ANÁLISIS Y PRUEBAS DE DESEMPEÑO EN DISPOSITIVOS AP MEDIANTE IMPLEMENTACIÓN DE UN FIRMWARE BASADO EN EL KERNEL DE LINUX

Universidad del Cauca

ALBERTO FERNANDO RODRÍGUEZ PABÓN
WILLIAM LEANDRO MORENO CASTRO

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Popayán
2007

ANÁLISIS Y PRUEBAS DE DESEMPEÑO EN DISPOSITIVOS AP MEDIANTE IMPLEMENTACIÓN DE UN FIRMWARE BASADO EN EL KERNEL DE LINUX

ALBERTO FERNANDO RODRÍGUEZ PABÓN
WILLIAM LEANDRO MORENO CASTRO

Universidad del Cauca

Trabajo de Grado para optar al título de
Ingeniero en Electrónica y Telecomunicaciones

Director:
Ing. Esp. Guefry Agredo Méndez

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Popayán
2007

A quienes con su trabajo hicieron que pudiera salir adelante, que pudiera tener un hogar, mis mejores amigos, mis maestros y ejemplos a seguir, mi padre Alberto y mi madre Amparo

A mis hermanos por todo su apoyo, afecto, entendimiento y buenos sentimientos.

A Todos aquellos amigos que siempre estuvieron ahí, ayudándome y apoyándome en mi trabajo.

A todos aquellos que confiaron en mí.

Alberto Fernando Rodríguez Pabón

A todos los que creen en mí y para las mujeres que más amo…
Mi mamá Elvia, Eddy, Alexandra y Catalina.

William Leandro Moreno Castro

AGRADECIMIENTOS

Quienes desarrollaron este documento agradecen a todos quienes de una u otra forma hicieron posible, la realización del mismo.

A Nuestro creador por darnos la vida y guiarnos a lo largo de este camino, por apoyarnos en cada meta que emprendíamos.

A Guefry Leider Agredo Méndez, por su estrategia, apoyo, paciencia, guía y entrega para la elaboración de este trabajo.

A todos los Ingenieros, Profesores y guías que nos enseñaron durante este proceso de pre-grado.

A nuestras familias y a todas aquellas personas que aportaron para hacer posible el cumplimiento de esta meta.

A todos y cada uno de Ustedes

Muchas Gracias…

Alberto y William

CONTENIDO

LISTA DE FIGURAS

pag.

LISTA DE TABLAS

pag.

LISTA DE ACRONIMOS

AC	Access Category (Categoria de Acceso).
ADC	Analog-to-Digital Converter (Convertidor Analogico Digital).
AES	Advanced Encryption Standard (Algoritmo Estándar de Cifrado Avanzado).
AU	Access Unit (Unidad de Acceso, de la empresa Alvarion).
BSR	Base Station Radio (Estación Radio Base).
BSS	Basic Service Set (Conjunto Básico de servicios).
CFP	Content Free Period (Periodo Libre de Contención).
CFRate	Content Free Rate (Tasa de Periodos Libres de Contienda).
CISC	Complex Instruction Set Computer (Set de Instrucciones Complejas para Procesador).
CPE	Costumer Premise Equipment (Equipo de Premisa de Usuario).
CSMA/CA	Carrier Sense Multiple Access (Método de Aacceso al Medio por Testeo de Portadora con Evasión de Colisiones).
CTS	Clear To Send (Libre Para Enviar).
DAR	Digital Audio Editing And Recording (Grabador y Eitor de Audio Digital).
DBPSK	Differential Binary Phase Shift Keying (Modulación por Desplazamiento de Fase Binaria Diferencial).
DCF	Distributed Coordination Function (Función de Coordinación Distribuida).
DMT	Discreet Multitone Modulation (Modulación Discreta por Multi-tono).
DQPSK	Differential Quadrature Phase Shift Keying (Modulación por Desplazamiento de Fase en Cuadratura Diferencial).
DSL	Digital Subscriber Line (Linea Digital de Subscriptor, tecnología de acceso de ultima milla).
DSSS	Direct Sequence Spread Spectrum (Tecnología de Espectro Ensanchado por Secuencia Directa).
DVR	Digital Video Recorder (Grabadoras de Video Digital).
EAP	Extensible Authentication Protocol (Protocolo de Autenticación Extensible).

EE	Espectral Efficiency (Eficiencia Espectral).
EPROM	Erasable Programmable Read Only Memory (ROM Borrable Programmable).
ETRI	Electronics and Telecommunications Research Institute (Instituto de Investigaciones de Electrónica y Telecomunicaciones).
FDM	Frequency Division Multiplex (Multiplexación por División de Frecuencia).
FHSS	Frequency Hopping Spread Spectrum (Tecnología de Espectro Ensanchado por Salto de Frecuencias).
FSF	Free Software Foundation (Fundación Para el Software Libre).
FSK	Frequency Shift Keying (Modulación por Desplazamiento de Frecuencia).
GNU	GNU No es Unix (Sistema Operativo de Código Abierto).
GPL	General Public License (Licencia General Pública de GNU).
HD-DVD	High Definition-DVD (Reproductores DVD de Alta Definición).
ICV	Integrity Check Value (Valor de Chequeo de Integridad).
IDR	Indoor Data Radio (Equipo Para el Subscriptor Interno, de la empresa Airspan).
IDU	Indoor Unit (Unidad Indoor, de la empresa Alvarion).
IFS	Inter Frame Spacing (Espaciado Entre Tramas).
IOS	Internetwork Operating System (Sistema Operativo de Cisco).
IV	Initialization Vector (Vector de Inicializacion).
LAN	Local Area Network (Red de Área Local).
MAC	Media Control Access (Control de Acceso al Medio).
MACA	Multiple Access with Collision Avoidance (Método de Acceso con Evasión de Colisión).
MIB	Management Information Base (Estructura Base de Información de Administración).
MIC	Message Integrity Code (Códigos de Integridad de Mensaje).
MIMO	Multiple Input Multiple Output (Entrada Múltiple Salida Múltiple).
MPDU	MAC Protocol Data Unit (Unidades de Datos MAC).
NAT	Network Address Traslation (Traducción de Direcciones de Red).
NAV	Network Allocation Vector (Vector de Asignación de Red).

Ndbps	Número de Bits de Datos Por Funcionamiento del Símbolo.
OSI	Open System Interconnection (Modelo de Interconexión Abierto).
PAPR	Peak to Average Power Ratio (Cociente Pico-a-Medio de Energía).
PC	Point Coordination (Punto de Coordinación).
PCF	Point Coordination Function (Función de Coordinación Puntual).
PIFS	PCF Interframe Space (Método PCF de Ínter Espaciado de Tramas).
PLCP	Physical Layer Convergenve Protocol (Protocolo de Convergencia de Capa Física).
PMD	Physical Medium Dependent (Sistema Físico Dependiente del Medio).
PSK	Phase Shift Keying (Modulación por Desplazamiento de Fase).
PSK	Pre-Shared Key (Modo de Clave Compartida, se usa en conjunto con WPA).
PtMP	Point to Multi Point (Punto a Multipunto).
PtP	Point to Point (Punto a Punto).
PVR	Personal Video Recorder (Grabadora Personal de Video Digital).
QAM	Quadrature Amplitude Modulation (Modulación por Amplitud en Cuadratura).
QoS	Quality of Service (Calidad de Servicio).
RADIUS	Remote Authentication Dial-In User Server (Servidor de Autenticacion y Autorización).
RISC	Reduced Instruction Set Computer (Set de instrucciones Reducidas para Procesador).
ROM	Read Only Memory (Memoria de Solo Lectura).
RTP	Real Time Protocol (Protocolos de Tiempo Real).
SDA	Subscriber Data Adapters (Adaptador de la Unidad del Subscriptor de Airspan).
SGI	Silicon Graphics Inc (Empresa dedicada al desarrollo de sistemas de computación avanzada para procesadores).
SIFS	Short Interframe Space (Espacio Ínter Trama Corto).
SOC	Sistem On a Chip (Sistema Embebido en un Chip).
SPR	Subscriber Premises Radio (Equipo Para el Subscriptor Ubicado en las Instalaciones del Cliente de Airspan).
SU	Subscriber Unit (Unidad del Subscriptor de Alvarion).

TCO	Total Cost of Ownership (Costo Total de Propiedad).
TGn Sync	Task Group 'n' synchronization (Consorcio que reune empresas como Agere, Atheros, Intel, Sony, Nortel, Samsung, Qualcomm, Philips, Panasonic para el desarrollo en MIMO).
TKIP	Temporal Key Integrity Protocol (Protocolo de Integridad de Calve Temporal).
WAN	Wide Area Network (Red de Área Extensa).
WDS	Wirless Distribution System (Sistema de Distribución Inalámbrico).
WECA	Wireless Ethernet Compatibility Alliance (Alianza Para la Compatibilidad de Ethernet).
WEP	Wired Equivalency Privacy (Sistema de Cifrado Incluido en el Estándar IEEE 802.11 Como Protocolo Para Redes Inalámbricas).
WiBRO	Wireless Broadband (Equivalente a WiMax Coreano).
Wi-Fi	Wireless Fidelity (Es un conjunto de estándares para redes inalámbricas basado en las especificaciones IEEE 802.11).
WiMAX	Worldwide Interoperability for Microwave Access (Interoperabilidad Mundial para Acceso por Microondas).
WLAN	Wireless LANs (Redes de Área Local Inalámbricas).
WPA	Wi-Fi Protected Access (Acceso Protegido Wi-Fi).
WWiSE	World-Wide Spectrum Efficiency (Es un consorcio que está compuesto por compañías como Airgo Networks, Broadcom, Motorota, Nokia, Conexant Systems, STMicroelectronics, Texas Instruments, France Telecom, NTT, para el desarrollo de MIMO).

RESUMEN

El esquema de construcción y los diferentes tipos de memorias flash que se fabrican hoy en día, ha permitido el surgimiento de equipos en los cuales se aprovecha la versatilidad que requieren los dispositivos embebidos. En un principio éstos se consideraban como inalterables, debido a que demandaban de una actualización directamente en su hardware para cambiar alguna funcionalidad o añadir una característica adicional, lo cual resultaba bastante costoso y dispendioso.

Hoy en día, es posible contar con memorias reprogramables que son capaces de alojar un software altamente especializado llamado firmware, el cual es desarrollado y compilado para una arquitectura objetivo específica, en la cual se pueden realizar modificaciones e inserciones de características adicionales mediante programación software. Estas modificaciones, al incluir código abierto, son cubiertas por la licencia GPL, la cual permite a los programadores introducir sus desarrollos en un hardware que antes no podia ser transformado.

Dicha circunstancia dió origen a este trabajo, el cual se centra en la construcción de un firmware basado en el kernel de Linux para un dispositivo especifico, este proceso se muestra mediante la creación de una aplicación, partiendo del código desarrollado en lenguaje C/C++ y lo lleva hasta su implementación funcional dentro del hardware, mostrando el camino hacia el aprovechamiento de su potencialidad para realizar funciones adicionales a las que presenta originalmente.

Se utiliza como marco teórico los avances del software de código abierto y el análisis del estándar con el fin de implementar características adicionales en el firmware prototipo, y realizar la comparación con el original para el logro de los objetivos de este trabajo de grado.

INTRODUCCIÓN

Hoy en día, los computadores están presentes en todas las áreas de la actividad humana: en el hogar, la oficina, los bancos, las escuelas, las universidades, la industria, etc. Aunque en algunos casos los computadores realizan sus funciones en forma aislada, en otros existe la necesidad de intercambio de información con dispositivos como cajas registradoras, computadores de mano, impresoras y computadores. Esto significa que para la construcción de la mayor parte de los equipos de cómputo que se instalan, se debe tener en cuenta los equipos hardware y software que se utilizarán para la comunicación entre estos. Dentro de la amplia gama de opciones disponibles se encuentran características diferenciadoras como lo son las redes cableadas (ya sea por cobre o fibra óptica), y las redes inalámbricas, las cuales le han dado al hombre la posibilidad de movilizarse y estar conectado al mundo proporcionándole innumerables aplicaciones para las labores que realiza.

A menudo los usuarios encuentran problemas como la dependencia del proveedor de la tecnología debido a la incompatibilidad que se puede presentar entre los diversos proveedores, así como también el tiempo que puede tardar una actualización, desde un firmware (sistema operativo que funciona sobre un hardware determinado) para un punto de acceso, enrutador o un equipo de premisa de usuario (*CPE, Costumer Premise Equipment*) hasta un controlador para una tarjeta de red con el último sistema de seguridad. Por otra parte, si el fabricante lo permite, es posible modificar el firmware de algunos dispositivos de red. Esto es de gran trascendencia para países como Colombia en los cuales se abren caminos hacia el desarrollo de este tipo de tecnología, debido a que anteriormente un cambio de firmware implicaba que se debía contar con una gran capacidad tecnológica para manufacturar chips con alta escala de integración y un alto grado de desarrollo en hardware y software.

Hoy en día es posible realizar este trabajo de una manera más flexible; incorporar nuevas capacidades y características, derivadas de la utilización del software de código abierto basado en el robusto y potente kernel de Linux, en este momento se abre un camino infinito de posibilidades para los desarrolladores al proporcionar herramientas que permiten utilizar sus conocimientos dentro de un firmware para dispositivos activos en las redes de comunicaciones.

Es importante tener en cuenta la creciente popularidad de los sistemas operativos de libre distribución como Linux, en el cual se permite la lectura y modificación de muchos de sus parámetros, cosa que no es posible con sistemas que no son de código abierto. Además ofrece una gran estabilidad y un bajo precio, por lo cual se está explotando en las redes del mundo; haciendo que su crecimiento vaya en aumento junto con las aplicaciones servidoras que hacen funcionar Internet.

En países como en Colombia, el desarrollo en firmware para dispositivos hardware es escaso, las empresas que han diseñado hardware no tienen los medios para manufacturar chips con alta escala de integración, limitándose a la producción de chips de mediana y baja escala; pero cuando es necesario la alta o muy alta escala de integración, se deben llevar los planos del desarrollo al exterior para poder utilizar la última tecnología. Hoy por hoy, el proceso de desarrollo de firmware para dispositivos específicos está cambiando; se cuenta con memorias ROM, RAM, FLASH, entre otras, las cuales son cada vez más rápidas, de mayor capacidad y sobretodo más económicas; en concordancia con esto, algunos productores de tecnología han aprovechado esta ventaja y comienzan a incorporar este tipo de memorias en sus productos, ya que una memoria en términos genéricos puede albergar cualquier tipo de firmware y permite su posterior actualización. Además, para diferentes tipos de aplicaciones se cuenta con un gran número de posibilidades de desarrollo

y aprovechamiento más amplio de los sistemas hardware, logrando así sacar el máximo provecho a los dispositivos activos de red.

Este tipo de desarrollo es más eficiente, debido a que se abre la posibilidad para que entes diferentes al fabricante optimicen y mejoren aplicaciones dentro de un firmware, esto hace que sea más robusto y pueda ser reparado por otras personas, de manera que el nivel de contribución es mayor, porque está en el interés de todos mejorar esta base común y solucionar algún tipo de problema o incrementar las características de una plataforma sin esperar a que el fabricante distribuya una actualización.

La reutilización del software permite que se haga uso del saber y el conocimiento desarrollado en cada aplicación realizada, en vez de comenzar siempre desde cero (como es el caso de la industria de software actual). Se sabe de antemano que es posible empezar un proyecto desde unas bases establecidas, esto es equivalente a la manera en la que la ciencia se desarrolla: no se parte de cero, se parte de los descubrimientos previos y se innova sobre el conocimiento que ya se tiene. Se tiene la posibilidad de adaptar el software a las necesidades, esta es una capacidad conocida como personalización, la cual es importante en el desarrollo de este proyecto.

Actualmente, la Universidad del Cauca no cuenta con una forma de implementar muchos de los conocimientos recibidos sobre redes inalámbricas en dispositivos reales y de producción; llegar a este tipo de implementaciones representa una gran ventaja para los desarrollos futuros de los estudiantes, debido a que el hardware en el que se va a realizar ya se encuentra probado y aceptado. Este proyecto, enmarcado dentro del Grupo de Nuevas Tecnologías en Telecomunicaciones de la Universidad del Cauca, es una gran herramienta, la cual se debe comprender y utilizar ampliamente, con miras a generar entre la comunidad universitaria aplicaciones y desarrollos sobre un hardware que sirva de laboratorio y a la vez pueda ser incorporado directamente en el ámbito comercial. Es esta la razón de ser de este trabajo de grado, establecer un punto de partida de descubrimiento y análisis en el cual se da una introducción al funcionamiento y utilización de las herramientas para la creación y desarrollo de un firmware para dispositivos embebidos basado en el kernel de Linux, con la esperanza en que de aquí en adelante se pueden lograr muchos logros basados en la capacidad de los estudiantes para descubrir sus propios límites.

A manera de recuento, en el capítulo uno se encuentra una introducción a la situación actual de las redes inalámbricas y del software de código abierto a nivel nacional y mundial, de igual manera se escribe sobre el estado del arte del firmware de los sistemas embebidos, mostrando los conceptos con el fin de crear iniciativas a los lectores sobre el potencial de desarrollo que provee la asociación de estas tecnologías.

En el capítulo dos se da una mirada al estándar 802.11 en los puntos clave donde se puede realizar algún desarrollo que permita mejorar su funcionamiento, además, se proporciona información técnica sobre la construcción, funcionamiento y elaboración de un firmware.

En el capítulo tres se presenta la construcción paso a paso de un firmware basado en el kernel de Linux junto con el nuevo módulo que se incorporará en el firmware.

En el capítulo cuatro se presentan los cálculos teóricos de rendimiento esperados y se comparan con los obtenidos en la práctica por los dispositivos comerciales reales, se analizan y validan las herramientas de medición y por último se compara el funcionamiento del nuevo firmware con el anterior sacando las conclusiones más relevantes.

Al final de este documento se tiene la posibilidad de realizar experimentación y desarrollo de aplicaciones que ofrezcan nuevas capacidades y saquen mayor provecho a las plataformas hardware existente. Según los gurús del UNIX esto se denomina *hacks*, de donde se desprendió

el termino *hacker*, esto indica que son modificaciones que se le pueden hacer a un programa o máquina para mejorar o alterar su funcionamiento.

1 ANTECEDENTES DE LAS REDES INALÁMBRICAS Y ESTADO DEL ARTE DEL CÓDIGO ABIERTO

1.1 PROBLEMÁTICA DEL USO DE LAS REDES INALÁMBRICAS A NIVEL MUNDIAL

Desde finales de los años ochenta, las redes inalámbricas se han incrementado masivamente en todo el mundo. Los países con mayor tecnología y desarrollo las usan con variadas finalidades, principalmente en las comunicaciones militares, esto debido a que la comunicación en forma inalámbrica tiene una característica muy importante que la hace sobresalir sobre cualquier tipo de esquema o arquitectura de red, esta característica es la movilidad.

Cabe destacar que un factor por el cual el avance en las redes inalámbricas se ha detenido levemente es la seguridad en la transmisión de la información, ya que el factor de riesgo se incrementa cuando el medio de transmisión es abierto, es decir, el medio de transmisión y la información están disponibles para todo el que tenga las herramientas adecuadas como una tarjeta de red inalámbrica, un *Sniffer*[1] y otras aplicaciones que pueden vulnerar redes.

1.1.1 Realidad en las Redes Inalámbricas

No es un secreto que las ventajas en las redes inalámbricas saltan a la luz cuando se ponen en consideración en el ámbito comercial, es decir, las ventas marcan el rumbo de la tecnología en cualquier nivel y son los consumidores quienes la colocan a prueba, algunos con pocos conocimientos y otros con mayor conocimiento.

Existen muchos grupos en el mundo que investigan, desarrollan y experimentan con redes inalámbricas libres, estos grupos crecen con gran furor en cada continente día a día. Entre estos grupos se encuentran:

- Irish Wan, es una organización no lucrativa que está construyendo una red de banda ancha en Irlanda. La red está siendo construida por entusiastas de todas partes del país y un día se espera cubrir un porcentaje enorme de Irlanda.

- BC Wireless, es una organización no lucrativa de pueblos canadienses, dedicada a la creación de comunidades a través de redes inalámbricas y software de código abierto a través de la provincia de Columbia Británica.

- NycWireless, es una organización no lucrativa que aboga y permite el crecimiento del acceso inalámbrico libre y público a Internet en New York City y sus alrededores.

- Seattle Wireless, es una comunidad inalámbrica de pueblos sin ánimo de lucro, en los alrededores de Seattle y Washington, sus metas incluyen la creación de una red metropolitana de banda ancha, así como las herramientas para ayudar a conseguir dicho fin, sin la intervención de ninguna compañía de telecomunicaciones.

[1] Aplicación capaz de detectar las tramas que se propagan por el medio de transmisión

- France Wireless, es una asociación no lucrativa que tiene como meta el desarrollo de una red de área extensa inalámbrica, con la filosofía de utilizar software libre, sin pago y para todas las personas.

- Redlibre, es un proyecto grande en España, donde se reúne mucha gente con unos objetivos en común: Crear una red libre, acercar la tecnología favoreciendo la comunicación de la sociedad y crear una red de emergencia para su uso en caso de catástrofe.

1.1.2 Ventajas de las Redes Inalámbricas

- Movilidad

Dentro de la zona de cobertura de este tipo de redes, los nodos y equipos se pueden comunicar prescindiendo de cables para intercambiar información, por ejemplo, se puede realizar cualquier tipo de presentación en Internet y teniendo un computador con interfaz inalámbrica se puede acceder a esta información desde y hacia cualquier lugar del mundo.

- Flexibilidad en la planificación

Antes de cablear un lugar, edificio, oficinas, hogares, entre otros, se debe pensar detenidamente en la distribución física de los equipos, mientras que con una red inalámbrica sólo existe la preocupación de que las estaciones inalámbricas queden dentro del área de cobertura.

- Diseño

Los transmisores y receptores son bastante pequeños y pueden integrarse dentro de un dispositivo y llevarlo en un bolsillo. Cada vez aumentan el número de equipos que traen consigo dispositivos de conexión inalámbrica, estos pueden ser: enrutadores, computadores de escritorio, portátiles, computadores de mano, celulares, entre otros.

- Independencia de un medio físico

Ante eventos inesperados, como un usuario que desconecta un cable, hasta una catástrofe mayor, en el cual una red cableada podría llegar a quedar completamente inhabilitada, una red inalámbrica se sobrepone mejor este tipo de percances, debido a no se necesita de un medio físico para la propagación de la información, además de factores como la fácil y rápida instalación de equipos en cualquier lugar.

1.1.3 Inconvenientes de las Redes Inalámbricas

Después de una etapa inicial, donde las redes inalámbricas se introdujeron al mercado mundial y fueron mejoradas en los años siguientes, aún existen inconvenientes que delimitan las fronteras de este tipo de tecnología. Esas fronteras no siempre están en el ámbito tecnológico, sino también, en la regulación hecha por los entes que manejan y gestionan el uso de este tipo de redes.

- Calidad de Servicio

Las redes inalámbricas ofrecen una calidad de servicio que no se compara con las redes cableadas, es decir, la calidad del servicio (QoS, Quality of Service) no es lo suficientemente alta, esto debido a diversos factores como el medio de transmisión, entre otros. Se habla de velocidades que no superan habitualmente los 108 mega bits por segundo, en contraste con los 10 giga bits por segundo que se pueden alcanzar utilizando cobre o los 160 Gbps mediante fibra óptica [1]. Por otra parte, se debe tener en cuenta también la tasa de error debido a las interferencias, la cual se puede situar alrededor de 10e-4 frente a 10e-10 de las redes cableadas. Esto significa que hay 6 órdenes de magnitud de diferencia, es decir, se habla de 1 bit erróneo cada 10.000 bits. Esto puede llegar a ser imposible de implantar en algunos entornos industriales con fuertes campos electromagnéticos y ciertos requisitos de calidad.

En las actuales redes 802.11a/b/g, no existe una definición para el trato a los diferentes tipos de tráfico que se encuentran en una red inalámbrica, no se diferencia entre alta o baja prioridad. Esta deficiencia se nota en aplicaciones de voz y video, las cuales son catalogadas como de alta prioridad y presentan gran sensibilidad al retardo y al *jitter*[2].

Una de las dificultades en el método de acceso de las redes inalámbricas se presenta cuando una estación obtiene el control del medio, en este momento la estación decide cuanto tiempo necesita para transmitir de acuerdo a la cantidad de información y la tasa de transmisión a la que opera. Si una estación se encuentra funcionando a una baja velocidad de transmisión (1Mbps), necesitará un tiempo alto de ocupación del medio, lo cual produce que las demás estaciones sufran retardos y *jitter* alto. Esto es desfavorable cuando se utilizan protocolos de tiempo real (RTP, Real Time Protocol), para transportar voz o video.

Otro inconveniente se presenta cuando existen muchas estaciones transmitiendo al mismo tiempo, esto genera colisiones, lo cual reduce el ancho de banda para cada estación, tal como ocurre en los protocolos basados en detección de portadora (CSMA, *Carrier Sense Multiple Access*).

En general no existe una implementación de QoS para las redes inalámbricas actuales, debido a esta situación se estableció y finalmente se aprobó el 21 de Septiembre de 2005 [2] el estándar 802.11e en el cual se establece la enmienda para la capa física y la capa de acceso al medio como una mejora para QoS en las redes inalámbricas, pero aún falta su implementación efectiva por parte de los fabricantes.

- Costos

El costo de montaje de una red inalámbrica es cada vez menor, sin embargo, no se puede comparar con el costo actual de una red cableada. Los dispositivos que conforman una red inalámbrica se basan en tarjetas y puntos de acceso principalmente, y siguen siendo un poco más costosos que los adaptadores de red, switches y enrutadores ethernet para redes cableadas de gama baja.

Desde el punto de vista futuro, se prevé que en esta década, el montaje de equipos inalámbricos estará igualando el valor del montaje de equipos cableados, cuando se evalúa la capacidad de las redes inalámbricas de disminuir costos en el tendido de cable en largas distancias y en el costo total de propiedad (*TCO, Total Cost of Ownership*)

[2] Variación del retardo que sufren los paquetes al pasar por la red.

- Soluciones Propietarias

Como la estandarización total ha sido lenta, ciertos fabricantes han sacado al mercado algunas soluciones propietarias que sólo funcionan en un entorno homogéneo, por lo tanto se necesita estar muy ligado con el fabricante en la implementación de una red. Esto supone un gran problema para el mantenimiento del sistema, tanto para ampliaciones como para la recuperación ante posibles fallos. Cualquier empresa o particular que desee mantener su sistema funcionando se verá obligado a acudir de nuevo al mismo fabricante para comprar otra tarjeta o punto de acceso, entre otros.

- Restricciones

Para su funcionamiento, las redes inalámbricas deben amoldarse a la disposición del espectro radioeléctrico de cada país y debido a que operan en bandas no licenciadas, esta parte del espectro se encuentra muy saturado hoy en día. Además, concretamente en muchos países europeos como España, Francia o países asiáticos como Japón, existen limitaciones en el ancho de banda a utilizar por parte de ciertos estándares. Cabe resaltar que las frecuencias libres en Colombia son: 900 Mhz, 2.4 Ghz, 5.4 Ghz, 5.8 Ghz [3].

- Seguridad

En cuanto a la seguridad e integridad de la información que se transmite se han visto avances, como el reemplazo del protocolo de cifrado para redes inalámbricas (*WEP, Wired Equivalency Privacy*) por el protocolo de acceso protegido Wi-Fi (*WPA*, *Wifi Protected Access*), puesto que el primero presentó un bajo nivel de seguridad desde el principio.

Respecto al estándar 802.11i el cual se trata en la sección 1.1.4, todavía sufre de una condición inherente al medio inalámbrico: cualquiera puede acceder al medio de transmisión, esto propone un reto a todos los sistemas de seguridad que se desarrollen y por lo pronto hace que no sea recomendable utilizar redes inalámbricas en entornos críticos en los cuales un “robo” de datos pueda ser peligroso.

Por otra parte este tipo de comunicación podría interferir con otras redes de comunicación (policía, bomberos, hospitales, entre otros), argumento que se debe tener en cuenta en el diseño de este tipo de redes.

1.1.4 Seguridad Inalámbrica

Desde el punto de vista de la seguridad existen diferentes tipos de estándares que constituyen un avanzado puente tecnológico entre los débiles sistemas de los primeros dispositivos y los más recientes como lo es el estándar 802.11i.

Por defecto, IEEE 802.11 provee de un mecanismo de cifrado de datos llamado *WEP* que se ha mostrado débil en ámbitos empresariales, por lo cual no es una opción recomendable para entornos de altos requerimientos de seguridad. Con el tiempo se han evidenciado diversas formas de resolver algunos de los problemas (fundamentalmente autenticación y cifrado de datos con claves dinámicas) en redes con servidores de autenticación, usando para ello herramientas ofrecidas por los sistemas operativos.

El protocolo *WEP* se basa en dos componentes para cifrar las tramas que circulan por la red: el algoritmo de cifrado *RC4* y el algoritmo de chequeo de integridad *CRC*.

RC4 es un algoritmo de cifrado de flujo. Es decir, funciona expandiendo una semilla o *seed* para generar una secuencia de números pseudo-aleatorios de mayor tamaño. Esta secuencia de números pseudo-aleatorios se unifica con el mensaje mediante una operación *XOR* para obtener un mensaje cifrado. Uno de los problemas de este tipo de algoritmos de cifrado es que no se debe usar el mismo *seed* para cifrar dos mensajes diferentes, ya que obtener la clave sería trivial a partir de los dos textos cifrados resultantes. Para evitar esto, *WEP* especifica un *vector de inicialización (IV, Initialization vector),* el cual es un grupo de 24 bits que se modifica regularmente y se concatena a la contraseña. A través de esta concatenación se genera el *seed* o raíz, que sirve de entrada al algoritmo *RC4*, para evitar secuencias iguales; de esta manera se crean *seeds* nuevos cada vez que varía el *IV*.

- *WPA* (*Wi-Fi Protected Access*)

WPA nació en el último cuarto del año 2002, sin embargo, se publicó varios meses después por la Alianza Wi-Fi. Al mismo tiempo se anunció que estaba abierto el proceso de certificación para productos acordes con WPA.

WPA es un subconjunto de tecnologías que se han extraído del estándar 802.11i, los cuales se implementaron comercialmente para resolver los problemas de seguridad existentes en Wi-Fi. Este incorpora autenticación mediante 802.1x y el protocolo de autenticación extensible (*EAP, Extensible Authentication Protocol*), además, se establece el protocolo de integridad de calve temporal (*TKIP, Temporal Key Integrity Protocol*) como protocolo para garantizar la integridad de la información.

WPA posee dos características principales: una es proveer un sistema de acceso seguro y autenticación para usuarios, además de proteger los datos transmitidos de una forma más segura que *WEP*. La otra característica adicional que lo convierte en una solución idónea inmediata es la posibilidad de hacerlo funcionar en el hardware actual ya que sólo requiere cambios de software y firmware.

- Control de acceso y autenticación de usuarios

El uso de 802.1x implica normalmente la puesta en marcha de un servidor de autenticación y autorización (*RADIUS, Remote Authentication Dial-In User Server)*, que en la mayoría de los casos, no está al alcance de todos los usuarios de este tipo de redes. En entornos empresariales en los que exista un servidor *RADIUS*, éste actuará de forma similar a las soluciones más robustas. Sin embargo, en entornos domésticos o de pequeñas oficinas en los que no esté disponible tal servidor, existe un modo de trabajo que no necesita nada especial para funcionar pero que de todas formas ofrece una cierta seguridad en el acceso. Este modo específico de trabajo se llama modo de clave compartida (*PSK*, *Pre-Shared Key*). Como su nombre lo indica, el único requerimiento es compartir una clave entre los diferentes clientes que se van a autenticar en la red. Si la clave de un cliente inalámbrico coincide con la del punto de acceso se le otorga acceso o es denegado en caso contrario. Esta clave no se envía al punto de acceso al intentar la autenticación sino que es el origen de un trabajo criptográfico que finalmente conduce a la autenticación, por lo que no es posible averiguarla rastreando las emisiones. Es claro que no es tan seguro como el uso de un servidor *RADIUS* pero es suficiente en entornos que necesitan conectarse de forma segura a pocos equipos.

- Cifrado de datos

El cifrado de datos corresponde a la protección de la información que se intercambia. Uno de los principales problemas de *WEP* es su relativa debilidad desde el punto de vista criptográfico. Para descifrar las comunicaciones con *WEP* es necesario capturar una cantidad de datos intercambiados y analizarlos. En entornos de poco tráfico como pequeñas oficinas o casas particulares se pueden tardar muchos días en reunir la información necesaria y por lo tanto *WEP* constituye una solución bastante adecuada dada su relativa simplicidad. En entornos empresariales en los que el volumen de información que se transporta es muy elevado es posible que un atacante preparado rompa las claves *WEP* en cuestión de unas horas o minutos, usando un portátil a distancia y sin ser detectado. Debido a esto se introduce la utilización de algunas características del protocolo IEEE 802.1x mediante *WPA*. La mejora consiste en la regeneración de claves para el cifrado de paquetes y claves globales de manera obligatoria. En el caso de claves establecidas con un único cliente, el protocolo *TKIP* es el encargado de hacerlo, manteniendo sincronizadas las claves entre cliente y punto de acceso, mientras que en el caso de claves de tráfico global, se incluye un sistema que permite al punto de acceso el envío seguro de la nueva clave a los clientes que estén conectados.

TKIP incrementa el tamaño de las claves de 40 a 128 bits y sustituye las claves estáticas de *WEP* por claves dinámicamente generadas y distribuidas por el punto de acceso tras la autenticación de los usuarios. El sistema empleado elimina la predicción de las claves en la que se apoyan los posibles atacantes para descifrar *WEP*. Después de autenticar a un usuario, se utiliza 802.1X para generar una clave única que se utilizará en la sesión, *TKIP* envía esta clave al cliente que se acaba de incorporar y establece un sistema de gestión de claves que permite generar sincronizadamente una nueva clave única diferente para cada paquete de datos que se envía, que por lo tanto se cifra de manera única.

En el caso del modo *PSK*, la diferencia principal es que la clave inicial no es única para cada sesión sino que es la misma para todos los clientes (está basada en la clave previamente compartida) ya que no existen datos de autenticación únicos para cada usuario. Ello reduce la seguridad en gran medida puesto que si se averigua la clave compartida (mediante fuerza bruta, ataques de diccionario, o algún método de ingeniería social) se tendrá acceso a la red y los datos descifrados. De todos modos el sistema es más seguro que *WEP*, esto debido a que *WPA* no elimina el proceso de cifrado *WEP*, sino que lo fortalece con una clave de 128 bits y un vector de inicialización que pasa de 24 a 48 bits. Adicionalmente *WPA* mejora la integridad de la información cifrada. Al incrementar el tamaño de las claves, el número de llaves en uso, y al agregar un sistema de verificación de mensajes, *WPA* hace que la entrada no autorizada a redes inalámbricas sea más difícil.

- Integridad de los datos

En 802.11 con *WEP*, la integridad de los datos intercambiados se trata de conseguir a través de un valor de chequeo de integridad de 32 bits llamado (*ICV, Integrity Check Value*) que se anexa al tráfico intercambiado y se cifra con *WEP*. A pesar de que el *ICV* está cifrado se pueden emplear técnicas de criptoanálisis para cambiar bits de información en los paquetes y cambiar el *ICV* sin ser percibido.

En *WPA* existe un algoritmo llamado Michael, que se usa para el cálculo de códigos de integridad de mensaje *(MIC, Message Integrity Code*). *MIC* fue hecho para evitar que un atacante capture paquetes, los modifique y los reenvíe. Michael define una avanzada función matemática que se calcula en cada paquete por parte del emisor y el receptor, intercalándola el primero entre el ICV y

los datos en el paquete enviado y finalmente cifrando el conjunto con *TKIP*. Al recibir el paquete se compara el MIC calculado con el contenido en éste y si no coinciden se asume que los datos han sido modificados, descartándolo. Además, *WPA* incluye protección contra ataques de "repetición de tramas" (*replay attacks*), ya que incluye un contador de tramas. El algoritmo Michael fue el más fuerte que los diseñadores de *WPA* pudieron crear, bajo la premisa de que debía funcionar en las tarjetas de red inalámbricas más antiguas; sin embargo, también es susceptible a ataques.

- Soporte de cifrado avanzado

WPA permite usar opcionalmente el algoritmo estándar de cifrado avanzado *(AES, Advanced Encryption Standard*). Se trata de uno de los algoritmos de claves simétricas más seguro que existe en la actualidad pero que requiere un cierto tipo de conocimiento para implementarlo. Este es el motivo por el cual su uso es opcional en *WPA*, ya que una gran parte del hardware existente en los dispositivos inalámbricos es sencillo e insuficiente para ejecutar *AES*. Sin embargo, algunos fabricantes lo ofrecen en las actualizaciones de sus productos.

- Interoperabilidad de *WPA*

WPA puede operar simultáneamente con equipos antiguos basados en 802.11 y *WEP*, aunque con la desventaja de que las claves de cifrado no son dinámicas. Esta limitación es significativa ya que reduce su funcionalidad casi al mismo nivel que *WEP*.

Según [4] se afirma que: “Esta interoperabilidad entre dispositivos antiguos y nuevos no forma parte de la especificación WPA aunque sí es posible que aparezca reflejada en 802.11i (WPA2). La Alianza Wi-Fi recomienda que no se utilice este modo de trabajo mixto, debido a que casi invalida las ventajas de WPA. De hecho, el proceso de certificación Wi-Fi para WPA incluye una prueba para asegurar que el dispositivo que se está evaluando no soporta el modo mixto en su configuración por defecto, es decir, que si lo soporta será el administrador de red quien conscientemente lo deberá activar.”5

Para admitir la transición gradual de las redes inalámbricas basadas en *WEP* a *WPA*, un punto de acceso inalámbrico puede admitir ambos clientes *WEP* y *WPA* al mismo tiempo. Durante la asociación, el punto de acceso inalámbrico determina qué clientes utilizan *WEP* y cuáles utilizan *WPA*. La compatibilidad con una combinación de clientes *WEP* y *WPA* resulta problemática. La clave de cifrado global no es dinámica, porque los clientes basados en *WEP* no lo admiten, pero se mantienen todas las demás ventajas de los clientes *WPA*, como la integridad.

El uso de *WPA* requiere la actualización de todos los componentes que intervienen en una red inalámbrica, estos son, los puntos de acceso, las tarjetas de red inalámbricas, los controladores de dispositivo y el software que se instala en los equipos cliente, estos cambios se pueden realizar mediante la actualización de su firmware y en algunos casos que el fabricante no libere la actualización de éste, es necesario un reemplazo de hardware.

*WPA*2 está basada en el nuevo estándar 802.11i, el cual fue ratificado en Junio de 2004, mientras que *WPA*, por ser una versión previa se puede considerar de migración, ya que no incluye todas las características del IEEE 802.11i, por su parte se puede afirmar que *WPA*2 es la versión certificada del estándar 802.11i. La alianza Wi-Fi llama a la versión de clave pre-compartida *WPA-Personal* y *WPA2-Personal* y a la versión con autenticación 802.1x/*EAP* como *WPA-Enterprise* y *WPA2-Enterprise*. Se puede apreciar una comparación de los principales estándares en la Tabla 1.

Tabla 1. Estándares de Seguridad Inalámbricos

	WEP	WPA	802.11i (RSN, WPA2)
Algoritmo Cipher	RC4	RC4 (TKIP)	Rijndael (AES-CCMP)
Llave de Encriptación	40 bits	128 bits (TKIP)	128 bits (CCMP)
Vector de Inicialización	24 bits	48 bits (TKIP)	48 bits (CCMP)
Llave de Autenticación	Ninguno	64 bits (TKIP)	128 bits (CCMP)
Revisión de Integridad	CRC - 32	Michael (TKIP)	CCM
Llave de Distribución	Manual	802.1x (EAP)	802,1x (EAP)
Llava Unica a:	Red	Paquetes, Sesión, Usuario	Paquetes, Sesión, Usuario
Llave de Jerarquía	No	Derivada de 802.1x	Derivada de 802.1x
Negociación Cipher	No	Si	Si
Seguridad Ad-hoc (P2P)	No	No	Si (IBSS)
Pre-Autenticación (Lan Cableada)	No	No	Usa 802.1x (EAPOL)

1.1.5 Desafíos Tecnológicos

La gran mayoría de empresas desarrolladoras de equipos y sistemas para redes inalámbricas han empezado a invertir tiempo y conocimientos en el aprovechamiento del espectro electromagnético y desarrollo de aplicaciones en otras capas del modelo, es por esto que cada vez es más evidente el desarrollo de estructuras y características como priorización de tráfico o IP Precedente, el cual es un método previo donde se etiquetan y clasifican los paquetes que van a ser priorizados.

1.1.6 Nivel de Maduración

Una de las clasificaciones de las redes inalámbricas es la siguiente: [6]

- Sistemas vía satélite (*INMARSAT, IRIDIUM*)
- Redes de área extensa de transmisión de datos (*WATM*)
- Redes móviles privadas (*Wireless Ethernet*)
- Redes de telefonía celular públicos (*GSM, GPRS, UMTS*)
- Redes de telefonía sin hilos (*DECT*)
- Redes domésticas (*Home RF*)
- Redes de área personal (*Bluetooth*)

Cada tipo de red presenta diferente evolución, tanto en equipos como en forma de control y precio, además de un factor muy importante como es la utilidad de las mismas. Cada red en su grupo lleva un desarrollo escalonado y organizado hoy en día, es importante tener en cuenta que los grupos de desarrollo de cada tecnología siguen y buscan software y controladores que permitan aprovechar cada una de las características técnicas de los dispositivos.

1.2 SITUACIÓN DE LAS REDES INALÁMBRICAS EN COLOMBIA

Prácticamente en esta década, se empezó a hablar de redes inalámbricas de transmisión de datos en el país, esto posiblemente por la falta de conocimiento sobre el tema y principalmente por el valor que pudo llegar a tener en su momento el montaje de redes extensas de comunicación basadas totalmente en enlaces inalámbricos. Hoy en día, existen muchos grupos de personas que se han dedicado al desarrollo de aplicaciones sobre este tema, han combinado las diferentes características tanto de hardware como de software, para hacer desarrollos que mejoren entornos de trabajo. Actualmente se puede contar con redes inalámbricas totalmente libres en algunas ciudades de Colombia, una de las más representativas es Medellín.

1.2.1 Empresas que Ofrecen Soluciones

Empresas como Coldecón S.A., Flycom, Enred, Telesat S.A., Geonet y ETB entre otras, ofrecen soluciones de comunicación e interconexión masiva mediante el uso de Redes Inalámbricas, generando un incremento en el auge de la Banda Ancha en Internet, lo cual da como resultado la introducción de nuevas tecnologías de Acceso a Colombia. Es importante destacar que la gran mayoría de empresas que ofrecen servicios y utilizan sistemas de Redes Inalámbricas, se basan principalmente en el uso de equipos con frecuencias libres, ya sea en bandas de 900 Mhz, 2.4 Ghz, 5.1 Ghz, 5.2 Ghz, 5.4 Ghz o 5.8 Ghz.

Según resolución 2064 de 2005 [7] del ministerio de comunicaciones de Colombia, se establece para la operación de los sistemas de distribución punto a punto y punto multipunto para acceso de banda ancha inalámbrica, la banda de frecuencias radioeléctricas comprendida entre los 3400 MHz a los 3600 MHz, se especifica además la adjudicación de 3 licencias a nivel nacional de 42 Mhz del espectro radioeléctrico, las cuales han sido asignadas a Telecom, ETB y EPM y dos licencias a nivel regional de 28 Mhz, las cuales se adjudicarán por concurso a operadores particulares. Las propuestas fueron recibidas hasta el 14 de julio de 2006 y el criterio de selección del ministerio es la capacidad de cobertura actual y futura, teniendo en cuenta lo determinado por la Comisión de Regulación de Telecomunicaciones en el documento “Promoción y Masificación de la Banda Ancha en Colombia Versión II” [8], en el cual recalca que uno de cuyos objetivos es el “Fomento al uso de nuevas tecnologías, tanto alámbricas como inalámbricas y sus consideraciones normativas y regulatorias, como es el caso del estándar 802.16 (*WiMAX*) y PLC entre otros”.

Otras empresas como Telebucaramanga y Telesat han utilizado tecnologías de acceso de ultima milla en conjunto con WiFi, en lo que se puede llamar pre WiMax, en este caso se utiliza una tecnología de acceso con equipos outdoor, radios de gran capacidad pero costosos, en combinación con WiFi, con lo que se logra utilizar CPE indoor mas económicos. Esta clase de CPE es la que al final dominará el mercado, saltándose el cambio de tecnología, es decir un CPE indoor de gran capacidad a un precio moderado.

Es de resaltar que en las soluciones planteadas con tecnología de acceso de ultima milla y WiFi, las frecuencias pertenecen a la banda libre, por lo que han proliferado notablemente las redes inalámbricas en edificios y barrios. Esto presenta un reto constante para las empresas, quienes deben mantener una buena calidad de servicio para los clientes, por medio de una buena gestión de este tipo de redes.

A nivel mundial se ve el incremento de compañías capaces de ofrecer un sin número de soluciones corporativas, desde el software necesario para la organización de una empresa hasta los accesos de última milla con salida a todos los servicios que estas requieren. Por otro lado, Colombia es un país en el cual el crecimiento de empresas que laboran en el campo de las TIC (Tecnologías de la Información y Comunicación) es cada vez más grande, de ahí que grupos de jóvenes universitarios formen día a día Pequeñas y Medianas empresas con carácter tecnológico, los cuales hacen que

se experimente con nuevas tecnologías y conocimientos, los cuales incrementan del desarrollo de las TIC en el país.

1.2.2 Empresas que Participan en Desarrollo

Parques industriales, como Parquesoft, apoyan y aplican soluciones en el desarrollo de software en diferentes situaciones, una de estas es el software para dispositivos móviles, que además de hacer funcionar unos equipos, ayuda a sacar mayor ventaja con aplicaciones determinadas, abriendo caminos en el sector de los servicios y el entretenimiento. Existen otras empresas como Coldecón S.A. que hace desarrollos privados sobre enlaces inalámbricos para clientes empresariales y presta servicios telemáticos para clientes que necesiten acceso en lugares distantes donde la única forma es el acceso inalámbrico.

1.2.3 Redes de Comunicación Libres

La idea de una red libre se puede adaptar perfectamente al concepto del software libre, cuyo concepto se resume en 4 premisas básicas [9]:

- La libertad de usar el programa con cualquier propósito.
- La libertad de estudiar cómo funciona el programa y adaptarlo a cualquier tipo de necesidades. El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias.
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

Tal como se menciona en Altred [10], estos 4 puntos se pueden aplicar a una red de la siguiente forma:

- La libertad de usar la red con cualquier propósito.
- La libertad de estudiar cómo funciona la red y adaptarla a cualquier tipo de necesidades. El acceso al código fuente es una condición previa para esto. El código fuente de una red pueden ser las instrucciones para hacer una copia de una red ya existente o enseñar a crear y configurar los elementos de una red.
- La libertad de crear nodos, repetidoras o clientes.
- La libertad de mejorar la red y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

Dentro de una red de computadores se comparten recursos de cualquier tipo, este es el objetivo de la sociedad de la información y es el caso de Internet. Aunque por ahora el acceso a Internet tiene un costo por el cual se debe pagar a un Proveedor, lo cual no cumple con la premisa de proporcionar a todas las personas acceso a las tecnologías de la información, pero se puede pensar en un futuro en una red alternativa como las planteadas por las comunidades inalámbricas para garantizar que el conocimiento esté al alcance de todo el mundo.

Por ahora, Internet es una gran fuente de información que hoy en día es necesaria para todos, por eso es importante que las personas, sin tener en cuenta su estrato y/o situación económica, tengan derecho a ingresar de una forma económica, sencilla y libre a la red de redes (Internet).

Además de acceso gratuito a Internet, una red libre puede prestar otros servicios a la comunidad como lo menciona redlibre en sus objetivos [11]:

- Acercar la tecnología y favorecer la comunicación de la sociedad: una Red Libre fomenta la instrucción técnica de los usuarios y acerca las nuevas tecnologías a los ciudadanos, eliminando muchas de las barreras que hoy en día existen para el pleno desarrollo de la sociedad de la información y creando nuevos canales de comunicación entre las personas de una manera absolutamente libre y gratuita.

- Crear una red de emergencia para su uso en caso de catástrofe: en este caso y en el consiguiente colapso de las redes de comunicación habituales, la Red Libre será una alternativa de comunicación al no depender de los canales, medios de transmisión habituales, permitiendo conexión a la red desde cualquier punto y en todo momento para servir en caso de emergencia y atender a las necesidades de comunicación y transmisión de voz y datos que puedan surgir.

Estos puntos son solo dos de las muchas aplicaciones que se le pueden dar a una red libre, también puede utilizarse para compartir archivos, recursos tales como impresoras y algún otro hardware, jugar en red, hacer video conferencia y emisoras de radio (*streaming media*), fomentar la educación *On-Line*, entre otros.

Para comenzar a crear una red libre, se necesita hardware y software. El software a utilizar debe ser software libre (*GNU, General Public License*) y/o *open source*; esto para lograr que la totalidad de la red sea libre en cuanto a su estructura de *Backbone*. Los clientes de la red libre pueden utilizar software con cualquier tipo de licencia y la responsabilidad de cumplir con los términos de cada licencia utilizada por el cliente es responsabilidad de él.

A medida que pasa el tiempo, son más los grupos de que se agregan al desarrollo y evolución en diferentes temas de esta ciencia y tecnología, algunos de estos grupos son:

MedellinWireless**,** IbagueWireless**,** BogotaWireless y PopayanWireless**.**

1.2.4 Tendencias Tecnológicas de las Redes Inalámbricas

En un principio, los requerimientos de las redes eran bajos, y lo que se transportaba era únicamente voz, con el pasar del tiempo la cantidad de voz fue aumentando y consigo el aumento del ancho de banda necesario para su transporte, un gran salto se introdujo cuando se empezó a integrar voz y datos en las redes, estos datos se fueron convirtiendo poco a poco en tráfico de toda clase, e incluso después de un tiempo la voz se convierte en datos para su transporte y luego aparece el tráfico multimedia, este tipo de tráfico requiere una gran cantidad de ancho de banda, por lo cual la tecnología sigue avanzando para suplir estas necesidades de voz datos y video; todo en una misma conexión y controlado por el protocolo IP, esta evolución se puede apreciar en la Figura 1 [12].

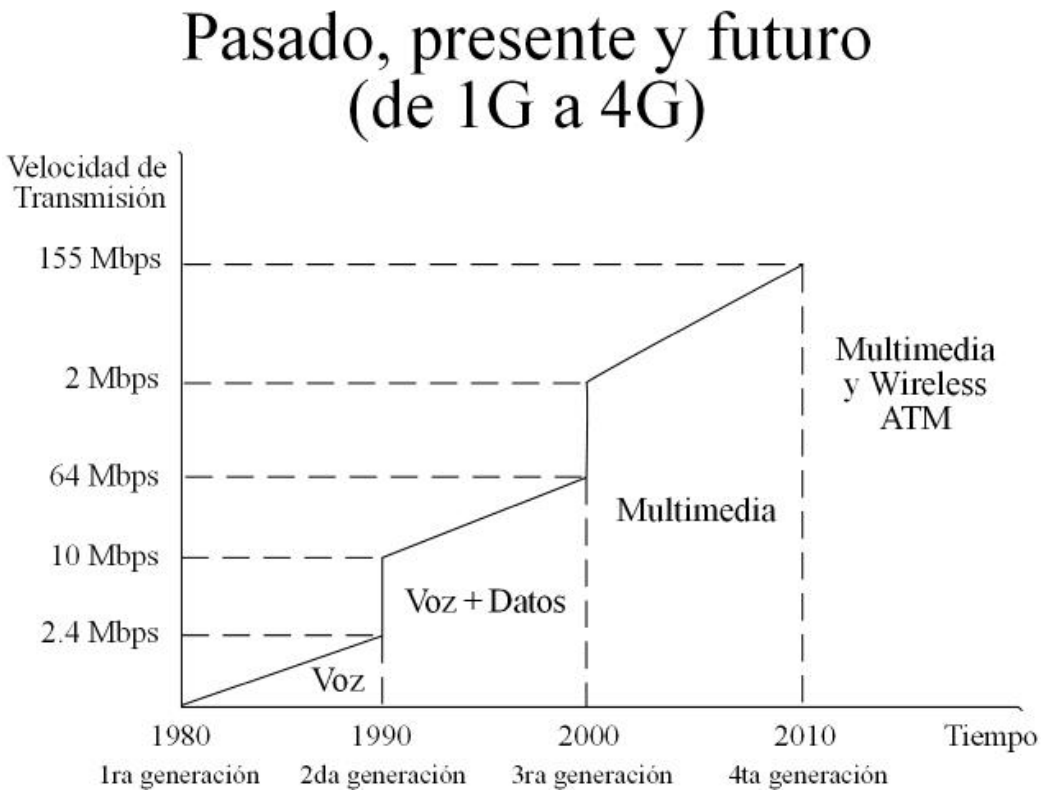

Figura 1. Pasado, Presente y Futuro de las Redes

1.2.5 El Futuro

La mayoría de tecnologías tienden a ser controladas por dispositivos sin cables, esto hace que cada día crezca el afán de conseguir mejores comunicaciones y equipos más robustos en todas las áreas. Las comunicaciones celulares, el control de periféricos, la comunicación de datos de alta velocidad entre muchas otras opciones han sido mejoradas con el tiempo.

El incremento en el uso de la Banda Ancha exige que cada proveedor de soluciones sea capaz de ofrecer comunicaciones de alta velocidad sin ningún tipo de inconvenientes en transmisión, debido a que las aplicaciones como Telefonía IP o videofonía tienen límites en cuanto a retardos y perdida de paquetes, ya sea por ruido, interferencia o condiciones atmosféricas.

En un futuro cercano, se vislumbra la aparición de IPTV la cual aumentará en gran medida los requerimientos de ancho de banda. Esto plantea un nuevo punto de vista para el proveedor de servicios de Internet y un gran panorama. A nivel general los ISP cuentan con 2 tipos de clientes, por un lado se encuentran los clientes corporativos que manejan un gran volumen de información, utilizando canales de datos para protocolos como SMTP, HTTP y SSL, para el correo, navegación y VPN´s respectivamente, además un cliente corporativo puede demandar video para cámaras de vigilancia, esto hace que los clientes corporativos consuman un alto tráfico cada uno. De otro lado se encuentran los clientes residenciales de banda ancha, los cuales manejan volúmenes de información moderados y consumen en promedio una décima parte del ancho de banda de un cliente corporativo promedio, pero a su vez, son diez veces más numerosos que los corporativos. El gran cambio puede surgir con los nuevos servicios de IPTV, debido a que los clientes que se

interesarían en este tipo de servicios son los clientes del hogar, y al ser tan numerosos demandarían un alto volumen de tráfico para los proveedores.

A nivel mundial la tendencia se encuentra en la tecnología de cuarta generación, la cual ofrece un gran ancho de banda y alta movilidad, algunas empresas están desarrollando a pasos gigantes y con grandes aspiraciones tecnologías como *WiMAX* y *WiBRO* (*Wireless Broadband*), estas prometen llegar de manera masiva a los clientes con alta movilidad y gran ancho de banda, empresas como Arispan [13] tiene productos certificados bajo el estándar 802.16d (WiMAX fijo) mediante sus celdas macromax y micromax. Alvarion, otra de las empresas líderes en WiMAX certificada en 802.16d, presentó en abril de 2006 su sistema 4Motion [14] el cual es compatible con 802.16e y se espera para finales del 2007 la puesta en marcha la comercialización de este producto certificado. Otra tecnología alineada con 802.16e es WiBRO, desarrollada en Corea por el instituto de investigaciones de electrónica y telecomunicaciones (ETRI, Electronics and Telecommunications Research Institute) y la industria coreana, principalmente por Samsung. Esta es una tecnología que ha logrado alta movilidad en sus primeras demostraciones, 60 Km/h con celdas de 1 a 5 Km y tasas de transferencia agregada de 20 a 30Mbps. En la Figura 2, se puede observar un panorama de las comunicaciones móviles hasta el momento [15], observando la tendencia hacia 4G la cual representa velocidades de hasta 1Gbps fijo y hasta 100Mbps en movimiento.

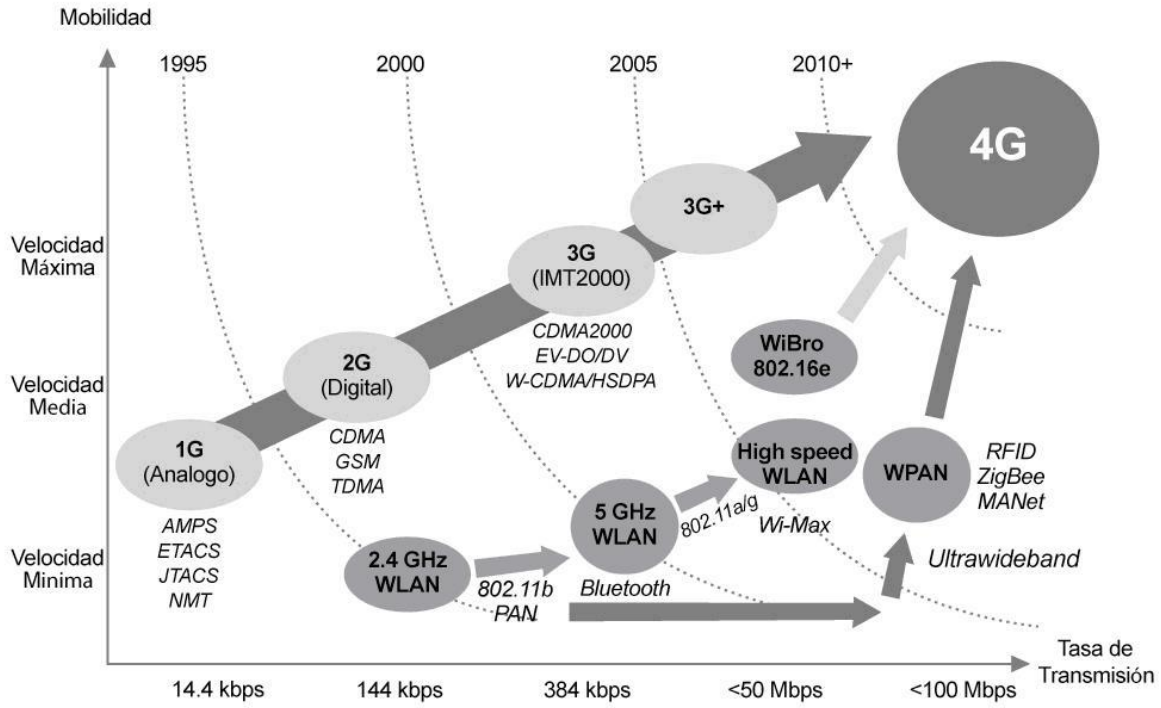

Figura 2. Presente y Futuro de las Redes Inalámbricas

1.3 ESTADO DEL ARTE DEL FIRMWARE

Una definición desde el ámbito legal dice sobre el firmware que es: “software que se almacena y ejecuta desde la memoria de solo-lectura (*ROM, Read Only Memory*) del computador. Su grado de

fijación al equipo puede ser absoluto, como en el caso de los programas implantados vía circuitos trazados, o relativo, como en el caso de los *ROM* reprogramables (*EPROM*). Para efectos jurídicos y por su propia naturaleza, el Firmware se considera accesorio del hardware y, por ende, sigue su misma suerte" [16].

Esto hace referencia a que el Firmware es considerado como software y debido a esto lo rigen las mismas leyes, estas leyes incluyen las licencias relacionadas al código abierto, el desarrollo a partir de código abierto y licencias de uso público (*GPL, General Public Licence*). Esto hace del Firmware un software con el cual se puede experimentar, modificar y mejorar al igual que se hace con el software de código abierto, el cual se convierte en una compilación en constante evolución, un premio a los desarrolladores de software en un campo en el cual antes solo los electrónicos y diseñadores de hardware tenían cabida.

El Firmware o programación en firme, es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria tipo *ROM*, que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Al estar integrado en la electrónica del dispositivo es en parte hardware, pero también es software, ya que proporciona lógica y dispone de algún tipo de lenguaje de programación. Funcionalmente, el Firmware es el intermediario, una interfaz entre las órdenes externas que recibe el dispositivo y su electrónica, ya que es el encargado de controlar a ésta última para ejecutar correctamente las órdenes externas.

Se encuentra Firmware en memorias ROM de los sistemas de diversos dispositivos periféricos, como en monitores de video, unidades de disco, impresoras, entre otros, pero también en los propios microprocesadores, chips de memoria principal y en general en cualquier circuito integrado.

Muchos de los Firmwares almacenados en ROM están protegidos por Derechos de Autor.

El programa BIOS de un computador es un Firmware cuyo propósito es activar una máquina desde su encendido y preparar el entorno para la instalación de un sistema operativo complejo, así como responder a otros eventos externos (botones de pulsación humana) y al intercambio de órdenes entre distintos componentes del computador.

En un microprocesador el Firmware es el que recibe las instrucciones de los programas y las ejecuta en la compleja circuitería del mismo, emitiendo órdenes a otros dispositivos del sistema.

Existen diversos Firmwares embebidos en diferentes sistemas que se utilizan hoy en día y los que se utilizarán en un futuro, es de especial importancia los que se encuentran basados en Linux, debido a las características de su licencia y posibilidad de modificación. A continuación se hace un recuento de algunos de estos dispositivos.

1.3.1 Media Center

Existen actualmente varias opciones de media center o centros de entretenimiento digital los cuales cuentan con varias opciones de hardware, sistemas operativos y firmware, que van desde los pequeños dispositivos para escuchar o ver videos de un computador a un televisor, hasta los complejos media center basados en Windows o en Linux. El media center ofrecido por la casa Microsoft es un sistema operativo adaptado, el cual se especializa en realizar funciones especificas de entretenimiento para el hogar, como reproducir videos, sintonizar televisión, reproducir y grabar audio entre otras funciones, el cual puede ser instalado en un computador basado en la plataforma PC al igual que se instala el sistema operativo Windows, es necesario que el hardware cumpla con los requisitos que el sistema necesita, como tarjeta de video, audio, sintonizador de televisión, lector de DVD y otros; siempre y cuando se disponga de controladores.

Los Media Hub no tienen forma de computador, sino más bien parecidos a un VHS o un DVD para hacerlo menos intimidante hacia el usuario inexperto que un computador. Fueron creados como un dispositivo hardware dotado de un disco duro de gran capacidad, y microprocesador, cuentan con funciones de audio, video, súper video, video compuesto, sonido envolvente, capacidad para sintonizar televisión, CATV, ver fotografías digitales, quemador de CD, DVD, acceso a Internet, entre otras capacidades, además incorporan un control remoto para ser controlado a distancia. Estos nuevos sistemas están diseñados para ofrecer al usuario un centro de entretenimiento digital en su hogar y dejar todo el trabajo a un solo equipo con todas las funciones de un hogar moderno y acceso a todos los medios posibles, de una manera rápida, fácil de manejar y robusta.

Existen también dispositivos que ofrecen la posibilidad de interactuar con los recursos del PC, ya sea para visualizarlos en el televisor o escucharlos en un equipo de audio. Estos artefactos se conectan por medio de una interfaz al computador y por otra realizan la función de pasarela entre todos los recursos de entretenimiento que tiene el usuario y los controla remotamente. Productos como el *Media MVP*, de *Hauppauge* (Música, Videos, Fotos), el cual ofrece características diversas de reproducción de audio y video, estaciones de radio de Internet, grabadora en diferentes formatos y otras características [17] son representativos de este género. Se puede apreciar que todas estas funcionalidades se encuentran soportadas por el Kernel de Linux, de manera que el cualquier fabricante puede personalizar su kernel, empaquetarlo e introducirlo en un dispositivo embebido. En la actualidad existe una tendencia por introducir Linux en muchos de los dispositivos comunes como las PDA y teléfonos celulares, existen organizaciones como *handhelds.org* [18], con mucho conocimiento sobre Linux y dispositivos embebidos, los cuales desarrollan día a día sobre plataformas móviles y dispositivos de mano.

Fabricantes como Intel y Hewlett Packard, también implementan estas funcionalidades utilizando el Kernel de Linux, el cual es robusto y eficiente para llevar a cabo todas estas tareas cuando se habilitan en él las características adecuadas. Existen proyectos como MythTv el cual es una grabadora personal de audio y video digital (PVR, *Personal Video Recorder-Digital Video Recorder*) creado para convertir un Linux en un Media Hub, desarrollado por *Robert Kulagowski* con la colaboración de muchas personas para convertir el computador en un centro de entretenimiento digital [19], con la posibilidad de modificar o añadir funciones adicionales.

Además existen en el mercado sistemas basados en Linux exitosos como el TiVo[20] que es un dispositivo que cumple con los requerimientos de un sistema de entretenimiento digital para el hogar, dispositivos como estos se encuentran en la mira de los proveedores de televisión satelital como DirecTV, de donde nació un producto que permite decodificar la señal directamente desde el satélite sin necesidad de utilizar el decodificador tradicional, esto permite, dependiendo de las limitaciones del operador de televisión, implementar funciones de adelantar, atrasar, pausar, reservar programas, grabar y otras características que antes no se podían implementar con la televisión convencional.

El sistema TiVo es el que más clientes tiene en Estados Unidos y por una suma más elevada se garantiza el sistema de manera vitalicia para el usuario, claro que los clientes están sujetos a la subsistencia de la empresa. Este es un terreno libre en Colombia el cual plantea posibilidades de negocio en este campo que está ganando bastantes adeptos en el mundo.

La sintonización de televisión, sonido envolvente, reproducción, grabación de video/audio y otras características de software embebidas en el hardware adecuado tienen una característica en común: están soportadas por el Kernel de Linux, un Kernel adaptado a unas necesidades especificas y pertinentes, el cual además de ejecutarse en un computador, puede ser encapsulado en un hardware, el cual adopta las características especiales de un dispositivo como cualquier otro que existe en el hogar o empresa y que brinda capacidad de modificación y actualización.

A nivel de los dispositivos de redes activos se observa un sistema operativo a partir de algunos enrutadores que realizan funciones bastante complejas. Dentro de estos se encuentran, desde la serie cisco 2500 en adelante, los cuales son bastante reconocidos en el mercado y solo permiten actualización de su sistema operativo (*IOS, Internetwork Operating System*) por su fabricante. Algunos sistemas de Cisco poseen su IOS en una tarjeta extraíble los cuales ofrecen la posibilidad de adquirir y reemplazar dicha tarjeta de manera muy sencilla, pero se está supeditado a su disponibilidad, adquisición y el considerable precio que se debe pagar por dicha actualización.

También se encuentran diferentes proyectos encaminados a un producto en particular, se puede observar cómo se utiliza el hardware de redes para realizar tareas como controlar cargas a distancia mediante dispositivos inalámbricos, utilizando para esto la flexibilidad de Linux.

Estudiando la construcción de los dispositivos para redes inalámbricas, se encuentra que los fabricantes ofrecen diversos productos como tarjetas madre con las características estándar para que un productor de hardware lo ensamble y venda bajo su marca. Estos dispositivos se basan en la arquitectura estándar con los chips adecuados para manejar el procesamiento central, manejo de las interfaces inalámbricas, interfaces cableadas y memoria entre otros. En la Figura 3 de la siguiente sección se muestra el esquema de construcción de un enrutador inalámbrico.

1.3.2 Esquema de Construcción de un Enrutador Inalámbrico

Esta clase de dispositivos cuentan con una interfaz conectada a un switch hardware de 5 puertos que soportan el protocolo 802.3/u, 4 de estos puertos pertenecen a la red de área local (*LAN, Local Area Network*) (1 al 4), el quinto es el puerto de enlace a la red de área extensa (*WAN, Wide Area Network*), además, posee un puerto inalámbrico que soporta los protocolos 802.11b/g. Los puertos están conectados al modulo enrutador, los cuales se identifican como `eth0, eth1 y eth2`, en términos de Linux. La interfaz `eth0` del enrutador hace la conexión al módulo *switch*, en el cual se pueden conectar los clientes para la red LAN cableada del cliente, un *switch o hub* adicional para dar acceso a más computadores, la interfaz `eth1` hace la conexión con el puerto *WAN* el cual permite conectarse con el mundo exterior, realizar enrutamiento o traducción de direcciones (*NAT, Network Address Traslation*) mediante la conexión a Internet que se disponga (ADSL, CABLE y otras tecnologías.), finalmente la interfaz `eth2` hace la conexión al puerto inalámbrico en el cual se conectan los clientes de manera inalámbrica, como computadores de escritorio o equipos portátiles, también se pueden conectar al dispositivo puente para ampliar el área de cobertura y otros enrutadores mediante el sistema de distribución inalámbrico (WDS, Wirless Distribution System). En la figura 3, se puede apreciar el esquema de un enrutador inalámbrico.

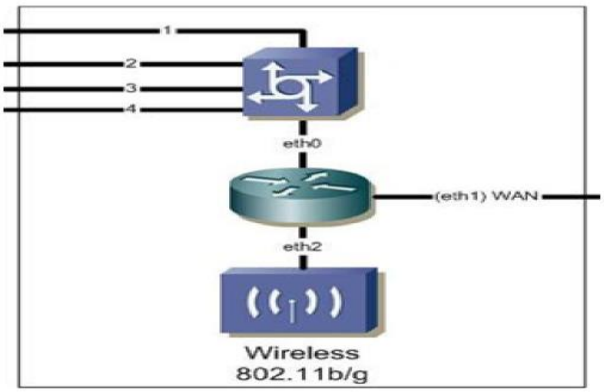

Figura 3. Esquema de Construcción de un Enrutador Inalámbrico

La manera en la cual se conectan las interfaces `eth0` (puertos *switch*) y la `eth2` (puerto inalámbrico) es mediante software, entre éstas se crea una interfaz virtual llamada `br0` en la cual se encuentran incluidas las interfaces `eth0` y `eth2` esto hace que tanto en los cuatro puertos de la red cableada como la red inalámbrica se encuentren en el mismo segmento en capa 2 del modelo OSI (*Open System Interconection*).

1.3.3 Microprocesadores

Un microprocesador es un conjunto de circuitos electrónicos altamente integrado para cálculo y control computacional. El microprocesador es utilizado como Unidad Central de Proceso en un sistema microcomputador y en otros dispositivos electrónicos complejos como cámaras fotográficas e impresoras, y como añadido en pequeños aparatos extraíbles de otros dispositivos más complejos como por ejemplo equipos musicales de automóviles.

Los microprocesadores modernos están integrados por millones de transistores y otros componentes empaquetados en una cápsula cuyo tamaño varía según las necesidades de las aplicaciones a las que van dirigidas, y que van actualmente desde unos pocos milímetros a unos centímetros. Las partes lógicas que componen un microprocesador son, entre otras: unidad aritmético-lógica, registros de almacenamiento, unidad de control, unidad de ejecución, memoria caché y buses de datos, control y dirección.

Parámetros significativos de un procesador son su ancho de bus (medido en bits), la frecuencia de reloj a la que trabajan (medida en hertzios), y el tamaño de memoria caché (medido en kilobytes).

Existen una serie de fabricantes de microprocesadores, como IBM, Intel, Zilog, Motorola, Cyrix y AMD, que lo largo de la historia y desde su desarrollo inicial, han mejorado enormemente su capacidad, partiendo desde los Intel 8080, Zilog Z80 o Motorola 6809, hasta los recientes Intel Itanium, Transmeta Efficeon o Pentium D. Actualmente los nuevos microprocesadores pueden manejar instrucciones de hasta 256 bits, después de haber pasado por los de 128, 64, 32, 16, 8 y 4 bits.

1.3.4 Tipos de Arquitecturas

Actualmente existen dos tipos diferenciados de arquitecturas, estas son: la CISC y RISC.

- Arquitectura basada en CISC (*Complex Instruction Set Computer*)

Es un tipo de arquitectura en donde cada instrucción puede utilizar varias instrucciones de bajo nivel como por ejemplo carga desde memoria, una operación aritmética y almacenamiento en memoria, todo en una sola instrucción. Sus funciones se caracterizan por ser muy amplias y permitir operaciones complejas entre operandos situados en la memoria o en los registros internos.

Este tipo de arquitectura dificulta el paralelismo entre instrucciones, por lo que en la actualidad, la mayoría de los sistemas CISC de alto rendimiento implementan un sistema que convierte dichas instrucciones complejas en varias instrucciones simples del tipo RISC, llamadas generalmente microinstrucciones.

Adicionalmente, la naturaleza compacta de la arquitectura CISC resulta en programas más pequeños, cada vez menos llamados a memoria, lo cual para 1960 fue un gran resultado en cuanto al ahorro en costos de los computadores de la época.
Los CISC pertenecen a la primera corriente de construcción de procesadores, antes del desarrollo de los RISC. Ejemplos de ellos son: Motorola 68000, Zilog Z80 y toda la familia Intel x86 usada en la mayoría de computadores personales del planeta.

- Arquitectura Basada en RISC (*Reduced Instruction Set Computer*)

El objetivo de las máquinas diseñadas con esta arquitectura es permitir la segmentación y el paralelismo en la ejecución de instrucciones y reducir los accesos a memoria. Las máquinas RISC protagonizan la tendencia actual de la construcción de microprocesadores. Power PC, DEC Alpha, MIPS, ARM, son ejemplos de algunos de ellos.

RISC es una filosofía de diseño de CPU para computadora que está a favor de conjuntos de instrucciones pequeños y simples que toman menor tiempo para ejecutarse. El tipo de procesador más comúnmente utilizado en equipos de escritorio, el x86, está basado en CISC en lugar de RISC, aunque las versiones más recientes traducen instrucciones basadas en CISC x86 a instrucciones más simples basadas en RISC para uso interno antes de su ejecución.

La idea fue inspirada debido a que muchas de las características que eran incluidas en los diseños tradicionales de CPU para aumentar la velocidad estaban siendo ignoradas por los programas que eran ejecutados en ellas. Además, la velocidad del procesador en relación con la memoria de la computadora a la que accedía era cada vez más alta. Esto conllevó a la aparición de numerosas técnicas para reducir el procesamiento dentro de la CPU, así como de reducir el número total de accesos a memoria.

Terminología más moderna se refiere a esos diseños como arquitecturas de carga-almacenamiento.

Este tipo de microprocesadores cuenta con las siguientes características:

- Instrucciones de tamaño fijo y presentado en un reducido número de formatos.
- Sólo las instrucciones de carga y almacenamiento acceden a memoria por datos.
- Disponen de muchos registros de propósito general.

1.3.5 Tipo de Procesador MIPS

MIPS es el acrónimo de *Microprocessor without interlocked pipeline stages*, el cual es un tipo de procesador desarrollado por *MIPS Computer Systems Inc*. Este tipo de procesador se basa en la arquitectura RISC. El diseño de MIPS es usado actualmente en la línea de productos de la compañía SGI (*Silicon Integrated Graphics*), y tienen amplia aplicación en dispositivos embebidos.

Un dispositivo embebido es un computador de propósito especial, el cual está completamente encapsulado en un hardware, tiene requerimientos específicos y tareas predefinidas, es un hardware programado, una combinación de hardware y software el cual facilita la producción en masa y la consolidación de variedad de aplicaciones. Lo contrario a los computadores de propósito general que tienen muchas funciones y están cargados de software para muchas tareas.

Algunos dispositivos que operan con MIPS son: Equipos con Windows CE, enrutadores Cisco, enrutadores Linksys, consolas de Nintendo 64, consolas de Sony PlayStation, Sony PlayStation 2 y handheld Sony PSP entre otros. Para finales de los 90 se estimó que uno de cada 3 chips RISC producidos fue basado en el diseño de MIPS. Las primeras implementaciones de MIPS fueron basadas en 32 Bits.

En 1981 un equipo liderado por John L. Hennessy de la Universidad de Stanford comenzó a trabajar en lo que sería el primer procesador MIPS. El concepto básico iba a incrementar el rendimiento a través del uso a fondo de las "*instruction pipelines*", una técnica conocida pero difícil de implementar. Generalmente se denomina "pipeline" cuando se separan las tareas de una instrucción en varios pasos, de esta forma se empieza a trabajar en el siguiente paso de una instrucción incluso antes de que la instrucción precedente se termine. En contraste, los diseños tradicionales esperaban completar la instrucción entera antes de pasar a otra, de tal modo que se dejan áreas de la CPU vacías a medida que el proceso continua.

Existe un simulador libre disponible para los MIPS R2000/R3000 llamado SPIM [21], que sirve para la mayoría de los sistemas operativos (específicamente UNIX o GNU/Linux; OS X Mac; MS WINDOWS 95, 98, NT, 2000, XP; y DOS), el cuál se puede utilizar para aprender el lenguaje de programación ensamblador de los procesadores MIPS.

Algunos casos de la vida diaria en los cuales se encuentran procesadores MIPS son:

- DVD

Dispositivos emergentes como las Grabadoras de Video Digital (DVR, *Digital Video Recorder*), reproductores DVD, estándar, reproductores DVD de alta definición (HD-DVD, High Definition-DVD) y Grabadores de Audio Digital (*DAR, Digital Audio Editing And Recording*), ofrecen la más alta calidad de audio y video desde la perspectiva técnica llamada Sistema en un Chip (SOC, *Sistem On a Chip*) sirviendo a las aplicaciones dando el máximo rendimiento y mínima latencia, mientras opera en ambientes donde el consumo de corriente y disipación de calor son altamente controlados.

- Automóviles

Los diseñadores de aplicaciones telemáticas para automóviles incluyendo productos de inteligencia, información y entretenimiento, son guiados por la demanda de funcionalidad de las comunicaciones y controles más poderosos exactos y robustos. Los fabricantes se ven obligados a mantener los costos bajos y penetrar rápidamente el mercado, para esto se requieren diseños que entreguen el máximo rendimiento y un excepcional bajo consumo de corriente

- Redes

Los *gateways* que entregan voz, datos y video, como los básicos para tecnología DSL, cable-modem, enrutadores y puntos de acceso, demandan rapidez, tamaño reducido y bajo consumo de corriente. Los procesadores MIPS son los líderes en el sector de cable-modem, DSL´s y la tecnología emergente 802.11 inalámbrica.

Idealmente satisfacen las aplicaciones de banda ancha, mediante los procesadores MIPS32 4KE™, los cuales son altamente integrados y de bajo costo.

- Televisión y entretenimiento digital

Los procesadores basados en MIPS se utilizan en los SOC´s para productos de TV dentro de la industria, desde los avanzados grabadores de video digital hasta los "set-top box" Los procesadores de este tipo permiten habilitar nuevos servicios y características de contenido y funcionalidad mientras reducen el costo del sistema, además, contribuyen a convertir los productos de TV convencionales en verdaderos "*media center*" o "*media gateways*", los cuales se convertirán en el centro del hogar en el futuro en lo que a entretenimiento digital se refiere. Para realizar estas labores estos procesadores implementan núcleos altamente programables y ofrecen la posibilidad de añadir funciones definidas por el usuario, con estas capacidades, se pueden incluir funcionalidades adicionales en los productos como seguridad, audio y otros.

- Networking

Empresas, *carriers* (transporte de información) y sistemas de almacenamiento para redes demandan mucha rapidez, más robustez, a precios más económicos. Esta clase de procesadores es solicitada por su alto desempeño, eficiencia en el consumo de corriente, alta integración y manejo de arquitecturas de 32 y 64 bits, estos son diseñados para manejar tráfico con altos volúmenes de datos, voz y video a altas velocidades. Además ofrecen los beneficios de una arquitectura estándar, incluyendo un completo rango de herramientas y software. De hecho, los procesadores MIPS son una de las dos arquitecturas de procesadores soportados por las IOS de Cisco.

- Tarjetas Inteligentes

Soluciones de nueva generación basadas en procesadores MIPS de 32 bits y aplicaciones de seguridad de datos ofrecen rígidos niveles de seguridad y rendimiento en criptografía, bajo consumo de corriente y bajo costo, permitiendo habilitar diversos tipos de criptografía por software en vez de hardware, esto significa mayor flexibilidad, menos material, y en caso de una brecha en el algoritmo de seguridad, una rápida actualización en vez de reemplazo.

Hasta este momento se ha visto el gran potencial que tienen las comunicaciones inalámbricas en el mundo y en Colombia, los ejemplos de redes libres que existen en todo el mundo y aun van en aumento, la idea de un mundo donde circule la libre información y por medios inalámbricos también de manera libre, bajo los conceptos del software de código abierto. Estos conceptos han evolucionado al punto en que los desarrolladores intervienen de manera altruista compartiendo información, estudiando las tecnologías a fondo con el fin de descubrir sus potenciales y debilidades, brindando soluciones a todos los tropiezos, para desarrollar uno o varios productos finales, los cuales a su vez se van actualizando y mejorando cada vez más, de esto se trata la sociedad de la información, en la cual se comparte la ciencia día a día para el crecimiento de la base de conocimiento universal, pretendiendo alcanzar las metas más altas. Una introducción a los procesadores describe la idea de cómo se puede interactuar con ellos de la manera en que se hace hoy en día, mediante el uso de memorias flash; las cuales brindan la posibilidad de escribir nuestro propio código para un firmware que antes no era posible ni siquiera comprender, y menos en Colombia. Es este un punto de partida, que presenta una introducción a las herramientas que se brindan para afrontar el mundo de las telecomunicaciones, mediante la modificación del firmware de un dispositivo de una manera ilimitada, siendo los estudiantes de la Universidad del Cauca participes de este reto. En los próximos capítulos se encuentran disponibles las herramientas, solo hace falta utilizarlas con destreza.

2 CARACTERIZACIÓN DEL ESTÁNDAR 802.11 Y EVALUACIÓN DE APORTES BASADOS EN CÓDIGO ABIERTO

Es cada vez más evidente el gran crecimiento enfocado al desarrollo y aceptación de las comunicaciones móviles y en concreto de las redes de área local inalámbricas *(WLAN, Wireless LANs*). La función principal de este tipo de redes es la de proporcionar conectividad y acceso a las tradicionales redes cableadas (Ethernet, Token Ring, entre otros tipos), pero con la flexibilidad y movilidad que ofrecen las comunicaciones inalámbricas.

El momento decisivo para la consolidación de estos sistemas fue la conclusión del estándar IEEE 802.11 en junio de 1997, de ahí en adelante se continuó con el proceso de maduración y desarrollo del hardware, software y firmware para adicionar nuevas características, tal como se plantea en este trabajo, se muestra la potencialidad y resultados a los que se puede llegar con el correcto uso de las herramientas basadas en las fuentes de firmwares libres.

En el estándar se encuentran las especificaciones tanto físicas como a nivel MAC que deben tenerse en cuenta a la hora de implementar una red de área local inalámbrica, del mismo modo se muestra de manera particular la posibilidad de realizar aportes que aunque no cambian radicalmente la funcionalidad básica del estándar aplicado a las redes inalámbricas, favorecen la mejora de características tal como se muestra en este documento. En la Figura 4 se puede observar las áreas donde se pueden realizar desarrollos que aporten nuevas características a los sistemas electrónicos, en este caso específico del desarrollo va encaminado a la parte intermedia, el “firmware”.

Figura 4. Áreas para el Desarrollo

El estándar 802.11 ha sufrido diferentes variaciones que han llevado a significativas mejoras, algunas de estas son:

- 802.11 Especificación para 1-2 Mbps en la banda de los 2.4 GHz, usando tecnología de salto de frecuencias (*FHSS - Frequency Hopping Spread Spectrum*) o de secuencia directa (*DSSS - Direct Sequence Spread Spectrum*).

- 802.11b Extensión de 802.11 para proporcionar 11Mbps usando DSSS. Wi-Fi (*Wireless Fidelity*) Promulgado por la alianza de compatibilidad (*WECA, Wireless Ethernet Compatibility Alliance*) para certificar productos.

- 802.11b Equipos capacitados para operar con otros fabricantes.

- 802.11a Extensión de 802.11 para proporcionar 54Mbps usando multiplexación por división de frecuencia ortogonal (*OFDM, Orthogonal Frequency Division Multiplexing*) en 5 Ghz.

- 802.11g Extensión de 802.11 para proporcionar 20-54Mbps usando DSSS y OFDM. Es compatible hacia atrás con 802.11b. Tiene mayor alcance y menor Consumo de potencia que 802.11a.

- 802.11e Extensión que mejorará las conexiones de vídeo y voz basada en calidad de servicio (*QoS, Quality of Service*).

Los sistemas WLAN no sustituyen a las tradicionales redes cableadas, más bien las complementan. En este sentido, el objetivo fundamental de las redes WLAN es el de proporcionar las facilidades no disponibles en los sistemas cableados y formar una red integral donde coexistan los dos tipos de sistemas.

2.1 GENERALIDADES SOBRE REDES DE ÁREA LOCAL INALÁMBRICAS

2.1.1 Definición de Red de Área Local Inalámbrica

Una red de área local inalámbrica puede definirse como a una red de alcance local, alrededor de 100 metros, que tiene como medio de transmisión el aire. Debido a su medio de transmisión, tiene desventajas a la hora de realizar trasmisiones principalmente en distancias variables debido a los ajustes automáticos de potencia que deben hacer los dispositivos, ya que se introducen diferentes factores al medio como son las interferencias y los obstáculos. Los factores nocivos para la comunicación que se introducen sobre el medio hacen que este no ofrezca un alto valor de confiabilidad, razón por la cual los aportes encaminados a proporcionar más estabilidad a las transmisiones en el medio y mejor control sobre la transmisión y recepción de información juegan un papel importante en el desarrollo tanto de hardware como de software. Implementaciones como las ofrecidas por este tipo de trabajos condicionan y dan pie a mejoras sustanciales tanto en el rendimiento como en la estabilidad de este tipo de redes.

El atractivo fundamental de este tipo de redes es la facilidad de instalación y el ahorro que supone la eliminación del medio de transmisión cableado. Aún así, debido a que sus prestaciones son menores en lo referente a la velocidad de transmisión, la cual se sitúa entre los 2 y los 108 Mbps frente a los 10 Mbps y 10000 Mbps ofrecidos por una red cableada, las redes inalámbricas son la alternativa ideal para hacer llegar una red tradicional a lugares donde el cableado no lo permite, en general las WLAN se utilizarán como un complemento de las redes fijas.

2.1.2 Aplicaciones de los Sistemas de Red Inalámbrica

Entre las aplicaciones más sobresalientes de las redes de área local que se puede encontrar actualmente se tiene:

- Implementación de redes de área local en edificios históricos, de difícil acceso y en general en entornos donde la solución cableada no es viable.

- Posibilidad de reconfigurar la topología de la red sin añadir costos adicionales. Esta solución es muy típica en entornos cambiantes que necesitan una estructura de red flexible que se adapte a estos cambios.

- Redes locales para situaciones de emergencia o congestión de la red cableada. Estas redes permiten el acceso a la información mientras el usuario se encuentra en movimiento. Habitualmente esta solución es requerida en hospitales, fábricas, almacenes.

- Creación de grupos de trabajo eventuales y reuniones. En estos casos no vale la pena instalar una red cableada, es mejor implementar una red de área local inalámbrica para un corto tiempo.

- En ambientes industriales con severas condiciones ambientales este tipo de redes sirve para interconectar diferentes dispositivos y máquinas.

- Interconexión de redes de área local que se encuentran en lugares físicos distintos. Por ejemplo, se puede utilizar una red de área local inalámbrica para interconectar dos o más redes de áreas locales cableadas situadas en dos edificios distintos.

2.2 ARQUITECTURA Y TECNOLOGÍAS DE MODULACIÓN

2.2.1 Arquitectura de Capas 802.11

La capa física proporciona una serie de servicios a la capa MAC o capa de acceso al medio. Diferentes tecnologías de capa física se definen para transmitir por el medio inalámbrico.

La capa física de servicios consiste en dos protocolos:

1. Una función de convergencia de capa física, que adapta las capacidades del sistema físico dependiente del medio (*PMD, Physical Medium Dependent*). Esta función es implementada por el protocolo de convergencia de capa física (*PLCP, Physical Layer Convergenve Protocol*), que define una forma de mapear unidades de datos MAC (*MPDU, MAC Protocol Data Unit*) en un formato de tramas susceptibles de ser transmitidas o recibidas entre diferentes estaciones o STASs a través de la capa *PMD*.

2. Un sistema *PMD*, cuya función define las características, un medio de transmitir y recibir a través de un medio sin cables entre dos o más *STAs*. La comunicación entre *MACs* de

diferentes estaciones se realizará a través de la capa física mediante de una serie de puntos de acceso al servicio, donde la capa *MAC* invocará las primitivas de servicio.

Además de estas capas, se puede distinguir la capa de gestión. En esta capa se observa la estructura base de información de administración (*MIB, Management Information Base*) que contienen por definición las variables de gestión, los atributos, las acciones y las notificaciones requeridas para manejar una estación. Consiste en un conjunto de variables donde se puede especificar o contener el estado y la configuración de las comunicaciones de una estación.

2.2.2 Tecnologías Utilizadas en las Redes Inalámbricas

El estándar 802.11 define varios métodos y tecnologías de multiplexación para implementaciones en redes inalámbricas como la técnica de espectro ensanchado de secuencia directa (*DSSS, Direct Sequence Spread Spectrum*), espectro ensanchado de saltos de frecuencia (*FHSS, Frecuency Hopping Spread Spectrum*) y la multiplexación por división en frecuencias ortogonales (*OFDM, Orthogonal Frequency Division Multiplexing*).

2.2.2.1 Tecnología de Espectro Ensanchado de Secuencia Directa *(DSSS)*

Esta técnica consiste en la generación de un patrón de bits redundante llamado señal de chip para cada uno de los bits que componen la señal de información y la posterior modulación de la señal resultante mediante una portadora de RF. En recepción es necesario realizar el proceso inverso para obtener la señal de información original.

La secuencia de bits utilizada para modular cada uno de los bits de información es la llamada secuencia de *Barker* y tiene la siguiente forma:

```
+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1
```

DSSS tiene definidos dos tipos de modulación para aplicar a la señal de información una vez se sobrepone la señal de chip, tal como especifica el estándar IEEE 802.11: la modulación por desplazamiento de fase binaria diferencial (DB*PSK, Differential Binary Phase Shift Keying*) y la modulación por desplazamiento de fase en cuadratura diferencial (DQ*PSK, Differential Quadrature Phase Shift Keying*) proporcionando unas velocidades de transferencia de 1 y 2 Mbps respectivamente.

En el caso de Estados Unidos, América Latina y de Europa la tecnología de espectro ensanchado por secuencia directa, DSSS, opera en el rango que va desde los 2.4 GHz hasta los 2.4835 GHz, con un ancho de banda total disponible de 83.5 MHz.

Este ancho de banda total se divide en un total de 14 canales con un ancho de banda por canal de 5 MHz de los cuales cada país utiliza un subconjunto de los mismos según las normas reguladoras para cada caso particular. Muchos países Europeos utilizan los canales 10 y 11 ubicados en una frecuencia central de 2.457 GHz y 2.462 GHz respectivamente. En países Latinoamericanos como Colombia, son utilizados los primeros 11 canales.

En la Figura 5 se muestra la distribución de los 3 canales independientes.

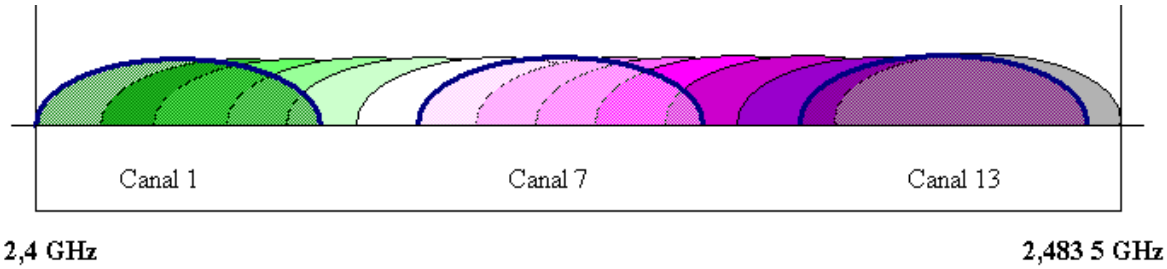

Figura 5. Distribución de Canales Independientes

En topologías de red que contengan varias celdas, ya sean solapadas o adyacentes, los canales pueden operar simultáneamente sin apreciarse interferencias en el sistema si la separación entre las frecuencias centrales es como mínimo de 30 Mhz. Esto significa que entre los 83.5 MHz de ancho de banda total disponible se puede obtener un total de 3 canales independientes que pueden operar simultáneamente sin que aparezcan interferencias en un canal procedente de los otros dos canales. Esta independencia entre canales permite aumentar la capacidad del sistema de forma lineal con el número de puntos de acceso operando en un canal que no se esté utilizando y hasta un máximo de tres canales.

- Tecnología de Espectro Ensanchado de Salto de Frecuencia (FHSS)

La Tecnología de Espectro Ensanchado de Salto de Frecuencia consiste en transmitir una parte de la información en una determinada frecuencia durante un intervalo de tiempo llamado *dwell time*, inferior a 400 ms. Pasado este tiempo se cambia la frecuencia de emisión y se sigue transmitiendo a otra frecuencia. De esta manera cada tramo de información se va transmitiendo en una frecuencia distinta durante un intervalo muy corto de tiempo.

Cada una de las transmisiones a una frecuencia concreta se realiza utilizando una portadora de banda estrecha que va cambiando (saltando) a lo largo del tiempo. Este procedimiento equivale a realizar una partición de la información en el dominio temporal.

El orden en los saltos de frecuencia que el emisor debe realizar viene determinado según una secuencia pseudo-aleatoria que se encuentra definida en unas tablas que tanto el emisor como el receptor deben conocer. La ventaja de estos sistemas frente a los sistemas DSSS es que con esta tecnología se puede tener más de un punto de acceso en la misma zona geográfica sin que existan interferencias si se cumple que dos comunicaciones distintas no utilizan la misma frecuencia portadora en un mismo instante de tiempo.

Si se mantiene una correcta sincronización de estos saltos entre los dos extremos de la comunicación, el efecto global es que aunque se va cambiando de canal físico, con el tiempo se mantiene un único canal lógico a través del cual se desarrolla la comunicación.

Para un usuario externo a la comunicación, la recepción de una señal FHSS equivale a la recepción de ruido impulsivo de corta duración. El estándar IEEE 802.11 describe esta tecnología mediante la modulación en frecuencia (*FSK, Frequency Shift Keying*) y con una velocidad de transferencia de 1 Mbps ampliable a 4 Mbps bajo condiciones de operación óptimas. En Colombia existen empresas con equipos trabajando en este tipo de tecnología. Regionalmente empresas como ETB con sus equipos Airspan Wipll utilizando un rango de 28 Mhz, separados en 2 tramos de 14 Mhz, uno para *upstream* y otro para *downstream*, utilizan el FHSS como técnica de modulación

para ofrecer a sus usuarios servicios de banda ancha residencial y corporativa de manera masiva. Coldecón S.A. es otra empresa que utiliza este tipo de equipos en frecuencias libres para entregar a usuarios servicios de banda ancha.

En la Figura 6 se pueden observar los tipos de equipos montados por algunas empresas colombianas que utilizan la técnica de FHSS. El equipo situado al lado superior izquierdo se denomina estación radio base (*BSR, Base Station Radio*), que es un equipo que se ubica en la estación base de los nodos de acceso, el siguiente es el equipo para el subscriptor ubicado en las instalaciones del cliente (*SPR, Subscriber Premises Radio*) de manera externa (*outdoor*). El equipo en la parte superior derecha es un equipo para el subscriptor interno (*IDR, Indoor Data Radio*), y el equipo de la parte inferior de la figura es el adaptador de la unidad del subscriptor (*SDA, Subscriber Data Adapters*), que es el equipo encargado de alimentar eléctricamente a los equipos *oudoor* y también brinda la posibilidad de hacer el etiquetamiento de *VLANs*.

Figura 6. Equipos Airspan Wipll FHSS

- Tecnología de Multiplexación por División en Frecuencias Ortogonales (OFDM)

La Multiplexación por División en Frecuencias Ortogonales (OFDM), también a veces llamada Modulación Discreta por Multi-tono (*DMT, Discreet Multitone Modulation*), es una técnica compleja de modulación para la transmisión, basada en el concepto de Multiplexación por División de Frecuencia (*FDM, Frequency Division Multiplex*) donde cada canal de frecuencia se modula mediante una técnica de modulación más simple. En OFDM las frecuencias y la modulación de FDM se ordenan para ser ortogonales, intentando eliminar la interferencia entre los canales.

Este tipo de modulación es conocida hace más de cuatro décadas, sin embargo, es hecha popular hoy en día por el costo más bajo y la disponibilidad de la señal numérica que procesan componentes.

Una de las principales características de OFDM es que las modulaciones *low-rate* (modulaciones con los símbolos relativamente largos comparados a las características del tiempo del canal) son menos sensibles a la multi-dirección, es decir, es mejor enviar un número bajo de ondas cortas en paralelo que enviando una forma de onda larga.

Esto es exactamente lo que hace OFDM: divide el espectro de la frecuencia en sub-bandas pequeñas.

- Comparación con FDM

En FDM, las señales múltiples se envían al mismo tiempo, pero en diversas frecuencias. Generalmente se conoce sobre FDM usado para radio y televisión: normalmente, cada estación difunde en una banda de frecuencia particular (gama de frecuencias) o el canal.

- OFDM toma este concepto más a fondo: En OFDM, un solo transmisor transmite en diversas frecuencias ortogonales (es decir las frecuencias que son independientes con respecto a la relación relativa de la fase entre las frecuencias). Además, porque las frecuencias son cercanas, cada uno tiene solamente sitio para una señal de banda estrecha.

- Esta técnica de la modulación en conjunto con el uso de las técnicas avanzadas de modulación en cada componente, da lugar a una señal con alta resistencia a interferencia.

- Características

Una señal portadora de OFDM es la suma de un número de sub-portadoras ortogonales, cada una contiene los datos de banda base sobre cada sub-portadora, la cual es modulada independientemente y comúnmente usando un tipo de modulación de amplitud de cuadratura (*QAM, Quadrature Amplitude Modulation*) o de fase (*PSK, Phase Shift Keying*). Esta señal compuesta de banda base se utiliza típicamente para modular una portadora principal de RF.

- Ventajas

Las ventajas de usar OFDM son varias, incluyendo alta eficiencia del espectro, resistencia contra interferencia multidireccional y facilidad de filtrar el ruido, (si una gama particular de frecuencias sufre de interferencia, las portadoras dentro de esa gama pueden ser adaptadas para funcionar más lentamente). Algunas formas de *DSL* (*Digital Subscriber Line*) utilizan esta característica en tiempo real, para asignar el ancho de banda a un canal que necesite más velocidad de transmisión de datos.

Una ventaja importante de usar sub-portadoras múltiples es que cada portadora funciona en una tasa de bits relativamente baja, la duración de cada símbolo es relativamente larga. Si se envía, por ejemplo, un millón de fragmentos por segundo sobre un solo canal, la duración de cada fragmento debe ser de microsegundos. Esto crea inconvenientes en la sincronización y el retiro de interferencia multidireccional. Si el mismo número de fragmentos por segundo se separan entre N sub-portadoras, la duración de cada fragmento puede ser más larga por un factor de N, y la disminución de los inconvenientes de la sincronización y de la sensibilidad multidireccional se reducen notablemente. Para las comunicaciones móviles, el efecto de *Doppler* sobre la

sincronización de la señal es otra exigencia que causa dificultades para algunos esquemas de la modulación.

COFDM también tiene generalmente un espectro casi “blanco”, mostrando características favorables de interferencia electromagnética con respecto a otras señales.

Los sistemas que utilizan COFDM usan algunas de las sub-portadoras para llevar las señales experimentales que se utilizan para la sincronización de la frecuencia. (La pérdida de sincronización causa errores en los datos descifrados).

En una amplia difusión, los receptores pueden beneficiarse de recibir señales de varios transmisores simultáneamente dispersos, puesto que los transmisores interferirán solamente en forma destructiva con otro en un número limitado de sub-portadoras, mientras que en general reforzarán realmente la cobertura sobre un área. Esto es muy beneficioso en muchos países, pues permite la operación de las redes nacionales que utilizan Radio Frecuencia, y evita la réplica de las mismas en diversas frecuencias portadoras que es necesario con FM u otras formas de difusión de radio. También, porque el índice binario (cociente del número de los paquetes transferidos entre dos dispositivos por segundo) se retrasa con eficacia en cada sub-portadora, los efectos de la “imagen secundaria” se reducen mucho.

- Desventajas de OFDM

OFDM sufre de variaciones por canal en el tiempo, o de presencia de una compensación de la frecuencia portadora. Esto es debido a que las sub-portadoras están a pequeños espacios cerca en frecuencia. La sincronización imperfecta de la frecuencia causa una pérdida en la ortogonalidad de la sub-portadora que degrada seriamente funcionamiento.

La señal es la suma de una gran cantidad de sub-portadoras, ésta tiende a tener un alto cociente pico-a-medio de energía (*PAPR, Peak to Average Power Ratio*). También, es necesario reducir al mínimo la intermodulación entre las sub-portadoras, que levantarían el nivel de piso del ruido y del canal. Por esta razón el trazado de circuito debe ser muy lineal o la frecuencia ser muy exacta.

- Principal Uso

OFDM se utiliza en muchos sistemas de comunicaciones, por ejemplo: ADSL, los estándares inalámbricos LAN 802.11a y 802.11g, difusión audio de Digital, WiMAX, y PLC. La tecnología de OFDM se utiliza además en radioenlaces punto a punto (*PtP, Point to Point*) y punto a multipunto (*PtMP, Point to Multi Point*).

- Uso en ADSL

OFDM se utiliza en las conexiones del ADSL que siguen el estándar de G.DMT (ITU G.992.1).

COFDM no interfiere fácilmente con otras señales, esta es la razón principal por la cual se utiliza con frecuencia en aplicaciones tales como módems del ADSL, en los cuales los alambres de cobre existentes se utilizan para alcanzar conexiones de datos de alta velocidad. La carencia de interferencia significa que ningún alambre necesita ser sustituido (si no sería más barato sustituirlos por fibra). Sin embargo, el DSL no se puede utilizar en cada par de cobre, la interferencia puede llegar a ser significativa si más del 25% de las líneas telefónicas en una central se utilizan para el DSL.

- Uso en Alianza del Powerline de HomePlug

OFDM es utilizado por los dispositivos de HomePlug para extender conexiones de ethernet a otros lugares en un hogar a través del cableado de la energía. La modulación adaptativa es particularmente importante con un canal ruidoso tal como el cableado eléctrico.

- Redes de área local de la radio (LAN) y redes del área metropolitana (MAN)

OFDM también se utiliza en entornos inalámbricos LAN y MAN, incluyendo IEEE 802.11a/g y el alternativo europeo en desuso HIPERLAN/2 y WiMAX.

Algunos radio aficionados han enganchado sobre el equipo comercial ADSL transmisores-receptores para radiar sus señales, y son capaces de cambiar las bandas usadas a las radiofrecuencias que el usuario ha licenciado.

En la Tabla 2, se observa un listado de las ocho velocidades de datos especificadas en la capa física de 802.11a. Se utilizan cuatro diversos esquemas de la modulación: B*PSK*, 4-QAM, 16-QAM, y 64-QAM. Cada esquema de ejecución más alto en modulación requiere una condición mejor del canal para la transmisión correcta. Estos esquemas de modulación se juntan con varios esquemas de codificación de corrección de error para dar una multiplicidad del número de los bits de datos por funcionamiento del símbolo (Ndbps).

Tabla 2. Información de Velocidades

Velocidad de Datos (Mbit/s)	**Modulación**	**Tarifa de la codificación**	**Ndbps**	**duración de la transferencia de 1472 octetos (µs)**
6	B*PSK*	el 1/2	23	2012
9	B*PSK*	3/4	36	1344
12	4-QAM	el 1/2	48	1008
18	4-QAM	3/4	72	672
24	16-QAM	el 1/2	96	504
36	16-QAM	3/4	144	336
48	64-QAM	2/3	192	252
54	64-QAM	3/4	216	224

En Colombia, empresas como Coldecón S.A. – Telesat S.A., utilizan equipos de radio de la marca Alvarion en su versión Breeze Access, que operan bajo la modulación OFDM en frecuencias no licenciadas de 5.2 Ghz, 5.8Ghz y 5.4Ghz, para ofrecer a los usuarios accesos a Internet de Banda Ancha.

En la Figura 7 se muestran algunos de los equipos Alvarion que trabajan en OFDM y que son instalados por las empresas Colombianas que usan esta tecnología.

Las dos unidades de color negro se denominan unidades *indoor* (*IDU, Indoor Unit*), las cuales son las que alimentan de energía la unidad *oudoor* con 55V y proporcionan el conector RJ45 para el equipo de borde del usuario. La unidad en forma de rombo se denomina unidad del subscriptor (*SU, Subscriber Unit*) que consta de una antena y un chasis acoplado directamente para evitar perdidas por cables RF, este equipo es un *bridge* inalámbrico y se ubica en las instalaciones del

usuario . Por ultimo, se encuentra la unidad de acceso (*AU, Access Unit*), la cual se ubica en el nodo central y soporta hasta 512 SUs.

Estos equipos están basados en el estándar 802.11a, pero no son compatibles con él. Alvarion introdujo en sus equipos mejoras significativas sobre el estándar como una alta inmunidad al ruido, un algoritmo que ajusta los parámetros de potencia en las unidades remotas llamado ATPC, la presencia de 8 niveles de modulación que se ajustan mediante un algoritmo y el soporte de hasta 512 *SU* por sector.

Figura 7. Equipos Alvarion OFDM

Estos equipos tienen la finalidad de establecer una red de área metropolitana de acceso, para ofrecer servicios de banda ancha a sus subscriptores.

2.2.3 Múltiple Entrada Múltiple Salida (MIMO)

Las redes de área local inalámbricas *(WLAN, Wireless Local Area Network*) ya son parte del entorno de las empresas y organizaciones. Los usuarios conocen sus innumerables beneficios, pero también saben de las desventajas de este tipo de redes. Los principales inconvenientes radican principalmente en la pobre capacidad de ancho de banda y la limitada cobertura de la señal.

El cambio repentino del nivel de fuerza la señal es un problema común que ocurre en los clientes inalámbricos con IEEE 802.11a/b/g (un Portátil, una PDA u otro dispositivo con una tarjeta inalámbrica). Si un dispositivo se mueve de un lugar a otro, la fuerza de la señal que se recibe del punto de acceso fluctúa del máximo al mínimo, o viceversa. Ocasionando, en la mayoría de casos, que el enlace se vea interrumpido entre el punto de acceso y el dispositivo cliente.

MIMO es una tecnología de radiocomunicaciones que se refiere a enlaces de radio con múltiples antenas en el lado del transmisor y del receptor. Debido a las múltiples antenas, la diversidad espacial puede ser explotada para mejorar el desempeño del enlace inalámbrico, haciendo la señal más fuerte, confiable y transmisiones más rápidas. En la Figura 8 se muestra el funcionamiento básico de MIMO.

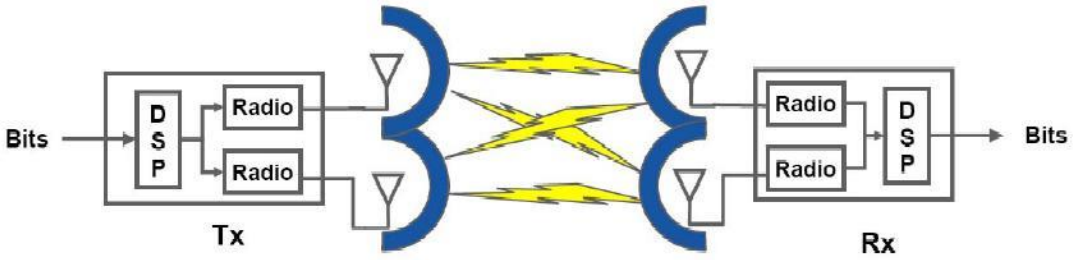

Figura 8. Funcionamiento de MIMO

Entre las características de MIMO está el incremento de la cobertura hasta 8 veces y el aumento de velocidad es hasta 6 veces la de las actuales redes IEEE 802.11g.

Aunque en la actualidad MIMO es una tecnología no estandarizada, ya está considerada en el estándar 802.11n de la IEEE, el cual piensa liberarse a finales de este año o principios del 2007. Los consumidores ven a MIMO como una nueva clase de productos inalámbricos categorizados como “pre-n”, debido a que se anticipan al estándar 802.11n de IEEE.

Visto de otra manera, ya existen varios dispositivos hardware en el mercado, principalmente Puntos de Acceso o Enrutadores Inalámbricos con más de una antena, de esta forma es posible realizar un desarrollo más especializado en el campo de las antenas con control por software logrando de cierto modo que se puedan emitir y recibir señales en diferentes tipos y frecuencias, buscando realizar un desarrollo local y aportes a este nuevo estándar.

En la actualidad, dos son los grupos principales que están contendiendo y sometiendo propuestas para el estándar IEEE 802.11n. Estos consorcios son TGn Sync y WWiSE. Ambas propuestas son muy similares, pero difieren en varios detalles. TGn Sync (Task Group 'n' synchronization)[22], es un consorcio que incluye compañías como Agere, Atheros, Intel, Sony, Nortel, Samsung, Qualcomm, Philips, Panasonic, entre otros. TGn Sync propone expandir el tamaño del canal de 20 MHz a 40 MHz, permitiendo así un caudal eficaz máximo de 315 Mbps con multi-canalización por división espacial MIMO.

Este cambio podría disminuir el número de canales disponibles de 22 a 11 en la banda de 5 GHz. WWiSE (*World-Wide Spectrum Efficiency*) es un consorcio que está compuesto por compañías como Airgo Networks, Broadcom, Motorota, Nokia, Conexant Systems, STMicroelectronics, Texas Instruments, France Telecom, NTT, entre otros. WWiSE propone utilizar el canal existente de 20 MHz, dos antenas MIMO, y cambios en la capa de control de acceso al medio (*MAC*, *Media Control Access*) para permitir una capacidad de canal de 135 Mbps.

En la figura número 9, se puede ver claramente las capacidades que puede ofrecer MIMO en el funcionamiento y mejor uso de las redes inalámbricas. Este tipo de desarrollo se basa principalmente en la maduración y nuevos sistemas hardware debido a que se necesitan 2 o más radios sincronizados para tal fin.

El mercado de productos y chips MIMO en lo que respecta a productos en el mercado, actualmente Airgo Networks [23] fabrica chips de una tecnología propietaria conocida como True MIMO, liderando el mercado con productos “pre-n”.

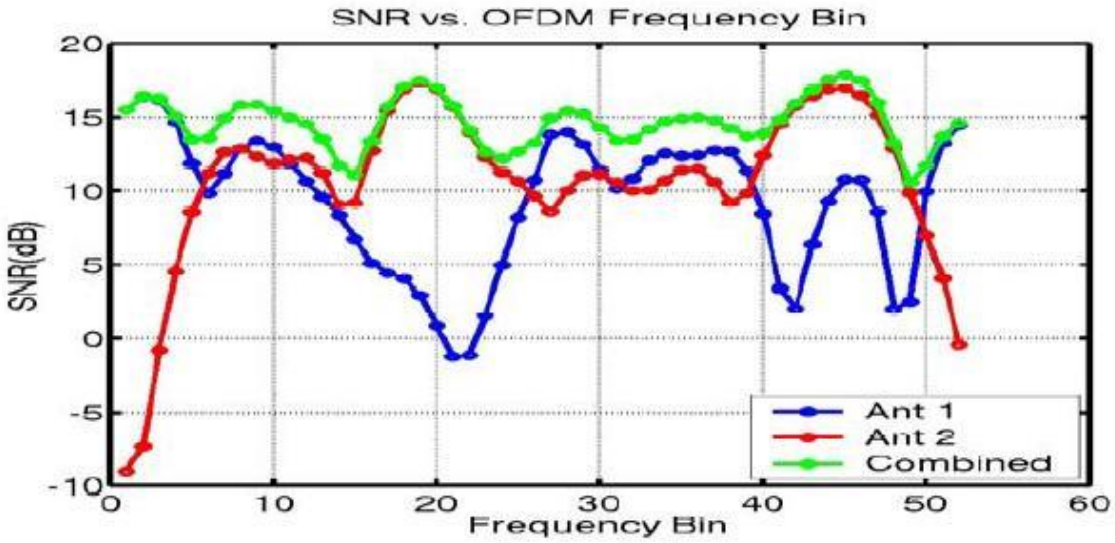

Figura 9. Relación SNR y Frecuencia

Otros fabricantes de chips MIMO son Atheros, Video54y Athena, entre los fabricantes de equipos para redes inalámbricas con tecnología MIMO se encuentran Belkin, Linksys, Samsung, Netgear, entre muchos otros. La mayoría de ellos utiliza el chip True MIMO de Airgo Networks.

Netgear, otro fabricante de equipos, ha lanzado productos bajo el nombre de "*RangeMax*" el cual utiliza una mezcla de la tecnología Super-G de Atheros y BeamFlex [24] de Video54, pero también producen una línea de productos que utiliza True MIMO de Airgo. Otros fabricantes como D-Link han optado por la tecnología MIMO conocida como channel-bonding spectrum-hogging [25].

En las Figuras 10 y 11 se muestran los equipos más representativos que utilizan MIMO, en este caso un punto de acceso inalámbrico y una tarjeta de red PCI.

Figura 10. Punto de Acceso MIMO

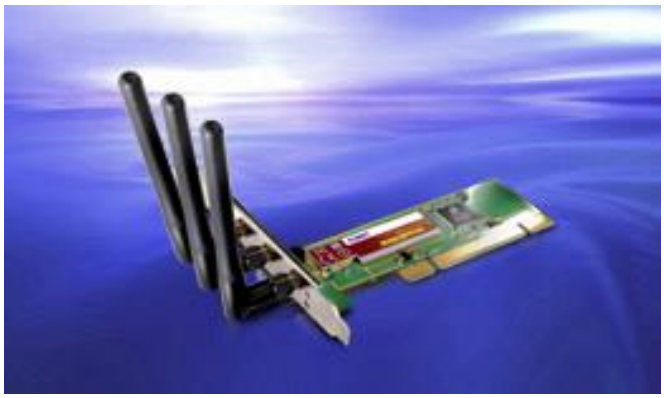

Figura 11. Tarjeta PCI MIMO

- Funcionamiento de MIMO

La propagación multi-trayectoria es una característica de todos los ambientes de comunicación inalámbricos. Usualmente existe una ruta o trayectoria principal desde un transmisor en el punto "A" al receptor en el punto "B", pero ocurre que algunas de las señales transmitidas toman otras trayectorias, irrumpiendo objetos, la tierra o capas de la atmósfera. Aquellas señales con trayectorias menos directas, llegan al receptor desfasadas y atenuadas.

Una estrategia para negociar con señales débiles multi-trayectoria es simplemente ignorarlas, pero las señales multi-trayectoria con potencia alta pueden ser demasiado fuertes como para ignorarse y son capaces de degradar el desempeño de los equipos inalámbricos basados en los estándares actuales. MIMO toma ventaja de la propagación multi-trayectorias para incrementar el caudal eficaz, cobertura y fiabilidad de las señales.

Más allá de combatir las señales multi-trayectoria, MIMO pone señales multi-trayectoria a trabajar acarreando y concentrando más información. Cada una de estas señales son moduladas y transmitidas por una serie antenas al mismo tiempo y en el mismo canal de frecuencia. El empleo de múltiples formas de onda constituye un nuevo tipo de radio comunicación, la cual es el único medio para mejorar los tres parámetros básicos del desempeño del enlace (cobertura, velocidad y calidad de la señal).

MIMO tiene la habilidad de multiplicar la capacidad, la cual es un sinónimo de velocidad. Una medida para medir la capacidad inalámbrica es conocida como la eficiencia espectral (*EE, Espectral Efficiency*). La EE es el número de unidades de información por unidad de tiempo por unidad de ancho de banda, denotada usualmente como bps/Hz (bits por segundo sobre Hertz). Si se transmiten múltiples señales, conteniendo diferentes ráfagas de información, sobre el mismo canal, se puede doblar o triplicar la eficiencia espectral. Más eficiencia espectral da como resultado más velocidad de información, más cobertura, más usuarios y una mejor calidad de la señal.

Los transmisores MIMO aprovechan las cualidades de OFDM. Como se explicó anteriormente OFDM es una técnica de modulación digital que divide la señal en varios canales de banda angosta a diferentes frecuencias. Dentro de las bondades de OFDM incluyen: gran eficiencia espectral, resistencia en contra de interferencia por multi-trayectorias, filtrado de ruido externo.

Los principales bloques de procesamiento de un transmisor utilizando MIMO incluyen dos antenas de transmisión con dos moduladores OFDM idénticos, convertidores analógico-digital, moduladores analógicos de radio frecuencia, amplificadores de potencia y antenas con patrón omnidireccional.

Un transmisor MIMO con dos antenas es un modulador digital que alimenta dos cadenas analógicas idénticas de convertidores analógico-digital, moduladores analógicos de radio frecuencia y dos antenas idénticas omnidireccionales.

De esta manera, la transmisión MIMO-OFDM es exactamente la misma, como si dos transmisiones OFDM simultáneas ocurrieran en el mismo canal, pero con diferentes datos digitales.

2.3 NIVEL DE ACCESO AL MEDIO (MAC)

Los diferentes métodos de acceso de IEEE802 están diseñados según el modelo OSI y se encuentran ubicados en el nivel físico y en la parte inferior del nivel de enlace o subnivel MAC. Además, la capa de gestión MAC controla aspectos como la sincronización y los algoritmos del sistema de distribución, que se define como el conjunto de servicios que precisa o propone el modo de infraestructura.

La arquitectura MAC del estándar 802.11 se compone de dos funcionalidades básicas: la función de coordinación distribuida (*DCF, Distributed Coordination Function*) y la función de coordinación puntual (*PCF, Point Coordination Function*).

2.3.1 Función de Coordinación Distribuida (*DCF)*

Se define función de coordinación como la funcionalidad que se determina, dentro de un conjunto básico de servicios (*BSS, Basic Service Set*), cuándo una estación puede transmitir y/o recibir unidades de datos de protocolo a nivel MAC a través del medio inalámbrico. En el nivel inferior del subnivel MAC se encuentra la función de coordinación distribuida y su funcionamiento se basa en técnicas de acceso aleatorias de contienda por el medio.

El tráfico que se transmite bajo esta funcionalidad es de carácter asíncrono ya que estas técnicas de contienda introducen retardos aleatorios y no predecibles no tolerados por los servicios sincrónicos.

Las características de DCF se pueden resumir en estos puntos:

- Utiliza MACA (*Multiple Access with Collision Avoidance*) (*CSMA/CA con RTS/CTS*) como protocolo de acceso al medio.
- Es necesario reconocimiento ACKs, provocando retransmisiones si no se recibe.
- Usa campo Duration/ID que contiene el tiempo de reserva para transmisión y ACK. Esto quiere decir que todos los nodos escuchan el medio para saber cuando el canal volverá a quedar libre.
- Implementa fragmentación de datos.
- Concede prioridad a tramas mediante el espaciado entre tramas (*IFS, Inter Frame Spacing*).
- Soporta *Broadcast* y Multicast sin ACKs

2.3.2 Función de Coordinación Puntual (*PCF)*

Por encima de la funcionalidad DCF se sitúa la función de coordinación puntual, PCF, asociada a las transmisiones libres de contienda que utilizan técnicas de acceso no aleatorias. El estándar IEEE 802.11, en concreto, define una técnica de interrogación circular desde el punto de acceso para este nivel. Esta funcionalidad está pensada para servicios de tipo sincrónico que no toleran retardos aleatorios en el acceso al medio.

Estos dos métodos de acceso pueden operar conjuntamente en una misma celda o conjunto básico de servicios dentro de una estructura llamada super-trama. Un parte de esta super-trama se asigna al periodo de contienda permitiendo al subconjunto de estaciones que lo requieran transmitir bajo mecanismos aleatorios. Una vez finaliza este periodo el punto de acceso toma el medio y se inicia un periodo libre de contienda en el que pueden transmitir el resto de estaciones de la celda que utilizan técnicas no aleatorias.

El uso de PCF es totalmente compatible con el modo DCF y el funcionamiento es transparente para las máquinas cliente. De esta manera, un cliente se asociará de modo que pueda actuar en el periodo libre e Contención (*CFP, Content Free Period*), declarándose como de consulta continua (CF-Pollable[3]), o por el contrario, se situará su vector de asignación de red (*NAV, Network Allocation Vector*) según las indicaciones del punto de coordinación(*PC, Point Coordination*). Se observa la grafica de los tiempos utilizados pro el *NAV* en la Figura 12

Existe un nodo organizador o director, llamado punto de coordinación o PC. Este nodo tomará el control mediante el método *PIFS* (*PCF interframe space*), y enviará un *CF-Poll* a cada estación que pueda transmitir en *CFP*, concediéndole poder transmitir una trama *MPDU*. El *PC* mantendrá una lista a la que le hace Poll donde tendrá todos los datos de las estaciones que se han asociado al modo *CF-Pollable*. La concesión de transmisiones será por riguroso listado y no permitirá que se envíen dos tramas hasta que la lista se haya completado.

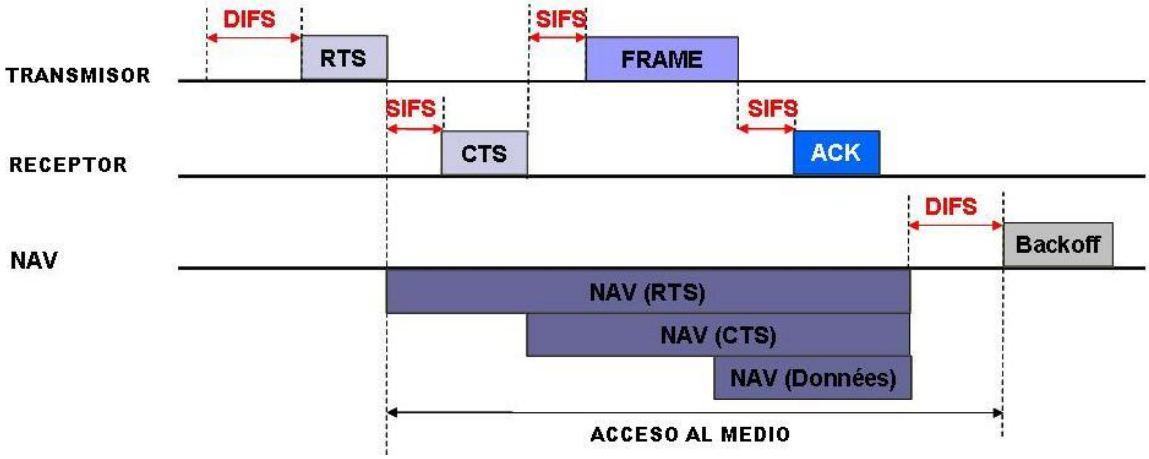

Figura 12. Vector de Asignación de Red

El nodo utilizará una trama para la configuración de la super-trama, llamada *Beacon*, donde establecerá una *CFRate* o tasa de periodos de contienda. Pese a que el periodo de contienda se puede retrasar por estar el medio ocupado, la tasa se mantendrá en el siguiente periodo con medio libre.

[3] Estación capaz de responder a las consultas

La transmisión de CF-*Polls* espera un tiempo *SIFS* (*Short interframe space*). Si una estación no aprovecha su *CF-Poll* se transmite a la siguiente en el listado para hacer *Poll*.

Las estaciones que no usen el *CF*, situarán su *NAV* al valor del final del *CF* y luego lo reiniciarán para poder modificarlo en el periodo de contienda en igualdad de condiciones.

Un problema importante en el solapamiento de redes inalámbricas ocurrirá cuando varios sistemas con coordinación puntual compartan una tasa *CFRate* semejante. Una solución es establecer un periodo de contienda entre *PCs* para ganar el medio esperando un tiempo *DIFS*+ *BackOff*[4] (*1-CWmin*). Sin embargo, se puede encontrar con mayores dificultades que exigirían un estudio diferenciado.

2.3.3 Espaciado entre Tramas *IFS*

El tiempo de intervalo entre tramas se llama *IFS*. Durante este periodo mínimo, una estación estará escuchando el medio antes de transmitir. Se definen cuatro espaciados para dar prioridad de acceso al medio inalámbrico, como se observa en la Figura 13.

- *SIFS* (*Short IFS*). Este es el periodo más corto. Se utiliza fundamentalmente para transmitir los reconocimientos. También es utilizado para transmitir cada uno de los fragmentos de una trama. Por último, es usado por el *PC* o punto de acceso para enviar testigo a estaciones que quieran transmitir datos sincrónicos.

- *PIFS* (*PCF*). Es utilizado por *STAs* para ganar prioridad de acceso en los periodos libres de contienda. Lo utiliza el *PC* para ganar la contienda normal, que se produce al esperar *DIFS*.

- *DIFS* (*DCF*). Es el tiempo de espera habitual en las contiendas con mecanismo *MACA*. Se utiliza para el envío de tramas *MAC MPDUs* y tramas de gestión *MMPDUs*.

- *EIFS* (*Extended IFS*). Controla la espera en los casos en los que se detecta la llegada de una trama errónea. Espera un tiempo suficiente para que le vuelvan a enviar la trama u otra solución.

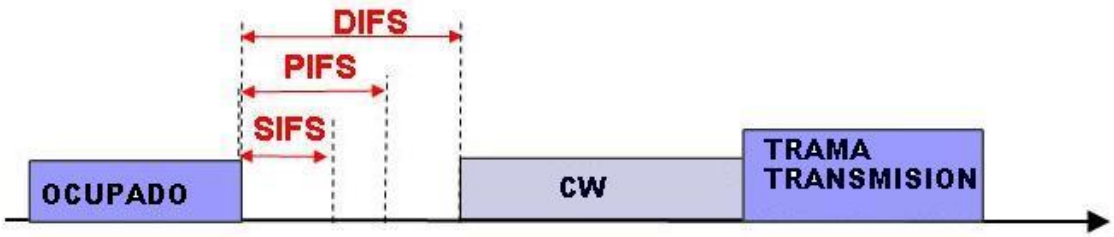

Figura 13. Espacio Intertrama

[4] Algoritmo que calcula un número con poca probabilidad de repetirse dentro de una secuencia.

2.3.4 Protocolo de Acceso al Medio *CSMA/CA* y *MACA*

El algoritmo básico de acceso a este nivel es muy similar al implementado en el estándar IEEE 802.3 y es llamado *CSMA/CA*. Este algoritmo funciona de la siguiente manera:

- Antes de transmitir información una estación debe escuchar el medio, o canal inalámbrico, para determinar su estado (libre / ocupado).

- Si el medio no está ocupado por ninguna otra trama, la estación ejecuta una espera adicional llamada espaciado entre tramas (*IFS, Inter Frame Space*).

- Si durante este intervalo temporal, o bien ya desde el principio, el medio se determina ocupado, entonces la estación debe esperar hasta el final de la transacción actual antes de realizar cualquier acción.

- Una vez finalizada la espera del *IFS*, la estación ejecuta el llamado algoritmo de *Backoff*, según el cual se determina una espera adicional y aleatoria escogida uniformemente en un intervalo llamado ventana de contención (*CW, Contention Window*).

    - El algoritmo de *Backoff* da un número aleatorio y entero de ranuras temporales (*slot time*) y su función es la de reducir la probabilidad de colisión que es máxima cuando varias estaciones están esperando a que el medio quede libre para transmitir.

- Mientras pasan los slot time (ranura de tiempo) de espera calculados por el algoritmo de *Backoff*, la estación continúa escuchando el medio, de tal manera que si el medio se determina libre durante un tiempo de al menos *IFS*, los *slot time* de espera disminuyen hasta cero.

    - Si el medio no permanece libre durante un tiempo igual o superior a *IFS* el algoritmo de Backoff queda suspendido hasta que se cumpla esta condición.

    - Cada retransmisión provocará que el valor de CW, que se encuentra entre *CWmin* y *CWmax* se duplique hasta llegar al valor máximo. Por otra parte, el valor del *slot time* es 20µseg.

- Una vez finalizada la espera de los *slot* time del algoritmo de *Backoff* el cliente puede transmitir

Sin embargo, *CSMA/CA* en un entorno inalámbrico y celular presenta una serie de problemas. Los dos principales problemas que se puede detectar son:

- Nodos ocultos. Una estación cree que el canal está libre, pero en realidad está ocupado por otro nodo que no oye.

- Nodos expuestos. Una estación cree que el canal está ocupado, pero en realidad está libre pues el nodo al que oye no le interfiere para transmitir a otro destino.

La solución que propone 802.11 es MACA o *MultiAccess Collision Avoidance*. Según este protocolo, antes de transmitir el emisor envía una trama *RTS* (*Request to Send*), indicando la longitud de datos que quiere enviar. El receptor le contesta con una trama *CTS* (*Clear to Send*), repitiendo la longitud. Al recibir el *CTS*, el emisor envía sus datos.

Los nodos seguirán una serie de normas para evitar los nodos ocultos y expuestos:

- Al escuchar un *RTS*, hay que esperar un tiempo por el *CTS*
- Al escuchar un *CTS*, hay que esperar según la longitud

La solución final de 802.11 utiliza *MACA* con *CSMA/CA* para enviar los *RTS* y *CTS*.

2.3.5 Estándar IEEE 802.11e

El estándar IEEE 802.11e es una propuesta que define los mecanismos utilizados en una red inalámbrica para proporcionar QoS a aplicaciones en tiempo real como voz y vídeo. En este nuevo estándar, se hace una distinción entre aquellas estaciones que no utilizan los servicios QoS, que se denominan *nQSTA*, y aquellas que si los utilizan, llamadas *QSTA*. Para proporcionar soporte *QoS*, en IEEE 802.11e se introduce una tercera unción de coordinación, llamada función de coordinación híbrida (*HCF, Hybrid Coordination Function*). *HCF* incorpora dos nuevos mecanismos de acceso al canal: acceso distribuido al canal (*EDCA, Enhanced Distributed Channel Access*) y acceso controlado al canal (*HCCA, HCF Controlled Channel Access*). La principal característica de *HCF* es la definición de cuatro categorías de acceso (*AC, Access Category*) y de ocho flujos de tráfico *TS* (*Traffic Stream*) a nivel *MAC*. Cuando un paquete procedente de las capas superiores llega a la capa *MAC*, es etiquetado con un identificador de prioridad de usuario (*TID, Traffic Identification*) acorde con sus necesidades de *QoS*. Este identificador puede tomar valores de 0 a 15. Si el *TID* del paquete tiene valores de 0 a 7, es mapeado con respecto a las cuatro *AC*, usando el método *EDCA* para acceder al canal. Si por el contrario el identificador *TID* tiene valores de 8 a 15, usará la función *HCCA* para acceder al medio, quedando almacenado el paquete en la cola de *TS* correspondiente a su *TID*. Otra característica incluida en este nuevo estándar es el concepto de oportunidad de transmisión (*TXOP, Transmison Opportunity*), que es un intervalo de tiempo en el cual la estación que lo posee tiene permiso para enviar sus tramas.

- *EDCA* (*Enhanced Distributed Channel Access*)

El método de acceso al medio *EDCA*, pretende mejorar el funcionamiento de *DCF*, tratando de forma preferencial a las aplicaciones con restricciones en el tiempo. Para realizar esta diferenciación, *EDCA* introduce dos métodos: El primero de ellos es asignar distintos *IFS* a cada categoría de acceso. Para ello, el estándar introduce un nuevo tiempo de espera entre tramas (*AIFS, Arbitration InterFrame Space*). El valor de *AIFS* es:

$$AIFS[AC] = AIFSN[AC] \times SlotTime + SIFS$$

Donde se utiliza un número de arbitraje entre tramas (*AIFSN, Arbitration InterFrameSpace Number*), para la diferenciación entre las distintas *AC*.

El segundo método utilizado es asignar distintos tamaños de ventana *CW* para cada *AC*. Con este segundo método, el estándar pretende asignar menores tiempos de espera a las estaciones más prioritarias cuando estas tengan que efectuar el mecanismo de *Backoff*. Estos tamaños se obtendrán mediante la asignación de distintos tamaños límite de ventana *CWmin* y *CWmax*. Otro

factor utilizado para la distinción en *EDCA*, es la duración del *TXOP* (*TXOPLimit*). Este parámetro limita el tiempo en el que una estación tiene los derechos para transmitir, sin que el resto de estaciones le disputen el canal.

El funcionamiento del mecanismo es distribuido. Dos o más *AC* dentro de una misma *QSTA* pueden poner a 0 su contador de *Backoff* en el mismo instante. Si esto ocurre, ambos flujos intentarán mandar los datos produciéndose una colisión, que en el estándar han denominado colisión interna. Siempre que esto se produzca, la capa *MAC* ofrecerá la oportunidad de transmisión al flujo más prioritario, tratando el de menor prioridad igual que si se hubiera producido una colisión real.

2.4 FRAGMENTACIÓN EN REDES INALÁMBRICAS

Un cliente o punto de acceso, utiliza la fragmentación para dividir las tramas en fragmentos que se envían por separado al destino. Debido a que el emisor transmite cada fragmento independientemente, el receptor contesta con un reconocimiento separado para cada fragmento.

El campo del control de la secuencia de cada fragmento incluye un sub-campo del número del fragmento, indicando el número del fragmento principal. Si el número es cero para el primer fragmento, entonces se inician los incrementos uno por uno para cada fragmento sucesivo de un grupo particular. El fragmento de un solo *bit* más el campo del fragmento en el fragmento principal indica sí un marco es o no, el último de una serie de fragmentos y el campo del fragmento se fija a "1" si la estación fuente va a enviar elementos adicionales del mismo fragmento principal. Se fija en cero si no hay más fragmentos para enviar.

La estación de destino vuelve a montar los fragmentos nuevamente dentro del marco original usando los números del fragmento encontrados en el principal. Después de asegurarse que el marco es completo se empieza a procesar. Aún cuando la fragmentación implica más gastos indirectos, su uso puede dar lugar a un funcionamiento mejor si se utiliza correctamente.

2.4.1 Fragmentación Dinámica

La fragmentación dinámica no existe como tal, actualmente los cambios en fragmentación se realizan de manera manual en los dispositivos que manejan este tipo de características (por ejemplo en los enrutadores inalámbricos, puntos de acceso, entre otros). Por esta razón el fin principal de este Trabajo es ofrecer en dentro de un firmware basado totalmente en código abierto características como ésta que mejorarán el desempeño de las trasmisiones y recepciones de datos en una red de área local inalámbrica sometida a los efectos de la interferencia, utilizando para ello un algoritmo creado para evaluar el nivel de interferencia y modificar el valor de la fragmentación de forma dinámica.

Normalmente el "*Fragmentation Threshold*" es un parámetro que puede variar el administrador del sistema manualmente, en la mayoría de los equipos inalámbricos se tiene un valor por defecto y presenta la opción de cambio dentro de un rango determinado.

En el capítulo 4 se mostrarán los resultados obtenidos con las pruebas más pertinentes y en diferentes condiciones de interferencia en los canales de transmisión.

El valor de la fragmentación está fijado de fábrica en 2346 bytes y el rango de trabajo está entre 256 y 2346 octetos. Esto especifica el tamaño máximo para un paquete antes de que los datos sean fragmentados en múltiples paquetes. Si se experimenta una alta tasa de error en la

transmisión de los paquetes, se puede variar levemente el umbral de la fragmentación. Fijar el umbral de la fragmentación demasiado bajo puede dar lugar a un bajo rendimiento en el funcionamiento de una red, por eso es recomendable que los cambios en la tasa de *bytes* de fragmentación se hagan de manera corta.

En la figura 14 se puede observar que los tamaños máximos de los fragmentos *IP* pueden variar, y tal como se había mostrado antes, con el fin de mejorar el rendimiento de la red inalámbrica.

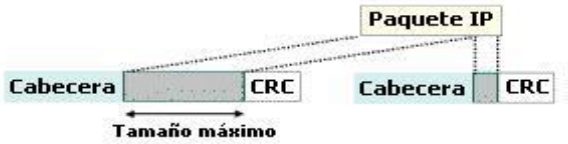

Figura 14. Cambio de Tamaño de los Fragmentos

2.4.2 Características de la Implementación de la Fragmentación

El uso de la fragmentación puede aumentar la confiabilidad de las transmisiones debido a que al enviar tramas más pequeñas, las colisiones son mucho menos probables de que puedan ocurrir, además, en caso de que ocurran, como el paquete fragmentado es de menor tamaño, se pierde menos información en dicho paquete.

El objetivo de construir una característica adicional para el firmware, se basa en realizar el control de la fragmentación de manera automática, esto significa que en presencia de ruido la fragmentación de los paquetes disminuye hasta lograr que se mitigue en lo posible la interferencia por un canal adyacente o incluso en el mismo canal. Por otro lado, si la red no presenta ruido, la fragmentación se hace innecesaria, debido a que baja el rendimiento a la red; en este caso el programa debe realizar el aumento en el tamaño de la fragmentación y tratar de estar siempre en el nivel más alto.

Un método para descubrir si es necesaria la fragmentación es supervisar las colisiones. Si se encuentra un número relativamente grande de colisiones, entonces se debe intentar usar la fragmentación. Esto puede mejorar el rendimiento de la red si el umbral de la fragmentación es el correcto.

Este trabajo mostrará que se pueden hacer cambios en la fragmentación de tal manera que se pueda mejorar la transmisión de datos entre 2 puntos de manera inalámbrica. Toda la base de este desarrollo se encuentra en la manera más efectiva de hacer cambios a nivel de firmware para lograr este tipo de resultados favorables a la comunicación.

Si están ocurriendo pocas colisiones (menos de 5 por ciento), no es necesario hacer cambios, es importante tener este dato en cuenta, esto debido a que los cambios en fragmentación ocasionarían cambios en el rendimiento y esto genera consecuencias a la hora de trasmitir o recibir información por este tipo de medios.

Si están ocurriendo colisiones significativas, es necesario intentar fijar el umbral de la fragmentación en un valor de 1.000 octetos en primer lugar, después se puede realizar un aumento hasta que se encuentren los mejores resultados. Después de iniciar la fragmentación, es necesario realizar muestreos continuos para saber si es beneficioso el valor fijado.

El uso de herramientas de la simulación en 802.11 puede ayudar en la determinación de los tamaños óptimos del umbral de la fragmentación, pero es necesario tener conocimiento del tema y desarrollar un modelo de simulación para cada red, esto debido a que las condiciones varían de acuerdo a muchos factores como ubicación, interferencia entre otros, es difícil tratar exactamente la interferencia en radio frecuencia en la red real, por lo cual se establece un método más práctico: la fragmentación dinámica.

2.4.3 Reducción al Mínimo de Interferencia en 802.11

La Interferencia es hoy en día uno de los inconvenientes más problemáticos en las redes inalámbricas debido al gran aumento de este tipo de redes y a la escasa o nula planificación en la escogencia de las frecuencias.

- Impacto de la Interferencia en *RF*

Cada estación 802.11 transmite solamente paquetes cuando no hay otra estación trasmisora. Si otra estación se dispone a enviar un paquete, las otras estaciones esperarán hasta que el medio está libre. Aunque todo esto es un poco más complejo, estos son los conceptos básicos.

La Interferencia en radiofrecuencia interrumpe las operaciones normales del sistema, debido al protocolo de acceso al medio de 802.11. Una señal de RF que interfiere con la suficiente amplitud en la frecuencia indicada, pueden parecerse a una falsa estación 802.11 que transmite un paquete. Esto hace que estaciones legítimas esperen períodos del tiempo indefinidos hasta que la señal que interfiere termine de trasmitir.

En los peores casos, una señal que interfiere no sigue generalmente los protocolos, así que esta señal puede comenzar precipitadamente a trasmitir mientras que una estación legítima de 802.11 está en curso de transmitir un paquete. Si ocurre esto, la estación destino recibirá el paquete con errores, por lo que la estación fuente vuelve a transmitir los paquetes.

En algunos casos, el funcionamiento de 802.11 procura continuar la operación en presencia de interferencia automáticamente cambiando a una tasa de datos más baja, que ocasiona retardos en las trasmisiones inalámbricas. Los casos más críticos ocurren cuando las estaciones basadas en 802.11 dejan de trasmitir hasta que desaparece totalmente la señal de interferencia, que perfectamente pueden ser minutos, horas, o días.

- Fuentes de interferencia de *RF* que pueden causar problemas

Para las redes inalámbricas basadas en 2.4 Ghz, hay varias fuentes de señales que interfieren, tales como hornos microondas, teléfonos inalámbricos, dispositivos *Bluetooth*, y otros equipos basados en la misma tecnología. Los más perjudiciales de éstos son los teléfonos inalámbricos de 2.4 Ghz ya que se han generalizado su uso en hogares y compañías. Es evidente que cuando uno de estos equipos está trabajando en el mismo sitio que una red inalámbrica la pérdida de rendimiento en la red inalámbrica se hace notoria.

Los hornos microondas que funcionan muy cerca de un punto de acceso generalmente interfieren en el funcionamiento de las redes 802.11b principalmente. Los dispositivos *Bluetooth*, tales como computadoras portátiles y *PDAs*, también causan degradaciones del funcionamiento si estos funcionan muy próximos a estaciones 802.11, especialmente si la estación 802.11 es relativamente lejana (es decir, los niveles de señal son bajos) de la estación con la cual se está comunicando.

Los grupos de investigación de los estándares 802.11 y 802.15 están trabajando en un estándar que permita la coexistencia de dispositivos *Bluetooth* y de 802.11 sin que exista mayor interferencia.

- Consideraciones y Acciones al experimentar interferencia:

    - Analizar los niveles de potencia e interferencia del Espectro (Análisis espectral). Hacer esto antes de instalar una red inalámbrica. usando las herramientas que se consiguen en el mercado y en muchos casos son gratuitas.

    - Evitar que las fuentes que interfieren el buen funcionamiento. Una vez que se conozca las fuentes potenciales de interferencia del RF, se podría eliminarlas simplemente apagando estos dispositivos. Ésta es la mejor manera de evitar la interferencia; sin embargo, no es siempre práctica. Por ejemplo, no se puede solicitar a una empresa u hogar vecino que apague sus dispositivos de red inalámbrica, sin embargo, se puede evitar el uso de dispositivos como hornos, dispositivos Bluetooth donde residen los equipos de 802.11 instalados.

    - Proporcionar la cobertura adecuada. Uno de los mejores concejos para evitar la interferencia de 802.11b es asegurar que la red inalámbrica tiene señales fuertes a través de las áreas donde se sitúen las estaciones cliente. Si las señales inalámbricas son demasiado débiles, las señales que interfieren serán más incomodas.

    - Fijar los parámetros de la configuración correctamente. Si ya se conoce el medio donde se ubicara una red inalámbrica es necesario tener claro en que canal se debe poner la emisión y recepción de la red a instalar. Esto no es siempre muy eficaz debido a que los cambios de canal de las redes vecinas pueden darse por simple casualidad e interferir con otras redes.

    - Utilizar redes 802.11a debido a que la mayor cantidad de interferencia de RF está hoy en 2.4 Ghz (es decir, 802.11). Vale la pena recordar que 802.11a trabaja en la banda de 5Ghz.

2.5 *RTS/CTS*

Como característica opcional, el estándar 802.11 incluye la función de *RTS/CTS* al acceso de la estación de control al medio. Generalmente con la característica de hacer a los equipos más costosos, aunque últimamente se puede ver en equipos de gama media y baja, también en todos aquellos equipos en los que se pueda compilar y poner en funcionamiento firmwares basados en software libre. Con el uso apropiado de *RTS/CTS*, se pueden realizar algunos cambios para manejar de manera más estable la transmisión en el medio inalámbrico mejorando la estabilidad y rendimiento de una red.

- Funcionamiento de *RTS/CTS*

Si se activa *RTS/CTS* en un equipo específico, se evitará el envío de información hasta que otra estación terminal realiza su transferencia con control de *RTS/CTS*, tal como un punto de acceso. Una estación inicia el proceso enviando una trama de *RTS*. Un punto de acceso recibe el *RTS* y

responde con una trama de *CTS*. La estación debe recibir un marco de *CTS* antes de enviar las tramas de datos. El *CTS* también contiene un valor del tiempo que muestra a otras estaciones su estado para poder tener acceso al medio mientras que la estación inicial arranca su transición sus datos.

El control y manejo apropiado de *RTS/CTS* proporciona control positivo sobre el uso del medio compartido. La razón principal de usar *RTS/CTS* es reducir al mínimo colisiones entre estaciones ocultas. Esto ocurre cuando se separan los usuarios y los puntos de acceso en condiciones como las mostradas en la Figura 15. En estos casos el número de las retransmisiones es relativamente alto.

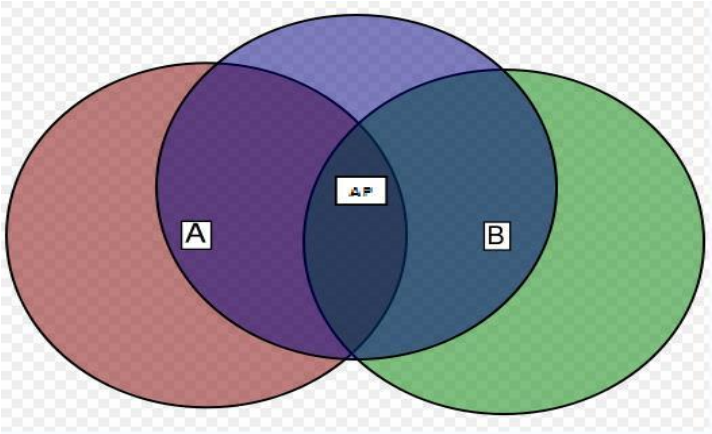

Figura 15. Cobertura de un Punto de Acceso con Nodo Oculto

En este caso hay dos usuarios finales (A y B) y un punto de acceso. La estación A y la estación B no puede oírse debido a la alta atenuación, sino que pueden ambos comunicarse con el mismo punto de acceso. Debido a esta situación, La estación A puede comenzar a enviar datos sin notar que la estación B está transmitiendo actualmente (o viceversa). Esto hará muy probablemente una colisión de paquetes entre la estación A y la estación B y ocurriría en el punto de acceso.

Consecuentemente, la estación A y la estación B necesitan retransmitir sus paquetes respectivos, lo que da lugar a un uso más alto del canal inalámbrico y a un procesamiento más alto en la red inalámbrica, lo cual baja su rendimiento.

Si la estación A o la estación B activan *RTS/CTS*, la colisión no sucede. Antes de transmitir, la estación B enviaría un *RTS* y recibiría un *CTS* del punto de acceso. El valor de la sincronización en los *CTS* hace que la estación A se mantenga detenida un tiempo hasta que la estación B deje de utilizar el canal. Así, el uso de *RTS/CTS* reduce colisiones y aumenta el rendimiento si hay estaciones ocultas en la red.

Es importante tener en cuenta, que el beneficio neto de transmitir más paquetes y de la reducción en las retransmisiones se pierde si no se tiene ningún nodo oculto, el uso de *RTS/CTS* aumenta la cantidad de procesamiento indirecto, lo que reduce el rendimiento de procesamiento de datos.

- Características de la implementación de *RTS/CTS*

Una de las mejores maneras de determinar si se activa *RTS/CTS* es supervisar las colisiones en las redes inalámbricas. Si se encuentra una gran cantidad de colisiones y los usuarios están separados probablemente fuera de máxima cobertura entre uno y otro, se puede activar *RTS/CTS* en el dispositivo del cliente si este lo permite. En la mayoría de los casos es suficiente con buscar las características avanzadas del dispositivo, activar la opción y guardar los cambios. De este modo es posible que no se necesite activar *RTS/CTS* en el punto de acceso. Después de recibir una trama de *RTS* del dispositivo del usuario, el punto de acceso responderá siempre con una trama de *CTS*.

Es importante saber que la movilidad del usuario puede cambiar los resultados. Un usuario que se movilice puede entrar a estado oculto por un período de tiempo corto, después de este periodo, puede estar más cercano a otras estaciones por mayor tiempo. Si las colisiones están ocurriendo entre los usuarios dentro de la red inalámbrica donde se sitúa este usuario, el problema puede ser el resultado de la alta utilización de la red o posiblemente de interferencia del RF.

Después de que *RTS/CTS* se activa, la prueba para determinar si el número de colisiones es menor al de antes, se observa si el rendimiento del procesamiento es mejor y es más alta tasa de trasmisión. Si es sensible la baja de rendimiento de la red ocasionada por el aumento de procesos y después de examinar la interferencia de la red y los otros equipos, se debe desactivar el *RTS/CTS.*

El método para activar *RTS/CTS* en puntos de acceso es diferente al de los dispositivos clientes. Para los puntos de acceso, se activa *RTS/CTS* fijando un umbral del tamaño del paquete del específico (0 a 2347 octetos) en la interfaz de la configuración del usuario. Si el paquete que el punto de acceso está transmitiendo es más grande que el umbral, iniciará la función de *RTS/CTS*. Si el tamaño del paquete es igual o menos que al umbral, el punto de acceso no ejecutará el *RTS/CTS*. La mayoría de los vendedores recomiendan el usar de un umbral de alrededor 500. El uso de 2347 octetos inhabilita *RTS/CTS* para el punto de acceso.

En la mayoría de los casos, activar *RTS/CTS* en el punto de acceso es infructuoso porque el problema de nodo oculto no existe de la perspectiva del punto de acceso. Forzar el punto de acceso para usar *RTS/CTS* aumentará perceptiblemente el procesamiento del equipo y reducirá rendimiento de la red.

2.5.1 Umbral *RTS*

Escrito de esta forma, esta característica sería un parámetro manejable en un equipo como tal, es decir, una variable y característica de fácil manipulación, normalmente por entorno web o línea de comandos. Umbral *RTS* es un mecanismo implementado para evitar el problema de "Nodo Oculto" básicamente. Nodo Oculto es una situación explicada anteriormente donde dos estaciones están dentro del área de alcance del mismo punto de acceso, pero no están dentro del área de alcance la una de la otra. Así, ellas son nodos ocultos entre sí. Cuando una estación oculta empieza la transmisión de datos con el punto de acceso, puede no percibir que la otra ya está usando el medio inalámbrico. Cuando las dos estaciones envían datos en el mismo instante, puede haber colisión de datos cuando lleguen simultáneamente al punto de acceso. La colisión causará la pérdida de los mensajes de ambas estaciones. Así, el mecanismo Umbral *RTS* ofrecerá la solución para evitar la colisión de datos. Cuando la función *RTS* esté activa, la estación y su punto de acceso usarán un protocolo a Solicitar para Enviar / Borrar para Enviar (*RTS/CTS*). La estación enviará un *RTS* al punto de acceso, informando que está por transmitir los datos. Después de recibirlos, el punto de acceso contesta con un mensaje *CTS* para todas las estaciones dentro del

área de alcance para solicitarles que posterguen la transmisión. Esto también confirmará, a la estación solicitante, que el punto de acceso ha reservado el canal para transmisión.

2.5.2 Nodo Oculto

En las redes *ethernet* las transmisiones se establecen mediante el protocolo CSMA/CD, que se encarga de detectar las colisiones. En estas redes los cables son el medio físico que contiene las señales y las distribuye a los nodos.

Las redes inalámbricas tienen unas características más ásperas en el sentido en que no todos los nodos pueden comunicar directamente con el resto de nodos. Teniendo un esquema como el que se muestra en la Figura 16:

Figura 16. Transmisión con Nodo Oculto

El nodo 2 puede comunicarse con ambos nodos, el 1 y el 3, pero hay algo que impide que los nodos 1 y 3 se comuniquen directamente. (El obstáculo en si mismo no es relevante; podría ser tan simple como que los nodos 1 y 3 se encuentran a una distancia y solo son capaces de comunicar con nodo 2). Desde la perspectiva de nodo 1, nodo 3 es un "nodo oculto".

Si el protocolo usado para transmitir es un simple "transmitir", será fácil para nodo 1 y nodo 3 transmitir simultáneamente, haciendo que nodo 2 sea incapaz de procesar nada. Además, los nodos 1 y 3 no tendrán conocimiento del error porque la colisión es a nivel local en el nodo 2. Las colisiones producidas por nodos ocultos pueden ser difíciles de detectar en redes inalámbricas debido a que los dispositivos inalámbricos son normalmente half-duplex; no transmiten y reciben al mismo tiempo.

Como se observa en la Figura 17, el nodo 1 tiene que mandar una trama de datos, para ello inicia el proceso enviando una trama RTS. La trama *RTS* tiene varios propósitos: Además de reservar el radio enlace para transmisión, también silencia a las otras estaciones que la oigan. Si la estación destino recibe un RTS, esta responde con un CTS. Al igual que la trama RTS, la trama *CTS* silencia las estaciones en la inmediata vecindad. Una vez que el intercambio *RTS/CTS* se completa, el nodo 1 puede transmitir sus tramas sin preocuparse de las interferencias producidas por nodos ocultos. Los nodos ocultos que se encuentren mas allá de la zona de la estación emisora son silenciados por el *CTS* del receptor. Cuando se usa el procedimiento *RTS/CTS*, cualquier trama debe ser positivamente aceptada.

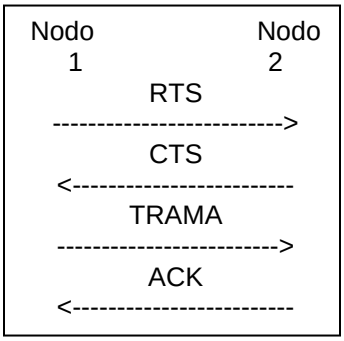

Figura 17. Transmisión entre Nodos

La transmisión *RTS/CTS* multi-trama, consume bastante capacidad, especialmente por la latencia adicional provocada antes de que las transmisiones puedan comenzar. Como consecuencia, solo es usado en entornos de alta-capacidad y entornos con considerable contención en transmisión. Para entornos de baja capacidad no es necesario.

Se puede controlar el procedimiento *RTS/CTS* configurando el umbral *RTS* (*RTS threshold*), si el driver para la tarjeta 802.11 lo permite. El intercambio *RTS/CTS* tiene lugar para tramas mayores que el threshold. Las tramas menores que el umbral RTS, son enviadas simplemente.

2.6 DESARROLLO DE NUEVOS PAQUETES

Con el transcurrir del tiempo se han venido realizando desarrollos en firmwares encaminados a agregar funcionalidades que incrementan el rendimiento y funcionamiento de los equipos hardware en las redes inalámbricas, algunos de estos desarrollos han sido:

- Servidor PPTP VPN
- Cliente y Servidor SSH
- Firewall
- Sistema de Distribución Inalámbrica
- Enrutamiento OSPF
- Enrutamiento RIP2
- Control de Potencia
- Selección de Antenas
- Soporte DDNS
- Wireless MAC address clone
- *WPA* sobre Sistema de Distribución Inalámbrica
- *WPA*/*TKIP* con *AES*
- Control de Ancho de Banda por protocolo
- Redireccionamiento de Puertos
- Wake On Lan
- Syslog Remoto
- Estadísticas Remotas de Ntop
- SNMP

En este capítulo se ha dado una introducción a las redes inalámbricas, pretendiendo realizar la caracterización del estándar mediante una abstracción de los problemas y posibles mejoras al estándar, se realiza un estudio de las funciones de coordinación puntual y distribuida, de las cuales se resalta el que no existe un manejo de la calidad del servicio, para esto, se crea el estándar 802.11e, el cual pretende realizar la implementación de una función de coordinación hibrida que permita dar prioridad al tráfico que lo necesite. Estos desarrollos sobre el estándar y cualquier otro, son módulos que se incorporan a él, y los fabricantes son libres de implementarlos en su nueva versión de firmware o no. También se mostraron algunas de las ventajas y desventajas de la modulación en OFDM, con lo que se da una idea de las limitaciones que se tienen actualmente en este campo, pero a la vez se abren nuevas posibilidades para investigar y realizar laboratorios en donde se apliquen los conocimientos impartidos en las asignaturas de señales.

Finalmente se expone el nuevo concepto de la fragmentación dinámica, un concepto original, que parte del análisis que se hace del estándar, en el cual se vislumbra la capacidad de manipular los parámetros que brinda el estándar y por otro lado se analizan las herramientas que ofrece el software de código abierto, en especial Linux, para implementar en un firmware características adicionales en cualquier capa del modelo OSI.

3 DESARROLLO E IMPLEMENTACIÓN DEL FIRMWARE EN SISTEMAS OPERATIVOS CON SOPORTE INALÁMBRICO

En la primera parte de este capítulo se hace una introducción a la instalación del sistema operativo Linux, el cual es la base fundamental sobre la que se desarrolla el proyecto. En la segunda sección se expone como se realiza la instalación y configuración de un host como punto de acceso. Luego se explica el proceso para llevar a cabo el desarrollo de la compilación e implementación de un firmware basado en el kernel de Linux y finalmente se muestra como se llevo a cabo la compilación e instalación de una de las características adicionales desarrolladas en el capítulo 2 para el nuevo firmware.

3.1 INSTALACIÓN DE SISTEMAS OPERATIVOS LINUX

Existen varios cientos de distribuciones, las más conocidas son: RedHat, Suse, Mandrake, Debian y Slackware, y Ubuntu. Aunque se puede apreciar que todas se basan sobre el mismo esquema fundamental, de manera que los conceptos basicos se pueden encontrar en referencias como [26], la cual es una buena guia de aprendizaje para el lector que desee familiarizarse más con Linux.

Se debe tener en cuenta que Linux necesita un procesador de la línea 386 como mínimo, y que la memoria mínima recomendable es 4MB; aún cuando una de las ventajas de Linux es el soporte de casi cualquier tipo de hardware, existen todavía algunas excepciones para algunos dispositivos, por lo que es recomendable leer con anterioridad el documento *Hardware-HowTo* que contiene cada distribución, con el fin de comprobar si el hardware sobre el que se va trabajar está soportado.

Actualmente todos los Sistemas Operativos Linux, basados en Unix vienen en CDs o DVDs y requieren características de hardware mínimas que varían de una a otra.

Dado que en los computadores actuales compatibles existe una gran variedad de hardware desarrollado por diferentes fabricantes, el sistema necesita unos controladores para cada tipo dispositivo, que en el caso del Linux se encuentran compilados en el propio núcleo. Por ello hay multitud de discos de arranque o *bootdisks*, cada uno con un Kernel diferente, evitando conflictos entre controladores.

Se selecciona SUSE 10 ya que esta distribución se encuentra bastante avanzada, la instalacion, configuración y mantenimiento se realiza mediante un asistente bastante intuitivo.

En el caso de SUSE 10, los pasos son los siguientes:

Se escoge la opción en la cual el dispositivo *boot* es el CDROM, luego se escoge la opción de instalación en el menú del CDROM, se deja la opción de partición recomendada por el sistema, se escogen los paquetes necesarios y se hace click en instalar.

3.2 CONTROLADORES PARA CONFIGURAR UN COMPUTADOR COMO PUNTO DE ACCESO

3.2.1 Controladores para Tarjetas Inalámbricas

Para poder hacer uso de un computador, ya sea de escritorio o portátil como un punto de acceso, es necesario tener en cuenta los modos de funcionamiento de un dispositivo inalámbrico, en Linux normalmente existen los siguientes:

- *Master*: Modo en el cual el dispositivo inalámbrico puede funcionar como punto de acceso inalámbrico.
- *Management* o Infraestructura: modo en el cual la tarjeta de red inalámbrica funciona como cliente de un punto de acceso.
- *Ad-Hoc*: Modo diseñado para soportar una pequeña red de equipos terminales sin punto de acceso.
- *Monitor*: Modo en el cual la tarjeta de red inalámbrica se comporta de manera pasiva en la cual solo escucha la información que se encuentra en el aire.

La gran mayoría de los dispositivos inalámbricos que funcionan en sistemas operativos de Microsoft Windows, se proporciona la posibilidad de trabajar en 2 modos, estos son:

- *Management*
- *Ad-Hoc*

Es decir, pueden ser clientes de puntos de acceso inalámbricos, es así cómo funcionan las tarjetas de Red inalámbricas creadas para desempeñarse bajo el estándar 802.11x de la IEEE.

El modo Ad-Hoc, realiza una conexión punto a punto entre dispositivos, pensado para redes provisionales con alrededor de cinco (5) equipos sin punto de acceso.

En equipos de escritorio o portátiles que tengan instalados sistemas operativos Linux, se puede aprovechar mejor las características de funcionamiento, debido a que los otros modos son funcionales. En este caso, mediante una tarjeta inalámbrica y los controladores adecuados se convierte un computador normal en un punto de acceso.

3.2.2 Requisitos Mínimos

Existen algunas características mínimas que deben tenerse en cuenta:

Hardware:

- Computador de escritorio (o portátil)
- Tarjeta (dispositivo) de red inalámbrica

Software

- Sistema operativo Linux con soporte Inalámbrico

- Controladores adecuados para la tarjeta de red

No todos los dispositivos de red inalámbricos, tienen suficiente desarrollo para poder soportar los diferentes modos de funcionamiento. En la mayoría de casos, aunque el hardware pueda permitir esto, no hay controladores (*Drivers*) que soporten dichas características.

Para el proyecto se utilizó una tarjeta inalámbrica D-LINK, la cual utiliza el chip AR 2414A-00 de la compañía Atheros. Específicamente se utilizó el siguiente hardware:

3 Enrutadores Inalámbricos Linksys WRT54G
1 Enrutador Inalámbrico Linksys WRT54GS
3 Computadores tipo Escritorio
3 Portátiles Toshiba - HP
2 Dispositivos de Red (Tarjetas) PCI DWL – G520
2 Dispositivos de Red (Tarjetas) PCMCIA DWL – G650

En cuatro (4) de los computadores se instaló Suse 10 como sistema operativo, las razones por las cuales se escogió esta distribución es el grado de madurez (versión 10), la estabilidad, la gran recopilación de controladores que posee, el kernel 2.6 y que cuenta con las últimas versiones de los compiladores más importantes como *GCC* (*GNU Compiler Collection*).

El MadWifi [27], es uno de los Controladores que maneja uno de los chips más populares de la compañía Atheros, este es un controlador genérico que se encuentra bastante desarrollado y permite utilizar cuatro modos de funcionamiento para las tarjetas inalámbricas basadas en esta clase de chip. Se escogieron las tarjetas D-Link debido a que cuentan con el chip Atheros el cual brinda la posibilidad de trabajar en todos los modos. En este caso, los cuatro dispositivos de red tienen chips de la compañía Atheros, los cuales son soportados por MadWifi, según la tabla de compatibilidad publicada en su página Web [28].

3.2.3 Instalación y Configuración

Para la instalación de los controladores en Linux Suse es necesario estar como root en el sistema operativo para poder hacer la instalación y compilación del software.

Nota: Para la adquisición del software se utilizó SVN (subversión) esta es una aplicación con la cual se pueden descargar aplicaciones y archivos:

```
svn checkout http://svn.madwifi.org/trunk madwifi-ng
```

Se crea el directorio `madwifi-ng`

```
/madwifi ~#
```

En este directorio se debe ejecutar siguiente comando:

`make` (para poder construir el controlador) y luego `make install` (para instalarlo)

Después de tenerlo instalado se debe ejecutar el comando:

```
modprobe ath_pci
```

Esto es para cargar el módulo del controlador.

MadWifi soporta interfaces virtuales, de manera que se puede especificar la creación de una interfase virtual de diversos tipos, en este caso se crea una interfaz virtual del tipo punto de acceso. Con los módulos cargados, se puede crear la interfaz inalámbrica con el siguiente comando:

```
wlanconfig ath0 create wlandev wifi0 wlanmode sta
```

Donde ath0 es el nombre de la interfaz, tal como funcionaría con eth0 para interfaces ethernet.

Con el comando `iwconfig`, se lista las interfaces de red que tiene el sistema operativo y se observa algo como lo siguiente:

```
/madwifi ~#  iwconfig

eth0        no wireless extensions.
Lo          no wireless extensions.
wifi0       no wireless extensions.
ath0        IEEE 802.11g  ESSID:""
            Mode:Managed      Frequency:2.457   GHz      Access   Point:
            00:00:00:00:00:00
            Bit Rate:0 kb/s   Tx-Power:20 dBm   Sensitivity=0/3
            Retry:off   RTS thr:off   Fragment thr:off
            Power Management:off
            Link Quality=0/94  Signal level=-95 dBm  Noise level=-95 dBm
            Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
            Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Según la configuración anterior, el modo de funcionamiento escogido fue *sta* el cual coloca el dispositivo en modo infraestructura, es decir, como cliente de un AP.

Sin embargo, el modo que se necesita para funcionar como punto de acceso es el modo master (punto de acceso).

Las opciones permitidas para el comando son:

```
wlanconfig ath0 create wlandev wifi0 wlanmode
[sta|adhoc|ap|monitor|wds|ahdemo]
```

En la sintaxis del anterior comando se puede ver los modos de funcionamiento que puede tener el dispositivo de red, de acuerdo a esto el comando que se necesita es el siguiente:

```
wlanconfig ath0 create wlandev wifi0 wlanmode ap
```

Con este comando se inicia la tarjeta en modo AP.

3.3 DESARROLLO DEL FIRMWARE BASADO EN LINUX

A mediados de 2002, la empresa Linksys utilizó el Kernel de Linux para la primera versión de su dispositivo inalámbrico WRT54G, el fabricante recibió las ventajas del código abierto plasmado en las licencias GPL como el acceso al código fuente y en particular el soporte que ofrece Linux a las

diversas clases de hardware embebido como memorias y procesadores, en particular los procesadores basados en MIPS, más tarde haciendo caso a la licencia GPL, este fabricante decidió liberar el código de su firmware el cual se encuentra disponible para la utilización y modificación de los desarrolladores a nivel mundial [29], el resultado fue el surgimiento de empresas y organizaciones que ofrecían por su parte un firmware adaptado para esta clase de dispositivos, cada uno con una característica en particular. Mientras tanto se fué organizando un proyecto conocido como openwrt el cual desarrollo un firmware soportado en el enrutador WRT54G/GS de Linksys.

Openwrt es una distribución de Linux para enrutador inalámbrico, el cual en vez de intentar saturar el firmware con muchas características, provee un mínimo de estas, ofreciendo la posibilidad de añadir paquetes desarrollados por los programadores. Para los usuarios esto significa que se tiene la capacidad de incluir características personalizadas, quitar los paquetes indeseados para hacer espacio a otros paquetes y para los desarrolladores significa que pueden centrarse en los paquetes sin tener que probar y lanzar un nuevo firmware desde el principio.

Basados en este principio se tomó la decisión de trabajar sobre openwrt, uno de los firmware más estables para crear una compilación, el cual permite la adición de paquetes y características adicionales que se desarrollarán en este trabajo.

Existen algunos firmwares reconocidos en el entorno, basados en el Kernel de Linux, dentro de los cuales se destacan:

Talismán, Alchemy, DD-wrt, Hyper wrt, Ewrt y BatBox.

Se escoje el openwrt debido a varias razones: Una de ellas, es debido a que es de los proyectos con mejor documentación, la cual se toma como referencia [30]. Además presenta lo indispensable para funcionar, es decir, soporte al procesador, controladores para las interfaces de red y demás hardware, integración los paquetes básicos para el funcionamiento del enrutador como iptables, consola para el acceso por telnet y ssh. Estos paquetes básicos se encapsulan en un archivo binario de 1.5MB, de manera que se tiene un espacio de 2.5MB en memoria para utilizar, este es un recurso escaso, el cual conviene economizar debido que solo se cuenta con 4MB de memoria flash en total, tambien se debe tener en cuenta que al tener solo los procesos necesarios ejecutándose, se reduce la posibilidad de fallas que puedan ocurrir por ocupación de memoria y cantidad de procesos consumiendo recursos.

Con el tiempo desarrolladores que han colaborado con el proyecto han incorporado a su método de compilación un sistema de *makefiles* para construir las herramientas de compilación (*toolchain*) y la compilación cruzada del firmware de una forma más genérica. Este es un punto de partida para la creación de un firmware más especializado, tal como lo demuestra el que la mayoría de los firmwares citados anteriormente tienen su origen total o parcial en openwrt.

Dentro del firmware más evolucionado se encuentra el dd-wrt, este firmware se analizó ampliamente debido a que posee diversas capacidades adicionales que han sido desarrolladas por colaboradores del proyecto en todo el mundo. Este firmware es recomendable para utilización a nivel de configuración de servicios y aplicación, ya que presenta módulos para el control de potencia de transmisión, servidor web, control de ancho de banda, QoS, interacción con servidores radius, aplicación para el montaje de *hotspot*, *keepalive* entre otros, los cuales solo necesitan ser configurados para su funcionamiento.

Esta clase de firmware no es recomendable para el desarrollo debido a que utiliza bastantes recursos del sistema, una compilación de dd-wrt está en el orden de los 3MB, lo cual limita a menos de 1MB para utilizar, además de cargar el sistema constantemente con todos los servicios que maneja.

3.3.1 Construcción de las Herramientas para la Compilación

Para comenzar, se debe tener en cuenta que los pasos necesarios para llevar a cabo todas las tareas de compilación se realizan con la herramienta *make*, el propósito de esta utilidad es determinar automáticamente qué piezas de un programa necesitan ser recompiladas y de acuerdo a un conjunto de reglas, lleva a cabo las tareas necesarias para alcanzar el objetivo definido, que en este caso es compilar un firmware. Para proyectos con varios cientos de líneas de código, permite agilizar el proceso de construcción de los programas, y en general, facilita el trabajo de compilación para uno o más archivos. De esta forma y con los archivos adecuados, *make* compila todos los programas fuentes. Si alguno de ellos sufre alguna modificación, sólo será recompilado aquel que fue modificado, junto con todos los programas que dependan de él. Por supuesto, es necesario indicarle a *make* la dependencia de programas, lo cual se realiza en el archivo *makefile*.

Al conjunto de *makefiles*, necesarios para la compilación se le denomina *buildroot*, el cual tiene dos objetivos: el primero es generar un conjunto de herramientas capaces de realizar la compilación cruzada, al cual se le denomina *toolchain*, y el segundo es generar un sistema de archivos raíz para un kernel sobre una arquitectura específica, esto es, un firmware con su sistema de archivos basado en el Kernel de Linux.

El *toolchain* para la compilación cruzada es un sistema de herramientas que permite compilar el código para un sistema determinado, el cual consiste de un compilador (en este caso, *GCC*), utilidades binarias como el ensamblador y el enlazador (en este caso, *binutils*) y una biblioteca estándar de *C* (por ejemplo: *GNU Libc*, *uClibc* o *dietlibc*). El *toolchain* para la compilación cruzada utiliza el *uClibc*.

GCC, es la base de la compilación, éste se encuentra bien documentado y es referencia obligada para el lector que quiere conocer mas de este proceso, la documentación de este proyecto se puede encontrar en en el anexo y en mayor profundidad en [31].

Un sistema Linux instalado en una estación de trabajo posee un *toolchain* de compilación, el cual se utiliza para compilar aplicaciones que se ejecuten en dicho sistema, por ejemplo si se utiliza un computador personal, el *toolchain* de compilación funciona en un procesador x86 y genera el código para un procesador x86. Este *toolchain* se denomina "*toolchain* de compilación del anfitrión", y más generalmente, la máquina en el cual él está funcionando se llama "sistema huésped". El *toolchain* de compilación es proporcionado en cada distribución Linux y bajo la mayoría de sistemas Linux se utiliza el *libc* de *GNU* como biblioteca estándar de *C*.

Según lo anterior, el *toolchain* de compilación que viene con cada sistema funciona y genera el código para el procesador de su sistema huésped. Debido a que para un sistema embebido se tiene un procesador diferente, se necesita un *toolchain* de compilación cruzada, esto es un *toolchain* de compilación que funciona en el sistema huésped pero que genera el código para un sistema de objetivo y procesador objetivo diferentes al sistema huésped. Por ejemplo, si el sistema huésped utiliza un procesador x86 y el sistema de objetivo utiliza un procesador *MIPS*, el *toolchain* regular de compilación de su anfitrión funciona en x86 y genera el código para x86, mientras que el *toolchain* de compilación cruzada funciona en x86 y genera el código para *MIPS*.

Es posible compilar directamente teniendo las herramientas adecuadas, *GCC binutils*, *uClibc* y todas las herramientas a mano, pero presenta diversos problemas debido a que se deben configurar todas las opciones para la compilación a mano y además existen incompatibilidades entre cada versión de *GCC*, *uClib* o *binutils*, esta forma es más dispendiosa y debido a esto se creó la herramienta *buildroot*.

El *buildroot* automatiza este proceso con el uso de *makefiles*, tiene una colección de parches para cada versión de *GCC* y *binutils* y realiza el trabajo para la arquitectura *MIPS* de la mayoría de los dispositivos inalámbricos.

3.3.2 Obtención de *Buildroot* para el Firmware

Para cualquier clase de desarrollo para firmware basado en Linux, en este caso tomando como base openwrt, se debe descargar directamente de openwrt mediante la herramienta *SVN* en su versión más actualizada. Se puede obtener en la siguiente dirección:

```
$ svn co https://svn.openwrt.org/openwrt/trunk/
```

La versión estable más recomendada de openwrt *buildroot* (*whiterussian*) para el desarrollo de paquetes se puede descargar mediante *SVN* en:

```
$ svn co https://svn.openwrt.org/openwrt/branches/whiterussian/
```

3.3.3 Uso de *Buildroot* en el Firmware

Buildroot es una herramienta de configuración similar a la que se puede encontrar en el kernel de Linux o en *busybox* [32]. Es posible ejecutar todo como usuario normal, no hay necesidad de ser usuario root para configurar y utilizar el *buildroot*. El primer paso es ejecutar el ayudante de configuración en el directorio en el cual se descomprime el openwrt.

```
/openwrt ~# make menuconfig
```

Para cada entrada de la herramienta de configuración, se puede encontrar la ayuda asociada que describe el propósito de la entrada.

Una vez que se configuren los paquetes, aplicaciones y utilidades, la herramienta de configuración genera un archivo de configuración (*.config*) que contiene la descripción de la configuración. Este archivo será utilizado por los *makefiles* para hacer lo que se requiere.

Luego se digita:

```
/openwrt ~# make
```

Este comando ejecuta el código consignado en los *makefiles*, tomando como parámetro lo escogido en el archivo `.config`, descarga las fuentes necesarias de los diferentes *mirror* en internet, configura y compila todas las herramientas seleccionadas, y finalmente genera la imagen del firmware para el sistema objetivo y los paquetes adicionales (dependiendo de las selecciones dentro del `make menuconfig`). Todos los archivos binarios para la arquitectura objetivo se pueden encontrar en la carpeta *`bin/`*. Se puede compilar las imágenes del firmware para las diversas arquitecturas objetivo en dos tipos diferentes de sistema de archivos: *jffs2 y squashfs*, estos son los dos sistemas de archivos especializados para sistemas embebidos.

jffs2: Contiene un sistema de archivos raíz re-escribible, que se amplía al tamaño de la imagen de la memoria flash disponible del hardware. La ventaja de utilizar jffs2 es que se puede realizar modificaciones dentro de toda la jerarquía de archivos, aunque se corre el riesgo de dejar inservible el enrutador si se configura de manera inadecuada y el sistema de archivos total tiene un tamaño superior al que se obtiene con *squashfs*.

Si se utiliza la imagen genérica del firmware, se debe escoger la imagen correcta para el tamaño de memoria flash, 8Mb o 4Mb, dependiendo del modelo del enrutador, debido a los diversos tamaños del *eraseblock* o mínima unidad borrable del chip de memoria flash del sistema de archivos *jffs*.

Squashfs: contiene un sistema de archivos raíz de solo lectura que usa un sistema de archivos *squashfs* modificado el cual utiliza un algoritmo de alta compresión denominado *LZMA* (*Lempel-Ziv-Markov chain-Algorithm*). Al iniciar, se puede crear un segundo sistema de archivos re-escribible, que contendrá las modificaciones al sistema de archivos raíz, incluyendo los paquetes que se instalan.

En el caso de utilizar el sistema *jffs2* no es posible dañar por error el sistema de archivos original, debido a que no se escribe directamente sobre él, en caso de error, se puede resetear el dispositivo a los valores por defecto y de esta manera se pierden todos los cambios realizados.

3.3.4 Personalización del Sistema de Archivos Objetivo para Requisitos Particulares

Hay dos maneras de modificar el sistema de archivos de la arquitectura objetivo para requisitos particulares los cuales son:

La primera es modificar el sistema de archivos para la arquitectura objetivo directamente con los requisitos particulares y luego reconstruir la imagen. El sistema de archivos para la arquitectura objetivo está disponible en: `/openwrt/build_ARCH/root/` donde *ARCH* es la arquitectura elegida, en este caso *MIPS*, se pueden realizar cambios aquí y ejecutar `make target_install`, lo cual reconstruirá la imagen del sistema de archivos. Este método permite hacer todo en el sistema de archivos para la arquitectura objetivo, pero al reconstruir el *toolchain*, las herramientas o paquetes creados se perderán.

La segunda es modificar el esqueleto del sistema de archivos para requisitos particulares que se encuentra disponible en: `/openwrt/package/base-files/default/`. Se pueden modificar los archivos de la configuración para requisitos particulares aquí, sin embargo, la jerarquía completa del sistema de archivos no está todavía presente porque se crea durante el proceso de la compilación, de manera que no se puede hacer todos los cambios en este esqueleto del sistema de archivos, pero los cambios que se hagan persisten aun cuando se reconstruya totalmente el *toolchain* de la compilación cruzada y las herramientas.

3.3.5 Modificar la Configuración de *Busybox* para Requisitos Particulares

Busybox es configurable, y se puede modificar para requisitos particulares. Su configuración se integra totalmente en el sistema principal del *menuconfig*. Este se puede encontrar en el menú de configuración:

```
"selección de paquetes de OpenWrt" = > "configuración de Busybox"
```

En esta parte *busybox* presenta todas sus opciones de configuración.

3.3.6 Modificar la Configuración de u*Clibc* para Requisitos Particulares

Al igual que *BusyBox, uClibc* ofrece varias opciones de configuración por lo que se permite seleccionar varias funcionalidades dependiendo de las necesidades y limitaciones. La manera más fácil de modificar la configuración del *uClibc* es seguir estos pasos:

- Hacer una primera compilación de *buildroot* sin modificar el *uClibc* para requisitos particulares.

- Entrar en el directorio: `/openwrt/toolchain_build_ARCH/uClibc/` y ejecutar `make menuconfig`. Con esto se inicia el asistente de configuración, este asistente es similar al que se usa para la configuración del Kernel de Linux, luego se realiza la configuración apropiada escogiendo los ítems adecuados.

- Se copia el archivo `.config` a la ruta:

  `toolchain/uClibc/uClibc.config ó`
  `toolchain/uClibc/uClibc.config-locale.`

Se utiliza el primero si no se ha seleccionado la ayuda del *locale*, esto es, si no se activo la casilla de verificación en la configuración de *buildroot*, y se utiliza el segundo si se ha seleccionado la ayuda del *locale* en la configuración del *buildroot*. *Locale* es el set de parámetros que definen el lenguaje, país o región.

- se ejecuta la compilación nuevamente.

También, se puede modificar simplemente con:

- `toolchain/uClibc/uClibc.config o`
- `toolchain/uClibc/uClibc.config-locale`

Sin ejecutar el ayudante de la configuración.

3.3.7 Funcionamiento del *Buildroot*

El *buildroot* es básicamente una colección de *makefiles* que descargan, configuran y compilan el software con las opciones correctas. También incluye algunos parches para el software, principalmente los que están implicados en el *toolchain* de compilación cruzada (*GCC binutils y uClibc*).

Hay básicamente un *makefile* por software y están divididos en tres secciones:

- *Package*, (el directorio *package/*) contiene los *makefiles* y los archivos asociados para todas las herramientas del usuario que *buildroot* pueda compilar y agregar al sistema de archivos raíz de la arquitectura objetivo. Hay un subdirectorio por cada herramienta.

- *Toolchain*, (el directorio *toolchain/*) contiene los *makefiles* y los archivos asociados para todo el software relacionado con el *toolchain* de la compilación cruzada: *binutils* ccache *GCC gdb Kernel-headers y uClibc*.

- *Target*, (el directorio *target/*) contiene los *makefiles* y los archivos asociados para el software relacionado con la generación de la imagen del sistema de archivos raíz de la arquitectura objetivo y el Kernel de Linux para los distintos sistemas de chips que vienen en las tarjetas madre de los enrutadores y puntos de acceso inalámbricos. Existen dos tipos de sistema de archivos soportados: *jffs2* y *squashfs*.

Cada directorio contiene por lo menos 2 archivos:

Makefile: es el *makefile* que descarga, configura, compila e instala el software.

`Config.in` es una parte del archivo de la descripción de la herramienta de la configuración, el cual describe opciones relacionadas con el software.

El *makefile* principal hace el trabajo con los pasos siguientes (una vez que se hace la configuración):

- Crea el directorio de descarga (`/openwrt/dl/` por defecto). Aquí es donde los archivos fuente comprimidos (*tarballs*) serán descargados. Es interesante saber que los *tarballs* están en este directorio porque puede ser útil para evitar descargas futuras.

- Crea el directorio build *(*`/openwrt/build_ARCH/` por defecto, donde *ARCH* es el tipo de arquitectura objetivo). Es aquí donde todas las herramientas del usuario son compiladas.

- Crea el directorio *build toolchain (*`/openwrt/toolchain_build_ARCH/` por defecto, donde *ARCH* es su arquitectura). Aquí es donde el *toolchain* de la compilación cruzada será compilado.

- Construye el directorio *staging (*`/openwrt/staging_dir_ARCH/` por defecto). Aquí es a donde el *toolchain* de la compilación cruzada será instalado.

- Se puede utilizar el mismo *toolchain* de compilación cruzada para otros propósitos, tales como compilación de otras aplicaciones, para esto se debe agregar `/openwrt/staging_dir_ARCH/bin` a su respectiva *PATH*, y luego utilizar *arch-linux-gcc* para compilar. Para la construcción de este directorio *staging*, primero se quita, y luego se crean varios subdirectorios y enlaces simbólicos dentro de él.

- Crea el directorio objetivo *(*`/openwrt/build_ARCH/root/` por defecto) y el esqueleto del sistema de archivos para la arquitectura objetivo. Este directorio contendrá el sistema de archivos raíz final. Para instalarlo, primero lo suprime, luego copia el esqueleto disponible en:

  `target/default/target_skeleton` y luego elimina directorios SVN/ inutiles.

- Prepara compila e instala en los subdirectorios *toolchain package y target*

3.3.8 Utilizar el *Toolchain* del *Uclibc*

Se puede compilar programas u otro software que no esté empaquetado en openwrt. Para hacer esto, se puede utilizar el *toolchain* que fue generado por el *buildroot*.

El *toolchain* generado por *buildroot* por defecto se encuentra en `/openwrt/staging_dir_ARCH.`

La forma más simple para usarlo es añadir.

```
/openwrt/staging_dir_ARCH/bin/
```

A la ruta de variable de entorno, luego usar el compilador adecuado: `arch-linux-gcc, arch-linux-objdump, arch-linux-ld`, para cada tipo de programa.

Por ejemplo, se puede añadir lo siguiente al `.bashrc` (considerando que se está construyendo el Kernel para una arquitectura basada en MIPS y el *buildroot* está localizado en `/openwrt/`)

```
export PATH=$PATH:~/openwrt/staging_dir_mipsel/bin/
```

O simplemente se puede compilar directamente con el compilador para *mipsel*:

```
mipsel-linux-uclibc-gcc -o paquete_ejemplo paquete_ejemplo.c
```

Importante: No se debe mover el *toolchain* a otro directorio por qué no funcionará, debido a que existe código fuertemente ligado en la configuración de *GCC*. Si se desea mover el directorio del *toolchain*, se debe referir a la siguiente sección 3.3.9, Utilizar el *toolchain* fuera del *buildroot*.

3.3.9 Utilizar el *uClibc Toolchain* fuera del *Buildroot*

Por defecto, la compilación cruzada se genera en el directorio `/openwrt/staging_dir_ARCH/` pero se puede instalar en otro directorio, esta puede ser usada para compilar otros programas o por otros usuarios. Moviendo el directorio directamente no se puede utilizar debido a que hay código que está fuertemente ligado al *toolchain*. Para realizar esto, se puede configurar el *buildroot* para que lo genere usando la herramienta de configuración.

```
build options -> Toolchain and header file location, which defaults to
staging_dir_ARCH/.
```

Y luego se modifica este último.

3.3.10 Ubicación de los Paquetes Descargados

Puede ser que sea útil saber que varios de los *tarballs* que son descargados por los *makefiles* están almacenados en el `DL_DIR` que por defecto es el directorio de `DL`. Es útil si se desea guardar una versión completa de *buildroot* conocida para trabajar con los *tarballs* asociados. Esto permitirá regenerar el *toolchain* y el sistema de archivos del sistema objetivo con exactamente las mismas versiones que se hicieron la primera vez.

3.4 CONFIGURACIÓN DEL FIRMWARE

Después de seguir las indicaciones de la sección 3.3.2 para descargar las fuentes necesarias para la compilación se procede a configurarlo y compilarlo.

3.4.1 Descargar las Fuentes Necesarias

Ejecutar el siguiente comando en el shell:

```
svn co https://svn.openwrt.org/openwrt/branches/whiterussian/
```

En el cual se encuentran las fuentes y las librerías necesarias. Estas fuentes se descargan al directorio `/whiterussian/openwrt`, donde se encuentra lo necesario para la compilación.

Aquí se puede encontrar el *readme* de la versión y los archivos descargados. Estos son los directorios que aparecen luego de la descarga:

```
build_mipsel, dl, docs, package, scripts, stagging_dir_mipsel, target,
toolchain_build_mipsel

archivos:
config.in, config.in.devel, license, makefile, readme, rules.mk
```

3.4.2 Ejecutar el `Make Menuconfig`

Para esto es necesario tener instalados los siguientes paquetes: *gcc, g++, binutils, patch, bzip2, flex, bison, make, gettext, unzip, libz-dev y libc headers.*

Luego se digita el siguiente comando dentro del directorio `/openwrt`:

```
/openwrt ~# make menuconfig
```

Con este comando se inicia el *makefile* que realiza la configuración del archivo de configuración del *buildroot*, con la cual se tiene acceso al menú de configuración, en este aparecen 4 ítems, en donde se puede elegir los paquetes básicos que se pueden instalar en el nuevo firmware, la configuración para el tiempo de ejecución, el sistema de archivos objetivo y la configuración del Kernel para el soporte de los diferentes dispositivos, como se observa en la Figura 18.

Se escogen las opciones de manera que el enrutador no quede sobrecargado con recursos que no se van a utilizar, que solo consumen procesamiento y espacio en memoria, con el fin de utilizar estos recursos en el desarrollo del nuevo módulo.

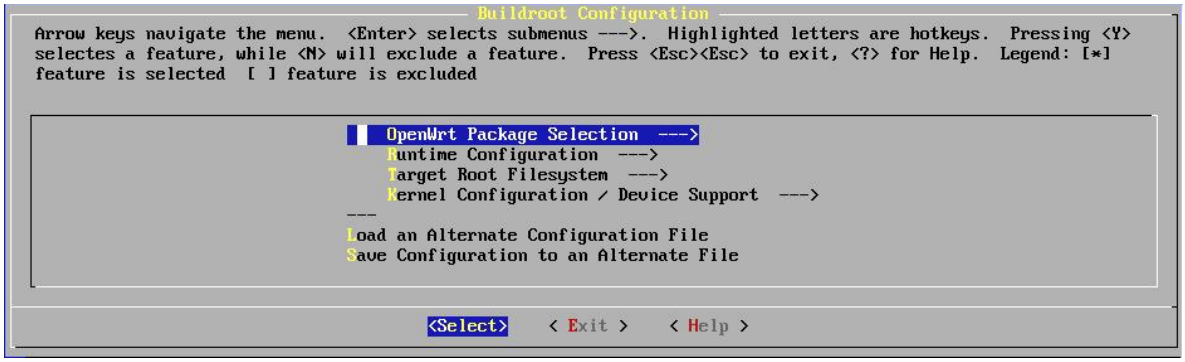

Figura 18. Menú de Configuración Principal

En el menú *package selection* que se muestra en la Figura 19, se pueden encontrar los paquetes que se van a compilar con el Firmware.

Esta opción permite escoger los paquetes que están disponibles en esta versión del *buildroot*, en el principio, a mediados del 2004 solamente había unas pocas aplicaciones desarrolladas para el sistema, hoy en día existen unas decenas de estas, las cuales han sido desarrolladas por diferentes personas del mundo, lo cual demuestra el gran interés que ha generado este desarrollo, debido a las infinitas posibilidades que ofrece a los programadores para controlar un hardware.

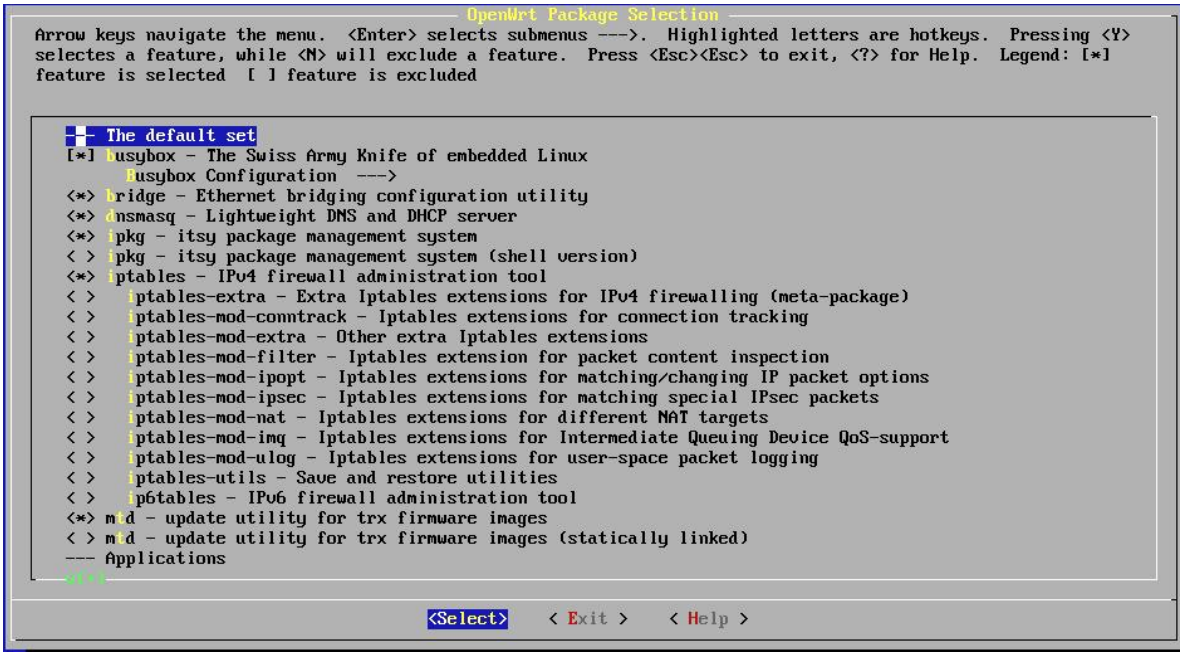

Figura 19. Menú de Selección de Paquetes

La segunda opción del menú principal es la configuración en tiempo de ejecución, Figura 20. En esta opción es posible escoger el tipo de acceso disponible, telnet abierto o telnet solo en modo a prueba de fallos.

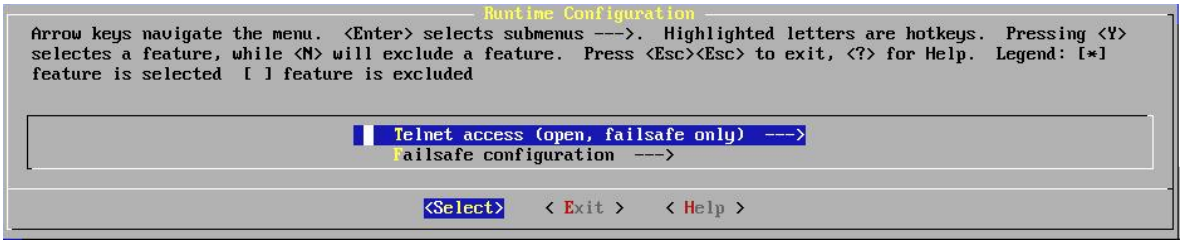

Figura 20. Menú de Configuración en Tiempo de Ejecución

En la Figura 21 se puede observar que se escoge el telnet de manera abierta (sin clave), como predeterminado para el modo *failsafe only*, esto quiere decir que se tiene acceso al dispositivo

mediante la aplicación telnet solamente cuando se encuentra en modo a prueba de fallos, esto es útil cuando hay una falla y se necesita recuperar el sistema.

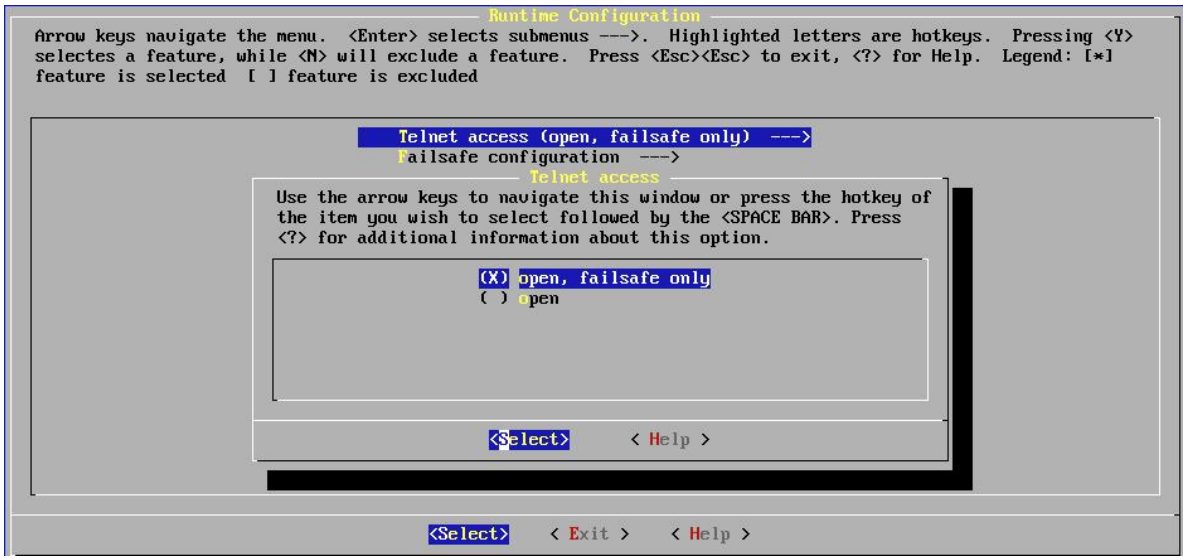

Figura 21. Menú de Configuración para Telnet

La segunda opción del submenú *run time configuration* es *failsafe configuration*, en este submenú se pueden configurar las opciones cuando el router se encuentre en modo *failsafe*, Figura 22.

Es recomendable conservar la dirección IP por defecto del enrutador Linksys original para evitar confusiones, esta es 192.168.1.1/24

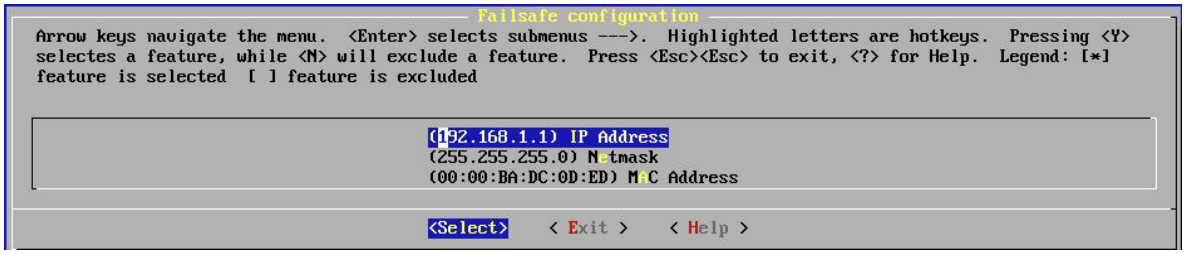

Figura 22. Escogencia de la Dirección IP por Defecto

En la opción 3 del menú principal, *target root filesystem* se puede escoger entre el sistema de archivos *jffs2* y *el squashfs* con compresión *LZMA*, Figura 23, o realizar la configuración para que se generen ambos sistemas de archivos, esto permite de escoger cualquiera de los dos tipos para insertar en el enrutador Linksys. En este caso se escogen ambas opciones de sistema de archivos.

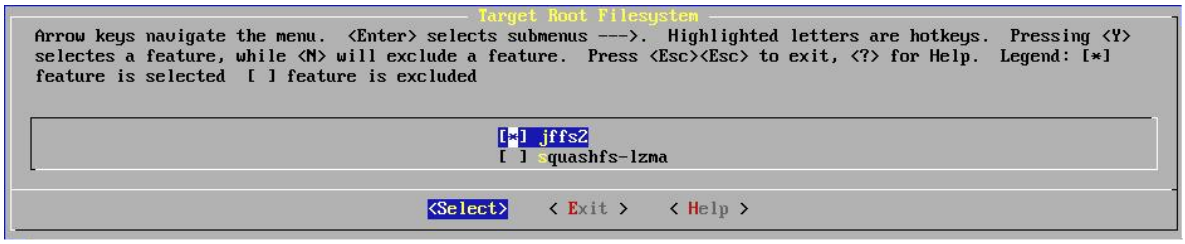

Figura 23. Menú de Configuración para el Sistema de Archivos

La cuarta opción del menú principal es la configuración del Kernel, donde se encuentran las opciones que tiene para configurar las utilidades y dispositivos disponibles en el Kernel de Linux, como se muestra en la Figura 24. En esta sección se encuentran ítems como la utilización del driver para el LED, el tipo de procesador, y módulos extra para IP.

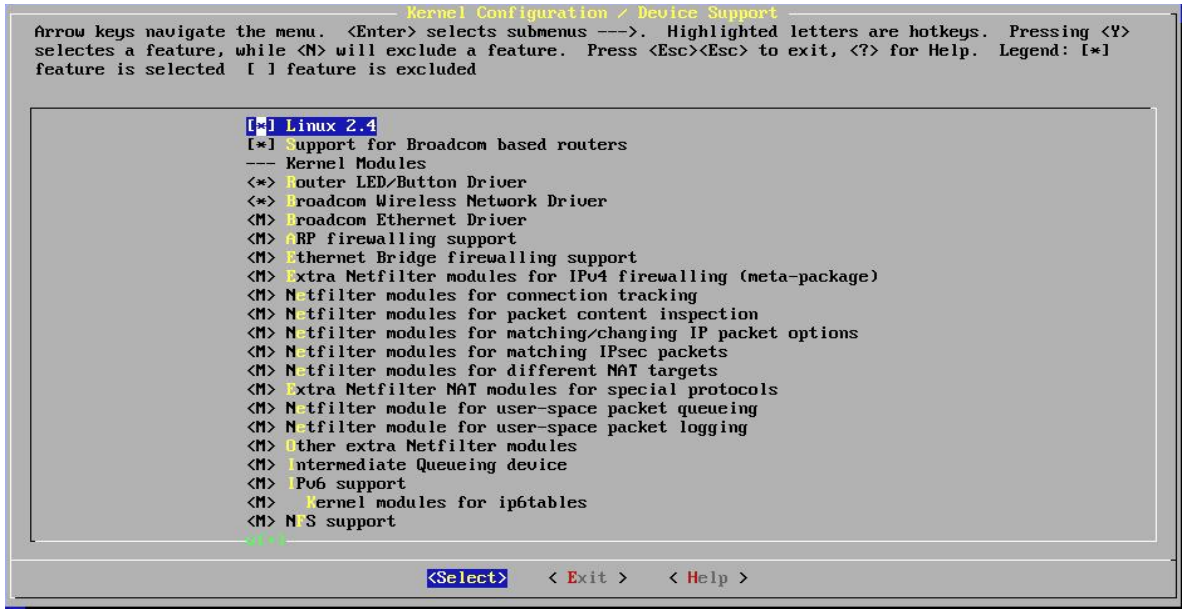

Figura 24. Menú de Configuración para Módulos Extra del Kernel.

Las últimas dos opciones son para cargar o guardar un archivo de configuración `.config`, a una ruta especifica, como se muestra en la Figura 25.

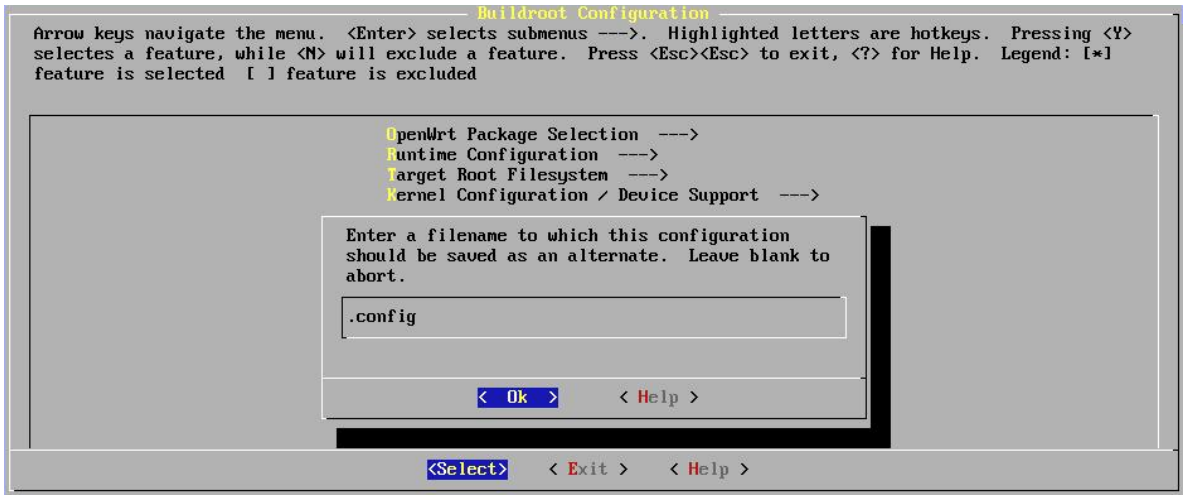

Figura 25. Guardar en un Archivo de Configuración y Ruta.

Por último se debe guardar la configuración del *buildroot* en el archivo `.config`, esto se hace cuando se sale de la utilidad de configuración como se muestra en la Figura 26.

Figura 26. Guardar la Configuración

Se aceptan los cambios y queda configurado el archivo de configuración `.config` del *buildroot*.

Ahora se debe salir de la configuración con *exit*.

3.4.3 Ejecutar `make`

Se ejecuta el siguiente comando:

```
/openwrt ~# make
```

Con este comando se inicia la compilación y construcción del Kernel, se descargan las fuentes necesarias desde Internet, se realiza la compilación cruzada mediante el *toolchain*, construcción del firmware con las aplicaciones seleccionadas.

Nota: cualquier librería que haga falta ocasiona un error en la compilación, por lo cual esta se detiene, de manera que se debe descargar e instalar las librerías necesarias para que la compilación se lleve a cabo satisfactoriamente.

En este momento se encuentra operativo el firmware, con las características configuradas mediante el menú de configuración, se realizan las pruebas las cuales indican su buen

funcionamiento y la estabilidad del sistema, estas pruebas son en condiciones ideales ya que no existen más dispositivos inalámbricos cercanos y el ruido se encuentra en niveles bajos y normales, alrededor de -96dBm. Las pruebas con ruido se consignan en el capítulo cuatro.

En esta compilación personalizada se cuenta en un kernel base que funciona de la manera esperada y estable, a partir de este momento se puede experimentar con él, modificarlo, personalizarlo, añadir paquetes que incluyan funciones adicionales o mejoras a las ya existentes ya que se trata de paquetes de código abierto.

Es posible incluir paquetes que incorporen los últimos adelantos en el manejo de QoS, clientes y servidores para los actuales desarrollos en seguridad, manejo de SNMP y muchos otros, esta es una base que da opciones ilimitadas para descubrir e implementar aplicaciones que mejoren el desempeño de esta clase de dispositivos.

3.5 FIRMWARE OPENSOURCE

3.5.1 Ampliando el Firmware con Software Adicional

Se puede ampliar el firmware básico con el software nuevo desarrollado. Para ampliar el firmware agregando software adicional, se debe tener en cuenta lo siguiente:

3.5.2 El Directorio `Package`

Primero, se debe crear un directorio dentro del directorio `package` para el software, en este caso se llamara: `paquete_ejemplo`.

3.5.3 El Archivo `Config.In`

Se debe crear un archivo llamado `config.in`. Este archivo contiene la parte de la descripción de las opciones relacionada con el software a crear, el cual será utilizado y exhibido en la herramienta de configuración. Debe contener básicamente lo siguiente:

```
config BR2_PACKAGE_paquete_ejemplo
        tristate "paquete_ejemplo - herramienta de ejemplo"
        default m if CONFIG_DEVEL
        help
        Comentario sobre que es el paquete_ejemplo.
```

Si el paquete depende de otro software o librería dentro del *buildroot*, se deben seleccionar estos paquetes en el archivo `config.in`, por ejemplo si `paquete_ejemplo` depende de la librería `bar`:

```
config BR2_PACKAGE_paquete_ejemplo
        tristate "paquete_ejemplo - herramienta de ejemplol"
        default m if CONFIG_DEVEL
        select BR2_PACKAGE_LIBBAR
        help
        Comentario sobre que es el paquete_ejemplo
```

Se pueden añadir opciones para configurar opciones particulares en el software `paquete_ejemplo`

3.5.4 `Config.In` en el Directorio `Package`

Para agregar un paquete nuevo a la herramienta de configuración, se necesita agregar la línea siguiente `a package/Config.in`, adaptada al `paquete_ejemplo`:

```
comment "Networking"
source "package/paquete_ejemplo/Config.in"
```

3.5.5 *Makefile* en el Directorio `Package`

Para agregar un paquete nuevo a la herramienta de configuración, se necesita editar el *makefile* y agregar una línea en `package/config.in`. Se debe localizar la línea que luce como la siguiente:

```
package-$(BR2_PACKAGE_paquete_ejemplo) += paquete_ejemplo
```

Esta línea agrega simplemente el objetivo `paquete_ejemplo` a la lista de los objetivos (*handhelds*) manejadas por el openwrt *buildroot*, con lo cual se ejecuta el *makefile* del `paquete_ejemplo`. Además de las dependencias por defecto, se puede hacer que el `paquete_ejemplo` dependa de otro paquete (por ejemplo una biblioteca) agregando una línea:

```
paquete_ejemplo-compile: bar-compile
```

3.5.6 El Archivo de Control de Ipkg

Adicionalmente, se debe crear el archivo de control, el cual contiene información acerca del paquete para ser leída por la utilidad *ipkg*. Esta utilidad es un sistema de gestión de paquetes muy liviano, funciona como un constructor de imágenes para memoria flash para los sistemas Linux embebidos y además permite la instalación/borrado de paquetes en un sistema. Fue diseñado para las instalaciones de Linux con limitaciones de almacenamiento, tales como dispositivos de mano o PDAs.

Este archivo puede ser creado como:

```
package/paquete_ejemplo/ipkg/paquete_ejemplo.control
```

Un ejemplo de archivo de control para *ipkg* es como el siguiente:

```
     1  Package: paquete_ejemplo
     2  Priority: optional
     3  Section: net
     4  Maintainer: paquete_ejemplo Software
<paquete_ejemplo@paquete_ejemplo.com>
     5  Source: http://paquete_ejemplo.com
     6  Depends: libbar
     7  Description: Descripción del paquete.
```

Generalmente se puede saltar la versión y campos de la arquitectura, pues éstas serán generadas por el script de `make-ipkg-dir.sh`, el cual es llamado desde el *makefile*. El campo *Depends* es importante, debido que la utilidad *ipkg* trae automáticamente todo el software que se encuentre en el campo "*dependend*" para añadirlo al sistema objetivo.

3.5.7 El *Makefile*

Se debe crear un archivo llamado *Makefile*, este es el archivo principal, donde se encuentran las reglas del *makefile*, estas reglas son las encargadas de llevar a cabo todas las tareas de descargar, configurar, compilar y de instalar el software. El siguiente es el ejemplo de archivo *makefile*, el cual se encuentra comentado al final.

```
 1  # $Id: buildroot-documentation.html wlx $
 2
 3  include $(TOPDIR)/rules.mk
 4
 5  PKG_NAME:=paquete_ejemplo
 6  PKG_VERSION:=1.0
 7  PKG_RELEASE:=1
 8  PKG_MD5SUM:=4584f226523776a3cdd2fb6f8212ba8d
 9
10  PKG_SOURCE_URL:=http://www.paquete_ejemplosoftware.org/downloads
11  PKG_SOURCE:=$(PKG_NAME)-$(PKG_VERSION).tar.gz
12  PKG_CAT:=zcat
13
14  PKG_BUILD_DIR:=$(BUILD_DIR)/$(PKG_NAME)-$(PKG_VERSION)
15  PKG_INSTALL_DIR:=$(PKG_BUILD_DIR)/ipkg-install
16
17  include $(TOPDIR)/package/rules.mk
18
19  $(eval $(call
PKG_template,paquete_ejemplo,paquete_ejemplo,$(PKG_VERSION)-
$(PKG_RELEASE),$(ARCH)))
20
21  $(PKG_BUILD_DIR)/.configured: $(PKG_BUILD_DIR)/.prepared
22          (cd $(PKG_BUILD_DIR); \
23                  $(TARGET_CONFIGURE_OPTS) \
24                  CFLAGS="$(TARGET_CFLAGS)" \
25                  ./configure \
26                  --target=$(GNU_TARGET_NAME) \
27                  --host=$(GNU_TARGET_NAME) \
28                  --build=$(GNU_HOST_NAME) \
29                  --prefix=/usr \
30                  --sysconfdir=/etc \
31                  --with-bar="$(STAGING_DIR)/usr" \
32          );
33          touch $@
34
35  $(PKG_BUILD_DIR)/.built:
36          rm -rf $(PKG_INSTALL_DIR)
37          mkdir -p $(PKG_INSTALL_DIR)
38          $(MAKE) -C $(PKG_BUILD_DIR) \
```

```
    39            $(TARGET_CONFIGURE_OPTS) \
    40            install_prefix="$(PKG_INSTALL_DIR)" \
    41            all install
    42          touch $@
    43
    44  $(IPKG_paquete_ejemplo):
    46          install -d -m0755 $(IDIR_paquete_ejemplo)/usr/sbin
    47          cp -fpR $(PKG_INSTALL_DIR)/usr/sbin/paquete_ejemplo
$(IDIR_paquete_ejemplo)/usr/sbin
    49          $(RSTRIP) $(IDIR_paquete_ejemplo)
    50          $(IPKG_BUILD) $(IDIR_paquete_ejemplo) $(PACKAGE_DIR)
    51
    52  mostlyclean:
    53          make -C $(PKG_BUILD_DIR) clean
    54          rm $(PKG_BUILD_DIR)/.built
```

Este ejemplo de *makefile* funciona para software binario. Para otro software más complejo tal como librerías o librerías más complejas con binarios múltiples, debe ser adaptado. Se pueden observar ejemplos de otros archivos *makefile* en el directorio `package/`.

Líneas 5-15: definición de variables.

`PKG_NAME`: El nombre del paquete, por ejemplo: `paquete_ejemplo`.

`PKG_VERSION`: La versión del paquete que debe ser descargado.

`PKG_RELEASE`: El número del lanzamiento que será añadido al número de versión del paquete ipkg.

`PKG_MD5SUM`: El archivo md5sum del software.

`PKG_SOURCE_URL`: Lista separada mediante espacios de los sitios del HTTP o del FTP de los cuales se descarga el archivo fuente necesario. Debe incluir la trayectoria completa al directorio donde el `paquete_ejemplo_SOURCE` puede ser encontrado.

`PKG_SOURCE` : El nombre tarball del paquete en el sitio web o en el sitio de descarga FTP. Como se puede ver se utilizan `PKG_NAME y PKG_VERSION`.

`PKG_CAT`: La herramienta que se necesita para la extracción del archivo software

`PKG_BUILD_DIR`: El directorio en el cual el software será configurado y compilado. Básicamente, es un subdirectorio de `BUILD_DIR` que se crea sobre la extracción del tarball.

`PKG_INSTALL_DIR`: El directorio en el cual el software será instalado. Es un subdirectorio de `PKG_BUILD_DIR`.

En la línea 3 y 17 se incluyen variables y rutinas comunes para simplificar el proceso de la creación del ipkg. Incluye rutinas para descargar, para verificar y para extraer los archivos de los paquetes de software.

La línea 19 contiene la línea que crea el ipkg.

Las líneas 21-33 definen el sistema objetivo y reglas asociadas que configuran el software, éstas dependen del sistema objetivo anterior (el archivo oculto en `.prepared`) de modo que se debe estar seguro que el software ha sido descomprimido. Para configurar éste, básicamente se ejecuta el script de configuración `/configurescript`. De la misma manera como se hace para la compilación cruzada, se deben proporcionar los argumentos para el sistema objetivo, el sistema host y la arquitectura. El prefijo es también fijado en `/usr`, no porque el software vaya a ser instalado en `/usr` en el sistema huésped, sino en el sistema de archivos objetivo. Finalmente se crea un archivo `.configured` para marcar el software como configurado.

Las líneas 35-42 definen un sistema objetivo y una regla que compila el software. Este sistema objetivo creará el archivo binario en el directorio de la compilación, y depende del software que está configurado (por lo tanto hace referencia al archivo `.configured`). Instala luego el binario resultante en: `PKG_INSTALL_DIR`. Este básicamente ejecuta el `make install` dentro del directorio *source* del *tarball*.

Las líneas 44-50 definen un sistema objetivo y las reglas asociadas que crean el paquete ipkg, el cual se puede encajar opcionalmente en la imagen del firmware que resulta. Se Instalan manualmente todos los archivos que se desean integrar en el ipkg. `RSTRIP` une todos los binarios y bibliotecas recursivamente.

Finalmente se llama a `IPKG_BUILD` para crear el paquete.

3.6 CONSTRUCCIÓN DE UN IPKG

La construcción de un paquete *.ipkg* se puede realizar mediante tres formas, mediante la utilización del *buildroot* que se usa para la compilación del openwrt, mediante la utilización del *SDK*, o mediante la compilación nativa dentro del WRT54G/GS/GL.

3.6.1 El Openwrt SDK

El SDK es un kit de desarrollo para la construcción de un paquete ipkg. Al usar el SDK no se requiere un buildroot completo. El SDK es una pequeña versión de este, que incluye el toolchain y todos los archivos requeridos, bibliotecas y las cabeceras o *headers* para compilar aplicaciones para el openwrt.

3.6.2 Requisitos

- Una distribución reciente de GNU/Linux
- La aplicación GNU make (por lo menos versión 3,80)

3.6.3 Uso del SDK para el Firmware

Para mostrar este procedimiento se va a crear y empaquetar el programa *wlx*, este es un programa totalmente nuevo, creado a partir del código escrito en lenguaje C, un lenguaje robusto y potente, con el cual es posible realizar prácticamente cualquier tarea dentro de un dispositivo.

Una vez se tiene el código de *wlx*, se debe proceder a compilar, el compilador utilizado solo puede ser *GCC*, esta es una limitación que se tiene hasta el momento, pero que a su vez garantiza la

portabilidad del software, esto es porque en realidad gcc está construido como un compilador cruzado, lo cual quiere decir que una compilación exitosa con gcc en forma nativa para cualquier plataforma(i386, i686, x64), probablemente provee una compilación exitosa mediante el compilador cruzado para otra plataforma como *mipsel*, esta es la forma como se realiza el desarrollo de aplicaciones con miras a insertarlas en el openwrt, se trabaja en una plataforma PC con el compilador gcc para Linux y luego de obtener los resultados esperados, se procede a utilizar el compilador cruzado con la seguridad de que la compilación será exitosa.

Uno de los problemas que se presentaron al utilizar *GCC* en conjunto con *uClibc* para compilar las nuevas funcionalidades para el firmware, es debido a la utilización de uClibc, debido a que ésta es una versión comprimida de *glibc*.

uClibc es una librería de C desarrollada para sistemas embebidos, la cual es una versión comprimida de *glibc* (la librería de C estándar para Linux) y permite realizar la compilación cruzada para los diferentes tipos de procesadores que se utilizan en dispositivos embebidos. uClibc no tiene implementadas todas las funciones que tiene el *glibc* para Linux, por lo cual se encuentran excluidas algunas funciones de C/C++ que no son muy utilizadas.

Aunque como es de esperar en el software de código abierto, el grupo de desarrolladores continúan trabajando para incluir más funciones dentro de este proyecto

Superadas estas limitaciones, al final de esta sección se estará en capacidad de construir un paquete *.ipkg* totalmente nuevo, partiendo del código escrito en lenguaje C, con lo cual se brinda la oportunidad de insertar funcionalidades adicionales en un dispositivo embebido quedando totalmente funcional dentro de su firmware.

El código fuente para el programa *wlx* en su primera versión es una prueba que realiza unas funciones básicas, la segunda versión está desarrollada en C++, y se ejecuta correctamente en el enrutador. Esto es la base para cualquier aplicación que se pueda desarrollar en C o en C++, con lo cual se cumple el objetivo de implementar una característica adicional partiendo del código escrito en lenguaje C/C++.

Una aplicación más funcional para implementar en el enrutador la cual se denomina *ucfrag*[5], establece un mecanismo de fragmentación automática, con la cual se pueden minimizar los efectos del ruido en un enlace inalámbrico. Esto se logra mediante la toma de muestras realizadas de los tiempos de latencia entre el AP y una estación, luego se realizan los ajustes en el parámetro de fragmentación del paquete en presencia de ruido. El código en lenguaje c para *wlx*, así como para *ucfrag* se encuentran en el anexo.

3.6.4 Obtención e Instalación del SDK

El SDK se puede descargar de

```
http://downloads.openwrt.org/whiterussian/newest/.
```

Se puede descargar en el directorio *home*, no es necesario utilizar la cuenta *root*, luego se descomprime el *tarball*.

```
lxuser@nx:~/home>wget
http://downloads.openwrt.org/whiterussian/newest/OpenWrt-SDK-Linux-i686-
1.tar.bz2
```

[5] Paquete *.ipkg* desarrollado por el grupo de trabajo para insertar en el firmware prototipo.

```
lxuser@nx:~/home bzcat OpenWrt-SDK-Linux-i686-1.tar.bz2 | tar -xvf –
```

Ahora se ingresa en el nuevo directorio.

```
lxuser@nx:~/home cd OpenWrt-SDK-Linux-i686-1
```

3.6.5 Creación de los Directorios

Se deben crear los siguientes directorios:

Directorio *ipkg*: Es donde se encuentra el archivo de control que contiene la información sobre el paquete
Directorio *patches*: Es donde se guardan los parches por ejemplo:

```
lxuser@nx:~/home>OpenWrt-SDK-Linux-i686-1
mkdir -p package/wlx/ipkg
mkdir -p package/wlx/patches
```

Para compilar más de un paquete en un orden especial se puede hacer de la siguiente manera:

```
lxuser@nx:~/home>OpenWrt-SDK-Linux-i686-1
mkdir -p package/100-wlx/ipkg
mkdir -p package/100-wlx/patches
mkdir -p package/200-wlx/ipkg
mkdir -p package/200-wlx/patches
```

3.6.6 Creación de los Archivos Requeridos

Para la creación de un paquete *.ipkg* se necesita de tres archivos fundamentales, estos son el archivo de configuración, el archivo *makefile* y el archivo de control, a continuación se muestra el contenido de cada uno de los archivos y la ruta en la que se encuentran.

3.6.7 Archivo de Configuración

Ruta: `package/wlx/Config.in`

```
config BR2_WLX
      prompt "Este es wlx version 0,1"
      tristate”wlx”
      default m if CONFIG_DEVEL
      help
      Esta es la primera version de wlx con ella se demuestra la
      capacidad para insertar un software totalmente nuevo, el cual
      inserta características adicionales al firmware del dispositivo.
```

3.6.8 Archivo *Makefile*

Ruta: `package/wlx/Makefile`

Se utiliza el comando `md5sum` para crear el `PKG_MD5SUM` del *tarball* original. Según la concepción del código abierto es posible contribuir con el proyecto a la comunidad en general, de manera que es posible ubicar las fuentes del programa en *SourceForge*. Este es un servidor de almacenamiento FTP en el cual se pueden alojar las fuentes para que estén disponibles al público en general. Para descargar de *SourceForge* se utiliza *@SF/* y se elige uno de los servidores *mirror* donde se encuentre el paquete `PKG_SOURCE_URL`.

Debido a que el código de el programa *wlx* es un paquete de prueba, solo se sube a un FTP local.

A continuación se especifica el *makefile* para el paquete *wlx*.

```
# $Id$

include $(TOPDIR)/rules.mk

PKG_NAME:=wlx
PKG_VERSION:=0.1
PKG_RELEASE:=1
PKG_MD5SUM:=32d2ab80d12bfd9bda9cc5e6b5485352

PKG_SOURCE_URL:=ftp://66.128.47.29/ftproot/
PKG_SOURCE:=$(PKG_NAME)-$(PKG_VERSION).tar.gz
PKG_CAT:=zcat

PKG_BUILD_DIR:=$(BUILD_DIR)/$(PKG_NAME)-$(PKG_VERSION)
PKG_INSTALL_DIR:=$(PKG_BUILD_DIR)/ipkg-install

include $(TOPDIR)/package/rules.mk

$(eval$(call                  PKG_template,WLX,$(PKG_NAME),$(PKG_VERSION)-
$(PKG_RELEASE),$(ARCH)))

$(PKG_BUILD_DIR)/.configured: $(PKG_BUILD_DIR)/.prepared
#Since there is no configure script, we can directly go to the building
step
      touch $@

$(PKG_BUILD_DIR)/.built:
      rm -rf $(PKG_INSTALL_DIR)
      mkdir -p $(PKG_INSTALL_DIR)/usr/bin
      mkdir -p $(PKG_BUILD_DIR)/src
      cp $(PKG_BUILD_DIR)/$(PKG_NAME).c $(PKG_BUILD_DIR)/src
      $(TARGET_CC)          $(PKG_BUILD_DIR)/src/$(PKG_NAME).c          -o
$(PKG_BUILD_DIR)/$(PKG_NAME)
      $(CP) $(PKG_BUILD_DIR)/PKG_NAME $(PKG_INSTALL_DIR)/usr/bin
      touch $@

$(IPKG_WLX):
      install -d -m0755 $(IDIR_WLX)/usr/bin
      $(CP) $(PKG_INSTALL_DIR)/usr/bin/wlx $(IDIR_WLX)/usr/bin
      $(RSTRIP) $(IDIR_WLX)
      $(IPKG_BUILD) $(IDIR_WLX) $(PACKAGE_DIR)

mostlyclean:
```

```
make -C $(PKG_BUILD_DIR) clean
rm $(PKG_BUILD_DIR)/.built
```

3.6.9 Archivo de Control

Ruta: `/wlx/ipkg/wlx.control`

```
Package: wlx
Priority: optional
Section: misc
Maintainer: Name <wmoreno@unicauca.edu.co>
Source: http://www.unicauca.edu.co/~wmoreno
Description: primera versión de wlx
```

3.6.10 El Paquete de Parches

Esta primera versión del paquete wlx no tiene parches por lo que esta sección no se implementa, sin embargo, se muestra como es su construcción.

Ruta: `/wlx/patches/100-wlx.patch`

```
cd package/wlx/patches
wget http://66.128.33.107/wlx_2.1.1-4.diff.gz
gunzip wlx_2.1.1-4.diff.gz
mv wlx_2.1.1-4.diff 100-wlx.patch
```

Se puede aplicar tantos parches como se necesiten. Para aplicarlos en un orden especial se deben nombrar como:

```
100-wlxparche1.patch
200-wlxparche2.patch
```

3.6.11 Compilación del Paquete

El comando *make* compila cada paquete que se ha creado en el directorio del `/package`.

```
lxuser@nx:~/home /OpenWrt-SDK-Linux-i686-1>
make clean && make compile
```

Cuando la compilación finaliza exitosamente se tiene listo para utilizar un paquete *ipkg* desarrollado para el firmware openwrt dentro del directorio:

```
lxuser@nx:~/home/OpenWrt-SDK-Linux-i686-1/bin/packages
```

Se comprueba el contenido del directorio.

```
lxuser@nx:~/home/OpenWrt-SDK-Linux-i686-1/bin/packages>ls
wlx_0.1-1_mipsel.ipk
```

El comando `md5sum` genera número de comprobación que se utiliza para verificar la autenticidad del paquete, este número debe incluirse en el archivo *makefile*, mediante este archivo realiza la comprobación luego de la descarga del paquete. Para la creación del número de comprobación se utiliza el comando `md5sum`, este se debe aplicar al archivo `tar` *o* `tar.gz`, esto se realiza de la siguiente manera.

Mediante una consola *bash*, se ingresa en la ruta donde se encuentra el archivo `.tar.gz`
Luego se digita el comando siguiente:

```
lxuser@nx:~/>md5sun wlx-0.1.tar.gz
lxuser@nx:~/>32d2ab80d12bfd9bda9cc5e6b5485352
```

Lo cual genera como resultado el número de comprobación, con el cual el archivo *makefile* reconoce si el paquete ha sido descargado correctamente, es decir, que su información es correcta y no ha sido alterada.

Este número es verificado inmediatamente después de la descarga, por lo cual si existe algún error, el compilador asume que el archivo tiene un error y detiene su ejecución.

3.6.12 Insertar las Características Adicionales en el Nuevo Firmware

Es posible insertar nuevos paquetes utilizando la aplicación *wget*, la cual está diseñada para permitir la fácil inclusión de nuevos paquetes en el firmware del dispositivo, para realizarlo de esta manera es necesario contar con un servidor WEB donde alojar el paquete `wlx_0.1-1_mipsel.ipk`

El primer paso para realizar la instalación de un nuevo paquete es abrir un terminal de consola en el enrutador, en este caso se tiene lo siguiente

```
Router #
```

Estando en esta consola se utiliza el comando `wget` para descargar el archivo `wlx_0.1-1_mipsel.ipk` de un servidor WEB, en este caso el paquete se encuentra en el servidor WEB de la universidad del cauca, cuya ruta completa es:

`www.unicauca.edu.co/~wmoreno/wlx_0.1-1_mipsel.ipk`, de manera que el comando completo para descargar es el siguiente:

```
Router # wget www.unicauca.edu.co/~wmoreno/wlx_0.1-1_mipsel.ipk
```

El programa *wget* se encarga de descargar el nuevo software en el dispositivo a la carpeta `/tmp`. Esta es la carpeta por defecto para las descargas. Para verificar que el software se descargo al dispositivo se utiliza el comando `ls`.

```
Router #ls
wlx_0.1-1_mipsel.ipk
```

Para realizar la instalación del paquete se hace uso de la aplicación *ipkg*, esta aplicación descomprime e instala el software, el comando es el siguiente:

```
Router # ipkg install wlx_0.1-1_mipsel.ipk
```

Cuando se complete el proceso que lleva a cabo *ipkg*, el software se encuentra ejecutable dentro del enrutador. Se puede verificar la existencia del nuevo paquete en el entorno del enrutador presionando dos veces la tecla TAB, en este momento el enrutador lista en la consola todos los comandos permitidos en el dispositivo.

```
Router #
awk
.
..
Cron
wl
wlx
```

Dentro de esta lista de comandos se observa al final el comando `wlx`, al digitar este comando, se ejecuta el código binario desarrollado en *C* dentro del entorno del dispositivo.

Con esto se observa el logro del objetivo de adicionar una cualidad adicional al enrutador, ya que se cuenta con una aplicación que puede disponer de los recursos del dispositivo hardware para cualquier finalidad y se descubre un sinfín de posibilidades con las cuales se puede controlar un dispositivo embebido. Esta es una funcionalidad adicional con la que se puede implementar desde un paquete para interactuar con una interfaz externa, hasta establecer políticas para brindar calidad de servicio e incluso un servicio nuevo a quien aplicarlas.

Es de resaltar la gran importancia que brinda el código abierto, y en especial Linux, el cual ofrece la capacidad de interactuar con un dispositivo hardware de una manera ilimitada, en cualquier nivel que se requiera.

3.7 Compilación Nativa de un Paquete para el Nuevo Firmware

Es posible realizar la compilación para plataformas para dispositivos embebidos en forma nativa, esto quiere decir, compilar el paquete directamente en el entorno del dispositivo, utilizando el compilador nativo para *mipsel*, para esto se requiere ampliar la memoria notablemente, mínimo hasta 150Mb, lo cual se puede lograr insertando una memoria externa al dispositivo. Debido a que Linux ofrece el soporte para manejar diversos tipos de sistemas de archivos, el WRT54G/GS/GL ofrece la posibilidad de adaptar una memoria externa a sus puertos de propósito general (GPIO, General Purpose Input/Output), estos son chips de bajo costo que proporcionan puertos especiales de comunicación para sistemas embebidos, cada uno de estos puede ser configurado por software como dispositivo de entrada o salida y soportan la transmisión de datos serial.

La inserción de una memoria digital externa tipo *SD (Secure Digital*) en el hardware del dispositivo, se puede encontrar en varias paginas de internet como www.against.org, esta actividad no se realizó dentro del marco de este trabajo ya que implica insertar directamente la memoria tipo *SD* a los pines de los puertos *GPIO* y se corre el riesgo fallo en cualquier parte del proceso, lo que puede resultar en la avería de ambos dispositivos.

Ya que el compilador cruzado ofrece las características necesarias para el desarrollo de aplicaciones para el openwrt, se deja como información para el lector con imaginación todas las modificaciones hardware que se pueden realizar, como lo muestran varios proyectos donde se ha catalogado al WRT54G/GS como un dispositivo sin límites.

3.7.1 Obtención del *Toolchain* Nativo

Cuando se cuenta con más memoria disponible se puede proceder a descargar el *toolchain* para *mipsel* nativo, este se puede encontrar en (44), una vez descargado se descomprime en la unidad de almacenamiento externo, los archivos descomprimidos tienen un tamaño de 120Mb en memoria.

3.7.2 Configuración de la Memoria Externa

Para la configuración del dispositivo de memoria externa se cuenta con el siguiente script, el cual puede ser modificado dependiendo de las necesidades.

```
#!/bin/sh

# Terminar los procesos que consumen recursos innecesarios en el sistema
#killall logger
#killall syslogd
#killall telnetd
#killall crond
#killall klogd
#killall udhcpc
#killall httpd
# Desactivación de los módulos para el sistema de archivos ext3 y para
jbd(Journaling Block #Device).
#rmmod ext3
#rmmod jbd

#En este ejemplo se tienen dos particiones  1.ext2  2.swap
#Montaje del sistema de archivos con las opciones noatime (No actualizar
los tiempos de acceso inode, para mayor velocidad de acceso) y async
(método de acceso asincrónico).

mount /dev/mmc/disc0/part1 /mnt -o noatime async

# Creación de la partición swap
#swapoff -a
#mkswap  /dev/mmc/disc0/part2
#swapon  /dev/mmc/disc0/part2

#Cambio del punto de montaje de la part1
mount -o move /tmp /mnt/tmp

echo " *** exit *** to back - Para volver al sistema"

#Establecer /mnt/ como raiz
chroot /mnt/ /bin/ash -
echo " ***  De vuelta al sistema original ***"

mount -o move /mnt/tmp/ /tmp/
umount /mnt
```

3.7.3 Compilación de las Fuentes

Luego de ejecutar este script, se debe hacer lo siguiente:

Localizarse en el directorio `/home`

Descargar la fuente del paquete por ejemplo `wlx-0.1.tar.gz`

```
tar -xvzf wlx-0.1.tar.gz

cd wlx-0.1

configure (1 minuto aproximadamente)

make (1 minuto aproximadamente)
```

Ahora se tiene un nuevo binario nuevo en el directorio `/src/`

Copiar el binario al directorio `/tmp`

```
exit
```

Ahora se puede copiar el binario al directorio `/usr/bin` con lo cual se puede ejecutar el programa `wlx`. Con lo cual se logra de manera nativa acceder mediante un programa desarrollado en lenguaje C a los recursos del dispositivo embebido.

A manera de recuento, en la primera parte de este capítulo se hace una introducción a la instalación del sistema operativo Linux el cual es la base fundamental sobre la que se desarrolla el proyecto. En la segunda sección se expone como se realiza la instalación y configuración de un host como punto de acceso. Luego se explica el proceso para llevar a cabo el desarrollo de la compilación e implementación de un firmware basado en el kernel de Linux y finalmente como se llevó a cabo la compilación e instalación de una de las características adicionales desarrolladas en el capítulo 2 para el nuevo firmware.

Después de realizar la implementación de un paquete ipkg, el cual es capaz de modificar la fragmentación de los paquetes que viajan por la red inalámbrica se evidencian las oportunidades que ofrece este tipo de desarrollos, se influye directamente en la capa de acceso al medio para mejorar un aspecto que no se había considerado, habilitar una opción que permite realizar la fragmentación dinámicamente para mitigar los inconvenientes causado por el ruido en la frecuencia en que trabaja el radio enlace. Desde este momento se puede inferir que se pueden realizar desarrollos en cualquiera de las capas hacia arriba del modelo OSI, esto es, desde la capa de acceso al medio hasta la capa de aplicación.

Ahora es posible realizar las mediciones del prototipo obtenido y las comparaciones con el firmware Linksys original y el hostAP[6], con miras a obtener buenos resultados en cuanto a lo desarrollado.

En este momento se encuentra operativo el firmware, con las características configuradas mediante el menú de configuración, se realizan las pruebas que indican su buen funcionamiento y la estabilidad del sistema, estas pruebas son en condiciones ideales ya que no existen más

[6] Software capaz de realizar las funciones de un punto de acceso en un computador.

dispositivos inalámbricos cerca y el ruido se encuentra en niveles bajos y normales, alrededor de -96dBm. El resultado de éstas y otras pruebas se consignan en el capítulo 4.

A partir de este momento se cuenta con una base, un kernel que funciona de la manera esperada y confiable, es aquí cuando se puede comenzar a experimentar con él, modificarlo, personalizarlo, añadir paquetes que incluyan cualidades adicionales o mejoras a las ya existentes, es posible incluir paquetes que incorporen los últimos adelantos en el manejo de QoS, clientes y servidores para los últimos desarrollos en seguridad, manejo de SNMP y muchos otros, esta es una base que da opciones ilimitadas para descubrir e implementar aplicaciones que mejoren el desempeño de esta clase de dispositivos, teniendo el control desde la capa 2 del modelo OSI, hasta la de capa de aplicación, esta es una gran herramienta, solo hace falta descubrir las posibilidades que puede brindar.

De acuerdo con el capítulo 2 se realizó la implementación de un módulo *ipkg* buscando realizar el control activo de la fragmentación sobre un dispositivo punto de acceso para evitar la pérdida de paquetes debida al ruido presente en el medio. Las implementaciones de QoS garantizan determinada calidad de servicio ya sea por puertos, protocolos etc., a una determinada estación o aplicación. Pero no se trabaja sobre el canal en si, es decir, que si se tiene un radio-enlace con mala calidad, el cual presenta pérdidas de paquetes y retardos, se pierde la garantía que ofrece QoS, esta se encuentra supeditada a la estabilidad del canal.

Lo que se logró con este trabajo fue incluir un módulo capaz de mitigar los efectos del ruido utilizando para ello una herramienta que proporcionó el protocolo 802.11 pero que se configura de manera manual, volviéndola automática, esto es un control de fragmentación dinámico para casos donde se evidencie la presencia de ruido en el medio de transmisión. Es posible realizar aun mayores consideraciones respecto al ruido, y generar herramientas que contribuyan a mitigar este tipo de inconvenientes que se presentan hoy en día, esto sería un gran adelanto para el desarrollo de las redes inalámbricas en el mundo entero, con este proyecto se abren las puertas a este desafío a la Universidad del Cauca, y se proporcionan las herramientas básicas para seguir con su desarrollo.

Es conocido de manera oficial que ingenieros ex-trabajadores de Alvarion, una de las más reconocidas marcas a nivel mundial en investigación y desarrollo para redes inalámbricas fundaron una empresa y crearon un producto bajo la marca InspiAir. Este dispositivo presenta características innovadoras sobre el estándar 802.11b, tales como un área de cobertura más extensa, una alta inmunidad al ruido, *hand-off* transparente, un método de *spanning tree* entre los puntos de acceso, entre otros beneficios, que además siguen aumentando a medida que se desarrolla el producto.

InspiAir está basado en un algoritmo privativo llamado *Virtual Transmitting Management,* VTM por sus siglas en ingles, el cual es el núcleo de la tecnología. Estos puntos de acceso son montados sobre una plataforma dada de baja por Alvarion, y funcionan con CPE´s o tarjetas inalámbricas estándar, con los cuales se consiguen resultados sorprendentes en cuanto a la utilización del canal en presencia de ruido. Por supuesto el secreto de los ex ingenieros de Alvarion no es revelado pero los resultados saltan a la vista, según las pruebas realizadas InspiAir es posible trabajar en un solo canal todas las celdas de una ciudad. En junio de 2006 se realizaron pruebas en la ciudad de Cali por un representante de InspiAir obteniendo excelentes resultados. Una de las pruebas realizadas demostró tiempos de respuesta al ping menores a 7ms en presencia de otros puntos de acceso en el mismo canal.

Este es un ejemplo claro de las ventajas que se pueden lograr modificando el firmware de dispositivos activos de red, no es necesario modificar el hardware para lograr una ventaja sobre los competidores. Es necesario desarrollar un estudio detallado sobre el estandar, el cual pueda aportar las bases requeridas para crear una aplicación capaz de resolver los problemas a los cuales se enfrentan las redes inalambricas en el mundo.

Este es un camino donde faltan muchas cosas por descubrir y que según la experiencia se pueden desarrollar, uno de los aportes de este trabajo de grado demuestra como en presencia de ruido y utilizando el modulo para el control de la fragmentación dinámica, se puede mitigar el ruido en dispositivos punto de acceso estándar de una manera visible, los tiempos de respuesta pasan de 500ms en promedio a menos de 10 ms en promedio, además de mostrar el camino para los desarrolladores ya que tienen la posibilidad de modificar un firmware de manera ilimitada. Solo es cuestión de imaginación para seguir descubriendo mayores aplicaciones e incluso se está en la capacidad realizar modificaciones al estándar.

4 PRUEBAS DE LOS DISPOSITIVOS DE RED INALAMBRICA COMERCIAL Y LOS PROTOTIPOS SOFTWARE Y FIRMWARE DESARROLLADO

4.1 INTRODUCCIÓN A LAS PRUEBAS DE DESEMPEÑO

Después de desarrollar el contenido teórico/práctico de este documento, se plantea a continuación un esquema de pruebas y resultados basados en comparaciones de rendimiento con dispositivos del tipo punto de acceso funcionando con firmware original de los equipos Linksys wrt54g, el firmware desarrollado por el grupo de trabajo (desarrollo, compilación y creación de paquetes) y un computador con *madwifi* funcionando como un punto de acceso para soportar usuarios corrientes.

La diferencia de rendimiento entre los diferentes tipos de firmware es notoria a la hora de actuar en condiciones normales, del mismo modo que la implementación de los aportes en software para garantizar mejor estabilidad y el mayor beneficio a la hora de trasmitir y recibir información.

4.1.1 Plan de Trabajo

En el siguiente listado se muestra el plan de trabajo seguido para la organización de las prácticas y toma de resultados:

- Consecución y ubicación de los equipos.
- Configuración de los equipos con diferentes firmwares y sistema operativos.
- Instalación de los paquetes creados en el firmware desarrollado por el grupo de trabajo (*ucfrag y ucrts*[7])
- Instalación de Sistema Operativo Linux con *madwifi*
- Instalación del Software para transmisión y recepción de tráfico.
- Instalación del Software para la medición y comparación de las características.
- Configuración de Software según las condiciones
- Inicio de diferentes tipos de transmisiones en los diferentes ambientes.
- Toma de Resultados.

4.1.2 Iniciación de Pruebas

Teóricamente estas pruebas se basaron en condiciones como las mostradas a continuación:

Inicialmente se realizaron pruebas de desempeño en transmisión de cada uno de los equipos con los diferentes firmwares. Estas pruebas se efectuaron con unas condiciones iniciales básicas, se configuró un punto de acceso y un cliente inalámbrico mediante una tarjeta de red 802.11g (a 54 Mbps) y un equipo servidor conectado a un punto de acceso vía ethernet (a 100 Mbps) con firmware original y con el firmware ucwrt[8], al igual que un cliente vinculado inalámbricamente al un computador con *madwifi* funcionando como punto de acceso como se muestra en la Figura 27.

---

[7] Paquete *.ipkg* desarrollado por el grupo de trabajo para implementar el el firmware prototipo.
[8] Firmware prototipo basado en el kernel de Linux, creado por el grupo de trabajo.

Mediante el software *LantrafficV2 [33]* se ejecutaron las pruebas de máximo *throughput* (MT) donde se generó tráfico desde la tarjeta de red ethernet de un servidor y lo trasmitió a los puntos de acceso (con los diferentes firmwares) y luego fue recibido por la tarjeta de red inalámbrica mostrando el máximo desempeño.

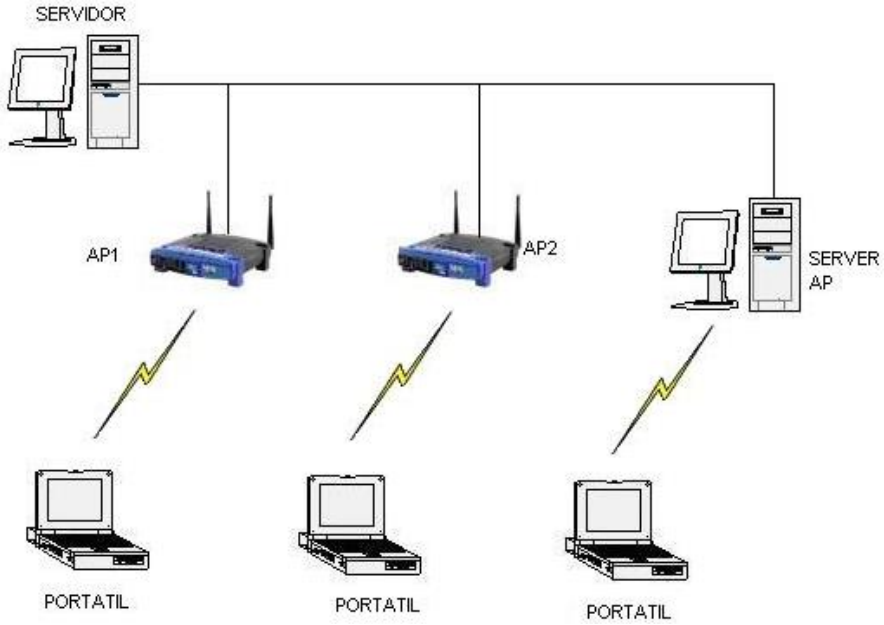

Figura 27. Arquitectura de Configuraciòn de Equipos

Con el software se entregó 2 tipos de límites de paquetes desde la interfaz ethernet. Uno de 56.000 Kbps y otro de 35.000 Kbps, se realizó de esta manera considerando que la máxima transferencia en condiciones teóricas entre un punto de acceso y un cliente inalámbrico vinculado a este sobre el estándar 802.11g solo puede llegar a los 54 Mbps. El uso de los dos tipos de límites de paquetes se realizó fundamentalmente pensando que además de la entrega de información por medio inalámbrico era necesario considerar que el procesamiento en un punto de acceso aumenta notoriamente si el uso de sus interfaces es más alto, teniendo en cuenta que la carga del procesador del equipo punto de acceso podría variar y bajar el rendimiento. En el casos de los Linksys wrt54g, el procesador funciona a 200Mhz en condiciones normales.

Continuando con el plan de pruebas, es necesario tener en cuenta que al usar canales iguales o adyacentes, se producen efectos de interferencias por traslape de las bandas de frecuencia, las pruebas fueron hechas con equipos que trabajan en frecuencias libres del orden de los 2.4 GHz sobre el estándar 802.11g, en este caso específico se utilizó el canal 1 (2.412 MHz) con interferencias sobre el mismo canal o en el canal 2 (2.417 MHz). En caso donde dos o más equipos utilicen exactamente el mismo canal, se da el problema llamado técnicamente “terminal expuesto”, que se traduce en que un dispositivo impide la transmisión porque escucha la transmisión de otro dispositivo operando en la misma frecuencia y obviamente ocupando el canal, pero que está conectado a un punto de acceso diferente como se muestra en la Figura 28, a pesar de que el inicio de la transmisión se efectúa debido a que el canal seguiría ocupado por un tiempo no estimado, el envío y recepción de información se ve afectado directamente. De forma muy similar a este caso, se implementaron pruebas donde los canales de transmisión eran adyacentes y al encontrarse con portadoras diferentes, cada uno de los puntos de acceso debería detectar el otro como ruido. Otro problema típico que se genera en un ambiente donde se encuentran varias celdas o áreas de cubrimiento de un punto de acceso, es el problema de “nodo oculto” explicado

en el capítulo 2 de este documento donde el RTS/CTS se convierte en un parámetro y función muy importante para la mejora en la transmisión. Dado que en redes CSMA el método de acceso múltiple requiere la detección de portadora para determinar el estado libre u ocupado del medio, la incapacidad de realizar esta operación se traduce en un problema de desempeño.

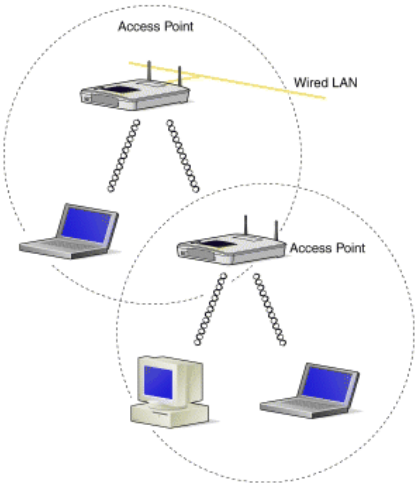

Figura 28. Puntos de Acceso Continuos

Debido a estos dos fenómenos se generan muchos inconvenientes para los usuarios, por ejemplo una baja en la tasa efectiva de servicio y retardo excesivo en la transmisión de paquetes. Se estableció un escenario en el cual se presentó el problema de "Terminal expuesto" cuando hay dos punto de acceso y el escenario donde se involucra el problema del "Nodo Oculto", comparando los resultados obtenidos en forma empírica con el mejor desempeño que puede esperarse desde un punto de vista teórico.

Adicionalmente se analizó el fenómeno de las interferencias causadas por la utilización de canales adyacentes. Para realizar las mediciones se propuso un método empírico para obtener los parámetros de relevancia de una red Inalámbrica, escenario que se amplía y complementa con condiciones donde se incluyó el problema del "Nodo oculto". Los resultados obtenidos de este trabajo demostraron que en un ambiente de múltiples celdas el uso de RTS/CTS mejoró la utilización del canal, manteniendo un equilibrio en el acceso entre los dispositivos clientes en un ambiente de múltiples celdas. Con respecto a la utilización de canales intermedios, se concluyó que el utilizarlos da como resultado una significativa disminución del *throughput* en las transmisiones pero más estabilidad a la hora de encontrarse un nodo oculto.

De este modo el desarrollo de un firmware capaz de mantener la estabilidad en uno o más enlaces inalámbricos y de soportar los cambios en la fragmentación y el RTS/CTS de manera automática se convierte en un aporte fundamental ya que potencializa la utilización de los medios inalámbricos para la transmisión de datos.

Después de culminar la etapa de desarrollo, el resultado de la misma es un firmware basado en software libre. El firmware llamado Universidad del Cauca WRT (*ucwrt*), es capaz de soportar la adición de nuevas características, paquetes de software y ser más estable, además que como se obtiene desde las fuentes de código tanto de hardware como del código original de *openwrt* se tiene la posibilidad de crear un firmware inicial (archivo *.bin*) con capacidades personalizadas de acuerdo a la necesidad, todo lo anterior se probó en laboratorio y se muestra en las figuras siguientes de este documento.

Con este aporte funcional y operativo se evidencia el cumplimiento de uno de los objetivos donde se trazó la creación de un prototipo de firmware capaz de soportar características especiales. En la Figura 29, se muestra el entorno de administración vía WEB del *ucwrt*, desde donde se pueden realizar varios cambios y mejoras en la interfaz, sin embargo, el firmware tiene la capacidad de ser operado mediante consola.

Figura 29. Consola de Administración WEB ucwrt

La Figura 30 muestra el entorno de Administración con conexión de *SSH* (*Secure Shell*), todo el firmware es basado en comandos y su administración permite realizar cualquier cambio sobre consola del mismo modo que hacer la instalación y desinstalación de paquetes adicionales.

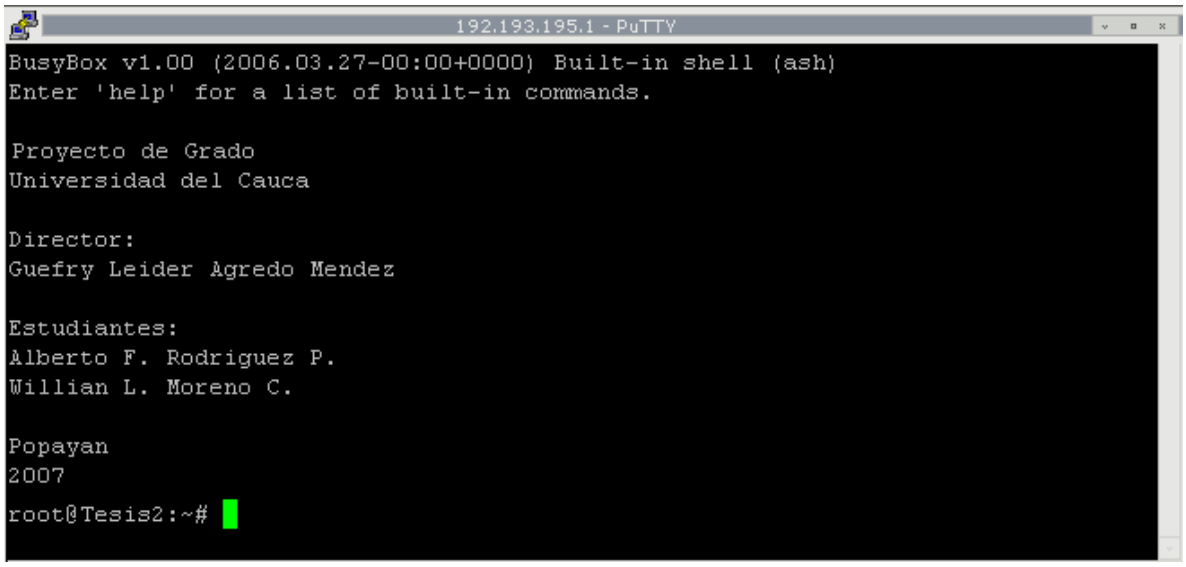

Figura 30. Consola de Administración Por SSH ucwrt

Los equipos Linksys wrt54g, originalmente vienen con un firmware con código y programación propietaria, todo el control y administración, dependen de una interfaz web, tal como se muestra en

la Figura 31. Debido a que únicamente se pueden realizar ajustes sobre la interfaz web, la potencialidad del equipo no es aprovechada en su totalidad. Esto hace que el desarrollo del firmware *ucwrt* tenga un valor agregado bastante importante y además de que sus raíces son basadas en software libre.

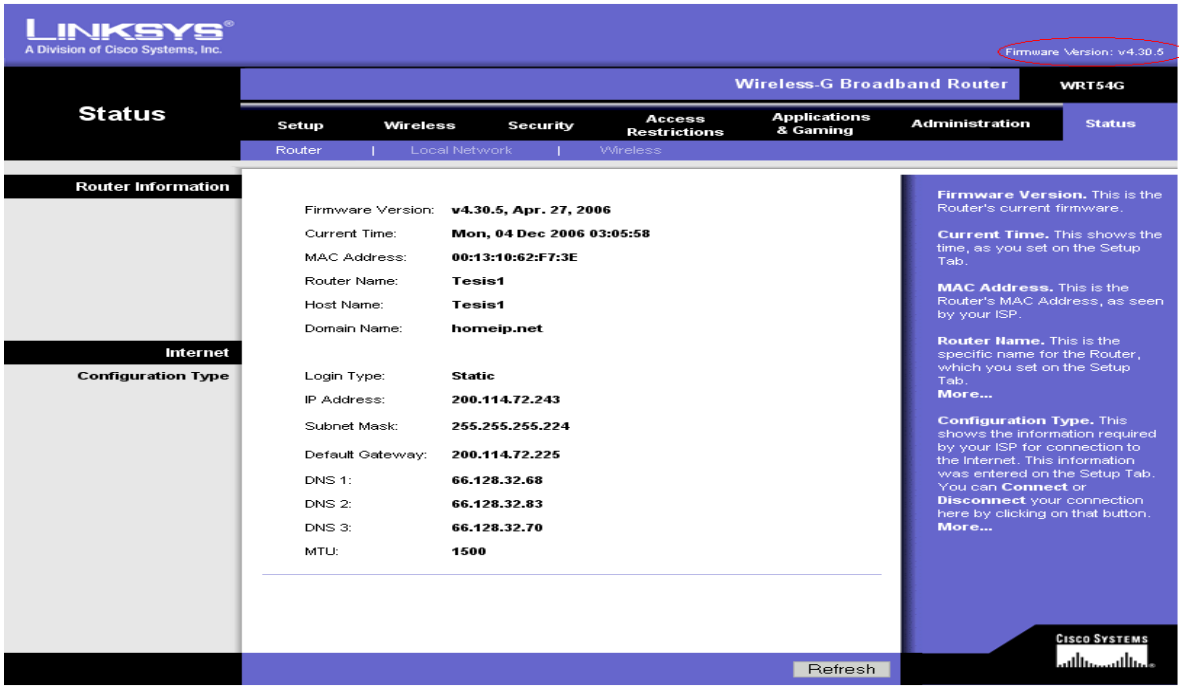

Figura 31. Consola de Administración WEB Linksys Original

- Configuración de punto de acceso en un computador

Debido a que el *HostAP* ya es obsoleto y con regularidad funcionaba en tarjetas de red inalámbricas con chips PRISM, y con la necesidad de cumplir el objetivo de la comparación de los diferentes tipos de equipos y firmware, se realizó la comparación con un software más avanzado y con la capacidad de tener y manejar varias características necesarias para un buen funcionamiento de las redes de este tipo. El software utilizado es *MADWIFI* y es también basado en código abierto.

Madwifi no funciona sobre todas los dispositivos hardware inalámbrico, en el caso de los adquiridos por el grupo de trabajo, habían dispositivos cuyo chip era producido por la compañía *Atheros* y donde los controladores de Madwifi funcionan perfectamente.

Las fuentes y paquetes de MadWifi se adquirieron en el sitio web:

`http://www.madwifi.org`

Una vez obtenido e instalado los paquetes y código se realizó la instalación del software y las librerías de control y gestión inalámbrica que el software necesita para funcionar, estas librerías vienen en los CDs de instalación de la distribución de Linux, en este caso *Suse* 10.1.

Una vez instalado el software se ejecutaron las líneas de comando para configuración:

```
#modprobe ath_pci
```
(Donde se instaló el módulo para el funcionamiento del hardware de la tarjeta de red.)

```
#wlanconfig ath0 destroy
```
(Para destruir cualquier interfaz que estuviera creada por un controlador anterior)

```
#wlanconfig ath0 create wlandev wifi0 wlanmode ap
```
(Para configurar la interfaz ath0 en modo Punto de Acceso)

De este modo se pudo obtener configurado un computador como punto de acceso, obviamente fueron necesarios otros parámetros para la configuración total de la interfaz.

A continuación se muestra el resultado después de escribir el comando `iwconfig` en la consola del computador

```
ath0      IEEE 802.11g  ESSID:"TesisUdC3"  Nickname:"unicauca"
          Mode:Master       Frequency:2.422    GHz       Access    Point:
          00:0D:88:C6:E0:02
          Bit Rate:0 kb/s   Tx-Power:18 dBm   Sensitivity=0/3
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=48/94  Signal level=-47 dBm  Noise level=-95 dBm
          Rx invalid nwid:7316  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Con el comando `iwconfig` se listan las interfaces y muestra cuales son inalámbricas y su configuración. Como se mostró en el capítulo 1, existen diferentes formar de configurar una interfaz inalámbrica, en este caso es necesario que esta funcione en modo *Master* para que opere como un Punto de Acceso.

Por ejemplo una configuración básica de la interfaz fue:

```
#iwconfig ath0 essid "TesisUdC3" channel 3 key off
```

(donde se configuró el nombre de broadcast del AP, el canal de funcionamiento y la opción de apagado para la clave wep)

Una vez configurada la interfaz inalámbrica y considerando que se debió concluir la configuración de un Punto de Acceso fue necesario que se establecer una configuración Bridge entre las interfaces donde operó el PC (Punto de Acceso), para que esto se pudiera realizar, se adicionó una interfaz llamada br0 en el PC para que esta hiciera de puente entre la interfaz inalámbrica y la ethernet del PC.

Los siguientes comandos muestran como se configuró la interfaces de manera funcional:

```
#brctl addbr br0
#brctl addif br0 ath0
#brctl addif br0 eth0
#ifconfig ath0 0.0.0.0 promisc
#ifconfig eth0 0.0.0.0 promisc
#ifconfig br0 up

unicauca:~ # ifconfig br0 192.193.194.3 netmask 255.255.255.0 up
```

La configuración final de las interfaces en el computador con Linux Suse, allí se muestran las interfaces inalámbricas, ethernet y tipo bridge (br0).

```
ath0      Link encap:Ethernet  HWaddr 00:0D:88:C6:E0:02
          inet6 addr: fe80::20d:88ff:fec6:e002/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:253081 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6955762 errors:0 dropped:83 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:26951536 (25.7 Mb)  TX bytes:1555676910 (1483.6 Mb)

br0       Link encap:Ethernet  HWaddr 00:0B:6A:0D:D2:DE
          inet addr:192.193.194.3  Bcast:192.193.194.255
          Mask:255.255.255.0
          inet6 addr: fe80::20b:6aff:fe0d:d2de/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:383060 errors:0 dropped:0 overruns:0 frame:0
          TX packets:340356 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:18683965 (17.8 Mb)  TX bytes:510466443 (486.8 Mb)

eth0      Link encap:Ethernet  HWaddr 00:0B:6A:0D:D2:DE
          inet6 addr: fe80::20b:6aff:fe0d:d2de/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:7131376 errors:0 dropped:0 overruns:0 frame:0
          TX packets:484926 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1590954056 (1517.2 Mb)  TX bytes:535429308 (510.6 Mb)
          Interrupt:11 Base address:0xd800

eth2      Link encap:Ethernet  HWaddr 00:08:A1:9D:E5:E8
          inet addr:192.193.195.3  Bcast:192.193.195.15
          Mask:255.255.255.240
          inet6 addr: fe80::208:a1ff:fe9d:e5e8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5702 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5295 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:796681 (778.0 Kb)  TX bytes:813474 (794.4 Kb)
          Interrupt:12 Base address:0xec00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:9342 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9342 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:731212 (714.0 Kb)  TX bytes:731212 (714.0 Kb)

wifi0     Link encap:UNSPEC  HWaddr 00-0D-88-C6-E0 02-00-00-00-00-00-00
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:330716 errors:0 dropped:0 overruns:0 frame:1431793
```

```
TX packets:5337757 errors:3475 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:199
RX bytes:36630157 (34.9 Mb)  TX bytes:1020585156 (973.3 Mb)
Interrupt:10 Memory:d0ee0000-d0ef0000
```

4.2 HARDWARE Y SOFTWARE UTILIZADO PARA PRUEBAS DE RENDIMIENTO

El hardware utilizado para la ejecución de las pruebas esta descrito en la tabla 3.

Tabla 3. Equipos Utilizados para Pruebas

Cantidad	Descripción	Interfaz Eth	Interfaz Wifi
1	Computador Desktop Intel Pentium 4 HT	S2/1 - 100Mbps	Si/54Mbps/Dlink dwl-g520
1	Computador Desktop Intel Pentium D - ASUS	Si/2 - 100Mbps	Si/54Mbps/Dlink dwl-g520
1	Computador Desktop AMD XP 2600	Si/2	No
2	Portátiles Toshiba	Si/100Mbps	Si/54Mbps/Intel Pro
1	Portátil HP	Si/100Mbps	Si/54Mbps/Linksys
4	WRT54G	Si/5 100Mbps	Si/54Mbps/Linksys

En la Figura 32 se muestran se muestran algunos de los dispositivos utilizados para las pruebas y desarrollo del firmware (ucwrt), paquetes (ucfrag y ucrts) entre otras pruebas y desarrollos.

Figura 32. Algunos de los Equipos Utilizados para el Proyecto

El software utilizado para la ejecución de las pruebas se muestra en la Tabla 4:

Tabla 4. Software Utilizado para Pruebas

DUMeter	Software para medir tráfico
LanTraffic V2	Software para generar y recibir trafico
Network stumbler	Software detector de redes 802.11x
Ping Plotter	Software para graficar tiempos de respuesta al ping y jitter

4.3 ESCENARIO DE PRUEBAS Y COTAS

Antes de iniciar las mediciones experimentales, es importante calcular el Máximo *Throughput* (MT) que puede esperarse teóricamente en un escenario ideal en que un dispositivo móvil se comunica con un Punto de acceso. El cálculo inicial se hizo basado en la no existencia de interferencias de otro dispositivo. Para obtenerlo, es necesario tener en cuenta:

1. El canal debe estar libre de errores (Se garantiza con software de análisis de red inalámbrica la no interferencia de otros canales). En la Figura 33 se puede observar que antes de iniciar las pruebas se realizó un escaneo del espectro en búsqueda de canales utilizados. El software utilizado garantiza que no haya conexiones inalámbricas utilizando el estándar 802.11 en un área de cobertura donde se encuentra la tarjeta de red inalámbrica donde se ejecutó la aplicación. Sin embargo, es importante aclarar tal como se mostró en el documento que existen dispositivos domésticos como hornos microondas, teléfonos inalámbricos, dispositivos bluetooth, teclados y ratones Inalámbricos que también utilizan frecuencias en el orden de los 2.4 GHz, los cuales son capaces de ocasionar interferencia, pero al no estar utilizando las misma portadora de los canales del estándar 802.11 no pueden ser detectados por el software utilizado.

El software de Netstumbler muestra cuando un punto de acceso está operando en modo Máster es decir funcionando como punto de acceso y es capaz de recibir conexión de clientes de manera inalámbrica.

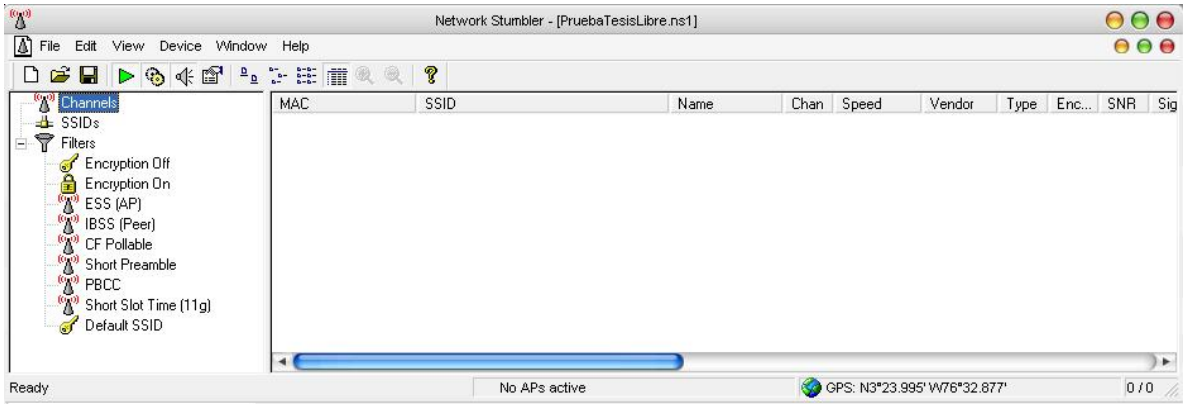

Figura 33. Software Netstumbler – Todos los canales libres

2. Para iniciar la transmisión se confirmó la existencia de un solo punto de acceso activo con SSID activo (TesisUdC2) y solamente un dispositivo cliente capaz de recibir los paquetes enviados desde el punto de acceso, en este caso el cliente tenía a disposición una tarjeta de Red inalámbrica DLink DWL-G520. En la Figura 34 se puede observar que después de encender el punto de acceso solo se detecta el punto de acceso utilizado en las pruebas.

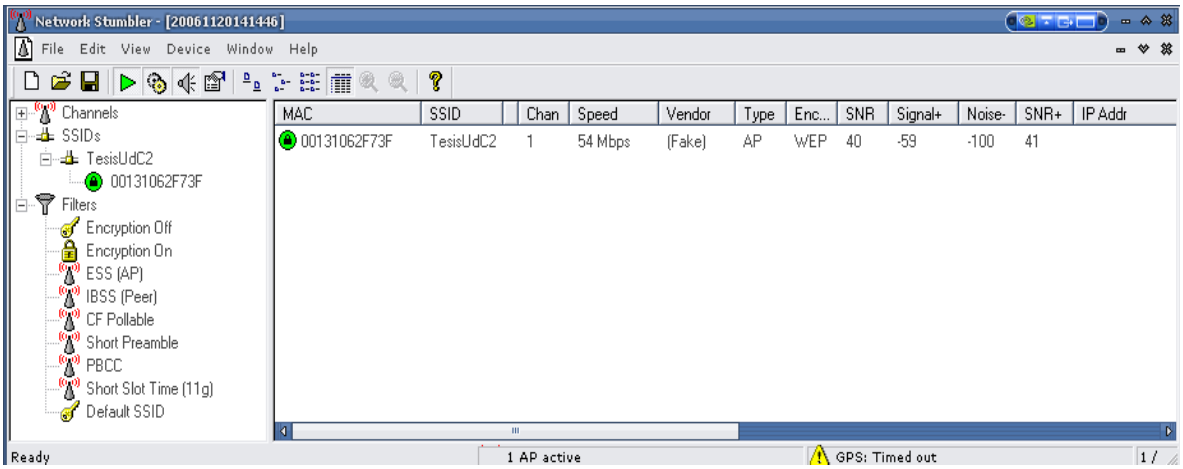

Figura 34. Software Netstumbler - Punto de Acceso (SSID : TesisUdC2)

Idealmente, el esquema utilizado se muestra en la Figura 35 donde la arquitectura de la red instalada es ideal para las pruebas.

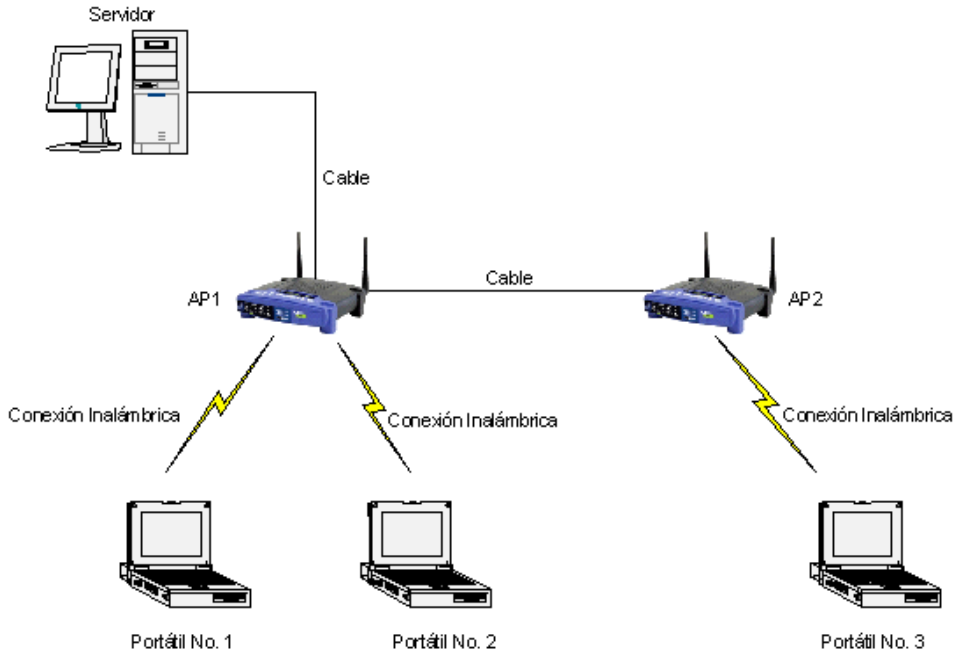

Figura 35. Disposición de Equipos para Pruebas

Se establecieron las cotas máximas esperadas en un ambiente experimental tomado como se explicó anteriormente sobre las dos mediciones, una de con tasa de transferencia a 35.000 Kbps Figura 36.

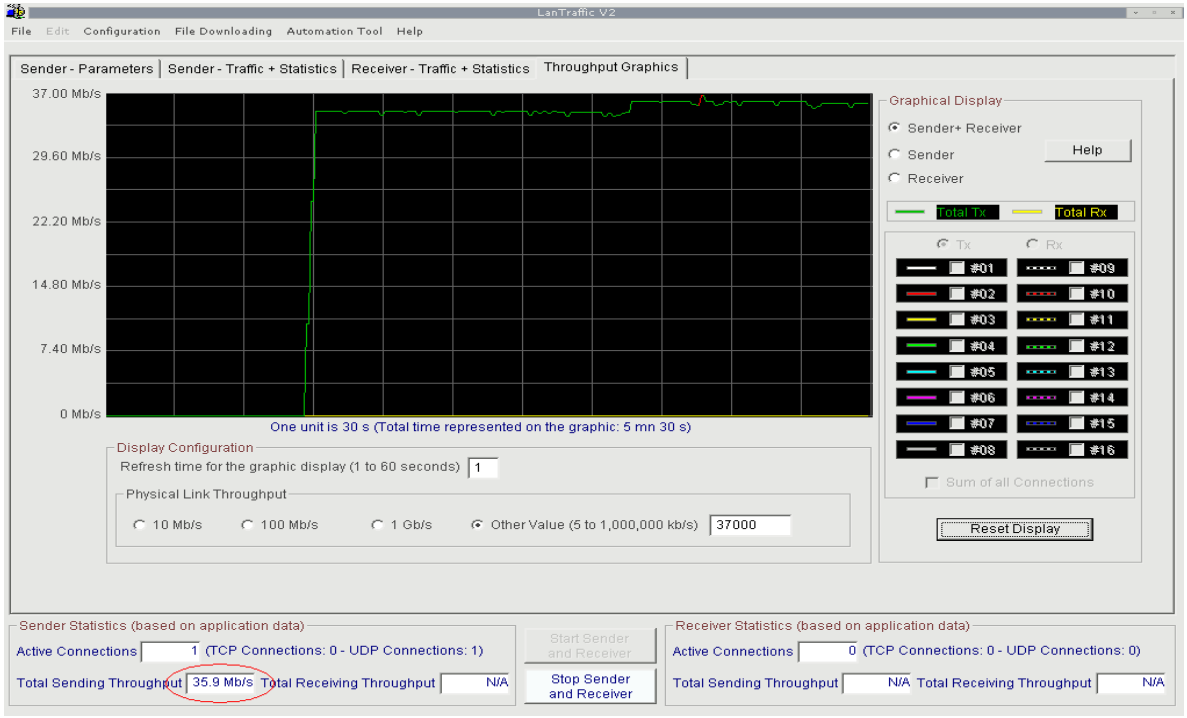

Figura 36. Emisión de Tráfico a 35.000 Kbps

Ambas tasas de transferencia fueron aplicadas a los dos tipos de equipos con diferente firmware, para de este modo tener un punto de referencia más claro sobre la transmisión y recepción en condiciones ideales.

El software *Lantraffic V2* es capaz de enviar y recibir tráfico tanto en TCP como en UDP en diferentes tipos de puertos. Para las pruebas realizadas específicamente se utilizó tráfico UDP debido a que mediante este protocolo la transmisión no es orientada a la conexión, de este modo se puede utilizar el canal en ambos sentidos sin recibir la confirmación de los ACK.

Los puertos por cliente utilizados fueron el 2020 y 2040 (pueden ser cualquiera) respectivamente, tal como se observa en la Figura 37. En la Figura No. 38 se muestran los cambios realizados para cambiar la tasa de transmisión.

Ambas muestras se realizaron con cada uno de los firmwares utilizados, tanto el original de la compañía Linksys versión 4.30.5 para el punto de acceso, como el desarrollado de nombre *ucwrt* por el grupo de trabajo.

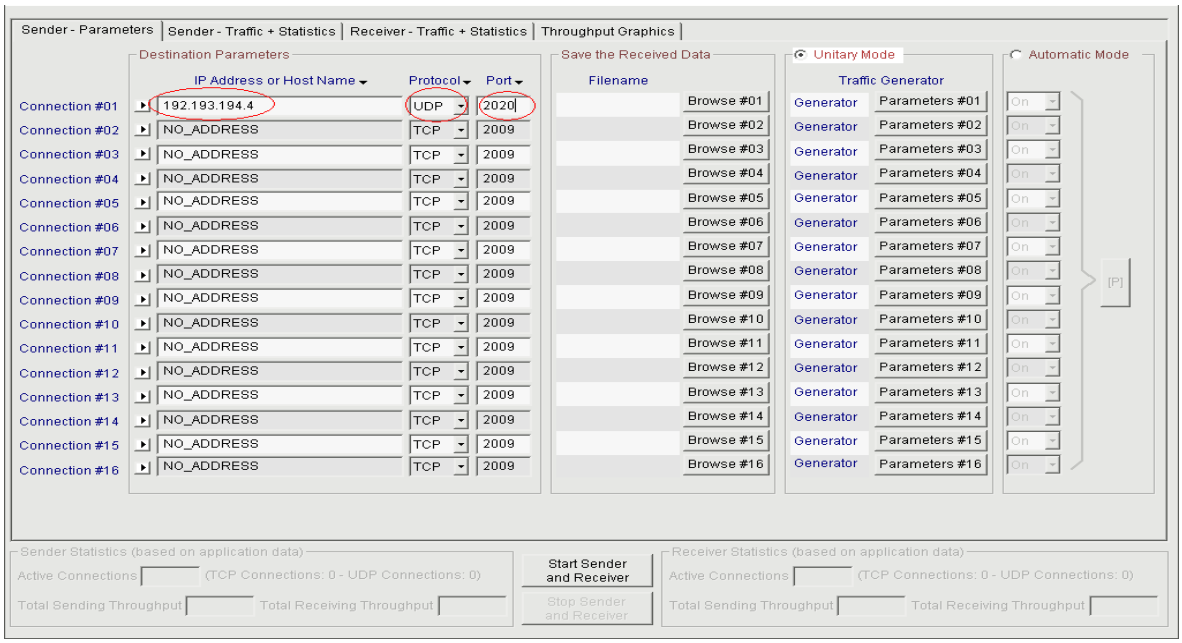

Figura 37. Configuración de IP, Protocolo y Puerto

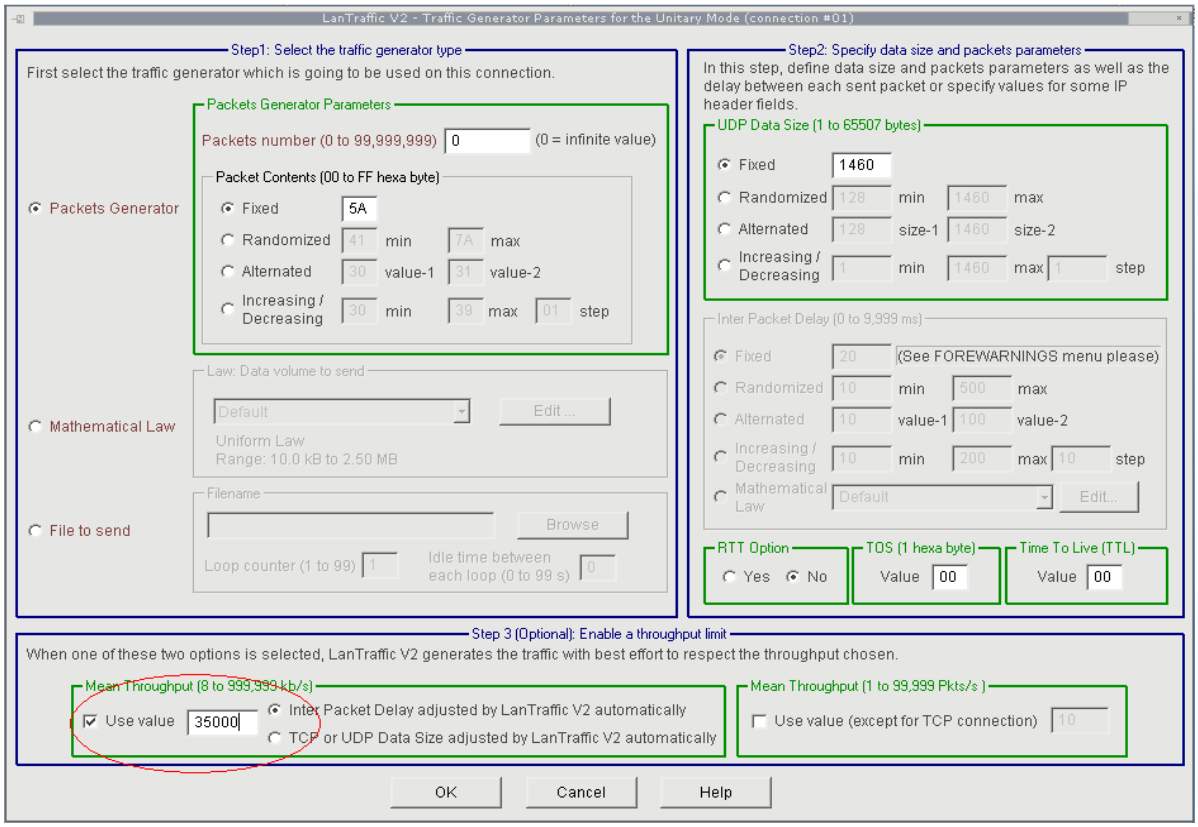

Figura 38. Configuración de Transmisión a 35.000 Kbps

Con el software *Netstumbler* se escaneó nuevamente el espectro de 2.4 GHz en busca de Canales utilizados, en la Figura 39 se observan los *SSIDs* detectados y los canales que cada uno tiene, en este caso se realizaron las pruebas en canales adyacentes (1 y 2).

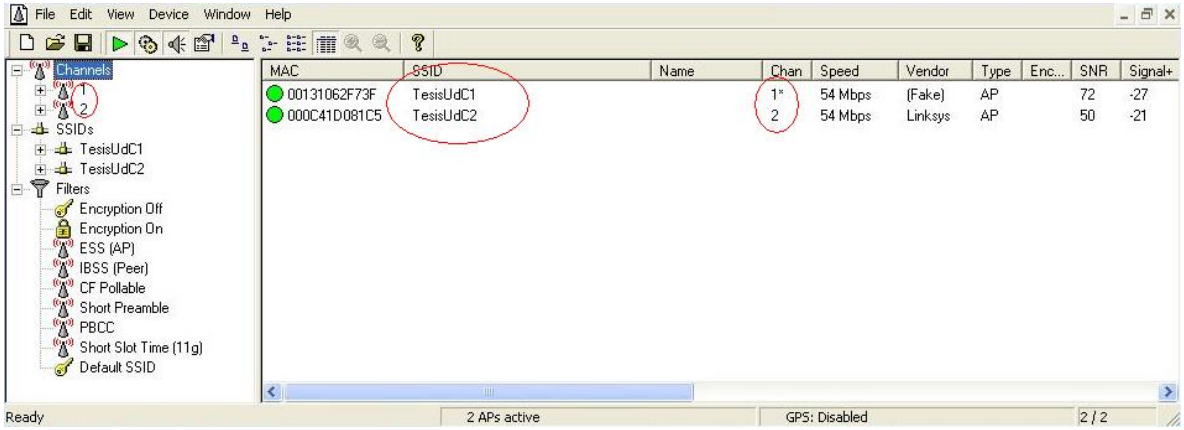

Figura 39. Detección de los dos Canales Adyacentes

En la Figura 40 Se puede observar el máximo *throughput* obtenido con una transmisión de 56.000 Kbps con el firmware Original de Linksys, esta trasmisión logró un MT de 28.8 Mbps. En la Figura 41 la transmisión generada fue de 35.000 Kbps y el MT de recepción fue de 30.8 Mbps. Existe una variación considerable en la recepción, esta variación se debe a que la tasa de transmisión alta hace que se presente un uso excesivo del procesador del punto de acceso cuando realiza el rechazo de paquetes, ocasionando una disminución en la transmisión, después de varias pruebas se pudo establecer que a medida que el tráfico se aumentaba en un puerto ya sea ethernet o Wifi de un punto de acceso, el procesamiento del equipo aumentaba, esto ocasionaba la disminución en rendimiento gradualmente, del mismo modo que al tener muchos paquetes y servicios instalados y operativos dentro del punto de acceso (básicamente en ucwrt), también se ocasionaba la caída en rendimiento por procesamiento de servicios.

Por lo anterior, se vuelve concluyente la necesidad de personalizar los paquetes y servicios necesarios para el funcionamiento de un punto de acceso que en la mayoría de ocasiones tiene en funcionamiento más características que las necesarias y adecuadas para un trabajo específico.

La emisión de tráfico se hace desde un PC conectado vía ethernet al punto de acceso, de este modo el equipo inalámbrico realiza únicamente el cambio de 802.3 a 802.11.

En el firmware ucwrt, la tasa máxima de transferencia fue de 28.7 Mbps, este valor es inferior al obtenido con el firmware original, sin embargo, al empezar a realizar cambios en el medio, los resultados obtenidos son notablemente inclinados hacia el ucwrt. En la Figura 42 se muestra la recepción con una tasa de envió a 35.000 Kbps y en la Figura 43 la recepción con una tasa de envió a 56.000 Kbps.

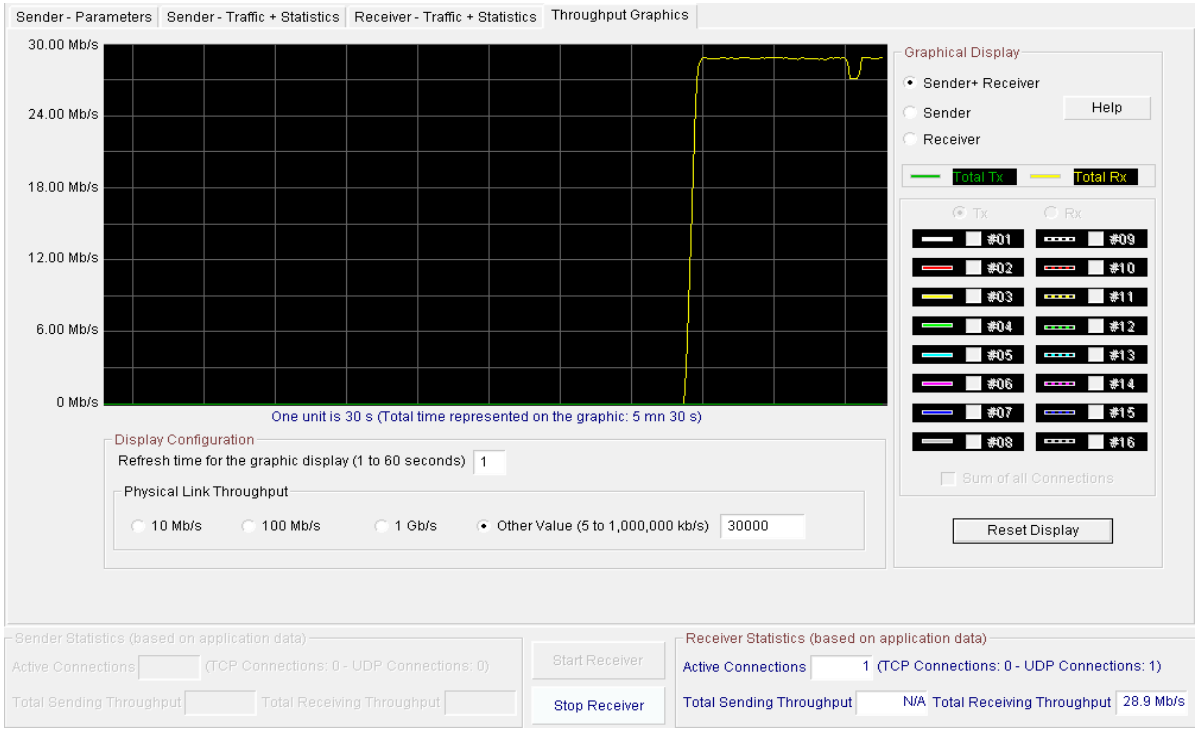

Figura 40. Recepción con Firmware Linksys Original 56.000 Kbps

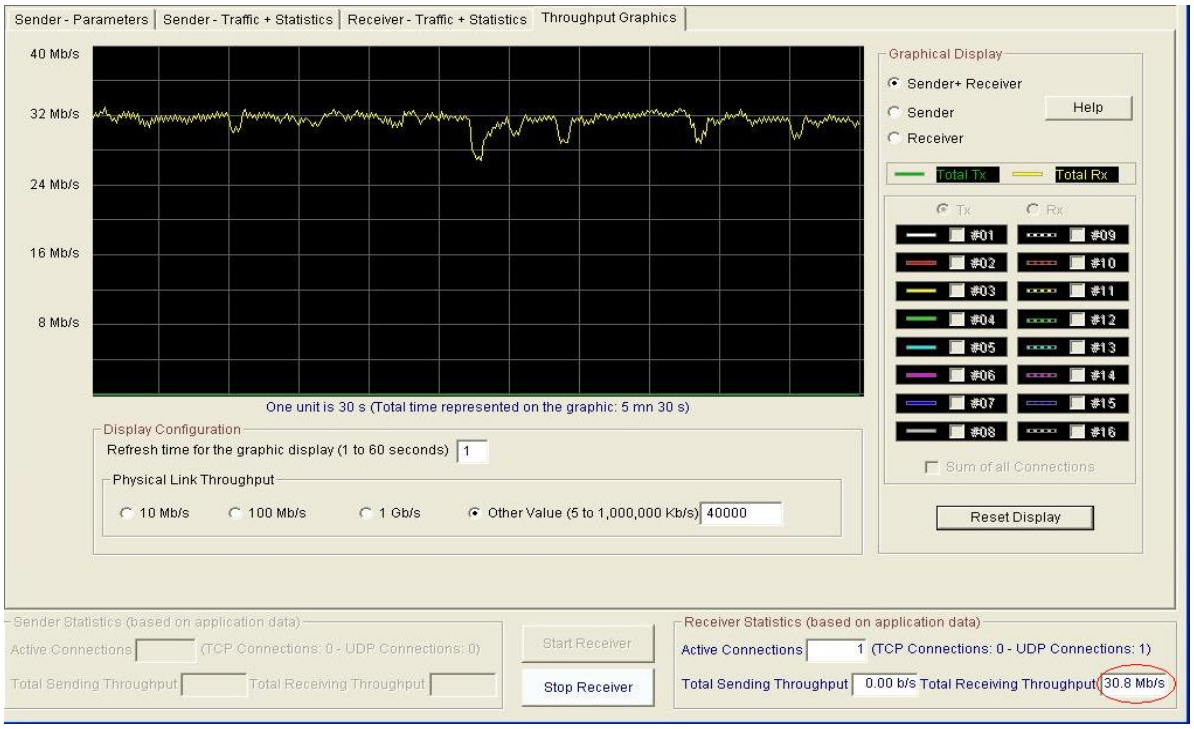

Figura 41. Máximo *Throughput* Linksys Firmware Original

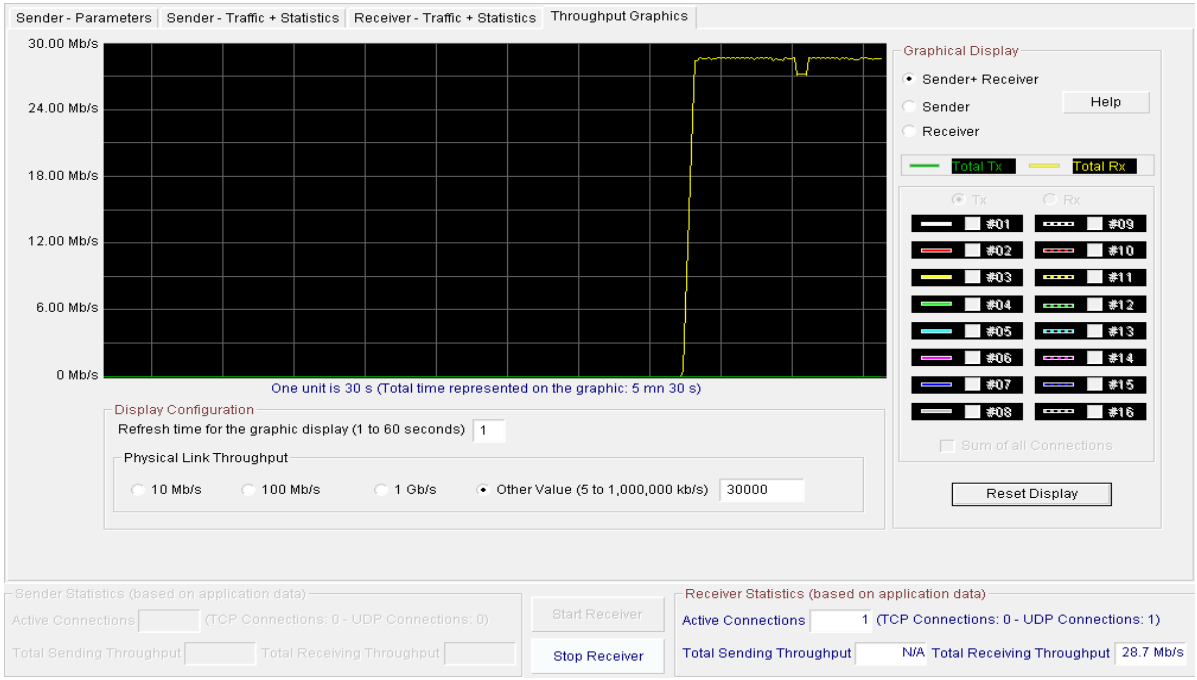

Figura 42. Máximo *Throughput* ucwrt Generado a 35.000 Kbps

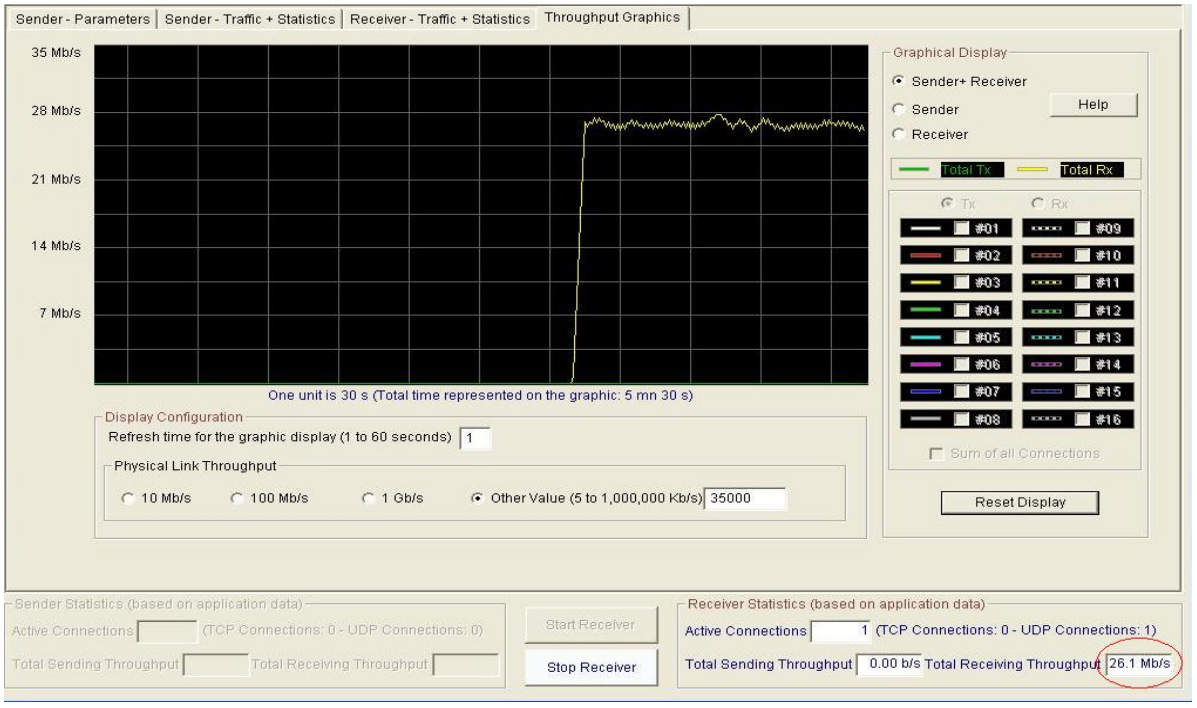

Figura 43. Recepción Firmware ucwrt Generado a 56.000 Kbps

Es visible que la recepción de paquetes en MadWifi es superior, ya que se mantuvo por encima y constante en los 32 Mbps, sin embargo, el costo de instalar y tener en funcionamiento un computador para este fin es alto. En la Figura 44 se muestra el resultado de la recepción con MadWifi.

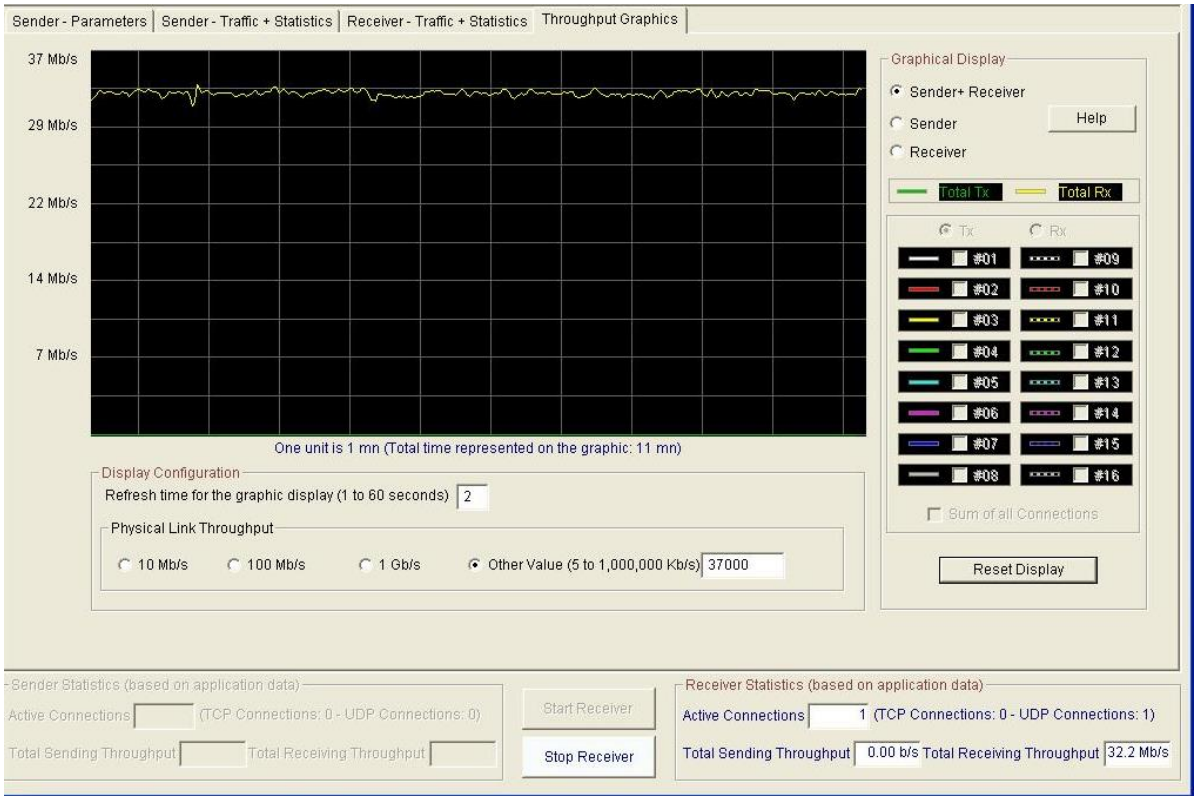

Figura 44. *Throughput* Máximo AP con MadWifi

Se construyó un escenario donde se realizaran por lo menos dos transmisiones, una para capturar la información de los paquetes transmitidos y otra para generar ruido. Un PC como servidor con dos tarjetas de red conectadas en redes diferentes, cada una de estas redes con un punto de acceso, uno para tomar las muestras y medidas y otro para generar ruido en transmisión.

En la Figura 45 se observa la generación del tráfico para cada tarjeta, uno con tráfico de 35.000 Kbps y otro con tráfico de 36.000 Kbps. La Figura 46 muestra los valores generados. Varían los tráficos enviados en 1000 Kbps únicamente para poder observar más claramente en las gráficas las emisiones de paquetes.

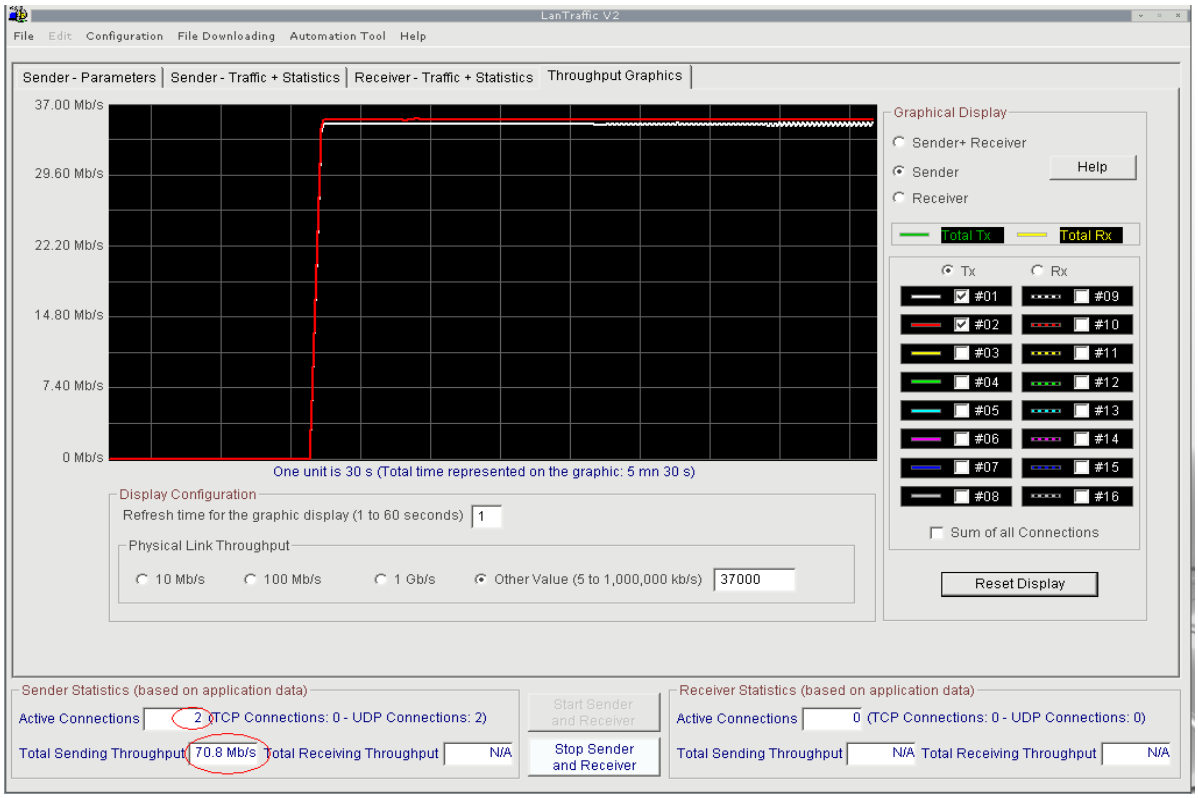

Figura 45. Generación de Trafico para dos Puntos de Acceso en Diferentes Redes

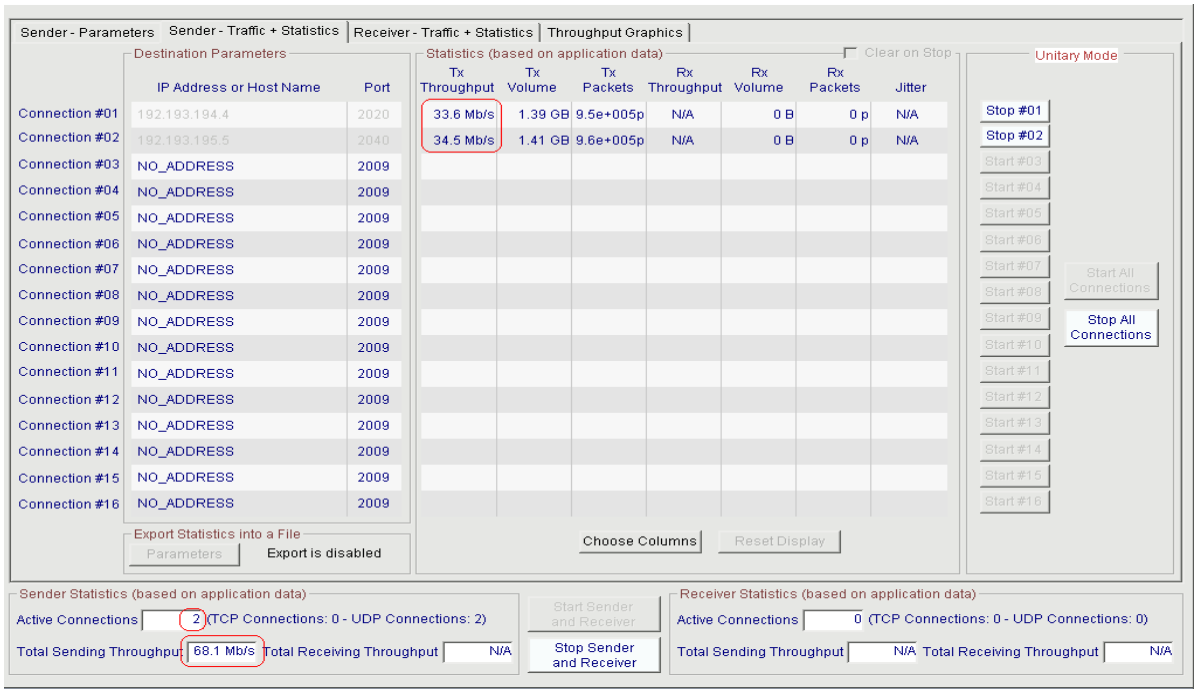

Figura 46. Muestra de Generación de Trafico para dos Puntos de Acceso

De acuerdo a los máximos valores de recepción, donde se logró los 30.8 Mbps, es decir el 57% aproximado de la capacidad teórica de red de 54 Mbps proporcionada por el punto de acceso WRT54G con cualquiera de los firmwares instalados, se planteó que la generación máxima de tráfico si es adecuada a 15.000 Kbps, de este modo poder disminuir el procesamiento del equipo a la hora de procesar los paquetes transmitidos y recibidos.

Ahora, después de haber obtenido el máximo *throughput* en los firmwares, donde es destacable el que ambos equipos son capaces de soportar un alto rendimiento con una conexión, es primordial mostrar lo ocurrido cuando el medio de transmisión empezó a ser cambiante, es decir, tuvo interferencias y variaciones en Terminal Expuesto y Nodo Oculto.

Para la generación de tráfico de información y ruido, se hizo con dos tipos diferentes de flujos de información, ambos UDP, el ruido a 30.000 Kbps y el tráfico a 15.000 Kbps. La disminución en la tasa máxima de transferencia a 15.000 Kbps fue realizada debido a que al iniciar una transmisión con Ruido el valor máximo fue cercano a los 15 Mbps.

En la Figura 47 se Observa la generación de tráfico. En este caso, inicialmente se realizó la generación de tráfico para toma de muestras y 20 segundos después la generación del ruido.

En la Figura 48 se muestra lo contrario, es decir, primero se general el ruido y luego el tráfico.

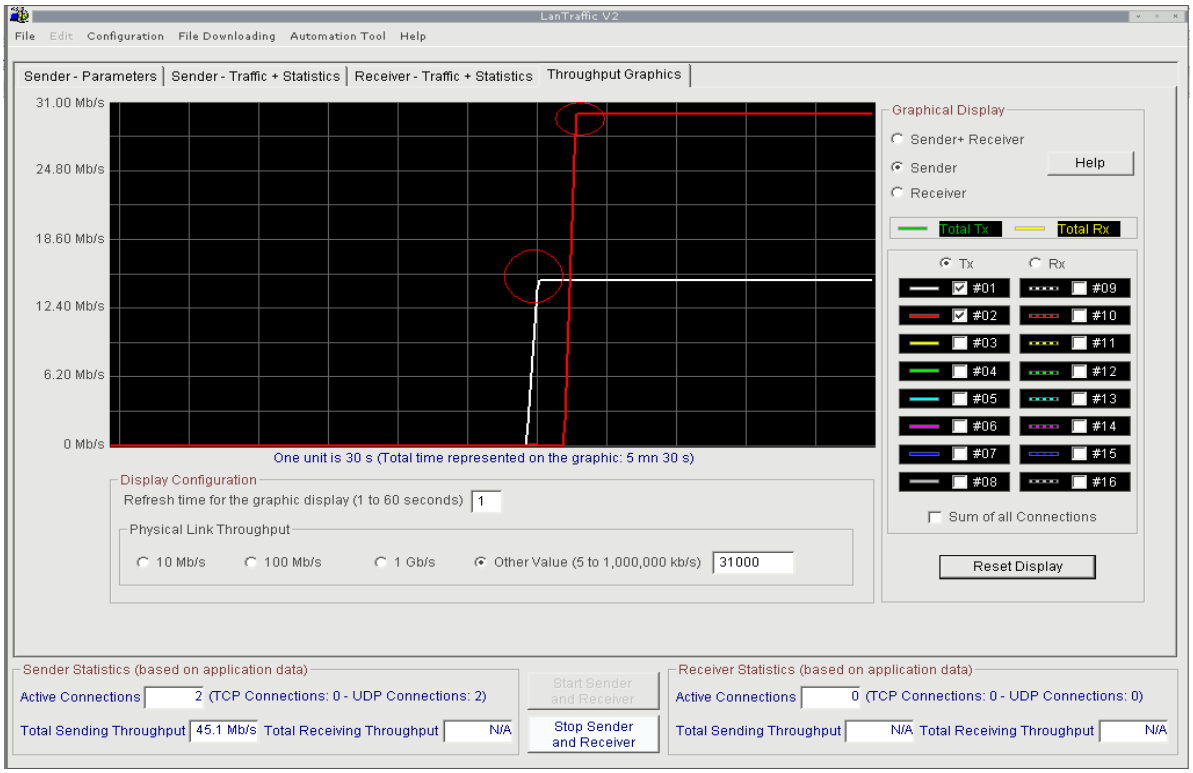

Figura 47. Generación de Tráfico y Ruido

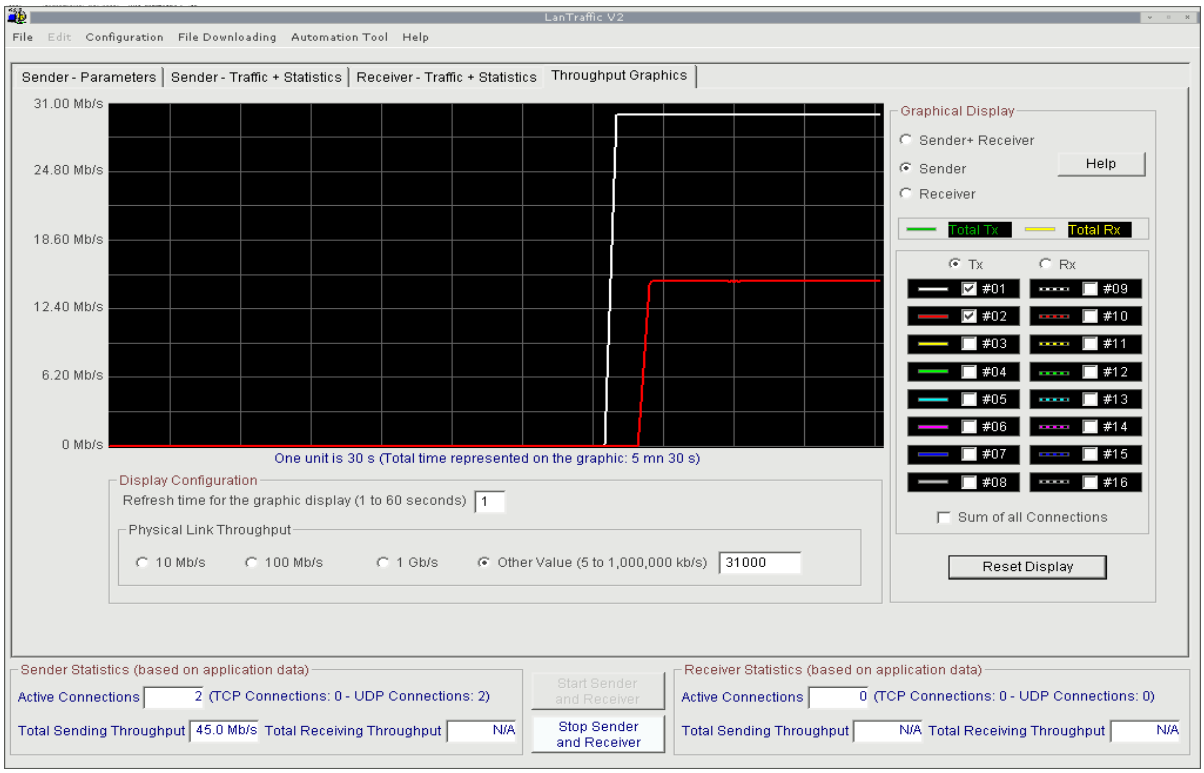

Figura 48. Generación de Ruido y Trafico

El cálculo tiene que considerar que el intercambio de mensajes entre el punto de acceso y el dispositivo cliente sigue un protocolo establecido. El protocolo básico de la Función de Coordinación Distribuida (DCF: Distributed Coordination Function) esto significa que una estación que recibe un paquete para ser transmitido por el medio inalámbrico, primero establece un valor inicial para un contador de cuenta regresiva que contabiliza los espacios que detecta libres. Cuando alcanza la cuenta 0 transmite. Esta cuenta la realiza mientras detecta el canal libre después de una transmisión y un tiempo adicional DIFS (Distributed InterFrame Space). Si su transmisión es recibida exitosamente, el destinatario emite un reconocimiento (ACK) después de un tiempo SIFS (Short InterFrame Space). En caso de estar en presencia del fenómeno de Nodo oculto, se precede al envío de datos de un intercambio RTS (Request To Send) enviado por el emisor siempre y cuando la característica este activa y disponible, seguido de un CTS (Clear To Send) del receptor, con tiempos entre mensajes SIFS, como se ilustra en la Figura 49.

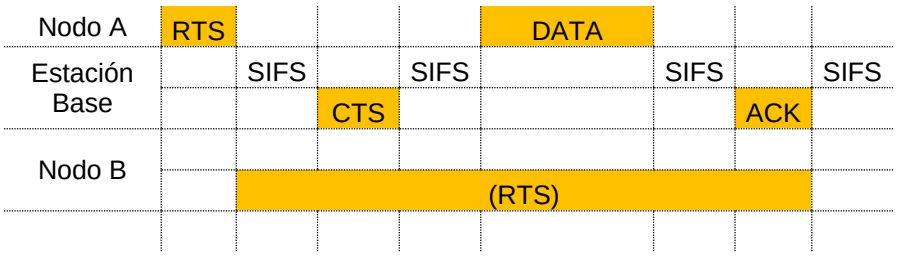

Figura 49. Protocolo RTS/CTS usado en IEEE 802.11

El *throughput* se obtiene entonces aplicando las siguientes ecuaciones:

Ecuación No. 1:

$$MT = \frac{L_{paquete}}{L_{paquete} / R_{datos} + T_{control}}$$

Ecuación No. 2:

$$T_{control} \mid_{\sin RTS} = \frac{CW_{\min}}{2} + 2T_{delta} + 2T_{PCLP} + \frac{ACK}{R_{datos}}$$

Ecuación No. 3:

$$T_{control} \mid_{conRTS} = \frac{CW_{\min}}{2} + 4T_{delta} + 4T_{PCLP} + \frac{RTS + CTS + ACK}{R_{datos}}$$

Donde la definición y valores de los parámetros de la ecuaciones de la (1) a la (3) para 802.11b se encuentran en la tabla 5.

Tabla 5. Parámetros IEEE 802.11b

Definición	Significado	Valor
Lpaquete	Tamaño del paquete transmitido	1472[bytes]
Rdatos	Tasa de transmisión bruta	11[Mbps]
CWMin	Ventana de Contienda Mínima	32
MACheader	Agregado MAC de 28 bytes min	28 [bytes]
ACK	Datos en una trama MAC ACK	14 [bytes]
RTS	Datos en una trama MAC RTS	20 [bytes]
CTS	Datos en una trama MAC CTS	14 [bytes]
TPCLPS (opt)	PCLP para tasas de 2, 5.5 y 11 Mbps	96[ms]
TPCLPL para tasas > de 1 Mbps		192[ms]
TDIFS	Duración de un tiempo DIFS	50[ms]
TSIFS	Duración de un tiempo SIFS	10[ms]
S	Duración de intervalo de contienda	20[ms]
Tdelta	Tiempo Tx/Rx y procesamiento	1[ms]

El tamaño del paquete de datos escogido en la tabla anterior corresponde al máximo tamaño que puede tomar un paquete IP en una red ethernet. El valor de la ventana de contienda CWmin es de 32 como valor mínimo, ya que es el valor que viene por defecto en las componentes WiFi. Dado que el valor inicial del contador del contador de *backoff* se distribuye aleatoriamente en forma uniforme, se toma el valor promedio para este cálculo en las ecuaciones (2) y (3). La estructura de una trama MAC se muestra en la Figura 50. En la Figura 51 se muestra la composición del encabezado MAC. Además, como se desprende de la tabla 5, el estándar define que el PLCP (Physical Convergence Layer Procedure) puede ser de dos tipos, según sea la calidad del enlace. El PLCP largo (long) se usa normalmente como opción por defecto, a veces sin poder sustituirlo por el PLCP corto, por lo cual tiene más sentido evaluar el *throughput* con el PLCP largo. En la

figura 50 se muestra la estructura de este encabezado de la capa física, que normalmente se transmite a una tasa de 1 Mbps, como aparece en la tabla 5.

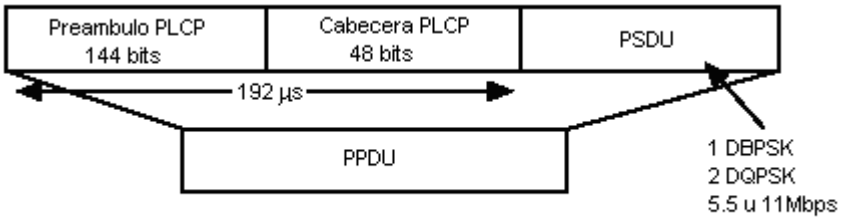

Figura 50. Formato Long PLCP PPDU

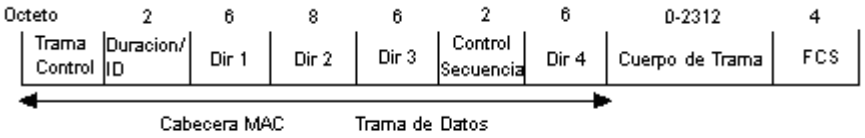

Figura 51. Formato Trama MAC

Normalmente los datos del encabezado se transmiten a la tasa nominal a la que opera el terminal de la red con condiciones de enlace más pobres. Si la tasa de transmisión bruta se fija a 54 Mbps, entonces se obtienen los valores de las ecuaciones (4) y (5) para la máxima tasa efectiva de servicio (MT) con PLCP, al reemplazar los valores de la tabla 1 en las ecuaciones (1) a (3):

MT Mbps con RTS = 16 Mbps (4)
MT Mbps sin RTS = 26 Mpbs (5)

Se observa en estos resultados que es mucho más conveniente trabajar en altas tasas de transmisión bruta, sin intercambio RTS/CTS. Sin embargo, en presencia del fenómeno de terminal oculto, será más conveniente usar el mecanismo RTS/CTS, como se desprende de las pruebas experimentales que se mostrarán más adelante en esta sección.

Fué necesaria la creación de un entorno donde operan varias redes inalámbricas WiFi, sin planificación de frecuencias, para analizar de qué manera se deteriora el *throughput* debido a la falta de planificación.

El máximo *throughput* por enlace se producirá cuando el medio esté siendo siempre ocupado sin producirse colisiones. El MT del canal va a ser el obtenido en (4) o (5), pero el MT por enlace será la mitad al obtenido anteriormente, ya que el canal es compartido en el tiempo. De esta manera la máxima tasa efectiva de servicio por enlace para el escenario desarrollado usando el PLCP largo será el siguiente:

MT Mbps con RTS = 9 Mbps
MT Mbps sin RTS = 14 Mbps

Este resultado se explica porqué el medio es compartido en el tiempo.

Después de realizar las pruebas adicionado factores externos, es visible y sobresaliente la diferencia en el uso de un equipo con firmware original y otro con el firmware trabajado por el grupo. En la Figura 52, se puede observar que cuando se adicionó interferencia externa sobre el

mismo canal, la tasa de transferencia en el firmware ucwrt estuvo por encima de los 14 Mbps y estable. Mientras en la Figura 53 se observa que la transferencia rodeó los 7 Mbps en el firmware original de Linksys con ruido externo. En la Figura 54 el se adicionó ruido externo a la transmisión utilizando como Punto de Acceso un computador con Madwifi, los resultados fueron más inestables en recepción.

Figura 52. Recepción Firmware *ucwrt* con Ruido

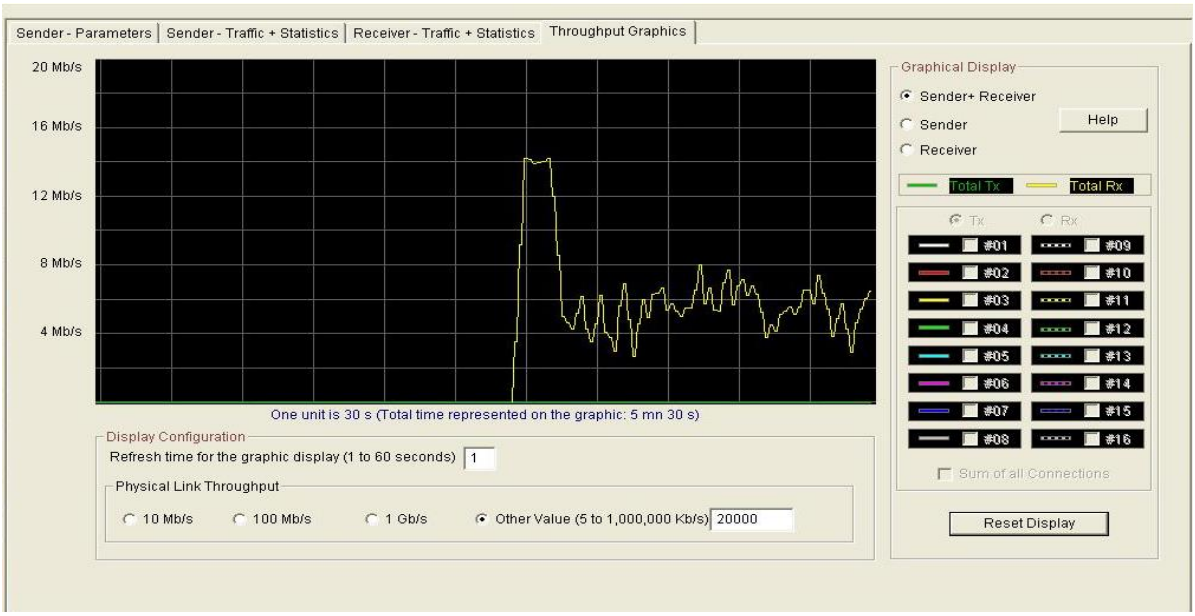

Figura 53. Recepción firmware Linksys con Ruido

Figura 54. Recepción AP (madwifi) con Ruido

Después de observar la gráficas, es concluyente la diferencia en estabilidad de transmisión de cada uno de los firmwares, el ucwrt puede mantener la transmisión más estable aun con ruido presente en el medio, mientras que el firmware Original de Linksys no presenta estabilidad en la transmisión.

La potencialidad del firmware basado en código abierto hace que el rendimiento sea superior en condiciones cambiantes, una de las principales razones es que se tiene la posibilidad de únicamente instalar los paquetes necesarios e ir adicionando poco a poco el software necesario.

En cuanto al procesamiento también es notoria la respuesta, el equipo realiza cambios de manera más veloz y por otro lado es completamente administrable ya que no solo se realiza el acceso mediante web, también por SNMP y SSH para manejo por consola.

Mostrando más a fondo el desarrollo de los aportes en este documento, con la creación de los paquetes que ayudarían con el mejor uso de las variables en dispositivos con firmware de código abierto, una vez activado el paquete de *ucfrag*, encargado de los cambios en fragmentación dinámica, la diferencia es notoria. En la Figura 55 el paquete empezó a funcionar y como se muestra, logró mantener la transmisión por encima de los 16 Mbps, cabe resaltar que las caídas mostradas en la figura, ocurrieron debido a los cambios de fragmentación y reinicio de interfaces, esto debido a que el equipo debe realizar este proceso para poder aplicar los cambios, sin embargo, la transmisión continua sin necesidad de volver a realizar las peticiones de autenticación y asociación, mientras que el firmware de Linksys no tiene la posibilidad realizar el cambio dinámico de fragmentación. Además, cuando se aplican cambios de manera manual en la fragmentación del firmware de Linksys, se observa que el equipo se reinicia, lo cual produce que los clientes se desasocien, aunque un momento más tarde, se asocian de nuevo.

Con esto se observa el cumplimiento de uno de los objetivos principales de este trabajo, implementar una característica adicional en el punto de acceso de manera funcional, que explote un poco más los recursos del hardware.

Es decir, el equipo realiza un muestreo mediante un ACK a un equipo vinculado a él, de acuerdo al resultado, el equipo es capaz de realizar un análisis de la información y realizar un cambio en la fragmentación de los paquetes que normalmente es 2346 y la varía de acuerdo a una tabla editable.

Tanto con la característica renombrada anteriormente como con el paquete *ucrts* se puede evidenciar el cumplimiento de uno de los objetivos donde se buscaba crear paquetes adicionales que ayuden al buen funcionamiento de una red inalámbrica en algún medio normal de trabajo.

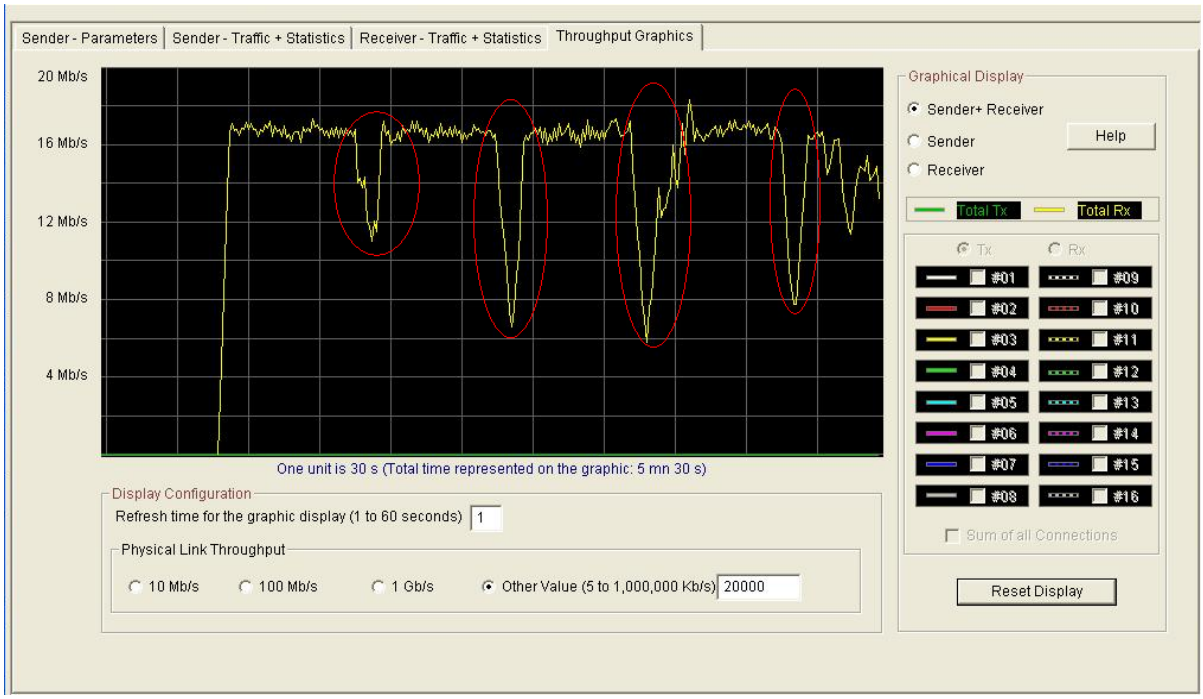

Figura 55. Cambios Automáticos en la Fragmentación

A continuación se muestra el código utilizado para realizar el paquete ucfrag donde al unirse con los scripts del cron del equipo, donde se encuentra instalado el ucwrt se obtiene un nuevo concepto y aporte sustancial al funcionamiento, para el grupo de trabajo el funcionamiento del *ucfrag*. Esto se llama funcionalmente y operativamente Fragmentación Dinámica.

PAQUETE UCFRAG

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void fx_execute(float);

int main(int argc, char *argv[])
{
```

```
FILE *f;
char linea[80], aux[64];
char *ip;
int counter;
float avg, min, max, frag;

if(argc!=2){
  printf("Escriba la direccion ip\n\r");
  return -1;
}
ip = argv[1];
printf("ip = %s\n\r", ip);

aux[0] = '\0';
strcat(aux, "ping -c 5 ");
strcat(aux, ip);
strcat(aux, " > /tmp/result.txt");
printf("%s\n\r", aux);

system(aux);
system("more /tmp/result.txt | grep -i avg > /tmp/result2.txt");

if ((f = fopen("/tmp/result2.txt", "rt")) == NULL) {
  printf("No se encuentra el archivo!\n\r");
  return -1;
}

fgets(linea, 80, f);
printf("%s\n\n\r", linea);
fclose(f);

char *tokenptr;

printf("%s\n\r","dividido: ");

counter = 0;
tokenptr=strtok(linea, "/");
while (tokenptr != NULL){
  printf("%s\n\r", tokenptr);
  tokenptr=strtok(NULL, "/");
  counter++;
  if(counter==4)
    avg = atof(tokenptr);
}

if ((f = fopen("tabla", "rt")) == NULL) {
  printf("No se encuentra el archivo de distribucion!\n\r");
  return -1;
}

while(!feof(f)){
  fgets(linea, 80, f);
  tokenptr=strtok(linea, "\t");
  min = atof(tokenptr);
  tokenptr=strtok(NULL, "\t");
  max = atof(tokenptr);
  tokenptr=strtok(NULL, "\t");
  frag = atof(tokenptr);
  if(min<=avg && avg<max){
    fx_execute(frag);
```

```
      fclose(f);
      return 0;
    }
  }

  fclose(f);
  return 0;
}

void fx_execute(float frag)
{
  char *fragstr,nvram1[20];
  int dec, sgn;

  // fragstr = ecvt(frag, 0, &dec, &sgn);
  sprintf(fragstr, "%0.00f", frag);
  printf("%0.00f  %s\n\r", frag, fragstr);

  nvram1[0] = '\0';
  strcat(nvram1, "nvram set wl0_frag=");
  strcat(nvram1, fragstr);
  printf("%s\n\r", nvram1);
  system(nvram1);
  system("nvram commit");
  system("/etc/init.d/S40network restart");
}
```

Archivo Tabla1

```
root@Tesis2:/www/cgi-bin/webif# cat tabla1
0       50      2346
50      100     2146
100     150     1946
150     200     1746
200     300     1546
300     400     1446
400     500     1346
500     600     1146
600     700     946
700     800     746
800     10000   546
root@Tesis2:/www/cgi-bin/webif#
```

El paquete de `ucfrag` es capaz de realizar los cambios dinámicos en fragmentación sin necesidad de reiniciar el equipo y así poder mantener un *throughput* más alto.

En la Figura 56 es evidente que la tasa de transmisión estuvo cerca de los 10 Mbps para el firmware original de Linksys mientras se realizaba la transmisión para ambos equipos. El envío y recepción de información es constante, pero si se quisiera realizar cambios en la fragmentación, fuese necesario reiniciar el equipo, labor que haría que la transmisión se interrumpiera. En este caso no tendría el firmware ucwrt punto de comparación, ya que en un equipo normal con firmware propietario, los cambios en fragmentación se realizan después de editarlos y al reiniciar el equipo, razón por la cual los clientes vinculados a él se desvincularían o perderían conexión.

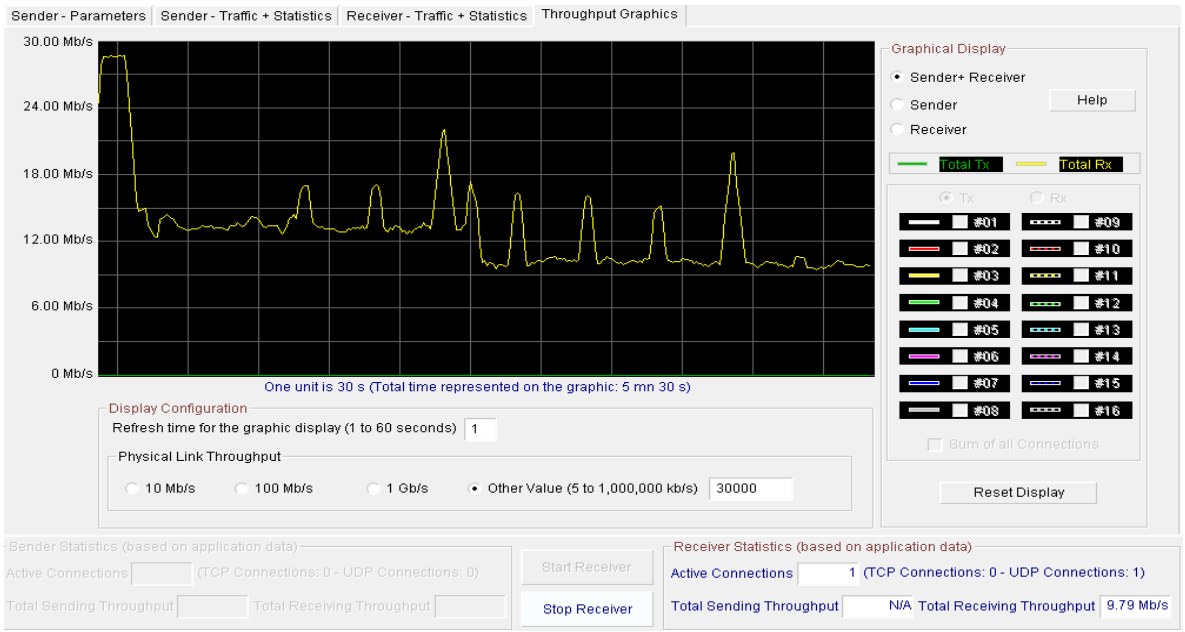

Figura 56. Transmisión de Información en Linksys con Firmware Original

Instalando otro punto de acceso intermedio, favoreciendo a un ambiente especial donde se simula un nodo oculto tal como se explicó en el capítulo 2 de este documento, se activo en los puntos de acceso utilizados para las pruebas esta característica. Logrando diferencia en los resultados, la Figura 57 muestra como la tasa de transferencia siendo menor a las conocidas anteriormente, sobrepasa en 2 megas a la mostrada en la Figura 58 con un firmware original del punto de acceso.

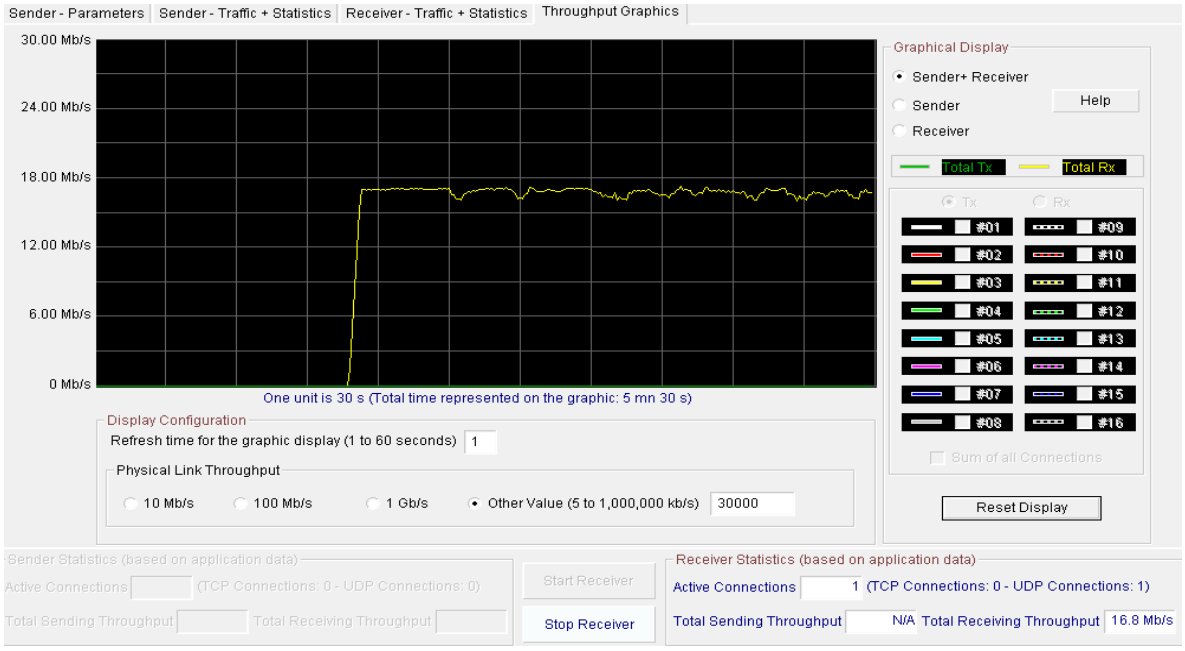

Figura 57. Recepción con vinculación a *ucwrt* con CTS activo

Para el funcionamiento de este escenario se tuvo en cuenta que el punto de acceso no estuviera dentro de la cobertura de otro punto de acceso pero que sus clientes si lo hicieran, de este modo se garantizó que los clientes pudieran estar en la cobertura de ambos puntos de acceso y tuvieran la capacidad de vincularse a uno u otro.

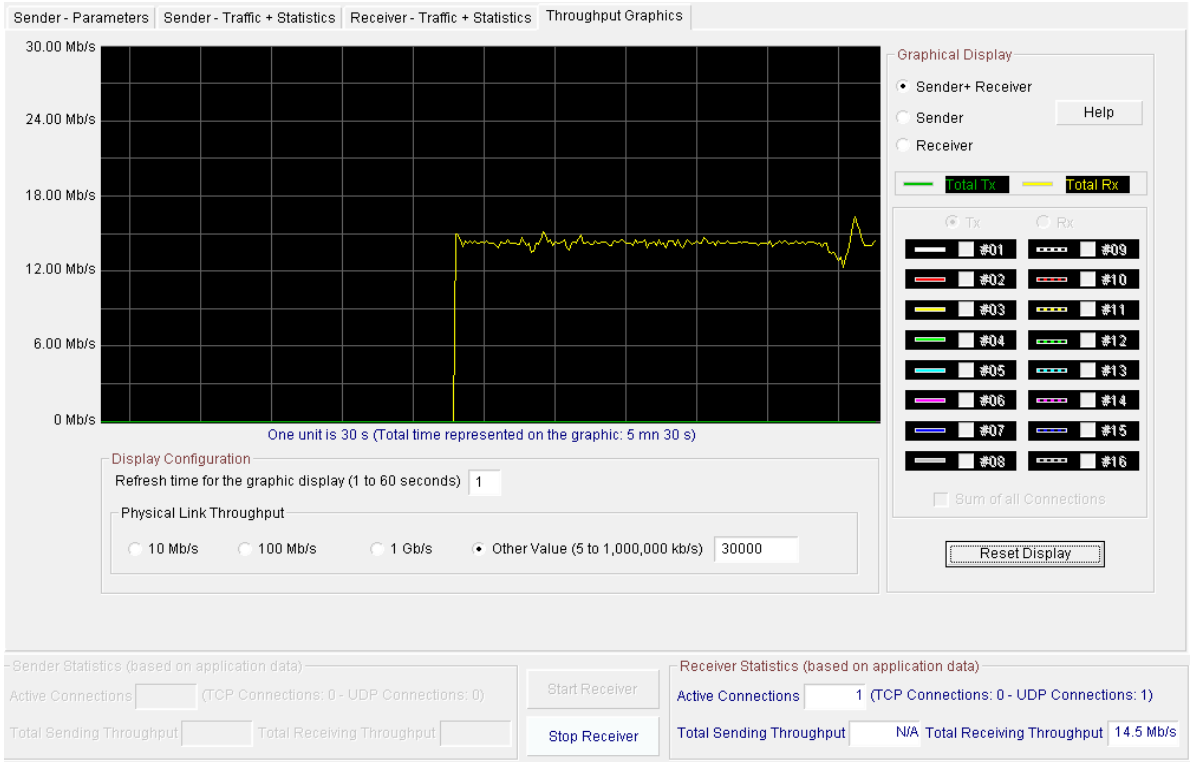

Figura 58. Recepción con vinculación a Firmware Linksys Original con CTS Activo

El paquete ucrts favorece al funcionamiento de punto de acceso debido a que evita la baja de rendimiento ocasionada por el nodo oculto.

Después de evaluar detenidamente los resultados, es evidente que la capacidad de realizar cambios dinámicos en los tamaños de fragmentación de paquetes y RTS/CTS, ayuda sustancialmente a la estabilidad y mejor desempeño de la red, además de mostrar más confiabilidad en ambientes cambiantes con interferencias. Las redes inalámbricas basadas en el estándar 802.11, cada día son más utilizadas y más dispositivos incorporan alguna interfaz de este tipo, los puntos de acceso son el núcleo de estas redes, en este orden de ideas, es fundamental que haya herramientas como las desarrolladas durante este trabajo capaces de potencializar el uso de las redes de este tipo además de hacer de manera automática una adaptación en los parámetros que mejoren el desempeño.

Teniendo en cuenta la variación de latencia en las medidas de rendimiento, se muestra en la Figura 59 esta medición en el firmware ucwrt con los paquetes de ucfrag funcionando, las variaciones fueron menores, mientras que en la Figura 60 los valores de Jitter fueron un poco superiores no son sensiblemente altos, en este caso se midieron sobre un Sistema Operativo Linux con Madwifi.

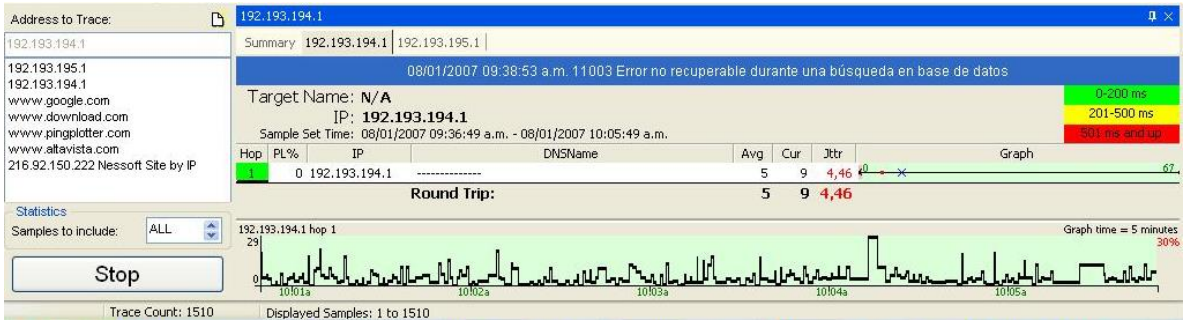

Figura 59. Jitter en Firmware ucwrt con ucfrag Activo

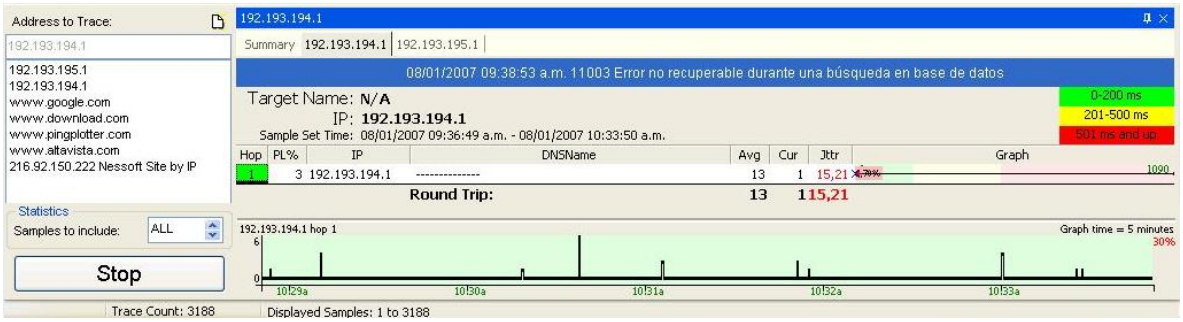

Figura 60. Jitter en Linux con Madwifi

En la Figura 61, las medidas de Jitter fueron superiores en comparación a las obtenidas con los sistemas operativos basados en código abierto. A pesar que con el ucwrt los valores fueron mejores, en el PC Linux con madwifi hubo algunos inconvenientes que posiblemente se ocasionaron debido a que el computador tenía carga de procesador por otras aplicaciones en el sistema operativo y esto ocasiona perdida en el rendimiento.

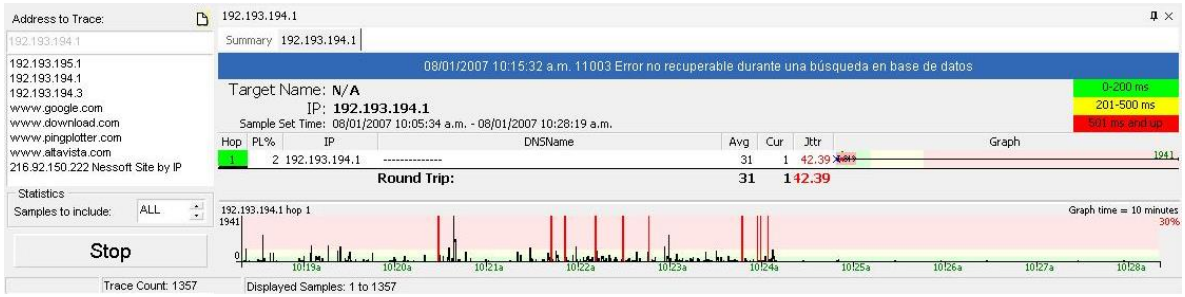

Figura 61. Jitter en Firmware Original de Linksys

Una vez realizadas las pruebas se llenaron los datos de las tablas, de acuerdo al tipo de prueba, en la Tabla 6 donde se midio el *throughput* máximo sin interferencia en firmware Linksys original, se muestra la variación de la velocidad. La Tabla 7 muestra *throughput* máximo sin interferencia en firmware ucwrt y la Tabla 8 donde se midió el *throughput* máximo sin interferencia en un computador con Madwifi, muestran de forma favorable que el resultado obtenido sin interferencias es superior en el firmware original del equipo, sin embargo estas condiciones son hoy en dia condiciones ideales de funcionamiento.

Ahora, es visible que en un ambiente normal, la interferencia en frecuencias de 2.4Ghz es bastante alta, es por esto que al revisar tablas como la Tabla 9 donde se midió el *throughput* máximo con interferencia en firmware Linksys original, la Tabla 10 que muestra el *throughput* máximo con interferencia en el firmware ucwrt y la Tabla 11 donde se especifica el *throughput* máximo con interferencia en un computador con Madwifi, la balanza se mueve hacia el firmware desarrollado por el grupo de trabajo, el ucwrt fue capaz de mantener un nivel de transmisión superior a los otros fimwares o computador, de esta forma, crece la importancia de desarrollos sobre este tipo de plataformas y pricipalmente utilizando software libre.

La Tabla 12 hace referencia a la medida de Jitter en diferentes tarjetas y firmwares, las variaciones pueden tener diferentes motivos, sin embargo, las conexiones de clientes realizadas sobre estos clientes donde se obtuvo mejores resultados en la transmisión y recepción mostraron que el firmware del grupo de trabajo tuvo un desempeño más adecuado y la medición del Jitter fue inferior.

Los tiempos de asociación de un cliente a una tarjeta de red, varian por diferentes causas, estas pueden ser interferencias, procesamiento de código en el computador o máquina donde está instalada la tarjeta entre otros. El grupo de trabajo realizó unas mediciones de los tiempos de asociación para poder comparar si había diferencia en esta información, la Tabla 13 revela la medida de tiempos de asociación en diferentes tarjetas y firmwares, estos tiempos son similares. Para estas pruebas no se tuvo en cuenta los factores externos que podían ocasionar que hubiese demoras en la asociación.

Tabla 6. *Throughput* Máximo Sin Interferencia en Firmware Linksys Original

Canal de vinculación	1
Canal de Emisor de Ruido	No Usado
Equipo Emisor de Ruido	NA
Potencia del Emisor	NA

***Throughput* en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
56000	Linksys WRT54G v3	Linksys Orig.	2346	26.1	No
56000	Linksys WRT54G v3	Linksys Orig.	2246	25.6	No
56000	Linksys WRT54G v3	Linksys Orig.	2146	27.8	No
56000	Linksys WRT54G v3	Linksys Orig.	2046	26.7	No
56000	Linksys WRT54G v3	Linksys Orig.	1946	26.9	No
56000	Linksys WRT54G v3	Linksys Orig.	1846	29.4	No
56000	Linksys WRT54G v3	Linksys Orig.	1746	29.5	No
56000	Linksys WRT54G v3	Linksys Orig.	1646	28.5	No
56000	Linksys WRT54G v3	Linksys Orig.	1546	38.1	No
56000	Linksys WRT54G v3	Linksys Orig.	1446	25.8	No
56000	Linksys WRT54G v3	Linksys Orig.	1346	28.5	No
56000	Linksys WRT54G v3	Linksys Orig.	1246	27.7	No
56000	Linksys WRT54G v3	Linksys Orig.	1146	25.7	No
56000	Linksys WRT54G v3	Linksys Orig.	1046	27.2	No

***Throughput* en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
35000	Linksys WRT54G v3	Linksys Orig.	2346 / Def	30.8	No
35000	Linksys WRT54G v3	Linksys Orig.	2246	30.2	No
35000	Linksys WRT54G v3	Linksys Orig.	2146	30.1	No
35000	Linksys WRT54G v3	Linksys Orig.	2046	30.7	No
35000	Linksys WRT54G v3	Linksys Orig.	1946	28.8	No
35000	Linksys WRT54G v3	Linksys Orig.	1846	29.9	No
35000	Linksys WRT54G v3	Linksys Orig.	1746	30.1	No
35000	Linksys WRT54G v3	Linksys Orig.	1646	30.3	No
35000	Linksys WRT54G v3	Linksys Orig.	1546	30.6	No
35000	Linksys WRT54G v3	Linksys Orig.	1446	30.4	No
35000	Linksys WRT54G v3	Linksys Orig.	1346	29.5	No
35000	Linksys WRT54G v3	Linksys Orig.	1246	29.9	No
35000	Linksys WRT54G v3	Linksys Orig.	1146	28.5	No
35000	Linksys WRT54G v3	Linksys Orig.	1046	30.1	No

Tabla 7. *Throughput* Máximo Sin Interferencia en Firmware ucwrt

Canal de vinculación	1
Canal de Emisor de Ruido	No Usado
Equipo Emisor de Ruido	NA
Potencia del Emisor	NA

***Throughput* en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
56000	Linksys WRT54G v3	Ucwrt	2346 / Def	24.1	No
56000	Linksys WRT54G v3	Ucwrt	2246	24	No
56000	Linksys WRT54G v3	Ucwrt	2146	22.6	No
56000	Linksys WRT54G v3	Ucwrt	2046	23.4	No
56000	Linksys WRT54G v3	Ucwrt	1946	23.6	No
56000	Linksys WRT54G v3	Ucwrt	1846	23.2	No
56000	Linksys WRT54G v3	Ucwrt	1746	23.6	No
56000	Linksys WRT54G v3	Ucwrt	1646	24.5	No
56000	Linksys WRT54G v3	Ucwrt	1546	24	No
56000	Linksys WRT54G v3	Ucwrt	1446	23.4	No
56000	Linksys WRT54G v3	Ucwrt	1346	26.4	No
56000	Linksys WRT54G v3	Ucwrt	1246	25.1	No
56000	Linksys WRT54G v3	Ucwrt	1146	23.4	No
56000	Linksys WRT54G v3	Ucwrt	1046	23	No

***Throughput* en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
35000	Linksys WRT54G v3	Ucwrt	2346 / Def	26.1	No
35000	Linksys WRT54G v3	Ucwrt	2246	26.1	No
35000	Linksys WRT54G v3	Ucwrt	2146	26	No
35000	Linksys WRT54G v3	Ucwrt	2046	25.8	No
35000	Linksys WRT54G v3	Ucwrt	1946	26.1	No
35000	Linksys WRT54G v3	Ucwrt	1846	25.7	No
35000	Linksys WRT54G v3	Ucwrt	1746	25.8	No
35000	Linksys WRT54G v3	Ucwrt	1646	25.7	No
35000	Linksys WRT54G v3	Ucwrt	1546	26.1	No
35000	Linksys WRT54G v3	Ucwrt	1446	25.1	No
35000	Linksys WRT54G v3	Ucwrt	1346	24.8	No
35000	Linksys WRT54G v3	Ucwrt	1246	24.6	No
35000	Linksys WRT54G v3	Ucwrt	1146	24.5	No
35000	Linksys WRT54G v3	Ucwrt	1046	24.3	No

Tabla 8. *Throughput* Máximo Sin Interferencia en Computador con Madwifi

Canal de vinculación	1
Canal de Emisor de Ruido	No Usado
Equipo Emisor de Ruido	NA
Potencia del Emisor	NA

Throughput **en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
56000	Linksys WRT54G v3	MadWifi	2346 / Def	27.5	No
56000	Linksys WRT54G v3	MadWifi	2246	27.4	No
56000	Linksys WRT54G v3	MadWifi	2146	27.4	No
56000	Linksys WRT54G v3	MadWifi	2046	27.2	No
56000	Linksys WRT54G v3	MadWifi	1946	26.8	No
56000	Linksys WRT54G v3	MadWifi	1846	26.2	No
56000	Linksys WRT54G v3	MadWifi	1746	26.1	No
56000	Linksys WRT54G v3	MadWifi	1646	25.5	No
56000	Linksys WRT54G v3	MadWifi	1546	25.3	No
56000	Linksys WRT54G v3	MadWifi	1446	25.2	No
56000	Linksys WRT54G v3	MadWifi	1346	24.9	No
56000	Linksys WRT54G v3	MadWifi	1246	24.7	No
56000	Linksys WRT54G v3	MadWifi	1146	24.5	No
56000	Linksys WRT54G v3	MadWifi	1046	24	No

Throughput **en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
35000	Linksys WRT54G v3	MadWifi	2346 / Def	32.1	No
35000	Linksys WRT54G v3	MadWifi	2246	32.3	No
35000	Linksys WRT54G v3	MadWifi	2146	32.1	No
35000	Linksys WRT54G v3	MadWifi	2046	32	No
35000	Linksys WRT54G v3	MadWifi	1946	31	No
35000	Linksys WRT54G v3	MadWifi	1846	31.4	No
35000	Linksys WRT54G v3	MadWifi	1746	29.2	No
35000	Linksys WRT54G v3	MadWifi	1646	30.4	No
35000	Linksys WRT54G v3	MadWifi	1546	31.5	No
35000	Linksys WRT54G v3	MadWifi	1446	30.9	No
35000	Linksys WRT54G v3	MadWifi	1346	30.6	No
35000	Linksys WRT54G v3	MadWifi	1246	30.2	No
35000	Linksys WRT54G v3	MadWifi	1146	32.1	No
35000	Linksys WRT54G v3	MadWifi	1046	31.8	No

Tabla 9. *Throughput* Máximo Con Interferencia en Firmware Linksys Original

Canal de vinculación	1
Canal de Emisor de Ruido	2
Equipo Emisor de Ruido	Linksys wrt54g
Potencia del Emisor	28mW

***Throughput* en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
56000	Linksys WRT54G v3	Orig. Linksys	2346 / Def	6.2	Si
56000	Linksys WRT54G v3	Orig. Linksys	2246	6.8	Si
56000	Linksys WRT54G v3	Orig. Linksys	2146	6.8	Si
56000	Linksys WRT54G v3	Orig. Linksys	2046	6.7	Si
56000	Linksys WRT54G v3	Orig. Linksys	1946	6.9	Si
56000	Linksys WRT54G v3	Orig. Linksys	1846	6.5	Si
56000	Linksys WRT54G v3	Orig. Linksys	1746	6.7	Si
56000	Linksys WRT54G v3	Orig. Linksys	1646	6.5	Si
56000	Linksys WRT54G v3	Orig. Linksys	1546	6.6	Si
56000	Linksys WRT54G v3	Orig. Linksys	1446	6.8	Si
56000	Linksys WRT54G v3	Orig. Linksys	1346	7.3	Si
56000	Linksys WRT54G v3	Orig. Linksys	1246	7.2	Si
56000	Linksys WRT54G v3	Orig. Linksys	1146	7	Si
56000	Linksys WRT54G v3	Orig. Linksys	1046	7.3	Si

***Throughput* en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
35000	Linksys WRT54G v3	Orig. Linksys	2346 / Def	7.1	Si
35000	Linksys WRT54G v3	Orig. Linksys	2246	7.3	Si
35000	Linksys WRT54G v3	Orig. Linksys	2146	7.3	Si
35000	Linksys WRT54G v3	Orig. Linksys	2046	7.1	Si
35000	Linksys WRT54G v3	Orig. Linksys	1946	6.7	Si
35000	Linksys WRT54G v3	Orig. Linksys	1846	7.3	Si
35000	Linksys WRT54G v3	Orig. Linksys	1746	7.8	Si
35000	Linksys WRT54G v3	Orig. Linksys	1646	8.2	Si
35000	Linksys WRT54G v3	Orig. Linksys	1546	7.6	Si
35000	Linksys WRT54G v3	Orig. Linksys	1446	7.9	Si
35000	Linksys WRT54G v3	Orig. Linksys	1346	7.4	Si
35000	Linksys WRT54G v3	Orig. Linksys	1246	7.2	Si
35000	Linksys WRT54G v3	Orig. Linksys	1146	7.8	Si
35000	Linksys WRT54G v3	Orig. Linksys	1046	7.9	Si

Tabla 10. *Throughput* Máximo Con Interferencia en Firmware ucwrt

Canal de vinculación	1
Canal de Emisor de Ruido	2
Equipo Emisor de Ruido	Linksys wrt54g
Potencia del Emisor	28 mw

***Throughput* en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
56000	Linksys WRT54G v3	UCwrt	2346 / Def	15.1	Si
56000	Linksys WRT54G v3	UCwrt	2246	14.6	Si
56000	Linksys WRT54G v3	UCwrt	2146	14.4	Si
56000	Linksys WRT54G v3	UCwrt	2046	14.1	Si
56000	Linksys WRT54G v3	UCwrt	1946	14.1	Si
56000	Linksys WRT54G v3	UCwrt	1846	14.2	Si
56000	Linksys WRT54G v3	UCwrt	1746	13.8	Si
56000	Linksys WRT54G v3	UCwrt	1646	14.1	Si
56000	Linksys WRT54G v3	UCwrt	1546	14.2	Si
56000	Linksys WRT54G v3	UCwrt	1446	13.9	Si
56000	Linksys WRT54G v3	UCwrt	1346	13.8	Si
56000	Linksys WRT54G v3	UCwrt	1246	13.7	Si
56000	Linksys WRT54G v3	UCwrt	1146	13.4	Si
56000	Linksys WRT54G v3	UCwrt	1046	13.2	Si

***Throughput* en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
35000	Linksys WRT54G v3	UCwrt	2346 / Def	16.1	Si
35000	Linksys WRT54G v3	UCwrt	2246	15.6	Si
35000	Linksys WRT54G v3	UCwrt	2146	15.4	Si
35000	Linksys WRT54G v3	UCwrt	2046	15.1	Si
35000	Linksys WRT54G v3	UCwrt	1946	15.1	Si
35000	Linksys WRT54G v3	UCwrt	1846	15.2	Si
35000	Linksys WRT54G v3	UCwrt	1746	14.8	Si
35000	Linksys WRT54G v3	UCwrt	1646	15.1	Si
35000	Linksys WRT54G v3	UCwrt	1546	15.2	Si
35000	Linksys WRT54G v3	UCwrt	1446	14.9	Si
35000	Linksys WRT54G v3	UCwrt	1346	14.8	Si
35000	Linksys WRT54G v3	UCwrt	1246	14.7	Si
35000	Linksys WRT54G v3	UCwrt	1146	14.4	Si
35000	Linksys WRT54G v3	UCwrt	1046	14.2	Si

Tabla 11. *Throughput* Máximo Con Interferencia en Computador con Madwifi

Canal de vinculación	1
Canal de Emisor de Ruido	2
Equipo Emisor de Ruido	Linksys wrt54g
Potencia del Emisor	28 mw

***Throughput* en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
56000	Linksys WRT54G v3	MadWifi	2346 / Def	10.5	Si
56000	Linksys WRT54G v3	MadWifi	2246	10.5	Si
56000	Linksys WRT54G v3	MadWifi	2146	10.6	Si
56000	Linksys WRT54G v3	MadWifi	2046	10.8	Si
56000	Linksys WRT54G v3	MadWifi	1946	11	Si
56000	Linksys WRT54G v3	MadWifi	1846	10.6	Si
56000	Linksys WRT54G v3	MadWifi	1746	10.9	Si
56000	Linksys WRT54G v3	MadWifi	1646	10.8	Si
56000	Linksys WRT54G v3	MadWifi	1546	10.9	Si
56000	Linksys WRT54G v3	MadWifi	1446	9.8	Si
56000	Linksys WRT54G v3	MadWifi	1346	9.9	Si
56000	Linksys WRT54G v3	MadWifi	1246	10.7	Si
56000	Linksys WRT54G v3	MadWifi	1146	10.3	Si
56000	Linksys WRT54G v3	MadWifi	1046	10.6	Si

***Throughput* en Transmisión sobre eth (Kbps)**	**Marca Punto de Acceso Principal**	**Firmware Punto de Acceso**	**Tamaño Fragmentación**	**Tasa de Recepción en Cliente (Mbps)**	**Transmisión Interferida**
35000	Linksys WRT54G v3	UCwrt	2346 / Def	12.6	Si
35000	Linksys WRT54G v3	UCwrt	2246	12.4	Si
35000	Linksys WRT54G v3	UCwrt	2146	12.3	Si
35000	Linksys WRT54G v3	UCwrt	2046	12.5	Si
35000	Linksys WRT54G v3	UCwrt	1946	12.5	Si
35000	Linksys WRT54G v3	UCwrt	1846	12.7	Si
35000	Linksys WRT54G v3	UCwrt	1746	12.8	Si
35000	Linksys WRT54G v3	UCwrt	1646	11.3	Si
35000	Linksys WRT54G v3	UCwrt	1546	11.8	Si
35000	Linksys WRT54G v3	UCwrt	1446	12.4	Si
35000	Linksys WRT54G v3	UCwrt	1346	12.6	Si
35000	Linksys WRT54G v3	UCwrt	1246	11.8	Si
35000	Linksys WRT54G v3	UCwrt	1146	11.9	Si
35000	Linksys WRT54G v3	UCwrt	1046	11.6	Si

Tabla 12. Medida de Jitter en Diferentes Tarjetas y Firmwares

Firmware	Tarjeta WPC54gS				Tarjeta DWL-G650				Tarjeta DWL-G520				Tarjeta Intel Pro 2000			
	Muestra 1	Muestra 2	Muestra 3	Muestra 4	Muestra 1	Muestra 2	Muestra 3	Muestra 4	Muestra 1	Muestra 2	Muestra 3	Muestra 4	Muestra 1	Muestra 2	Muestra 3	Muestra 4
Linksys Original	42	40	36	40	32	31	33	31	30	30	36	37	35	36	35	35
ucwrt	14	15	13	15	16	16	18	17	20	19	20	20	20	19	19	19
madwifi	28	25	26	24	29	29	29	28	27	28	27	30	31	32	27	27

tiempo en segundos

Tabla 13. Medida de Tiempos de Asociación en Diferentes Tarjetas y Firmwares

Firmware	Tarjeta WPC54gS				Tarjeta DWL-G650				Tarjeta DWL-G520				Tarjeta Intel Pro 2000			
	Muestra 1	Muestra 2	Muestra 3	Muestra 4	Muestra 1	Muestra 2	Muestra 3	Muestra 4	Muestra 1	Muestra 2	Muestra 3	Muestra 4	Muestra 1	Muestra 2	Muestra 3	Muestra 4
Linksys Original	3.6	3.7	3.6	3.4	4.1	3.2	3.3	4	5.1	5	4.6	4.9	3.4	3.4	3.8	4
ucwrt	2.8	2.9	3.3	3.1	3.5	3.6	3.1	3.9	4	4.1	4.1	4.8	4.1	2.9	2.9	3.5
madwifi	2.9	2.9	2.7	2.6	3	3.5	3.7	4	4.1	3.6	3.6	3.4	4.2	4.3	3.5	3.7

tiempo en segundos

4.4 INFRAESTRUCTURA DE LAS PRUEBAS

Para los experimentos citados anteriormente se utilizaron tres enrutadores inalámbricos Linksys WRT54G y un Linksys WRT54GS funcionando como puntos de acceso, ya que permitieron variar fácilmente muchos parámetros de configuración.

Un parámetro importante que ofrecen estos dispositivos es el control de potencia de transmisión, la que puede ser ajustada de 28 mW a 250 mW, permitiendo de esta forma controlar el radio de cobertura de un punto de acceso. Otro parámetro importante que se puede variar en estos dispositivos es el umbral de activación RTS/CTS. En las estaciones clientes se utilizaron tarjetas Intel Pro 2000, Linksys WPC65GS, DWL G520 y DWL G650. La particularidad de estas tarjetas es que se conectan a diferentes puertos, y algo muy importante es que pueden ser fácilmente recubiertas de un material que absorba la señal electromagnética, introduciendo una atenuación adicional a la de la propagación natural de 20 dB, aproximadamente. En un portátil Toshiba Intel Centrino de 1.7GHz, 512 MB de RAM se colocó una tarjeta Dlink DWL G650, la cual permite trabajar en modo de "escucha", del mismo modo que su tarjeta integrada Intel Pro Wireless.

Lo mismo se hizo en un Portátil HP ze4547. La idea de utilizar un sniffer (husmeador) en la interfaz de aire es la de poder verificar las configuraciones de hardware como el intercambio de tramas RTS/CTS y ocupación del canal inalámbrico que se utilizarán en las pruebas para minimizar interferencias. Los computadores genéricos utilizados como servidores (conectados en la puerta ethernet de los puntos de Acceso) son AMD 2600 XP, Pentium 4 HT y Pentium D, de 256MB, 1GB y 1GB de memoria RAM respectivamente. En los servidores y clientes se utilizó como S.O. Suse Linux v10.0 y Windows XP para evitar alterar la configuración original del S.O. del disco duro que tenían. *LanTraffic V2* funcionó perfectamente como herramienta para establecer transmisiones entre cliente y servidor, permitiendo la configuración de varios parámetros y características para paquetes TCP como UDP. Características de *LanTraffic V2* aprovechadas en pruebas son:

- El servidor maneja múltiples conexiones. Es posible especificar la cantidad de datos a ser transferidos, la tasa de generación que se desea que permanezca activa la transmisión.

- Cuando sea apropiado, las opciones pueden ser especificadas en k (kilo-) y M (mega-). De este modo, se puede especificar 128K en vez de 131072 bytes.

- *LanTraffic V2* reporta *throughput*, retardo (delay), jitter (variación del retardo) y pérdidas de datagramas en intervalos de tiempo especificados.

4.5 VALIDACIÓN DE LOS DISPOSITIVOS CLIENTES (TERMINALES)

Para asegurar que los portátiles no fueran la limitante en relación a la máxima tasa de transferencia que pueden enviar, se realizaron comparaciones entre tarjetas inalámbricas de diferentes tipos y fabricantes. En la prueba se utilizaron las siguientes tarjetas inalámbricas Liksys WPC54GS, adaptador PCMCIA, Dlink DWL-650, adaptador PCMCIA y Dlink DWL-G520 adaptador PCI.

Se utilizó el acceso básico (protocolo WiFi sin intercambio de mensajes RTS/CTS) para realizar estas mediciones. La prueba se realizó durante cinco minutos y consistía en enviar datos desde una sola estación al servidor utilizando *LanTraffic V2*. El *throughput* de saturación es la máxima tasa de transferencia sin pérdida de paquetes. Debido a que el fin de las pruebas no es determinar

el *throughput* de cada uno de los dispositivos de red cliente por separado, se tomaron los valores promedios de las pruebas. Es notoria la diferencia en transmisión sobre los diferentes tipos de firmware instalado bajo los enrutadores utilizados.

CONCLUSIONES Y RECOMENDACIONES

Una vez desarrollado el piloto, fue evidente que se hace necesario tener un proceso organizado para el desarrollo, los cambios continuos en las fuentes y constantes mejoras obligan a obtener resultados de manera más veloz. El estándar de IEEE 802.11 no ha cambiado notoriamente, pero fue importante tener un conocimiento más específico sobre aquellas funciones básicas y principales.

Después de analizar el estándar y documentar más a fondo la base del conocimiento, se evidencia el progreso local, tanto en la ciudad como en el país, donde se pueden obtener cada vez mejores resultados en aplicaciones, diseño y gestión de redes inalámbricas.

Fue enriquecedor e importante poder introducir funcionalidades adicionales a un producto inicialmente creado fuera del entorno universitario local, todos aquellos aportes que hacen que el desarrollo pueda ser más específico y personalizado favorecen el desempeño del producto firmware y paquetes para este caso.

El rendimiento del producto firmware en cuanto a su funcionamiento fue favorable, después de las comparaciones en ambientes cambiantes, se notó que la estabilidad era superior y los aportes de los paquetes ayudan a este fin.

Toda cualidad adicional que pueda ofrecerse en este tipo de desarrollos tiene un valor agregado importante, ya que ofrece una característica diferenciadora en el mercado, en este caso, la posibilidad de mejorar la estabilidad de un Punto de Acceso y ofrecer mejores prestaciones a la hora de entregarse como un producto funcional.

Existen varios aportes en software (paquetes) que se pueden realizar, del mismo modo que desarrollar firmwares para otro tipo de equipos o de otras marcas, se hace necesario tener las fuentes de código con los *drivers* de hardware para iniciar el proceso. Normalmente una empresa productora de equipos no ofrece el código de programación, sin embargo, trabajos como el realizado por el grupo muestran que es posible realizar avances en búsqueda de mejoras tecnológicas.

Una vez realizadas todas las prácticas se obtuvieron los resultados sobre el funcionamiento superior del prototipo firmware desarrollado por el grupo de trabajo en comparación con los existentes probados y del computador con sistema operativo Linux con madwifi, configurado como Punto de Acceso. Se evidenció con claridad que una vez un equipo tenga menor carga en procesamiento de funciones y características, el rendimiento y estabilidad serán superiores.

Al examinar detenidamente el funcionamiento y rendimiento de una red, y después de observar que con los aportes realizados por el grupo de trabajo mejoran el rendimiento, es significativa la posibilidad de ofrecer diferentes tipos de configuraciones y perfiles de acuerdo a las necesidades y requerimientos de los usuarios.

Técnicamente, el factor seguridad debe ser una preocupación a la hora de crear una red inalámbrica, de este modo, la capacidad que ofrecen tanto el firmware como los paquetes de mejorar el rendimiento, hacen que la implementación de estándares y métodos de seguridad inalámbrica como WPA, WEP entre otros, puedan funcionar sin ningún problema sin sacrificar rendimiento en las redes inalámbricas actuales.

Las velocidades de conexión, transmisión y recepción de datos son importantes debido al intercambio y manejo de información; en el trabajo se mostró que se podían obtener velocidades prácticas reales bastantes altas y que favorecen a los usuarios por que presentan una relación costo beneficio buena.

Aunque el manejo de entornos Linux no es estándar en todas las distribuciones específicamente el control por X-Window, cada vez es más evidente que este tipo de sistemas operativos son más confiables y dan cualidades de funcionamiento mejores, es por esta razón que la incursión en tecnologías de software libre donde la adquisición de información y fuentes es mas fácil toma gran fuerza a la hora de desarrollar cualquier tipo de implementación con el fin de favorecer a un mayor número de personas y con menor precio de los productos.

Es posible ampliar las capacidades del punto de acceso incluyendo una memoria tipo SD, que en la actualidad es de 1GB, en la cual se pueden instalar muchos más paquetes adicionales.

Las librerías y fuentes están desarrollándose constantemente mediante la contribución de las diferentes personas involucradas en el *open source*, esto hace que las actualizaciones frente a los fallos sean rápidas, siempre y cuando el tema sea de interés. En un inicio existían pocos desarrolladores y aplicaciones, hoy en día ya son alrededor de cien aplicaciones y se puede observar que en este caso el tema es bastante interesante y está creciendo a un ritmo acelerado.

Es posible implementar aplicaciones para dispositivos embebidos que pueden situarse desde capa 2 del modelo OSI, hacia arriba, es decir que se tiene control desde la capa de acceso al medio hasta la capa de aplicación.

Una vez terminadas las pruebas se pudo observar que en el escenario analizado se muestran tres problemas básicos en redes inalámbricas modelados: el problema del terminal oculto, el problema del terminal expuesto y el fenómeno de captura. Éste se produce por efectos del mecanismo de *backoff* exponencial con que opera el protocolo en conjunto con el hecho de que los nodos de la red sólo tienen acceso parcial a la actividad que se registra en sus enlaces. Cuando no se utiliza intercambio RTS/CTS entre puntos de acceso y estaciones inalámbricos, en situaciones de alta carga en la red, siempre una estación inalámbrica captura el canal, lo cual conlleva a una mayor tasa de transmisión de esta estación, en deterioro de las otras que estén haciendo alguna transmisión, inclusive hacia otro punto de acceso cercano. La solución a este problema es utilizar el intercambio RTS/CTS, ya que en situaciones de alta carga no permite que una estación se apodere del canal, equipando así la tasa de las transmisiones.

Es óptimo utilizar los mismos canales RF en celdas cercanas, ya que, como se vio en los tres escenarios, se generan problemas de interferencias entre los canales, produciéndose así una baja en el *throughput* de los enlaces. Con respecto a la utilización de canales traslapados, se puede concluir que es recomendable siempre usar sólo canales no traslapados, ya que si se utilizan canales cercanos, como en las mediciones hechas, se genera una baja en el desempeño de la red. Por lo anterior es recomendable siempre hacer una planificación detallada de redes inalámbricas en los lugares en donde se desee implementar esta tecnología, para obtener el máximo desempeño posible.

REFERENCIAS BIBLIOGRAFICAS

[1] HUAWEI LAUNCHES INDUSTRY-LEADING 40GBPS PER WAVELENGTH DWDM TRANSMISSION SYSTEM. [en línea] Página Web versión HTML.4.0 Longgang District Shenzhen - China, 2005. [citado 8 de Enero de 2007]. Disponible en Internet http://www.huawei.com/news/view.do?id=466&cid=-1001

[2] IEEE-SA STANDARDS BOARD STANDARDS REVIEW COMMITTEE. [en línea] Página Web versión HTML.4.0 Piscataway, New Jersey, 2005. [citado 8 de enero de 2007]. Disponible en Internet: < http://standards.ieee.org/board/rev/905recomm.html />.

[3] COLOMBIA. MINISTERIO DE COMUNICACIONES. Resolución número 000689 de 2004. Bogotá, 2004; p.3.

[4] ALARCÓN, JOSÉ M [en línea] Página Web versión HTML.4.0. NewTec Ediciones 2003 España [citado 8 de Enero de 2007]. Disponible en Internet: <http://www.windowstimag.com/atrasados/2003/81_nov03/articulos/seguridad_5.asp>

[5] SEGURIDAD INALÁMBRICA AVANZADA. [en línea] Página Web versión HTML.4.0 España, 2005. [citado 8 de enero de 2007]. Disponible en Internet: http://www.windowstimag.com/atrasados/2003/81_nov03/articulos/seguridad_3.asp

[6] INTRODUCCION A LAS COMUNICACIONES MOVILES [en línea] Página Web versión HTML.4.0 Albacete - España, 2005. [citado 11 de Abril de 2006]. Disponible en Internet http://www.info-ab.uclm.es/asignaturas/42638/pdf/cap1.pdf />.

[7] COLOMBIA. MINISTERIO DE COMUNICACIONES. Resolución número 2064 de 2005. Bogotá, 2004; p.2. disponible en Internet: <http://www.mincomunicaciones.gov.co/mincom/src/user_docs/Archivos/normatividad/2005/Resolucion/R2064de2005.pdf>

[8] COLOMBIA. MINISTERIO DE COMUNICACIONES. LineamientosPoliticaBandaAnchaII de 2005. Bogotá, 2004; p.5. disponible en Internet: <http://www.crt.gov.co/Documentos/ActividadRegulatoria/MasificacionBandaAncha/LineamientosPoliticaBandaAnchaII.pdf>

[9] THE FREE SOFTWARE DEFINITION. . [en línea] Página Web versión HTML.4.0 Boston, MA - USA, 2007. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.gnu.org/philosophy/free-sw.html>

[10] DEFINICIÓN. [en línea] Página Web versión HTML.4.0 Medellin - Colombia, 2005. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.altred.net/pagina/node/27>

[11] OBJETIVOS Y MISIÓN DE REDLIBRE. . [en línea] Página Web versión HTML.4.0 España, 2003. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.redlibre.net/objetivos.php>

[12] INTRODUCCION A LAS COMUNICACIONES MOVILES [en línea] Página Web versión HTML.4.0 Albacete - España, 2005. p.16 [citado 11 de Abril de 2006]. Disponible en Internet: http://www.info-ab.uclm.es/asignaturas/42638/pdf/cap1.pdf />.

[13] AS.MAX. [en línea] Página Web versión HTML.4.0 Boca Raton, FL - USA, 2003. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.airspan.com/products_group.aspx?ProductGroupID=1>

[14] INTRODUCING 4MOTION MOBILE WIMAX. [en línea] Página Web versión HTML.4.0 Tel Aviv, Israel, 2005 -2007. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.alvarion.com/mobilewimax/>

[15] INTRODUCCIÓN A LASTECNOLOGÍASINALÁMBRICAS. [en línea] Página Web versión HTML.1.0 Merida - Venezuela, 2006. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.wilac.net/descargas/documentos/8va_eslared/01_Intro_tec_inalambricas.pdf>

[16] DICCIONARIO BÁSICO DE DERECHO INFORMÁTICO E INFORMÁTICA JURÍDICA. [en l ínea] Página Web versión HTML.4.0 San Jose – Costa Rica, 2004. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.hess-cr.com/secciones/dere-info/diccionario/f.shtml>

[17] MEDIA MVP PRODUCTS. . [en línea] Página Web versión HTML.4.0 New York, 2006. [citado 9 de enero de 2007]. Disponible en Internet <http://www.hauppauge.com/pages/prods_mvp.html>

[18] HANDHELDS.ORG. [en línea] Página Web versión HTML.4.0 MIT, USA, 2007. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.handhelds.org/geeklog/index.php>

[19] MYTHTV. [en línea] Página Web versión HTML.4.0 OSL, Oregon - USA, 2006. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.mythtv.org>

[20] THE BRAIN INSIDE THE BOX. [en línea] Página Web versión HTML.4.0 Alviso, CA, 2006. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.tivo.com>

[21] A MIPS32 SIMULATOR. [en línea] Página Web versión HTML.4.0 Wisconsin - USA, 2006. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.cs.wisc.edu/~larus/spim.html>

[22] TASK GROUP N SYNC. [en línea] Página Web versión HTML.4.0 USA, 2006. [citado 9 de enero de 2007]. Disponible en Internet: http://www.xs4all.nl/~jrme/tgnsync.html

[23] HOW MIMO WORKS. [en línea] Página Web versión HTML.4.0 Palo Alto, CA, 2005 - 2006. [citado 9 de enero de 2007]. Disponible en Internet: <http://www.airgonetworks.com/mimo/how/>

[24] TERCER SEMINARIO MATARO. [en línea] Página Web versión HTML.4.0 , 2005 - . [citado 9 de enero de 2007]. Disponible en Internet: <pof.eslack.org/writings/80211n-mataro.pdf>

[25] MIMO LA PRÓXIMA GENERACIÓN DE LA TECNOLOGÍA WI-FI. [en línea] Página Web versión HTML.4.0 México, 2006. [citado 9 de enero de 2007]. Disponible en: <Internet:http://www.eveliux.com/index.php?option=content&task=view&id=81&Itemid=>

[26] NAPIER, DAVID BANDEL Y ROBERT. Linux Edición Especial. Madrid : Prentice Hall, 2001

[27] GENERAL INFORMATION. [en línea] Página Web versión HTML.4.0 USA, 2005 - 2006. [citado 9 de enero de 2007]. Disponible en Internet: <http://madwifi.org/wiki/MadWifi>

[28] HARDWARE SUPPORTED BY MADWIFI. [en línea] Página Web versión HTML.4.0 USA, 2005 - 2006. [citado 9 de enero de 2007]. Disponible en Internet: http://madwifi.org/wiki/Compatibility

[29] GPL CODE . [en línea] Página Web versión HTML.4.0 Irvine, CA, 2005 - 2007. [citado 9 de enero de 2007]. Disponible en Internet: http://www.linksys.com/servlet/Satellite?c=L_Download_C2&childpagename=US%2FLayout&cid=1115417110138&packedargs=sku%3D1133202177241&pagename=Linksys%2FCommon%2FVisitorWrapper

[30] OPENWRT . [en línea] Página Web versión HTML.4.0 , 2005. [citado 8 de enero de 2007]. Disponible en Internet: http://openwrt.org/

[31] GCC, THE GNU COMPILER COLLECTION . [en línea] Página Web versión HTML.4.0 Boston, MA 2006. [citado 8 de enero de 2007]. Disponible en Internet: http://gcc.gnu.org/

[32] BUSYBOX: THE SWISS ARMY KNIFE OF EMBEDDED LINUX. [en línea] Página Web versión HTML.4.0 USA, 1999 - 2006. [citado 9 de enero de 2007]. Disponible en Internet: http://www.busybox.net/

[33] OMNICOR, Aplicación. [en línea] Página Web versión HTML.4.0 USA - 2006. [citado 9 de enero de 2007].Descarga disponible en Internet: <http://www.omnicor.com/netest.htm>