ref       Uso Interfaz
10.1.1.0         0.0.0.0           255.255.255.0   U      0      0        0 wlan0
root@tesis-PC:/home/tesis#
```

Fuente: Autores del proyecto

5. CONCLUSIONES

Con base a la literatura revisada se evidenció que existen muchos protocolos de software libre para redes inalámbricas, la mayoría de ellos desconocidos para nosotros como autores del proyecto. Muchos de ellos no tienen soporte, son muy antiguos y se descartaron de entrada. Por ello, para la realización de este proyecto se eligieron los protocolos de software libre que tenían soporte y que se han ido actualizando con el tiempo.

La aplicación de los protocolos de enrutamiento existentes, varía dependiendo de las necesidades de la red, por lo que se debe analizar las ventajas y desventajas, novedades, originalidad e impacto de estos protocolos, ya que el desempeño de éstos varía según las aplicaciones a ser soportadas y las características propias de las redes Ad Hoc, tales como: movilidad, capacidad de procesamiento, ancho de banda, etc.

Al momento de seleccionar una metodología de evaluación de herramientas de software libre se deben tener en cuenta varios criterios, tales como su licencia, el que cuente con un apartado para la evaluación de criterios funcionales, la posibilidad de adaptarla a las necesidades, entre otros; según lo anterior la metodología que más se adaptó a las necesidades del proyecto de grado fue la metodología Qualification and selection of open source software (QSOS).

Mediante la evaluación de los protocolos de enrutamiento realizada por medio del modelo de comparación, se pudo determinar que el protocolo OLSR es el de

mayor grado de cumplimiento a las características evaluadas y por tanto el protocolo que se escogió para emular su funcionamiento.

OLSR es un protocolo proactivo, se adapta bien a redes de alta movilidad, reduce la sobrecarga de mensajes de control y tiene una buena compatibilidad de hardware, ya que este protocolo está disponible para instalar en numerosas plataformas, tanto fijas como móviles.

BIBLIOGRAFÍA

AMCHITE, Federico. CHUNG, Marcelo y AGUIAR, Juan. Evaluación en performance de redes AD-HOC mobiles. En: Google Scholar. 2003.

CALDERÓN, Oscar y QUINTERO, Víctor M. Un nuevo aspecto de la movilidad: redes ad hoc - conceptos. En: Revista Colombiana de Tecnologías de Avanzada. 2004.

HIDALGO, Francisco J. Estudio de viabilidad de la utilización de redes inalámbricas Ad-Hoc en edificios departamentales, publicado en Diciembre de 2008. Recuperado el 19 de Agosto de 2013, de riunet: http://riunet.upv.es/bitstream/handle/10251/13183/TesinaMaster_FcoJavierHidalgo.pdf?sequence=1

HUAMÁN, Jose C. (s.f.). Networking I. Recuperado el 17 de Agosto de 2013, de scribd: <http://es.scribd.com/doc/86039433/40/Desventajas-de-las-Redes-Inalambricas>

MURAZZO, María. RODRIGUEZ, Nelson y MARTINEZ, Matías. Evaluación del retardo de los protocolos de ruteo reactivos para redes MANet. En: Google Scholar.

PALMA, Antonio Manuel. Análisis de protocolos de enrutamiento para redes de sensores inalámbricas. Trabajo de grado ingeniería técnica de telecomunicación. Madrid: Universidad Carlos III de Madrid. Departamento de teoría de la señal y comunicaciones, 2009.

PIERMATTEI, Gerardo R. y GARCIA, Bruno E. Análisis de los protocolos de ruteo OLSR y AODV en redes Ad-Hoc IBSS. Trabajo final de cátedra Redes de Información. Argentina: Universidad Nacional de Río Cuarto. Facultad de Ingeniería, 2011.

RODRIGUEZ, Eduardo. DECO, Claudia. BURZACCA, Luciana y PETINARI, Mauro. Redes inalámbricas de uso comunitario: un análisis comparativo de protocolos. En: Google Scholar. 2000.

RUPÉREZ, Delfín. SANDOVAL, Ana PERAL, Alberto y GARCÍA, Luis. Gestión de riesgos reactiva y chequeo de errores en OLSR. En: Google Scholar.

SAUMETT, Miguel. CASTRO, Harold E. Análisis del desempeño del protocolo de enrutamiento DSR bajo diferentes modelos de movilidad. En: Google Scholar. 2007.

TAQUET, Eduardo Jacob. SAIZ, Purificación y ASTORGA, Jasone. Propuesta de plataforma de evaluación de protocolos de rutado en redes móviles ad-hoc. En: Unión Científica Internacional de Radio (URSI).

TANENBAUM, Andrew S. "Redes de Computadores". Cuarta Edición. Prentice Hall, 2003.

TÉLLEZ, Carlos Felipe. Detección de intrusos y seguridad en redes móviles ad-hoc. En: Google Schollar. 2002.

TROYANO, Albert B. Protocolos de encaminamiento en redes inalámbricas mesh: un estudio teórico y experimental, publicado en Junio de 2011. Recuperado el 20 de Agosto de 2013, de openaccess:
http://openaccess.uoc.edu/webapps/o2/bitstream/10609/8164/1/abatistet_TFM_0611.pdf

TRUJILLO, María Benito. Evaluación experimental de redes malladas basadas en el Protocolo B.A.T.M.A.N. Trabajo de grado ingeniería técnica de telecomunicación. Madrid: Universidad Carlos III de Madrid. Departamento Ingeniería Técnica de telecomunicación, 2010. 187 p.

VARELA, C., & DOMINGUEZ, L. (s.f.). Redes Inalámbricas. Recuperado el 17 de Agosto de 2013, de blyx:
<http://www.blyx.com/public/wireless/redesInalambricas.pdf>