DISEÑO, IMPLEMENTACIÓN Y CARACTERIZACIÓN DE UN PROTOTIPO DE RED TIPO MALLA INALÁMBRICA A PARTIR DE ENRUTADORES LINKSYS WRT54GL

Andrés Felipe Alba Hernández

Javier Fernando Arias Torres

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA

2010

DISEÑO, IMPLEMENTACIÓN Y CARACTERIZACIÓN DE UN PROTOTIPO DE RED TIPO MALLA INALÁMBRICA A PARTIR DE ENRUTADORES LINKSYS WRT54GL

Andrés Felipe Alba Hernández
Javier Fernando Arias Torres

Trabajo de grado para optar por el título de Ingeniero Electrónico

Director
M.Sc. José de Jesús Rugeles Uribe

Codirector
Ing. Ricardo Andrés Díaz Suarez

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA
2010

A mi padre Luis Enrique y mi abuela María Antonia
quienes ya no están presentes,
pero sus enseñanzas siempre estarán en mí.

A mi mamá Isabel que con su tenacidad hizo todo lo necesario
para apoyarme en toda mi carrera universitaria y me dio
la libertad para desarrollarme como persona y profesional.

A mi hermano Sergio Enrique quien siempre ha estado
a mi lado y seguirá acompañándome en la vida.

A mi novia Vanessa por su amor y compañía,
por crecer junto a mí y motivarme siempre para ser mejor.

A mi amigo Javier con quien hemos compartido gran parte de
nuestras vidas y quien me ha acompañado
tanto en los momentos buenos como en los difíciles.

A los profesores y personas que en la primaria y el colegio
formaron los cimientos para llegar a ser lo que soy hoy.

A todos mis familiares y amigos por acompañarme,
hacerme reír, llorar, jugar, aprender y amar
simplemente por ayudarme a vivir.

-Andrés Felipe Alba Hernández-

A Dios por brindarme la salud y bienestar para alcanzar mis sueños.

A mis queridos padres Marco Aurelio y Luz Marina por su sacrificio y apoyo incondicional.

A mis hermanos Rafael, Luz Adriana y Marco Andrés por sus consejos y recomendaciones.

A mi novia Margarita por su amor y motivación para culminar mis metas.

A mi compañero de trabajo y gran amigo Pipe por su ayuda desinteresada en todo momento.

A mi familia y amigos por acompañarme a culminar esta etapa.

-Javier Fernando Arias Torres-

AGRADECIMIENTOS

Para empezar queremos agradecer a nuestros padres por su apoyo en esta y en todas las etapas que hemos tenido que superar para llegar hasta aquí, también a nuestras familias y amigos por estar siempre dispuestos a brindarnos su apoyo. A nuestras novias por su compañía y motivación durante el desarrollo del trabajo de grado.

Dentro de las instituciones vinculadas al proyecto queremos manifestar nuestra gratitud al grupo de Conectividad y Procesamiento de Señal (CPS) dirigido Ph.D. Óscar Gualdrón González por brindarnos las herramientas necesarias para el desarrollo del trabajo, al director del proyecto M.Sc. José de Jesús Rugeles, a nuestro codirector y amigo Ing. Ricardo Andrés Díaz por sus sugerencias y recomendaciones. A Colciencias por el soporte económico sin el que no sería posible el desarrollo del proyecto "Diseño, construcción y caracterización de un enlace autónomo de comunicación wifi para áreas rurales alimentado con energía solar" dentro del que se encuentra enmarcado el presente trabajo.

Finalmente damos gracias a todas las personas que de alguna forma colaboraron en nuestra formación tanto académica como personal para culminar está etapa y afrontar los nuevos retos por venir.

RESUMEN

TITULO: Diseño, implementación y Caracterización de un prototipo de red tipo malla inalámbrica a partir de enrutadores Linksys WRT54GL.[1]

AUTOR: Andrés Felipe Alba Hernández, Javier Fernando Arias Torres.[2]

PALABRAS CLAVES: Redes inalámbricas enmalladas, Protocolos de enrutamiento, Herramientas de análisis de tráfico, Linksys WRT54G.

DESCRIPCIÓN:

En este trabajo de grado se realizó la caracterización de desempeño de los protocolos de enrutamiento BATMAN y OLSR sobre cuatro topologías de red tipo MALLA, a partir de las métricas tasa de transferencia, número de saltos, retardos, *jitter* y potencia. Se documento la funcionalidad de los protocolos de enrutamiento OLSR, BATMAN, BABEL, OSPF y se determinaron sus características en cuanto a la detección de rutas erróneas, reordenamiento dinámico de la red, tiempo de convergencia, cabeceras, tipos de mensajes, tablas, algoritmo de selección de rutas entre otros elementos que permitan entender cómo trabaja el protocolo. Se hizo la comparación de las herramientas de análisis de tráfico Iperf, Netperf y DITG, documentación de las opciones de cada una de ellas.

Se caracterizó la tasa de transferencia respecto a la potencia con dos pruebas: una en ambiente *indoor* y otra en ambiente *outdoor,* los resultados de estas pruebas muestran curvas de comportamiento similares de las que se concluye que para casos prácticos la potencia de emisión no debe superar los 120[mW].

En este trabajo se compararon los firmware disponibles para el linksys WRT54GL de los que se selecciono OPENWRT, se documentó la forma de implementar los protocolos en este y se describen las topologías de las pruebas. Finalmente se presentan los resultados obtenidos para cada topología, basados en las métricas tasa de transferencia tanto para TCP como para UDP, *jitter* y retraso para estas topologías se concluyó que OLSR presenta un mejor desempeño. Queda entonces una experiencia en el grupo CPS y en la Universidad Industrial de Santander para continuar el desarrollo de proyectos con linksys WRT54GL, redes inalámbricas enmalladas, análisis de tráfico en redes o protocolos de enrutamiento.

---

[1] Trabajo de grado

[2] Facultad de ingenierías Físico-Mecánicas

Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones.

Director: José de Jesús Rugeles Uribe. Codirector: Ricardo Andrés Díaz Suárez

SUMMARY

TITLE: Design, implementation and Characterization of a wireless mesh network prototype with Linksys WRT54GL routers.[3]

AUTHORS: Andrés Felipe Alba Hernández, Javier Fernando Arias Torres.[4]

KEY WORDS: Wireless mesh networks, routing protocols, network performance tools, Linksys WRT54G.

DESCRIPTION:

In this project was made the performance characterization of routing protocols BATMAN and OLSR over four mesh network topologies, from the throughput, number of hops, delay, jitter and power metrics. It documents the functionality of OLSR, BATMAN, BABEL and OSPF routing protocols and was determined their specificities regarding the detection of erroneous routes, dynamic rearrangement of the network, convergence time, headers, message types, tables, route selection algorithm and other elements that let understand how the protocol works. We compare the Iperf, netperf and DITG network performance tools, and was documented the options for each one of them.

The characterization of throughput versus emits power was made by two test: one at indoor environment and the other one at Outdoor environment, results of these tests show similar performance curves which concluded that for practical cases the transmission power must not exceed 120 [mW].

In this project were compared available firmware to linksys WRT54GL and was selected OpenWRT, also was documented how to implement the protocols on this routers and describe testbeds topologies. Finally results were presented for each topology and based on throughput over TCP and UDP, jitter and delay metrics and was concluded that for these topologies OLSR delivery a better performance. There remains an experience in the CPS group and Universidad Industrial de Santander to continue the development of projects with linksys WRT54GL, wireless mesh networks, network performance tools and routing protocols.

---

[3] Degree Work.

[4] Faculty of Physic-mechanic Engineering, School of Electrical, Electronic and Telecommunication Engineering.

Director: José de Jesús Rugeles Uribe. Codirector: Ricardo Andrés Díaz Suárez

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

1. CAPITULO UNO – DESCRIPCIÓN DEL PROYECTO

1.1 CONTEXTUALIZACIÓN

Las redes inalámbricas enmalladas son redes ad-hoc implementada con nodos enlazados de manera inalámbrica que trabajan basados en una jerarquía plana, auto configurable, aprendida dinámicamente y con configuración dinámicas de rutas.

La escalabilidad y la fiabilidad de la redes tipo malla han permitido el desarrollo de gran variedad de proyectos como: redes enmalladas metropolitanas, aldeas WIFI y soluciones de conectividad rural. La utilidad de las redes depende de los servicios que sé puedan prestar sobre estas y de la correcta delimitación de la red para aprovecharla al máximo sin saturarla, de esta manera caracterizar la tasa de trasferencia es fundamental para el diseño, implementación y gestión de redes inalámbricas enmalladas.

La familia de estándares 802.11 ha permitido múltiples desarrollos con redes *WLAN*[5], en el momento la mayoría de los equipos se desarrollan bajo el 802.11g y 802.11b, sin embargo dentro de estos estándares no definen apropiadamente los aspectos para implementar redes inalámbricas en topología malla por lo que los fabricantes han creado mecanismos propios para construir las mallas. Ante la necesidad de conseguir estandarización, interoperabilidad de equipos, desarrollo de aplicaciones y servicios apropiados en Enero del 2005 el *IEEE Working Group 802.11*[6] aprobó un *Task Group*[7] denominado 802.11s con lo que se inició el desarrollo de un estándar que busca suplir las necesidades ya mencionadas, el primer borrador D0.01[8] del estándar fue aceptado en marzo del 2006, el D1.00 fue aprobado en noviembre del mismo año bajo la modalidad de voto por carta, el estándar continua en desarrollo pero el hecho que se esté construyendo es evidencia de la relevancia que están tomando las redes enmalladas (1).

Este trabajo de grado está enmarcado en el proyecto de Colciencias **"***Diseño, construcción y caracterización de un enlace autónomo de comunicación wifi para áreas rurales alimentado con energía solar***",** en proyectos para zonas rurales se utiliza la tecnología WIFI ya que es la más económica comparada con otras y es ideal para países en desarrollo como Colombia (2) (3) (4), a su vez parece ser que la mejor forma de distribuir la conexión a Internet en zonas remotas y de escasos recursos es por medio de redes enmalladas (5). El interés

---

[5] Wireless Local Area Network: Red inalámbrica de área local.

[6] Working Group: Grupo de trabajo

[7] Grupo de tarea.

[8] Draft 0.01: hace referencia a primer borrador.

gubernamental y social en proyectos de este corte es un intento por reducir la brecha digital y así aumentar las oportunidades de los individuos en las comunidades en las que se desarrollan los mismos, sin embargo no es suficiente mejorar la conectividad sino que además se deben abarcar aspectos como la capacidad para usar las **TICs** (6)(Tecnologías de la información y las comunicaciones), en este sentido los posibles servicios educativos prestados sobre las redes pueden ayudar a mejorar las competencias de los individuos.

El proyecto pretende realizar un prototipo de red enmallada utilizando enrutadores Linksys WRT54GL y caracterizar el *troughput*, generando una experiencia en el grupo **CPS** (Conectividad y procesamiento de señal) que sirva para el proyecto WIFI-Rural y futuros proyectos con redes enmalladas. Los enrutadores usados poseen firmware de código libre, permiten fácil modificación de hardware, son de bajo costo y muy versátiles para proyectos de este tipo (7), se han utilizado en proyectos de redes enmalladas metropolitanas tales como el proyecto Montevideo libre (8), se encuentra bastante documentación acerca de los mismos y los procesos usados para su modificación.

Finalmente se espera incentivar el desarrollo de nuestra propia red metropolitana para que los ciudadanos tengan acceso a las TICs, un ejemplo que lidera estas iniciativas se encuentra en París consta de una red con más de 400 nodos que brinda conectividad a todos los habitantes y visitantes de la ciudad, generando intercambio cultural, educación y facilitando el desarrollo de ideas y servicios que ayuden a impulsar la economía.

1.2 MOTIVACIÓN Y JUSTIFICACIÓN

Sur América presenta un nivel de penetración a Internet del 36.5% (estadísticas de diciembre del 2009) mientras que el mayor porcentaje esta en Norte América con un 76.2%, en estas cifras se visualiza la brecha digital que afrontamos, si tenemos en cuenta la desigualdad socio-económica que afronta el país se puede concluir que la brecha se hace aún mayor para la mayoría de la población, una forma de afrontar la desigualdad en conectividad que es un aspectos relevante es la ejecución de proyectos de redes metropolitanas y wifi-rural (4). Las redes enmalladas parecen ser la mejor alternativa para desarrollar estas ideas y no se cuenta en la Universidad Industrial de Santander con un trabajo previo en esta área, así este trabajo de investigación pretende dejar como producto experiencia y documentación en la implementación de una red enmallada y la caracterización de la tasa de transferencia en la misma que es necesaria para la correcta gestión de la red.

1.3 OBJETIVOS

1.3.1 OBJETIVOS GENERALES

Diseñar, implementar y caracterizar un prototipo de red tipo malla inalámbrica empleando enrutadores linksys WRT54GL.

1.3.2 OBJETIVOS ESPECÍFICOS

- Realizar un análisis comparativo de los diversos tipos y versiones de firmware existentes para el enrutador Linksys WRT54GL con el propósito de establecer las configuraciones posibles sobre el dispositivo.

- Diseñar y construir un prototipo de| red inalámbrica tipo malla compuesta por módulos que constan de: enrutadores Linksys WRT54GL, baterías de alimentación, antenas omnidireccionales y equipos de medida.

- Evaluar la tasa de transferencia con diferentes protocolos de enrutamiento utilizando herramientas de análisis de tráfico.

- Medir los cambios en la tasa de transferencia ocasionadas por variaciones de potencia radiada en los nodos de la red inalámbrica tipo malla en ambientes *indoor.*

- Documentar los procesos y resultados del estudio para dejar una experiencia en la configuración de redes tipo malla y protocolos de enrutamiento que permita complementar los alcances del proyecto de investigación: "Diseño, construcción y caracterización de un enlace autónomo de comunicación wifi para áreas rurales alimentado con energía solar".

1.4 DESARROLLO DEL PROYECTO

En este proyecto se implemento una red en topología de malla con enrutadores Linksys WRT54GL, se midió el cambio en la tasa de transferencia y el *jitter* cuando se modifica el protocolo de enrutamiento, la potencia de los nodos y distancia de enlace entre otros parámetros. Los resultados y la documentación contribuirán al proyecto de Colciencias *"Diseño, construcción y caracterización de un enlace autónomo de comunicación wifi para áreas rurales alimentado con energía solar"* y a futuros trabajos en redes enmalladas.

En la primera fase se realizó un recopilación bibliográfica de los siguientes tópicos: redes enmalladas o topología malla (WMN), redes de área metropolitana (MAN), WIFI – Rural, redes comunitarias, estándares 802.11a, b, g, estándar 802.16, acceso al medio en redes inalámbricas, algoritmos de planificación, seguridad en redes, los avances del desarrollo del estándar 802.11s, protocolos de enrutamiento, configuración de enrutadores, información referente al enrutador Linksys WRT54GL hardware y firmware, implementación de redes inalámbricas usando este enrutador, medición de la tasa de trasferencia y herramientas para la medición de tasa de transferencia. De esta primera etapa se documentaron los tópicos más relevantes para el proyecto como: enrutamiento en el capítulo 2, firmware y configuración de los enrutadores linksys WRT54GL en el capítulo 3 y herramientas de análisis de tráfico en el capítulo 4. Además esta fase fue el soporte teórico para afrontar el proyecto.

Como segunda fase se analizaron las herramientas de análisis de tráfico y la forma de procesamiento de los datos, la experiencia en esta etapa fue fundamental para el desarrollo de las pruebas.

Después se trabajo en la configuración del firmware para poder implementar los protocolos de enrutamiento, se realizó el diseño del prototipo de un nodo malla, se diseño la topología malla usando filtrado por MAC, y se definió la topología del banco de pruebas. Esto permitió proceder a construir el banco de pruebas y desarrollar las pruebas según la metodología planteada, lo anterior y sus resultados se documentan en el capitulo cinco.

Los resultados, conceptos y las experiencias más relevantes se documentan dando como resultado el presente libro que sirve como referencia para futuros proyectos en el área de redes inalámbricas enmalladas, enrutamiento o temas afines.

2. CAPITULO DOS – CONCEPTOS DE ENRUTAMIENTO

2.1 CONCEPTOS BÁSICOS DE ENRUTAMIENTO

En redes los protocolos determinan como se mueven los datos dentro de ella. Estos se clasifican como IGP (*Interior Gateway Protocol) y* EGP (*Exterior Gateway Protocol)* También se pueden catalogar como protocolos no enrutables (*nonroutable)* y protocolos enrutables (*routable)*, los enrutables por su parte pueden utilizar vector distancia ó estado de enlace (9) (10). Las clasificaciones anteriores ayudan a conceptualizar la forma como se implementan y trabajan los protocolos en redes reales que es el objetivo de este capítulo, a su vez se discutirán las métricas utilizadas por los protocolos para determinar el mejor camino y se analizará el problema de bucles.

2.1.1 INTERIOR GATEWAY PROTOCOL Y EXTERIOR GATEWAY PROTOCOL

Para poder discutir esta clasificación es necesario primero definir que son sistemas autónomos y dominios de enrutamiento.

- **Dominios de enrutamiento:** Este concepto hace referencia a los enrutadores y redes que utilicen el mismo protocolo de enrutamiento.
- **Sistemas autónomos (SA):** Un sistema autónomo está compuesto por un conjunto de dominios de enrutamiento que están conectados dentro de la misma red. Entonces, en un sistema autónomo hay varios protocolos y pueden coexistir diferentes TICs, esto es común en las organizaciones actuales debido a sus divisiones geográficas y administrativas. Determinar el límite de un sistema autónomo es más bien establecer un límite lógico, para decir que hasta allí va la red de determinada organización ó un ISP. Sin embargo, no es posible verlo como un límite físico o técnico.

2.1.1.1 INTERIOR GATEWAY PROTOCOL

Un IGP es cualquier protocolo que pueda trabajar como un dominio de enrutamiento, estos ven o conocen únicamente a las redes y los enrutadores estén dentro del mismo sistema autónomo e intercambian información de enrutamiento solo con aquellos que pertenezcan al mismo dominio, algunos de estos protocolos son: RIP, OSPF y IGRP.

2.1.1.2 EXTERIOR GATEWAY PROTOCOL

Los EGP son protocolos que permiten la comunicación entre **SA** para facilitar el tráfico a través de internet, estos reconocen los **SA** ignorando los IGP, los enrutadores de borde deben tener interfaces con IGP y otras con EGP, el protocolo BGP (Border Gateway Protocol) es el más popular de los EGP.

2.1.2 PROTOCOLOS ENRUTABLES Y NO ENRUTABLES

2.1.2.1 PROTOCOLOS NO ENRUTABLES

No tienen información de la capa de red y están basados en la difusión por lo que agregan tráfico a la red generando ineficiencia. Dos ejemplos de estos protocolos son: DEC`s LAT (Local Area Transport) y NetBEUI (Network Basic Input/Output System).

En muchos casos las redes que utilizan estos protocolos podrían cambiar el enrutador por un dispositivo de capa dos como un switch.

2.1.2.2 PROTOCOLOS ENRUTABLES

Necesitan información de la capa de red para determinar por que ruta o rutas van a enviar los paquetes de datos, utilizan métricas para determinar el mejor camino o los mejores caminos. Se pueden clasificar en los que usan Vector Distancia y los que utilizan estado de enlace.

2.1.3 MÉTRICAS

Son parámetros que permiten comparar una ruta con otra para que el enrutador pueda escoger el mejor camino, basado en una o varias métricas. Las más comunes son: saltos, ancho de banda, retraso, fiabilidad, carga y costo.

2.1.3.1 SALTOS

Hace referencia al número de redes que tiene que atravesar un datagrama para llegar a su destino, no siempre la ruta que tenga menos saltos es la que da una mejor taza de transferencia ya que hay otros factores que influyen, aún así es una métrica útil, en especial si todos los enlaces tienen mas o menos el mismo ancho de banda y confiabilidad y si los enrutadores tienen retardos similares.

2.1.3.2 ANCHO DE BANDA

Como su nombre lo indica aquí el parámetro hace referencia al *ancho de banda digital*, es decir *la cantidad de bits que se pueden enviar por segundo*. Aunque el enlace de mayor

ancho de banda generalmente proporciona mejor taza de transferencia, existen otros factores como los retardos y la fiabilidad que pueden hacer que esto no sea cierto. Por otro lado que una ruta tenga la mejor taza de transferencia no implica que sea la mejor ruta porque para ciertos casos puede ser más importante la disponibilidad del servicio.

2.1.3.3 RETRASO

Esta métrica determina el tiempo que demora un enrutador en procesar, hacer el proceso de cola y sacar por otra interfaz un datagrama.

2.1.3.4 FIABILIDAD

Para poder establecer el valor de esta métrica el enrutador toma una trama especifica de tiempo y en ella observa los problemas reportados por los enlaces, el número de datagramas perdidos, fallas en el enlace y errores de las interfaces, así el enlace que reporte menos problemas en total obtendrá la mejor métrica, es muy útil si lo que se requiere es disponibilidad de servicio.

2.1.3.5 CARGA

En esta métrica lo que se hace es tomar una ventana de tiempo en la cual se mide el tráfico en determinado enlace, representando la máxima carga como 255, es decir el enlace está ocupado en un 100% de su capacidad. Es una métrica útil para hacer balance de carga y en redes malla cobra importancia ya que siempre hay más de una ruta. De esta forma si una ruta está sobrecargada se pueden usar las otras para en definitiva tener una mejor taza de transferencia.

2.1.3.6 COSTO

Es un parámetro que es ajustado por el administrador de la red, permite tomar en cuenta otras cosas como el costo de enviar datos por determinado enlace.

2.1.4 CONVERGENCIA

Hace referencia al tiempo que tarda un protocolo en actualizar la tabla de rutas de los enrutadores de la red encargados del enrutamiento, es un parámetro importante ya que si las actualizaciones no están a tiempo en los enrutadores estos verán una red muy diferente a la en que realmente están inmersos.

2.1.5 TIPO DE SERVICIO

Tos por sus siglas en ingles hace referencia a 8 bits de la cabecera de IP reservados para el tipo de servicio, donde los tres primeros bits indican la importancia del servicio, el cuarto representa el requerimiento de poco retardo, el quinto el requerimiento de alta taza de transferencia y el sexto alta fiabilidad, los dos últimos son usados para el ECN (notificación de congestión explicita).

2.1.6 VECTOR DISTANCIA

El vector distancia intenta determinar a qué distancia se encuentran los enrutadores y así escoger el mejor camino, generalmente utiliza la métrica de saltos para valorar la distancia. Protocolos como RIP v1, RIP v2 y IGRP utilizan vector distancia, los dos primeros usan la métrica de saltos mientras que IGRP toma en cuenta ancho de banda y retardo para calcular la distancia.

2.1.6.1 BUCLES DE ENRUTAMIENTO

Los bucles de enrutamiento son un problema común en los protocolos que trabajan con vector distancia, estos se originan a partir de información errónea en la tabla de rutas o debido a un cambió en la topología que no ha sido actualizada. Este problema hace que paquetes queden atrapados consumiendo recursos y sin llegar a su destino, para solucionar este problema se utilizan algunas técnicas que se mencionan a continuación.

2.1.6.2 CONTEO A INFINITO

Este mecanismo para eliminar bucles utiliza el número de saltos y el TTL, el número de saltos permitidos depende del protocolo, por ejemplo en RIP un paquete puede hacer 15 saltos mientras que en IGRP se le permite llegar hasta 255, después de superar este número el paquete debe ser descartado por el enrutador, por otra parte cuando el tiempo de vida TTL de la cabecera de datagrama llegue a cero este será descartado por los enrutadores, así tenemos dos mecanismos que evitan que el paquete siga indefinidamente por la red, para esto tenemos que tener una idea del tamaño de la red para no limitarla por error ya que por ejemplo si usamos RIP el tamaño se limita a 15 saltos.

2.1.6.3 HORIZONTE DIVIDIDO *(SPLIT HORIZON)*

Los enrutadores no reenvían un datagrama por la misma interface que lo recibieron, evitando bucles y sobrecarga innecesaria a la red.

2.1.6.4 ENVENENAMIENTO DE RUTA

Cuando un enrutador encuentra que por una interfaz ha llegado un datagrama con un número de saltos superior al permitido envía un mensaje en difusión avisando que esa ruta es invalida, en este caso la difusión se hace por todas las interfaces incluso por la que llego el datagrama, es decir se rompe la regla del horizonte de partida, obviamente esta ruta se elimina de la tabla.

2.1.6.5 TIEMPO REFRACTARIO DE CAÍDA *(HOLDDOWN TIMER)*

Si se recibe un mensaje que indique que cierta ruta esta caída se inicia un conteo que normalmente supera el tiempo de convergencia, mientras este conteo no finalice el enrutador no tomara en cuenta información de la ruta caída incluso si llega del mismo enrutador que le indico que estaba caída, después de que este tiempo pase si se recibe un mensaje de la ruta caída entonces se tomara de nuevo como ruta activa y se podrá incluir en la tabla de rutas. Este procedimiento es necesario debido al fenómeno de convergencia ya que puede que lleguen datagramas de la ruta envenenada debido a que algunos enrutadores aún no se han enterado que esta ruta es inválida.

2.1.7 ESTADO DE ENLACE

Los protocolos basados en estado de enlace son más complejos que los que utilizan vector distancia, congestionan menos la red porque no hacen difusión periódicamente y no envían toda la tabla de rutas, en cambio los protocolos de estado de enlace solo envían la información que cambio de la tabla de ruta y lo hacen únicamente cuando un evento ocurre, es decir en el momento que aparecen nuevos enlaces o se cae alguno existente. Además están en capacidad de tomar mejores decisiones de enrutamiento ya que tienen en cuenta más métricas como ancho de banda, retraso, fiabilidad y carga.

Todos los protocolos basados en estado de enlace envían la máscara de subred permitiendo enrutamiento sin clase y brindando así mayor versatilidad al usuario para administrar la red, al estar en capacidad de seleccionar el número de redes y estaciones de las mismas.

2.1.8 TABLAS

Todos los protocolos de estado de enlace deben manejar tres tablas:

2.1.8.1 TABLA DE VECINOS

Se crea a través de los mensajes **"hello"** enviados por cada enrutador para reconocer que enrutadores están conectados a el por cada interfaz.

2.1.8.2 MAPA DE TOPOLOGÍA

También conocida como base de datos de estado de enlace, esta se construye a través de las actualizaciones, se hace un mapa de la topología de todo el dominio de enrutamiento, un enrutador conoce todas las redes y subredes del área y la forma para llegar a ella.

2.1.8.3 TABLA DE RUTAS

Se crea utilizando el algoritmo Dijikstra's SPF *(Short Path First)* dentro del mapa de topología, para así colocar el camino con menor costo en la tabla de ruta.

2.1.9 ACTUALIZACIONES

Cuando una actualización llega a un enrutador este inmediatamente la envía a todos aquellos con los que tenga una relación, así se procesa la actualización al mismo tiempo en todos los enrutadores y con esto el tiempo de convergencia se reduce a diferencia de los protocolos basados en vector distancia que primero procesan la información y luego tienen que esperar a que llegue el momento de difusión periódica para enviar la actualización.

El principal defecto del uso de estado de enlace son los requerimientos de cómputo para seleccionar la mejor ruta y los requerimientos de memoria para guardar las tres tablas.

2.2 PROTOCOLOS DE ENRUTAMIENTO

2.2.1 RFC1583 OSPF

OSPF *(Open Shortest Path First)* es un protocolo de estado de enlace que utiliza el algoritmo de Dijkstra, con OSPF se espera un mejor desempeño que con un protocolo de vector distancia, pero los requerimientos de hardware son más fuertes. OSPF puede ser utilizado en redes grandes y medianas ya que no tiene un número máximo de saltos, se puede denominar también un protocolo de disparo o eventos porque solo envía actualizaciones cuando ocurre algún cambio en la red, reduciendo el tiempo de convergencia, como solo envía información de los cambios que ocurrieron reduce también la sobrecarga de la red (9) (10).

2.2.1.1 CARACTERÍSTICAS DEL PROTOCOLO

- **Multidifusión:** Se utiliza para enviar los paquetes a los enrutadores designados y a los enrutadores de soporte *(backup)*, la dirección IP utilizada para los enrutadores designados es 224.0.0.6 mientras que para los otros nodos OSPF es 224.0.0.5. La Multidifusión reduce el número de sistemas que procesan la información.
- **Rápida convergencia:** Inmediatamente se percibe un cambio en la red se envían los paquetes de actualización que se procesan en paralelo en todos los nodos, reduciendo el tiempo de convergencia y manteniendo la red actualizada.
- **Enrutamiento sin clase:** El protocolo soporta mascara de subred variable, VLSM *(variable length subnet mask*).
- **Soporta varias clases de métricas:** Este protocolo permite implementar varias métricas como ancho de banda, fiabilidad, carga y retardo. Sin embargo no todas las implementaciones de OSPF usan todas las métricas, comúnmente las implementaciones traen por defecto el uso únicamente de ancho de banda.
- **Calidad de servicio:** OSPF permite a los enrutadores direccionar los paquetes basados en un nivel de servicio requerido por la aplicación.
- **Autenticación:** Los nodos pueden usar una clave de protección, intercambiando paquetes únicamente con nodos autorizados que posean la misma clave.
- **Costo de rutas:** Los enrutadores pueden enviar paquetes a través de rutas redundante ya sea en igual cantidad si el costo de las rutas es el mismo o de manera desigual si es diferente, en otras palabras se puede hacer balance de carga.
- **Áreas:** Se puede dividir el dominio de enrutamiento en varías áreas o se puede tomar todo el domino como una única área. Los enrutadores comparten paquetes de actualización únicamente con los nodos de su misma área.

2.2.1.2 BASES DE DATOS

OSPF mantiene tres bases datos para su funcionamiento: la de adyacencia también conocida como tabla de vecinos, base de datos de estado de enlace o mapa de la topología y finalmente la base de datos de direccionamiento o tabla de rutas.

- **TABLA DE VECINOS**

El primer pasó que siguen los enrutadores que tienen implementado OSPF es reconocer los vecinos que están conectados directamente al mismo segmento. Para construir este listado de vecinos el enrutador sigue los siguientes pasos:

i) Se transmiten paquetes hello para darse a conocer con sus vecinos, este mensaje se envía a la dirección 224.0.0.5 en multidifusión.

ii) Los enrutadores OSPF reciben la nueva ruta y la anexan a su tabla de vecinos y al mismo tiempo responden con mensajes hello identificándose ellos mismos.

Los paquetes hello deben tener ciertos parámetros para que los enrutadores los reconozcan y formen una relación de vecinos, estos parámetro serán detallados a continuación cuando se hable de los tipos de paquetes OSPF.

- **MAPA DE TOPOLOGÍA**

Los enrutadores construyen su base de datos de estado de enlace creando un mapa completo de toda la topología de red dentro de su misma área OSPF, identificando cada red y subred conectada y el camino para llegar a ella. Así se crea una estructura de árbol de la red donde la raíz es el nodo que creó la base y las derivaciones corresponden a las rutas para llegar a cada uno de los nodos de su misma área.

- **TABLA DE RUTAS**

La base de datos de direccionamiento se crea a partir del mapa de topología, el algoritmo SPF *(Short Path First)* toma la topología de la red y aplica el algoritmo de Dijkstra para encontrar el camino más corto a cada destino y coloca esta trayectoria en la tabla de rutas.

2.2.2 BABEL

Es un protocolo de vector distancia inspirado en DSDV *(Destination-Sequenced Distance Vector routing)* que se diseño para ser robusto y eficiente tanto en redes cableadas como en redes MANET, es decir en ambientes que utilizan enrutamiento basado en prefijo y en aquellas que usan enrutamiento plano (11).

2.2.2.1 CARACTERÍSTICAS

La principal característica de BABEL es que no causa bucles de enrutamiento, ni agujeros negros durante la reconvergencia, el término agujero negro se refiere a lugares en la red

donde el tráfico entrante es descartado sin informar a la fuente que el paquete no llego a su destino.

Si se detecta un evento en la red BABEL esta hace una rápida reconvergencia con una configuración tal que la red quede libre de bucles y conserve la conectividad, pero esta configuración no es necesariamente la más óptima, la mayoría de las veces este proceso no requiere intercambio de paquetes y en el peor de los casos se requiere un intercambio de paquetes proporcional al diámetro de la red, en este último BABEL converge lentamente a la configuración óptima, el tiempo de convergencia esta en escala de minutos.

2.2.2.2 PROPIEDADES:

- Cuando un prefijo es originado por un enrutador, babel no sufre bucles de enrutamiento.
- Cuando un prefijo es originado por múltiples enrutadores, BABEL debe crear un bucle de enrutamiento transitorio para este prefijo en particular; este lazo desaparece en un tiempo proporcional al diámetro y los enrutadores no vuelven a participar en un lazo de enrutamiento con el mismo prefijo.
- Cualquier agujero negro desaparece en un tiempo proporcional al diámetro de la red luego de que el evento ha ocurrido.

Babel proporciona una estimación de la calidad del canal muy flexible que permite implementar gran variedad de métricas, por ejemplo se puede implementar una métrica basada en la estadística de pérdida de paquetes.

Los nodos de BABEL pueden asociarse en una misma red aunque sus parámetros de configuración sean diferentes, así un nodo sin acceso a la red eléctrica debe usar intervalos largos en tiempo de actualización y mensajes *HELLO* para conservar energía, mientras que uno con un ambiente de alta movilidad debe reducir estos tiempos, dos nodos de este tipo pueden pertenecer a una misma red BABEL lo que hace a este protocolo muy útil en ambientes heterogéneos y redes MANET.

2.2.2.3 LIMITACIONES

Babel envía actualizaciones de la tabla de rutas periódicamente lo que en redes muy estables genera un tráfico innecesario, para estas sería adecuado usar otro protocolo como OSPF.

Por otra parte BABEL impone un tiempo de espera cuando un prefijo es sacado dejándolo inhabilitado por un tiempo, esto hace a inviable el uso de BABEL en redes móviles que implementen prefijo de agregación automática.

2.2.2.4 REGISTRO IANA

IANA (*Internet Assigned Numbers Authority)* ha registrado el puerto número TBD, llamado "babel", para el uso del protocolo BABEL. IANA ha registrado el IPv6 grupo multidifusión TBD y el IPv4 grupo multidifusión TBD para el uso de BABEL.

2.2.2.5 CONSIDERACIONES DE SEGURIDAD

Babel es un protocolo completamente inseguro, un atacante puede atraer el tráfico de datos avisando rutas con baja métrica. Este problema se puede resolver con mecanismo de seguridad en capas inferiores o utilizando criptografía con los paquetes BABEL, la opción de ignorar los datos contenidos en un paquete de Babel más allá de la longitud del cuerpo declarado por la cabecera se ha diseñado para tal propósito.

Por otra parte con la información de los anuncios realizados por BABEL en todo el dominio de enrutamiento, es posible predecir en algunos casos y con una precisión razonable la localización física de un nodo, para reducir este problema se puede usar un cambio periódico del router-id y la dirección IP.

2.2.2.6 OPERACIÓN DEL PROTOCOLO

Cada nodo en una red BABEL debe tener un ***router-id***, es decir identificación de nombre que es una cadena de 8 octetos que debe ser única en todo el dominio de enrutamiento.

Los nodos BABEL intercambian mensajes del protocolo BABEL, uno o varios de estos mensajes conforman un paquete BABEL que es enviado en un datagrama UDP.

La fuente de un paquete BABEL siempre es una dirección unidifusión pero los paquetes pueden ser enviados tanto en unidifusión como en multidifusión, a excepción de los mensajes *Hello* y *ACK* todos los mensajes BABEL pueden ser interpretados por un nodo sin necesidad de determinar la dirección de destino del paquete.

BABEL aplica un nivel moderado de ***Jitter*** a sus mensajes, un mensaje que va a salir es guardado en el buffer por un tiempo con un pequeño retardo aleatorio. Lo anterior se realiza con dos propósitos: el primero es eliminar la sincronización en los nodos a la hora de hablar a través de la red y el segundo es dar tiempo para que se puedan incluir varios mensajes en un solo paquete.

El retraso aplicado depende también de la urgencia del mensaje ya que está sujeto a todos los tiempos que se mencionan a continuación, por ejemplo si un mensaje esta cerca del tiempo especificado para "morir" pues debe ser enviado rápidamente.

2.2.3 OLSR RFC3626

OLSR (*Open Link State Routing Protocol*) es un protocolo diseñado para redes ad-hoc móviles conocidas como redes MANET[11]. Los nodos OLSR utilizan la información local para hacer el enrutamiento ya que esta no queda centralizada sino que se difunde por los MPR, por este motivo el protocolo funciona bien en redes grandes con la misma configuración de una red pequeña (12).

2.2.3.1 RELACIONES ENTRE NODOS

Establecer algunas relaciones entre nodos es importante para analizar OLSR:

- **Vecino:** Un nodo **A** es vecino de un nodo **B**, si **A** puede escuchar al nodo **B** y viceversa.
- **Vecino a dos saltos:** Es aquel nodo que puede ser escuchado por un vecino.
- **Vecino a dos saltos estricto:** Un nodo que es vecino de un vecino.
- **Multipoint Relay (MPR):** Es un nodo que ha sido elegido por su vecino para retransmitir todos los mensajes de difusión que recibe del nodo que lo selecciono como **MPR**.
- **MPR Selector (MS):** Son todos los nodos que tienen a dicho nodo como MPR, es decir todos los que lo seleccionaron.

2.2.3.2 PAQUETE OLSR

La estructura básica de cualquier paquete OLSR es la que se presenta a continuación (13).

<table>
<tr><td>1 BYTE</td><td>2 BYTE</td><td>3 BYTE</td><td>4 BYTE</td></tr>
<tr><td colspan="2">Tamaño del paquete</td><td colspan="2">Número de secuencia de paquete</td></tr>
<tr><td>Tipo de Mensaje</td><td>Tiempo V</td><td colspan="2">Tamaño del mensaje</td></tr>
<tr><td colspan="4">Dirección de origen</td></tr>
<tr><td>Tiempo de vida</td><td>Número de Saltos</td><td colspan="2">Secuencia del mensaje</td></tr>
</table>

Mensaje		
Tipo de Mensaje	Tiempo V	Tamaño del mensaje
Dirección de origen		
Tiempo de vida	Número de saltos	Número de secuencia del mensaje
Mensaje		

.

.

.

TABLA 2.1 ESTRUCTURA DE DATOS DE UN MENSAJE OLSR.

2.2.3.3 TABLA. ESTRUCTURA BÁSICA PAQUETE OLSR

A continuación se describe cada campo de la estructura anterior:

- **Tamaño del paquete:** El tamaño en bytes del paquete.

- **Número de secuencia del paquete (PSN):** Es un número que se incrementa cada vez que un nuevo paquete OLSR es transmitido. Las secuencias de paquetes se toman por cada interfaz, es decir todos los paquetes transmitidos por una interfaz están secuencialmente enumerados.

- **Tipo de mensaje:** En un rango de 0 a 127 especifica que tipo de mensaje está en el campo "mensaje". Los números del rango que no se usan son para futuras extensiones.

- **Tiempo V:** El tiempo de validez indica por cuanto tiempo un nodo receptor debe considerar la información contenida en el mensaje como valida, sin haber obtenido una actualización más reciente.

- **Tamaño del mensaje:** Este campo da el tamaño del mensaje desde el inicio del campo "tipo de mensaje" hasta el comienzo del siguiente campo "tipo de mensaje".

- **Dirección de origen:** Contiene la dirección del nodo que genero el mensaje sin importar los nodos intermedio por los que paso.

- **Tiempo de vida:** Este campo contiene el número de saltos que un mensaje puede ser transmitido. Antes de transmitir un mensaje el nodo decrementa en uno este

valor, si un mensaje tiene tiempo de vida 0 o 1 no es transmitido. Este número permite al nodo escoger su campo de acción.

- **Conteo de saltos:** Es el número de saltos que un mensaje ha tenido, el nodo de origen sitúa este valor en cero y los otros lo van incrementando en uno.

- **Número de secuencia del mensaje:** Cada mensaje tiene un único número de secuencia que es colocado por el nodo que lo origino, de esta forma se sabe que un mensaje no es retransmitido más de una vez.

2.2.3.4 MULTIPOINT RELAY (MPR)

Cada nodo debe seleccionar MPRs que serán los encargados de llevar todo el tráfico de control de la red, si un nodo envía un mensaje en difusión este es redireccionado únicamente por sus MPRs. Un nodo selecciona su MPR ó MPRs de tal forma que pueda llegar a través de ellos a todos sus vecinos a dos saltos. Para poder realizar la selección de los MPRs es necesario tener información acerca de los vecinos y los vecinos a dos saltos, para esto se utilizan los denominados paquetes HELLO.

2.2.3.5 PAQUETES HELLO

Estos paquetes permiten conocer los vecinos, tipos de vecinos y estado de los enlaces.

Si un enrutador A inicia envía un mensaje HELLO de inicio a un nodo B, este devuelve un mensaje HELLO con la dirección del enrutador A, finalmente el nodo A envía un mensaje HELLO anunciando que el enlace es simétrico.

Los mensajes HELLO contienen la lista de todos los vecinos del nodo, de esta manera los nodos que reciben el mensaje conocen que vecinos a dos saltos tienen a través de del enrutador que origino el mensaje HELLO. Además cada mensaje HELLO indica si el enrutador de origen seleccionó a este vecino como MPR.

Después de procesar los paquetes HELLO cada enrutador debe tener su lista de vecinos y el estado de enlace de cada uno de ellos (13).

<table>
<tr><td>0</td><td>1</td><td>3</td><td>4</td></tr>
<tr><td>0123456789</td><td>0123456789</td><td>123456789</td><td>123456789</td></tr>
<tr><td colspan="2">Reservado</td><td>Tiempo H</td><td>Willigness</td></tr>
<tr><td colspan="4">Dirección de la interfaz vecina</td></tr>
</table>

Dirección de la interfaz vecina		
. . .		
Código enlace	Reservado	Tamaño del mensaje
Dirección de la interfaz vecina		

TABLA 2.2 ESTRUCTURA DE UN PAQUETE HELLO.

- **Reservado:** Este campo debe ser llenado con ceros para concordar con las especificaciones del protocolo.

- **Tiempo H:** Espacio que especifica el intervalo de mensajes HELLO usado por el nodo en esta interfase en particular. El tiempo de los mensajes es representado por su *mantissa* (cuatro bits más significativos) y por su *exponent* (cuatro bits menos significativos)

- **Código de enlace:** Campo que especifica información acerca del enlace entre las interfaces del emisor y la lista de interfaces de vecinos. Además de la información acerca del estado del vecino.

Si el código de enlace es menor o igual a quince este se puede interpretar en los siguientes dos campos de dos bits.

7	6	5	4	3	2	1	0
0	0	0	0	Tipo de vecino		Tipo de enlace	

TABLA 2.3 INTERPRETACIÓN CÓDIGO DE ENLACE

- **Tipo de enlace:**

  - ***UNSPEC_LINK:*** Indica que no se especifica información del enlace.
  - ***ASYM_LINK:*** Indica que el enlace es asimétrico.
  - ***SYM_LINK:*** Indica que el enlace es simétrico en la interfase.
  - ***LOST_LINK:*** Indica que los enlaces no se ven o se han perdido.

- **Tipo de vecino:**

  - ***SYM_NEIGH:*** Indica que el vecino tiene como mínimo un enlace simétrico con este nodo.
  - ***MPR_NEIGH:*** Indica que el vecino tienen como mínimo un enlace simétrico y ha sido seleccionado como MPR.

- ***NOT_NEIGH:*** Indica que los nodos no han llegado a ser vecinos simétricos aún.

Cuando una combinación no es lógica el paquete se descarta, por ejemplo si esta marcado SYM_LINK como tipo de enlace y a su vez NOT_NEIGH como tipo de vecino, entonces el paquete es descartado ya que si tiene un enlace simétrico por lo menos tiene un vecino.

- **Tamaño de Mensaje:** Define el tamaño del mensaje, contado en bytes y medido desde el inicio del campo "código de enlace" hasta el próximo campo de "código de enlace".

- **Dirección de interface de vecino:** Como su nombre lo indica la dirección de dicho vecino por esa interfaz.

2.2.3.6 MODO DE OPERACIÓN OLSR

OLSR usa datagramas UDP para transmitir la estructura de paquetes ya descrita en los numerales anteriores.

2.2.3.6.1 PROCESAMIENTO DE PAQUETES

Los paquetes que llegan a un nodo son manejados de la manera que se describe a continuación:

a) Si el paquete no cumple con un tamaño mínimo se asume que no contiene mensaje y se descarta.
b) Si el valor del TTL es menor o igual a cero se descarta.
c) Se decide que proceso ser realiza al paquete dependiendo del tipo de mensaje que contenga, si el mensaje es duplicado no se procesa. Un mensaje duplicado es aquel que tiene el mismo número de secuencia e IP origen.
d) La retransmisión de un paquete depende del algoritmo del tipo de mensaje.

2.2.3.6.2 PERCEPCIÓN DEL ENLACE

Los enlaces se perciben a través de los paquetes HELLO descritos en secciones anteriores, si estos se escuchan por ambos nodos el enlace sería simétrico, si solo uno de ellos lo escucha sería asimétrico.

2.2.3.6.3 DETECCIÓN DE VECINOS

Durante la percepción del enlace un nodo recibe mensajes HELLO y de allí crea una tabla con todos los vecinos de los cuales recibió mensajes HELLO y el estado de enlace de cada uno de ellos. Esta tabla se modifica cuando un estado de enlace cambia, si no se recibe información de un enlace durante un tiempo establecido entonces el enlace se borra de la tabla. Esta detección de vecinos es fundamental para el proceso ya descrito de selección de MPRs.

2.2.3.6.4 DIFUSIÓN DE MENSAJES DE CONTROL DE TOPOLOGÍA

Los mensajes TC *(Topology Control)* contienen lo necesario para el cálculo de las rutas, dentro de ellos está la información de todos los enlaces de tal forma que los vecinos saben a donde llegan desde dicho nodo. Estos mensajes son difundidos de un nodo a todos sus MPRs.

2.2.3.6.5 CÁLCULO DE RUTAS

Los enrutadores deben tener una tabla de enrutamiento basada en las tablas información de enlace y de la topología, que es actualizada cuando alguno de los siguientes campos varía: Enlace, vecino, vecino a dos saltos o topología.

2.2.3.6.6 INTERFACES NO OLSR

Existen nodos que tienen interfaces de otro dominio de enrutamiento, para tener conectividad con estas se introduce información externa de enrutamiento a través de mensajes HNA (Host and Network Association).

2.2.3.6.7 NOTIFICACIÓN DE CAPA DE ENLACE

Aún cuando OLSR no trabaja con la información de capa de enlace, si esta se tiene es incluida en los mensajes HELLO para mantener dicha información tanto de los vecinos como los MPRs.

2.2.3.6.8 INFORMACIÓN REDUNDANTE DE LA TOPOLOGÍA

Los mensajes de anuncio HNA incluyen información de enlaces hacia nodos vecinos que no necesariamente tienen a este nodo como MPR. El mensaje de anuncio contiene información de todos los enlaces de los nodos vecinos.

Existen tres niveles de redundancia:

- Sin redundancia: Sólo se emite información del grupo que ha elegido el nodo como MPR.
- Redundancia media: Se emite información del grupo que ha elegido el nodo como MPR y también información de los nodos que este ha elegido como MPR.
- Redundancia alta: Se emite información de todos los enlaces hacia los vecinos.

2.2.3.6.9 MPR REDUNDANTE.

Esta funcionalidad especifica la habilidad del nodo de seleccionar MPR redundantes. Aunque la redundancia crea mucho más tráfico y pierde eficiencia el mecanismo de MPR, se tiene una gran ganancia al asegurar la llegada de los paquetes a sus destinos. Esta funcionalidad es útil para situaciones en que la red tiene mucha movilidad ya que mantiene una buena cobertura con los MPR.

2.2.3.6.10 MÉTRICA

El RFC del protocolo no especifica la métrica a usar, simplemente menciona un estado de enlace lo que deja abierta la posibilidad de la métrica, algunas implementaciones usan número de saltos pero como ya se menciono esta métrica no siempre es la más adecuada, en implementaciones de Freifunk y Open Wrt la métrica usada es ETX *(Espected Transmition Count)*.

ETX es una medida de la calidad del enlace en una red inalámbrica, que determina el número espera de transmisiones necesarias para enviar un paquete sin ningún error a su destino. El número varía de uno a infinito, tener uno como valor del ETX indica una perfecta transmisión tener infinito representa un enlace caído. Esta métrica representa una probabilidad basada en los eventos pasados y por este motivo es un número real.

ETX = [número de transmisiones / número de paquetes recibidos]

2.2.4 B.A.T.M.A.N.

B.A.T.M.A.N. (*Better Approach To Mobile Ad-hoc Networking)* es un protocolo de enrutamiento proactivo para redes inalámbricas enmalladas Ad-hoc, incluidas las redes móviles. Este protocolo contiene información acerca de todos los nodos a los que puede acceder directamente o a través de múltiples saltos. Sin embargo un nodo en particular no conoce la topología total de la red y utiliza únicamente la información local para determinar las rutas, así el enrutamiento está distribuido en toda la red. Con datos locales se selecciona el mejor siguiente salto para cada destino de la red sin que sea necesario calcular la ruta completa, esto hace más rápido y eficiente el protocolo. B.A.T.M.A.N. es un protocolo que corre sobre UDP y utiliza el puerto 4305 asignado por la IANA (14).

2.2.4.1 DETECCIÓN DEL CAMINO:

Como ya se menciono el mecanismo para seleccionar rutas usa únicamente la información local, el proceso se basa en unos mensajes denominados OGM que son enviados por todos los enrutadores en difusión, la idea básica del proceso es que las mejores rutas de mayor ancho de banda, mas fiabilidad y menor retardo serán las que lleven los mensajes OGM a una mayor rata. Así el vecino por el cual se presente la mayor rata de mensajes OGM será seleccionado como el mejor salto para llegar a un determinado destino y entrara en la tabla de rutas sin preocuparse de la ruta completa para llegar al destino, por eso se denomina enrutamiento distribuido. Cada mensaje OGM es único ya que lleva una IP origen y un número de secuencia, así cada nodo cuenta únicamente una vez este mensaje y evita que mensajes duplicados afecten el cálculo del estado del mejor vecino. Sin embargo vale la pena resaltar que no todos los mensajes OGMs son aceptados, eso depende de la configuración realizada en la denominada "acceptance-function" que define reglas para saber si aceptar o no un OGMs, está función también es la encargada de evitar bucles y otros problemas relacionados con enrutamiento, además de darle flexibilidad al diseñador de redes para escoger los parámetros que cree relevantes a la hora de seleccionar una ruta.

2.2.4.2 FORMATO DE PAQUETES B.A.T.M.A.N.

Originator Message
HNA Mensajes (Opcional)
…

TABLA 2.4 FORMATO GENERAL DE PAQUETES.

Originator Message (OGM): Es un campo de doce octetos. Que contiene el OGM descrito más adelante.

HNA Mensajes (Opcional): Este campo de 5 octetos está asociado al mensaje OGM que lo procede. Su descripción se amplía más adelante.

ORIGINATOR MESSAGE (OGM)

Versión U D	TTL	Bandera GW
Número de Secuencia	Puerto GW	
Dirección de Origen		

TABLA 2.5 FORMATO MENSAJE OGM

- **Versión:** Si este campo contiene una versión diferente a la que está corriendo el enrutador entonces este ignora el mensaje.

- **D:** Indica si el mensaje proviene de un vecino, es decir de un enlace directo.

- **U:** Indica si el nodo vecino es bidireccional o no.

- **TTL:** Indica el tiempo de vida del mensaje y determina el número de veces que un mensaje OGM será retransmitido.

- **Bandera GW:** Banderas de puerta de enlace.

- **Número de secuencia:** Un número que se incrementa con cada mensaje OGM enviado con el fin de que este sea único.

2.2.4.3 FORMATO MENSAJE HNA

Dirección de Red
Mascara subred….

TABLA 2.6 FORMATO MENSAJE HNA

- Mascara subred de la red anunciada.
- Dirección IPv4 de la red anunciada.

2.2.4.4 LISTA DE ORIGINADORES

Este listado contiene todas las interfaces de las que se recibieron mensajes OGM, si un nodo tiene más de una interfaz y se recibieron OGMs por varias de ellas entonces este estará en la lista varías veces, una por cada interface. En la lista se mantiene la siguiente información por cada nodo:

- **IPV4 del Originador:** Contiene la dirección de la interface por la que se origino el mensaje.
- **Tiempo última actualización:** Contiene la fecha y hora de la última actualización recibida.
- **Número de secuencia del enlace bidireccional:** Es necesario grabar el número de secuencia del OGM para hacer luego el proceso de comprobación de enlace bidireccional.
- **Número de secuencia actual:** El nuevo número de secuencia del OGM que ha sido aceptado.
- **Lista HNA:** Todas las redes anunciadas por el originador con su rango IP y su máscara subred.
- **Capacidades de Puerta de enlace:** Este campo se usa si el originador es puerta de enlace y anuncia sus parámetros.

- **Lista de información de vecinos:** Esta lista contiene los vecinos con la siguiente información de cada uno.
    - **Ventana deslizante:** Un número de secuencia dentro de la ventana es marcado si un OGM es recibido.
    - **Conteo de paquete:** La cantidad de números de secuencia de la ventana.
    - **Ultimo tiempo validado:** La fecha y hora de último paquete recibido por este vecino.
    - **Ultimo TTL:** El TTL del último OGM recibido a través de este vecino.

2.2.4.5 DIFUSIÓN DE LOS MENSAJES OGM

Cada nodo genera periódicamente los mensajes OGM por todas sus interfaces, estos se originan cada cierto tiempo determinado por el denominado ORIGINATOR_INTERVAL por todas sus interfaces B.A.T.M.A.N, se aplica un retardo aleatorio en estos intervalos para evitar colisiones. Los mensajes inicializan de la siguiente manera:

- **Versión:** La versión del protocolo.
- **Banderas:** Se activa la bandera de directamente conectado y en cero la de unidireccional.
- **TTL:** Se coloca en el tiempo deseado entre el tiempo de vida máximo y mínimo.
- **Número de secuencia:** Es un valor arbitrario del primer mensaje que se incrementa con cada nuevo mensaje.
- **GWFlags:** Esta bandera se activa únicamente si el enrutador ofrece conexión a internet.
- **GWPort:** Si el enrutador ofrece conexión a internet este indica el puerto por el cual se llevara la conexión, si no ofrece internet este campo estará en cero.
- **Dirección del Originador:** Este campo se completa con la dirección IP de la interface que origino el mensaje.

2.2.4.6 RECEPCIÓN DE MENSAJES OGM

Al recibir un mensaje se realiza el siguiente proceso:

1. Si se recibe un mensaje OGM que contiene una versión diferente a la interna el mensaje se descarta.
2. Si la dirección del mensaje corresponde al a de una de las interfaces del nodo este se descarta.
3. Si la dirección del emisor es una de la direcciones de difusión de alguna de las interfaces entonces este se ignora.
4. Si la dirección de los originadores es idéntica a la de cualquiera de las interfaces del nodo entonces este se procesa de manera especial como se explica más adelante y luego se descarta.
5. Si la bandera unidireccional esta activada entonces el mensaje se descarta.

6. Si el mensaje OGM es recibido vía un enlace bidireccional y contiene un nuevo número de secuencia entonces los OGM deben procesarse como se describe más adelante.
7. El mensaje OGM es redifundido como se describe más adelante si:

    - El mensaje OGM es recibido de un vecino a un salto.
    - El mensaje OGM fue recibido vía un enlace bidireccional, vía el mejor enlace y no esta duplicado.

2.2.4.7 COMPROBACIÓN DE BIDIRECCIONALIDAD DEL ENLACE

Este mecanismo es utilizado para comprobar que un enlace lleva información en ambas direcciones. Por este motivo el número de secuencia y la interface de un OGM es guardado si:

- El mensaje se recibe por la misma interface que fue originado.
- La bandera de directamente conectado está activada.
- El número de secuencia es marcado con el número del último mensaje OGM difundido por dicha interface.

2.2.4.8 RANKING DE VECINOS

Al recibir OGM de un nodo se realiza lo siguiente:

- El conteo de paquetes se actualiza.
- Si el número de secuencia OGM es mayor que el que está guardado en el nodo entonces:
    - El nuevo número de secuencia de la interfaz se graba con el número que contiene este mensaje.
    - El último tiempo TTL del mensaje es actualizado.
    - La ventana deslizante de todos los enlaces conocidos del originador debe ser actualizada.
- Si la ventana deslizante de el enlace por el cual es recibido el OGM contiene el número de secuencia mayor ósea el mejor ranking entonces este es seleccionado como el mejor enlace, en caso contrario el mejor enlace no cambia.

2.2.4.9 REDIFUSIÓN DE OGM A OTROS NODOS

La redifusión un mensaje implica el cambio de algunos campos del mensaje, los cambios son los siguientes:

- El TTL debe disminuir en uno. Si el TTL se hace cero el paquete debe ser ignorado.

- El campo de directamente conectado debe estar activado si fue recibido por un vecino directo y redifundido por dicha interface.
- La bandera unidireccional debe ser activada si se hace redifusión del OGM y fue recibido por un enlace bidireccional.

2.2.4.10 ENRUTAMIENTO

El demonio de B.A.T.M.A.N crea una lista donde mantiene todos los originadores de mensajes OGM, la ya mencionada lista de Originadores. De la misma manera el demonio de B.A.T.M.A.N mantiene una entrada de enrutamiento para cada originador conocido y para cada anuncio HNA. Estas entradas tienen las interfaces de salida y la dirección IP del siguiente salto a través del cual debo llegar al originador.

2.2.4.11 SELECCIÓN Y ESTABLECIMIENTO DE RUTAS

Si un OGM o un HNA de un originador desconocido es recibido este se adhiere a la tabla de rutas, y el mejor vecino de enlace bidireccional para dicho vecino es seleccionado como la puerta de enlace a dicho destino. Si un mejor enlace vecino cambia su ranking la tabla de rutas es actualizada.

2.2.4.12 ELIMINACIÓN DE LA TABLA DE RUTAS

En caso de no recibir mensajes OGM o HNA de un originador conocido durante un periodo de tiempo superior al intervalo de salida y el tamaño de ventana la ruta se considera expirada y se elimina de la tabla. Se recomienda normalmente seleccionar el intervalo de salida aproximadamente diez veces el tamaño de la ventana por el tamaño del intervalo originador.

2.2.4.13 ANUNCIOS DE PUERTA DE ENLACE

Un nodo con conexión a internet tiene debería actuar como puerta de enlace a internet. Esta es anunciada con GWFlags transmitidas con los paquetes OGM. Si un nodo no provee acceso a internet esta bandera estará en cero. En caso contrario este campo contiene el valor aproximado del ancho de banda del acceso a internet.

0	1	2	3	4	5	6	7
S	DESCARGA				SUBIDA		

TABLA 2.7 CAMPO GWFLAGS

Ancho de banda de descarga = 32 x (S + 2) x $2^{DESCARGA}$ [kbits/s]

Ancho de banda de subida = ((up+1) x (Ancho de banda de descarga)) / 8 [Kbits/s]

2.2.4.14 ENCAPSULACIÓN Y TUNNELING DE LA PUERTA DE ENLACE

Los clientes encapsulan toda la información enviada al nodo puerta de enlace, sin embargo los paquetes que vienen de internet a los clientes no son encapsulados. El encapsulamiento consiste en un datagrama IP/UDP, el nodo GW reconoce el datagrama guiado por el número de puerto de la cabecera UDP.

2.2.4.15 SEGURIDAD

B.A.T.M.A.N. es un protocolo que recibe información de otros nodos y por lo tanto es vulnerable a ataques, por lo que se recomienda complementarlo con codificación y tecnologías de autenticación. Como B.A.T.M.A.N. no conoce toda la topología de la red entonces recibe paquetes de cualquier fuente lo que dificulta la confidencialidad y por lo cual es fundamental implementar otros mecanismos de protección. Dos vulnerabilidades graves de B.A.T.M.A.N. es que un nodo malicioso cree sobre flujo de OGM o manipule las rutas a través de OGM, el diseñador de redes debe tener en cuenta esto.

3. CAPITULO TRES – CONFIGURACIÓN E IMPLEMENTACIÓN DE PROTOCOLOS EN LINKSYS WRT54GL

3.1 HARDWARE LINKSYS.

3.1.1 ALIMENTACIÓN

Todas las versiones del WRT54G requieren la misma alimentación de 12V DC. 1A, exceptuando la versión 1.0, con su fuente de poder de 5V DC 2A, esto debido, a que esta primera versión de la serie contaba con la compatibilidad de PoE (Power over Ethernet) en donde la fuente de alimentación proviene de los pares de cable no usados del cable de Ethernet (7).

3.1.2 ARQUITECTURA DEL PROCESADOR

Toda la serie WRT54G implementa procesadores Broadcom MIPS (Microprocessor without Interlocked Pipeline Stages), que cuentan con RISC (Reduce Instruction Set Computer), es decir un set de instrucciones reducido, en el cual el tiempo de ejecución de las mismas es menor, y donde también se disminuye el número de accesos a la memoria. Es importante señalar que la mayoría del software libre basado en Linux como es el caso del WRT54G está hecho para la plataforma intel x86, que implementa otra arquitectura basada en un set de instrucciones más complejos CISC (Complex Instruction Set Computing), de la que desciende la mayoría de procesadores de computadores de escritorio (7). Debido a lo anterior se hizo necesario la incursión de un Cross Compiler, el cual es un compilador capaz de crear código ejecutable para una plataforma diferente a la que se está corriendo, esto con el fin de que el firmware sea soportado por el procesador del Router.

El enrutador WRT54G se compone principalmente de 3 funciones, el procesamiento, control de acceso al medio de Ethernet (Ethernet MAC), y MAC del wireless.

A través de los distintos lanzamientos de la serie WRT54G, se observa la implementación de dos familias de procesadores Broadcom: BCM47xx y BCM5352.

BCM47xx: esta serie está compuesta por dos versiones, el BCM4704 en el cual este integrado solo cumplía la labor de procesamiento con una velocidad de reloj de 125MHz, mientras que el chip Broadcom BCM2050 se encarga del radio wireless, por otra parte, inicialmente el procesador ADMTec ADM6996 se encargo de la MAC de Ethernet, para más adelante en la versión 2.2 del Router ser reemplazado por el chip Broadcom BCM5325.

Con la inclusión de la versión BCM4712 se incluyo en un solo integrado la CPU y la MAC del wireless con el sistema SoC (System on a Chip), manteniendo por separado en esta familia el integrado de la MAC de Ethernet, además de incrementar la velocidad de procesamiento a 200MHz.

La segunda familia de procesador, el Broadcom BCM5352 hizo su primer aparición en la versión 4.0 del Router, en la que su principal novedad fue la inclusión de la nueva generación de la arquitectura SoC, ya que incorporo las 3 funciones de CPU, MAC del Ethernet y wireless MAC en solo integrado, manteniendo la velocidad del procesador en 200MHz.

3.1.3 ALMACENAMIENTO

La capacidad de almacenamiento como parte fundamental del funcionamiento del dispositivo, ya que es ahí donde se almacena todo el software del equipo, es decir el firmware, está a cargo del tipo de memoria no volátil, específicamente de memoria flash, siendo las versiones 1,2 y 3 del WRT54GS las de mayor capacidad con 8MB. La versión Linksys WRT54GL que la usada en el presente proyecto tiene una capacidad de memoria de 4MB.

3.1.4 MEMORIA RAM.

Los modelos del router WRT54G usan memoria SDRAM (Synchronous Dynamic Random Access Memory), es decir memoria RAM de acceso síncrono que se conecta al reloj del sistema, con la capacidad de leer o escribir a un ciclo de reloj por acceso sin estados de espera intermedios. Esta memoria esta soldada directamente sobre la tarjeta principal con una capacidad de 16MB en la versión inicial del router.

3.1.5 SERIE LINKSYS WRT54G

El firmware linksys WRT54G se basó en Linux por lo que el código fuente tuvo que ser liberado para cumplir con la licencia GNU GPL, abriendo la posibilidad a terceros de modificarlo y crear funciones extras, esto fue lo que hizo interesante a la serie WRT54G llevando a la compañía incluso a lanzar el WRT54GL, versión que fue creada para facilitar el *"Hacking".* Esta versión permite retirar las antenas, tiene espacio en la PCB para montar un cable JTAG en caso de daño del firmware, espacio para instalar un puerto serial y espacio para ampliar la memoria con una memoria externa. El diagrama de bloques y una fotografía del dispositivo se presentan a continuación.

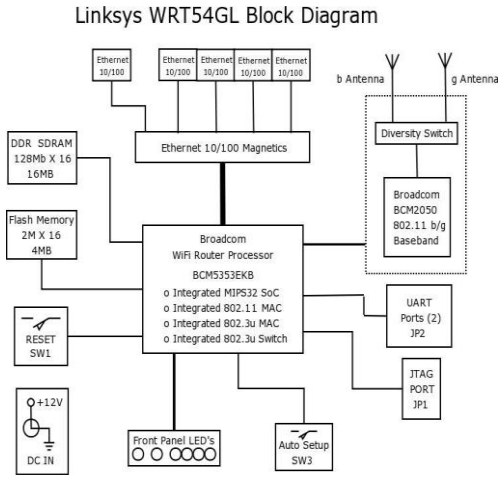


FIGURA 3.1 DIAGRAMA DE BLOQUES DEL LINKSYS WRT54GL[9]

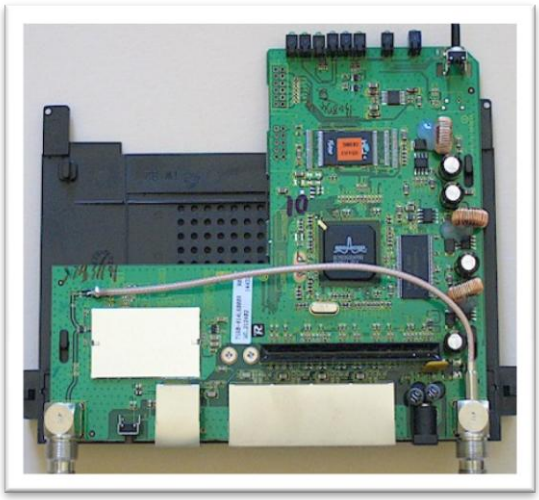


FIGURA 3.2 LINSYSWRT54GL

Como se observa en las dos figuras anteriores el linksys WRT54GL implementa un integrado SoC (System-on-Chip), tiene cinco puertos Ethernet, el diversity chip y la memoria.

En los siguientes párrafos se hace una breve descripción de algunos firmwares que sirve para visualizar cual sería el apropiado según el uso que se le espera dar a los enrutadores.

---

[9] Diagrama extraído de: Paul Asadoorian, Larry Pesce, "Linksys WRT54G Ultimate Hacking", Syngress Publishing, Inc.

3.2 FIRMWARE.

El firmware es un bloque de instrucciones con un fin específico y directamente ligado a un hardware, se encuentra en memoria no volátil y es el encargado de controlar a más bajo nivel el hardware, en otras palabras este funciona de interfaz entre el hardware y el software, un ejemplo de firmware es el que se monta sobre la BIOS de un computador y sobre este se soporta el sistema operativo.

3.2.1 INSTALACIÓN

La instalación de un firmware puede realizarse a través de la interfaz Web o por TFTP (*Trivial File Transfer Protocol*), la primera de ellas no es soportada por todos los firmware, además la interfaz Web debe guardarse en memoria y si este recurso es limitado en el enrutador, entonces se debe evaluar que tan útil sería tenerla, por otro lado es mucho más sencillo hacer cambios a través de la Interfaz Web. Finalmente es necesario saber instalar el firmware a través de TFTP ya que si el firmware está dañado es la única forma de hacerlo, existe otra manera de instalar el firmware, a través de JTAG *(Joint Test Action Group)* Sin embargo requiere modificaciones de hardware y no se discutirá por el momento.

- ***Instalación por interfaz Web:*** Es la manera más sencilla de realizar la instalación del firmware, simplemente se busca en la interfaz una pestaña que indique actualización de firmware o en ingles *firmware upgrade* después se debe dar la dirección donde se encuentra el código, a continuación se presenta una imagen de este procedimiento usando el firmware que viene de fabrica.

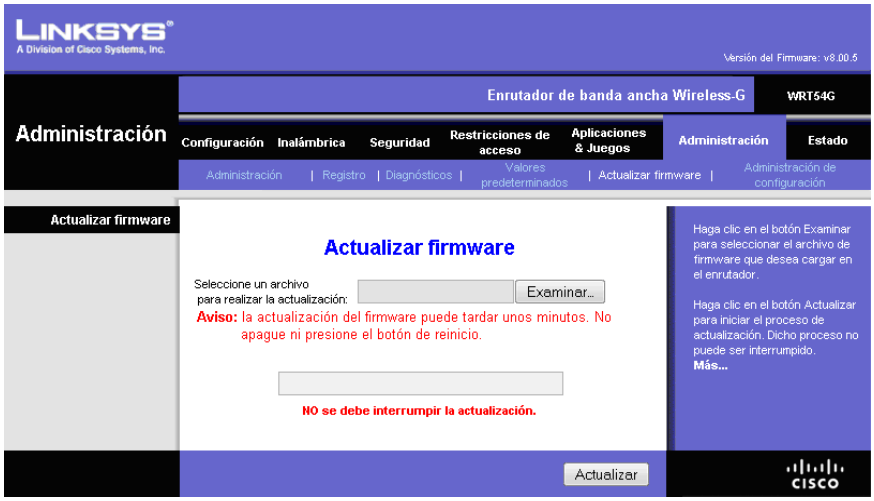


FIGURA 3.3 ACTUALIZACIÓN FIRMWARE POR INTERFAZ WEB.

- ***Proceso de arranque:*** Cuando el linksys arranca el PMON (versiones 1 y 1.1) ó CFE que son cargadores de arranque toman el control del proceso de arranque,

esta revisa si existen particiones en la NVRAM (Memoria no volátil de acceso aleatorio) si no existen entonces las realiza usando los valores almacenados en el cargador de arranque, después revisa el parámetro *boot_wait* si este parámetro esta encendido es decir esta en *on*, el espera conexiones de un servidor TFTP por un tiempo predeterminado que comúnmente es de tres segundos, a continuación realiza un chequeo de redundancia cíclica sobre el firmware, si es correcto el arranque se ejecuta normalmente, de lo contrario el cargador de arranque esperara hasta que un nuevo firmware se reciba vía TFTP. Cuando esto último sucede el enrutador hace varias cosas que vale la pena mencionar, crea algunas IP con las que hace cosas como responder solicitudes ARP para la dirección 192.168.1.1, escuchar mensajes ARP de difusión y responder a la solicitud ICMP de ping. Por otra parte cuando se inicia el modo de *boot_wait* el inicia TFTP sin necesidad de contraseña. Una ventaja de los cargadores de arranque es que difícilmente se dañan porque no se permite escritura en la partición de la NVRAM en que se encuentra este.

- ***Instalación del firmware por TFTP:*** Para instalar el firmware vía TFTP se aprovecha el *boot_wait* que es el estado en el que el enrutador esperara una conexión de servidor TFTP, sin embargo no siempre esta habilitada esta opción así que se pueden usar tres técnicas para activarlas según la versión y los conocimientos del usuario: Ping Hack, a través de líneas de comando del sistema operativo ó como última opción actuando directamente sobre el cargador de arranque a través de puerto serial. Existen diferentes comandos para TFTP y depende del sistema operativo que se utilice, esto se debe tomar en cuenta siempre que se quieran hacer modificaciones a través de esta vía.

3.2.2 VERSIONES DE LOS FIRMWARE

3.2.2.1 FIRMWARE ORIGINAL

Basado en Linux sirvió como base para crear otros, no es *hackable*, soporta WPA-PSK y WPA2-PSK, solo algunas versiones soportan WPA (Wifi protected access) y WEP (wireless encryptation protocol). Este firmware es adecuado para usuarios promedio que simplemente quieren que su enrutador funcione de inmediato y no están interesados en hacer modificaciones (15).

3.2.2.2 VXWORKS

Es la última versión de firmware, es de tipo propietario y no es basado en Linux, no es muy versátil para el *Hacking* y al igual que el original es solo para usuarios promedio, ya que no presenta opciones interesantes (16).

3.2.2.3 DD-WRT

Creado por BrainSlayer esta basado en la versión original del firmware, algunas características que lo diferencian del original son: la posibilidad de incrementar la potencia de transmisión, soporta portal cautivo vía wifidog, soporta WDS *(Wireless Distribution System)*, puente o tipo malla y calidad de servicio tanto por las interfaces Wireless como LAN (7).

Las distintas versiones del firmware soportan varias características presentadas en la siguiente tabla:

Feature	Micro	Mini	Standard	Standard_ nokaid	VoIP	VPN
Chillispot			X	X	X	X
HTTPS Web management			X	X	X	X
IPv6			X	X	X	X
kaid			X			
MMC/SD card support			X	X		X
NoCat			X	X	X	X
OpenVPN						X
PPTP/PPTP client		X	X	X	X	X
radvd	X	X	X	X		
RFlow			X	X	X	X
Samba client			X	X	X	X
SIPatH					X	
SNMP			X	X	X	X
SSH		X	X	X	X	X
UPnP		X	X	X	X	X

TABLA 3.1 CARACTERÍSTICAS VERSIONES DD-WRT[10]

Descripción de las características:

- ***Chillispot:*** Un portal cautivo que permite autenticar usuarios antes de que se conecten a la red.
- ***HTTPS Web Management:*** Permite administrar desde interfaz Web, se recomiendan activar desde allí encriptación SSL *(secure sockets layer)*.
- ***Ipv6:*** Es una nueva versión de IP que permite más direcciones, mayor seguridad entre otras características.

[10] Extraída del libro: Syngress Linksys WRT54G Ultimate Hawking.

- ***Kaid:*** Soporta Kai un método desarrollado para que usuarios de consolas como Xbox, PlayStation2, PlayStation portable y Nintendo GameCube entre otros puedan jugar en línea.
- ***MMC/SD card support:*** Permite almacenamiento extra vía tarjeta SD.
- ***NoCat:*** Es un portal cautivo como Chillispot.
- ***OpenVPN:*** Para realizar conexiones VPN esta es una buena opción.
- ***PPTP client:*** Es una tecnología similar a VPN pero no es tan segura ni flexible se basa en el protocolo PPTP *(Point to point tunneling protocol)*
- ***Radvd:*** Para usuarios con Ipv6 este programa envía advertencias de enrutamiento.
- ***RFlow:*** Es una solución de monitoreo remoto para routers que corran DD-WRT.
- ***Samba client:*** Este software permite conectar a almacenamiento compartido en Windows y servidores Samba vía SMB *(Server Message Block)*
- ***SIPatH:*** Un servidor de voz sobre IP para sistemas embebidos con Linux.
- ***SNMP:*** *Simple Network Management Protocol* permite usar dispositivos de monitorio y hacer cambios en la configuración.
- ***SSH:*** *Secure Shell* es un protocolo de la capa de red que permite acceder por comandos pero tiene características de seguridad como encriptación.
- ***UPnP:*** Es un protocolo para realizar pequeños cambios al cortafuegos por comunicación con sus clientes y reglas del mismo, se debe usar cuidadosamente porque no se conocen detalles de las implicaciones que tiene sobre la seguridad.

Finalmente es necesario decir que el DD-WRT tiene gran cantidad de características al alcance de la mano y muchas se pueden modificar fácilmente por su interfaz Web, sin embargo es el firmware que más recursos consume, el cual se puede descargar de su portal web (17).

3.2.2.4 EWRT

Este firmware usa la misma interfaz gráfica del firmware Original, agrega varias características que no posee el original, dentro de las que esta el anexar portales cautivos para funcionar como hotspot, la principal diferencia es permitir conexión SSH, la mayoría de las características adicionales deben trabajarse por ventana de comandos, es ideal para trabajar enrutadores que están en zonas lejanas (7). Desde la versión Ewr-0.4.4 se suspendió el desarrollo del proyecto hasta nuevas noticias o a la espera de ser adoptado, sin embargo se dejo como soporte todos los archivos y paquetes correspondientes a esta última versión (18).

3.2.2.5 FAIRUZAWRT

Fue creado para probar el concepto de *diferent attack vector* sobre una red basada en seguridad de ataques, de esta prueba se generaron mas versiones creando así un firmware enfocado a la seguridad, FairuzaWRT tiene muchas de las características de los firmware de código abierto, pero vale la pena recalcar las herramientas para reconocimiento y ataque que posee (7):

- ***Arping:*** Una herramienta de ping basada en ARP.
- ***Arpspoof:*** Es parte de un dsniff, usado para hacer suplantación de IP por falsificación de la tabla ARP.
- ***Dnsspoof:*** Parte también de un dsniff, esta herramienta permite interpretar y redireccionar solicitudes DNS.
- ***Dsniff:*** Es como un sniffer pero además de capturar los paquetes los decodifica, puede hacer sniffing protocolos de clear-text[11] incluyendo nombres de usuario y claves, incluyendo Telnet, TFP y POP3.
- ***Fping:*** Permite hacer ping on steroids, soporta multitarea de lectura y salidas que pueden ser fácilmente cambiadas a otras estructuras más sencillas para el análisis.
- ***Fping6:*** Permite los mismo que Fping pero para Ipv6.
- ***Fprobe:*** Una prueba para mirar el flujo de la red
- ***Hexdump:*** Una utilidad para pasar archivos binarios a hexadecimal y formato ASCII.
- ***Hping2:*** Un paquete para transmisión TCP/IP con herramientas para la elaboración de análisis.
- ***Hydra:*** Una herramienta para escuchar redes y testear gran cantidad de servicios como: Telnet, FTP, http, HTTPS, http-Proxy, SMB, SMBNT, MS-SQL, MySQL, REXEC, RSH, RLOGIN, CVS, SNMP, SMTP-AUTH, SOCKS5, VNC, POP3, IMAP, NNTP, PCNFS, ICQ, SAP/R3, LDAP2, LDAP3, Postgres, Teamspeak, Cisco auth, Cisco enable, y Cisco AAA.
- ***Mailsnarf:*** Parte también de un dsniff, es un sniffer para clear-text en protocolos de correo electrónico.

---

[11] Clear–text hace referencia a todo lo que este escrito en un formato que pueda ser entendido directamente por un ser humano sin procesamiento, prácticamente se refiere a texto sin encriptar.

- ***Msgsnarf:*** Dentro de un dsniff, permite hacer sniffer de aplicaciones de mensajería instantánea en clear-text.
- ***Nbtscan:*** Un escáner de nombres de servidores NetBios.
- ***Netcat:*** Una herramienta de lectura y escritura de datos a través de redes TCP/IP.
- ***Nmap:*** Un escáner de puerto y herramienta de identificación por servicio remoto.
- ***Sshmitm:*** Una herramienta para lanzar ataques SSH MITM *(man in the middle)*, parte de un dsniff.
- ***Tcpdump:*** Un sniffer por línea de comando.
- ***Tcpkill:*** Usa suplantación TCP/IP para terminar sesiones de red.
- ***Tcpnice:*** Una herramienta para controlas la velocidad de conexiones TCP vía *spoofing* (suplantación).
- ***Urlsnarf:*** Dentro del dsniff, es útil para reunir URLs sobre la red.
- ***Webmitm:*** También del dsniff, una herramienta para hacer ataques MITM *(man in the middle)* http y SSL.

FairuzaWRT solo permite modificaciones por línea de comando, además la sesión de Telnet inicia activada y se debe deshabilitar por seguridad, en parte esto muestra que los que quieran usar este potente firmware deben tener conocimientos para usarlos, es una forma también de prevenir que cualquiera pueda usar una herramienta tan potente que en manos equivocadas puede ser desastrosa, por otro lado el tener una interfaz Web que solo muestra el estado de los ajustes ahorra espacio en memoria. A continuación se presenta una imagen del menú desplegado al establecer una sesión SSH.

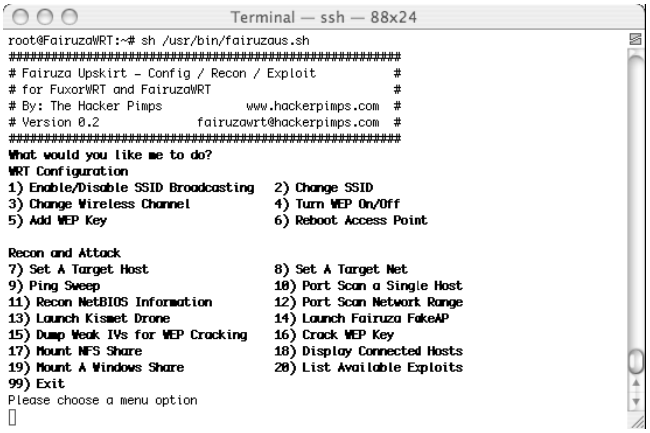


FIGURA 3.4 CONEXIÓN SSH CON FIRMWARE FAIRUZAWRT

- ***Aircrack y Aircrack-ng:*** Aircrack fue la herramienta más popular para escuchar redes inalámbricas, permite hacer sniff del tráfico inalámbrico y crackear claves WEP, el Aircrack-ng es una versión actualizada del Aircrack y contiene nuevas características y algunas mejoras en velocidad, la nueva versión permite hacer crack también de WPA-PSK, para dar una idea de lo que puede hacer podemos decir que se puede crackear una clave de 128 bits WEP en menos de 60 segundos.

Este firmware es ideal para profesionales en seguridad computacional, profesionales en pruebas de penetración y también para aquellos que desean acceder a una red con malos propósitos el cual se puede descargar de su portal web junto con su documentación (19).

3.2.2.6 SVEASOFT

Inicio con distribución gratuita pero ahora requiere una licencia comercial de 25 dólares el año, se realizaron varias versiones cada una con fines específicos denominadas talismán (20).

- ***Talisman Basic:*** Tiene las características básicas esperadas en el WRT54G y incluye interfaz Web y potencia inalámbrica transmitida.
- ***Talisman Micro:*** Reduce el tamaño del firmware, ideal para aquellos que solo tienen 2 MB de memoria, pero sacrifica las funcionalidades de PPoE, PPTP y DHCP.
- ***Talisman VPN:*** Agrega servidores IPSec y PPTP, es dedicado a crear VPN.
- ***Talisman Hotspot:*** Ideal para Hotspot como su nombre lo indica, tiene el portal cautivo del EWRT y además tiene su propio RADIUS *(Remote Authentication Dial in User Service)* y servidor SQL, además soporta RADIUS externo, RADIUS es un protocolo de autenticación y autorización para aplicaciones de acceso a la red o movilidad IP.
- ***Talisman Mesh:*** Ideal para crear redes tipo malla, para trabajar como proveedor inalámbrico o una comunidad basada en redes inalámbricas.
- ***Talisman VOIP:*** Esta versión basa su inhalación dentro de un SIP *(Session Initiation Protocol)* Usted puede usar este firmware para terminar localmente conexiones SIP o enviarlas a un proveedor VoIP/SIP.

Este firmware permite también la modificación de la potencia de la antena que por defecto esta en 50 mW la máxima potencia a la que se debe subir el radio es 89 mW sin embargo el firmware permite sobrepasar este valor, el máximo que alcanzan los modelos WRT54G, WRT54GL y WRT54GS es 251 mW pero después de 89 mW se corre el riesgo de sobrecalentamiento y posibilidad de incendio (7).

3.2.2.7 HYPERWRT

Este firmware se baso en gran parte del firmware original del enrutador, añadiendo como principal funcionalidad modificar la potencia transmitida razón por la cual tomo popularidad, además el aumento de la potencia ajustaba como un porcentaje, así el 100% equivale a 84 mW evitando poner en riesgo el equipo, sin embargo también es posible introducir el valor de la potencia manualmente y allí si se puede llegar a 251 mW cualquier valor por encima de este es rechazado, mantiene una interfaz web similar al original, permite 14 canales posibles del 1-11 para USA, 12-13 para Europa y el 14 para Japón. Adicionalmente este firmware provee un robusto sistema de QoS y Wake On Lan (WOL) (7).

El desarrollo del firmware fue detenido desde el año 2006, eliminando incluso el portal web oficial de soporte, no obstante se puede encontrar versiones de este firmware en servidores de otras comunidades.

3.2.2.8 FREIFUNK

El Firmware de Freifunk nace como iniciativa de software libre que tiene como fin soportar las redes inalámbricas libres en la región Alemana, pero su uso se ha extendido a diversas iniciativas libres alrededor del mundo. El archivo utilizado para los enrutadores linksys WRT54GL es _g+gl/ que se encuentra en los repositorios de Freifunk en la web (21).

El firmware está basado en OpenWrt se pueden instalar nuevos paquetes sin tener que esperar nuevas versiones del firmware, tiene la posibilidad de realizar sesiones SSH para implementar funcionalidades por comandos. Freifunk permite hacer varias modificaciones desde la interfaz web facilitando al usuario la modificación de parámetros de enrutamiento desde la interfaz. Una de las principales ventajas del Firmware freifunk es que ya trae instalado el OLSRD *(Optimized Link State Routing Protocol)* que fue diseñado para redes Ad – Hoc móviles MANET por sus siglas en ingles, se de los repositorios puede descargar una actualización que permite el uso de BATMAN también desde interfaz web.

3.2.2.9 OPENWRT

Existen diferentes paquetes del mismo con distintas características, tienen en común que todos permiten escoger entre dos tipos de sistemas de archivo ***SQUASHFS*** y ***JFFS2*** (*Journaling Flash File System),* la primera de estas es de solo lectura y tiene características interesantes como el borrado automático de archivos duplicados. Permite el uso de SSH y es un firmware ideal para personas con conocimientos en Linux ya que solo las últimas versiones tienen interfaz Web, permite hacer *Hacking* aunque la mayoría de las modificaciones deben hacerse por línea de comando, estos paquetes además soportan varias aplicaciones de las comunidades de software libre como: ***Wifidog*** que es un software que crea su portal cautivo y permite administrar hotspots que son puntos de

acceso inalámbrico publico, ***BlueZ Bluetooth driver subsystem*** y ***Snort*** que es un software de código abierto para seguridad y detección de intrusos, a pesar de su seguridad algunos paquetes tienen activado el demonio de Telnet por lo que se debe modificar el script por seguridad y evitar que se pueda abrir una conexión de este tipo que sería insegura (7).

Revisado lo anterior respecto a firmwares disponibles para los enrutadores linksys WRT54GL, teniendo en cuenta sus capacidades, cualidades y el soporte que entrega la comunidad de desarrollo (22) detrás de cada uno por lo que se decidió optar por el firmware OPENWRT para el desarrollo del proyecto de redes inalámbricas.

Todo ese desarrollo del firmware se ve reflejado en un mayor número de paquetes disponibles en los repositorios del OPENWRT lo cual da la posibilidad de instalarle y añadirle más funcionalidades entre ellas la gestión de redes inalámbricas tipo malla, como también en la publicación de nuevas versiones del firmware que ayudan a corregir errores de estabilidad y agregan mejoras como la inclusión de la interfaz Web luci.

Finalmente se presenta una tabla comparativa de los firmwares estudiados, aunque otros firmware presentan otras características interesantes el OpenWrt es el único que permite usar todas las opciones de protocolos de enrutamiento, además consta de buena documentación, actualizada y el proyecto aún está en marcha por lo que cada día se generan nuevas características que pueden ser fácilmente descargadas de los repositorios.

Funcionalidad	Firmware Original	VXWORKS	DD-WRT	EWRT	FAIRUZAWRT	SVEASOFT	HYPERWRT	FREIFUNK	OPENWRT
Basado en:	Linux	Propietario	Linux	Linux	Linux	Linux - Costo	Linux	Linux	Linux
Interface web	x	x	x	x	-	x	x	x	x
Requerimiento memoria flash	Normal	Normal	Alta	Normal	Baja	Normal	Normal	Normal	Normal
Telnet	x	x	x	x	x	x	x	x	x
SSH	-	-	x	x	x	x	x	x	x
Documentación	Manual	Manual	Muy Buena	Proyecto cerrado	Buena	Básica	Proyecto cerrado	Muy Buena	Muy Buena
Repositorios	No existen	No existen	Buenos	aceptables	Buenos	Solo con licencia	Cerrados	Buenos	Buenos
RIP	-	-	Interfaz LAN	-	-	-	-	-	x
OSPF	-	-	Interfaz LAN	-	-	-	-	-	x
OLSR	-	-	-	X	-	-	-	x	x
BATMAN	-	-	-	-	-	-	-	Corre sobre OLSR	x
BABEL	-	-	-	-	-	-	-	-	x
Variacion Potencia del radio	-	-	x	-	x	X	x	x	x
Instalados durante el proyecto	x	-	x	-	-	-	-	x	x

TABLA 3.2 COMPARACIÓN FIRMWARES.

3.3 INSTALACIÓN OPENWRT

La manera más fácil de instalar el firmware es desde la interfaz web original, en *Administración / actualizar firmware* y pulsar sobre el botón examinar se despliega la

ventana que permite buscar el archivo .bin que contiene el instalador del firmware. La versión de firmware a instalar es la 8.09.2 para el *kernel* 2.4 que se encuentra en la sección de descargas del portal web ***www.openwrt.org*** de la comunidad openwrt quienes desarrollaron el firmware. El proceso de instalación dura algunos minutos y durante este tiempo no se debe apagar el dispositivo.

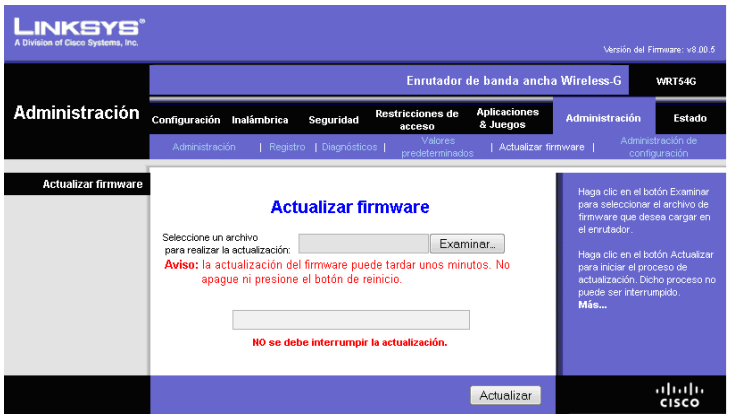


FIGURA 3.5 ACTUALIZAR FIRMWARE.

3.3.1 CONFIGURACIÓN DE LAS INTERFACES.

3.3.1.1 INICIO.

Después de la correcta instalación del firmware se accede por medio de un explorador de internet al portal de inicio de OPENWRT que generalmente se encuentra en la dirección 192.168.1.1, para poder modificar los parámetros predefinidos en el firmware se debe acceder a la pestaña de administración que se encuentra en la parte superior derecha. Esta solicita el usuario y la contraseña que por defecto es root y admin respectivamente.

FIGURA 3.6 INICIO OPENWRT.

Aunque la interfaz gráfica facilita las modificaciones en la configuración es necesario implementar comandos en la consola del router por medio de protocolos como Telnet y SSH. Por defecto el puerto habilitado para acceder a consola es el de Telnet mientras que el puerto SSH no está habilitado. Para mayor seguridad se deben utilizar sesiones SSH en lugar de Telnet, es necesario cambiar la contraseña por defecto para habilitar el puerto para SSH, esto a su vez deshabilita el puerto de Telnet. El cambio de contraseña se realiza después de presionar Administration, vamos a la pestaña System y finalmente Admin Password tal como se presenta en la siguiente figura.

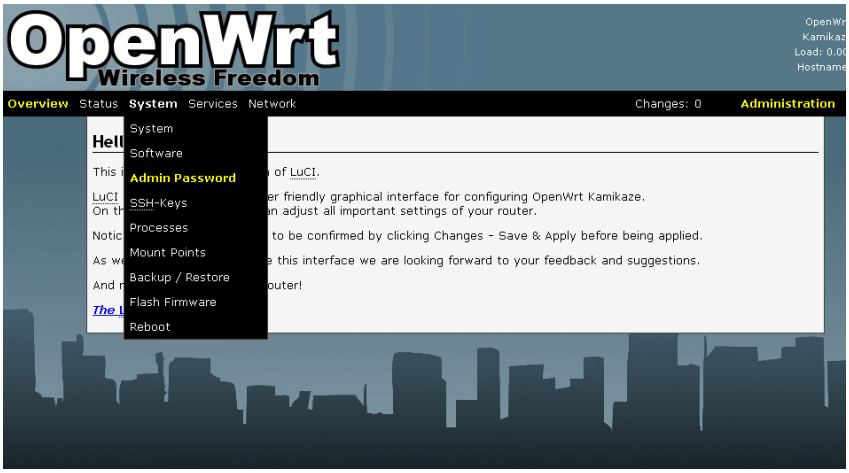


FIGURA 3.7 CAMBIO CONTRASEÑA

Primero se revisa que interfaces están creadas en enrutador, para lo que se accede a la pestaña *Network/Interface*, aquí deben estar WAN, LAN y WIFI, si alguna de estas falta se debe crear usando el botón *Add entry,* es importante escoger el dispositivo (*Device)* adecuado tal como se presenta en la siguiente figura.

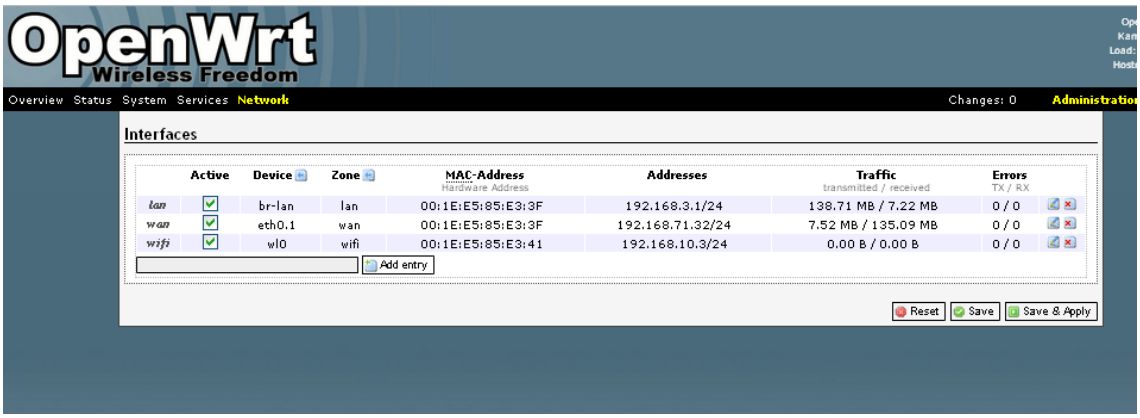


FIGURA 3.8 REDES OPENWRT

Otro paso importante es colocar una dirección IP de manera lógica con la arquitectura de red, para la red LAN se utilizo la siguiente lógica para asignar IP, la dirección de la interfaz cableada es 192.168.***x***.1 donde la ***x*** corresponde al número del nodo y para la interfaz inalámbrica 192.168.10.***x***, esta modificación se realiza fácilmente en la interfaz gráfica.

3.3.1.2 LAN.

Para la red cableada se dirige a la pestaña *Network/Interfaces/LAN* tal como se muestra a continuación, el ejemplo se realiza con el nodo 3.

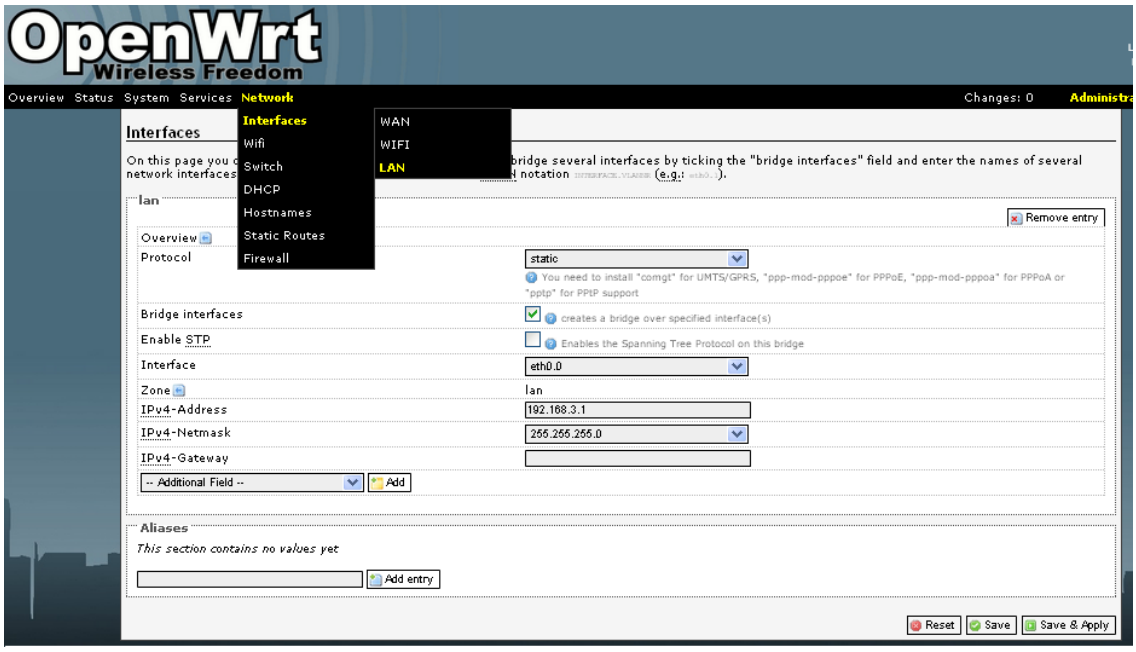


FIGURA 3.9 CONFIGURACIÓN RED DE ÁREA LOCAL OPENWRT.

3.3.1.3 INALÁMBRICA

De la misma manera para la interfaz inalámbrica se configuran los campos como se presenta ahora, accediendo a las modificaciones por *Network/Interfaces/WIFI.*

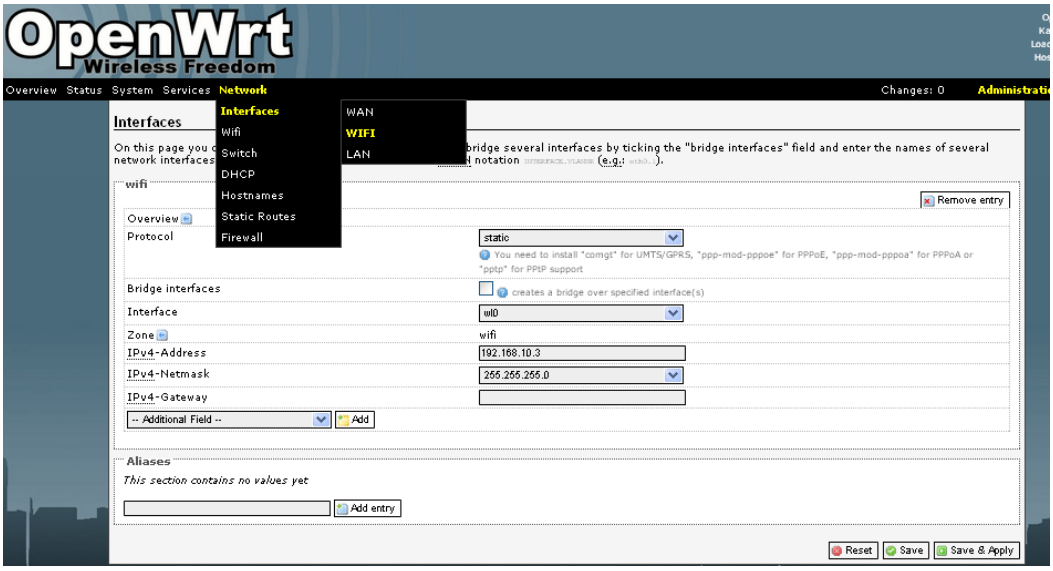


FIGURA 3.10 . CONFIGURACIÓN INTERFAZ INALÁMBRICA.

Es necesario configurar los parámetros de la interfaz inalámbrica WL0 accediendo a *Network/Wifi/WL0,* lo primero es habilitar la interfaz seleccionando la casilla *enable*, se escoge el canal en el que se va configurar la red, el ESSID debe ser el mismo para todos los nodos de la red, en este caso se uso **mesholsr**. El modo de operación de una red enmallada es Ad-Hoc, los campos en general se llenan tal como se presentan en la imagen a continuación.

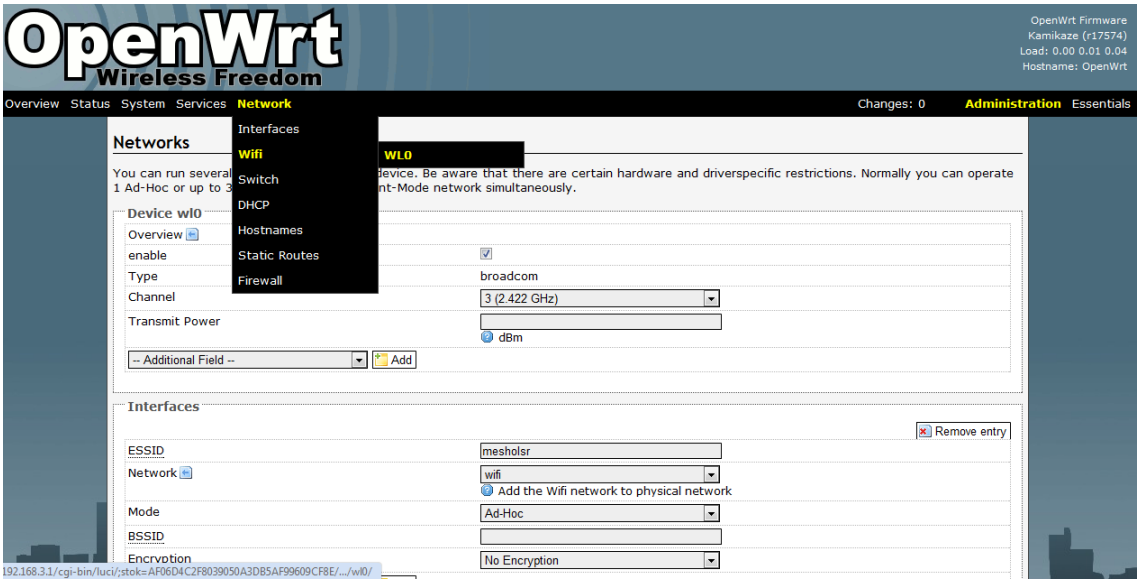


FIGURA 3.11 CONFIGURACIÓN INTERFAZ WL0.

3.3.1.4 WAN

Para configurar el puerto WAN que es el encargado de recibir la conexión a Internet, se accede a la pestaña *Network/Interfaces/WAN* y se habilita para que trabaje con DHCP si

está disponible, sino se llenan los campos de dirección IP, mascara de subred, puerta de enlace y servidor DNS de acuerdo con nuestro ISP.

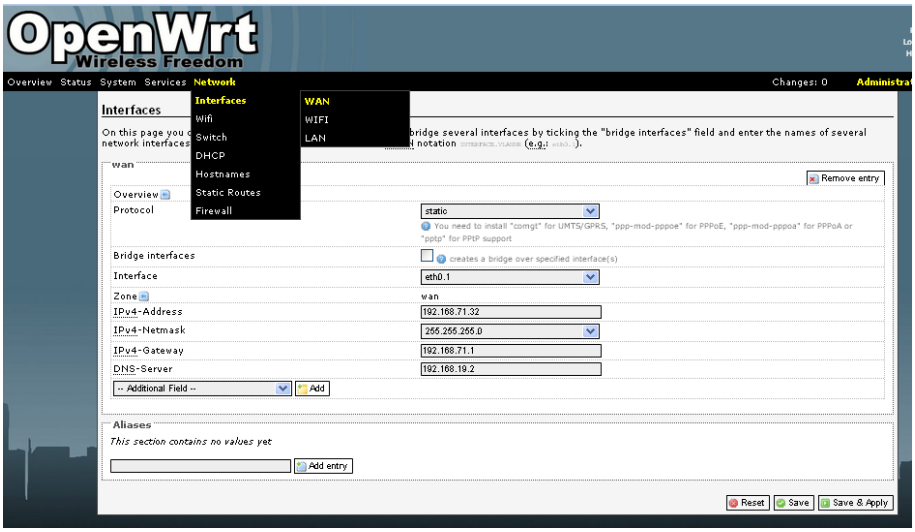


FIGURA 3.12 CONFIGURACIÓN WAN.

3.3.1.5 FIREWALL

El siguiente paso es configurar el Firewall accediendo a el a través de *Network/Firewall* y configurándolo de tal forma que permita la entrada, salida y redireccionamiento de paquetes para las interfaces lan y wifi tal como se muestra a continuación.

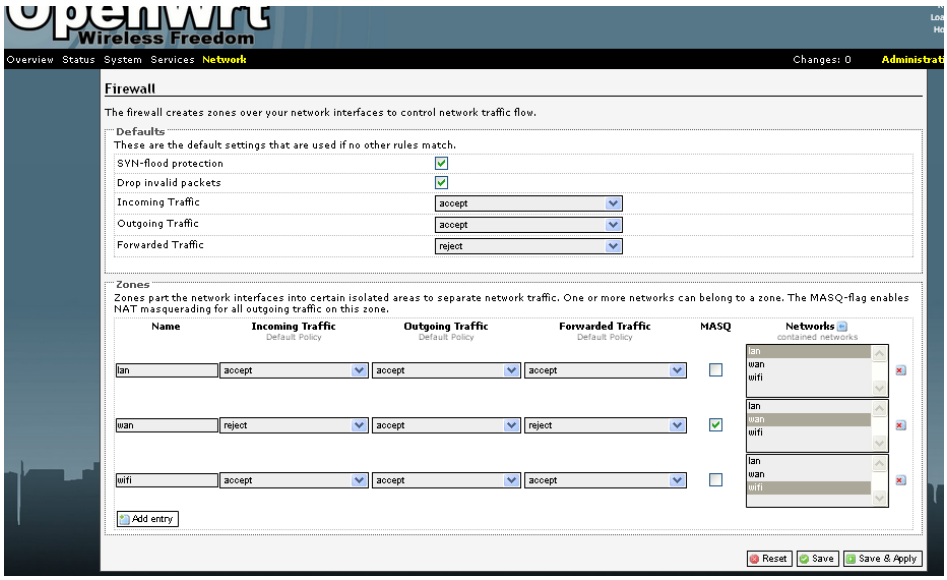


FIGURA 3.13 CONFIGURACIÓN FIREWALL.

3.3.1.6 INSTALACIÓN PAQUETES.

Replicando todo el proceso anterior con cada nodo se establece la red inalámbrica ad-hoc, pero sin protocolo de enrutamiento, lo cual implica que no tendrá la suficiente información en la tabla de rutas necesaria para acceder a otros nodos que estén fuera del área de cobertura de la interfaz inalámbrica, y a las interfaces lan de cada nodo mesh.

Para lo anterior es necesaria la instalación de diversos paquetes correspondientes a los protocolos de enrutamiento a implementar los cuales se encuentran en los repositorios de Openwrt a través de una sesión SSH a la dirección 192.168.3.1 (en este caso el 3) accediendo a la terminal de comandos del enrutador y en donde con el comando *opkg* es el encargado de la gestión de paquetes del firmware. Su sintaxis se presenta en la siguiente tabla:

Comando	Función
opkg update	Descarga la lista de paquetes disponibles en los servidores de los repositorios
opkg install nombre_paquete	Descarga el paquete seleccionado del repositorio y lo instala.
Opkg remove nombre_paquete	Desinstala o remueve el paquete seleccionado.

TABLA 3.3 SINTAXIS COMANDO OPKG.

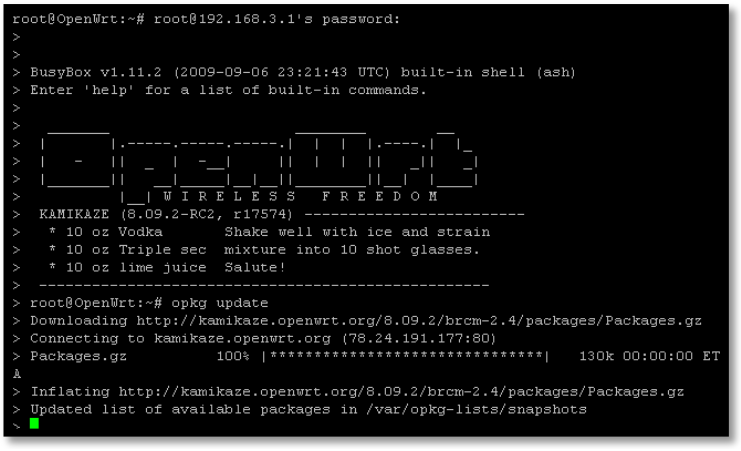


FIGURA 3.14 INSTALACIÓN PAQUETES POR COMANDOS.

También se pueden instalar los paquetes por medio de la interfaz web Luci, dirigiéndose a la pestaña System/Software donde en primer lugar solo se van a observar los paquetes instalados, es necesario presionar el botón de Update packages para descargar la lista de

paquetes disponibles y poder listarlos. Después de que realizar los cambios deseados se presiona el botón Perform Actions.

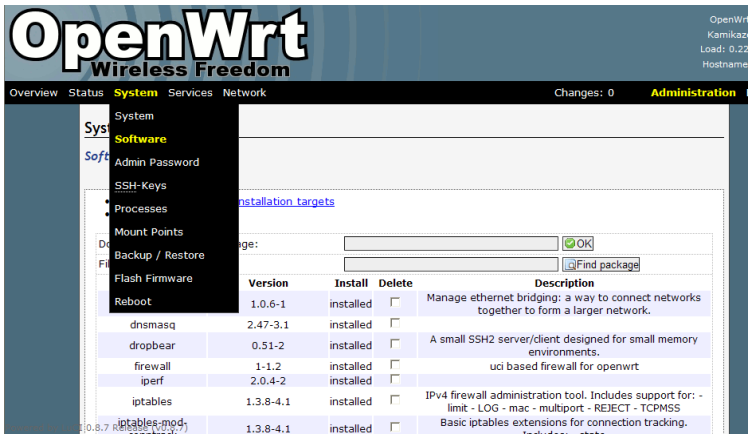


FIGURA 3.15 INSTALACIÓN PAQUETES POR INTERFAZ WEB.

3.4 IMPEDIMENTOS PARA IMPLEMENTAR OSPF Y RIP EN REDES INALÁMBRICAS AD-HOC

Configuradas correctamente las interfaces del enrutador Linksys es necesario implementar un protocolo de enrutamiento sobre la interfaz inalámbrica de cada enrutador la cual está configurada en modo ad-hoc.

Al inicio del trabajo de investigación se estudiaron los protocolos de enrutamiento y sus categorías más importantes de acuerdo al algoritmo usado para determinar las rutas, los cuales son Vector distancia y Estado de enlace, tomándose como punto de partida los protocolos RIP y OSPF como referencias principales respectivamente.

Para la implementación de los protocolos RIP y OSPF en el enrutador es necesaria la instalación del paquete Quagga que se encuentra en los repositorios del Openwrt. Quagga es una recopilación de software de enrutamiento que provee a sistemas de la familia Unix entre los cuales se encuentra FreeBSD, Solaris, NetBSD, Linux y por consiguiente para el firmware Openwrt. Quagga es una bifurcación del proyecto GNU Zebra el cual fue lanzado como parte del proyecto GNU y este demonio que se encarga de manejar las tablas de rutas del núcleo. Quagga soporta los protocolos OSPFv2, OSPFv3, RIP v1 y v2, RIPng y BGP-4 (23).

Se puede realizar la instalación por medio del comando: *opkg install quagga* como se observa a continuación.

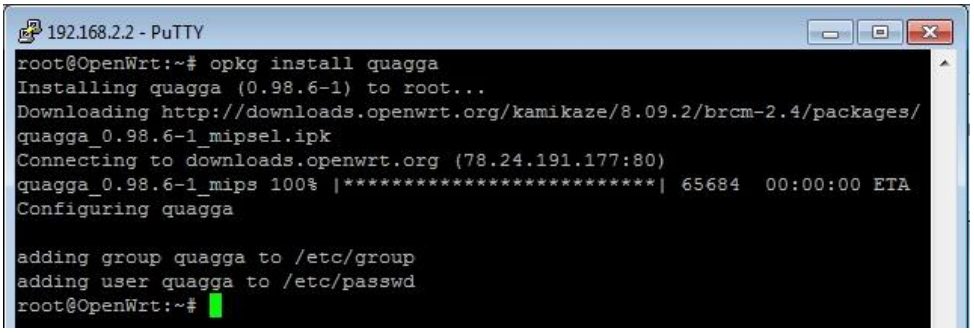


FIGURA 3.16 INSTALACIÓN QUAGGA POR COMANDOS.

Por medio de la interfaz Web del firmware además quagga es necesario seleccionar los paquetes correspondientes a los protocolos a ejecutar por el software quagga.

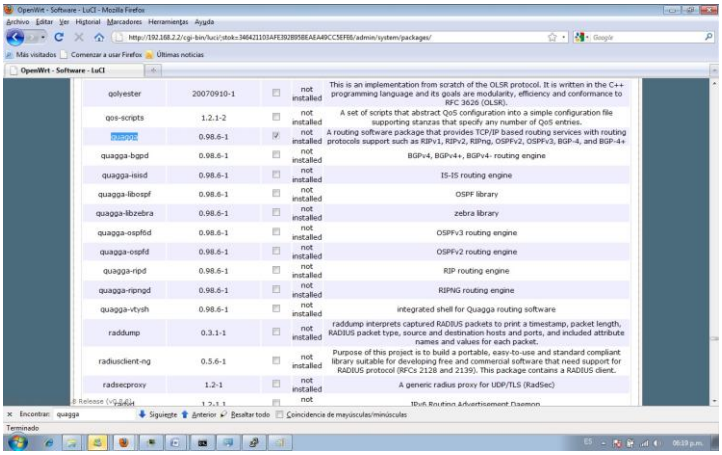


FIGURA 3.17 INSTALACIÓN QUAGGA POR INTERFAZ GRÁFICA.

Sin embargo estos protocolos a pesar de ser los más difundidos en cuanto IGP no fueron diseñados para redes inalámbricas ad-hoc sino para redes cableadas, motivo por el que presentan impedimentos y esto se pudo experimentar tanto con RIP como con OSPF en las siguientes topologías:

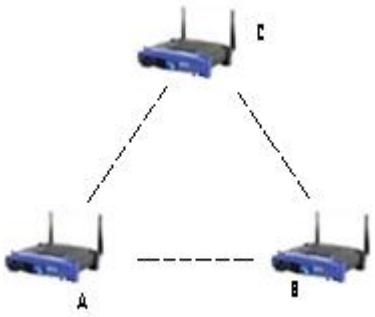


FIGURA 3.18 TOPOLOGÍA TRIANGULO.

Configurando las interfaces de los router Linksys WRT54GL de la forma descrita en el numeral 3.2 y organizándolos en una topología triangulo como la que se observa en la figura anterior, en esta cada uno de los nodos tiene en su área de cobertura a los otros dos nodos, el router A es el nodo 1; el router B es el nodo 2; y el router C es el nodo 3. En este caso los protocolos RIP y OSPF por medio del software de Quagga funcionan correctamente, ya que por ejemplo un equipo conectado a la interfaz LAN del nodo A puede comunicarse con otro equipo conectado a la interfaz LAN del nodo B o C ya que en la tabla de rutas del router A dictara que si el paquete se dirige a la subred 192.168.3.0/24 perteneciente a la interfaz LAN del router C su puerta de enlace es la dirección IP 192.168.10.3, y dado que la interfaz inalámbrica del router A pertenece a la subred 192.168.10.0/24 el paquete sale por esta.

Sin embargo si la topología es como la que se presenta en la figura topología en línea tres nodos en donde: el router A solo está conectado al router B, el router B con el router A y C, y el router C únicamente con el router B, las tablas de rutas creadas tanto con el protocolo RIP como con OSPF no saben resolver el caso anterior de un equipo conectado por la interfaz LAN del router A comunicarse a otro equipo conectado a la interfaz LAN del router C.

FIGURA 3.19 TOPOLOGÍA EN LÍNEA TRES NODOS.

Esto se debe a que tanto RIP como OSPF generan la tabla rutas de manera incorrecta ya que el router B informa al router A que para alcanzar la subred 192.168.3.0/24 del router C la puerta de enlace es la dirección IP 192.168.10.3 no obstante el router A en su tabla de rutas solo sabe resolver que para alcanzar la subred 192.168.10.0/24 el está directamente conectado por la interfaz inalámbrica luego cuando intenta enviar un paquete a la dirección 192.168.10.3 sale correctamente por su interfaz inalámbrica pero el router C al no estar en el rango de cobertura del router A el paquete se pierde. Esto se debe a que los protocolos RIP y OSPF generan las tablas de rutas solo las puertas de enlace para alcanzar únicamente otras subredes por ejemplo las de las interfaces LAN o dado el caso la interfaz WAN, mas no direcciones IP pertenecientes a equipos en particular de una subred.

Se puede resumir que la movilidad de los nodos dentro de la red inalámbrica ad-hoc conlleva a situaciones complejas con los vecinos que tienen cobertura por dos enrutadores distintos dificultando la elección del mejor camino.

Por lo cual debido a estas limitaciones que presenta el protocolo OSPF en redes inalámbricas ad-hoc han surgido iniciativas en grupos de trabajo para adaptar el protocolo

a estas nuevas topologías, desarrollándose drafts enfocados en diversas perspectivas tales como el trabajo de Kenneth Holter (24), como también el trabajo de Boeings (25) en donde en ambos trabajos parten de quagga para su implementación. Sin embargo no se ha llegado a estandarizar OSPF en este sentido.

Debido a los desafíos que presentan las redes inalámbricas ad-hoc tales como el gran número de vecinos inmediatos y permitir fácilmente la inclusión de nuevos nodos, la movilidad de los nodos cambiando constantemente la topología de red generando así una inundación de mensajes de alerta que congestionen la red, el permitir que se conecten a redes fijas para que estos nodos actúen como puertas de enlaces a otras redes ha resultado en la creación del grupo de trabajo MANET (Mobile Ad-hoc Network) en el Grupo de Trabajo de Ingeniería de Internet IETF (Internet Engineering Task Force) (26) siendo el encargado de estudiar los protocolos candidatos a ser estandarizados.

De acuerdo a lo anterior se estudio una lista de protocolos de enrutamiento para redes inalámbricas ad-hoc (27) y se contrasto con la lista de paquetes del repositorio de Openwrt para definir que protocolos de enrutamiento se pueden implementar en los Linksys WRT54GL arrojando como resultado los protocolos BABEL, OLSR Y B.A.T.M.A.N.

3.5 IMPLEMENTACIÓN Y CONFIGURACIÓN BABEL.

Para la instalación de Babel se requiere del paquete babeld y se realiza por medio del comando *opkg install babeld* como se ve a continuación.

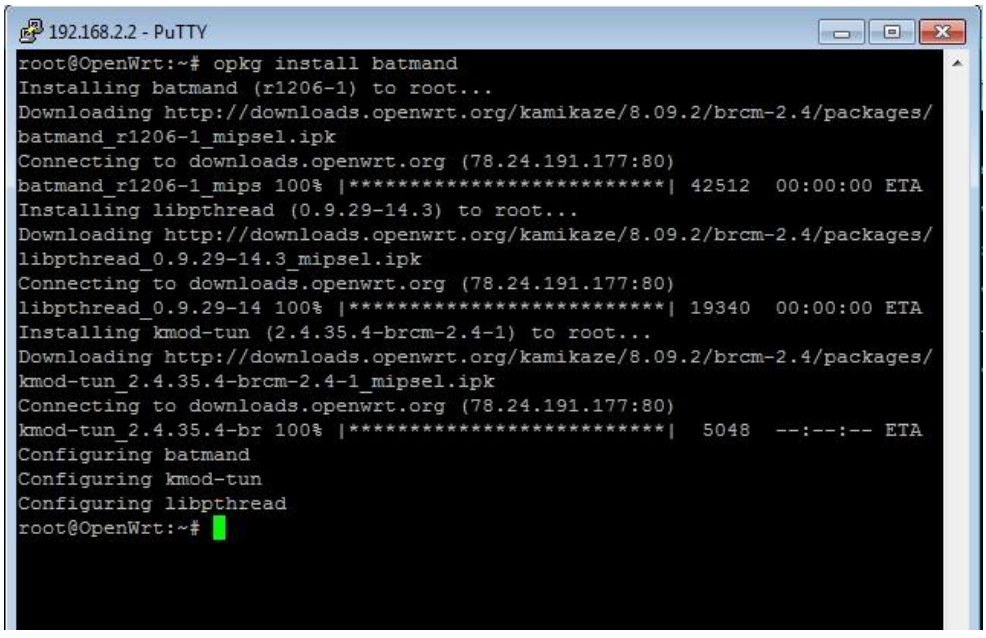


FIGURA 3.20 INSTALACIÓN POR COMANDOS BABELD.

También se advierte que se instalan automáticamente los paquetes necesarios para el correcto funcionamiento del comando babeld. Asimismo usando la interfaz gráfica se puede realiza la instalación como se observa en la siguiente figura.

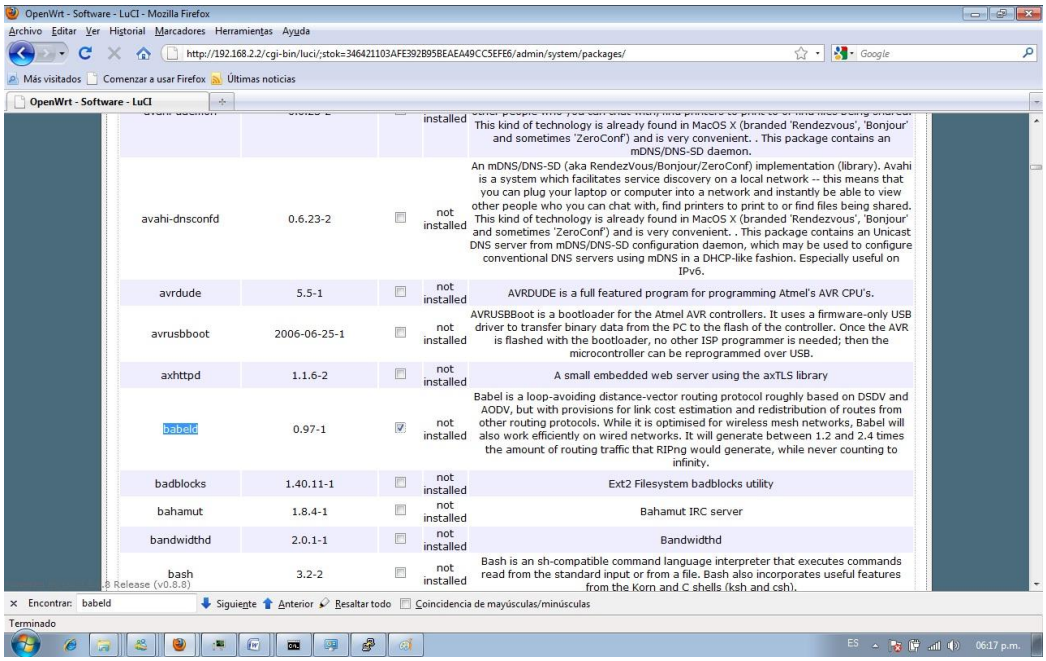


FIGURA 3.21 INSTALACIÓN POR INTERFAZ WEB BABELD.

No obstante para la configuración y puesta en marcha del protocolo únicamente se puede realizar por medio de la ventana de comandos usando la siguiente sintaxis (11):

babeld *opción*… [--] *interface*…

Las opciones se presentan en la siguiente tabla.

Opción	Función
-m dirección-multicast	Se especifica la dirección multicast a usar por el protocolo.
-p puerto	Selecciona el puerto UDP usado por el protocolo. Por defecto es 8475.
-h intervalo-hello	Selecciona el intervalo en segundos en que se programa el envío de paquetes hello en las interfaces inalámbricas. Por defecto es 4 segundos.

-H intervalo-cable-hello	Selecciona el intervalo en segundos en que se programa el envío de paquetes hello en las interfaces cableadas. Por defecto es 20 segundos.
-k prioridad	Selecciona el valor de prioridad usado cuando se instalan las rutas en el kernel. Por defecto es 0.
-P	Se ejecuta en modo pasivo, el demonio solo anunciara rutas redistribuidas.
-d nivel	Nivel de depuración. Valor 1 solicita la tabla de rutas cada iteración del lazo principal del demonio de babel. Valor 2 solicita adicionalmente la tabla cada vez que recibe o envía un paquete. Valor 3 adicionalmente arroja la tabla con cada interacción con el kernel. Por defecto es 0.
-t tabla	Usar la tabla de rutas del kernel para las rutas insertadas por babeld.
-c nombre_archivo	Especifica el nombre del archivo de configuración. Por defecto /etc/babeld.conf.
-C sentencia	Especifica una sentencia de configuración directamente en la línea de comandos.
interface…	La lista de las interfaces en que el protocolo va a operar.

TABLA 3.4 OPCIONES DEMONIO BABEL.

De acuerdo a las opciones anteriores se configuro el protocolo de babel con la siguiente línea de comando:

babel -C ’redistribute metric 256’ wl0

Sin embargo al momento de apagar y encender el enrutador o simplemente reiniciarlo es preciso volver a ejecutar el comando anterior por lo que se hizo necesario la creación de un script que al iniciarse el enrutador se inicie el demonio de babeld. Este script se crea por medio del editor de textos *vi* creando un archivo con el nombre S50inicio_babel en donde la letra S (Start) representa que es para iniciar el demonio y el numero 50 el tiempo en el que se inicia, este se archiva en el directorio /etc/rc.d/ en el cual están almacenados todos los demonios que inician en el enrutador.

Adicionalmente hay que otorgarle permisos de ejecución al archivo por medio del comando chmod. El contenido del archivo es la línea de comando de configuración del protocolo babel, es decir *babel -C ’redistribute metric 256’ wl0*.

3.6 IMPLEMENTACIÓN Y CONFIGURACIÓN OLSR.

Para implementar OLSR es necesario agregar tres paquetes de software que no vienen por defecto con el *Firmware*, estos son:

- olsrd-luci
- olsrd-luci-mod-dyn-gw
- olsrd-luci-mod-txtinfo

Se pueden instalar por medio de la ventana de comandos al haber iniciado una sesión SSH o por medio de la interaz web luci del openwrt.

A pesar de tener instalado los paquetes el servicio no se inicializa automáticamente por lo que se debe ir a la carpeta *etc/init.d*, luego se da el comando *./olsrd start* y este inicializa el demonio necesario para correr el protocolo.

Otra posibilidad es crear un archivo dentro del directorio rc.d de tal forma que cada vez que se inicie el enrutador se ejecute el demonio olsrd, en este caso particular se denomino S96inicio_olsrd, el 96 indica a los cuantos segundos se ejecuta, después se deben dar los permisos de ejecución al archivo.

Después de iniciar el demonio entonces se procede a configurar parámetros específicos del protocolo, como los anuncios HNA, para acceder a estos se ingresa a *Service/OLSR/HNA Announcements*. Los anuncios dan a conocer redes fuera del dominio de enrutamiento, si se coloca 0.0.0.0 anuncia todas las redes conectadas, lo anterior combinado con los plugins permite anunciar la red WAN para que le brinde conexión a Internet a los otros nodos y estos coloquen el nodo con conexión como ruta por defecto.

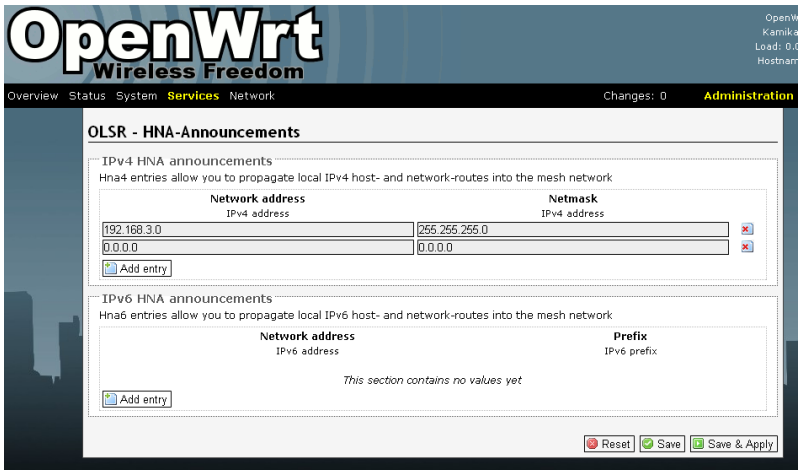


FIGURA 3.22 ANUNCIOS HNA OLSR.

Se tiene la posibilidad de ver cómo está funcionando el protocolo a través de la pestaña Status, donde se presenta la tabla de rutas, la topología de la red, los anuncios HNA en la

red y los anuncios de múltiples interfaces. A continuación se presenta una imagen de las posibilidades, seleccionando topología como ejemplo.

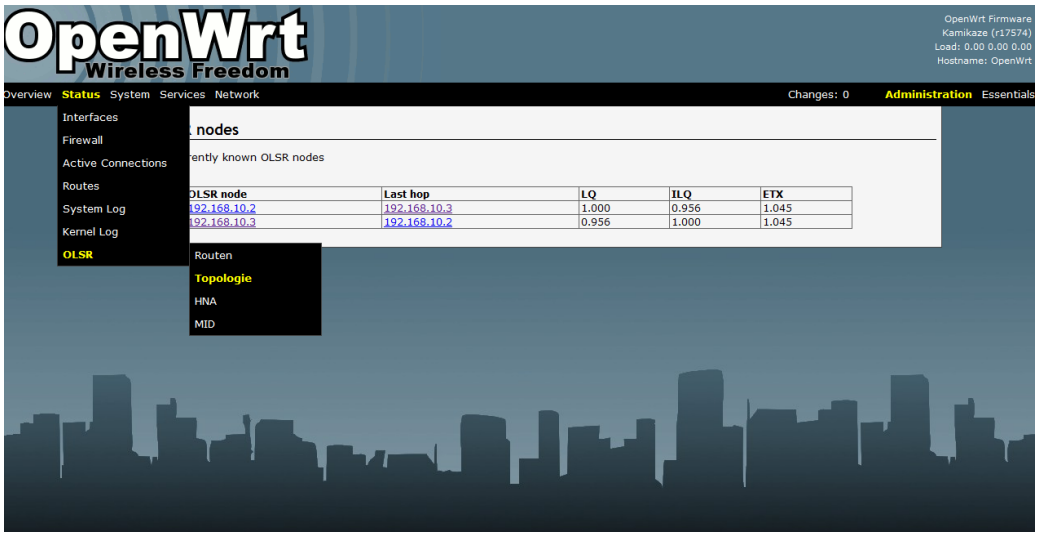


FIGURA 3.23 ESTADO TOPOLOGÍA OLSR.

3.7 IMPLEMENTACIÓN Y CONFIGURACIÓN B.A.T.M.A.N.

La instalación de B.A.T.M.A.N. se realiza por medio del paquete batmand que se encuentra en los repositorios de Openwrt usando del comando opkg install batmand donde además se instalan automáticamente otros paquetes necesarios para la ejecución de batmand, esto se observa en la siguiente figura.

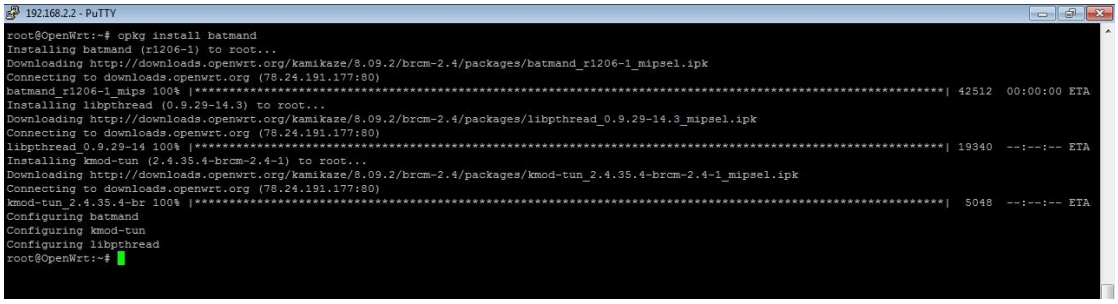


FIGURA 3.24 INSTALACIÓN POR COMANDO BATMAND.

También se puede realizar por medio de la interfaz gráfica seleccionando el paquete babeld de la lista y presionando el botón Perform action ubicado al final de la lista, esto se ve en la figura a continuación.

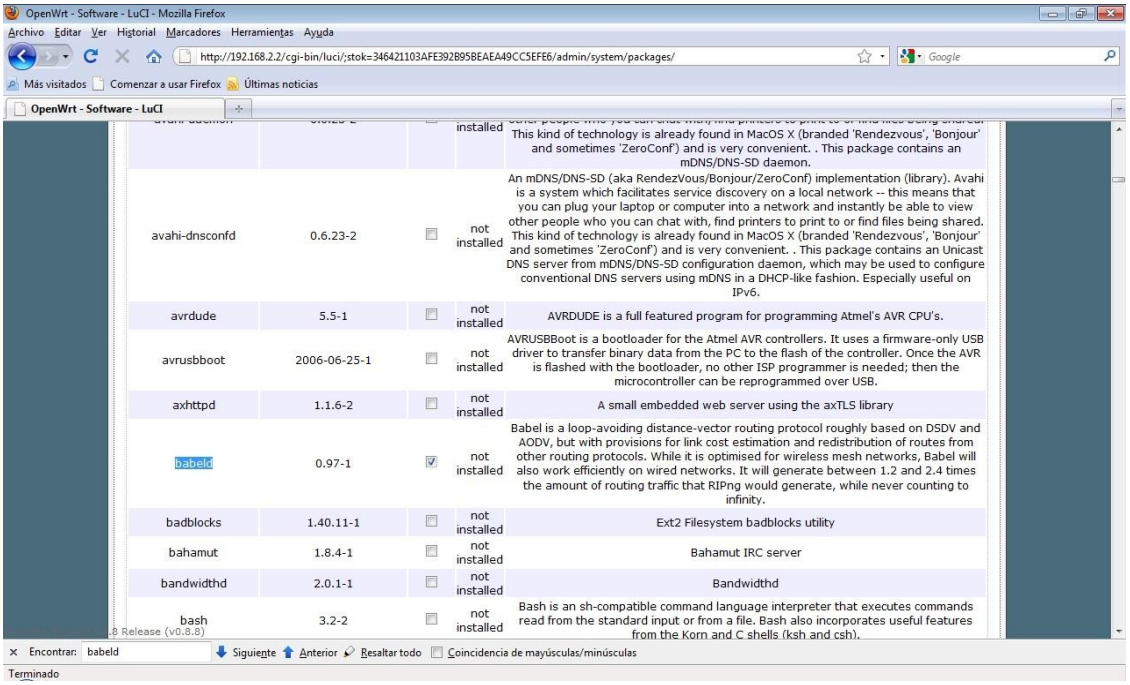


FIGURA 3.25 INSTALACIÓN POR INTERFAZ WEB BATMAND.

Para la configuración y ejecución del demonio de batmand se usa la siguiente sintaxis (28):

batmand [*opciones*] *interface*...

Las opciones se presentan en la siguiente tabla.

-a añadir redes para anunciar	Especifica las redes a añadir a la lista del demonio que van a estar disponibles para las otras redes. Esta opción se puede usar varias veces incluso cuando el demonio ya se este ejecutando. El parámetro se ingresa de la forma dirección-ip/mascara_subred.
-A borrar redes para anunciar	Especifica las redes a remover de la lista del demonio que van a estar disponibles para las otras redes. Esta opción se puede usar varias veces incluso cuando el demonio ya se este ejecutando. El parámetro se ingresa de la forma dirección-ip/mascara_subred.
-d nivel de depuración	El nivel de depuración se puede ajustar en 5 niveles: Por defecto: 0 -> depuración deshabilitada. Valores permitidos: 1 -> listar los vecinos. 2 -> listar las puertas de enlace. 3 -> observar el demonio batmand. 4 -> observar batmand de manera mas detallada. 5 -> uso de memoria y uso de procesador.
-g capacidad puerta de enlace	Se usa para informar a los demás nodos en la red el ancho de banda de internet disponible. Solo se ingresa un número seguido de manera opcional por kbit o mbit y el demonio supone el tipo de puerta de enlace apropiado. Se usa el "/" para separar tasa de descarga y de subida. Si se omite el valor de subida el demonio asume que es /5. Por defecto: 0 -> puerta de enlace deshabilitada. Valores permitidos:

	5000 5000kbit 5mbit 5mbit/1024 5mbit/1024kbit 5mbit/1mbit
-h	Menú de ayuda corto
-H	Menú de ayuda detallado.
-o intervalos en ms	Un nodo transmite mensajes broadcast llamados OMG(originator message) para informar a los vecinos de su existencia. Es el tiempo de espera antes de enviar el siguiente mensaje. Por defecto es 1000 ms (1 segundo). En una red mobile se sugiere valores menores a 1000 ms donde es necesario detectar rápidamente cambios en la topología. En una red estática se puede reducir el tráfico sobre la red aumentando este valor.
-p puerta de enlace preferida	Se selecciona la puerta de enlace deseada por el usuario. Sino la encuentra tomara la que índice la clase de enrutamiento.
-r clase de enrutamiento	La clase de enrutamiento se puede ajustar en 4 valores. El enrutador erigirá la puerta de enlace a internet basado en ciertos criterios a menos que la opción –p este activada. Por defecto: 0 -> no define ninguna ruta. Valores permitidos: 1 -> usar la conexión mas rápida. 2 -> usar la conexión mas estable. 3 -> usar la conexión de cambio mas rápido. 4 -> usar la conexión de cambio mas lento.
-v	Mostrar versión del paquete.

TABLA 3.5 OPCIONES DEL DEMONIO BATMAND.

De acuerdo a las opciones presentadas se ejecuto la siguiente sentencia de comando para la implementación de B.A.T.M.A.N. en el enrutador.

batmand –a 192.168.10.1/32 –a 192.168.1.0/24 wl0

En donde 192.168.10.1/32 corresponde a la dirección IP de la interfaz inalámbrica del nodo 1, la dirección 192.168.1.0/24 corresponde a la subred perteneciente a la interfaz LAN del enrutador y wl0 es el nombre de la interfaz inalámbrica.

No obstante en el instante en que se apague y encienda el enrutador o se reinicie es necesario volver a ejecutar la línea de comando anterior, por lo que también se creó un scritp para que el demonio se ejecute automáticamente al iniciar el enrutador como se explico en el numeral 3.5 en el caso similar de babeld.

4. CAPITULO CUATRO – PROCESOS DE MEDICIÓN

La medición del desempeño de una red es una tarea difícil de definir ya que existen múltiples métricas y algunos diseñadores de red pueden darle pesos diferentes a cada una de ellas, sin embargo esto realmente depende de la aplicación que se le va dar a la red como tal. En el desarrollo de este proyecto se tiene como objetivo obtener tres métricas que son: tasa de transferencia, el *Jitter* y el retraso, a su vez también es necesario definir cómo se va realizar el proceso de medición de potencia de los nodos en la red. En este capítulo primero se discuten las herramientas de análisis de tráfico y luego los procesos de medición de cada una de las métricas mencionadas.

4.1 HERRAMIENTAS DE ANÁLISIS DE TRÁFICO.

Existen múltiples herramientas de análisis de tráfico, para este proyecto se estudiaron algunas de uso libre de las cuales se documentaron: Netperf, DITG, Iperf y Jperf.

4.1.1 NETPERF

Es una herramienta desarrollada por la división de Redes de información de la compañía Hewllet-Packard creada para ayudar a sus clientes a medir el desempeño, esta herramienta está compuesta por dos programas netperf y netserver, el último de ellos funciona como servidor mientras el primero como cliente. Esta herramienta permite simular tres tipos de tráfico TCP y dos de UDP, estos tipos son (29):

- Una conexión TCP usando intentando saturar la red con una gran cantidad de datos.
- Una conexión TCP usando peticiones cliente y respuestas del servidor.
- Múltiples pares petición/respuesta, cada una como una conexión TCP separada.
- Un volcado unidireccional de datos UDP transferidos del cliente al servidor que intenta saturar la conexión.
- Una sesión petición respuesta usando UDP.

Se debe tener en cuenta que UDP no corrobora que el paquete enviado se halla recibido por lo que se deben tener en cuenta tanto las estadísticas del cliente como del servidor para saber la tasa de transferencia real en UDP.

COMANDO	DESCRIPCIÓN
-a *tamaño*	Define el tamaño del buffer de alineamiento en el sistema local.
-A *tamaño*	Define el tamaño del buffer de alineamiento en el sistema remoto.
-b *tamaño*	Especifica el tamaño la cantidad de datos que se van a transferir durante la prueba.
-c	Calcula el uso de la CPU en el sistema local.
-C	Calcula el uso de la CPU en el sistema remoto.
-d	Incrementa el nivel de depuración en el sistema local.
-f *unidades*	Cambia las unidades de medida mostradas en la prueba.
-F *archivo*	Especifica el archivo que se va enviar.
-h	Muestra la ayuda.
-H *estación*	Especifica el nombre de la estación o la dirección IP del equipo remoto que corre el *netserver.*
-i *min,max*	Especifica el mínimo y máximo número de iteraciones.
-l *tiempo*	Especifica el tiempo en segundos de la prueba.
-n *# CPUs*	Especifica el número de CPUs en la estación vecina.
-o *número*	Sitúa un de offset para el buffer de alineamiento del equipo local.
-O *número*	Sitúa un de offset para el buffer de alineamiento del equipo remoto.
-p *puerto*	Especifica el puerto del equipo remoto *netserver* al que se desea conectar.
-v *verbose*	Especifica el nivel de verbosidad en verboso.
-t *nombre*	Especifica el nombre de la prueba.

TABLA 4.1 COMANDOS NETPERF

La herramienta netperf se probo satisfactoriamente entre dos estaciones, sin embargo no se encontró en los repositorios del firmware por lo que en caso de seleccionarla es necesario descargarla y compilarla para el enrutador linksys WRT54GL. Además no se encontró una interfaz gráfica de la misma lo que implica que sería necesario desarrollar un software que realice las gráficas ó desarrollar una interfaz gráfica para agilizar el desarrollo de las pruebas.

4.1.2 DITG

En el estado del arte se encuentran múltiples trabajos para medir el desempeño de redes heterogéneas utilizando DITG. Esta herramienta es un generador de tráfico desarrollado por el departamento de informática y sistemas de la universita' degli Studi di Napoli ''Federico II", sus siglas DITG hacen referencia a ***Distributed Internet Traffic Generator***, este generador de tráfico se divide en cinco componentes que son (29) (30): ITGSend, ITGRecv, ITGLog, ITGDec, ITGplot e ITGapi, estos se describen a continuación:

4.1.2.1 ITGSEND

Se encarga de generar varios tipos de flujos, cada flujo se maneja como un hilo independiente, con un hilo separado actuando como maestro y coordinando los otros. Se puede generar un script que contenga todos los flujos que se desee generar. Tiene más opciones que las otras herramientas analizadas para el flujo enviado y permite generar tráfico UDP, TCP, ICMP, SCTP y DCCP (30).

4.1.2.2 ITGRECV

Este componente recibe y genera el un archivo log. En caso de que el nombre del archivo sea especificado tanto en el receptor como en el emisor el de este último se ignora. Una ventaja de tener componentes es que los archivos pueden ser guardados en diferentes equipos, por eso el ITGRecv en su sintaxis permite escribir una dirección IP del equipo donde se espera guardar el archivo log, por defecto esto se hace en la estación local. En caso de que se escriba una IP tanto en el emisor como en el receptor, se usa únicamente la del receptor (30).

4.1.2.3 ITGLOG

Este componente funciona como un servidor Log de la plataforma D-ITG, recibe los archivos log ya sea del emisor o el receptor, se utiliza cuando se desea grabar los archivos en otra estación diferente a las que se realiza la prueba, por ejemplo si el administrador de redes tiene su propio equipo y está realizando pruebas entre dos nodos de la red lo más probable es que desee que los archivos log queden en su equipo más no en los nodos en los que realiza la prueba. El ITGLog escucha en puertos dinámicos situados en el rango de [9003-10003] (30).

4.1.2.4 ITGDEC

Este componente es el decodificador de los archivos log generados en las pruebas, es necesario para analizar los resultados ya que calcula los valores promedio de tasa de bits, retraso y *Jitter,* es posible para todo el experimento ó para intervalos de tiempo (30).

4.1.2.5 ITGPLOT

Este componente es un script de octave que permite dibujar los datos contenidos en los archivos bitrate.dat, jitter.dat o packetloss.dat. Por defecto las gráficas se graban en formato EPS, sin embargo es posible cambiarlo por otros formatos suministrados por GNUPLOT. Además el componente permite colocar un titulo a la gráfica (30).

4.1.2.6 ITGAPI

Este último componente es un API de C++ que permite controlar el generador de tráfico de manera remota. Bastante útil si se desean lanzar sesiones simultáneas en varios nodos distintos de la red, o en varias redes, está función permite al administrador de redes controlar todo desde una sola estación (30).

4.1.3 IPERF Y JPERF

Iperf es una herramienta de análisis desempeño creada por la universidad de Illinois en el *National Laboratory for Applied Network Research (NLANR)* que puede correr tanto en ambientes Linux como Windows y permite hacer diferentes tipos de pruebas TCP y UDP. Poder realizar pruebas entre dos sistemas operativos diferentes es una gran ventaja, además Iperf puede correr como programa de fondo en Windows o como un demonio en Linux y sin interferir con las otras aplicaciones de la estación (29).

PARAMETROS	**DESCRIPCIÓN**
-f *formato*	Especifica las unidades de salida en bits, bytes, kilo bytes, etc.
-i *intervalo*	Especifica el intervalo en que Iperf va ir mostrando reportes de los resultados, por defecto los resultados se presentan simplemente al finalizar la prueba.
-l *tamaño*	Especifica el tamaño de los paquetes enviados en la prueba.
-n *número*	Especifica el número de paquetes enviados.

-p *puerto*	Especifica el puerto usado para contactar el servidor
-t *tiempo*	Especifica el tiempo de transmisión de paquetes.
-P *clientes*	Especifica el número de conexiones simultaneas a clientes
-S *tos*	Especifica tipo de servicio.
-m *MTU*	Imprime el máximo tamaño de segmento TCP
-o *nombre*	Especifica el nombre del archivo donde se va guardar el reporte
-w	Especifica el tamaño de la ventana.
-B *ipmulticast*	Permite realizar una prueba de tráfico multicast
-N	Sitúa TCP sin retardo, deshabilitando el algoritmo Nagle.
-V	Sitúa el dominio en IPv6
-u	Utiliza UDP en lugar de TCP que ya viene por defecto

TABLA 4.2 COMANDOS GENERALES IPERF.

PARAMETROS	**DESCRIPCIÓN**
-s	Especifica que corra como servidor.
-D	El servidor corre como demonio
-R	Remueve el servicio en win32

TABLA 4.3 COMANDOS SERVIDOR IPERF.

PARAMETROS	**DESCRIPCIÓN**
-b	Para UDP, especifica el ancho de banda enviado en bits/segundo
-c *IPservidor*	Especifica que está en modo cliente y la estación a la que se quiere conectar.
-d	Realiza una prueba bidireccional simultáneamente
-n *número*	Especifica el número de bytes a transmitir
-r	Hace un test bidireccional pero individualmente
-t *tiempo*	Tiempo en segundos para la transmisión

-F *archivo*	Especifica el archivo que va ser transmitido como entrada.
-L *número*	Puerto donde se va escuchar el test bidireccional
-P *número*	Número de clientes corriendo de forma paralela.
-T *número*	Tiempo de vida para multicast.

TABLA 4.4 COMANDOS CLIENTE IPERF.

Iperf permite hacer múltiples tipos de pruebas tanto en UDP y TCP como se puede observar analizando las opciones de los comandos ya presentados, por otro lado el tamaño del código es pequeño, está en los repositorios y se puede descargar e instalar para que corra sobre la versión reducida de Linux contenida por los enrutadores Linksys WRT54GL, esto abre la posibilidad de hacer pruebas directamente entre enrutadores o entre enrutadores y estaciones.

Jperf es una interfaz gráfica creada en Java que permite usar Iperf agilizando las pruebas, por este motivo puede utilizarse en cualquier equipo que contenga el respectivo JVM *(Java Virtual Machine).* Con Jperf es posible configurar tanto las opciones de servidor como las opciones del cliente, a continuación se presenta una imagen de la interfaz.

FIGURA 4.1 INTERFAZ GRÁFICA JPERF.

Después de estudiar las herramientas de análisis de tráfico se decidió utilizar Iperf en combinación con la interface Jperf porque que es versátil, presenta las opciones necesarias para los objetivos del presente proyecto, para las pruebas con estaciones la interfaz agiliza mucho las pruebas, es más liviana y rápida de usar que DITG, a diferencia de este último los reportes se generan directamente y no es necesario usar otro componente de software

para procesarlos y a diferencia de Netperf ya esta compilada por lo que se puede instalar en los enrutadores para hacer pruebas entre nodos directamente. Las tres herramientas presentaban resultados casi iguales para una configuración cliente servidor entre dos enrutadores cada uno conectado a su respectiva estación vía Ethernet, lo mismo ocurrió en modo ad-hoc por lo que cualquiera de las tres es útil en proyectos de análisis de tráfico, sin embargo DITG es la herramienta más completa en cuanto a opciones pero la que ocupa más espacio en memoria, es de gran utilidad cuando se desean controlar las pruebas remotamente y guardar datos en ciertas estaciones, en fin si las pruebas sacan alguna ventaja de que el programa sea distribuido y de cierta forma pueda operar como componentes independientes. Finalmente se presenta una tabla resumen de las funcionalidades que se tuvieron en cuenta para seleccionar Iperf.

Funcionalidad	Netperf	Iperf	DITG
Instalación en linksys		x	
Documentación	Media	Alta	Media
Requerimiento de memoria	Bajo	Bajo	Alto
Permite medir parámetros deseados	x	x	x
Interfaz Gráfica		x	x
Compilada	No	Si	No
Generación de reportes	Directa	Directa	Codificado

TABLA 4.5 COMPARACIÓN HERRAMIENTAS.

4.2 PROCESOS DE MEDICIÓN DE LAS MÉTRICAS

Para la caracterización de la red se propone medir parámetros como: tasa de transferencia, *jitter,* retraso y potencia. Los procesos como se midieron se presentan a continuación:

4.2.1 TASA DE TRANSFERENCIA

Es la cantidad de datos por unidad de tiempo que son enviadas a través de un sistema, la tasa de transferencia también se puede denominar throughput, para todas las pruebas la herramienta básica usada para medir la tasa de transferencia va ser Iperf sin importar si se hace o no a través de la interface Jperf, las pruebas de las topologías tienen una duración de 10 segundos mientras que las de potencia tendrán una duración de 100 segundos. Aunque se pueden realizar individualmente se realizo bidireccional para hacer más ágil el proceso, para TCP la prueba se puede ver en cualquiera de las dos estaciones

sin embargo para UDP es necesario tomar el ancho de banda desde la estación que actúa como servidor ya que esta es la que realmente sabe la cantidad de paquetes que llegan, UDP no confirma si los paquetes que envió llegaron por lo que el cliente siempre va ver el ancho de banda que se coloca en Iperf, por eso al hacer la prueba bidireccional se guardan los resultados de ambas estaciones para poder deducir el ancho de banda en cada dirección.

4.2.2 RETRASO

Se refiere al tiempo que tarda un paquete en ir desde la fuente hasta el destino, para aplicaciones prácticas existen problemas de sincronización de relojes por lo que se toma el tiempo que tarda un paquete en ir y volver a su fuente, este tiempo es denominado en redes rtt (*round trip time)* y es el que se va usar en el presente trabajo. Para medir se utiliza el comando ping presente en los sistemas operativos Windows y Linux, este genera el máximo retraso de una prueba, el mínimo y su promedio, este último será el valor que se toma en cuenta.

4.2.3 JITTER

Este término en redes se utiliza para definir la variabilidad del retraso de los paquetes a través de una red. Este proceso se hace tomando el promedio de la variación de retraso entre un paquete y otro para un número determinado de paquetes (30).

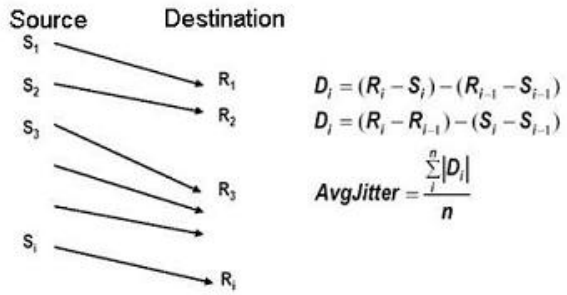


FIGURA 4.2 DIAGRAMA[12] *JITTER.*

R_i se refiere al tiempo en el que llego el primer bit del paquete i, y S_i al momento en que se envió dicho paquete, la diferencia (R_i - S_i) es equivalente al retraso, la diferencia [(R_i - S_i) - (R_{i-1} - S_{i-1})] es el *Jitter*, normalmente se toman n muestras de *Jitter* y se promedian

[12] Diagrama extraído de D-ITG V. 2.6.1d Manual, Mayo 2 del 2008.

obteniendo el *Jitter* promedio que es el que se va mostrar como resultado, este se obtiene en el receptor de una prueba UDP usando Iperf.

4.2.4 POTENCIA

Para la caracterización de la tasa de transferencia contra la potencia es necesario medir la potencia, después de varias pruebas preliminares se decidió que la mejor forma es hacerlo directamente del driver del enrutador a través usando el comando **iwlist scan**. Por otro lado modificar la potencia emitida por el enrutador es necesario instalar el paquete **wl**, en este proceso se usa el gestor de paquetes de Linux opkg, después de instalado el paquete es posible modificar la potencia de emisión utilizando el comando **wl txpwr.**

5. CAPITULO QUINTO – PRUEBAS DE POTENCIA, CONSTRUCCIÓN DEL BANCO DE PRUEBAS Y ANALISIS DE RESULTADOS

5.1 CARACTERIZACIÓN TASA DE TRANSFERENCIA RESPECTO A LA POTENCIA

Las pruebas presentadas en el presente capitulo muestran el comportamiento de la tasa de transferencia respecto a la potencia emitida por el enrutador, estas se realizaron en el edificio de ingenierías físico-mecánicas de la Universidad Industrial de Santander que es un ambiente *indoor,* también se realizaron pruebas *outdoor* para contrastar los resultados.

5.1.1 DESCRIPCIÓN DE LA PRUEBA.

En la figura se muestra la disposición de los enrutadores en el edificio, en una prueba realizada previamente se encontró que con línea de vista (entre los puntos A y B) a lo largo de todo el edificio las variaciones de la potencia son menores a 1dbm.

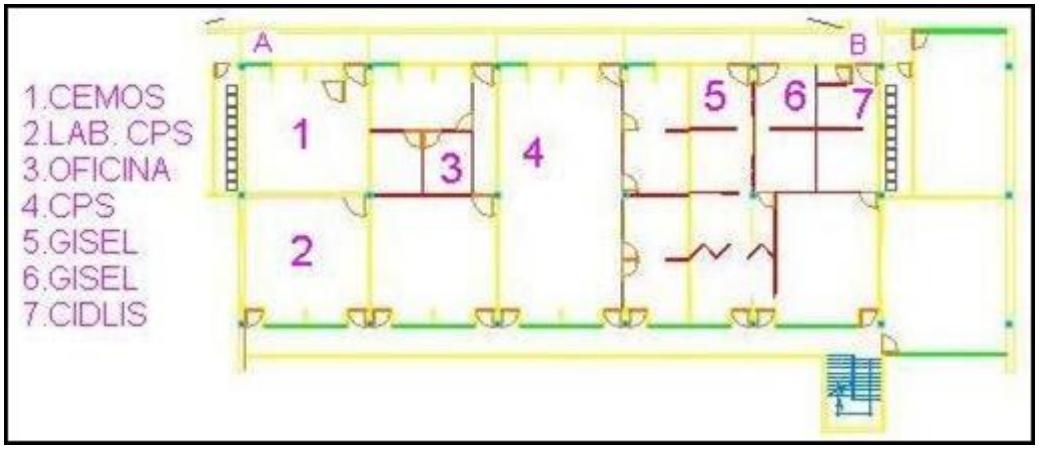


FIGURA 5.1 PLANO PRUEBAS POTENCIA INDOOR.

Las pruebas se realizaron dejando un enrutador en la ubicación 1 y moviendo otro a las posiciones 2, 3, 4, 5, 6, 7. En cada posición se realizaron mediciones con los valores por defecto de iperf, se midió el ancho de banda en intervalos de cien segundos para cada una de las siguientes potencias 1, 30, 60, 90, 120, 150, 180, 210, 250 en mW.

A diferencia de las pruebas con los puntos A y B, los puntos interiores tienen atenuación debido a las paredes, y muestran una clara variación del ancho de banda respecto a la potencia, los resultados se presentan en la tabla y pueden ser visualizados en la figura.

Potencia[mW]	1-2	1-3	1-4	1-5	1-6	1-7
1	11,1816	10,2727	9,2629	0,10174	0,0056	0,0066
30	12,6259	11,6252	12,6161	4,17333	1,4783	0,5674
60	13,0495	11,2366	12,7980	4,92927	2,5231	1,1214
90	12,9380	10,6929	11,9431	5,89751	3,3632	1,9623
120	13,0741	10,1007	10,3786	6,75035	3,5651	2,1382
150	5,2106	4,7353	5,3496	2,55944	4,0055	2,7109
180	6,1494	4,6305	5,3932	2,96657	3,5164	2,8184
210	5,8589	4,4969	5,2997	3,66249	3,5921	2,7131
250	5,6083	5,6779	4,2985	3,41254	3,6610	2,9536

TABLA 5.1 PRUEBAS POTENCIA AMBIENTE INDOOR.

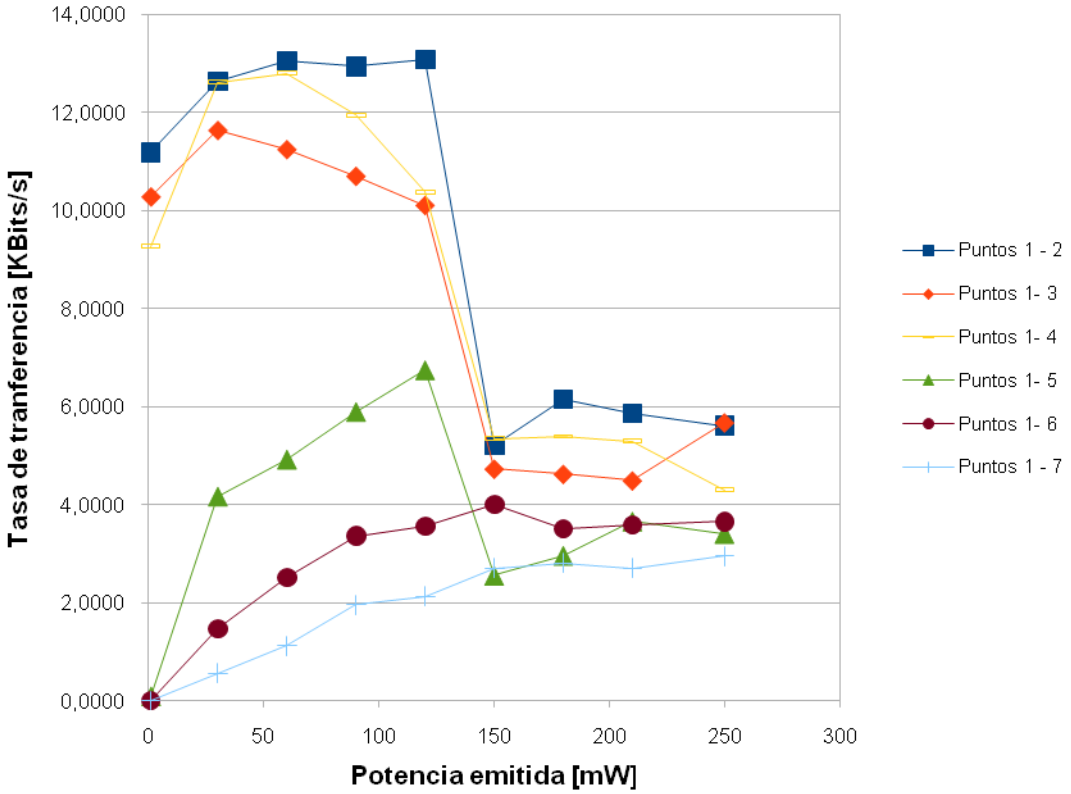


FIGURA 5.2 TASA DE TRANSFERENCIA VS POTENCIA INDOOR.

Detallando el comportamiento del ancho de banda en las cuatro primeras curvas se observa que todas entre 120 [mW] y 150 [mW] presentan una pendiente alta lo que muestra una caída en el ancho de banda, que luego se estabiliza. En las dos últimas gráficas no se aprecia esta pendiente, el fenómeno se comporta así ya que las pérdidas son tan altas que la tasa de transferencia es incluso menor al punto donde se estabiliza en los cuatro primeros casos y simplemente asciende hasta alcanzar el valor de los otros.
Cada punto en la figura anterior corresponde al promedio de la tasa de transferencia de la prueba realizada durante cien segundos a cierta potencia, a continuación se presenta una de las gráficas promediadas para obtener un punto.

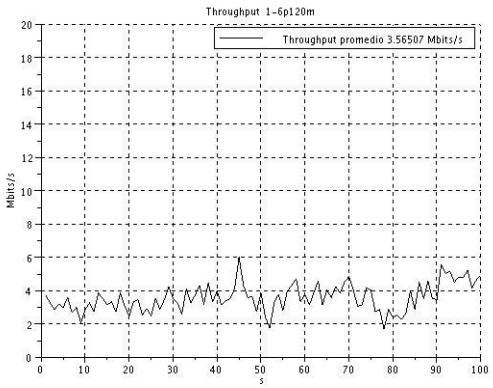


FIGURA 5.3 PRUEBA POTENCIA PUNTOS 1 Y 6 *INDOOR.*

De los resultados obtenidos se puede decir que la caída del throughput a partir de 120 [mW] ocurre debido a que la potencia emitida por los radios conlleva a problemas de no linealidad que afectan la señal e indirectamente el througput.

5.2 MEDICIÓN VARIACIONES DE POTENCIA RECIBIDA CONTRA DISTANCIA.

Para contrastar los datos obtenidos con atenuación por muros en ambientes *indoor* se realizó otra prueba en un ambiente *outdoor*, en la cancha de la Universidad industrial de Santander. Se ubicaron los enrutadores en los puntos A, B, C, D, E tal como se presenta en la siguiente figura y se realizaron las pruebas A-E, A-D, A-C y A-B de donde se obtienen las gráficas presentadas después del plano.

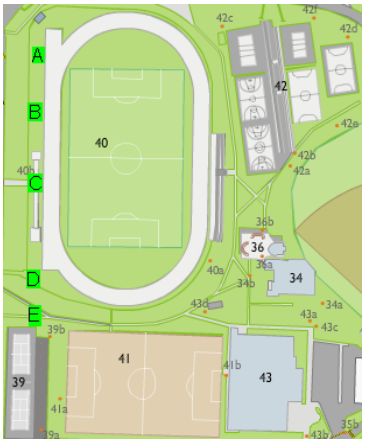


FIGURA 5.4 PLANO PRUEBAS AMBIENTA *OUTDOOR.*

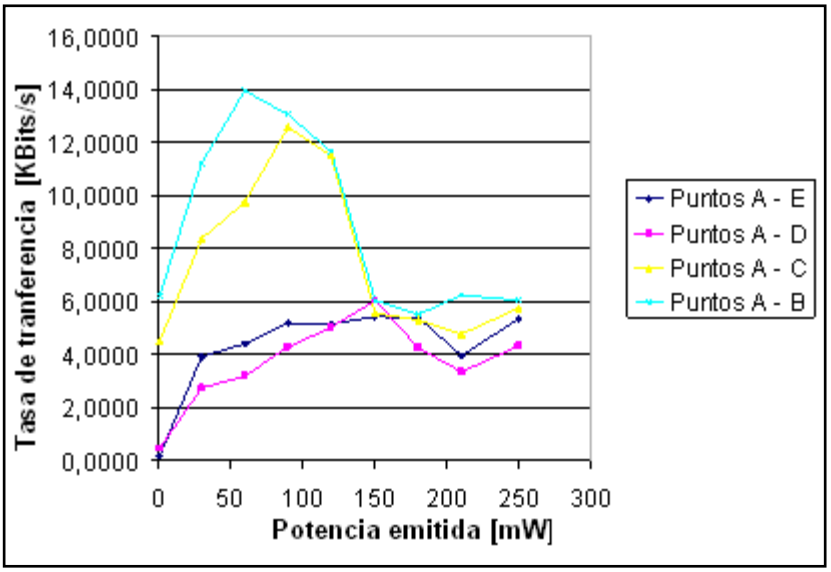


FIGURA 5.5 TASA DE TRANSFERENCIA VS POTENCIA *OUTDOOR.*

La gráfica anterior se relaciona bastante con la obtenida de las pruebas en ambientes *indoor*, en ambas se observa una clara caída del throughput a partir de 120 [mW] esto reafirma que la caída corresponde a problemas de linealidad como los que presenta cualquier sistema electrónico que utilice amplificadores, para los cuales la ganancia y el ancho de banda son inversamente proporcionales, así cuando se requiere una ganancia alta para superar los 120 [mW] de salida el ancho de banda cae y por tanto el throughput debe disminuir.

Potencia [mW]	**A - E**	**A - D**	**A - C**	**A - B**
1	0,1500	0,4448	4,4861	6,1941
30	3,9010	2,7173	8,3632	11,2135
60	4,4047	3,1997	9,7505	13,9681
90	5,1634	4,3043	12,5900	13,0444
120	5,1529	5,0315	11,5164	11,6632
150	5,4443	6,0295	5,5624	6,0434
180	5,3806	4,2578	5,3142	5,5228
210	3,9613	3,3592	4,7702	6,2382
250	5,3669	4,3298	5,7617	6,0222

TABLA 5.2 PRUEBAS POTENCIA AMBIENTE *OUTDOOR.*

El estudio del throughput vs potencia se realizo de manera experimental, sin tener en cuenta modelos de radio propagación ni simulaciones, el desarrollo de estas representaría un trabajo extenso y fuera del alcance del presente proyecto. Además para el desarrollo de modelos de radio propagación se requieren datos estructurales del edificio en planos CAD, con las respectivas capas y los materiales utilizados, datos que no están disponibles en el momento. No sobra afirmar que un trabajo de simulación de propagación puede llegar a ser un trabajo de grado por si solo lo que justifica nuestra metodología para determinar la relación througthput vs potencia.

5.3 DISEÑO Y CONSTRUCCIÓN DE PROTOTIPO DE NODO MALLA.

Para el desarrollo del prototipo del nodo primero se analizaron los elementos que debía poseer cada nodo. Dos enrutadores Linksys WRT54GL para que uno trabaje como columna vertebral de la rede tipo malla y brinde enrutamiento, el otro se configura para que su interfaz inalámbrica trabaje en la modalidad de punto de acceso inalámbrico, los dos enrutadores están conectados por su interfaz LAN de manera que trabajen en modo puente y pertenezcan a la misma subred 192.168.x.0/24.

Cada enrutador debe estar alimentado, aquí surge un punto importante en la diseño y es precisamente ¿cómo se va alimentar el dispositivo? los autores del presente texto decidieron que lo mejor es que el nodo posea una batería que sirva como fuente alterna en caso de que no exista una conexión cerca al lugar donde se va realizar la prueba, esta condición a su vez sirve como punto de partida para el futuro diseño de un nodo autónomo en que la batería sea cargada por energía solar por ejemplo. Para esto se decidió utilizar un juego de dos interruptores doble tiro que pueden conmutar la alimentación de la toma de corriente alterna a una batería que funcione como alimentación, sobre una regleta se conectan los adaptadores de cada enrutador y de allí se dirigen a los interruptores, de la misma forma la batería se conecta a los interruptores, estos seleccionan una de las dos fuentes y dirigen el fluido eléctrico directamente a los equipos, el esquema se presenta en la figura a continuación.

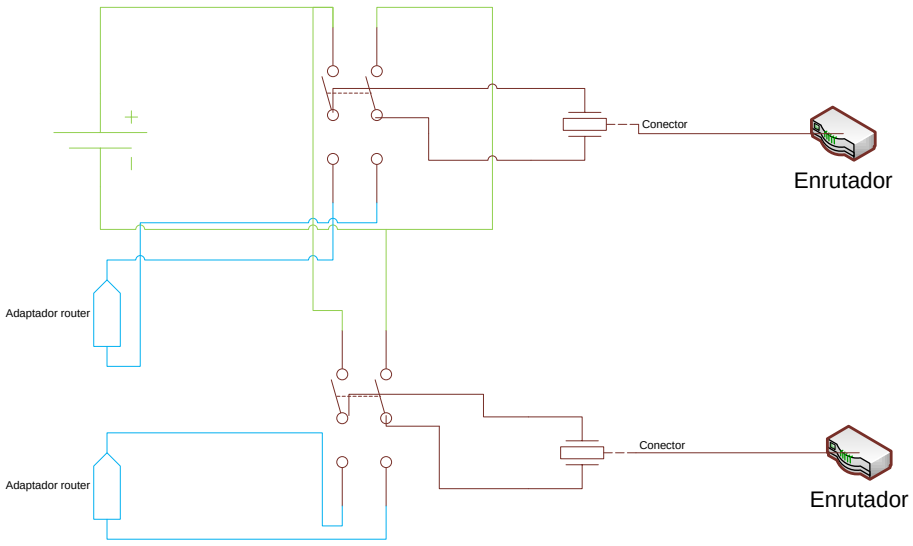


FIGURA 5.6 ESQUEMA ELÉCTRICO INTERCAMBIO ALIMENTACIÓN.

El montaje completo se realizo en una caja para exteriores *tableplast* de la serie G, los enrutadores se retiraron de sus cajas y se montaron sobre laminas plásticas separadas de las paredes de la caja y así evitan el calentamiento, además las perforaciones necesarias se hacen sobre la lamina sin dañar la caja, de esta forma los agujeros sobre la caja son únicamente los necesarios para sostener la lamina y sacar las antenas. Los resultados se presentan en la siguiente imagen:

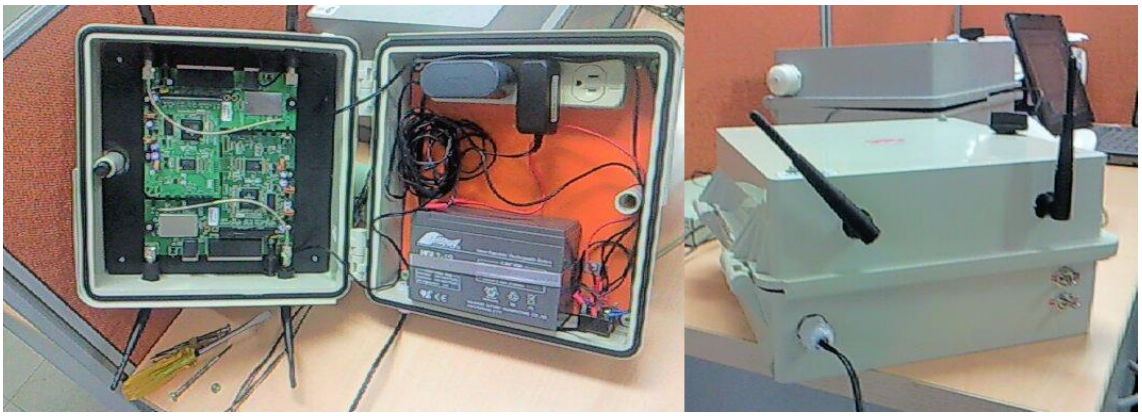


FIGURA 5.7 FOTOGRAFÍA PROTOTIPO NODO MALLA.

Este diseño es útil en ambientes *outdoor* en cuyo caso la batería sería usada más frecuentemente y se recomienda por tanto un regulador, dentro de la caja hay espacio suficiente para incluirlo, sería interesante combinar esto con un panel solar u otra fuente de energía alternativa que sea versátil en zonas rurales o sin conexión eléctrica. La caja seleccionada tiene protección UV, puede resistir hasta 70 ºC sin deformarse y cumple con las especificaciones para uso en exteriores.

Los elementos que constituyen el prototipo de nodo malla tienen un costo:
2 enrutadores Linksys WRT54GL -> $120 dólares.
Batería de alimentación 12V-12Ah -> $35 dolares.
Caja tablepast serie G -> $75 dolares.
Otros implementos -> $10 dolares.
Para un costo total de alrededor de $240 dolares, que es menor al 50% de nodos para redes tipo malla (routers Mesh) para ambientes outdoor que se encuentran en el mercado (31).

5.4 TOPOLOGÍAS BANCO PRUEBAS.

Debido a que las pruebas se van a realizar en ambientes *indoor* y el radio de cobertura de los enrutadores con la potencia de emisión óptima para obtener una mayor tasa de transferencia excede el área física de trabajo, y con el fin de obtener las topologías deseadas es necesario realizar filtrado por MAC. Se debe tener en cuenta que las pruebas fueron realizadas en un lugar con interferencia, aún así usando herramientas de análisis de espectro como el programa ***channalizer*** combinado con el ***WiSpy*** se selecciono el canal con menos interferencia en el lugar**.**

5.4.1 TOPOLOGÍA MALLA 3X3 CON CONEXIÓN ETHERNET

Se realizó un arreglo matricial compuesto de nueve nodos, en el que cada nodo está enlazado únicamente con el enrutador adyacente, es decir a una fila o a una columna de distancia exclusivamente como se presenta en la figura. Los nodos entre los que se realiza la prueba se conectan vía Ethernet al enrutador.

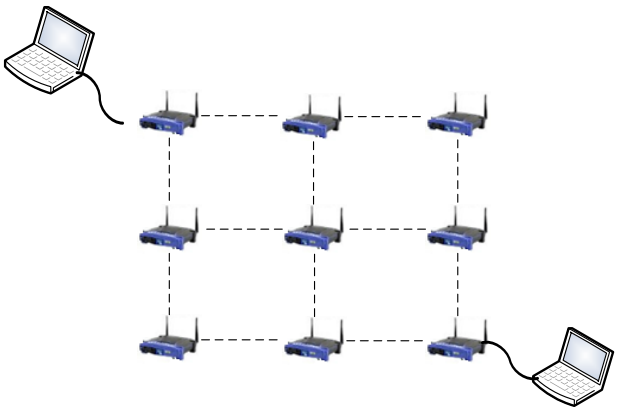
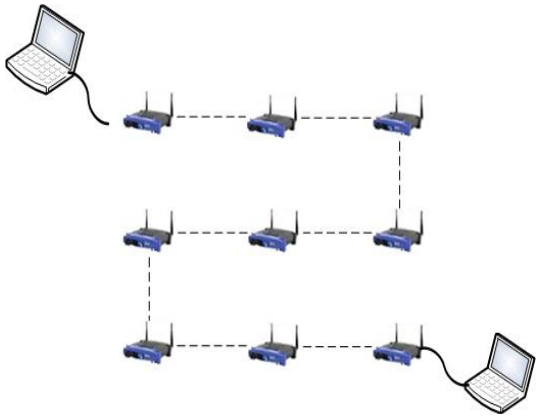


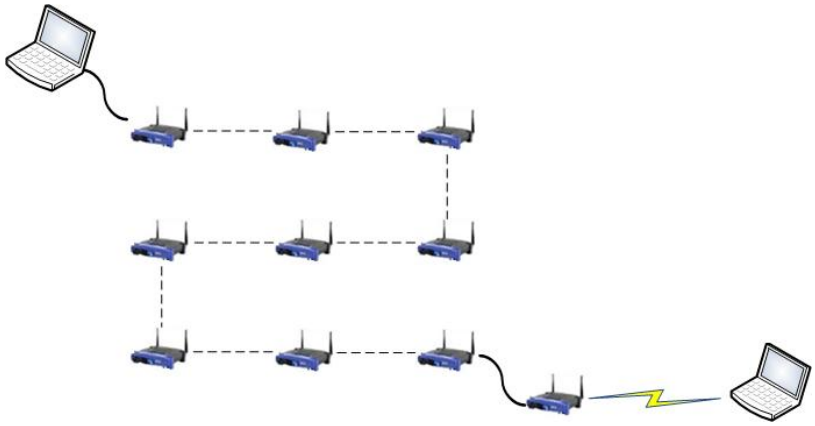
FIGURA 5.8 TOPOLOGÍA 3X3 CON CONEXIÓN ETHERNET.

5.4.2 TOPOLOGÍA MULTISALTO EN LÍNEA

Se configuran los enrutadores de tal forma que cada uno tenga enlazados máximo dos nodos adyacentes, como se observa a continuación, cada prueba se realiza entre dos estaciones ya sean conectadas vía Ethernet ambas ó una de ellas a un AP. Ambas situaciones se presentan en las figuras a continuación.



5.9 TOPOLOGÍA MULTISALTO EN LÍNEA



5.10 TOPOLOGÍA MULTISALTO EN LÍNEA CON AP

5.4.3 TOPOLOGÍA MALLA 5 NODOS DE ACCESO INALÁMBRICO

Se configuran los enrutadores para crear la malla de dos filas y tres columnas pero eliminando el primer elemento de la matriz y se le adiciona un punto de acceso inalámbrico que está conectado vía Ethernet. Así cada nodo brinda enrutamiento y acceso inalámbrico a la red, estos nodos están construidos tal como se presenta en el esquema denominado topología 5 nodos de acceso inalámbrico y se dispuso de manera física como se presenta luego en el plano de la topología 5 nodos de acceso inalámbrico.

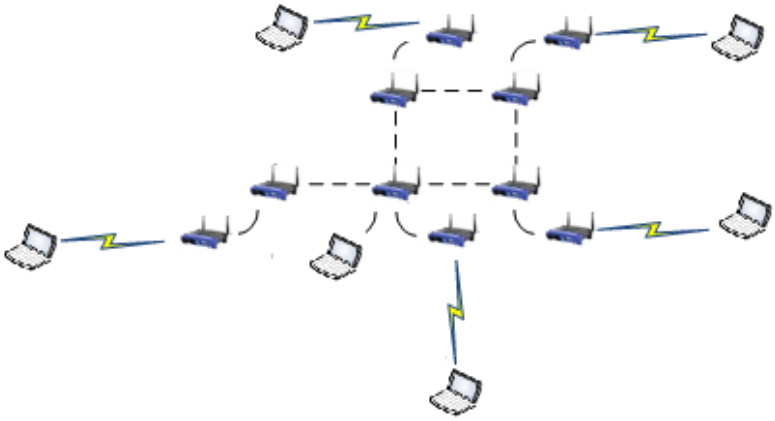


FIGURA 5.11 TOPOLOGÍA 5 NODOS DE ACCESO INALÁMBRICO.

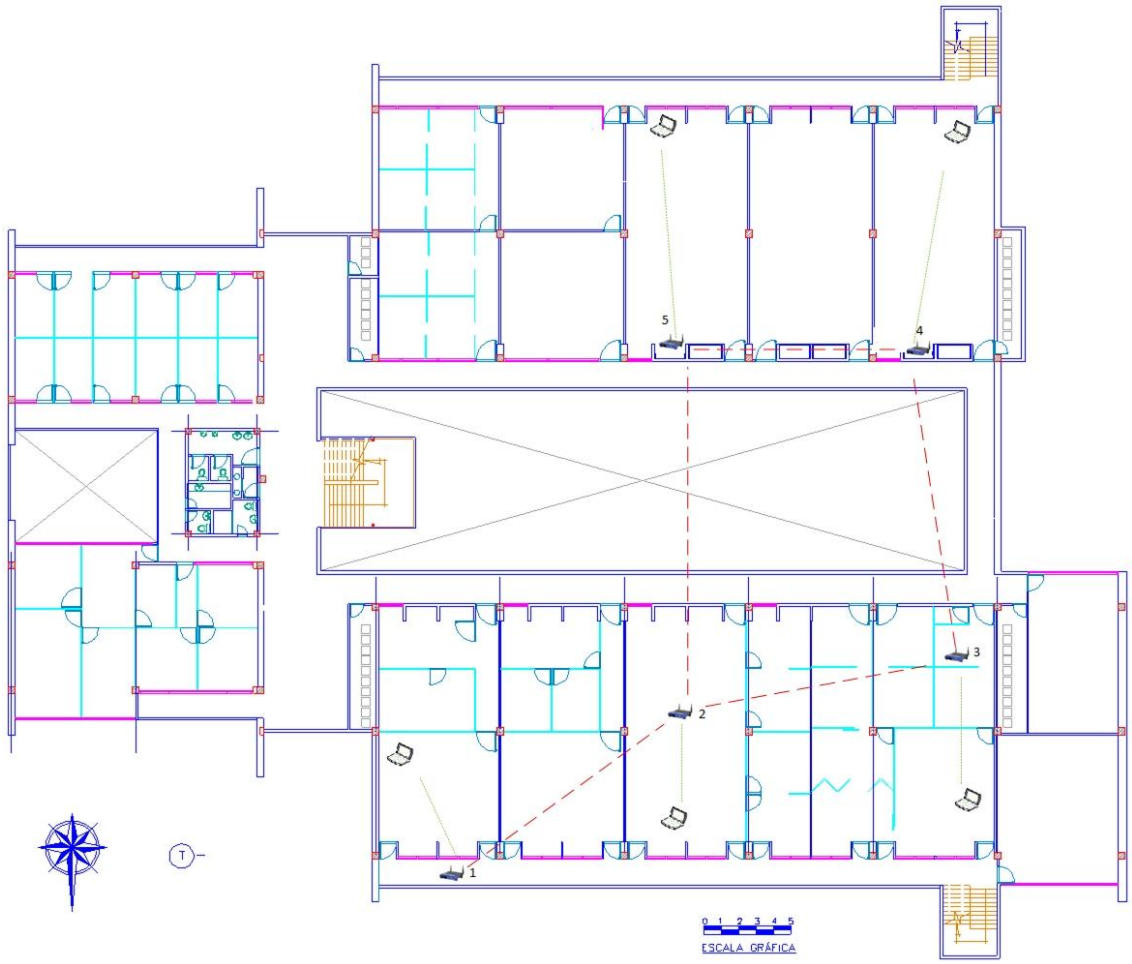


FIGURA 5.12 PLANO IMPLEMENTACIÓN DE LA TOPOLOGÍA 5 NODOS DE ACCESO INALÁMBRICO.

5.5 IMPLEMENTACIÓN DE LAS TOPOLOGÍAS

Para implementar las topologías anteriores fue necesario realizar filtrado por MAC ya que los enrutadores quedan en el rango de cobertura de los demás por lo que cada nodo quedaría con conexión directa con los otros. Para el filtrado se puede usar el comando iptables como se encuentra en múltiples trabajos, sin embargo la interfaz gráfica del OPENWRT permite realizar el filtrado por lo que se usó esta herramienta para configurar las topologías expuestas en el apartado 5.4. Para el filtrado se configuran las reglas de la siguiente manera, las primeras reglas aceptan paquetes UDP, TCP e ICMP de los nodos que se quieren tener como vecinos directos o conexión directa, después se coloca una regla que rechace todos los paquetes UDP, TCP e ICMP entrantes. En cualquier tabla de filtrado las reglas se van leyendo en orden y si alguna se cumple se aplica, por lo tanto si alguna de las primeras se cumple el paquete es aceptado, en caso contrario va llegar a la última y se descarta.

5.6 DESEMPEÑO DE LAS TOPOLOGÍAS

Para medir el desempeño de las topologías planteadas se utilizó la siguiente metodología que consiste en (12):

- Medir la tasa de transferencia desde cada nodo de la malla hacia todos los demás, es decir en una malla de **n x m** nodos se realizarían [n x m] x [(n x m) – 1] pruebas.
- Para cada prueba se lanza una sesión TCP de 10 segundos con la herramienta de medición Iperf.
- Para cada prueba se lanza una sesión UDP de 10 segundos con la herramienta de medición Iperf, esta permite medir también el *Jitter*.
- Para cada prueba se lanza un ping de 10 segundos y 84 bytes para medir el retraso.

Las pruebas se pueden realizar a diez segundos ya que este es el tiempo de establecimiento de la herramienta IPERF, es decir el tiempo mínimo para que la medición sea correcta (29).

Con los resultados obtenidos de la tasa de transferencia, retraso y *Jitter* de cada nodo hacia los demás se realiza un promedio con el que se da un valor representativo de la tasa de transferencia de la red para un determinado protocolo de enrutamiento.

Los resultados se agrupan de acuerdo al número de saltos necesarios para llegar al destino final, estos se promedian para obtener un único valor característico con los cuales se obtiene la gráfica, número de saltos contra ancho de banda. También se realiza lo mismo con la métrica de *Jitter* con el fin de obtener la gráfica de número de saltos vs *Jitter*.

5.6.1 DESEMPEÑO OLSR

La prueba se realizo tal como se describe en el apartado anterior, para la topología 3 x 3 con conexión Ethernet y corriendo el protocolo OLSR en el firmware OPENWRT. Para esta malla existen 72 combinaciones según la metodología planteada, sin embargo para cada nodo se mide TCP, UDP, *Jitter* y retraso lo que da un total de 288 datos. Estos se agruparon teniendo en cuenta el número de saltos realizados para llegar a su destino, las cuatro tablas se encuentran en los anexos.

De los resultados se puede inferir que el número de saltos tiene un impacto fuerte sobre las métricas, a medida que el número de saltos aumenta la tasa de transferencia tanto en TCP como en UDP disminuyen mientras que el *Jitter* y el retraso aumentan, de manera más concreta el desempeño disminuye notablemente con el número de saltos. Estos resultados junto a las pruebas de potencia permiten ver que para el diseño de una red enmallada se debe tener en cuenta la reducción de la tasa de transferencia debido potencia recibida y la reducción que generaría incluir otro enrutador, si se comparan estas reducciones se puede determinar el número de nodos necesarios para obtener el mejor desempeño dentro de la

cobertura requerida. Para visualizar el desempeño de la red enmallada usando el protocolo de enrutamiento OLSR, los promedios de las métricas para uno, dos, tres y cuatro saltos se agrupan en la siguiente tabla y se obtienen las gráficas que se presentan a continuación.

Número de Saltos	TCP [KBits/s]	UDP [KBits/s]	JITTER [ms]	DELAY [ms]
1	19269,1	19171,2	2.945	2,000
2	8261,0	11227,8	3.864	3,107
3	5614,9	7429,4	2.966	4,188
4	4232,0	5491,5	4.687	5,500

TABLA 5.3 PROMEDIOS MÉTRICAS RED ENMALLADA OLSR.

A continuación se presentan las gráficas de ancho de banda vs número de saltos, se observa que el ancho de banda es aproximadamente el ancho de banda entre dos nodos sobre el número de saltos, tanto para TCP como para UDP.

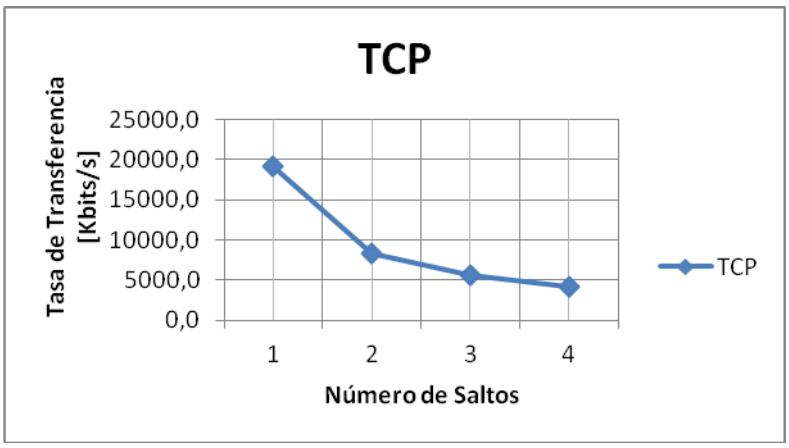


FIGURA 5.13 TASA DE TRANFERENCIA EN TCP CONTRA NÚMERO DE SALTOS OLSR.

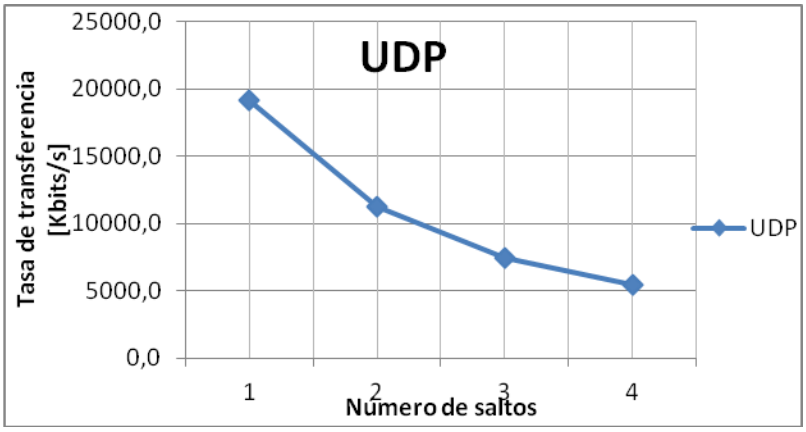


FIGURA 5.14 TASA DE TRANSFERENCIA EN UDP CONTRA NÚMERO DE SALTOS OLSR.

Ahora se presentan las gráficas *jitter* vs número de saltos y retraso vs número de saltos, es claro que a medida que el número de saltos aumenta lo mismo sucede con el retraso, no se puede decir lo mismo del comportamiento del *jitter.*

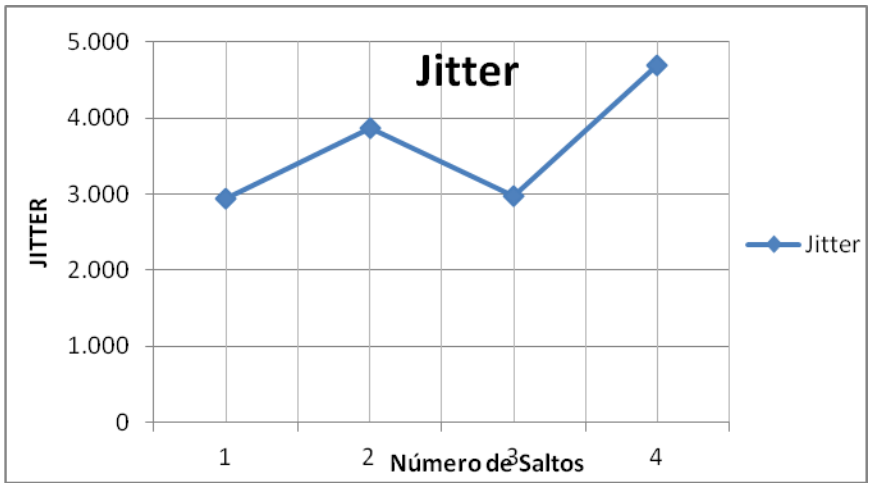


FIGURA 5.15 JITTER CONTRA NÚMERO DE SALTOS OLSR

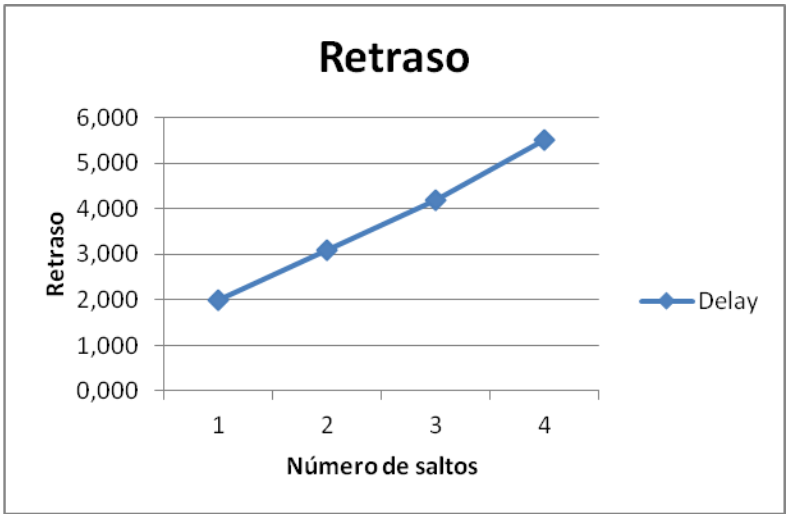


FIGURA 5.16 RETRASO CONTRA NÚMERO DE SALTOS OLSR.

5.6.2 DESEMPEÑO B.A.T.M.A.N.

De la misma manera como se realizaron las pruebas de OLSR se realiza la prueba con B.A.T.M.A.N. Los datos se procesan de manera similar obteniendo como resultado la siguiente tabla:

Número de saltos	TCP [KBits/s]	UDP [KBits/s]	JITTER [ms]	DELAY [ms]
1	13923,7	17250,2	3.319	2,118
2	7047,8	9543,6	3.423	3,591
3	4896,0	6398,1	13.099	5,286
4	3573,3	4665,3	4.926	6,000

TABLA 5.4 PROMEDIO MÉTRICAS ENMALLADAS.

Las gráficas de ancho de banda en Kbits/s contra número de saltos tanto en TCP como en UDP se presentan a continuación, se observa una caída de la tasa de transferencia respecto al número de saltos y al igual que en OLSR la tasa de transferencia es aproximadamente la encontrada entre dos nodos vecinos dividida en el número de saltos.

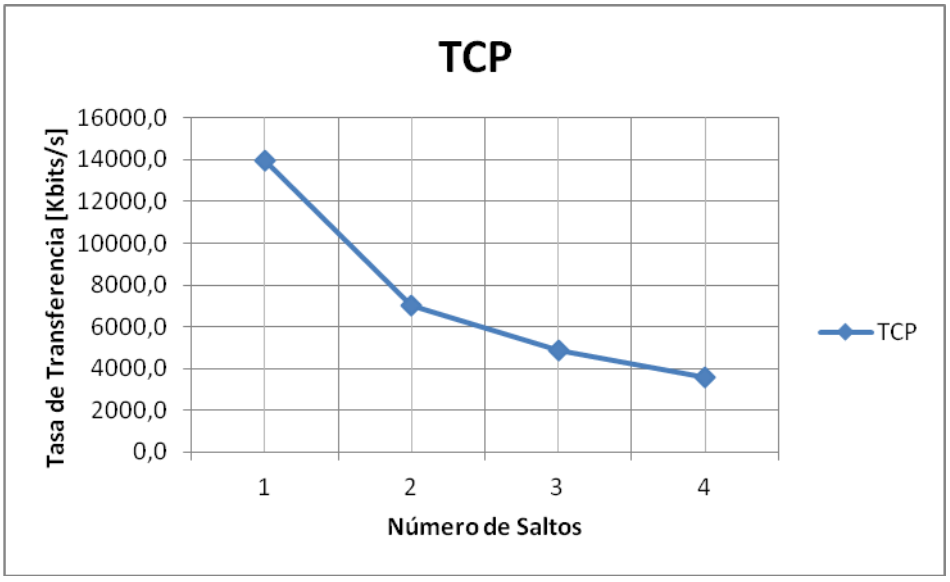


FIGURA 5.17 TASA DE TRANFERENCIA EN TCP CONTRA NÚMERO DE SALTOS BATMAN

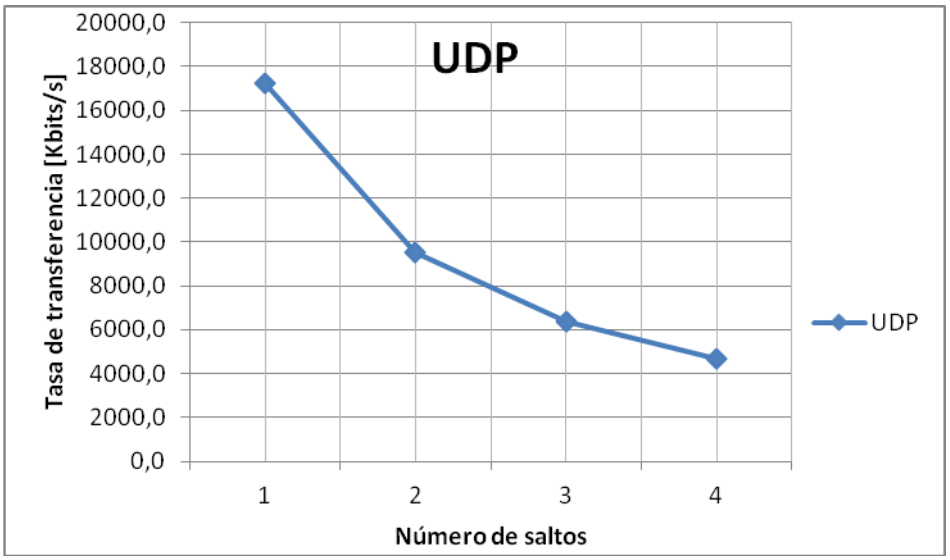


FIGURA 5.18 TASA DE TRANSFERENCIA EN UDP CONTRA NÚMERO DE SALTOS BATMAN

Ahora se presenta el resultado del retraso contra el número de saltos y del *jitter* contra el número de saltos, se observa también un aumento en el retraso a medida que aumenta el número de saltos.

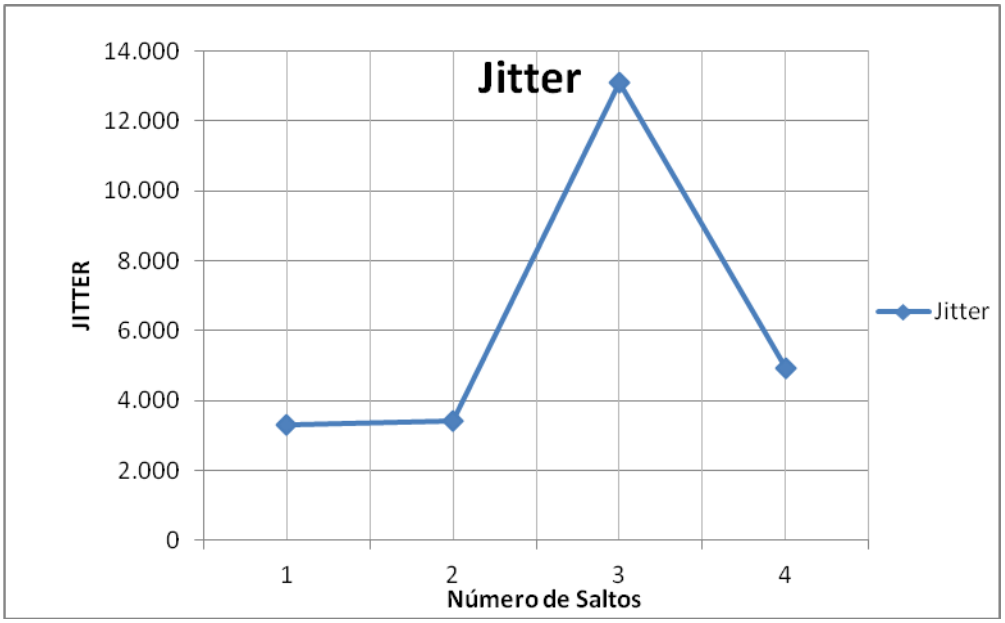


FIGURA 5.19 JITTER CONTRA NÚMERO DE SALTOS BATMAN

FIGURA 5.20 RETRASO CONTRA NÚMERO DE SALTOS BATMAN

5.6.3 COMPARACIÓN DE DESEMPEÑOS

Al analizar el comportamiento del ancho de banda respecto al número de saltos en TCP y UDP para ambos protocolos simultáneamente se observa una diferencia en la máxima tasa de 2 [Kbits/s], esta diferencia al igual que la tasa de transferencia se reduce a medida que aumenta el número de saltos, esta diferencia es aproximadamente la diferencia máxima dividida en el número de saltos, el fenómeno se observa en la siguientes dos gráficas.

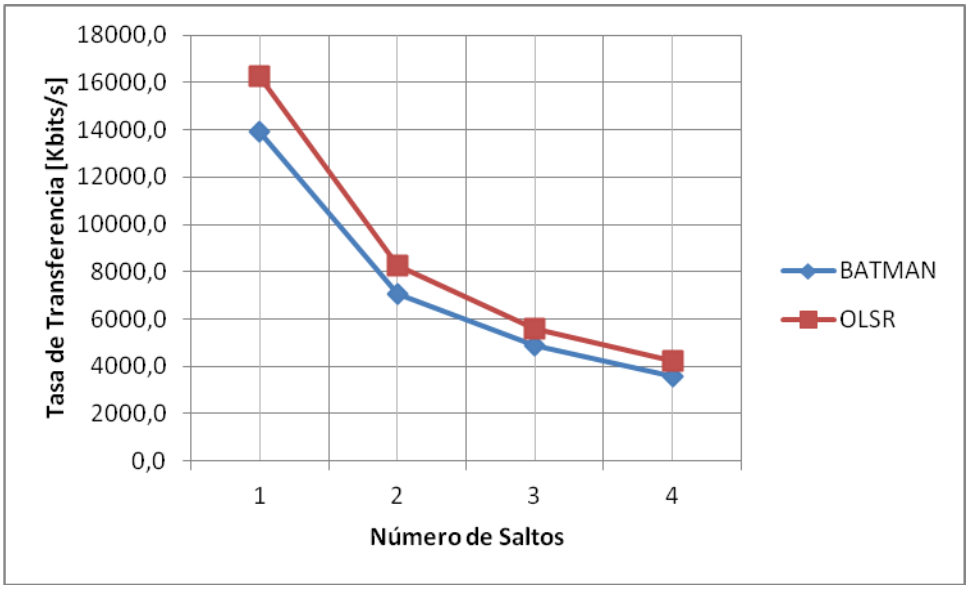


FIGURA 5.21 COMPARACIÓN PROTOCOLOS TCP

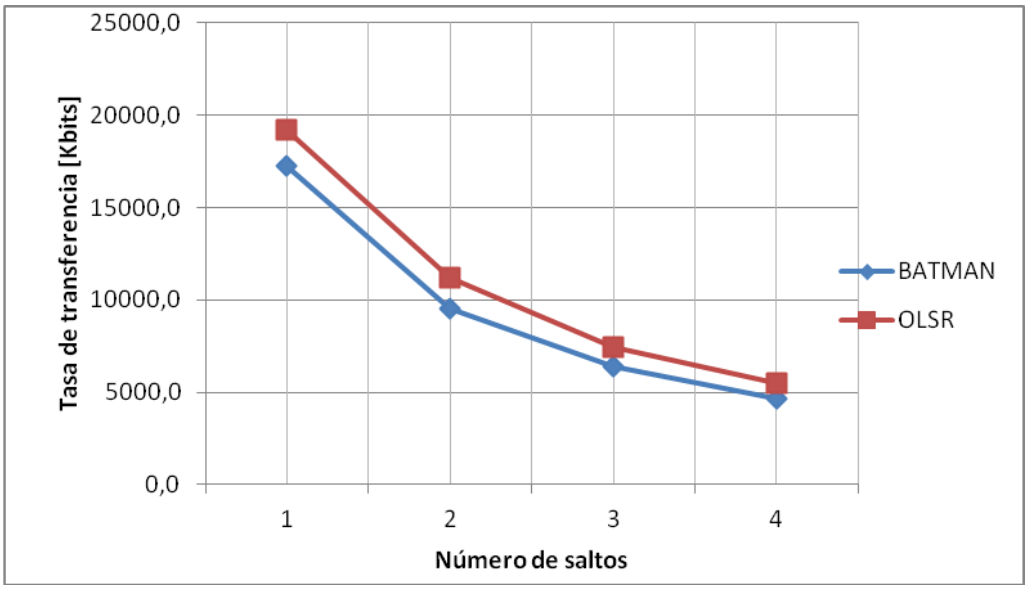


FIGURA 5.22 COMPARACIÓN PROTOCOLOS UDP

La diferencia de ancho de banda entre ambos protocolos se puede explicar debido a la cantidad de tráfico inyectado por cada uno de los protocolos, el tráfico de enrutamiento de OLSR corresponde a los mensajes *Hello* que son enviados a todos los vecinos pero únicamente cada cierto tiempo denominado intervalo *Hello* y la información de enrutamiento que es transmitida únicamente a los MPR, al parecer este flujo de tráfico es mucho menor que el suministrado a través de los mensajes OMG imprimidos a la red por BATMAN, uno de los motivos es que estos OMG se envían en difusión a diferencia de OLSR que solo lo envía a los OLSR, otro motivo es que el envío de los OMG es constante y se hace mucho más a menudo que la información de enrutamiento de OLSR que es enviada únicamente cuando hay modificaciones en la red y solo por los MPR. Comparando el *jitter* y retraso se observan comportamientos similares para ambos protocolos, aún así OLSR tiene un mejor desempeño en estos dos ítems también.

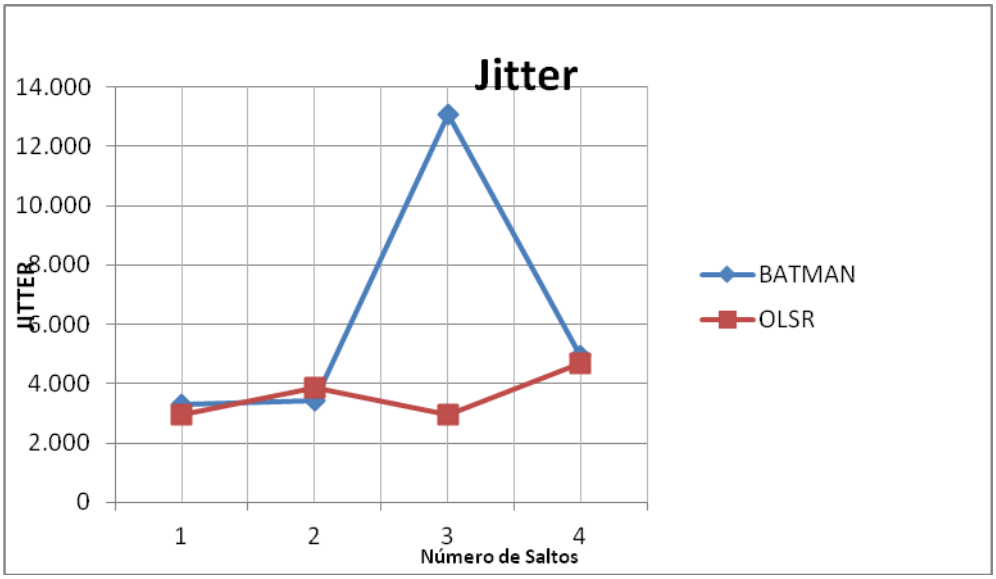


FIGURA 5.23 COMPARACIÓN PROTOCOLOS *JITTER.*

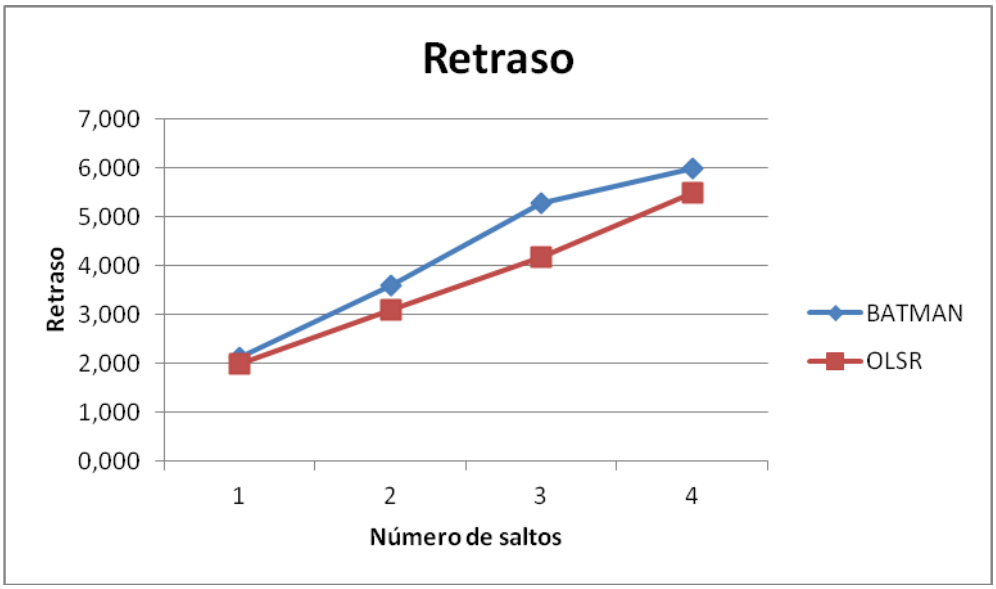


FIGURA 5.24 COMPARACIÓN PROTOCOLOS RETRASO.

Ahora se realiza el mismo análisis pero para la topología de multisalto en línea descrita en el apartado 5.4.3, tanto con conexión final vía Ethernet como vía AP.

Para iniciar se presentan los resultados de TCP para los dos protocolos a través de las tres gráficas que se muestran a continuación.

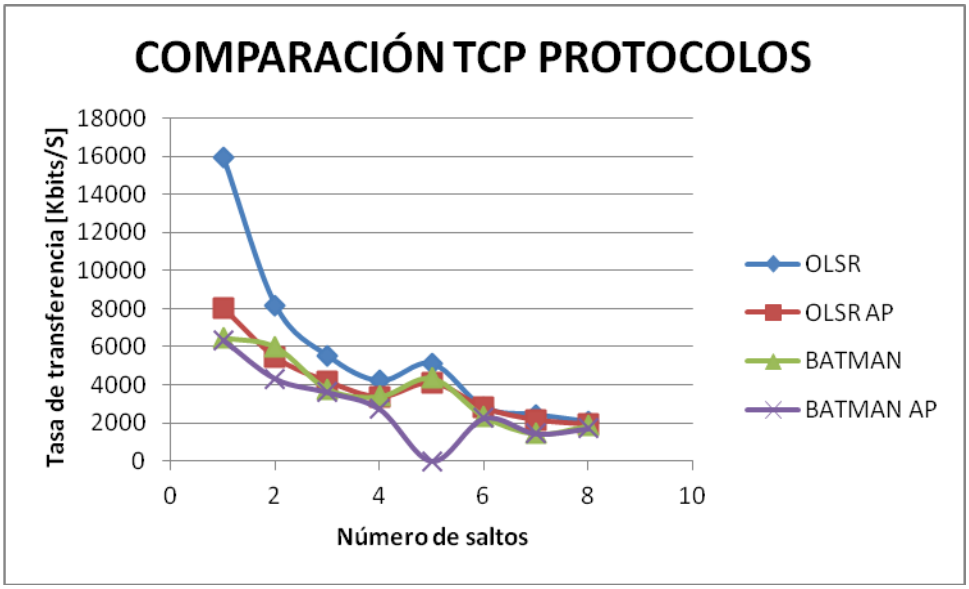


FIGURA 5.25 COMPARACIÓN TCP DE LAS PRUEBAS MULTISALTO.

La anterior figura muestra los resultados en TCP para OLSR y BATMAN con ambos nodos conectados vía Ethernet y también con uno de los nodos conectados vía AP, el trazo en color azul que corresponde a OLSR conectado vía Ethernet fue el que dio mejores resultados, como se observa la tasa de transferencia máxima es aproximadamente el doble de la obtenida con AP que es la traza roja, por otro lado BATMAN da un desempeño aún más bajo ya que la máxima tasa inicia en aproximadamente 6 [Mbits] que corresponde a la sexta parte de los 16 [Mbits] obtenidos con OLSR conexión Ethernet. Se observa que al igual que en OLSR el AP disminuye la tasa de transferencia de BATMAN, y por lo tanto el trazo morado siempre está por debajo de los demás dando el peor desempeño comparado con los otros tres. A continuación se presenta la comparación de las dos topologías pero con cada protocolo para que se pueda visualizar mejor como afecta el AP la tasa de transferencia.

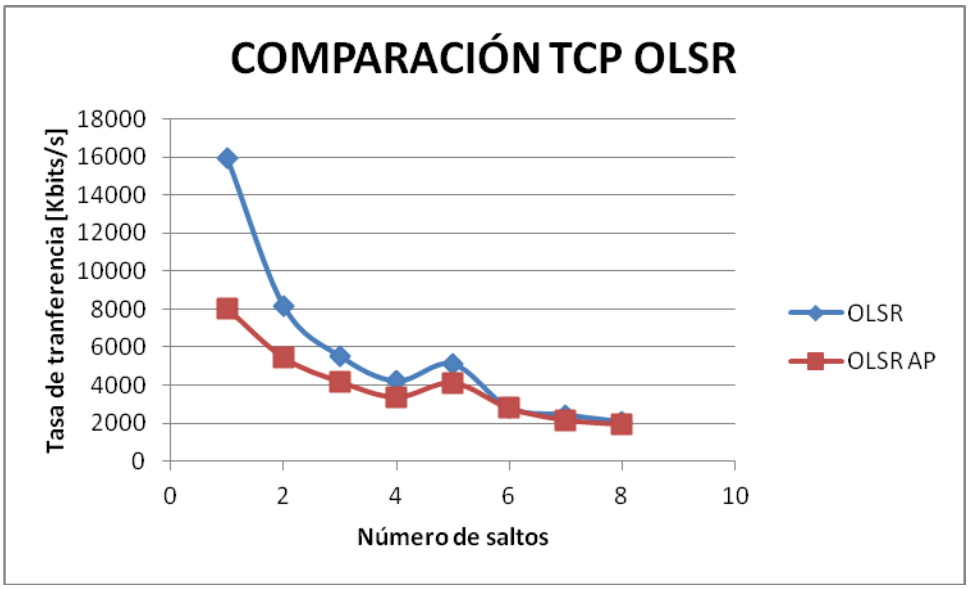


FIGURA 5.26 COMPARACIÓN TOPOLOGÍA MULTISALTOS TCP OLSR

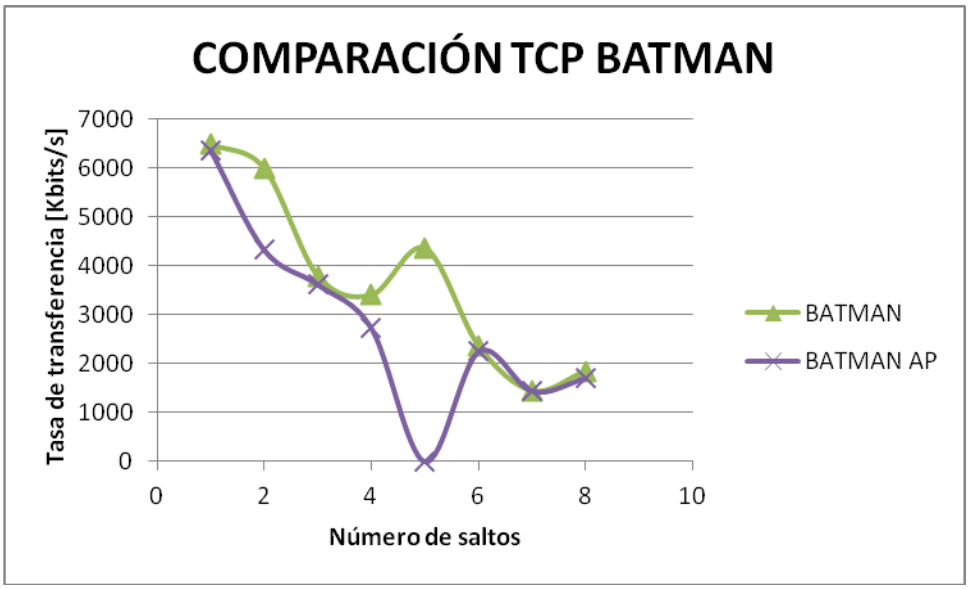


FIGURA 5.27 COMPARACIÓN TOPOLOGÍA MULTISALTOS TCP BATMAN.

Ahora se analizan los resultados obtenidos en UDP y se observa un comportamiento bastante similar aunque la diferencia de tasa de transferencia entre OLSR y BATMAN no es tan drástica como en TCP, además entre los saltos 2, 3,4 la topología BATMAN Ethernet logra superar en tasa de transferencia a OLSR con AP, aún así de forma general se puede decir que OLSR supera a BATMAN y que la topología con conexión vía AP siempre presenta menor desempeño que vía Ethernet.

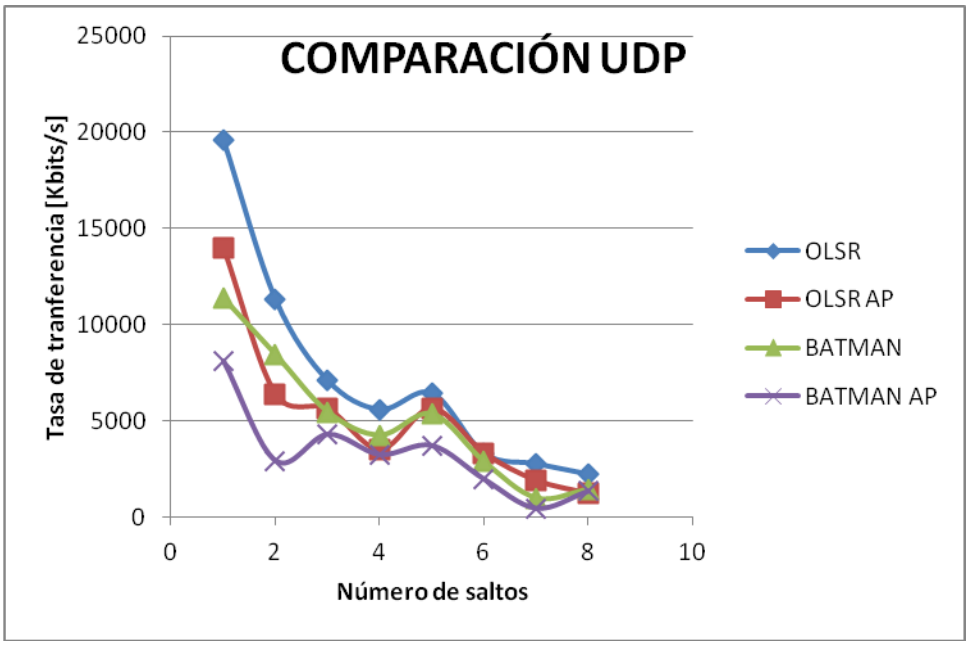


FIGURA 5.28 COMPARACIÓN UDP DE LAS PRUEBAS MULTISALTOS.

Aunque la diferencia entre los protocolos se reduce la diferencia entre utilizar Ethernet y AP para UDP aumenta y los trazos en general están un poco más separados que para TCP.

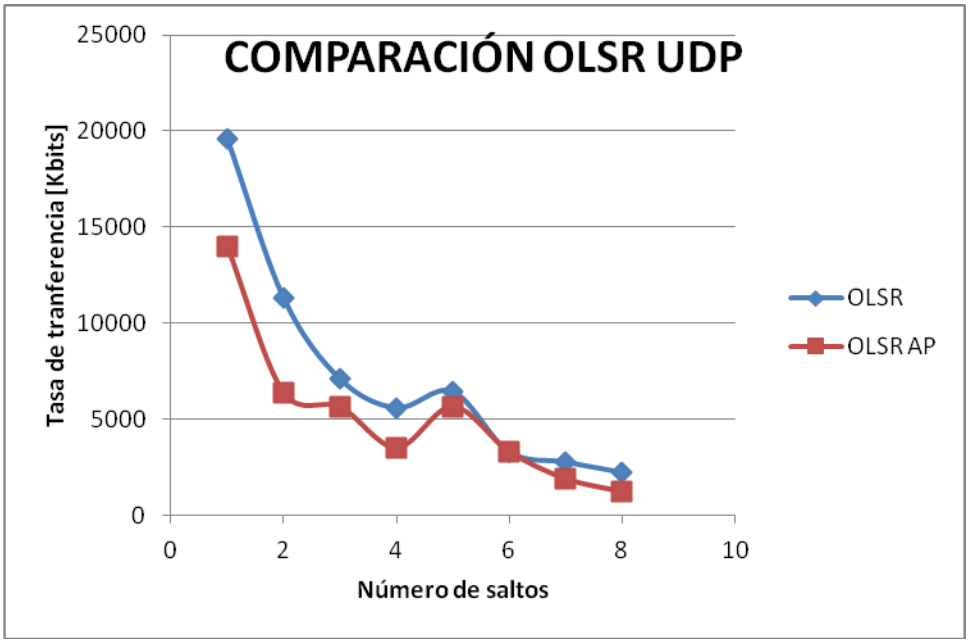


FIGURA 5.29 COMPARACIÓN TOPOLOGÍA MULTISALTOS UDP OLSR.

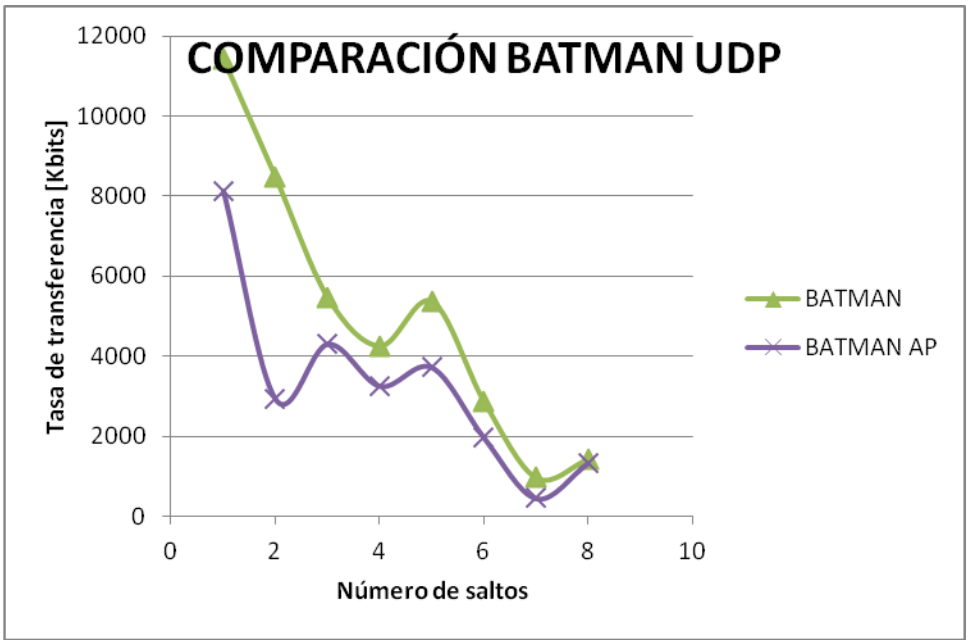


FIGURA 5.30 COMPARACIÓN TOPOLOGÍA MULTISALTOS UDP OLSR.

Siguiendo con la misma estrategia de comparación primero se colocan el *Jitter* de los dos protocolos con y sin AP, se ve una breve ventaja de OLSR respecto a BATMAN, pero si se observan las gráfica por separado se reconoce que durante ciertos saltos el *jitter* de OLSR es mayor, es necesario analizar las gráficas por separado ya que el problema de escala cuando están todas juntas no permite observar en detalle las diferencias en los valores de *Jitter* inferiores a 20 segundos. De 1 a 5 saltos es mejor BATMAN, pero de allí en adelante

se ve una ventaja clara de OLSR, por lo tanto de esta prueba no se puede definir cual protocolo es mejor en cuanto a *Jitter.*

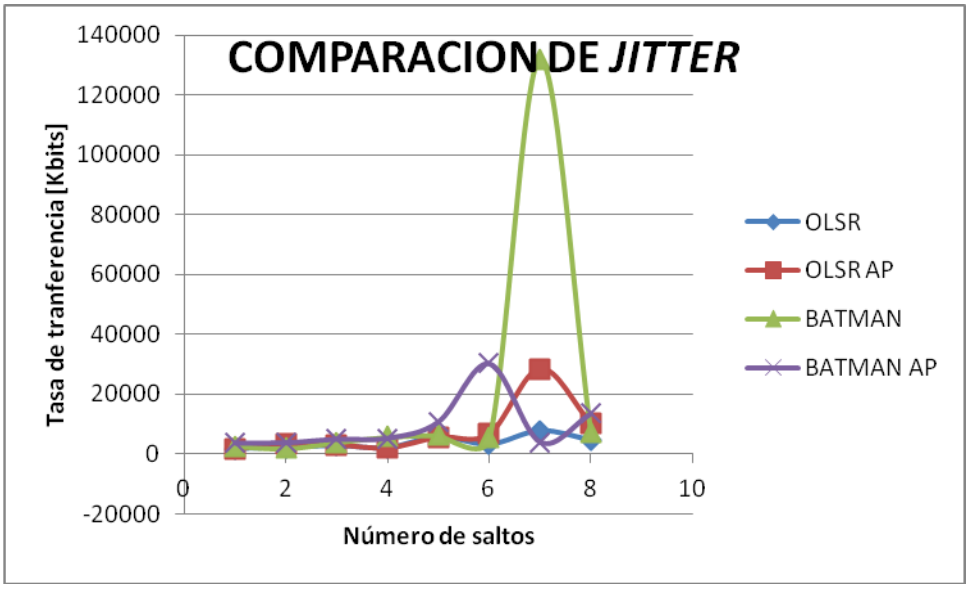


FIGURA 5.31 COMPARACIÓN TOPOLOGÍA MULTISALTOS *JITTER.*

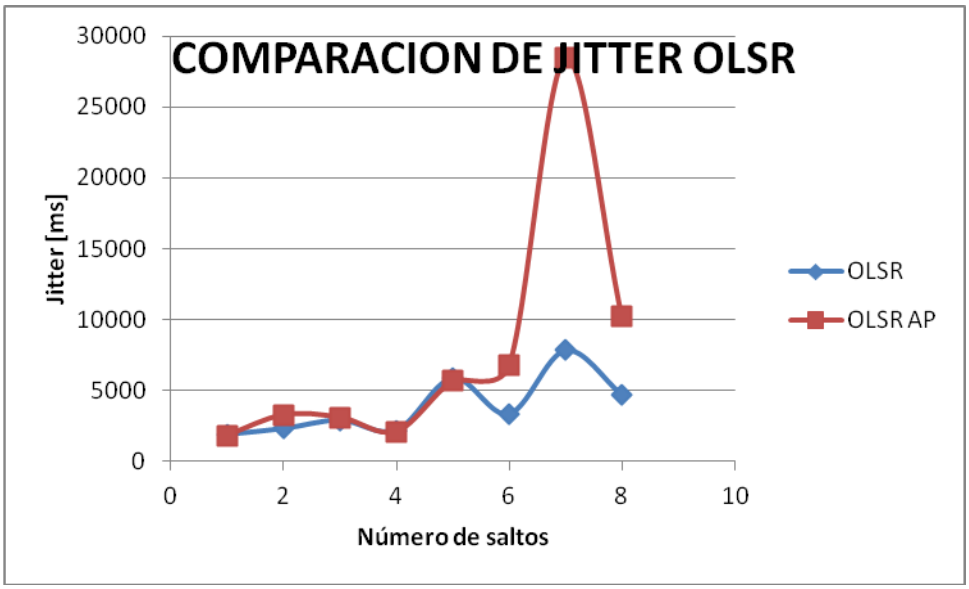


FIGURA 5.32 COMPARACIÓN TOPOLOGÍA MULTISALTOS OLSR *JITTER.*

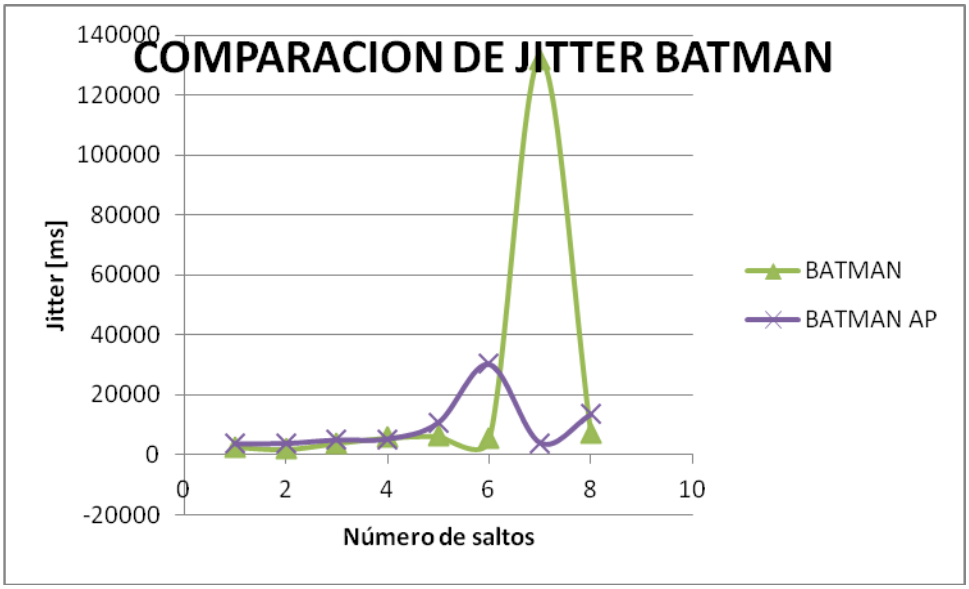


FIGURA 5.33 COMPARACIÓN TOPOLOGÍA MULTISALTOS BATMAN *JITTER.*

La última métrica que se analizó es el retraso, claramente se aprecia que el AP genera un retraso altísimo lo que hace insignificante el retraso producido por la malla como tal, por otro lado con punto de acceso OLSR presenta un mejor desempeño ya que tiene un retraso más bajo que BATMAN, sin embargo en las gráficas correspondientes a la topología conectadas vía Ethernet la diferencia entre los protocolos es mínima.

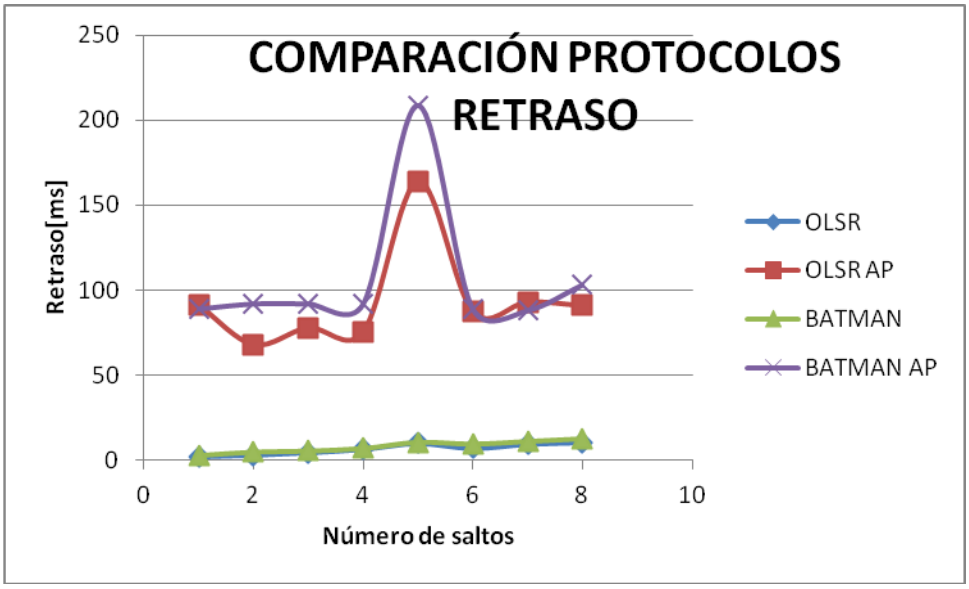


FIGURA 5.34 COMPARACIÓN TOPOLOGÍA MULTISALTOS RETRASO.

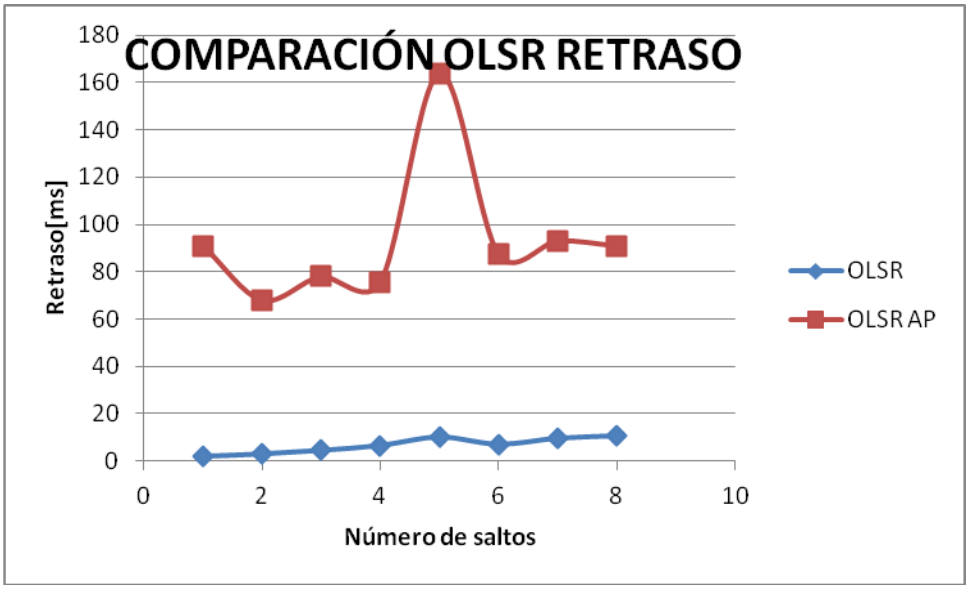


FIGURA 5.35 COMPARACIÓN TOPOLOGÍA MULTISALTOS RETRASO OLSR

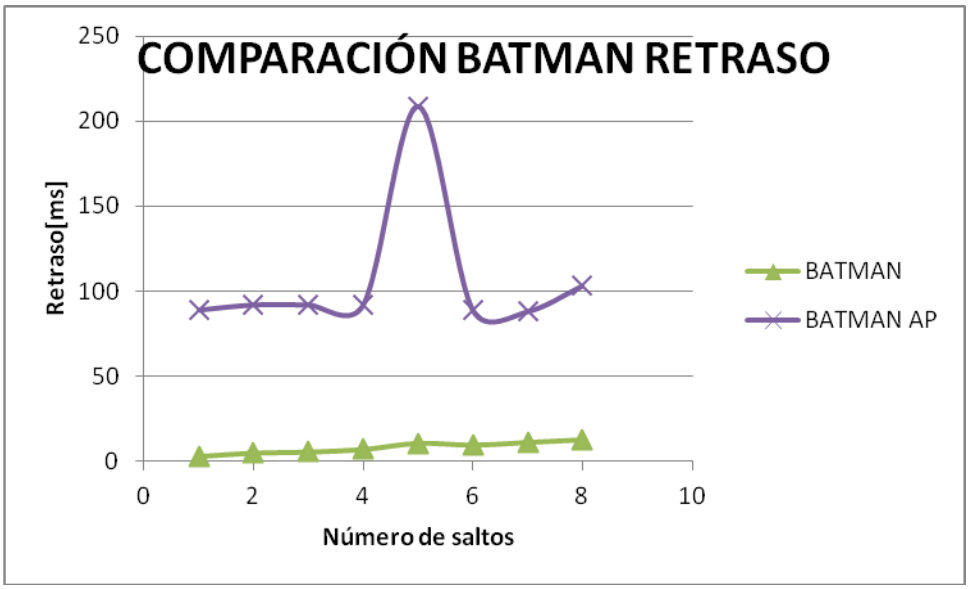


FIGURA 5.36 COMPARACIÓN TOPOLOGÍA MULTISALTOS RETRASO BATMAN.

Para concluir este capítulo se puede decir que en términos generales el protocolo OLSR presento un desempeño mejor que BATMAN, sin embargo se puede inferir con un análisis teórico que esto se debe a la topología de la red y sus condiciones, principalmente el hecho que esta es una red enmallada pero no es un rede MANET, está discusión se lleva a cabo en capitulo seis.

5.6.4 DESEMPEÑO EN LA TOPOLOGÍA 5 NODOS CON ACCESO INALÁMBRICO

La idea de esta última topología era observar el comportamiento de una red enmallada en un ambiente real no controlado, como es el tercer piso del edificio de ingenierías físico-mecánicas de la Universidad Industrial de Santander, y comparar los resultados de la

malla respecto a utilizar un único nodo como punto de acceso. Así utilizando la topología descrita en el numeral 5.4.4. Se realizo la prueba como si el nodo dos representará la puerta de enlace con conexión a internet y se estudio la tasa de transferencia desde una estación conectada vía punto de acceso al nodo en su sala y de allí al nodo dos de la malla. Los resultados se presentan en la siguiente tabla:

SALA	Tasa de transferencia (AP) (Kbits)		Tasa de transferencia (MESH) (Kbits)	
	Descarga	Subida	Descarga	Subida
1	8136	3781	3759	3383
2	13817	13782	15337	18205
3	9080	4606	3595	6963
4	12570	6979	3616	2869
5	12386	14586	6443	5957

TABLA 5.5 MESH VS AP

Como se puede observar para distancias pequeñas es decir donde los nodos MESH están todos dentro del rango de cobertura del AP no es útil la red enmallada ya que su tasa de transferencia es menor, es probable que la utilidad de la red enmallada en esta caso se presente para rangos de cobertura superiores al del AP ya que podemos escalar la red fácilmente.

Con esta misma topología se realizaron dos pruebas más, en la primera de ellas la estación de cada sala lanzaba una prueba IPERF de cien segundos, así sucesivamente hasta que hubiese una estación por cada sala con una prueba corriendo hacia el AP del nodo dos, así se comprobó como cae el ancho de banda con cada conexión, el resultado se presenta en la gráfica a continuación que fue tomada por una estación conectada al nodo dos como si fuera el proveedor de internet. Se observa claramente la caída de la tasa de transferencia cuando entra cada nodo.

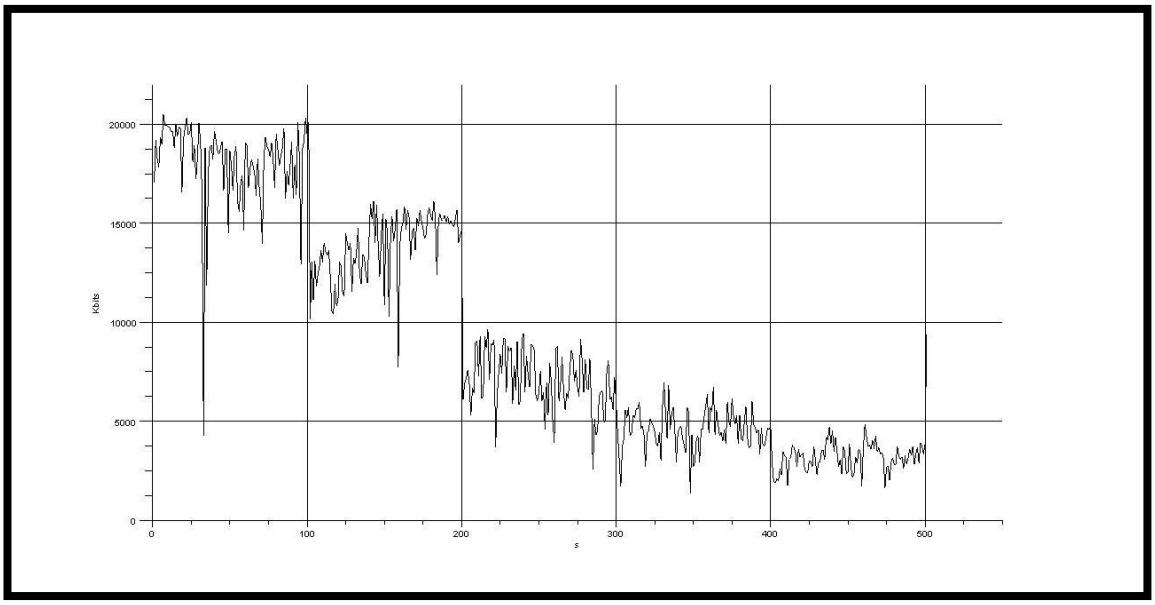


FIGURA 5.37 TASA DE TRANSFERENCIA PUERTA DE ENLACE CON ESTACIONES CONECTADAS A AP

La misma prueba se realizo pero esta vez las estaciones no se conectaban al AP sino al nodo *MESH* de cada sala y como se puede observar en la siguiente figura el desempeño fue muy inferior al presentado conectándose directamente al AP.

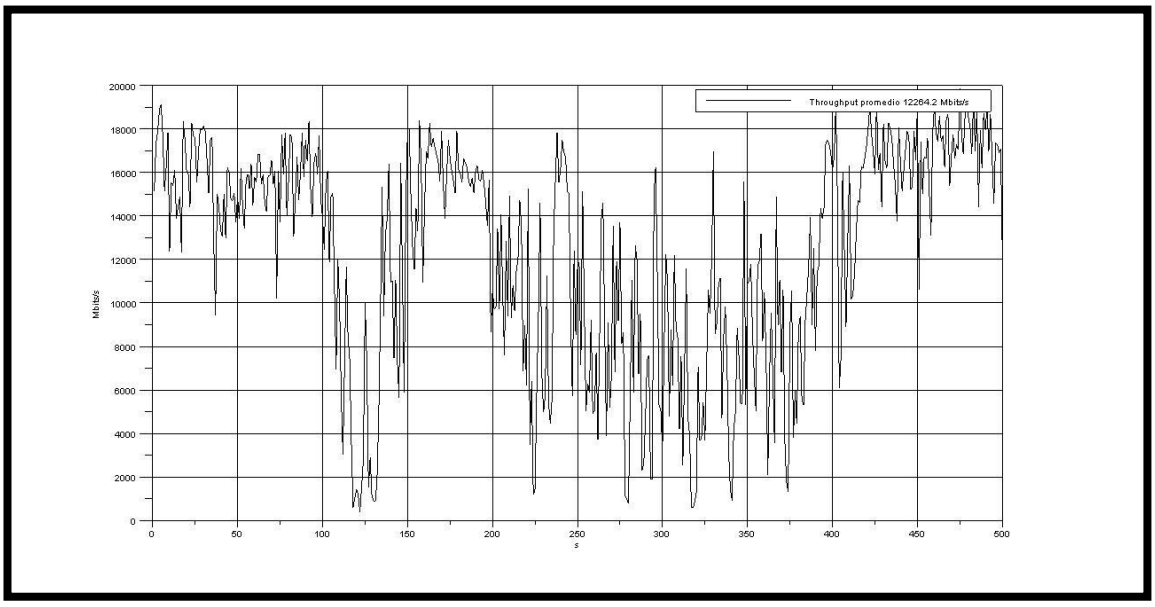


FIGURA 5.38 TASA DE TRANSFERENCIA PUERTA DE ENLACE CON ESTACIONES CONECTADAS A MESH

Una tarea para trabajos futuros puede ser realizar pruebas similares pero para distancias mayores, de esta manera se obtendría una comparación más profunda de si existe ventaja del uso de la red enmallada frente a un único AP para condiciones reales.

6. CAPITULO SEIS – CONCLUSIONES

En este capítulo se describen las conclusiones más importantes de este trabajo de investigación y que pueden servir de referencias para futuros desarrollos tanto en redes inalámbricas enmalladas como protocolos de enrutamiento, el segundo elemento son las recomendaciones de los autores acerca líneas futuras que pueden seguirse en lo tópicos tratados durante el proyecto de grado.

6.1 CONCLUSIONES

- Se determinó que OPENWRT es el firmware de uso libre más completo para construir redes enmalladas sobre los linksys WRT54GL, esto se debe a que es una distribución reducida de Linux para sistemas embebidos que tiene características de robustez y versatilidad, ya que cuenta con repositorios muy completos que posibilitan la instalación de múltiples paquetes orientados a enrutamiento de este tipo de redes, además cuenta con una comunidad de desarrollo y soporte excelente que brinda documentación detallada y actualizada.

- Se diseñó y construyó un prototipo de nodo malla tanto para ambientes *indoor* como *outdoor*, constituido por dos enrutadores Linksys WRT54GL con la posibilidad de alimentarlos por batería o directamente desde la red eléctrica, protegidos dentro de una caja para exteriores. Siendo el costo del nodo aproximadamente entre un 50-70% más económico que los equipos disponibles en el mercado para desplegar redes mesh.

- La herramienta para caracterizar el desempeño de la red seleccionada fue Iperf ya que esta puede ser instalada en equipos con capacidad de memoria reducida como el Linksys WRT54GL, y se encuentra en los repositorios de Openwrt. Adicionalmente se puede complementar con la interfaz gráfica Jperf basada en java para los sistemas operativos Windows y Linux que agiliza el trabajo.

- La tasa de transferencia con el protocolo de transporte UDP debe ajustarse en la herramienta IPERF de tal forma que el valor de ancho de banda sea un poco mayor a la tasa máxima de transferencia de la red ya que si se coloca un número demasiado grande la herramienta no funciona correctamente y un valor por debajo de la tasa máxima de transferencia esperada no satura el enlace.

- Se determinó que el filtrado por MAC es una alternativa para crear bancos de pruebas que permitan caracterizar el desempeño de topologías mesh, debido a que se puede filtrar los mensajes de difusión en los diferentes protocolos de enrutamiento e impedir enlaces entre ciertos nodos, sin embargo no fue efectivo para paquetes multicast.

- Se encontró que OLSR presenta un mejor desempeño para las topologías 3 x 3 y la topología en línea, ya que maneja una tasa de transferencia de aproximadamente 2 [MBits] por encima de BATMAN tanto para TCP como UDP.

- Las características que presenta BATMAN para redes MANET tiene desventajas ante el protocolo OLSR en el despliegue de redes tipo malla estáticas o que cambien poco en su topología. Basados en el estudio de los protocolos que se realizo se puede inferir que BATMAN ofrece excelentes prestaciones en redes tipo malla donde los cambios en la topología son constantes y donde es necesario un cambio de rutas permanente.

- RIP y OSPF se implementaron en los enrutadores Linksys WRT54GL por medio del software quagga disponible para Openwrt, y a pesar de ser los protocolos IGP mas difundidos no realizaron el enrutamiento correctamente porque fueron diseñados para redes cableadas y presentan impedimentos para trabajar en redes inalámbricas Ad-hoc en donde se comparte el mismo medio de transmisión.

- El protocolo de enrutamiento BABEL se implemento en los enrutadores linksys WRT54GL generando correctamente las tablas de rutas, sin embargo la difusión de mensajes es realizada de manera multicast, lo cual fue un impedimento para construir los bancos de pruebas que permitieran caracterizar su desempeño en las diferentes topologías.

- Existe una relación no lineal entre el bit rate y la potencia de emisión en los enrutadores linksys WRT54GL y se observa que para valores por encima de 120 [mW] disminuye significativamente la tasa de transferencia.

- En ambientes *indoor* la interferencia generada por la existencia de otras redes inalámbricas trabajando en el mismo canal de frecuencia en que operan los nodos tipo malla afectó negativamente la tasa de transferencia.

- En un ambiente *outdoor* para distancias menores a 50 [m] con línea de vista se encontró que la mayor tasa de transferencia se halla a una potencia de emisión del radio de 90 [mW**]**.

- El proyecto generó como productos este documento, un prototipo de nodo malla y dos videos de cómo configurar enrutadores linksys WRT54GL para un red enmallada con el protocolo OLSR, estos sirven como guía para proyectos futuros en redes inalámbricas enmalladas, protocolos de enrutamiento o trabajos con enrutadores linksys WRT54GL cumpliendo así el último de los objetivos planteados en el trabajo de grado.

6.2 RECOMENDACIONES

- Se recomienda el uso del generador de tráfico Iperf para caracterizar el desempeño de red en enrutadores Linksys, donde el tamaño de memoria es reducido.

- La herramienta DITG se puede utilizar cuando los recursos de memoria y procesador no incidan en las medidas además cuando se quieran controlar todas las pruebas de manera remota y guardar la información en distintos servidores.

- En caso de problemas durante la instalación de firmware que dejen el dispositivo inasequible por sus interfaces habituales, es útil el uso de un cable de fácil elaboración que se conecta al puerto JTAG ubicado en la placa del linksys para poderle instalar el firmware.

- Se recomienda el uso de disipadores de calor cuando se manejen potencias de emisión por encima de 90 [mW] y en ambientes *outdoor* donde es necesario montarlo dentro de una caja para exteriores que dificulta la refrigeración por lo tanto se evita el deterioro del dispositivo.

- En ambientes *indoor* en donde existen por lo general otras redes inalámbricas se recomienda hacer un estudio previo para seleccionar el canal menos congestionado.

- Si se utiliza frecuentemente la batería como fuente de alimentación se recomienda utilizar un regulador de tensión creando condiciones óptimas para el funcionamiento de los enrutadores.

- Se recomienda para futuros trabajos de redes enmalladas utilizar dispositivos que permitan incorporar más interfaces inalámbricas y donde la capacidad de memoria suficiente para instalar distribuciones Linux que permitan instalar, por ejemplo Voyage, Pebble.

6.3 LÍNEAS FUTURAS

Se resalta que este es el primer trabajo de grado en esta línea de investigación en la Universidad Industrial de Santander, además su componente experimental lo hace interesante ya que se encuentran pocos proyectos que trabajen de esta manera las redes inalámbricas enmalladas, y en la revisión bibliográfica realizada no se encontró en Colombia un precedente. Por estos motivos se espera que este sea el punto de partida para nuevas investigaciones en esta línea, aún queda espacio para probar topologías en ambientes *outdoor* con enrutadores Linksys WRT54GL. Las pruebas se pueden realizar dentro de la Universidad Industrial de Santander con una disposición de los nodos mesh en puntos estratégicos donde acude la mayoría de la comunidad, en este caso el filtrado por MAC no es necesario ya que se reemplaza por la distancia entre nodos que atenúa la señal. Otra alternativa es modificar el hardware del linksys usando antenas direccionales, esto último mejoraría la relación señal a ruido que afecta significativamente la tasa de transferencia.

Existen tres tipos de generaciones en redes inalámbricas enmalladas, la tercera generación utiliza canales diferentes para cada uno de los enlaces de una red inalámbrica tipo malla y para los puntos de acceso inalámbrico, esto no es posible con enrutadores linksys WRT54GL por lo que sería necesario buscar otros dispositivos que permitan el uso de varios radios, con mejor capacidad de procesamiento y almacenamiento, entregando un mayor desempeño pero que a su vez también permitan el uso de software libre.

Otra posibilidad abierta es construir nodos autónomos que se alimenten con paneles solares y permitan implementar topologías en zonas rurales donde no exista conexión eléctrica, de esta manera se pueden plantear proyectos reales donde se construyan redes inalámbricas en zonas rurales o urbanas. Una red inalámbrica enmallada dentro de la universidad sería un buen comienzo como se menciono, y la posibilidad de extenderlo a toda Bucaramanga sería un avance para romper la brecha digital, eso sería seguir el ejemplo de ciudades como parís donde existe una red pública de acceso libre con más de 400 nodos brindando acceso a cualquier persona.

Este proyecto sirve como base para el desarrollo de futuras iniciativas en redes inalámbricas en zonas rurales de difícil acceso a internet ó la creación de redes de datos comunitarias en campus estudiantiles como el de la Universidad Industrial de Santander.

7. REFERENCIAS

1. QUICK GUIDE TO IEEE 802.11 WG & ACTIVITIES. [En línea] Institute of Electrical and Electronics Engineers, Inc. (IEEE), 2009. http://www.ieee802.org/11/QuickGuide_IEEE_802_WG_and_Activities.htm.

2. **Sichitiu, Mihail L.** *Wireless Mesh Networks: Opportunities and Challenges.* España : The tenth IEEE symposium on computers and communications, Junio 2005.

3. **Yarali, Abdulrahman.** *Wireless Mesh Networking Technology for Commercial and Industrial Customers.* s.l. : IEEE.

4. **A.Yarali.** *WiMAX: A Key to Bridging the Digital Divide.* Richmond, VA : IEEE Southeast , 07 .

5. **Corinna "Elektra" Aichele, Rob Flickenger, Carlo Fonda, Jim Forster, Ian Howard, Tomas Krag y Marco Zennaro.** Redes Inalámbricas en los Países en Desarrollo. [En línea] 2007. http://wndw.net/.

6. **Comunicaciones, María del Rosario Guerra. Ministra de.** *COLOMBIA: PLAN DE GOBIERNO EN TICS 2006-2010.* Bogotá D.C. : s.n., Mayo 24 de 2007.

7. **Paul Asadoorian, Larry Pesce.** *Linksys WRT54G Ultimate Hacking.* s.l. : Syngress Publishing, Inc.

8. Proyecto Montevideo libre. [En línea] http://www.montevideolibre.org/.

9. **Osterloh, Heather.** *IP Routing Premier Plus.* Indianapolis : SAMS.

10. **Rubio, Lluis Faixó.** *Redes mesh basadas en puntos de acceso inteligentes 802.11 open source (I).* Universitat Politècnica de Catalunya : s.n., Septiembre de 2005.

11. **Chroboczek, J.** Babel — a loop-free distance-vector routing protocol. [En línea] Mayo 20 de 2010. http://www.pps.jussieu.fr/~jch/software/babel/.

12. **Johnson, David Lloyd.** *Performance Analysis of Mesh Network Indoor and Outdoor Wireless Testbeds.* University of Pretoria : s.n., October 2007.

13. **T.Clausen, P. Jacquet.** The Internet Engineering Task Force (IETF). *Optimized Link State Routing Protocol (OLSR).* [En línea] October 2003. http://www.ietf.org/rfc/rfc3626.txt.

14. **A. Neuman, C. Aichele, M. Linder, S. Wunderlich.** The Internet Engineering Task Force (IETF). *Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.).* [En línea] Apr 07, 2008. http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00.

15. **Cisco.** Linksys by Cisco. [En línea] http://www.linksysbycisco.com/LATAM/es/support/WRT54GL/download.

16. —. Upgrading VxWorks Firmware from the Console. [En línea] http://www.cisco.com/en/US/products/hw/wireless/ps441/products_tech_note09186a00800 93c37.shtml.

17. **Gottschall, Sebastian.** DD-WRT.com. [En línea] http://www.dd-wrt.com/.

18. **Irving Popovetsky, Brandon Psmythe.** Portless Networks. *EWRT.* [En línea] http://www.portless.net/.

19. **Pimps, The Hackers.** Information Security`s Shadow Agency. *FAIRUZA Project.* [En línea] http://www.hackerpimps.com/projects.html.

20. SVEASOFT. [En línea] http://www.sveasoft.com/.

21. Freifunk.net. *Freifunk Firmware (Espanol).* [En línea] http://wiki.freifunk.net/Freifunk_Firmware_(Espanol).

22. OpenWrt Wireless Freedom. [En línea] http://openwrt.org/.

23. **Ishiguro, Kunihiro.** Quagga Routing Suite. [En línea] http://www.quagga.net/.

24. **Holter, Kenneth.** Wireless Extensions to OSPF - Implementation of the Overlapping Relays Proposal. [En línea] 2 de Mayo de 2006. http://folk.uio.no/kenneho/index.php?page=studies&subpage=wospf.

25. **Phillip A. Spagnolo, Thomas Goff, Thomas R. Henderson, Gary Pei.** OSPFv3 MANET MDR. *Boeing Quagga Software.* [En línea] 2008. http://hipserver.mct.phantomworks.org/ietf/ospf/.

26. Mobile Ad-hoc Networks (manet). *IETF.* [En línea] http://datatracker.ietf.org/wg/manet/charter/.

27. List of ad-hoc routing protocols. [En línea] http://en.wikipedia.org/wiki/List_of_ad-hoc_routing_protocols.

28. **Tsai, Wesley.** B.A.T.M.A.N. Daemon HowTo. [En línea] Enero 25, 2009. http://downloads.open-mesh.org/batman/misc/batmand_howto.pdf.

29. **Blum, Richard.** *Network Performance Open Source Toolkit.* s.l. : Wiley Publishing, Inc., August 2003.

30. **Stefano Avallone, Alessio Botta, Alberto Dainotti, Walter de Donato, Antonio Pescapé.** D-ITG, Distributed Internet Traffic Generator. *DITG V. 2.6.1D Manual.* [En línea] Mayo 2, 2008. http://www.grid.unina.it/software/ITG/codice/D-ITG2.6.1d-manual.pdf.

31. Meraki. [En línea] http://meraki.com/.

8. ANEXOS

A. PROFUNDIZACIÓN PROTOCOLOS DE ENRUTAMIENTO.

MODO DE OPERACIÓN Y TIPO DE ANUNCIOS DE ESTADO DE ENLACE

Este protocolo puede correr sobre distintas arquitecturas de red, dependiendo de la arquitectura las tres bases de datos ya mencionadas varían. Además los tipos de anuncios de estado de enlace generados dependen también de la arquitectura de la red en la que se ubiquen los nodos.

TIPOS DE ANUNCIOS DE ESTADO DE ENLACE

OSPF consta de 6 tipos de estado de enlace, cada estado tiene asignado un LSA *(Link State Advertisement)* que se distingue por un respectivo número tal como se presenta en la siguiente tabla.

LSA	Advertencia	Descripción
1	Enrutadores conectados.	Describe los enrutadores directamente conectados a la red y el estado de todas las interfaces.
2	Redes conectadas.	Identifica todos los enrutadores conectados a la red local.
3	Resumen de enlace	Resume todas las subáreas que están por fuera del área.
4	Resumen de enlace	Resume las rutas externas que no pertenezcan al dominio de enrutamiento.
5	AS enlace externo	Describe una ruta para un destino en otro sistema autónomo.
7	AS enlace externo	Lleva información de rutas a través de una NSSA *(Not So Stubby Area)* red de resguardo.

TABLA 8.1 TIPOS ANUNCIOS DEL ESTADO DE ENLACE.

CATEGORÍAS DE TIPOS DE ANUNCIOS:

- Intra-área.
- Inter-área.
- Área externa.

INTRA-ÁREA: Los enrutadores envían mensajes que se propagan únicamente dentro de la misma área de origen. Estos describen los enlaces locales y las redes dentro del área. Corresponden al flujo tipo 1 que es enviado por todos los enrutadores y tipo dos que es enviado únicamente por los enrutadores designados. Los enrutadores envían mensajes LS tipo 1 a la ruta 224.0.0.6 que es la dirección correspondiente al grupo de multidifusión para los enrutadores designados y los enrutadores de soporte designados. Los enrutadores designados por otra parte envían mensajes LS tipo 2 a la dirección 224.0.0.5 a la cual están vinculados todos los enrutadores OSPF, este anuncio identifica todos lo enrutadores.

INTER-ÁREA: Cuando existen varias áreas en un dominio OSPF los enrutadores que conectan estas áreas se denominan enrutadores de borde ABRs por sus siglas en ingles *Area Border Routers.* Normalmente todas las áreas están conectadas a una columna vertebral denominada Área 0 que es la encargada de llevar el tráfico principal que proviene de todas las otras áreas. Los enrutadores de borde envían LSA tipo 3 y 4, los paquetes LSA tipo3 contienen todas las rutas OSPF dentro de una subárea OSPF, los enrutadores de borde informan de estas rutas enviando paquetes dentro del área 0, donde otros enrutadores de borde aprenden y propagan esta nueva información dentro de sus propias áreas. Los mensajes LS tipo 4 para identifican los enrutadores de borde entre sistemas autónomos *ABSR* por sus siglas en ingles *Autonomous System Boundary Routers*, estos enrutadores proveen acceso a enlaces en redes externas a ese sistema autónomo y fuera del dominio de enrutamiento.

ÁREA EXTERNA: Solo los enrutadores *ABSR* envían advertencias externas, estas se propagan a través de todo el sistema autónomo OSPF, excepto en las áreas de resguardo. Estos mensajes describen todas las rutas fuera del dominio de enrutamiento y obviamente del sistema autónomo, como podrían ser redes que manejen RIP, OLSR o cualquier otro protocolo. Únicamente los enrutadores de borde pueden manejar más de un protocolo de enrutamiento, para esto usan mensajes LS tipo 5 con los que informan de rutas fuera del sistema autónomo. Los enrutadores de conectados a un NSSA corren más de un protocolo y son los únicos que envían LS tipo 7, si el área externa se conecta directamente con el área 0 entonces los LS se convierten en 5.

CABECERAS DE LOS ANUNCIOS DE ESTADO DE ENLACE

Todos los LSA constan del mismo formato de cabecera que es de 20 bytes y contiene los siguientes campos.

- **Edad LS:** Contiene un valor de medida en segundos e indica el tiempo que pasa desde que el enrutador de origen envío este LSA.
- **Opciones:** Contiene los mismos valores descritos en los paquetes *hello*.
- **Tipo LS:** Define el tipo de LSA enviado.
- **ID de estado de enlace:** Identifica la dirección IP del enrutador de origen (en el caso de LSA tipo 1 y 4) o la dirección de la red de origen (en el caso de LSA tipo 2, 3, 5 y 7) en la que inicio el anuncio. Los tipos de identidades se presentan en la siguiente Tabla 2
- **Anuncio del enrutador:** Identifica el enrutador que origino el mensaje.
- **Número de secuencia LS:** Garantiza la entrega y recepción de los paquetes, cada vez que se envía una advertencia el enrutador de origen incluye una secuencia numérica que identifica el mensaje. El receptor usa la secuencia numérica para mantener el orden de los paquetes recibidos.
- **Suma de comprobación LS:** Se usa para comprobar si el paquete no se daño durante la transmisión por el canal. El enrutador que envía el mensaje usa un algoritmo para calcular dicho valor, el receptor recalcula el algoritmo. Si los valores no son iguales el paquete es descartado.
- **Longitud:** Contiene el tamaño del datagrama en bytes. Este valor dice que tantos datos se llevan.

Edad LS	Opciones	Tipo LS
ID Estado de enlace		
Anuncio del enrutador		
Número de secuencia LS		
Suma de comprobación LS	Tamaño	

TABLA 8.2 CABECERA DE MENSAJES LSA.

Valor	Descripción
1	ID de un enrutador esclavo
2	Enrutador Designado

3	Dirección IP de la red de donde nació el mensaje
4	ID de enrutadores de Borde de sistema autónomo
5	Dirección IP de la red de donde nació el mensaje

TABLA 8.3 TIPO DE ENRUTADOR.

IDENTIFICACIÓN DE ESTADOS DE ENLACE

ESTADOS DE LOS ENRUTADORES OSPF

Para empezar a intercambiar información de enrutamiento los nodos OSPF deben pasar por los siguientes estados en el respectivo orden que se presenta a continuación.

INICIALIZANDO

Cuando el administrador inicia o reinicia OSPF en alguna interfaz del enrutador, en este estado no existen DR ni BDR, la labor del enrutador en esta fase es darse a conocer a través de mensajes *hello* anunciando su presencia a todos los vecinos para que estos a su vez respondan y pueda crear la respectiva tabla de vecinos.

- **Tratamiento bilateral de los mensajes *hello*:** Cada enrutador al recibir un mensaje *hello* aprende de sus vecinos, sin embargo estos mensajes deben contener mínimo ID Área, *hello* y tiempo de muerte, autenticación e ID de resguardo. Los mensajes *hello* se continúan enviando en intervalos de cada diez segundos para mantener la relación bidireccional entre enrutadores.

POST INICIALIZADO

Después de tener su tabla de vecinos completa los enrutadores tienen suficiente información para elegir el enrutador designado y los enrutadores designados de soporte. En este estado todos los enrutadores establecen una relación maestro esclavo con los enrutadores designados y los enrutadores designados de soporte. Todos los enrutadores son esclavos de los DR y BDR, estos a su vez reciben actualizaciones de enrutamiento de todos los esclavos del segmento.

ENRUTADORES DESIGNADOS:

Generalmente se selecciona un Enrutador designado DR *(Designe router)* y un Enrutador designado de soporte BDR *(Backup Designe router)* por segmento, en cuya selección participan todos los enrutadores del segmento usando dos parámetros que ya habían sido difundidos en los mensajes *hello*. Estos son:

- ID de prioridad.

- ID del enrutador.

El enrutador con ID con prioridad más alta será asignado como DR, el segundo será el BDR, en caso de empate se utilizará la ID del enrutador para conocer cual es el designado. El administrador puede modificar cualquiera de estos dos parámetros lo que le da la capacidad de influir sobre esta decisión e indirectamente genera un parámetro de diseño de red.

Los enrutadores designados deben cumplir las siguientes funciones:

a) Recolectar todos los anuncios de los enrutadores locales.

b) Construir la base de datos de estado de enlace.

c) Difundir esta información a través de todos los enrutadores en el mismo segmento.

Aunque tanto el DR como el DBR reciben la información de enrutamiento de todos los enrutadores a través de la dirección 224.0.0.6 únicamente el DR es el encargado de distribuir la información a todos los enrutadores a través de la dirección 224.0.0.5. El BDR esta en modo de espera hasta que el DR por algún no esté disponible, en este caso el DBR asume el papel de DR, si el DR retorna este no retoma sus funciones, estas continúan realizándose por el BDR.

INTERCAMBIO

Durante este estado todos los enrutadores intercambian información de enrutamiento con el DR y el BDR, la base de datos de estado de enlace completa de toda la red la tiene únicamente del DR. Así cada enrutador construye su mapa de la topología. Sin pasar por este y los estados anteriores un enrutador no puede intercambiar información de enrutamiento. La primera vez que un enrutador entra en este estado tiene que recibir información de toda el área OSPF porque su mapa de topología esta vació, después el enrutador únicamente envía y recibe las actualizaciones con los cambios de la topología. A pesar de esto los enrutadores OSPF envían toda su información de enrutamiento cada 30 minutos solo para estar seguro que todos los enrutadores OSPF tienen su tabla de topología completa.

CARGANDO

El enrutador únicamente entra en este estado cuando estos reciben información conflictiva durante el estado de intercambio. Si la información recibida del DR difiere de la contenida en el mapa de topología el enrutador debe ingresar en estado cargando y enviar peticiones de estado de enlace para recibir información específica del mapa completo.

COMPLETO

Un enrutador alcanza este estado únicamente después de pasar por los otros estados. En el estado completo el enrutador construye su base de datos de direccionamiento e incluye

en ella la mejor ruta a cada destino corriendo el algoritmo SPF en todas las rutas identificadas dentro de la base de datos de estado de enlace.

Después de que un enrutador ha alcanzado el estado completo únicamente puede transitar a través de uno de los siguientes tres estados: Intercambio, cargando o completo.

Los enrutadores atraviesan los tres estados anteriores cuando algún cambio ocurre en su área, por ejemplo la caída o aparición de un nuevo enlace hace que el enrutador genere una actualización de disparo que es procesada por el DR de área y diseminada dentro de la misma, cuando otro enrutador recibe esta información entra en conflicto con la información que ya tenía, entonces entra en el estado cargando, en este estado solicita más información al DR, modifica su mapa de topología, luego corre SPF y de nuevo entra al estado de completo.

TIPOS DE ENRUTADORES OSPF

En un sistema autónomo OSPF un enrutador puede asumir diferentes roles, uno o varios dependiendo de su ubicación en la topología, lo que genera cuatro tipos de enrutadores:

- **Interno:** Enrutador en el que todas sus interfaces están contenidas dentro de una misma área OSPF y este únicamente corre el protocolo OSPF.
- **Columna vertebral *(backbone)*:** Es aquel que tiene por lo menos una interfaz conectada al área 0. Un enrutador que tiene todas las interfaces dentro del área cero funciona como un enrutador interno de la columna vertebral.
- **Enrutador de borde de área:** Esta situado en el borde entre varias áreas OSPF, generalmente conectando una subárea con el área cero en cuyo caso sería también un enrutador de la columna vertebral.
- **Enrutador de borde de sistema autónomo:** Se ubica en el límite de dos sistemas autónomo, este corre OSPF y algún otro protocolo de enrutamiento. Este se puede configurar de manera tal que envié anuncios de rutas que no usan OSPF dentro del dominio de enrutamiento OSPF, diseminando esta información de rutas externas a todas las áreas dentro del sistema autónomo OSPF.

MÚLTIPLES ÁREAS OSPF

OSPF puede implementarse en una sola área cuando las redes son de pequeño o mediano tamaño, sin embargo a medida que el tamaño de la red crece es recomendable dividir el dominio de enrutamiento en múltiples áreas. Una red grande configurada como una única área OSPF generaría una base de datos de estado de enlace muy grande, lo que conlleva a requerimientos de memoria elevados y requerimientos altos de procesamiento al correr el algoritmo SPF. Por otra parte entre más enrutadores más probabilidad de cambios en la red, lo que genera más tráfico y procesamiento. Al dividir en múltiples áreas todos estos problemas se reducen, ya que los enrutadores solo procesan información de las áreas a las que están directamente conectados.

TIPO DE ÁREAS OSPF

Existe tres tipos básicos de áreas OSPF que son las que definen los tipos de anuncios LSA que van a circular por la red y los enrutadores que van a generar los mismos. Estas áreas son:

- Columna vertebral (Área 0)
- Área Estándar.
- Las dos áreas de resguardo. (Reguardo estándar y NSSA – not so stubby area)

▪ **Columna Vertebral (Área 0):** Esta área funciona como la conexión de todas las otras áreas. Dentro de esta se propagan los mensajes generados dentro de su propia área, todos los mensajes de resumen entre áreas enviados por los enrutadores de borde de área y los enviados por los enrutadores de borde de sistemas autónomos. Esta área acepta todo tipo de anuncios excepto los de tipo 7.

▪ **Área Estándar:** Estas áreas funcionan como subáreas del área 0. Acepta LSA tipo 1,2, LSA tipo 3 de resumen de su misma área y LSA tipo 4 de ABR que conectan otras subáreas conectadas con el área 0, también acepta mensajes LSA tipo 5 que provienen obviamente de un enrutador ASBR. Las áreas estándar pueden estar conectadas a la columna vertebral a través de varios caminos para dar robustez a la red.

▪ **Las dos áreas de resguardo:** Un área de resguardo tiene únicamente un camino de entrada y uno de salida, en este caso no se necesita enviar actualizaciones a través de este enlace. Generalmente se utilizan las rutas por defecto o estáticas para eliminar el tráfico de las actualizaciones.

- Área estándar de resguardo
- NSSA (Not So Stubby Area)

El área 0 y ASBRs no puede ser parte de un área de resguardo, además el administrador debe configurar los enrutadores conectados dentro o al área de resguardo o a la red NSSA como enrutador de resguardo.

• **Área Estándar de Resguardo:** A pesar de tener una ruta por defecto esta área recibe los mensajes tipo 1, 2, 3 y 4, pero no acepta ningún anuncio de rutas externas ya sea LSA tipo 5 o 7.

• **NSSA (Not So Stubby Area):** Estas áreas únicamente aceptan LSAs tipo 1 y tipo 2 no acepta LSA tipo 5 de ASBRs, esta área transporta tráfico externo hasta el área 0. Recibe paquetes LSA tipo 7 del ASBRs que son convertidos a tipo 5 en el ABR. Una gran diferencia con el área estándar de resguardo es que en NSSA si se corre un protocolo de enrutamiento.

CAMPOS ESTÁNDAR OSPF

Todos los paquetes OSPF tienen los mismos campos en la cabecera de 24 bytes. La cabecera tiene los siguientes campos.

Versión	Tipo	Tamaño del paquete
ID del enrutador		
ID del Área		
Suma de comprobación		Tipo de Autenticación
Autenticación		
Hello, descripción las bases de datos, Peticiones de estado de enlace, actualizaciones de estado de enlace, acuse de estado de enlace más datos.		

TABLA 8.4 CABECERA DE OSPF

- **Versión:** Identifica la versión OSPF utilizada.
- **Tipo de paquete:** Identifica el tipo de paquete OSPF.
- **Tamaño de paquete:** Identifica el tamaño en bytes del datagrama, incluyendo su cabecera y contenido.
- **ID Enrutador:** Este valor identifica el enrutador del que nació el paquete OSPF. Se puede configurar manualmente o se puede dejar dinámico.
- **ID área:** Identifica el área de la que proviene el datagrama, el área cero siempre tiene la ID 0.0.0.0. Este valor varía por subáreas y puede ser configurado con el mismo número del subárea
- **Suma de comprobación:** Permite verificar si el paquete no se daño durante la trayectoria.
- **Autenticación:** 32 bits que permiten el uso de clave para que los enrutadores verifiquen que el paquete si pertenece a la red y que no se trata de un ataque a la red.
- **Tipo de autenticación:** Este campo determina qué tipo de autenticación maneja el paquete. En OSPF se pueden seleccionar diferentes tipos de autenticación por cada interfaz, así mismo existe la posibilidad de usar diferentes claves, pero todos los enrutadores de un área deben compartir la misma.

Las opciones de este campo se presentan a continuación:

Opción	Descripción
0	Sin autenticación
1	Clave Simple
2	Criptografía simétrica
Otras	Reservado por la IANA para propósitos experimentales.

TABLA 8.5 OPCIONES DE TIPO DE AUTENTICACIÓN.

TIPOS DE PAQUETES OSPF

Tipo	Nombre	Descripción
1	*Hello*	Establece y mantiene relaciones
2	Descripción de Base de datos	Resumen de la base de datos
3	Petición de estado de enlace	Solicita información de enrutamiento
4	Actualización de estado de enlace	Información de enrutamiento en respuesta a las solicitudes
5	Acuse de estado de enlace	Acuses de recibo de información de rutas.

TABLA 8.6 TIPOS DE PAQUETES

PAQUETES HELLO

Los paquetes *hello* permiten crear relaciones y mantenerlas con sus vecinos y son fundamentales para el desarrollo de la tabla de vecinos.

Versión	Tipo	Tamaño del paquete
ID del enrutador		
ID del Área		

Suma de comprobación	Tipo de Autenticación	
Autenticación		
Mascara de red		
Intervalo *hello*	Opciones	Prioridad enrutador
Intervalo de caducidad		
Enrutador Designado		
Enrutador Designado de soporte		
Vecino		
…		

TABLA 8.7 CAMPOS DEL PAQUETE *HELLO*

Como los campos de la cabecera ya fueron descritos a continuación se realiza una descripción de los otros campos que contiene un paquete *hello.*

- **Mascara de red:** Este campo lleva la máscara de subred de la interface.
- **Opciones:** Aquí se especifican las capacidades de OSPF que el enrutador soporta. Aquí se definen dos opciones el bit T y el bit E.
  - ✓ Bit T: Si el bit es diferente de cero indica si el enrutador soporta ToS/QoS.
  - ✓ Bit E: Los enrutadores que tienen este bit en uno tienen la capacidad para procesar información de enrutamiento de protocolos distintos a OSPF, un enrutador de área de resguardo que no procesa información de enrutamiento externan no tiene este bit activado por el contrario todos los ASBR lo deben habilitar.
- **Intervalo *Hello*:** Hace referencia a que tan seguido se van a enviar los mensajes *hello*, en redes de difusión los mensajes *hello* se envían cada 10 segundos. En otro tipo de redes por defecto se hace cada 30 segundos.
- **Prioridad del enrutador:** Este factor es fundamental a la hora de seleccionar los enrutadores designados, el valor por defecto de este depende del fabricante del equipo.

- **Intervalo de caducidad:** Intervalo que permite determinar cuando el enlace con un vecino se asume como caído. Por defecto un enrutador toma un enrutador como muerto después de que lleva cuatro intervalos *hello* sin escucharlo.
- **Enrutador designado:** Este campo contiene la dirección del DR que conoce este enrutador, si no conoce ninguno el valor es 0.0.0.0.
- **Vecinos:** En este campo se escriben las ID de todos los enrutadores que el enrutador conoce a través de sus paquetes *hello* locales.

DESCRIPCIÓN DE PAQUETES DE BASE DE DATOS OSPF

A continuación se presenta una descripción de los campos de un paquete tipo 2.

Versión	Tipo = 2	Tamaño del Paquete
Identidad del enrutador		
ID Área		
Suma de Comprobación	Tipo de autenticación	
Autenticación		
MTU de la interfaz	Opciones	0 0 0 0 0 I M MS
Número de la secuencia		
Cabecera de anuncios de estado de enlace		
…		

TABLA 8.8 PAQUETES TIPO 2

- **I (Init):** Cuando este bit está en 1 indica que este paquete corresponde al primer paquete transmitido de la descripción de la base de datos.
- **M (More):** Cuando este bit está en 1 indica que más paquetes describiendo la base de datos van a llegar.
- **MS (Master/Slave):** El MS o bit Maestro esclavo identifica si el enrutador que transmite es un maestro o un esclavo.
- **Número de secuencia:** Describe la secuencia de todos los paquetes enviados y el recibo de los acuses de cada paquete. El primer valor como en muchos protocolos es aleatorio y se reconoce por el bit I, el inició en un número aleatorio brinda más seguridad y robustez.

- **Cabecera de anuncios de estado de enlace:** Un enrutador puede incluir uno o más anuncios de estado de enlace con un paquete tipo 2. La cabecera de los mensajes LSA ya fue descrita.

PAQUETES DE PETICIÓN DE ESTADO DE ENLACE

Estos paquetes tipo 3 piden información de enrutamiento de algún vecino específico.

Versión	Tipo = 3	Tamaño del Paquete
Identidad del enrutador		
ID Área		
Suma de Comprobación		Tipo de autenticación
Autenticación		
Tipo de LS		
ID Estado de enlace		
Anuncio del enrutador		
…		

TABLA 8.9 PAQUETES TIPO 3

- **Tipo de estado de enlace:** Identifica el tipo de anuncio que se va hacer.
- **ID Estado de enlace:** Es una identificación única asignada a este anuncio en particular y con la cual será reconocido en todos los enrutadores.
- **Anuncio del enrutador:** Este campo identifica al enrutador que generó el anunció.

PAQUETES DE ACTUALIZACIÓN DE ESTADO DE ENLACE

Estos mensajes corresponden a los paquetes tipo 4 y se envían como respuesta a las peticiones. Estos paquetes contienen información de la condición de varios enlaces dentro de la red. Un solo paquetes puede incluir varios anuncios de estado de enlace.

Versión	Tipo = 4	Tamaño del paquete
ID enrutador		
ID Área		

Suma de comprobación	Tipo de autenticación
Autenticación	
Anuncio	
Anuncios de estado de enlace …	

TABLA 8.10 PAQUETES DE ACTUALIZACIÓN DE ESTADO DE ENLACE

El campo asignado a los anuncios del estado de enlace constituyen la mayor del paquete de actualización de estado de enlace, este campo contiene como su nombre lo indica los anuncios del estado de enlace. Cada anuncio LSA tiene la cabecera correspondiente.

PAQUETES DE ACUSE DE ESTADO DE ENLACE

Los paquetes de acuse tipo 5 sirven para informar que se recibió la información de enrutamiento. Su estructura se presenta a continuación.

Versión	Tipo = 5	Tamaño del Paquete
ID Enrutador		
Área ID		
Suma de comprobación		Tipo de autenticación
Autenticación		
Una cabecera de anuncio de estado de enlace		
…		

TABLA 8.11 PAQUETE DE ACUSE DE ACTUALIZACIÓN

B. ESTRUCTURA DE DATOS BABEL

NÚMERO DE SECUENCIA

Es un entero de 16 bits que se incluye en las actualizaciones de rutas enviadas por un nodo. Un nodo incrementa su número de secuencia cada vez que recibe una solicitud para un nuevo número de secuencia.

TABLA DE INTERFACES

Esta tabla contiene la lista de las interfaces en las cuales el nodo utiliza el protocolo. Cada fila de la tabla contiene el número de secuencia de los mensajes *HELLO* enviados por dicha interface, este número se incrementa cada vez que un mensaje *HELLO* es enviado. Además existen dos contadores por cada campo de la tabla, el *HELLO TIMER* que se encarga del periodo de envío de los mensajes *HELLO, IHU (I hear you)* y el *UPDATE TIMER* que se encarga del periodo de las actualizaciones.

TABLA DE VECINOS

Como su nombre lo indica contiene la lista de todas las interfaces vecinas sobre las cuales se ha recibido un paquete babel recientemente. Los campos de la tabla son indexados en parejas *interface-dirección* por lo que la relación no es entre nodos, así un enrutador con múltiples radios puede estar en varios campos de la tabla. Cada vecino en la tabla debe contener los siguientes datos:

- La interface local sobre la que es posible alcanzar el vecino.
- La dirección de enlace local de la interfaz del vecino.
- Un historial del recibo de paquetes *HELLO* de dicho vecino, esté es llevado por una secuencia de n bits que indica cuantos de los n mensajes *HELLO* enviados recientemente por el vecino han sido recibidos por el nodo.
- El valor del costo de la transmisión que estaba en el último paquete *IHU* ó infinito si el tiempo de espera *IHU* expiró.
- El número de secuencia esperado del próximo paquete *HELLO.*

En esta tabla existen dos contadores asociados con cada entrada de vecino, el contador *HELLO TIMER* que se inicia con el valor del intervalo contenido por los mensajes *HELLO* y el contador *IHU TIMER* que se inicializa con un múltiplo del intervalo que contienen los mensajes *IHU*.

DETECCIÓN DE VECINOS

Como en otros protocolos para este fin se utilizan los denominados paquetes *HELLO,* estos contienen el número de secuencia y el intervalo con el que se enviarán de manera periódica, sin embargo es común cuando se descubre un nuevo vecino o cuando las condiciones del enlace cambian de repente que un nodo envié paquetes fuera de este intervalo con el fin de acelerar el proceso para estimar el costo del canal.

Los nodos deben cambiar el *HELLO INTERVAL* que decrece siempre excepto justo antes de enviar un paquete *HELLO.* Cuando un nodo recibe un paquete *HELLO* compara el número de secuencia recibido **“nr”** con el número de

secuencia esperado **"ne"** dependiendo de la comparación se toma una de las siguientes acciones (11):

- Si los dos difieren en más de 2^16 entonces probablemente el nodo emisor se reinicio perdiendo la secuencia por lo que el vecino asociado en la tabla es eliminado.

- Si nr < ne entonces el nodo que envío el paquete incremento su intervalo sin que el receptor lo percibiera, por lo que el nodo receptor elimina las ultimas (ne – nr) entradas del historial *HELLO.*

- Si nr > ne entonces en nodo emisor disminuye su *HELLO INTERVAL,* por lo tanto algunos paquetes *HELLO* se perdieron, el nodo receptor añade 0 bits al historial *HELLO.*

  El nodo receptor añade un bit al historial *HELLO,* reiniciando el *HELLO TIMER* de vecinos y asignando a **ne** el valor **nr + 1**. Entonces coloca el *HELLO TIMER* en 1.5 veces el valor anunciado en el mensaje *HELLO* y el margen extra de retardo debido al Jitter de mensajes.

  Cuando el *HELLO TIMER* asociado a un vecino expira, se añade un 0 al historial *HELLO* e incrementa el número *HELLO* esperado. Si el historial de mensajes ya tenía un 0 en el historial entonces el vecino se elimina. De otra manera se reajusta el *HELLO TIMER* al valor anunciado en el último mensaje recibido de su vecino.

DETECCIÓN DE BIDIRECCIONALIDAD

Para comprobar la bidireccionalidad de un enlace los nodos envían periódicamente a sus vecinos mensajes IHU. Estos mensajes contienen su propio intervalo y son enviados con menos frecuencia que los *HELLO.* Aunque conceptualmente son unidifusión se deben enviar como multidifusión para agregar varios paquetes en un solo mensaje. Los mensajes *IHU* contienen dos partes, el ***rxcost*** y el intervalo entre *IHU*. Un nodo que recibe un *IHU* actualiza el ***txcost*** por el valor contenido en el mensaje y reajusta el *IHU TIMER* por un múltiplo del contenido en el mensaje. Si el *IHU TIMER* expira entonces el ***txcost*** se coloca en infinito.

Después de actualizar estos valores el nodo recalcula el costo de los vecinos y corre el proceso de la selección de rutas.

COSTO DE ENLACE

Los costos de enlace asocia a un vecino son computados de los valores en la tabla de vecinos, denominados historial de *HELLO* y su ***txcost.***

Por cada vecino el nodo computa los valores de costo de recepción ***rxcost.*** Este valor es se deriva del historial *HELLO*, que se combina con otras estadísticas del canal. Tanto ***txcost*** como el ***rxcost*** son utilizadas para determinar el costo del canal, sin embargo esto no está definido por BABEL y corresponderán a las políticas implementadas por el diseñador de red, pero debe satisfacer las siguientes condiciones:

- El costo es estrictamente positivo.
- Si el historial *HELLO* esta iniciado (0 bits), entonces el costo es infinito.
- Si el ***txcost*** es infinito, entonces el costo es infinito.

LA TABLA FUENTE

Esta se indexa por tripletes (prefijo, pleno, vecino), cada entrada de la tabla contiene:

- EL prefijo.
- La dirección del enrutador que origino este prefijo.
- La pareja número de secuencia y métrica, conocido como distancia de referencia a dicha fuente.

  Existe un contador asociado con cada entrada de la tabla, el *SOURCE GARBAGE COLLECTION TIMER* que es inicializado en un tiempo del orden de los minutos y reiniciado

LA TABLA RUTAS

Los elementos de dicha tabla se indexan a través de tripletes (prefijo, pleno, vecino), cada elemento de la tabla debe contener los siguientes datos:

- El anuncio del prefijo.
- El vecino que anuncia dicha ruta.
- La métrica con la que esta ruta fue anunciada por su vecino.
- El número de secuencia con la que la ruta fue anunciada.
- El siguiente salto de dicha ruta
- Una bandera que indica si esta ruta es seleccionada.

ADQUISICIÓN DE RUTAS

Cuando se recibe una actualización de un vecino primero se mira si este ya está en tabla de rutas, si no está:

- Si la actualización es inviable, esta es ignorada.
- Si la métrica es infinita, la actualización es ignorada.
- Si lo anterior no sucedió se crea una nueva entrada en la tabla, con un número de secuencia y una métrica igual a la métrica del la actualización.

Si la entrada ya existe entonces:

- Si la actualización es inviable, entonces depende de si los *router-ids* concuerdan, si no lo hacen la actualización es tratada como si su métrica fuera infinita, por otra parte si son diferentes la actualización es ignorada.
- Si la actualización es viable, entonces el número de secuencia, la métrica de referencia y la métrica son actualizadas, a menos que la métrica sea infinita el contador del tiempo para expirar es reiniciado a un múltiplo del intervalo incluido en la actualización.

  Cuando el contador de expiración se dispara, todo depende de la métrica, si esta es finita está se coloca en infinito y el contador de expiración se reinicia. Si esta es infinita, la ruta es sacada de la tabla.

SELECCIÓN DE RUTAS

BABEL fue diseñado para permitir flexibilidad en las políticas de selección de rutas, imponiendo únicamente dos condiciones:

- Una ruta con métrica infinita nunca es seleccionada.
- Una ruta inviable nunca es seleccionada.

Sin embargo estas dos condiciones no aseguran un enrutamiento estable por lo que a continuación se presentan algunas recomendaciones para una política correcta:

- Rutas con una métrica pequeña deben ser preferidas sobre rutas con una métrica mayor.
- El cambio de *router-ids* debe ser evitado.
- Rutas sobre vecinos estables deben preferirse sobre rutas sobre vecinos inestables.
- Rutas estables deben ser preferidas sobre inestables.
- Cambios en el siguiente salto deben ser evitados.

TIEMPO DE ESPERA

Cuando un prefijo es retractado porque todas las rutas son inviables, muy viejas o tiene métrica infinita y otro prefijo más corto que cubre al primero está disponible, no se puede utilizar esta ruta alternativa para enviar paquetes destinados a la primera sin correr el riesgo de crear un bucle. Para eliminar este problema se mantiene el campo en la tabla de rutas con una métrica infinita y los paquetes con este destino no son redireccionados a través rutas de prefijos menores. La métrica se mantiene en infinito hasta que una actualización haga la ruta disponible, si esta no llega antes del tiempo de espera entonces la ruta inicial se elimina de la tabla.

LA TABLA DE PETICIONES PENDIENTES.

Esta tabla contiene una lista de número de secuencia de peticiones que el nodo ha enviado y que no han sido respondidas aún, cada fila contiene los siguientes datos:

- El *router-id* y el número de secuencia empieza peticiones.
- Los vecinos a los que se ha reenviado la petición.
- Un entero indicando el número de veces que la petición ha sido reenviada sin obtener respuesta.

Hay un contador asociado con cada petición pendiente, que gobierna tanto el reenvío de peticiones como su tiempo de expiración.

PETICIONES DE RUTAS EXPLICITAS

Existen dos tipos de peticiones explicitas, petición de ruta que ocurre cuando se pide la actualización de prefijo y la actualización de el número de secuencia. Cada petición de actualización se hace a un prefijo específico con un número de secuencia.

Si un nodo recibe una petición explicita entonces este responde por la misma interfaz por la cual recibió la petición con una actualización de la información requerida, si no tiene la información para hacerlo entonces redirecciona la petición. Además guarda que peticiones han sido redireccionadas para no causar redundancia de peticiones.

En caso de que una ruta sea retractada o expire y no exista otra ruta disponible para el mismo prefijo entonces el nodo debe enviar una petición de numero de secuencia con el *Router Id* de la ruta que perdió y el número de secuencia contenido en su tabla fuente más uno, esta se envía por todas sus interfaces.

PAQUETES DE ACUSE

Los nodos BABEL tienen la capacidad de hacer peticiones con acusé, en las que cualquier vecino que reciba el paquete debe enviar un acusé dentro de un tiempo

determinado, aunque el uso de este tipo de peticiones es opcional en caso de que un nodo reciba una de este tipo es obligatorio contestarla.

Los ACK son enviados en unidifusión, mientras que las peticiones pueden ser enviadas en unidifusión ó multidifusión.

Si bien esta característica está presente no es muy recomendado utilizarla ya que puede generar demasiado tráfico, en vez de esto se pueden usar actualizaciones periódicas que aseguren que las nuevas rutas son propagadas por la red, otra posibilidad es usar estas peticiones de ACK solo con peticiones urgentes.

HORIZONTE DE PARTIDA

Si bien esta opción está disponible solo es útil en redes simétricas y tecnologías cableadas, ya que la técnica no es adecuada para tecnologías inalámbricas descentralizadas (la interface casi siempre es la inalámbrica por lo tanto no se puede dejar de enviar por la misma interface que se recibió) como en el caso de las redes enmalladas.

ENVÍO DE ACTUALIZACIONES

Los nodos BABEL anuncian a sus vecinos las rutas seleccionadas a través de paquetes de actualización en unidifusión o multidifusión difundiéndolos por todas las interfaces, sin embargo la tecnología multidifusión es significativamente más costosa que la unidifusión, así que un nodo debería preferir el envío de múltiples copias de la actualización en unidifusión cuando el número de vecinos es pequeño.

Adicionalmente, para asegurar que no existan agujeros negros babel envía actualizaciones con métrica infinita para prefijos eliminados recientemente.

ACTUALIZACIONES POR DISPARO

Cuando ocurre un cambio significativo en la topología de la red BABEL envía una actualización.

Un cambio en el *router-id* de un prefijo debe ser anunciado, la retracción de una ruta también es motivo para enviar una actualización y por último cuando existe un cambio significativo en la métrica de un prefijo el nodo debe anunciarla por medio de una actualización.

ACTUALIZACIONES PERIÓDICAS

Cada nodo BABEL anuncia periódicamente sus rutas por todas las interfaces incluidas aquellas recientemente retractadas, como BABEL usa actualizaciones por disparo y además se sabe que no sufre de bucles entonces no es necesario hacer actualizaciones periódicas muy seguido brindando la posibilidad de seleccionar un periodo largo.

ACTUALIZACIONES INVIABLES

Cuando se recibe una actualización inviable de una ruta actualmente seleccionada el nodo envía una petición de número de secuencia, el mismo proceso se hace si se recibe una actualización inviable de un vecino no seleccionado pero que podría anunciar rutas si fuera seleccionado.

PREVENCIÓN DE EXPIRACIÓN DE RUTAS

En caso de que una ruta esté cerca de su tiempo de expiración, un nodo BABEL debe enviar una petición unidifusión al vecino que anuncio la ruta, el resultado casi siempre es el refresco de la ruta o es retractada.

CODIFICACIÓN DEL PROTOCOLO

Los paquetes BABEL son enviados como el cuerpo de un paquete UDP**,** con conteo de saltos igual a uno, con destinos *unicast* y *multicast,* puede ir sobre IPv4 o IPv6. Para minimizar el número de paquetes y evitar la fragmentación en capas inferiores, un nodo babel debe intentar maximizar el tamaño de los paquetes enviados, y un MTU de a las interfaces de salida ajustado a las cabeceras de las capas inferiores. Sin embargo, BABEL puede recibir paquetes mayores.

FORMATO PAQUETE

Un paquete babel consiste en una cabecera de cuatro octetos seguida por una secuencia de mensajes BABEL.

1 OCTETO	2 OCTETO	3 OCTETO	4 OCTETO	…..
MAGIC	VERSION	TAMAÑO DEL CUERPO		CUERPO

TABLA 8.12 FORMATO PAQUETE

- **MAGIC:** Corresponde al número 42, aquellos paquetes que contengan un número diferente deben ser ignorados.
- **Versión:** Este documento especifica la versión 2 del protocolo BABEL. Paquetes con un octeto diferente deben ser ignorados.
- **Tamaño del cuerpo:** El tamaño en octetos del cuerpo.
- **Cuerpo:** El cuerpo del paquete hace referencia a una secuencia de mensajes.

FORMATO DEL MENSAJE

A excepción de Pad1, todos los mensajes siguen la siguiente estructura

Primer octeto	Segundo octeto	……
Tipo	Tamaño	Cuerpo …

TABLA 8.13 FORMATO MENSAJE.

- **Tipo:** Especifica el tipo de mensaje.
- **Tamaño:** El tamaño del cuerpo se determina acá, si el tamaño es mayor que el esperado en un determinado tipo de mensaje entonces este es ignorado.
- **Cuerpo:** En este campo va el cuerpo del mensaje, su interpretación depende del tipo de mensaje.

TIPO PAD1

Si el tipo de mensaje es 0 indica un mensaje Pad1. Este mensaje es ignorado en la recepción.

TIPO PADN

Primer octeto	Segundo octeto	……
Tipo	Tamaño	MBZ…

TABLA 8.14 FORMATO PADN.

- **Tipo:** Si está en uno indica un mensaje PadN.
- **MBZ:** Este campo se coloca en cero en la transmisión, este mensaje es ignorado en la recepción.

MENSAJE DE PETICIÓN DE ACUSE

Primer Octeto	Segundo Octeto	Tercer Octeto
Tipo = 2	Tamaño	Reservado
Nonce		Intervalo

TABLA 8.15 PETICIÓN DE ACUSE.

- **Tipo:** Si esta en dos indica que es un mensaje de petición de acuse.
- **Reservado:** Este debe estar en cero si no el paquete será ignorado.
- **Nonce:** Este campo es un valor arbitrario que vendrá en el acuse de respuesta.
- **Intervalo:** Este campo expresa un intervalo de tiempo en centisegundos después del cual el nodo que envió el paquete puede asumirlo como perdido.

MENSAJE DE ACUSE

Primer octeto	Segundo octeto	……
Tipo = 3	Tamaño	Nonce…

TABLA 8.16 MENSAJE DE ACUSE.

- **Tipo:** Si el valor es tres esto indica que es un mensaje de acuse.
- **Nonce:** Este debe ser el mismo valor que venía en el mismo campo de la petición de acuse. Como este valor no es globalmente único, estos mensajes deben ser enviados por unidifusión.

MENSAJE *HELLO*

Primer octeto	Segundo octeto	Tercer octeto
Tipo = 4	Tamaño	RESERVADO
Número de secuencia		Intervalo

TABLA 8.17 MENSAJE HELLO.

- **Tipo:** Si este campo contiene un 4 indica que es un mensaje ***HELLO.***
- **Reservado:** Este campo debe estar en cero y ser ignorado por el receptor.
- **Número de Secuencia:** El número de secuencia para mensajes *hello* del emisor por dicha interface.

- **Intervalo:** El límite máximo expresado en centisegundos en el cual el nodo emisor va enviar un nuevo mensaje *HELLO,* este campo no puede ser cero.

Mensaje *IHU:*

IHU significa "I Heard you" este mensaje es utilizado para confirmar que el canal es bidireccional.

Primer octeto		Segundo octeto	Tercer octeto
Tipo = 5	Tamaño	AE	Reservado
Costo de transmisión		Intervalo	
Dirección . . .			

TABLA 8.18 MENSAJE IHU.

- **Tipo:** Un mensaje tipo 5 indica que es un mensaje *IHU.*
- **AE:** Indica la forma de codificación del campo de dirección, este debe estar entre 1 y 3 en la mayoría de los casos. Como una optimización este campo debe ser cero si el mensaje es enviado en unidifusión, si el mensaje es asociado sobre conexiones punto a punto o cuando se comprueba un enlace bidireccional con un punto fuera del dominio de enrutamiento.
- **Reservado:** Este campo debe ser cero e ignorado en la recepción.
- **Costo de transmisión:** El costo de transmisión acorde con el nodo emisor de la interface cuya dirección es especificada en el campo Dirección. El valor 0xFFFF en Hexadecimal indica que esta interface no está disponible.
- **Intervalo:** Es el límite expresado en centisegundos después del cual el emisor puede enviar nuevos mensajes IHU; este no puede ser cero. El nodo receptor usara dicho valor para calcular el tiempo de espera para esta asociación simétrica.
- **Dirección:** La dirección de nodo de destino, en el formato especificado en el campo AE.

Conceptualmente los mensajes *IHU* están tienen un solo destino. Sin embargo estos pueden contener direcciones multidifusión ya se agregan varios mensajes *IHU* en un solo paquete.

Mensaje *Router-ID*

Primer octeto	Segundo octeto	Tercer octeto
Tipo = 6	Tamaño	RESERVADO
Router – Id		

TABLA 8.19 MENSAJE ROUTER-ID

Este mensaje establece el Router Id utilizado en los próximos mensajes de actualización.

- **Tipo:** Si este valor es 6 indica que es un mensaje de *Router-ID.*
- **Reservado:** Si este campo es cero el mensaje debe ser ignorado.

MENSAJE DE PRÓXIMO SALTO

El mensaje de próximo salto establece el siguiente salto en una familia de direcciones que está implícita en la subsecuencia de mensajes de actualización

Primer octeto	Segundo octeto	Tercer octeto	
Tipo = 7	Tamaño	AE	Reservado
Próximo Salto . . .			

TABLA 8.20 MENSAJE DE PRÓXIMO SALTO.

- **Tipo:** Si está en siete indica el que es un mensaje de próximo salto.
- **AE:** Este codifica el campo dirección, debe estar entre 1 y 3 pero no pude ser cero.
- **Reservado:** Este campo debe ser cero y debe ser ignorado por el receptor.
- **Próximo salto:** El próximo salto anunciado por los mensajes de actualización.

Si el campo AE no es apropiado el mensaje se ignora.

MENSAJE DE ACTUALIZACIÓN

Un mensaje de actualización anuncia o elimina una ruta. Como una optimización, este puede también el efecto de establecer una nuevo *router-id.*

Primer octeto	Segundo octeto	Tercer octeto	
Tipo = 8	Tamaño	AE	Banderas
Plen	Omitido	Intervalo	
Número de secuencia		Métrica	
Prefijo . . .			

TABLA 8.21 MENSAJE DE ACTUALIZACIÓN.

- **Tipo:** Si tiene un 8 es un mensaje de actualización.
- **AE:** Un tipo de codificación del prefijo, si esta en cero la métrica debe ser 0xFFFF, en cuyo caso este mensaje se retrae en todas las rutas previamente anunciadas por el emisor en esta interface.
- **Banderas:** Los bits de este campo especifican un manejo especial del mensaje. Cada nodo debe ser capaz de interpretar las banderas 0x80 y 0x40; una bandera desconocida tiene que ser ignorada.
- **Plen:** Este campo anuncia el tamaño del prefijo.
- **Omitido:** Es el número de octetos que van a ser omitidos al inicio del prefijo, y estos deben ser deben ser tomados de un mensaje de actualización anterior con la bandera en 0x80.
- **Intervalo:** Indica el límite expresado en centisegundos, después de este tiempo el emisor envía una nueva actualización para este prefijo. Este campo no debe ser cero y debe ser menor que diez. El nodo receptor utiliza este valor para calcular el tiempo de espera en la entrada de su tabla de rutas. El valor 0xFFFF (infinito) expresa que este anuncio no debe repetirse hasta que no se reciba una petición.
- **Número de secuencia:** Corresponde al número de secuencia originado por esta actualización.
- **Métrica:** La métrica del emisor de esta ruta. El valor 0xFFFF (infinito) significa que esta ruta es una ruta retraída.
- **Prefijo:** Este especifica el prefijo que había sido anunciado.

➢ Si la bandera 0x80 está en uno entonces este mensaje establece un nuevo prefijo para los siguientes mensajes de actualización dentro del mismo paquete.

- Si el bit 0x40 está en uno, entonces los octetos de menor orden del prefijo establecen un nuevo *router-id* para este mensaje y los siguientes en el mismo paquete.

MENSAJE DE PETICIÓN DE RUTA

Un mensaje de petición de rutas solicita al receptor el envió de la ruta a un prefijo determinado o toda la tabla de rutas.

Primer octeto	Segundo octeto	Tercer octeto	
Tipo = 9	Tamaño	AE	Reservado
Prefijo . . .			

TABLA 8.22 MENSAJE DE PETICIÓN DE RUTA.

- **Tipo:** Si este campo es un nueve indica que es un mensaje de petición de ruta.
- **AE:** Explica la codificación del prefijo. El valor 0 especifica que es una petición de toda la tabla de rutas.
- **Plen:** Esta es el tamaño del prefijo solicitado.
- **Prefijo:** Este campo especifica el prefijo solicitado.

PETICIÓN DE NÚMERO DE SECUENCIA

Un mensaje de este tipo pide al receptor el envío de una actualización de un prefijo con un número de secuencia determinado o redireccionar la petición si esta no se puede satisfacer localmente.

Primer octeto	Segundo octeto	Tercer octeto	
Tipo = 10	Tamaño	AE	Banderas
Número Secuencia		Conteo de saltos	Reservado
Router-ID			
Prefijo . . .			

TABLA 8.23 PETICIÓN DE NÚMERO DE SECUENCIA.

- **Tipo:** Si este en 10 indica que es el número de secuencia.
- **AE:** La codificación del prefijo es estipulada en este campo. Este no debe ser cero.
- **Plen:** Aquí se especifica el tamaño del prefijo.
- **Número de secuencia:** El número de secuencia que se está solicitando.
- **Conteo de saltos:** El máximo número de veces que este mensaje debe ser redireccionado más uno. Esté número no debe ser cero.
- **Prefijo:** Este campo especifica el prefijo del cual se solicita el número de secuencia.

C. TABLAS RESULTADOS DE PRUEBAS OLSR

1 SALTO	TCP[Kbits]	UDP[Kbits]	JITTER[ms]	RETRASO[ms]
1-2	14415	19272	1.343	2
1-4	15560	18651	1.363	2
2-1	15023	15926	4.845	2
2-3	13935	18919	16.415	2
2-5	16457	19215	1.502	2
3-2	15418	19372	1.419	2
3-6	16182	18560	2.261	2
4-1	15766	19087	2.869	2
4-5	16627	18530	1.253	2
4-7	16542	18502	1.162	2
5-2	16686	19215	1.502	2
5-4	16444	21680	1.401	2
5-6	17078	18270	1.846	2
5-8	16797	18531	1.162	2
6-3	15737	20588	1.371	2
6-5	24189	21049	2.047	2
6-9	16235	18158	1.862	2
7-4	16254	21616	1.551	2
7-8	16188	17829	14.586	2
8-5	15980	15980	1.324	2
8-7	15706	20658	1.665	2
8-9	15933	18233	2.311	2
9-6	16110	20811	1.423	2
9-8	15196	21456	2.203	2
PROMEDIOS	16269,0833	19171,1667	2.945	2

TABLA 8.24 RESULTADOS MÉTRICAS DE NODOS A UN SALTO.

2 SALTOS	TCP[Kbits]	UDP[Kbits]	JITTER[ms]	RETRASO[ms]
1-3	6834	10781	2.980	3
1-5	7939	10131	2.383	4
1-7	8226	10429	13.190	3
2-4	8604	11297	1.840	3
2-6	8670	11405	4.226	3
2-8	8618	11158	15.418	3
3-1	7045	9289	5.161	3
3-5	8271	10821	14.915	3
3-9	8135	11190	1.877	3
4-2	8173	11696	1.947	3
4-6	8794	12327	3.017	3
4-8	8558	11699	2.176	3
5-1	8036	11185	3.706	3
5-3	8225	10273	2.338	3
5-7	8624	11689	1.452	3
5-9	8369	11689	3.136	3
6-2	8533	11405	4.226	4
6-4	8500	11402	2.933	3
6-8	8559	11365	2.487	3
7-1	8043	11199	2.560	3
7-5	8421	11648	1.458	3
7-9	8336	11610	2.416	4
8-2	8519	11808	2.087	3
8-4	8461	11599	2.176	3
8-6	8447	11738	1.544	3
9-3	8121	10358	2.318	3
9-5	8323	11861	2.066	3
9-7	7925	11325	2.165	3
PROMEDIO	8261,03571	11227,75	3.864	3,10714286

TABLA 8.25 RESULTADOS MÉTRICAS NODOS A DOS SALTOS

3 SALTOS	TCP[Kbits]	UDP[Kbits]	JITTER[ms]	RETRASO[ms]
1-6	5462	7013	2.905	4
1-8	5555	7122	4.011	4
2-7	5651	7556	1.094	4
2-9	5778	7622	3.861	6
3-4	5621	7200	2.433	4
3-8	5307	7625	3.267	5
4-3	5527	7200	2.433	4
4-9	5784	7696	2.621	4
6-1	5395	7072	2.922	4

6-7	5854	7661	3.630	4
7-2	5658	7790	3.352	4
7-6	5788	7755	3.133	4
8-1	5288	7611	4.300	4
8-3	5547	6482	2.766	4
9-2	5821	7748	2.291	4
9-4	5802	7718	2.433	4
PROMEDIO	5614,875	7429,4375	2.966	4,1875

TABLA 8.26 RESULTADOS MÉTRICAS DE NODOS A TRES SALTOS

4 SALTOS	TCP[Kbits]	UDP[Kbits]	JITTER[ms]	RETRASO[ms]
1-9	4227	5443	2.591	6
3-7	4279	5619	6.339	5
7-3	4273	5427	2.328	5
9-1	4149	5477	7.489	6
PROMEDIO	4232	5491,5	4.687	5,5

TABLA 8.27 RESULTADOS MÉTRICAS DE NODOS A CUATRO SALTOS