EVALUACIÓN DEL DESEMPEÑO DE LOS MECANISMOS DE RECONOCIMIENTO DEL PROTOCOLO 802.11 MODIFICADOS PARA ENLACES PUNTO A PUNTO DE LARGA DISTANCIA

Alma Patricia Collazos Sarria
Diego Andrés Latorre Gómez

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Grupo I+D de Nuevas Tecnologías en Telecomunicaciones
Popayán, Marzo de 2010

EVALUACIÓN DEL DESEMPEÑO DE LOS MECANISMOS DE RECONOCIMIENTO DEL PROTOCOLO 802.11 MODIFICADOS PARA ENLACES PUNTO A PUNTO DE LARGA DISTANCIA

Alma Patricia Collazos Sarria
Diego Andrés Latorre Gómez

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

Director: Guefry Agredo Méndez – M. Sc.

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Grupo I+D de Nuevas Tecnologías en Telecomunicaciones
Popayán, Marzo de 2010

A quienes con su amor y apoyo me ayudaron a salir adelante, las personas que más amo en el mundo, mis padres.

A mi hermano 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.

Alma Patricia Collazos Sarria

Con un profundo amor a mis padres, quienes con su guía y gran amor, impulsaron mi carrera. A mis hermanas, por su gran apoyo, por compartirme sus alegrías y su vida.

.

Diego Andrés Latorre Gómez

AGRADECIMIENTOS

Los autores expresan su agradecimiento:

A la Universidad del Cauca y a la FIET por su formación académica y crecimiento en valores.

Al Ingeniero Guefry Agredo Méndez gratitud por sus valiosas orientaciones, dedicación y amistad.

A los ingenieros Francesco Gringoli creador de OpenFWWF y Sandra Salmerón de EHAS España por su apoyo incondicional en la búsqueda del conocimiento.

Por último, el más afectuoso agradecimiento a nuestras familias, quienes nos apoyaron incondicionalmente, no solo en este trabajo de grado, si no en nuestra vida.

El haber llegado hasta este punto en nuestras carreras académicas y personales ha sido gracias a Uds.

Los Autores
Popayán, Marzo de 2010

CONTENIDO

INTRODUCCIÓN **1**

1 PRINCIPIOS DEL PROTOCOLO IEEE 802.11, MÉTODOS DE RECONOCIMIENTO CONTEMPLADOS EN EL ESTÁNDAR Y LIMITACIONES EN LARGAS DISTANCIAS **3**

1.1 DESCRIPCIÓN GENERAL DEL PROTOCOLO IEEE 802.11 **3**
1.1.1 Generalidades 3
1.1.2 Componentes de la Arquitectura del Estándar IEEE 802.11 4
1.1.3 Nivel Físico del Protocolo IEEE 802.11 16

1.2 LIMITACIONES DEL PROTOCOLO 802.11 EN LARGAS DISTANCIAS **17**
1.2.1 Legislación 17
1.2.2 Pérdidas de Propagación 19
1.2.3 Sensibilidad de Receptores y Margen de señal 19
1.2.4 Constantes MAC 802.11 20
1.2.5 Colisiones en Enlaces de Larga Distancia 21
1.2.6 Interferencia entre Enlaces Adyacentes 21

1.3 MÉTODOS DE RECONOCIMIENTO **23**
1.3.1 ACK Estándar 23
1.3.2 ACK en Bloque 26
1.3.3 No ACK 30

1.4 CONSIDERACIONES **31**
1.4.1 ACK 31
1.4.2 Block ACK 31
1.4.3 No ACK 32

2 PLATAFORMA DE DESARROLLO A NIVEL MAC **33**

2.1 HERRAMIENTAS PARA EL DESARROLLO DE NUEVOS MECANISMOS MAC **33**
2.1.1 Sistema Operativo Linux 33
2.1.2 Enrutador WRT54GL 34
2.1.3 OpenWrt 34

2.2 ALTERNATIVAS DE DESARROLLO **35**
2.2.1 MadWifi 35
2.2.2 NS-2 37
2.2.3 OpenFWWF 37

2.3 COMPILACIÓN CRUZADA PARA OPENWRT EN LINUX **37**
2.3.1 Obtención del *Buildroot* de OpenWrt 39
2.3.2 Manejo del *Buildroot* de OpenWrt 39
2.3.3 Personalización del Sistema de Archivos Objetivo para Requisitos Particulares 40
2.3.4 Modificar la Configuración de *Busybox* para Requisitos Particulares 41
2.3.5 Modificar la Configuración de uC*libc* para Requisitos Particulares 41
2.3.6 Funcionamiento del *Buildroot* 41
2.3.7 Utilizar el *Toolchain* del *Uclibc* 42
2.3.8 Utilizar el *uClibc Toolchain* fuera del *Buildroot* 42

2.4 CONFIGURACIÓN y COMPILACIÓN DEL FIRMWARE openwrt con OPENFWWF **43**
2.4.1 Interfaz Web 43

2.4.2 Menú de Configuración 43
2.4.3 Instalación de OpenWrt 48
2.4.4 Adecuación de OpenFWWF 49
2.4.5 Conjunto de Instrucciones de OpenFWWF 50

2.5 PROCESOS DE TRANSMISIÓN Y RECEPCIÓN 55
2.5.1 Transmisión 55
2.5.2 Proceso de Recepción 57

2.6 PLATAFORMA DE DESARROLLO MAC 58
2.6.1 Puesta en Funcionamiento y Verificación 58
2.6.2 Criterios de Diseño del Mecanismo Block ACK. 59
2.6.3 Diagramas en Bloque del Mecanismo Block ACK 59
2.6.4 Limitaciones. 61

3 PARÁMETROS DE DESEMPEÑO Y HABILITACIÓN DE LOS MECANISMOS DE RECONOCIMIENTO 62

3.1 PARÁMETROS DE DESEMPEÑO 62
3.1.1 Throughput 62
3.1.2 Jitter 62
3.1.3 Porcentaje de Paquetes Perdidos 62

3.2 HERRAMIENTAS PARA EVALUAR EL DESEMPEÑO 64
3.2.1 Wireshark 64
3.2.2 Airmon-ng 65
3.2.3 Iperf 66

3.3 SOFTWARE PARA PRUEBAS 66
3.3.1 Elastix 67
3.3.2 X-Lite 67
3.3.3 Zoiper 68
3.3.4 VLC 69

3.4 CONFIGURACIÓN MECANISMOS DE RECONOCIMIENTO 70
3.4.1 Ubiquiti PowerStation 2 70
3.4.2 Activación de No ACK 71
3.4.3 Variación de ACKTimeout 72

4 RESULTADOS, ANÁLISIS Y CONCLUSIONES DE LA EVALUACIÓN DE LOS MECANISMOS DE RECONOCIMIENTO COMPRENDIDOS EN EL TRABAJO DE GRADO 74

4.1 PLAN DE PRUEBAS 74

4.2 CONFIGURACIÓN DE EQUIPOS Y ESCENARIOS DE PRUEBA 74
4.2.1 Configuración de Equipos 74
4.2.2 Escenarios de Prueba 77

4.3 CONFIGURACIÓN DE HERRAMIENTAS 80
4.3.1 Iperf 80
4.3.2 VLC 81
4.3.3 Wireshark 81
4.3.4 Elastix 82
4.3.5 X-Lite y Zoiper 83

4.4 RESULTADOS Y ANÁLISIS DE LA EVALUACIÓN PRÁCTICA DE LOS MECANISMOS DE RECONOCIMIENTO **83**
4.4.1 Enlace FIET – Tablazo 83
4.4.2 Enlace FIET – Cajete 90

CONCLUSIONES **98**

REFERENCIAS BIBLIOGRÁFICAS **99**

LISTA DE FIGURAS

Figura 1.1 Conjunto de Servicios Básicos. 4
Figura 1.2. Conjunto de Servicios Extendidos. 5
Figura 1.3. Conjunto de Servicios Básicos Independientes. 5
Figura 1.4. Sistema de Distribución Inalámbrico. 6
Figura 1.5. Diagrama de Flujo del Proceso de Función de Coordinación Distribuida. 10
Figura 1.6. Trama MAC General. 12
Figura 1.7. Campo Control de Trama. 13
Figura 1.8. Uso de tramas RTS y CTS. 15
Figura 1.9. Formato de la Trama ACK. 15
Figura 1.10. Diagrama de Contienda. 20
Figura 1.11. Diagrama de Superposición de Canales en IEEE 802.11 b/g. 22
Figura 1.12. Máscara de Espectro Transmitida en IEEE 802.11b. 23
Figura 1.13. Superposición de Armónicos en los Canales 1, 6 y 11. 23
Figura 1.14. Secuencia de Mensajes para el Mecanismo Block ACK. 27
Figura 1.15. Secuencia Típica del Mecanismo Block ACK Inmediato. 29
Figura 1.16. Secuencia Típica de Block ACK al Usar la Política Retardada. 29
Figura 2.1. Menú de Configuración Principal. 44
Figura 2.2. Menú de Configuración de Red. 45
Figura 2.3. Menú de Controladores Inalámbricos. 45
Figura 2.4. Menú de Selección de la Versión del *firmware* b43. 46
Figura 2.5. Menú de Componentes de LuCI. 46
Figura 2.6. Menú de Temas de LuCI. 47
Figura 2.7. Menú de Configuración del *kernel*. 47
Figura 2.8. Selección de Optimizar Espacio. 48
Figura 2.9. Acceso a OpenWrt a través de telnet. 49
Figura 2.10. Compilación del *firmware* OpenFWWF. 50
Figura 2.11. Proceso de Transmisión de OpenFWWF. 56
Figura 2.12. Proceso de Recepción de OpenFWWF. 57
Figura 2.13. Diagrama de Bloques Proceso de Transmisión con Block ACK. 60
Figura 2.14. Diagrama de Bloques Proceso de Recepción con Block ACK. 61
Figura 3.1. Captura de Wireshark. 65
Figura 3.2. Resultado de Ejecutar *airmon-ng*. 65
Figura 3.3. Esquema de Funcionamiento de *iperf*. 66
Figura 3.4. Página Inicial de Administración de Elastix. 67
Figura 3.5. Presentación general de X-lite. 68
Figura 3.6. Zoiper 2.0 para Linux. 69
Figura 3.7. Esquema General de Funcionamiento de VLC. 69
Figura 3.8. Ubiquiti PowerStation 2. 71
Figura 3.9. Presencia de IEEE 802.11 ACK. 71
Figura 3.10. 802.11 con No ACK. 72
Figura 4.1. Configuración de Red de los Enlaces. 74
Figura 4.2. Configuración Interfaz Ethernet del AP. 75
Figura 4.3. Configuración Inalámbrica del AP. 75
Figura 4.4. Configuración Interfaz Ethernet del Cliente. 76
Figura 4.5. Configuración Inalámbrica del Cliente. 76
Figura 4.6. Comprobación Existencia de la Red. 77
Figura 4.7. Perfil del Enlace FIET – Tablazo. 78
Figura 4.8. Perfil del Enlace FIET – Cajete. 79
Figura 4.9. Torre FIET. 80
Figura 4.10. Iperf TCP. 80
Figura 4.11. Esquema general de las pruebas con VLC. 81
Figura 4.12. Reproducción video *streaming*. 81
Figura 4.13. Interfaces de Wireshark. 82
Figura 4.14. Menú de Creación de Extensiones en Elastix. 82
Figura 4.15. Esquema General de Pruebas con Elastix. 83

Figura 4.16. *Throughput* Vs ACKTimeout. 84
Figura 4.17. *Throughput* Vs ACKTimeout. 85
Figura 4.18. *Jitter* Vs ACKTimeout. 86
Figura 4.19. Datagramas Perdidos (de un total de 893) Vs ACKTimeout. 86
Figura 4.20. Porcentaje de Paquetes Perdidos Vs ACKTimeout. 88
Figura 4.21. *Jitter* Máximo Vs ACKTimeout. 88
Figura 4.22. *Jitter* Promedio Vs ACKTimeout. 89
Figura 4.23. Throughput Vs ACKTimeout. 92
Figura 4.24. *Throughput* Vs ACKTimeout. 93
Figura 4.25. *Jitter* Vs ACKTimeout. 93
Figura 4.26. Porcentaje de Datagramas Perdidos Vs ACKTimeout. 94
Figura 4.27. Porcentaje de Paquetes Perdidos Vs ACKTimeout. 95
Figura 4.28. *Jitter* Máximo Vs ACKTimeout. 95
Figura 4.29. *Jitter* promedio Vs ACKTimeout. 96

LISTA DE TABLAS

Tabla 1.1. Espacio entre Tramas Medidos en Microsegundos según Espectro Ensanchado. 7
Tabla 1.2. Tamaño de la Ventana de Contención. 10
Tabla 1.3. Técnicas de transmisión en 802.11. 16
Tabla 1.4. Modos de operación del Nivel Físico de 802.11a/g. 18
Tabla 1.5. Distribución Mundial de Frecuencias para 802.11 b/g. 22
Tabla 1.6. Subcampo Política de ACK. 31
Tabla 2.1. Modos de Operación de Madwifi. 36
Tabla 2.2. Conjunto de Instrucciones Aritméticas y Lógicas. 51
Tabla 2.4. Conjunto de Instrucciones de Transferencia de Datos. 51
Tabla 2.4. Conjunto de Instrucciones de Rotación y Desplazamiento. 51
Tabla 2.6. Conjunto de Instrucciones de Ramificación. 52
Tabla 2.7. Otras Instrucciones. 52
Tabla 2.8. Descripción de Operandos. 52
Tabla 2.9. Registros. 53
Tabla 2.10. Bits de Configuración Registro de Condición 4. 54
Tabla 3.1. Factores que Afectan el Desempeño. 63
Tabla 3.2. Configuración *iperf*. 66
Tabla 3.3. Parámetros adicionales de *iperf*. 66
Tabla 3.4. Herramientas Software Utilizadas para Evaluar el Desempeño de los Mecanismos de Reconocimiento. 70
Tabla 3.5. Categorías de Acceso para WMM. 72
Tabla 4.1. Configuración Inalámbrica del AP. 76
Tabla 4.2. Características del Enlace FIET – Tablazo. 78
Tabla 4.3. Características del Enlace FIET – Cajete. 79
Tabla 4.4. Ejemplo Configuración *iperf*. 80
Tabla 4.5. Configuración *Softphones.*. 83
Tabla 4.6. Equivalencia entre distancia y ACKTimeout. 84
Tabla 4.7. *Throughput* Promedio Medido desde el AP hacia el Cliente (Mbps). 84
Tabla 4.8. *Throughput* Promedio Medido desde el Cliente hacia el AP (Mbps). 84
Tabla 4.9. *Throughput, Jitter* y Porcentaje de Datagramas Perdidos Medidos Desde el AP Hacia el Cliente. 85
Tabla 4.10. *Throughput, Jitter* y Porcentaje de Datagramas Perdidos Medidos Desde el Cliente Hacia el AP. 85
Tabla 4.11. Porcentaje de Paquetes Perdidos, *Jitter* Máximo y *Jitter* Promedio Utilizando la Ley de Compresión U del Códec G.711. 87
Tabla 4.12. Porcentaje de Paquetes Perdidos, *Jitter* Máximo y *Jitter* Promedio Utilizando la Ley de Compresión A del Códec G.711. 87
Tabla 4.13. ACK Duplicados, Retransmisiones y Paquetes HTTP Transmitidos. 89
Tabla 4.14. *Throughput* Medido desde el AP Hacia el Cliente y desde el Cliente hacia el AP (Mbps). 90
Tabla 4.15. *Throughput, Jitter* y Datagramas Perdidos Medidos desde el AP Hacia el cliente y desde el cliente Hacia el AP. 90
Tabla 4.16. Porcentaje de Paquetes Perdidos, *Jitter* Máximo y *Jitter* Promedio Utilizando la ley de compresión U y la ley de compresión A del Códec G.711. 90
Tabla 4.17. ACK Duplicados, Retransmisiones y Paquetes HTTP Transmitidos. 90
Tabla 4.18. Equivalencia entre Distancia y ACKTimeout. 91
Tabla 4.19. *Throughput* Medido desde el AP hacia el Cliente. 91
Tabla 4.20. *Throughput* Medido desde el Cliente hacia el AP. 91
Tabla 4.21. *Throughput, Jitter* y Porcentaje de Datagramas Perdidos medidos desde el AP hacia el Cliente. 92
Tabla 4.22. *Throughput, Jitter* y Porcentaje de Datagramas Perdidos medidos desde el Cliente hacia el AP. 92
Tabla 4.23. Porcentaje de Paquetes Perdidos, *Jitter* Máximo y *Jitter* Promedio utilizando la ley de compresión U del Códec G.711. 94
Tabla 4.24. Porcentaje de Paquetes Perdidos, *Jitter* Máximo y *Jitter* Promedio utilizando la ley de compresión A del Códec G.711. 94

Tabla 4.25. ACK Duplicados, Retransmisiones y Paquetes HTTP Transmitidos. 96
Tabla 4.26. *Throughput* medido desde el AP hacia el Cliente y desde el Cliente hacia el AP. 96
Tabla 4.27. *Throughput, Jitter* y Porcentaje de Datagramas Perdidos medidos desde el AP hacia el Cliente y desde el Cliente hacia el AP. 97
Tabla 4.28. Porcentaje de Paquetes Perdidos, Delta Máximo, *Jitter* Máximo y *Jitter* Promedio utilizando la ley de compresión U y la ley de compresión A del Códec G.711. 97
Tabla 4.29. ACK duplicados, Retransmisiones y Paquetes HTTP Transmitidos. 97

INTRODUCCIÓN

El área rural de países desarrollados y en vía de desarrollo se ha convertido en un sector poco atractivo para los operadores de redes de telecomunicaciones, quienes debido a la baja densidad de habitantes, prefieren no invertir en un mercado en el que difícilmente recuperarían los costos de la inversión. En cambio, emplean gran parte de sus recursos en desarrollo e investigación para las áreas urbanas de países industrializados y en vía de desarrollo cuya densidad de habitantes es alta [1] y en donde las soluciones más completas se basan en redes cableadas como el par de cobre, el cable coaxial, la fibra óptica y soluciones de acceso inalámbrico como WiFi[1] y el sistema de Interoperabilidad Mundial para Acceso por Microondas (WiMAX, *Worldwide Interoperability for Microwave Access*)[2].

Al considerar una zona rural, se aprecia fácilmente que una solución inalámbrica es conveniente para este entorno, pues, brinda buena cobertura y fácil acceso en áreas distantes, superando así, algunos aspectos que bloquean la inversión de los operadores de telecomunicaciones. A pesar de que existen algunas tecnologías inalámbricas en estas zonas, como el Sistema Global para Comunicaciones Móviles *(*GSM, *Global System for Mobile communications)*[3] o los Terminales de Apertura muy Pequeña (VSATs, *Very Small Aperture Terminals)*[4], éstas no brindan servicios de datos (Internet) a costos sostenibles para los usuarios rurales. Tecnologías de gran ancho de banda y largo alcance, como WiMAX, cuya infraestructura de red y terminales son de alto costo, requieren un mercado masivo de usuarios para ofrecer servicios económicos, a diferencia de WiFi, que se destaca por ser una propuesta económica y versátil para brindar acceso a Internet a zonas poco pobladas y faltas de infraestructura de redes.

WiFi nació como una solución de acceso inalámbrico para espacios reducidos, tales como hogares u oficinas, sin embargo, estas condiciones pueden variar con el uso de antenas externas y amplificadores[5] que permiten llevar la señal a lugares distantes. El alcance de esta tecnología se empieza a explotar con la interconexión entre sedes empresariales y el establecimiento de redes metropolitanas con nodos distantes. Estas experiencias motivaron a la comunidad a hacer de WiFi una plataforma para llegar a lugares de difícil acceso o con topografías complicadas. Entidades como la fundación Enlace Hispanoamericano de Salud (EHAS) y programas como el Fondo Regional para la Innovación Digital en América Latina y el Caribe (FRIDA) han fomentado proyectos para brindar conectividad a poblaciones rurales tecnológicamente desatendidas. El proyecto "*Análisis, uso, adecuación y apropiación de servicios sobre tecnologías inalámbricas en zonas de difícil acceso de las poblaciones indígenas del Cauca andino*" financiado por FRIDA y la Universidad del Cauca busca llevar una conexión de banda ancha a algunas zonas rurales del departamento del Cauca y al ser WiFi la solución más adecuada para este proyecto, surgen desafíos en cuanto a la implementación y la adaptación de dicha tecnología a enlaces de larga distancia.

Adecuar WiFi para su correcto funcionamiento en zonas rurales distantes implica el estudio del estándar IEEE 802.11, el cual define el uso de los dos niveles inferiores del modelo de Interconexión de Sistemas Abiertos (OSI, *Open System Interconnection*) en una Red Inalámbrica de Área Local (WLAN, *Wireless Local Area Network*). Este estándar se define apropiadamente para funcionar en entornos pequeños pero enfrenta muchas falencias cuando opera en largas distancias. Al utilizar WiFi para un ambiente para el que no fue creado se presenta la necesidad de ajustar varios parámetros, uno de ellos es el

---

[1] Término desarrollado por la Alianza WiFi (WFA, *WiFi Alliance*) para describir productos de Red de Área Local Inalámbricos (WLAN, *Wireless Local Area Network*) basados en los estándares IEEE 802.11.

[2] Estándar de transmisión inalámbrica de datos (IEEE 802.16) que proporciona tasas de transferencia de 124Mbit/s y una cobertura a distancias de entre 40-70 kilómetros de una estación base, utilizando tecnología que no requiere línea de vista.

[3] Sistema de telefonía móvil digital ampliamente utilizada en todo el mundo. Es considerada la segunda generación (2G) celular.

[4] Redes privadas de comunicación de datos vía satélite para intercambio de información.

[5] La potencia radiada por los amplificadores debe obedecer la regulación propia de cada país.

ACKTimeout[6], esto debido a que el tiempo de ida y vuelta de una trama[7] es mucho mayor en largas distancias comparado a una configuración *indoor* donde sólo dura unos microsegundos[8] [2] .

Según lo expuesto anteriormente, el objetivo general de este proyecto, ha consistido en evaluar el desempeño de los mecanismos de reconocimiento del estándar 802.11 modificados para enlaces punto a punto de larga distancia, de manera que se pueda encontrar una forma adecuada de llevar conectividad inalámbrica a las regiones apartadas del departamento. Así, se desarrolló el documento de la siguiente manera:

En el Capítulo 1, se hace una presentación del estándar 802.11 teniendo en cuenta los aspectos de interés para el desarrollo del proyecto. Se presentan las limitaciones que tiene el estándar 802.11 en enlaces de larga distancia, al igual que las limitaciones de los mecanismos de reconocimiento. Así, se dió cumplimiento al primer objetivo propuesto.

En el Capítulo 2 se expone una plataforma de desarrollo para el Control de Acceso al Medio de 802.11 que trabaja sobre los dispositivos que poseen un *chip* Broadcom 43XX. Esta plataforma surgió como resultado del proceso de desarrollo del segundo objetivo propuesto.

El Capítulo 3 presenta los parámetros de desempeño que se tuvieron en cuenta para evaluar el comportamiento de los mecanismos de reconocimiento, además, se presentan las herramientas que permitieron hacer estas mediciones y los procedimientos necesarios para alternar y modificar los mecanismos, dando cumplimiento al tercer objetivo propuesto.

Por último, en el Capítulo 4 se presentan los escenarios sobre los cuales se realizaron las pruebas, las configuraciones de los equipos de red y las aplicaciones. Finalizando con los resultados obtenidos y las conclusiones al respecto y culminando así el cuarto objetivo de este Trabajo de Grado.

[6] Tiempo de espera para el reconocimiento.
[7] La trama es la Unidad de Datos del Protocolo (PDU, Protocol Data Units), el nivel de Enlace de Datos del modelo OSI.
[8] La mayoría de las implementaciones asumen la recomendación de la IEEE para fijar el tiempo de propagación en el aire de 1µs, limitando el *ACK Timeout* a una distancia máxima de 300m.

1 PRINCIPIOS DEL ESTÁNDAR IEEE 802.11, MÉTODOS DE RECONOCIMIENTO CONTEMPLADOS EN EL ESTÁNDAR Y LIMITACIONES EN LARGAS DISTANCIAS

En este capítulo, inicialmente, se presentan las principales temáticas concernientes al estándar 802.11, base fundamental para el entendimiento de las limitantes que esta tecnología presenta en largas distancias como se evidencia en la sección 1.2. Finalmente, se describen los mecanismos de reconocimiento de 802.11 contemplados en el estándar, sección en que se fundamenta este estudio.

1.1 DESCRIPCIÓN GENERAL DEL ESTÁNDAR IEEE 802.11

1.1.1 Generalidades

IEEE 802.11 e IEEE 802.11x hacen referencia a una familia de protocolos desarrollados por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, *Institute of Electrical and Electronics Engineers*) para tecnologías de Red de Área Local Inalámbrica (WLAN, *Wireless Local Area*). Estos protocolos definen las tecnologías que implementan a 802.11 en el nivel Físico y el nivel Control de Acceso al Medio (MAC*, Medium Access Control)* del nivel de Enlace de Datos del modelo de Interconexión de Sistemas Abiertos (OSI, *Open System Interconnection*).

En 1997 se publicó la primera versión de IEEE 802.11, esta especificación, que operaba en la banda de 2.4 Ghz usaba los esquemas de transmisión Espectro Ensanchado por Salto de Frecuencia (FHSS, *Frecuency Hopping Spread Spectrum*) y Espectro Ensanchado por Secuencia Directa (DSSS, *Direct Sequence Spread Spectrum*). La especificación inicial permitía alcanzar velocidades de conexión entre 1 y 2 Mbps pero fue completamente reemplazada por la especificación 802.11b en el año 1999 [3] 802.11b. Aún permanece vigente y permite obtener una velocidad máxima de conexión de 11Mbps bajo la misma banda de frecuencia. En ese mismo año fue publicada la especificación 802.11a, que usa los mismos principios que 802.11b, pero utiliza Multiplexación por División de Frecuencias Ortogonales (OFDM, *Orthogonal Frequency Division Multiplexing,* opera en una frecuencia de 5 GHz, alcanza una velocidad máxima de 54 Mbps y resulta incompatible con los productos de 802.11b, motivo por el cual se desarrollaron pocos productos para el hogar y se prefirió incorporar el estándar al ambiente empresarial [4]. Posteriormente, basándose en los principios de 802.11a, se publicó una especificación que era compatible con 802.11b y que alcanza velocidades de 54Mbps, la 802.11g. En la actualidad la mayoría de productos son fabricados bajo las especificaciones b/g, mientras se continúa desarrollando la especificación 802.11n que se espera alcance una velocidad máxima de 300 Mbps [5]. Aunque los dispositivos que soportan 802.11 pueden sufrir interferencias por parte de hornos microondas, teléfonos inalámbricos y otros equipos que trabajan en la misma banda, características como interoperabilidad, bajos costos de producción y en especial el uso de bandas no licenciadas permiten que continúe en aumento su popularidad.

Es importante aclarar el significado del término WiFi. Éste hace referencia únicamente a los productos WLAN 802.11 a/b/g cuya interoperabilidad con otros está certificada por la WFA (*WiFi Alliance*) [6].

En el trascurso de este documento se abarcan temáticas concernientes al estándar 802.11, las cuales darán al lector una idea global del funcionamiento del mismo. De igual manera varios aspectos no serán tenidos en cuenta, pues no se relacionan directamente con el desarrollo del proyecto. Tecnologías como 802.11n u 802.11i serán omitidas, dando prioridad a ciertos componentes base de 802.11 a/b/g que se deben tener en cuenta para evaluar el desempeño de esta tecnología en largas distancias y plantear ciertas mejoras en los mecanismos de reconocimiento.

1.1.2 Componentes de la Arquitectura del Estándar IEEE 802.11

802.11 define dos componentes básicos, una estación, también llamada nodo o cliente, que usualmente es un Computador Personal (PC, *Personal Computer*) equipado con una tarjeta de red inalámbrica, y un Punto de Acceso (AP, *Access Point*). El AP consta de un transceptor de radio, una interfaz de red *Ethernet*, un *software* de interconexión de redes acorde al estándar 802.1d y actúa como la estación base para la red inalámbrica, dando acceso a múltiples estaciones inalámbricas.

Además de los componentes básicos, es necesario conocer todo el conjunto de elementos que conforman la arquitectura del estándar 802.11.

1.1.2.1 **Conjunto de Servicios**

El estándar IEEE 802.11 define tres topologías diferentes, también conocidas como Conjunto de Servicios (Service Set):

- **Conjunto de Servicios Básicos**

  Un Conjunto de Servicios Básicos (*BSS, Basic Service Set*) consta de un AP y una o más estaciones Inalámbricas tal como lo muestra la Figura 1.1 [7]. Usa el modo infraestructura, en el cual todo el tráfico inalámbrico viaja desde y hacia el AP, ya que las trasmisiones cliente a cliente no están permitidas. El BSS cubre una única celda y tiene un único Identificador de Conjunto de Servicio (SSID, *Service Set Identifier*).

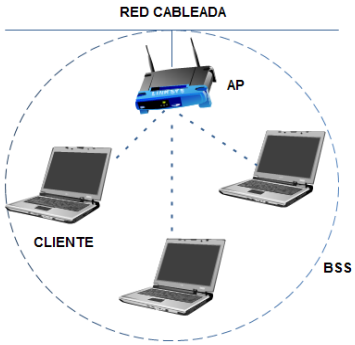


Figura 1.1 Conjunto de Servicios Básicos.

- **Conjunto de Servicios Extendidos**

  Un Conjunto de Servicios Extendidos (ESS, *Extended Service Set*) es un conjunto de dos o más BSSs que forman una subred como se muestra en la Figura 1.2 [7]. Un ESS cubre múltiples celdas, de tal manera que permite realizar *Itinerancia*[9] y no requiere de un mismo SSID para los BSSs.

[9] Capacidad del nodo cliente para cambiar de un área de cobertura a otra sin interrupción en el servicio o pérdida en conectividad.

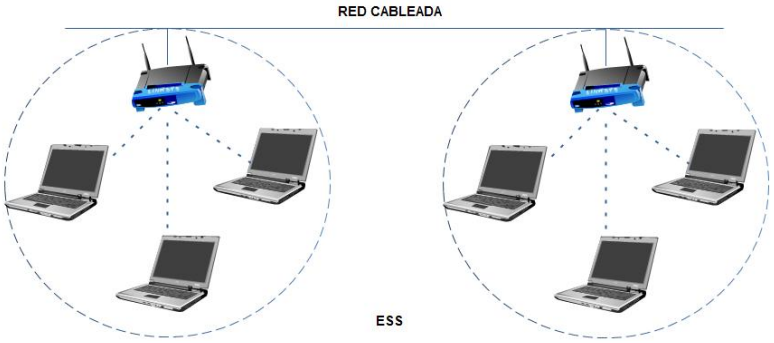


Figura 1.2. Conjunto de Servicios Extendidos.

- **Conjunto de Servicios Básicos Independientes**

  Un Conjunto de Servicios Básicos Independientes (IBSS, *Independent Basic Service Sets*) o red *ad hoc* consta de un grupo de estaciones 802.11 que se comunican directamente entre ellas, sin usar un AP o alguna conexión cableada tal como lo ilustra la Figura 1.3 [7]. Debido a que las estaciones realizan conexiones directas entre sí, también es conocida como una red Ad-Hoc.

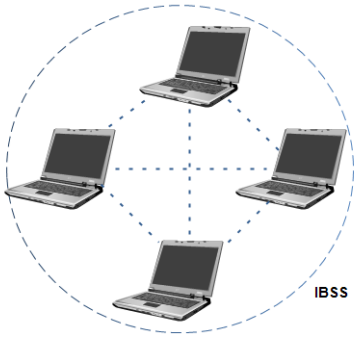


Figura 1.3. Conjunto de Servicios Básicos Independientes.

En el estándar no se encuentra un límite para el número de dispositivos que pueden conformar un IBSS, sin embargo, es posible que algunos miembros no se puedan comunicar con otros debido al problema de nodos escondidos[10] [7].

1.1.2.2 **Sistema de Distribución**

Además de existir de manera independiente, un BSS, puede formar parte de una red extendida, la cual se construye con múltiples BSSs. El componente usado para interconectar los BSSs es el Sistema de Distribución (DS, *Distribution System*). La mayoría de las implementaciones de 802.11 usa un AP como red de acceso a la red de transporte *Ethernet* 802.3, el cual sirve como medio del sistema de distribución.

[10] Situación que se presenta cuando un dispositivo inalámbrico puede transmitir con la potencia suficiente para que sea escuchado por un nodo receptor, pero no por otra estación que también desea transmitir y que por tanto no detecta la transmisión.

1.1.2.3 **Sistema de Distribución Inalámbrico**

Aunque el DS normalmente usa un *backbone ethernet* cableado, es posible que use una conexión inalámbrica. Un Sistema de Distribución Inalámbrico (WDS, *Wireless Distribution System*) puede operar con APs que tienen un único radio tal como lo muestra la Figura 1.4 [8] o con APs que poseen dos radios.

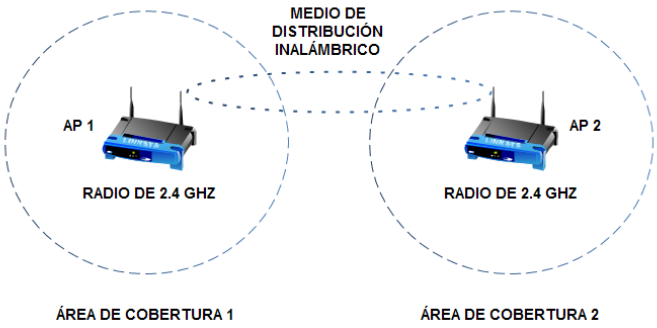


Figura 1.4. Sistema de Distribución Inalámbrico.

En el caso de la Figura 1.4, los APs dan acceso a las estaciones cliente y se comunican entre sí directamente. La desventaja de esta solución es que el *throughput*[11] puede ser afectado debido a la naturaleza *half duplex*[12] del medio, particularmente en un escenario en el que un AP no se puede comunicar con una estación y con otro AP al mismo tiempo.

El otro caso se presenta cuando cada AP tiene dos radios operando en frecuencias diferentes. El radio que opera en la frecuencia de 2.4 GHz provee acceso a las estaciones cliente, mientras que el de 5 GHz sirve como enlace WDS entre dos APs. De esta forma, el *throughput* no se ve afectado debido a que los radios de 2.4 GHz se pueden comunicar al mismo tiempo que los radios de 5 GHz.

1.1.2.4 **Identificador de Conjunto de Servicio**

El Identificador de Conjunto de Servicio (SSID) es un nombre usado para identificar una red inalámbrica 802.11. Las tres topologías contempladas en el estándar IEEE 802.11 utilizan el SSID para que los radios sean capaces de identificarse entre sí en un proceso conocido como escaneado activo o escaneado pasivo. El SSID es una característica configurable tanto para los APs como para las estaciones cliente.

La mayoría de los APs tienen la habilidad de esconder el SSID y así mantener el nombre de la red oculto para usuarios no autorizados. Sin embargo, es una forma muy insegura de restringir el acceso a la red y no se encuentra definida en el estándar.

1.1.2.5 **Área de Servicio Básico**

El Área de Servicio Básico (BSA, *Basic Service Area) es* el área física que cubre un AP en un BSS. Las estaciones cliente se pueden mover a través de esta área y mantener la comunicación con el AP. El tamaño y la forma de la BSA depende de muchas variables como: potencia de transmisión del AP, ganancia de la antena, estructura física alrededor del AP, etc.

---

[11] Número de bits transmitidos entre el punto de acceso y el cliente durante un periodo de tiempo (tasa de transferencia de datos).
[12] El medio permite transmisiones en los dos sentidos pero no de manera simultánea.

1.1.2.6 **Nivel MAC del Estándar IEEE 802.11**

El nivel de Control de Acceso al Medio se encarga principalmente de controlar el acceso al canal de radio y de la detección de portadora. El estándar 802.11 define en su nivel MAC una serie de funciones que permiten gestionar y mantener las comunicaciones entre estaciones 802.11.

1.1.2.7 **Mecanismo de Acceso al Medio en una WLAN**

El mecanismo de acceso al medio de una red *Ethernet* es conocido como Acceso Múltiple por Detección de Portadora con Detección de Colisiones (CSMA/CD, *Carrier Sense Multiple Access with Collision Detection*), y el de una WLAN es conocido como Acceso Múltiple por Detección de Portadora con Evasión de Colisiones (CSMA/CA, *Carrier Sense Multiple Access with Collision Avoidance*) [8]. Los dos mecanismos funcionan de manera similar; cuando una estación desea transmitir, primero debe determinar el estado del medio para ver si otro dispositivo ya está transmitiendo, en el caso de que otro dispositivo lo esté haciendo, entonces, la estación debe esperar hasta que el medio esté disponible. De manera que cuando un nodo CSMA/CD quiere transmitir y no hay ningún otro transmitiendo, puede empezar inmediatamente la transmisión y si una colisión ocurre mientras éste está transmitiendo, la colisión será detectada y el nodo detendrá la transmisión, seguidamente entra a un estado en el que espera un tiempo aleatorio para reintentar la transmisión. Por otra parte, las estaciones Inalámbricas al no poder transmitir y recibir simultáneamente, no tienen la capacidad de detectar una colisión durante la transmisión, es por esto que en lugar de transmitir inmediatamente queda libre el medio, esperan un tiempo aleatorio, y una vez trascurrido comienza el proceso de envío de los datos. Para comprobar que no ocurrió colisión usan un mecanismo de reconocimiento positivo de los paquetes recibidos.

La Función de Coordinación Distribuida (DCF, *Distributed Coordination Function*) es el principal método de acceso al medio especificado en el estándar IEEE 802.11, sin embargo, éste también contempla un método de acceso al medio opcional conocido como Función de Coordinación Puntual (PCF, *Point Coordination Function*), pero debido a que la mayoría de los dispositivos WiFi no lo soportan o simplemente no lo implementan [9], no será explicado en este documento. La Función de Coordinación Distribuida será analizada más adelante después de comprender algunos elementos importantes de CSMA/CA.

1.1.2.8 **Espacio entre Tramas**

Las especificaciones IEEE 802.11 a/b/g definen 4 tipos de Espacios entre Tramas (IFS, *InterFrame Space*): Espacio entre Tramas Corto (SIFS, *Short InterFrame Space*), Espacio entre Tramas DCF (DIFS, DCF *InterFrame Space*), Espacio Entre Tramas PCF (PIFS, PCF *InterFrame Space*) y Espacio entre Tramas Extendido (EIFS, *Extended InterFrame Space*). Estos se utilizan en una WLAN para enviar ciertos tipos de mensajes a través de la red o para manejar los intervalos durante los cuales las estaciones compiten por el medio. En una red inalámbrica, tanto las estaciones como los APs están sincronizados y por lo tanto usan cantidades de tiempo estándar (espacios) para realizar tareas. La Tabla 1.1 [7] muestra el tiempo en microsegundos de los espacios entre tramas.

Tabla 1.1. Espacio entre Tramas Medidos en Microsegundos según Espectro Ensanchado.

	DSSS	**FHSS**	**OFDM**
SIFS	10	28	16
PIFS	30	78	25
DIFS	50	128	34
EIFS[13]	-	-	-

[13] EIFS se deriva del tiempo SIFS, DIFS y el tiempo que toma transmitir una trama de control ACK a la velocidad de transmisión de datos más lenta, por esto no se especifica su duración.

A continuación se explican los tipos de espacios entre tramas más importantes del estándar 802.11:

- Espacio entre Tramas Corto

  El espacio entre tramas SIFS es el más corto de los cuatro y es un espacio de tiempo que va antes y después del envío de mensajes tales como Despejado para Transmitir (CTS, *Clear to Send Frame*) usado por las estaciones receptoras para dar respuesta a una trama Petición para Transmitir (RTS, *Request to Send Frame*)[14] y Reconocimiento (ACK, *Acknowledgement*) usado por la estación receptora para informar a la estación transmisora que los datos llegaron con el formato adecuado.

  SIFS da el nivel de prioridad más alto en una red inalámbrica. Cuando una estación va a transmitir, primero debe determinar el estado del medio y si éste se encuentra libre debe esperar cierta cantidad de tiempo para empezar a transmitir. Dicha cantidad de tiempo está determinada por la función que la estación necesita realizar. Cada función, de acuerdo a su prioridad, entra en una de las categorías de IFS. Así, las tareas de mayor prioridad entran en la categoría SIFS, en la cual esperan tiempos más cortos antes de empezar la transmisión.

- Espacio entre Tramas PCF

  El espacio entre tramas Función de Coordinación Puntual (PCF) lo utilizan los APs únicamente cuando la red está en modo PCF. Éste proporciona más prioridad que DIFS y menos que SIFS.

- Espacio entre Tramas DCF

  El espacio entre tramas Función de Coordinación Distribuida (DCF) se utiliza por defecto en todas las estaciones 802.11 que operan en modo DCF. Éstas usan el espacio DIFS para transmitir tramas de datos y de administración y tienen el nivel más bajo de prioridad.

- Espacio entre Tramas Extendido

  EIFS es un espacio de longitud variable y de mayor duración que los anteriores. Es usado como periodo de espera en el caso de que una estación escuche una trama cuyo FCS es incorrecto y desee transmitir, dando suficiente tiempo para que la estación destino reconozca la trama (puede haber sido correctamente recibida por ella) antes de que la primera comience a transmitir.

1.1.2.9 **Intervalo de Tiempo**

Es un periodo de tiempo estándar que se encuentra pregrabado en los radios WiFi y su valor depende de la técnica de transmisión que se use. Así, para FHSS el intervalo (*slot)*[15] de tiempo es de 50 µs, para DSSS es de 20 µs y para OFDM es de 9 µs.

Los *slots* de tiempo se utilizan de igual forma que el segundero de un reloj. Un nodo inalámbrico marca los *slots* de tiempo así como un reloj marca los segundos [1].

1.1.2.10 **Evasión de Colisiones**

En un entorno inalámbrico, la detección de colisiones se hace imposible, pues, los equipos de 802.11 no tienen la capacidad de transmitir y recibir información al mismo tiempo. Por consiguiente 802.11

[14] El mensaje RTS se usa por las estaciones transmisoras para reservar el medio con el fin de asegurar que ninguna estación esté transmitiendo.
[15] Se acostumbra dejar en inglés.

implementa el uso de Reconocimientos (ACKs, *Acknowledgements*), así, cada vez que un radio transmite una trama, si ésta es recibida correctamente, el radio receptor responderá con una trama de reconocimiento. De esta forma la estación transmisora sabrá que el envío de la información fue exitoso. En caso de que dicha estación no reciba el ACK, deberá retransmitir la información. Este método no detecta colisiones, pero si la estación transmisora no recibe la trama ACK, se asume que ocurrió una colisión, que se perdió la información o el mensaje ACK, debido a interferencias, baja relación señal a ruido, errores en la transmisión, etc.

1.1.2.11 **Duración/Identificación (Duration/Id)**

El campo Duración/Identificación (Duration/Id) hace parte del encabezado MAC de la trama 802.11. Cuando un cliente transmite una trama, el campo Duración contiene un valor de 0 a 32767 y representa el tiempo en microsegundos necesario para transmitir el ACK más el intervalo SIFS. El transmisor calcula cuánto tiempo tardará en llegar la trama ACK e incluye este valor en el campo duración del encabezado MAC de la trama transmitida.

1.1.2.12 **Detección de Portadora**

Existen dos formas de detectar la portadora:

- Detección Virtual de Portadora.

  Usa un temporizador conocido como Vector de Asignación de Red (NAV, *Network Allocation Vector*) que mantiene una predicción del tráfico futuro. Cuando un equipo 802.11 no está transmitiendo, está escuchando. En este caso, cuando escucha una trama de transmisión de otra estación, observa el encabezado de la trama y determina si el campo Duration/ID contiene un valor de duración o un valor ID. Si el campo contiene un valor de duración, la estación oyente pondrá el temporizador NAV en este valor. Así, se usará el NAV como un temporizador descendente que indica que el medio RF estará ocupado hasta que el temporizador llegue a cero.

- Detección Física de Portadora.

  Es posible que cuando una estación escuche el medio no detecte la trama transmitida por el radio transmisor y por lo tanto no pueda leer el valor del campo Duración ni establecer el temporizador NAV. Así, que CSMA/CA utiliza otra línea de defensa para asegurarse que una estación no transmita mientras otra ya lo está haciendo; se conoce como detección física de portadora y es realizada constantemente por todas las estaciones que no están transmitiendo o recibiendo. Uno de los propósitos de éste método es determinar si la transmisión de una trama está en espera para ser recibida por otra estación y en el caso de que el medio esté ocupado el radio intentará sincronizar con la transmisión. El otro propósito es determinar si el medio está ocupado antes de empezar a transmitir. Esto es conocido como Evaluación de Canal Libre (CCA, *Clear Channel Assessment*) y opera a nivel físico.

  La detección virtual de portadora y la detección física de portadora ocurren al tiempo, siendo la detección física el método principal para evitar colisiones.

1.1.2.13 **Tiempo Backoff Aleatorio**

Después de haber llevado a cabo la detección física y virtual de portadora, una estación puede competir por el medio durante una ventana de tiempo conocida como ventana de contienda (CW, *Contention Window*). En este punto, la estación selecciona un valor aleatorio para el tiempo *Backoff* en un rango de cero al valor inicial de la ventana de contienda. Luego, este valor se multiplica por los *slots* de tiempo, o periodos de tiempo que dependen de la técnica de transmisión que se use. Entonces, el temporizador

Backoff empieza a contar descendentemente de acuerdo a los *slots* y cuando éste llegue a cero, la estación puede empezar a transmitir [8].
El temporizador *Backoff* es un método que ayuda a minimizar la probabilidad de que dos estaciones transmitan al mismo tiempo, pero no es totalmente seguro. Así que si una estación no recibe un ACK, asume que existe una colisión y reenvía la trama que no fue reconocida. La Tabla 1.2 [8] muestra el tamaño de la ventana de contienda según el número de reintentos y según la técnica de transmisión que se use.

Tabla 1.2. Tamaño de la Ventana de Contención.

	INTENTO INICIAL	PRIMER REINTENTO	SEGUNDO REINTENTO	TERCER REINTENTO	CUARTO REINTENTO	QUINTO REINTENTO	SEXTO REINTENTO
802.11 FHSS, 802.11a/g	15 SLOTS	31 SLOTS	63 SLOTS	127 SLOTS	255 SLOTS	511 SLOTS	1023 SLOTS
802.11 DSSS, 802.11b.	31 SLOTS	63 SLOTS	127 SLOTS	255 SLOTS	511 SLOTS	1023 SLOTS	

1.1.2.14 Función de Coordinación Distribuida

La Figura 1.5 [8] muestra el diagrama de flujo del proceso de Función de Coordinación Distribuida (DCF, *Distributed Coordination Function*).

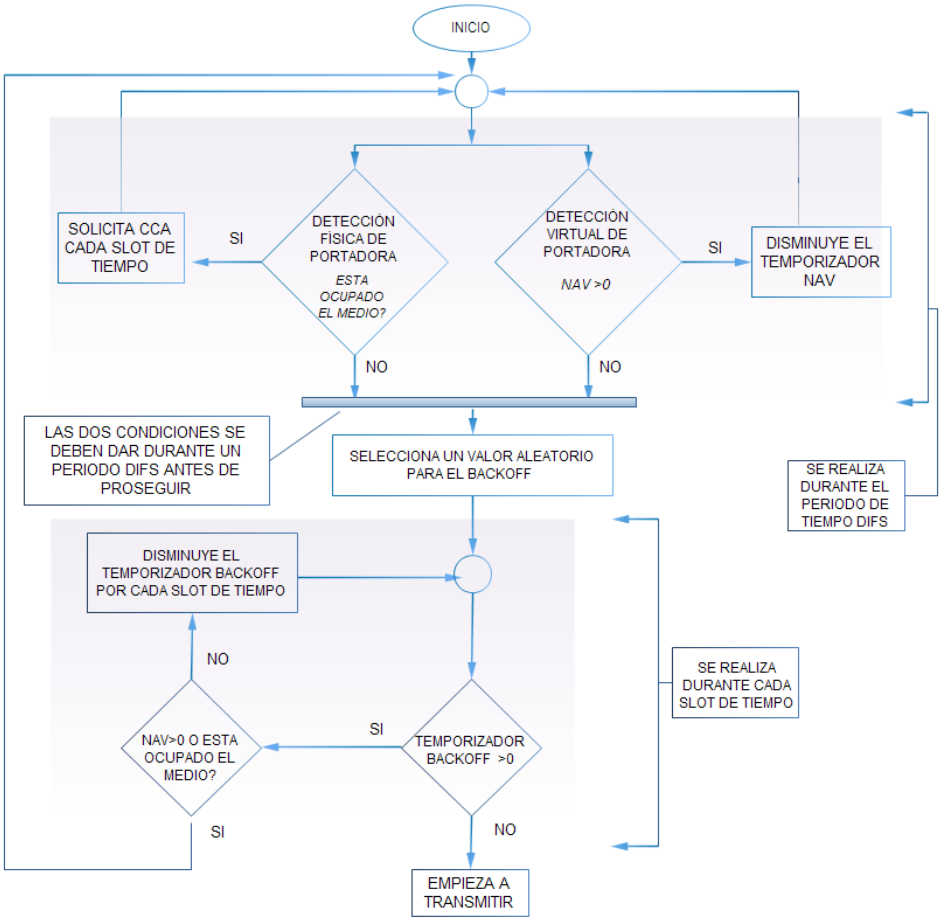


Figura 1.5. Diagrama de Flujo del Proceso de Función de Coordinación Distribuida.

El proceso comienza en la parte superior de la figura 1.5 (inicio) y continúa de forma descendente con dos puntos de decisión. Uno de ellos es la detección física de portadora y el otro es la detección virtual de portadora, y ambos se realizan simultáneamente. En el punto de decisión de la derecha, si el temporizador NAV es mayor que cero, significa que otra estación está usando y continuará usando el medio por un tiempo y por lo tanto la estación deberá esperar. Ésta disminuye el temporizador y vuelve al inicio y a los puntos de decisión. Las estaciones solo actualizan el valor del temporizador NAV cuando éste es mayor al que actualmente tienen.

En el punto de decisión de la izquierda, si la Evaluación de Canal Libre (CCA) indica que el medio está ocupado, significa que otra estación está transmitiendo y que la estación tiene que esperar. Ésta debe esperar un *slot* de tiempo y volver al inicio para realizar otra detección de canal libre. Cuando los dos puntos de decisión determinan que no hay tráfico inalámbrico durante un periodo DIFS, el proceso continúa.

Inmediatamente después, la estación selecciona un valor aleatorio para el *Backoff,* y luego pasa a un punto de decisión en el que evalúa si dicho temporizador es mayor que cero o no. En el caso de que sea mayor que cero, la estación deberá realizar de nuevo la detección física y virtual de portadora para determinar si alguna otra estación ha empezado a transmitir. Si detecta que el medio aún está libre, disminuirá el valor del temporizador *Backoff* y volverá al punto de decisión para ver si éste ya ha llegado a cero, y en caso de ser así, puede empezar a transmitir.

Si el temporizador *Backoff* no ha llegado a cero y otra estación ha empezado a transmitir, entonces, conserva el valor del *Backoff* y va al inicio donde vuelve a empezar todo el proceso. La única diferencia en este nuevo ciclo, es que la estación no vuelve a escoger un valor aleatorio para el *Backoff* sino que continúa con el valor que conservó en el ciclo inmediatamente anterior.

1.1.2.15 **Tipos de Tramas**

A diferencia de muchos estándares para redes cableadas, tales como el IEEE 802.3, el cual usa un único tipo de trama de datos, el estándar IEEE 802.11 define tres tipos diferentes de tramas:

- Tramas de Gestión.

  Las estaciones inalámbricas utilizan las tramas de gestión para acceder a una red o para abandonarla. En una red cableada, dichas tramas no son necesarias, debido a que al conectar o desconectar físicamente el cable de red se realizan estas funciones. Sin embargo, en un ambiente inalámbrico, la estación inicialmente debe encontrar una red compatible, luego autenticarse (en el caso de que le sea permitido conectarse) y asociarse con la red (típicamente con un AP) para que finalmente pueda acceder a una red cableada. A continuación se muestra la lista de los subtipos de tramas de gestión definidos en el estándar 802.11:

  - Faro (*Beacon*).
  - Solicitud de Exploración (*Probe request*).
  - Respuesta a Exploración (*Probe response*).
  - Autenticación (*Authentication*).
  - Desautenticación (*Deauthentication*).
  - Solicitud de Asociación (*Association Request*).
  - Respuesta a Asociación (*Association Response*).
  - Disociación (*Disassociation*).
  - Solicitud de Re asociación (*Reassociation Request*).
  - Respuesta a Re asociación (*Reassociation Response*).
  - Mensaje de Anuncio de Indicación de Tráfico (ATIM, *Announcement Traffic Indication Message*).

- Tramas de Control.

  Estas tramas se encargan de controlar la transmisión y la recepción de las tramas de datos. Deben ser escuchadas por todas las estaciones, así que deben ser transmitidas a una velocidad de transmisión de datos básica. A continuación se encuentran los subtipos de tramas de control del estándar 802.11 usados por DCF:

  - Ahorro de Energía (PS - POLL, *Power Save*).
  - Petición para Transmitir (RTS, *Request to Send*).
  - Despejado para Transmitir (CTS, *Clear to Send*).
  - Reconocimiento (ACK, *Acknowledgment*).

- Tramas de Datos.

  Las tramas de datos llevan la información que viene de protocolos de nivel superior. Las tramas nulas se usan en el modo ahorro de energía para informar al AP de los cambios en el estado del ahorro de energía. A continuación se encuentran los subtipos de tramas de datos usados por DCF.

  - Datos (*Data*).
  - Función Nula (sin datos, *Null Function*) [8].

1.1.2.16 **Formato y Análisis de Tramas**

Los tres tipos de tramas del estándar 802.11 se basan en una trama MAC general. Éstos usan porciones específicas de la misma y de ser necesario aumentan campos de acuerdo a sus propósitos. La Figura 1.6 [10] ilustra el formato general de trama de 802.11.

Control de Trama	Duración / ID	Dirección 1	Dirección 2	Dirección 3	Control de Secuencia	Dirección 4	Control de QoS	Cuerpo de la Trama	FCS
2 Bytes	2 Bytes	6 Bytes	6 Bytes	6 Bytes	2 Bytes	6 Bytes	2 Bytes	0 - 2312 Bytes	4 Bytes

Figura 1.6. Trama MAC General.

Para los campos de la trama general, se tiene que el campo *Duración/ID* contiene el tiempo en microsegundos que puede tardar el intercambio de tramas entre estaciones DCF, al igual que algunas características para el caso en el que exista ahorro de energía. Los campos *Dirección 1, 2, 3, 4* varían según el tipo de trama que se esté usando. El campo *Control de Secuencia* está constituido por el número de fragmento y el número de secuencia, los cuales definen la trama y el número de fragmento en la trama. El campo *Control de Calidad de Servicio* (QoS, *Quality of Service*) presente en cierto tipo de tramas, identifica el tipo de datos presentes en la carga útil de la trama e información relacionada al trato que debe recibir por parte de la red. Para terminar, el campo *Secuencia de Verificación de Trama* (FCS, *Frame Check Sequence),* es una secuencia de 16 bits que permite verificar la correcta transmisión de la trama y la detección de posibles errores en la misma [10].

A su vez, el campo Control de Trama se subdivide en 11 campos, como lo indica la Figura 1.7 [10].

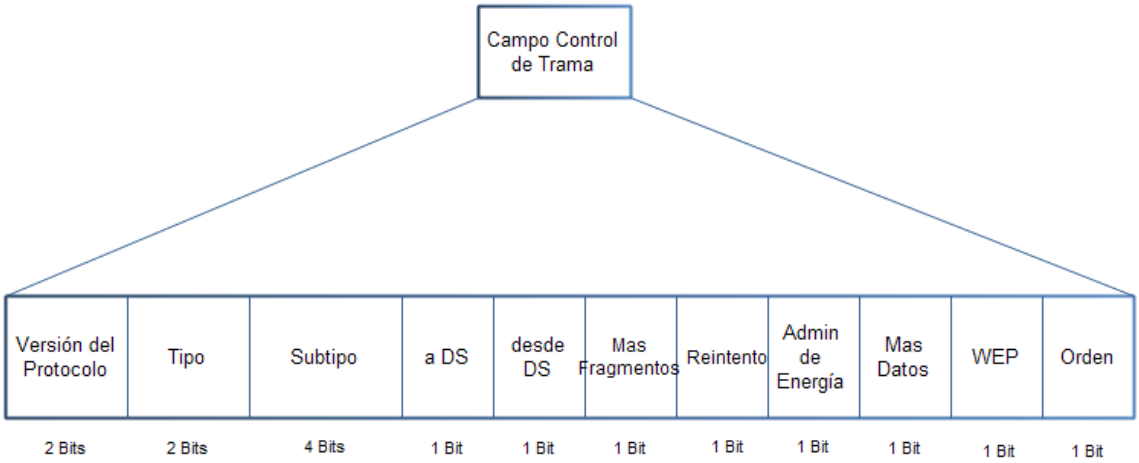


Figura 1.7. Campo Control de Trama.

El campo *Versión del Protocolo* específica la versión del protocolo MAC de 802.11, su valor siempre es cero debido a que hasta el momento solo existe una versión. El campo *Tipo* indica si se trata de una trama de gestión, de control o de datos. El campo *Subtipo* define el subtipo de trama. *A DS* muestra si la trama es dirigida al Sistema de Distribución o si está siendo enviada a una estación asociada a un AP, y *desde DS* muestra si la trama es proveniente del Sistema de Distribución. El campo *Más Fragmentos* especifica si la trama es única o si se debería esperar más fragmentos. El campo *Reintento* muestra si la trama está siendo retransmitida y de esta forma le permite al receptor descartar tramas duplicadas. *Administración de Energía* detalla si se está en modo de ahorro de energía o no. Cuando el campo *Más Datos* lleva el valor de 1, el AP le notifica a la estación que tiene más datos destinados para ella en el *Buffer*[16]. El campo *Trama Protegida* muestra si los datos que van en el cuerpo de la trama están siendo encriptados mediante algún algoritmo de cifrado y finalmente, cuando el campo *Orden* está en 1 indica que las tramas y fragmentos van a ser transmitidas en un orden estricto.

A continuación se explica el modo de operación de algunas Tramas de Gestión**,** conocimiento relevante para el completo entendimiento del estándar:

- Trama *Beacon*:

  Las tramas de gestión *Beacon* son tramas cortas enviadas desde el AP hacia las estaciones (en un BSS) o desde una estación hacia otra estación (en un IBSS) con el fin de organizar y sincronizar la comunicación inalámbrica. Estas tramas sincronizan a las estaciones cliente poniendo un sello de tiempo al momento exacto de la transmisión. Una vez el cliente reciba la trama *Beacon* cambia su reloj para igualar al reloj del AP y así quedar sincronizados. Estas tramas no solo sirven para sincronizar, también llevan la información que necesita una estación cliente antes de hacer parte de un BSS, por ejemplo, parámetros de la técnica de transmisión el SSID de la red, velocidades de transmisión de datos soportadas, etc. Las tramas *Beacon* también son utilizadas para llevar a cabo el proceso conocido como escaneado pasivo, mediante el cual, el AP informa a las estaciones cliente que se encuentra activo.

[16] Memoria de almacenamiento temporal de información. Se utiliza para compensar la diferencia de tiempos y velocidades que manejan los distintos dispositivos.

- Trama Solicitud de Exploración y Trama Respuesta a Exploración:

  A diferencia del escaneado pasivo, es posible que sea la estación cliente la que busque al AP, método que es conocido como escaneado activo. En este caso la estación cliente envía al AP la trama Solicitud de Exploración la cual puede contener el SSID de la red a la que se desea conectar o puede no especificar ningún SSID. En el primer caso, solo los APs que tengan configurado ese SSID responderán con la trama Respuesta a Exploración y en el caso en el que no se especifique el SSID todos los APs deberán responder. La trama Respuesta a Exploración contiene prácticamente la misma información que una trama *Beacon*.

- Trama Autenticación:

  La autenticación es el proceso mediante el cual el AP verifica la identidad de la estación cliente que está intentando conectarse. La estación cliente comienza éste proceso enviando una Trama Solicitud de Autenticación al AP y éste, puede rechazarla o aceptarla, y le notifica a la estación mediante la trama Respuesta a Autenticación.

- Trama Solicitud de Asociación y Trama Respuesta a Asociación:

  Una vez la estación cliente haya sido autenticada por el AP, está en capacidad de asociarse con el mismo. Cuando una estación cliente se asocia a un AP, se hace miembro de un BSS y puede enviar información a través del AP y a través del sistema de distribución. La estación cliente inicia el proceso con el envío de la trama Solicitud de Asociación y el AP le responde con la Trama Respuesta a Asociación [1].

A continuación se explica el modo de operación de las tramas de control RTS/CTS y se detalla el formato y funcionamiento de las tramas ACK debido a que son fundamentales en el desarrollo de este trabajo de grado.

- Trama Petición para Transmitir (RTS, *Request to Send*) y Trama Despejado para Transmitir (CTS, *Clear to Send*):

  Una estación hace parte de un BSS, únicamente cuando está en capacidad de comunicarse con el AP. Sin embargo, es posible que la estación no sea capaz de escuchar otras estaciones cliente ni de ser escuchada por las mismas. Esto puede representar un problema si se tiene en cuenta que las estaciones realizan la detección virtual de portadora escuchando la transmisión de otra estación y realizan la detección física de portadora escuchando el medio. Así que si una estación no puede escuchar a las demás estaciones ni ser escuchada, hay una alta posibilidad de que ocurra una colisión.

  Las tramas Petición para Transmitir y Despejado para Transmitir hacen parte de un mecanismo que utiliza la detección virtual de portadora para evitar dichas colisiones. El mecanismo consiste en reservar el medio antes de transmitir una trama de datos. Cuando una estación desea transmitir, primero envía una trama RTS a la estación receptora (la estación receptora puede ser el AP o una estación cliente), informándole a las demás estaciones que el medio va a estar ocupado por un tiempo y que deben esperar hasta que la trama CTS, la trama de Datos y el ACK sean enviados. Después, la estación receptora (que puede ser el AP o una estación cliente) responde con la trama CTS e informa de nuevo a las demás estaciones que deben esperar hasta que la trama de Datos y el ACK sean enviados. Debido a que el AP es el encargado de enviar alguna de las dos tramas, todas las estaciones cliente del BSS deberían escuchar por lo menos la trama que fue transmitida por el AP. La Figura 1.8 [8] muestra el envío de información entre dos PCs inalámbricos en un BSS usando las tramas RTS y CTS.

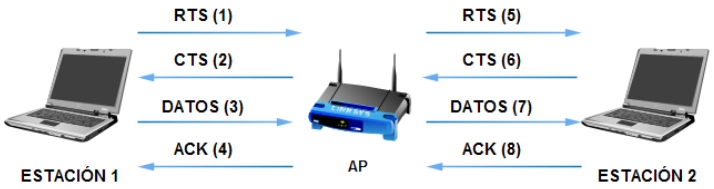


Figura 1.8. Uso de tramas RTS y CTS.

En un BSS el envío de información entre dos estaciones cliente implica la participación de un AP[17], es por esto que la Figura1.8 muestra realmente dos intercambios de información. El primero es entre la Estación 1 y el AP y el segundo es entre el AP y la Estación 2. Durante el primer intercambio, la estación 2 actualiza su temporizador NAV cuando escucha la trama RTS (1) o la trama CTS (2). Dependiendo que tan lejos esté la Estación 2 de la Estación 1 es posible que escuche o no la trama RTS (1), sin embargo, debido a que la Estación 2 hace parte de un BSS es evidente que debe escuchar la trama CTS (2). Una vez la trama ACK (4) haya sido enviada, las estaciones están en libertad de competir por el medio para luego transmitir. Si el AP es el que gana el acceso al medio comenzará el segundo intercambio de tramas RTS/CTS. La estación 1 escucha el RTS (5) que envía el AP, establece el valor de su temporizador NAV, y espera hasta que el ACK (8) haya sido enviado. Después de esto todas las estaciones tienen el derecho a transmitir.

Cuando una estación va a transmitir, primero realiza el proceso DCF (ver 1.1.2.14) y luego envía la trama RTS. Sin embargo, cuando la estación receptora va a transmitir la trama CTS, sólo tiene que esperar un tiempo SIFS. De igual forma, La transmisión de la trama de datos y del ACK no requieren del proceso DCF, las estaciones únicamente se ven obligadas a esperar un tiempo SIFS.

El uso de RTS/CTS solo es aconsejado cuando se quiere reducir la probabilidad de que dos estaciones colisionen debido a que una no puede escuchar a la otra, en el caso de que se verifique que esto no ocurre, es preferible no usarlo, pues el tráfico de paquetes RTS/CTS disminuye el desempeño de la red [8].

- Trama de Reconocimiento (ACK, *Acknowledgment*):

Los reconocimientos son un componente clave del método de control de acceso al medio en 802.11. Debido a que el medio es inalámbrico, la única forma de que una estación sepa que una trama transmitida fue recibida correctamente es que la estación receptora se lo notifique. Esta notificación es realizada usando una trama ACK. La Figura 1.9 [8] muestra el formato de dicha trama, la cual consta de 14 octetos de información.

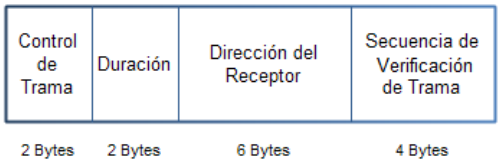


Figura 1.9. Formato de la Trama ACK.

Cuando la estación receptora recibe una trama de datos, copia de ésta la dirección MAC de la estación transmisora y la pone en el campo Dirección del Receptor, para luego enviar la trama ACK.

[17] En un IBSS los mensajes RTS/CTS son intercambiados directamente entre las estaciones, es decir, sin la participación de un AP.

La transmisión de una trama de reconocimiento tiene un trato especial y por lo tanto no se realiza el proceso Función de Coordinación Distribuida (DCF). Cuando la estación receptora va a enviar el ACK no necesita competir por el medio, únicamente debe esperar un tiempo SIFS e inmediatamente después, puede transmitir la trama. Cabe resaltar que éste intervalo de tiempo es más corto que el intervalo DIFS por dos slots, situación que favorece a la estación receptora pues le garantiza acceder al medio antes de que otra estación lo haga [8].

1.1.3 Nivel Físico del Estándar IEEE 802.11

El nivel físico es la interfaz entre el nivel MAC y el Canal. Se subdivide en dos niveles:

- Nivel Procedimiento de Convergencia del nivel Físico (PLCP, *Physical Layer Convergence Procedure*):

  Su función principal es mapear la PDU[18] del nivel MAC o MPDU a un formato de trama susceptible de ser transmitido o recibido por las estaciones a través del nivel PMD. Este proceso lo realiza agregando campos a la MPDU, generando así la PDU de PLCP o PPDU. También lleva a cabo CCA mediante el cual detecta si el canal está siendo usado o no.

- Nivel Dependiente del Medio Físico (PMD, *Physical Medium Dependent*):

  Este nivel se encarga de la transmisión y recepción de señales, para lo cual usa las técnicas de modulación y codificación, determina el canal que se va a usar y realiza la conversión de señales a secuencias binarias.

El estándar IEEE 802.11 define varias técnicas de transmisión que se usan con cada uno de los estándares. En la Tabla 1.3 [8] se muestra la relación de estas técnicas con la familia de protocolos 802.11 y las velocidades de transmisión de datos que pueden alcanzar.

Tabla 1.3. Técnicas de transmisión en 802.11.

Protocolo	Frecuencia (GHz)	Velocidad (Mbps)	Técnica de Transmisión
802.11	2.4	1, 2	FHSS, DSSS
802.11b	2.4	1, 2, 5.5, 11	CCK, DSSS
802.11a	5.8	6, 9, 12, 18, 24, 36, 48, 54	OFDM
802.11g	2.4	6, 9, 12, 18, 24, 36, 48, 54	OFDM, DSSS, CCK
802.11n	2.4, 5.8	78, 104, 108, 117, 120, 130, 145, 162, 180, 216, 240, 243, 270, 300	OFDM, DSSS

A continuación se explica cómo operan algunas de las técnicas de transmisión de nivel físico del estándar IEEE 802.11 [11]:

- FHSS:
  La técnica FHSS consiste en transmitir la señal en una portadora que "salta" de frecuencia en frecuencia, en función del tiempo. El cambio periódico de frecuencia de la portadora reduce la interferencia producida por otra señal originada en un sistema de banda estrecha, afectando solo si ambas señales se transmiten en la misma frecuencia en el mismo momento. Un patrón de saltos determina las frecuencias de portadora en cada momento. Para recibir correctamente la señal, el receptor debe conocer el

[18] Unidad de Datos del Protocolo (PDU,*Protocolo Data Unit*).

patrón de saltos del emisor, sincronizarse con éste y sintonizar la frecuencia correcta en el momento correcto.

- DSSS:
  La técnica de transmisión DSSS multiplica la señal a transmitir por una señal de "ruido", la cual es una pseudo-secuencia de 1s y -1s a una frecuencia mucho mayor que la señal original y se le conoce como Secuencia de Barker, Código de Dispersión o Ruido Pseudo-aleatorio. La señal resultante es una expansión en frecuencia de la señal original.

  La señal se recupera en el receptor multiplicando la señal recibida por la misma secuencia de 1s y -1s para dar como resultado la señal original.

- CCK:
  La técnica de transmisión conocida como Cifrado de Código Complementario (CCK, *Complementary Code Keying)* es un conjunto de 64 palabras de 8 bits que se usan para codificar datos a velocidades de transmisión de 5.5 Mbps y 11 Mbps. Estas palabras tienen propiedades matemáticas únicas que les permiten ser correctamente distinguidas de otras por el receptor aún en presencia de ruido e interferencia multitrayecto.

  CCK trabaja en conjunto con la tecnología DSSS, aplicando operaciones matemáticas a dichos códigos. Esto permite que los códigos representen un mayor volumen de información por ciclo de reloj. El transmisor puede entonces enviar múltiples bits de información con cada código DSSS, suficiente para alcanzar una velocidad de 5.5 y 11 Mbps en lugar de los 2 Mbps del estándar original.

- OFDM:
  La tecnología de Multiplexación por División de Frecuencias Ortogonales (OFDM) consiste en fraccionar un canal de gran ancho de banda en un número de subportadoras ortogonales, las cuales se emiten en paralelo para aumentar la velocidad de transmisión de datos, mitigando el gasto que implican las bandas de guarda utilizadas para evitar interferencia entre los canales FDM con la selección de canales que se solapan pero no se interfieren, es decir, que son ortogonales.

1.2 LIMITACIONES DEL ESTÁNDAR 802.11 EN LARGAS DISTANCIAS

El proyecto "*Análisis, Uso, Adecuación y Apropiación de Servicios sobre Tecnologías Inalámbricas en Zonas de Difícil Acceso de las Poblaciones Indígenas del Cauca Andino*" desarrollado por los grupos de investigación y desarrollo GNTT, GESC, IDIS y GESIG de la Universidad del Cauca y financiado por el Fondo Regional para la Innovación Digital en América Latina y el Caribe FRIDA propone como uno de sus objetivos brindar una conexión de banda ancha que intercomunique a las comunidades indígenas del Cauca. Pensando en alcanzar este propósito y conociendo que la distancia entre estas comunidades es superior a la distancia en la que operan tecnologías de bajo costo como WiFi, surge la necesidad de realizar un estudio con el cual no solo se pueda tener un canal de datos disponible manejando variables como potencia de transmisión y ganancia de antenas, sino que aspectos intrínsecos de la tecnología colaboren con el buen funcionamiento de una red de tal envergadura. Así, en esta sección del documento se muestran las limitantes que presenta esta tecnología en largas distancias y se da cumplimiento al Objetivo Específico número 1 "*Determinar las limitaciones de los mecanismos de reconocimiento del protocolo 802.11 en enlaces de larga distancia*" propuesto en el anteproyecto de este Trabajo de Grado.

1.2.1 Legislación

Cuando se quiere usar WiFi para enlaces de largas distancias se encuentran ciertos aspectos que impiden el despliegue de esta tecnología para interconectar lugares distantes. Primero que todo, el

estándar especifica las frecuencias de trabajo en 2.4 GHz y 5.8 GHz. Las leyes existentes permiten el uso de ese rango de frecuencias sin la necesidad de licencias de funcionamiento en la mayoría de los países, pero se encuentran restricciones en el nivel de potencia al cual se puede transmitir. Ese nivel máximo depende del país y del tipo de enlace. Para Colombia la normativa vigente para la banda de 2.4 GHz según la Resolución 689 del 21 de abril de 2004 del Ministerio de Comunicaciones establece: Los sistemas que operen en la banda de 2.400 a 2.483,5 MHz que sean utilizados exclusivamente para operaciones fijas punto a punto, pueden emplear antenas de transmisión con ganancia direccional mayor a 6 dBi siempre y cuando la máxima potencia pico de salida de un transmisor de 30dBm sea reducida en 1 dB por cada 3 dB que la ganancia direccional de la antena exceda los 6 dBi, además exige que para la correcta operación de los sistemas de acceso inalámbrico y redes inalámbricas de área local, que utilicen técnicas de transmisión de espectro ensanchado y modulación digital, de banda ancha y baja potencia, sólo se aceptarán equipos de conformidad con las normas técnicas de la Comisión Federal de Comunicaciones (FCC, Federal Communications Comission).

Con respecto a los enlaces punto a multipunto, la normatividad colombiana con la Ley 1689 de 2007 establece "*que en la actualidad existen internacionalmente múltiples y diversos aparatos y dispositivos inalámbricos de banda ancha y baja potencia que utilizan antenas omnidireccionales con potencias iguales o algo superiores a los 100 mW, que cumplen con las demás disposiciones de la Resolución 689 de 2004*".

Aún con estas restricciones en potencia la señal puede alcanzar distancias de varios kilómetros siempre y cuando se garantice línea de vista.

En IEEE 802.11 se definen diferentes velocidades de transmisión de datos, lo cual se consigue empleando diferentes tipos de modulación y codificación, la Tabla 1.4 [8], muestra cómo se establecen estas relaciones para 802.11a/g.

Tabla 1.4. Modos de operación del Nivel Físico de 802.11a/g.

Modo	Modulación	Tasa de Codificación	Tasa de Datos	Sensitividad Mínima del Receptor
1	DBPSK	1/2	6 Mbps	-82 dBm
2	DBPSK	3/4	9 Mbps	-81 dBm
3	DQPSK	1/2	12 Mbps	-79 dBm
4	DQPSK	3/4	18 Mbps	-77 dBm
5	16QAM	1/2	24 Mbps	-74 dBm
6	16QAM	3/4	36 Mbps	-70 dBm
7	64QAM	2/3	48 Mbps	-66 dBm
8	64QAM	3/4	54 Mbps	-65 dBm

A estas diferentes combinaciones les corresponde diferentes Niveles Mínimos de Recepción. Esto indica que cuanto mayor sea la velocidad de transmisión de datos, más potencia debe ser recibida para tener una señal lo suficientemente alta para mantener el enlace a dicha velocidad. La diferencia en la sensibilidad del receptor para velocidades de transmisión de 1 y 11 Mbps, aunque depende de los equipos suele ser de más de 10dB, lo cual equivale casi a cuadriplicar con 1 Mbps el alcance que se tiene con 11 Mbps.

Basándose en lo anterior y teniendo en cuenta la restricción que impone la Resolución 689 sobre la potencia de emisión, se puede notar que para enlaces de larga distancia es necesario limitar la velocidad a niveles bajos, de manera que se pueda garantizar un enlace estable. Otras consideraciones sobre la factibilidad o la conveniencia de hacer enlaces de uno u otro tipo dependerán del estudio del nivel MAC.

1.2.2 Pérdidas de Propagación

Las pérdidas de propagación en espacio libre se calculan teniendo en cuenta solo la atenuación de la potencia de las ondas electromagnéticas según recorren el trayecto desde el transmisor hasta el receptor, sin que se tome en consideración ninguna otra componente que la propagación directa. No se toman en cuenta otras fuentes de pérdidas tales como reflexiones, difracciones, etc. Las Pérdidas de Espacio Libre PEL son proporcionales al cuadrado de la distancia entre el transmisor y el receptor, y también al cuadrado de la frecuencia de la señal [12].

Las pérdidas de propagación se expresan en decibelios dB y se calculan con la fórmula (1), donde *d* es la distancia y *f* es la frecuencia [12]:

$$PEL(dB) = 92.45 + 20\log_{10}(d[Km]) + 20\log_{10}(f[GHz]) \quad (1)$$

Para una misma distancia, el cambio en las pérdidas de propagación en espacio libre puede aumentar hasta en 0.3 dB de acuerdo al canal elegido en la banda de 2.4 GHz, por tanto se puede considerar la expresión (2) como una buena aproximación a las pérdidas:

$$FSL(dB) \cong 100 + 20\log_{10}(d[Km]) \quad (2)$$

Al calcular las pérdidas de propagación de esta manera se ignoran factores importantes como las irregularidades del terreno, la curvatura de la tierra y las características específicas del lugar. Por eso, si es necesario conocer de una manera más acertada las pérdidas de propagación de acuerdo al contexto topográfico, se puede considerar usar el modelo de Longley-Rice con el apoyo de mapas digitales de elevaciones con la mayor precisión posible, pues éste considera parámetros propios del terreno sobre el cual se realiza el radio enlace, como son la transimpedancia de la superficie del suelo, la mínima refractividad superficial mensual media, entre otros [12].

1.2.3 Sensibilidad de Receptores y Margen de señal

El estándar define la sensibilidad como el nivel de señal recibida suficiente para mantener la tasa de error de bit BER < 10^{-5}. La sensibilidad de los receptores varía de acuerdo a la velocidad de transmisión de datos del enlace y a la calidad del receptor. Como es de esperar, existe una variación entre el balance de enlace teórico y el real, esencialmente debido a:

- La propagación responde a un modelo real de terreno irregular y no a un modelo de espacio libre, aunque en unos casos este margen puede ser pequeño.
- La potencia real de los transmisores puede ser menor a la nominal.
- La alineación de las antenas puede no ser la mejor y puede ir deteriorándose por efecto del viento y otros fenómenos meteorológicos.
- Aunque sea un enlace fijo, con línea de vista y mucha directividad, los enlaces presentan un comportamiento no determinístico, y se ven afectados por un desvanecimiento que se puede modelar estadísticamente.

Las consideraciones anteriores hacen necesario que se trabaje con un margen de señal por encima a la sensibilidad del receptor, de manera que el enlace sea más confiable. Este margen puede ser definido por el diseñador, de acuerdo a los aspectos más relevantes que pueden afectar el enlace.

De esta manera, la potencia recibida se puede calcular con el balance de enlace:

$$P_{Rx} = P_{Tx} - L_{Tx} + G_{Tx} - L_p + G_{Rx} - L_{Rx} > S_{Rx} + M \quad (3)$$

Siendo P_{Rx} la potencia recibida por el receptor, P_{Tx} la potencia transmitida por el transmisor, G_{Tx} la ganancia de la antena del transmisor, L_{Tx} y L_{Rx} las pérdidas generadas por el cable y los conectores en el sistema transmisor y receptor respectivamente, G_{Rx} la ganancia de la antena del receptor, S_{Rx} la sensibilidad de recepción y M el margen mínimo por encima de la sensibilidad que se debe tener en cuenta para asegurar la confiabilidad y continuidad del enlace.

1.2.4 Constantes MAC 802.11

El estándar IEEE 802.11 fue diseñado para funcionar en ambientes con varios usuarios que compiten por el acceso al canal. Los radios inalámbricos de esos hosts son *half-duplex* y no pueden escuchar al medio y transmitir al mismo tiempo, es por esto que solo una estación puede transmitir correctamente en un instante de tiempo [13]. Por esta razón, para todos los dispositivos 802.11 a/b/g se definen un número de constantes con el fin de detectar la portadora en el aire y evitar colisiones como también para retransmitir las tramas no reconocidas. Las constantes que se consideran son `Slottime`, `CTSTimeout` y `ACKTimeout`. La constante `Slottime` es más importante para evitar colisiones cuando hay múltiples estaciones conectadas o cuando se trata de crear o simular una comunicación *Full-Duplex*, mientras que los `ACKTimeout` son más importantes para comunicaciones punto a punto. La Figura 1.10 [**Error! Bookmark not defined.**] presenta otras constantes que pueden limitar la distancia máxima del enlace como lo son SIFS, DIFS y PIFS, cuya descripción se encuentra en la sección 1.1.2.8.

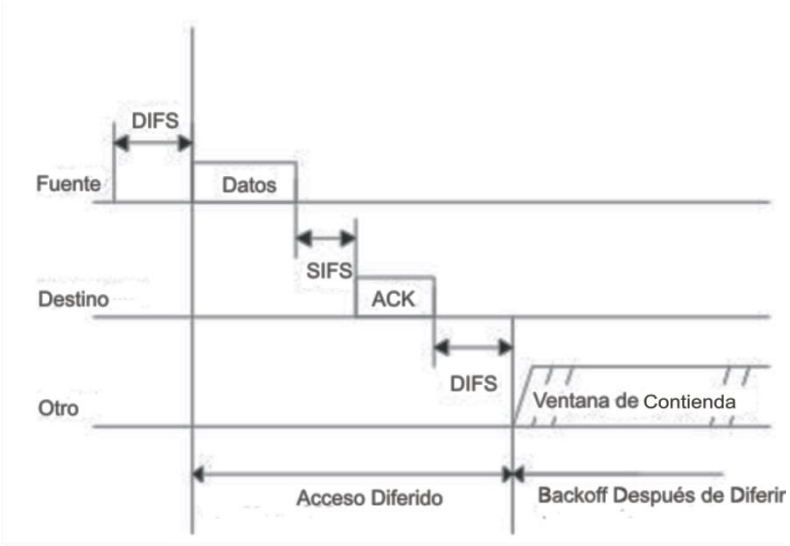


Figura 1.10. Diagrama de Contienda.

ACK, Mecanismo de Recuperación del Nivel de Enlace

El nivel MAC de 802.11 usa un protocolo simple de '*stop and wait*', donde cada trama es reconocida independientemente. Al recibir exitosamente un paquete, el nodo receptor debe enviar un reconocimiento dentro de un periodo de tiempo `ACKTimeout`, de lo contrario el transmisor tendrá que reenviar el paquete. Este mecanismo tiene algunas desventajas:

- A medida que la distancia aumenta, el retardo de propagación también lo hace, y el transmisor deberá esperar un mayor período de tiempo para que el ACK llegue. Esto disminuye la utilización del canal.
- Si el ACK toma más tiempo en llegar que el `ACKTimeout`, el transmisor tendrá que retransmitir innecesariamente y desperdiciar ancho de banda.

El problema radica en que el estándar IEEE define el tiempo de propagación en el aire como 1 µs (±10% SIFS), en este tiempo las ondas de radio aproximadamente viajan 300 metros. Con referencia a la figura 1.10, si el `ACKTimeout` es menor que el tiempo que le toma a la parte final del último paquete de datos en propagarse al receptor junto con un periodo SIFS y además junto con el tiempo que le toma al inicio del ACK de ese paquete en regresar al transmisor, el nivel MAC del transmisor asumirá que el paquete se ha perdido y retransmitirá el paquete de datos innecesariamente. Este paquete terminará colisionando con el ACK que va en camino, induciendo el mecanismo de *Backoff* y reduciendo de esta manera el *throughput* del enlace. De otro lado, si el tiempo de espera para el `ACKTimeout` es muy largo, el transmisor esperará mucho tiempo innecesariamente antes de retransmitir alguna trama perdida, reduciendo así también el *throughput* del enlace [7].

Al considerar el `CTStimeout`, se puede sacar conclusiones similares, aunque RTS/CTS es solo para reducir las colisiones con nodos escondidos y puede ser desactivado en la mayoría de los casos.

En la sección 1.3 se encuentra en profundidad el funcionamiento de los diversos métodos de ACK presentes en 802.11.

1.2.5 Colisiones en Enlaces de Larga Distancia

Dado que el aire es un medio compartido, las redes WLAN tienen que considerar la posibilidad de que se presenten colisiones al igual que en las tradicionales redes cableadas. La diferencia es que, no hay ningún medio por el cual la estación transmisora pueda determinar que en realidad hubo una colisión. Es imposible detectar una colisión en una red WLAN. Por esta razón, las redes WLAN utilizan el método de Acceso Múltiple por Detección de Portadora con Evasión de Colisiones [7] como se describió en la sección 1.1.3.

CSMA/CA junto con los datos de control que son usados en una WLAN, conllevan a un sobre encabezado que usa aproximadamente el 50% del ancho de banda disponible en una WLAN. Este sobre encabezado junto con los campos utilizados por RTS/CTS que mejoran la evasión de colisiones, son responsables de que el *throughput* de una red 802.11 sea aproximadamente solo la mitad de la velocidad de transmisión de datos a nivel Físico.

1.2.6 Interferencia entre Enlaces Adyacentes

Otra fuente importante de errores es la interferencia originada por la operación de enlaces adyacentes en el mismo canal o en canales sobrepuestos. Aunque la interferencia entre enlaces adyacentes se puede evitar usando canales no adyacentes, la operación de WiFi en una banda no licenciada y con un número creciente de usuarios, dificulta el correcto funcionamiento de la red.

Es importante conocer que el estándar IEEE802.11 divide la banda de 2.4 GHZ en 14 canales separados como se muestra en la Tabla 1.5 [8]. Aunque existen los 14 canales, la regulación propia del territorio determina que canales pueden ser usados. La Tabla 1.5 muestra también qué canales pueden ser usados en ciertas regiones del mundo.

Tabla 1.5. Distribución Mundial de Frecuencias para 802.11 b/g.

Canal	Frecuencia Central (GHz)	E.E.U.U.	COLOMBIA	EUROPA	JAPÓN	ESPAÑA	FRANCIA
1	2.412	X	X	X	X		
2	2.417	X	X	X	X		
3	2.421	X	X	X	X		
4	2.427	X	X	X	X		
5	2.432	X	X	X	X		
6	2.437	X	X	X	X		
7	2.442	X	X	X	X		
8	2.447	X	X	X	X		
9	2.452	X	X	X	X		
10	2.457	X	X	X	X	X	X
11	2.462	X	X	X	X	X	X
12	2.467			X	X		X
13	2.472			X	X		X
14	2.484						

Los canales de 802.11 b/g son de 22 MHz de ancho y se designan por su frecuencia central. El problema de traslape se presenta dado que la separación entre canales es solo de 5 MHz. La Figura 1.11 [8] muestra cómo se sobreponen los canales adyacentes y que algunos como los canales 1, 6 y 11 están plenamente separados.

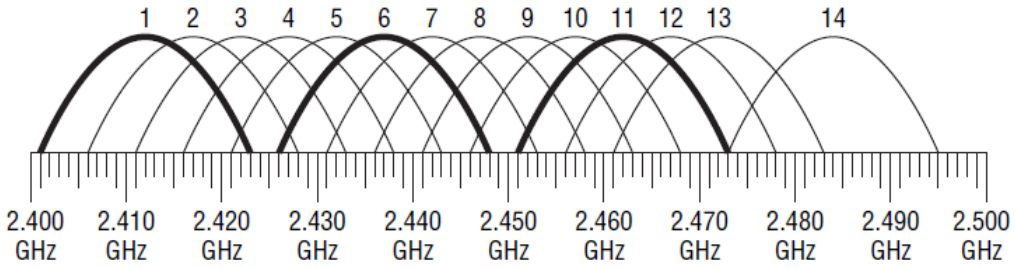


Figura 1.11. Diagrama de Superposición de Canales en IEEE 802.11 b/g.

Además de la frecuencia portadora o lóbulo principal, también se generan bandas laterales, como se muestra en la Figura 1.12 [8]. La IEEE especifica que el primer lóbulo lateral debe estar ubicado entre ±11 MHz y ±22 MHz de la frecuencia central y debe ser de por lo menos 30 dB más bajo que el lóbulo principal. También especifica que cualquier lóbulo lateral adicional más allá de ± 22 MHz de la frecuencia central debe estar al menos 50 dB por debajo del nivel más alto del lóbulo principal.

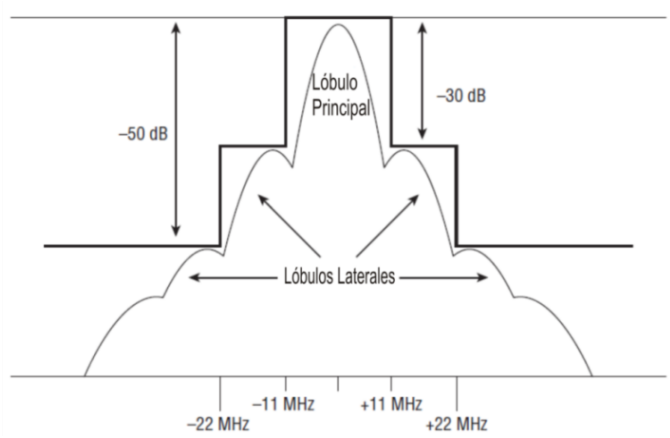


Figura 1.12. Máscara de Espectro Transmitida en IEEE 802.11b.

Debido a la existencia de los lóbulos laterales, los canales 1, 6 y 11 se alcanzan a superponer como se observa en la Figura 1.13 [8]. El problema de interferencia alcanza a afectar a los canales que se presume están separados, aún más a aquellos adyacentes.

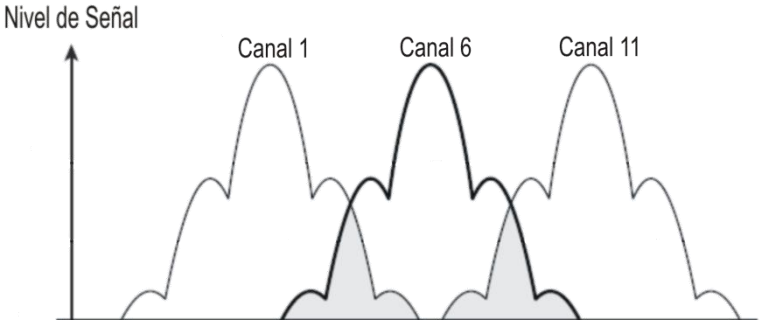


Figura 1.13. Superposición de Armónicos en los Canales 1, 6 y 11.

En un entorno de operación de varios kilómetros en el cual pueden existir muchas redes inalámbricas es necesario verificar que la interferencia causada por cualquier señal existente en uno de los 11 canales no tiene la potencia suficiente para causar un decremento significativo en el desempeño de la red.

1.3 MÉTODOS DE RECONOCIMIENTO

Los mensajes de reconocimiento hacen parte de los seis mensajes de control descritos en 802.11 y es uno de los componentes claves del funcionamiento de CSMA/CA. En este sentido y dado que 802.11 funciona en un medio inalámbrico que no puede garantizar la transmisión exitosa de los datos, existe una manera de conocer si una trama transmitida fue apropiadamente recibida, esto es, notificando a la estación transmisora este hecho. Dada la importancia de estos mensajes en el buen funcionamiento de un enlace inalámbrico basado en 802.11 y que son la base de estudio de esta investigación, se describen a continuación los diversos métodos de reconocimiento contemplados en el estándar.

1.3.1 ACK Estándar

La recepción de algunos tipos de tramas, requieren que la estación receptora responda con un reconocimiento, generalmente una trama ACK. Este proceso se conoce como reconocimiento positivo y se lleva a cabo si la FCS de la trama recibida es correcta.

La no recepción de una trama ACK que se espera le indica a la estación que inició el intercambio de tramas que un error ha ocurrido. Puede suceder, que la estación destino haya recibido la trama correctamente, y que el error haya ocurrido en la transferencia o en la recepción de la trama ACK. Para quien inició el intercambio de tramas esta condición es indiferente de un error que ocurrió en la trama inicial.

1.3.1.1 **Procedimiento ACK**

Después de transmitir una Unidad de Datos de Protocolo MAC MPDU que requiere una trama ACK como respuesta, la estación deberá esperar un intervalo de tiempo `ACKTimeout`, con un valor `aSIFSTime`[19] + `aSlotTime`[20] + `aPHY-RX-START-Delay`[21], iniciando el temporizador al confirmar el fin de la transmisión por parte del nivel físico. Si el nivel físico no indica que ha empezado a recibir una MPDU durante el intervalo `ACKTimeout`, la estación concluye que la transmisión de la trama de datos ha fallado, y deberá invocar su procedimiento de *Backoff* hasta la finalización del `ACKTimeout`. Por el contrario, si la indicación ocurre durante el intervalo `ACKTimeout`, la estación deberá esperar la correspondiente indicación de fin de la recepción por parte del nivel físico para determinar si la transmisión de la MPDU fue exitosa. El recibimiento de una trama ACK válida enviada por el receptor de una MPDU que requiere reconocimiento, deberá ser interpretado como un reconocimiento exitoso, permitiendo que la secuencia de tramas continúe, o termine sin reintentos, como es apropiado para el progreso de una secuencia de tramas en particular. El reconocimiento de algo más, incluyendo cualquier otro tipo válido de tramas deberá ser interpretado como una falla en la transmisión de la MPDU. En esta instancia, la estación deberá invocar su procedimiento de *Backoff* y puede procesar la trama recibida. La única excepción es que el reconocimiento de una trama de datos válida enviada por el receptor de una trama PS-Poll deberá ser también aceptada como un reconocimiento exitoso de una trama PS-Poll [14].

Una trama ACK con las características descritas en 1.1.3.9 debe ser generada en las siguientes situaciones:

- Un AP debe generar una trama ACK tras la recepción exitosa de una trama que requiere reconocimiento con el campo a *Sistema de Distribución* (To DS) habilitado.
- Una estación debe transmitir una trama ACK únicamente cuando recibe exitosamente una trama *unicast* del tipo que requiere reconocimiento, pero no lo debe hacer tras recibir una trama del tipo *broadcast* o *multicast*.
- Tras la recepción exitosa de una trama que requiere reconocimiento, debe iniciar la transmisión de la trama ACK después de un periodo SIFS, sin importar el estado ocupado/libre del medio como se aprecia en la Figura 1.10.

1.3.1.2 **Procedimientos de Recuperación y Límites de Retransmisión**

La recuperación de errores es un proceso que intenta transmitir de nuevo las tramas que en una secuencia de intercambio se infiera han fallado. Los reintentos deben continuar, para cada secuencia de intercambio de tramas fallida, hasta que la transmisión sea exitosa, o hasta que el límite relevante de intentos sea alcanzado. Las estaciones deben mantener un Contador Corto de Reintentos (SRC, *Short Retry Count*) y un Contador Largo de Reintentos (LRC, *Long Retry Count*) para cada Unidad de Datos de Servicio MAC (MSDU[22], *MAC Service Data Unit*) o Unidad de Datos del Protocolo de Gestión MAC

---

[19] `aSIFSTime`: Tiempo nominal en microsegundos que el nivel físico y el nivel MAC requieren para recibir el último símbolo de una trama, procesar la trama, y responder con el primer símbolo de la trama de respuesta lo más pronto posible. `aSIFSTime` hace referencia al concepto SIFS de la sección 1.1.3.2.

[20] `aSlotTime`: duración del slot de tiempo que el nivel MAC usará para definir los periodos PIFS y DIFS. `aSlotTime` hace referencia al concepto slot de la sección 1.1.3.3.

[21] `aPHY-RX-START-Delay`: Retardo en microsegundos desde que inicia el preámbulo hasta que la estación cambia a modo de recepción.

[22] MSDU: Información que es entregada como una unidad entre Puntos de Acceso al Servicio MAC (SAPs). Varias MPDUs conforman una MSDU.

(MMPDU[23], *MAC Management Protocol Data Unit*) que espera ser transmitida. Estos contadores se incrementan y se reinician independientemente uno del otro.

La recuperación de errores siempre es responsabilidad de la estación que inicia la secuencia de intercambio de tramas. Muchas circunstancias pueden causar que un error que sucede requiera recuperación. Por ejemplo, una trama CTS puede no retornar después que una trama RTS es transmitida. Esto puede ocurrir debido a una colisión con otra transmisión, debido a interferencias en el canal durante la trama RTS o CTS, o debido a que la estación que recibe la trama RTS tiene activa la condición de Escucha de Portadora Virtual (Virtual CS, *Virtual Carrier Sense*) indicando un periodo de tiempo de medio ocupado.

Después de que una trama RTS ha sido transmitida, la estación realiza el procedimiento CTS. Si la transmisión RTS falla, el SRC y el Contador Corto de Reintentos de la Estación (SSRC, *Station Short Retry Count*) para la MSDU o para la MMPDU aumentan. Este proceso debe continuar hasta que el número de intentos de transmitir esa MSDU o MMPDU alcance el Límite Corto de Reintentos (dot11ShortRetryLimit[24]), de manera una trama RTS se reconoce exitosamente con una trama CTS que actúa como un ACK.

Cuando se transmite una trama que requiere reconocimiento, la estación debe realizar el procedimiento ACK, como se especifica en 1.3.1.1. El SRC para una MPDU de datos o el SSRC de una MMPDU se incrementará cada vez que se transmita una trama MAC de menor o igual longitud que el Umbral RTS (`dot11RTSThreshold`[25]) y falle para esa MPDU o MMPDU. Este SRC y SSRC debe reiniciarse cuando una trama MAC de longitud menor o igual que el `dot11RTSThreshold` sea exitosa para esa MPDU o MMPDU. El LRC para una MPDU o el SLRC de una MMPDU se incrementará cada vez que se transmita una trama MAC de longitud mayor que una `dot11RTSThreshold` y falle para esa MPDU o MMPDU. Este LRC y el Contador Largo de Reintentos de la Estación (SLRC, *Station Long Retry Count*) deben reiniciarse cuando una trama MAC de mayor longitud que un `dot11RTSThreshold` sea exitoso. Todo intento de retransmisión de una MPDU de datos o una MMPDU en que falle el procedimiento de ACK una o más veces deberá realizarse con el campo Reintento fijado en 1.

Los reintentos para transmisiones fallidas deberán continuar hasta que el SRC para las MPDU o MMPDU sea igual a dot11ShortRetryLimit o hasta que el LRC para las MPDU o MMPDU sea igual a `dot11LongRetryLimit`[26]. Cuando cualquiera de estos dos límites sea alcanzado, los intentos por retransmitir deben detenerse, y la MPDU o MMPDU deben descartarse.

Otro caso se presenta cuando una estación en modo Ahorro de Energía PS, en un ESS, inicia una secuencia de intercambio de tramas transmitiendo una trama PS-Poll para solicitar los datos del AP. En el evento en que ni la trama ACK ni una trama de datos sean recibidas desde el AP en respuesta a la trama PS-Poll, la estación deberá retransmitir la secuencia, transmitiendo otra trama PS-Poll a su conveniencia. Si el AP envía una trama de datos en respuesta a la trama PS-Poll, pero falla en recibir la trama ACK en reconocimiento a su trama de datos, la siguiente trama PS-Poll de la misma estación puede generar una retransmisión de la última MSDU. Esta MSDU duplicada puede ser filtrada en la estación receptora usando el mecanismo normal de filtrado de tramas duplicadas. Si el AP responde a una PS-Poll transmitiendo una trama ACK, entonces la responsabilidad sobre la recuperación de errores de la entrega de la trama de datos se asigna al AP, debido a que los datos son transferidos en una secuencia de intercambio de tramas subsecuente, la cual es iniciada por el AP. El AP deberá intentar entregar una MSDU a la estación que transmitió la PS-Poll, usando alguna secuencia válida de

[23] MMPDU: Unidad de datos intercambiada entre dos entidades MAC, usando servicios del nivel físico, para implementar el protocolo de gestión/control MAC. Varias MPDUs conforman una MMPDU.

[24] `dot11ShortRetryLimit`: Límite de reintentos de transmisión usado con tramas cortas, es decir cuya longitud es menor que `dot11RTSThreshold`.

[25] `dot11RTSThreshold`: Atributo que controla el mecanismo RTS/CTS permitiendo que las estaciones se configuren en este modo siempre, nunca o solo con tramas de longitud mayor que una especificada.

[26] `dot11LongRetryLimit`: Límite de reintentos de transmisión usado con tramas largas, es decir cuya longitud es igual o mayor que `dot11RTSThreshold`.

intercambio de tramas para una MSDU direccionada individualmente. Si la estación PS que transmitió la PS-Poll retorna al estado Doze[27] después de transmitir la trama ACK en respuesta a la recepción exitosa de su MSDU, pero el AP falla al recibir esta trama ACK, el AP retransmitirá su MSDU hasta que el limite relevante de reintentos sea alcanzado.

1.3.2 ACK en Bloque

El mecanismo de ACK en Bloque o `Block ACK`[28] descrito en [15] y en esta sección, hace referencia al especificado por el estándar 802.11e que funciona según la prioridad establecida por el tipo de datos que se transmiten. Es importante conocer que al incluir las capacidades de QoS, se provee diferente trato a aplicaciones, usuarios, o tipo de datos para tratar de garantizar cierto nivel de desempeño en el flujo de la información. Para llevar a cabo esto, se define la Función de Coordinación Híbrida (HCF, *Hybrid Coordination Function*) que debe ser implementada en todas las estaciones QoS. Ésta combina funciones de DCF y PCF con otras propias, mecanismos específicos de QoS y subtipos de tramas que permiten que un componente llamado Coordinador Híbrido (HC, *Hybrid Coordinator*) maneje eficientemente las secuencias de intercambio de tramas QoS. HCF usa una técnica de acceso al canal basado en contienda, llamado Acceso al Canal Distribuido Mejorado (EDCA, *Enhanced Distributed Channel Access*) y un acceso controlado al canal para transferencias de datos sin necesidad de competir por el medio, llamado Acceso Controlado al Canal HCF (HCCA, *HCF Controlled Channel Access*).

El mecanismo `Block ACK` mejora la eficiencia del canal agregando varios reconocimientos dentro de una trama. Hay dos tipos de mecanismos de `Block ACK`: Inmediato y Retardado. El mecanismo `Block ACK` Inmediato es apropiado para tráfico de gran ancho de banda y baja latencia mientras que el mecanismo `Block ACK` Retardado es apropiado para aplicaciones que soporten una latencia moderada.

Para este mecanismo se define al transmisor que usa el mecanismo `Block ACK` como el Originador, y al receptor como Receptor.

El mecanismo `Block ACK` se inicia con el intercambio de tramas Petición/Respuesta ADDBA (*Add Block Acknowledgement*). Después de la inicialización, los bloques de tramas de datos QoS se pueden transmitir desde el Originador al Receptor. Un bloque se puede iniciar ganando una Oportunidad para Transmitir (TXOP, *Transmission Option*) o ganando la contención EDCA teniendo en cuenta que el número de tramas en el bloque está limitado al igual que la cantidad de memoria *Buffer* que se debe reservar por el Receptor. La MPDU con el bloque de tramas es reconocida con una trama de control BlockAck, que debe ser solicitada con una trama BlockAckReq.

El mecanismo de `Block ACK` no requiere el establecimiento de un Flujo de Tráfico (TS[29], *Traffic Stream*); de cualquier manera, una estación QoS que use la facilidad TS puede escoger señalizar su intención de usar el mecanismo `Block ACK` para la consideración del planificador en la asignación de TXOPs. Los reconocimientos de tramas pertenecientes al mismo Identificador de Tráfico (TID[30], *Traffic Identifier*), pero transmitidas durante múltiples TXOPs, pueden combinarse en una sola trama Block ACK dando la oportunidad al Originador de tener flexibilidad con respecto a la transmisión de datos MPDU. Éste puede dividir el bloque de tramas a través de TXOPs, separar la transferencia de datos y el intercambio de Block ACK, e intercalar bloques de MSDU para diferentes TIDs o Receptores.

La Figura 1.14 [14] ilustra la secuencia de mensajes para el establecimiento, transferencia de datos y Block ACK, y la finalización del mecanismo.

---

[27] Doze: Estado en que se encuentra una estación que solicita PS y no es capaz de transmitir o recibir y consumir baja potencia.

[28] Es más usual encontrar referencias al método de Reconocimiento en Bloque como `Block ACK`, por esta razón se tratará de ésta manera en el documento.

[29] Traffic Stream (TS): Conjunto de MSDUs que se deben entregar de acuerdo a los parámetros de QoS indicados en una especificación de tráfico TSPEC particular. Los TSs solo tienen significado para las entidades que soportan QoS en el servicio de datos MAC.

[30] Traffic Identifier (TID): Cualquier de los identificadores usados por las entidades de niveles más altos para distinguir las MSDUs de las entidades MAC que soportan Calidad de Servicio. El TID es asignado a un MSDU en los niveles superiores de MAC.

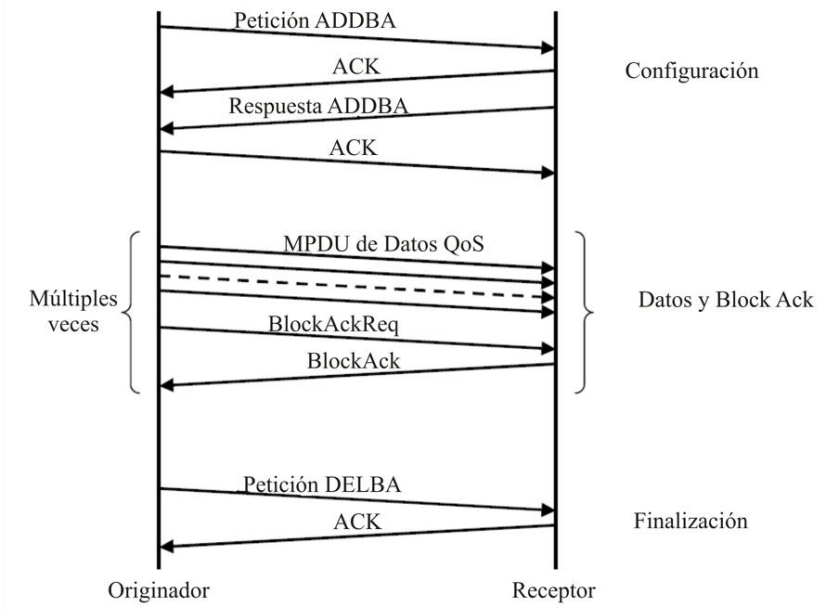


Figura 1.14. Secuencia de Mensajes para el Mecanismo `Block ACK`.

1.3.2.1 **Configuración y modificación de parámetros de Block ACK**

Una estación que usa el mecanismo `Block ACK` para la transmisión de tramas de datos QoS debe revisar si la estación receptora es capaz de participar en el mecanismo de `Block ACK`, descubriendo y examinando sus bits de capacidad de `Block ACK` Retardado o Inmediato. Si la estación es capaz de participar, el Originador envía una trama de Petición ADDBA indicando el TID para el cual el `Block ACK` se está configurando. Los campos Política de Block ACK (ver Tabla 1.6) y Tamaño del Buffer en la trama de Petición de ADDBA se sugieren y pueden ser modificadas por el Receptor. La estación receptora deberá responder con una trama de Respuesta de ADDBA. Esta estación, tiene la opción de aceptar o rechazar la petición hecha por el Originador. Si acepta, se crea un acuerdo de `Block ACK` entre los dos puntos interesados en la transmisión, indicando el tipo de `Block ACK` y el número de buffers que reservó para el bloque. Por otro lado, si la estación rechaza la petición, el Originador no podrá usar el mecanismo de `Block ACK`.

Si el mecanismo de `Block ACK` se configura para un TS, la negociación del ancho de banda debería preceder la configuración del mecanismo de `Block ACK`.

Una vez se haya configurado el intercambio de `Block ACK`, las tramas de datos y ACKs se transfieren usando el procedimiento descrito en 1.3.2.2.

1.3.2.2 **Transferencia de Datos y Reconocimientos**

Tras configurar el intercambio de bloques de tramas según el procedimiento en 1.3.2.1, el originador puede transmitir un bloque de tramas de datos QoS separadas por un periodo SIFS, sin exceder el número total de tramas acordadas en el campo tamaño de buffer de la trama de respuesta ADDBA. Cada una de las tramas debe tener el subcampo Política de ACK fijado en `Block ACK` dentro del campo Control de QoS. El campo dirección del receptor de la tramas debe ser la dirección *unicast* del destino. El Originador solicita el reconocimiento de tramas de datos QoS pendientes enviando una trama de petición BlockAckReq y debe mantener un registro de Block ACK para el bloque.

Sujeto a las limitaciones que establece una TXOP de acuerdo al mecanismo usado para acceder al canal, el originador puede:

- Separar las tramas del bloque y Block ACK en distintas TXOPs.
- Dividir una trama bloque en múltiples TXOPS.
- Dividir las transmisiones de MPDUs de datos enviadas bajo la política de `Block ACK` en múltiples TXOPs.
- Intercalar MPDUs con diferentes TIDs en la misma TXOP.
- Secuenciar o intercalar MPDUs para diferentes receptores en una TXOP.

Un mecanismo de protección (HCCA, RTS/CTS, o el mecanismo de protección para receptores sin Velocidad de transmisión Física Extendida (ERP, *Extended Rate* PHY)) debería ser usado para reducir la probabilidad que otras estaciones usen el medio durante la TXOP. Si no se usa un mecanismo de protección, entonces la primera trama que se envía como un bloque debe tener una trama de respuesta y deberá tener el campo Duración fijado para que las NAVs tengan valores apropiados en todas las estaciones en la BSS.

El originador deberá usar una secuencia de inicio de control de Block ACK para señalizar la primera MPDU en el bloque para la cual se espera un reconocimiento. Las MPDUs en el *buffer* del receptor con un valor de control de secuencia que precede el valor de la trama inicial, se llaman MPDUs *precedentes*. El receptor deberá reensamblar cualquier MSDU completa desde el *Buffer* indicando esto a su nivel superior y deberá entonces liberar los b*uffers* usados por las MPDUs precedentes.

El receptor debe mantener un registro de Block ACK con la siguiente información:

- Dirección del Originador.
- TID.
- Registro de *buffer* de reordenamiento indexado por el valor de la secuencia de control de la MPDU recibida. Este registro mantiene los estados de reconocimiento de las tramas de datos recibidas por el Originador.

Si la política de `Block ACK` inmediato se usa, el receptor deberá responder a una trama `BlockAckReq` con una trama `BlockAck`. Si el Receptor envía la trama `BlockAck`, el Originador actualiza sus registros y reenvía cualquier trama que no haya sido reconocida con la trama `BlockAck` o en algún otro bloque o individualmente.

Si se usa la política de `Block ACK` retardado, el receptor deberá responder a una trama de `BlockAckReq` con una trama ACK. Entonces, el receptor deberá enviar su respuesta Block ACK en una TXOP obtenida posteriormente. Una vez los contenidos de la trama `BlockAck` han sido preparados, el Receptor deberá enviar esta trama en la TXOP más rápida posible usando la prioridad más alta. El originador deberá responder con una trama ACK al recibir la trama `BlockAck`. Por otro lado si se usa la política de `Block ACK` retardado y si el Coordinador Híbrido (HC, *Hybrid Coordinator*) es el receptor, entonces, el HC puede responder con una trama `+CF-Ack`[31] si la trama `BlockAckReq` es la trama final de un intercambio de tramas para obtener una TXOP. Si se usa la política de `Block ACK` retardado y el HC es el originador, entonces el HC puede responder con una trama `+CF-Ack` si la trama `BlockAck` es la última trama del intercambio de tramas de las TXOP.

[31] La trama +CF-Ack es el mensaje de reconocimiento *Contention Free*-ACK o libre de contienda, que se envía al usar el método de acceso al medio HCCA.

La trama `BlockAck` contiene reconocimientos para las MPDUs de hasta 64 MSDUs previos. En la trama `BlockAck`, la estación reconoce solo las MPDUs comenzando con la secuencia inicial de control hasta la MPDU con el número de secuencia más alto que ha sido recibida, y la estación deberá fijar los bits en el mapa de bit de `Block ACK` correspondiente a todas las otras MPDUs a 0. El estado de las MPDUs anteriores al rango del número de secuencia para las cuales el Receptor tiene registro deben ser reportadas como exitosamente recibidas. Si la trama `BlockAck` indica que una MPDU no fue recibida correctamente, el Originador deberá reintentar esa MPDU sujeto al límite de tiempo apropiado de la MPDU.

La figura 1.15 [14] muestra una secuencia típica de intercambio de tramas `BlockAck` usando el mecanismo `Block ACK` inmediato.

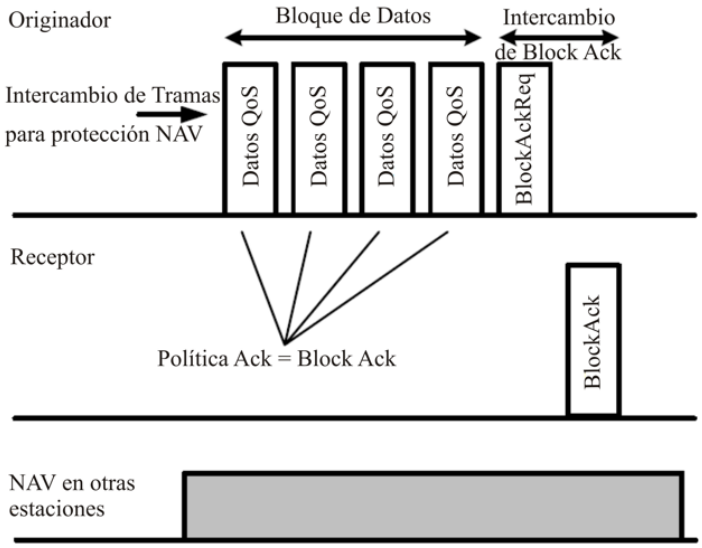


Figura 1.15. Secuencia Típica del Mecanismo `Block ACK` Inmediato.

La figura 1.16 [14] ilustra una secuencia típica de intercambio de tramas `BlockAck` usando el mecanismo `Block ACK` retardado. En ésta se observa claramente cómo el mensaje `BlockAck` no es recibido inmediatamente termina el intercambio de tramas de datos.

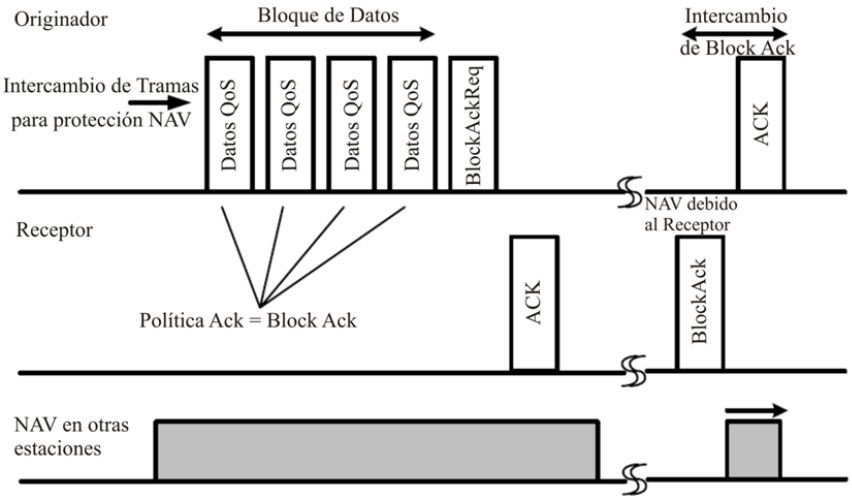


Figura 1.16. Secuencia Típica de `Block ACK` al Usar la Política Retardada.

El originador puede continuar transmitiendo MPDUs al receptor después de transmitir la trama BlockAckReq, pero antes de recibir la trama BlockAck (aplica solo a Block ACK Retardado). La trama BlockAck debe incluir el estado de las tramas recibidas entre el número de secuencia inicial y la transmisión de la trama BlockAckReq. La trama BlockAckReq debe ser descartada si todas las MSDUs referenciadas por este BlockAckReq han sido descartadas en el *Buffer* de transmisión debido a la expiración de su tiempo de vida máximo.

Con el fin de mejorar la eficiencia, los originadores pueden enviar tramas MPDU con el subcampo politica de ACK en las tramas de control de QoS fijado en ACK Normal si solo unas pocas MPDUs están disponibles para la transmisión. El registro de BlockAck deberá ser actualizado independientemente de la política de ACK utilizada para un TID con un Block ACK activo. Cuando hay un número suficiente de MPDUs, el Originador puede usar Block Ack. La recepción de tramas de datos QoS que usa la política ACK Normal no debe ser utilizada por el receptor con el fin de reiniciar el contador para detectar Block ACKTimeouts. Esto permite que el Receptor borre el BlockAck si el originador no vuelve a usar Block ACK.

El esquema general de funcionamiento del mecanismo Block ACK descrito anteriormente, sirve de base para la creación de un mecanismo Block ACK adecuado a largas distancias, en que el tipo de tráfico sea indiferente, ya que el tiempo que demora la transmisión de un extremo al otro y los tiempos de respuesta con los ACK no permiten garantizar un tratamiento especial para cierto tipo de tráfico que necesita cumplir requerimientos específicos de retardo y *jitter*[32]. Es por eso, que este trabajo de grado buscó implementar un mecanismo Block ACK que brinde las mejores prestaciones sobre un enlace inalámbrico de larga distancia para todo tipo de tráfico, además de realizar la comparación con los mecanismos de reconocimiento ya existentes, con la premisa de encontrar el mejor mecanismo sobre el escenario planteado.

1.3.3 No ACK

El subcampo de política de ACK se fija en 10 en todas las tramas que el transmisor no requiera reconocimiento, para que el receptor no realice acción alguna al recibir una trama. Esta combinación también se usa para tramas *broadcast* y *multicast* que usan el formato de tramas QoS.

El uso de No ACK está determinado por las políticas en la estación QoS. Cuando se usa esta política, no hay recuperación a nivel MAC, y la fiabilidad de este tráfico se reduce de acuerdo a la fiabilidad del tráfico con otras políticas de reconocimiento, debido a la mayor probabilidad de tramas perdidas por interferencia, colisiones, o parámetros del canal que varían en el tiempo. Un mecanismo de protección como transmitir usando HCCA, RTS/CTS, o el mecanismo de protección para receptores no ERP debe ser usado para reducir la probabilidad de que otras estaciones transmitan durante la TXOP.

La Tabla 1.6 [14], resume las diferentes opciones de métodos de reconocimiento para el estándar 802.11 cuando la opción QoS se encuentra activa.

[32] Jitter: Cambio o variación de latencia entre paquetes de datos que se reciben consecutivamente.

Tabla 1.6. Subcampo Política de ACK.

Bits Control de QoS		Descripción
Bit 5	Bit 6	
0	0	Normal ACK. El receptor envia un ACK o una trama QoS +CF-ACK después de un SIFS. Es el valor por defecto para las tramas que no requieren QoS.
1	0	No ACK. El receptor no emite respuesta tras la recepción de una trama.
0	1	Sin reconocimiento explícito. Puede haber una trama de respuesta a la trama recibida, pero no es un ACK o una trama de datos del tipo +CF-ACK.
1	1	Block ACK. El receptor registra el estado indicado en la trama y se prepara para el momento en que le llegue una trama BlockAckReq a la cual responde usando el procedimiento descrito en 1.3.2.

1.4 CONSIDERACIONES

Después de estudiar los problemas presentes al usar 802.11 en enlaces de larga distancia y los métodos de reconocimiento contemplados en el estándar se pueden definir los principales problemas que presentan los mecanismos de reconocimiento.

1.4.1 ACK

La observación del mecanismo ACK permite inferir que al aumentar la distancia para la cual fue diseñado el estándar, se incumple con los parámetros establecidos para dicha distancia, específicamente un impacto en el desempeño, pues el tiempo de viaje de los paquetes en el aire cambia de 1µs a Xµs, de manera que el tiempo que tiene un paquete de reconocimiento para ser entregado al origen no es suficiente para recibirlo dentro de las normas establecidas por CSMA/CA, hecho que obliga al sistema a retransmitir cuantas veces sea necesario hasta que un paquete ACK con igual número de secuencia reconozca el paquete originalmente transmitido.

Basado en el mismo principio de incumplimiento de ACKTimeout, si la solución fuese incrementarlo hasta un valor superior no adecuado para la distancia del enlace, se evidencia que si los paquetes de información emitidos presentan alguna distorsión durante el tiempo de tránsito en el aire, y son recibidos o incluso en el caso de que no lo fuesen, el tiempo de reacción de la estación transmisora para identificar que la trama no fue correctamente recibida sería superior al tiempo realmente necesario para retransmitir la trama y la red quedaría momentáneamente en un estado de no utilización ante la espera de un ACK inexistente.

De ahí que sea necesario ajustar el ACKTimeout en orden de poder establecer un correcto funcionamiento de la red y lograr que el tiempo de utilización de la red sea similar a las condiciones normales de funcionamiento de una WLAN y que el número de retransmisiones sea lo menor posible.

1.4.2 Block ACK

En el caso del Block ACK se presenta el mismo problema anteriormente descrito, pues al igual que en el mecanismo ACK, un mensaje de reconocimiento debe viajar desde el receptor hasta el transmisor indicando el correcto procedimiento de recepción del bloque de tramas, siendo más crítico el problema ya que la perdida de dicha trama implica la retransmisión de un grupo de tramas de las cuales algunas pudieron ser bien recibidas. El hecho anterior obliga a pensar la manera en que las estaciones deben responder ante la pérdida del mensaje ACK y qué número de tramas seria el indicado de transmitir para que en el caso de ser necesaria una retransmisión no se vea altamente disminuida la velocidad de transmisión efectiva.

1.4.3 No ACK

El esquema de funcionamiento de No ACK funciona sobre la base del mejor esfuerzo, es decir, la red no provee alguna garantía de que los datos enviados se recibieron correctamente y no se toma alguna medida de precaución, quedando las tramas a la merced del comportamiento del canal de radio y a la probabilidad de que un flujo de datos llegue correctamente sin la necesidad de retransmitirlo.

Después de estudiar los problemas presentes al usar 802.11 en enlaces de larga distancia y los métodos de reconocimiento contemplados en el estándar se da cumplimiento al primer objetivo específico propuesto en el Anteproyecto y se puede continuar con el proceso de investigación para determinar qué mecanismo es el que mejor se comporta en dicho tipo de enlaces. Para esto, en los siguientes capítulos se propuso una arquitectura práctica experimental bajo escenarios similares a los planteados en el proyecto Frida que permitió probar los mecanismosde reconocimiento, seleccionar uno adecuado para largas distancias y establecer si los mecanismos existentes son o no, adecuados en escenarios no contemplados en el diseño original de WiFi.

2 PLATAFORMA DE DESARROLLO A NIVEL MAC

Una vez identificados los mecanismos de reconocimiento a tener en cuenta y qué dificultades presentan en largas distancias, fue necesario explorar un conjunto de herramientas que permitieran realizar los ajustes necesarios para que el funcionamiento de las redes inalámbricas en largas distancias sea el adecuado. Este proceso exploratorio se presenta en la primera parte de este capítulo. En la segunda sección se expone cómo se crea un escenario de compilación cruzada para poder configurar el *firmware* de código abierto OpenWrt junto con OpenFWWF en el enrutador WRT54GL y de qué manera funciona el controlador de la tarjeta inalámbrica con la cual trabaja éste dispositivo. Finalmente se presentan algunos criterios de diseño que se deberían tener en cuenta al iniciar el desarrollo de un mecanismo Block ACK.

2.1 HERRAMIENTAS PARA EL DESARROLLO DE NUEVOS MECANISMOS MAC

Para desarrollar nuevas funcionalidades de nivel MAC, sin recurrir a equipos especializados ni incurrir en muchos gastos, es necesario el uso de herramientas de código libre y equipos versátiles de bajo costo como el enrutador WRT54GL, que trabajando en conjunto, permiten crear plataformas de desarrollo tan avanzadas para introducir nuevos esquemas de reconocimiento en 802.11. En las siguientes secciones se presentan los elementos que componen la arquitectura de dicha plataforma que facilitaron el proceso de desarrollo del objetivo específico 2 “Adecuar los mecanismos de reconocimiento ACK y Block ACK a las características de enlaces de larga distancia” propuesto en el Anteproyecto.

Como base de partida se eligió Linux, como sistema operativo de código abierto en el cual se puede trabajar con los firmwares disponibles para el enrutador WRT54GL.

2.1.1 Sistema Operativo Linux

GNU/Linux[33] es un sistema operativo que aparte de distribuirse libremente, posee una funcionalidad, adaptabilidad y robustez que lo ha establecido como una poderosa alternativa a los sistemas operativos propietarios como UNIX o Windows. Empresas del sector como IBM, Hewlett-Packard y otras gigantes del mundo de la computación lo han adoptado como plataforma de desarrollo de sus servicios y promueven su evolución. Además de ser un sistema ampliamente orientado a servidores y ambientes empresariales, sus desarrolladores se han preocupado por hacerlo amigable y brindar aplicaciones de uso diario en los hogares para también posicionarse en ese sector.

A pesar de que existe un gran número de implementaciones, cada una con características diferentes, todas se basan en el mismo componente básico, el *kernel*. Este componente es el encargado de dar la funcionalidad básica de Linux y a partir de él las empresas desarrollan aplicaciones e interfaces de acuerdo al público al que orientan su distribución. Es así que aparecen distribuciones como: RedHat, Suse, Mandrake, Debian, Slackware y Ubuntu, entre muchas otras que aparentan ser muy distintas por su presentación y gestión de archivos pero que están basadas en un mismo esquema fundamental.

La flexibilidad que brinda Linux hace posible instalarlo en servidores y supercomputadores, computadores de escritorio, sistemas embebidos, celulares y casi cualquier dispositivo cuyo nivel de procesamiento y arquitectura hagan posible albergar los componentes mínimos de funcionamiento de Linux [16]. Gracias a esto enrutadores como el WRT54GL de Linksys poseen un sistema operativo de libre distribución y código abierto en el cual se pueden realizar modificaciones y convertirlo más que en un enrutador inalámbrico, en un minicomputador capaz de realizar funciones más complejas que amplían las capacidades para las cuales fue diseñado.

[33] GNU es un proyecto que se desarrolló con el objetivo de crear un sistema operativo completo tipo Unix de *software* libre al que se llamó sistema GNU. Debido a que el núcleo de GNU no se finalizó, se implementó el núcleo Linux. La combinación de GNU y Linux es el sistema operativo GNU/Linux. Sin embargo, se popularizó con el nombre de Linux, y por esta razón se hace referencia a él de esta manera.

La distribución sobre la que se trabajó en este proyecto fue Ubuntu, pues, reúne las características de funcionamiento y compatibilidad que se requieren para compilar y adecuar los *firmwares* que se usaron.

2.1.2 Enrutador WRT54GL

El enrutador WRT54G, antecesor del WRT54GL, surge a mediados del 2002 como una anticipación a la especificación 802.11g, con la premisa de alcanzar velocidades de 54 Mbps, superando ampliamente a los 11 Mbps de 802.11. El WRT54G, como cualquier otro enrutador inalámbrico, fue destinado a compartir la conexión de Internet, ya sea cable o Línea de Abonado Digital (DSL, *Digital Subscriber Line*), con los clientes conectados a sus interfaces *Ethernet* y con los clientes al alcance de su interfaz inalámbrica. Pero a diferencia de los demás enrutadores de su tiempo, su *firmware* está basado en componentes Linux, motivo que obligó a Linksys a liberar el código fuente de su *firmware* para así cumplir con los términos de la Licencia Pública General (GPL, *General Public License*) bajo la cual opera Linux [17]. Para continuar con la comercialización de estos enrutadores decidieron lanzar la versión WRT54GL (la letra L representa la presencia de Linux en el dispositivo) y continuar con versiones WRT54G con el *firmware* propietario de Linksys.

Con la disposición de este código, muchos desarrolladores se dedicaron a implementar nuevas características, ampliando las capacidades y fiabilidad del dispositivo hasta convertirlo en una plataforma inalámbrica con características que solo enrutadores de altas prestaciones pueden brindar. Por ejemplo, algunas de los desarrollos para el WRT54GL permiten [17]:

- Crear un Sistema de Distribución Inalámbrico (WDS, *Wireless Distribution System*) o una red en malla.
- Habilitar el enrutador como repetidor o puente.
- Crear un servidor de Red Privada Virtual (VPN[34], *Virtual Private Network*) o de Voz sobre IP (VoIP[35], *Voice over IP*).
- Autenticar usuarios con Servidor Remoto de Autenticación de Usuarios (RADIUS[36], *Remote Authentication Dial-In User Server*).
- Controlar el ancho de banda por protocolo.
- Soportar IPv6[37].
- Aumentar la potencia de transmisión.
- Crear y acceder a *log*[38].
- Correr varias aplicaciones Linux.

Gracias a la liberación del código fuente se creó una plataforma libre y de bajo costo para la experimentación y la evaluación de redes basadas en las especificaciones 802.11, permitiendo así el desarrollo de este proyecto y de muchos otros relacionados con la investigación de este tipo de redes.

2.1.3 OpenWrt

Con el fin de controlar el funcionamiento del enrutador, es necesario usar un software, *firmware*, que trabaje como sistema operativo, este se encarga de convertir un simple enrutador inalámbrico en un minicomputador.

La primera gran implementación de un *firmware* de código abierto para el enrutador WRT54GL fue OpenWrt, el cual, en vez de saturar el *firmware* con muchas características, provee un mínimo de éstas,

[34] Una VPN es una red virtual que se crea dentro de otra red real, como puede ser Internet. Es una estructura de red corporativa implantada sobre una red de recursos de carácter público que utiliza el mismo sistema de gestión y las mismas políticas de acceso que se usan en las redes privadas.
[35] VoIP es una tecnología que permite la transmisión de voz a través de redes IP en forma de paquetes de datos.
[36] Es un protocolo de autenticación y autorización para aplicaciones de acceso a la red o movilidad IP.
[37] El Protocolo de Internet versión 6 (IPv6, *Internet Protocol version 6*) fue diseñado para reemplazar el Protocolo de Internet versión 4 y una vez esté implementado en todo el mundo proporcionará $340x10^{39}$ direcciones IP.
[38] Un log es un registro oficial de eventos durante un periodo de tiempo en particular.

ofreciendo la posibilidad de añadir aplicaciones desarrollados por los programadores. Este sistema de personalización, permite que los programadores desarrollen aplicaciones y las pongan a disposición de los usuarios sin tener que lanzar un nuevo *firmware* y que los usuarios instalen y desinstalen los paquetes que consideren necesarios. Existen otros *firmwares* como DD-WRT que ofrecen gran cantidad de características directamente en sus distribuciones y no permiten seleccionar qué paquetes instalar, disminuyendo la memoria libre de trabajo y del sistema en los enrutadores, siendo recomendables para su utilización a nivel de configuración de servicios y aplicación, ya que presenta módulos para: el control de potencia, servidor web, control de ancho de banda, QoS, interacción con servidores RADIUS, aplicación para *Hotspots*[39], entre otros [18].

Basados en este principio se tomó la decisión de trabajar sobre OpenWrt, uno de los *firmwares* más estables para crear una compilación, el cual permite la adición de paquetes y nuevas características. Las siguientes razones muestran porque OpenWrt fue apropiado para el desarrollo de este proyecto [19]:

- OpenWrt permite incluir nuevos controladores *hardware* dentro del *firmware*, dando soporte a nuevas características a este nivel y ofreciendo la posibilidad de aumentar el número de dispositivos en los cuales puede ser instalado.
- Posee una plataforma propia o *Kit* de Desarrollo de *Software* (SDK, *Software Development Kit*) para el desarrollo de nuevas aplicaciones.
- Se puede instalar una versión básica del *firmware*, con elementos necesarios para el correcto funcionamiento del estándar 802.11b/g dejando gran parte de su memoria libre para una ejecución más estable.
- Se encuentra disponible toda la documentación necesaria acerca de su funcionamiento.

De las características más importantes que se encontraron en OpenWrt se destacan el soporte de WMM y posibilidad de modificar algunas variables privadas del entorno inalámbrico, como lo es el tiempo de espera para el ACK o `ACKTimeout`. Además, OpenWrt no solo implementa EDCA y TXOP para cumplir con lo estipulado en la certificación MultiMedia Inalambrica (WMM, *Wireless Multimedia*) que expide la WFA, si no que incluye el mecanismo No ACK (disponible únicamente si el *hardware* del enrutador lo soporta). Estos dos aspectos permitieron evaluar el comportamiento de un enlace 802.11 punto a punto de larga distancia con los mecanismos de reconocimiento ACK con ACKtimeout modificado y No ACK

En un principio se creyó posible realizar la totalidad del proyecto con OpenWrt, pero ante la detenida revisión y exploración del sistema, se encontró que el *firmware* no manejaba todos los dispositivos del WRT54GL directamente, sino que relega el control de ellos a software independiente, de manera que para realizar cambios a nivel MAC es necesario modificar dicho *software.* En el mismo sentido se encontró que el modulo controlador de la tarjeta inalámbrica Broadcom que viene con el WRT54GL, es propietario y consta de dos partes, una embebida en el chip que se carga al iniciar el dispositivo y otra que comunica el hardware con el *firmware* OpenWrt. Así, a continuación se presentan las alternativas estudiadas.

2.2 ALTERNATIVAS DE DESARROLLO

Ante la imposibilidad de modificar el firmware embebido en el chip Broadcom, fue necesario explorar otras posibilidades que permitiesen variar la manera en que funciona el nivel MAC, ya fuese con un controlador inalámbrico que sea totalmente controlado por software, o un firmware inalámbrico programable para algún chip específico.

2.2.1 MadWifi

El Controlador Multibanda de Atheros para WiFi (MadWifi, *Multiband Atheros Driver for Wireless Difelity*) es uno de los controladores inalámbricos más avanzados para WLAN disponibles en Linux, este como tal

[39] Zona de cobertura Wi-Fi, en la que uno o varios puntos de acceso proporcionan servicios de red.

es un controlador de dispositivos WLAN basados en los chips de Atheros para el kernel de Linux. El controlador como tal es de código abierto pero depende de una Capa de Abstracción del Hardware (HAL, *Hardware Abstraction Layer*) que se encuentra en forma binaria y pertenece a Atheros, pues este controlador soporta únicamente ese tipo de dispositivos [20].

El controlador trabaja de tal forma que la tarjeta WLAN aparece como una interfaz de red normal para el sistema, pero adicionalmente soporta extensiones que permiten configurar la mayoría de aspectos inalámbricos y agregar capacidades que no ofrece directamente el fabricante.

Los modos de operación que soporta se encuentran los descritos en la Tabla 7. Además, permite emitir múltiples SSID, soporta WMM, cambio de AP sin interrupciones si la señal disminuye, entre otras.

Tabla 2.1. Modos de Operación de Madwifi.

Modo	Operación
Estación	Opera como una estación cliente típica de WLAN.
AP	Actúa como el Punto de Acceso para las estaciones cliente WLAN.
Ad-hoc	Modo IBSS, es decir, el dispositivo se conecta punto a punto con otro sin necesidad de un AP
Adhoc Demo	Es una vieja implementación del modo Ad-hoc de Atheros.
Monitor	Este modo permite capturar todas las tramas de 802.11 al alcance del dispositivo.
WDS	Usado para enlazar varios APs con el fin de crear grandes redes inalámbricas.

A pesar de las grandes ventajas que presenta Madwifi, éste tiene una gran limitante, la HAL.

2.2.1.1 **Capa de Abstracción de Hardware**

A diferencia de muchos otros dispositivos inalámbricos, los chips de Atheros pueden trabajar en un amplio rango de frecuencias y el software controlador puede manejar muchos aspectos del funcionamiento del radio. Pero las agencias regulatorias tienen leyes que restringen el uso del espectro electromagnético como se explicó en el Capítulo 1 y otras verifican la compatibilidad entre equipos inalámbricos como la WFA. Estas leyes y certificaciones conllevan a los fabricantes a desarrollar soluciones que fortalezcan el cumplimiento legal e interoperabilidad, de manera que el autor original de MadWifi, Sam Leffler, y Atheros acordaron usar una HAL como bloqueo de ciertos registros y procedimientos hardware [21].

La HAL es entonces un software en el que se encaminan todos los procedimientos de acceso al hardware y bloquea ciertos aspectos de su funcionamiento. Es importante resaltar que la HAL es un módulo del kernel y se presenta en modo binario, hecho que impide su fácil modificación.

El estudio de MadWifi y conversaciones con la ingeniera Sandra Salmerón Ntunumu de EHAS España permito clarificar que ciertas variables necesarias para la creación de un mecanismo en bloque se encuentran en una posición privilegiada dentro de la HAL de MadWifi. Asimismo, Sandra Salmerón propuso el Simulador de Red 2 (NS-2, *Network Simulator* - 2) como plataforma para crear y evaluar el funcionamiento de Block ACK.

2.2.1.2 **Futuro de MadWiFi**

Tras la popularidad alcanzada por MadWifi y la cantidad de usuarios y desarrolladores interesados en liberar la MAC para generar nuevas capacidades para las tarjetas de inalámbricas y dada la flexibilidad de los chips Atheros, surgieron dos proyectos denominados *ath5k* y *ath9k*, cuyo principal objetivo es ofrecer un driver que no dependa de una HAL cerrada, sino que las instrucciones puedan ser dirigidas directamente al hardware y así poder tener control total de dichas tarjetas [20] [22].

A largo plazo estos proyectos van a reemplazar a MadWifi, y de hecho, todo el trabajo de sus creadores se ha enfocado en este objetivo, pero la ejecución directa de las instrucciones aún se encuentra en una etapa prematura, de modo que la labor de diseñar un mecanismo de reconocimiento en bloque para estos drives es aún una tarea difícil de lograr.

2.2.2 NS-2

El Simulador de Red 2 (NS-2, *Network Simulator - 2*) es un simulador de eventos discretos orientado a la investigación de redes. Con NS-2 es posible simular protocolos de TCP, enrutamiento y *multicast* sobre redes cableadas e inalámbricas [23].

Se debe tener en cuenta que NS-2 no es un producto refinado ni terminado, sino que es un producto de un esfuerzo continuo de investigación y desarrollo, por tal razón aún se encuentran errores en el software que constantemente se están corrigiendo. Los usuarios del sistema son NS-2 son responsables de verificar que sus simulaciones no son invalidas a causa de dichos errores y de que los modelos implementados en el simulador son realmente los modelos que se desean.

Aunque NS-2 se presenta como una alternativa para desarrollar el Block ACK, se decidió no usarlo debido a que la esencia de este Proyecto de Grado es hacer una evaluación práctica del funcionamiento de los mecanismos de reconocimiento y realizar un aporte significativo al proyecto financiado por FRIDA.

2.2.3 OpenFWWF

Al continuar con el estudio de alternativas para desarrollar el mecanismo de Block ACK se encuentra el *Firmware* Abierto para Redes WiFi (OpenFWWF, *Open FirmWare for WiFi Networks*), éste es un proyecto que busca proveer una plataforma sencilla y barata para implementar nuevos mecanismos de Control de Acceso al Medio MAC, convirtiéndose en una alternativa a las simulaciones y a costosas plataformas *ad-hoc*. Actualmente se encuentra en la versión 5.2 [24].

OpenFWWF provee un *firmware* abierto para dispositivos basados en los *chips* Broadcom/Airforce de la familia 43XX. Su desarrollo comenzó con las especificaciones disponibles en [25] debido a que trabaja en conjunto con el controlador b43 agregándole la posibilidad de manejar las interacciones entre el procesador MAC, los registros *hardware* y el mecanismo de operación que controla las tramas de datos en el proceso de transmisión y recepción [24].

Este *firmware* está diseñado para trabajar como AP, como cliente o para funcionar en modo *ad-hoc* de igual manera que el *firmware* propietario, además cumple con los esquemas de acceso DCF de los modos b y g. El *firmware,* sin embargo, no implementa el procedimiento RTS/CTS *handshake* (esta limitación no afectó el desarrollo del presente trabajo, pues, un enlace punto a punto no requiere dicho procedimiento) ni funciona con la aceleración de criptográfica de *hardware*.

Gracias a las bondades encontradas en OpenFWWF y a la posibilidad de modificar el nivel MAC de 802.11 directamente sobre el *firmware* embebido para las tarjetas de la familia 43XX de Broadcom (presente en el enrutador WRT54GL), se definió usarlo bajo OpenWrt para crear una plataforma de desarrollo MAC.

A continuación se presentan los procedimientos necesarios para poner en funcionamiento dicha plataforma.

2.3 COMPILACIÓN CRUZADA PARA OPENWRT EN LINUX

Los enrutadores inalámbricos son prácticamente PCs de bajas prestaciones, en los cuales el tamaño de la memoria es reducido, en este sentido, OpenWrt es un *firmware* pensado para ocupar tan poco espacio como sea posible, pero con todas las prestaciones que se puedan incluir en este espacio. Es por esto,

que para compilar un paquete o compilar el *firmware* completo, se usa una técnica que se llama compilación cruzada.

La compilación cruzada es un proceso necesario, pues, al ser el enrutador un dispositivo con características limitadas, no es posible incluir las librerías y el *software* requerido para generar un código compatible con el microprocesador que posee, de manera que la programación y compilación se realiza en otro dispositivo que permite realizar estas tareas. Para éste caso concreto se usó un PC con procesador x86, Linux y las herramientas necesarias para compilar un *software* que finalmente fue ejecutado en el enrutador WRT54GL cuyo procesador opera bajo la arquitectura MIPS[40].

El proceso de compilación se inicia al ejecutar la instrucción *make* en la consola de Linux, la cual determina automáticamente qué piezas de un programa necesitan ser compiladas o recompiladas y de acuerdo a un conjunto de reglas definidas en un archivo *Makefile*, que describe las relaciones entre los archivos del programa, 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, automatizando qué código construir y cómo construirlo, liberando a los programadores de esta tarea para concentrarse en desarrollar el código. 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 [26].

Al conjunto de *Makefiles* y parches necesarios para la compilación se le denomina *buildroot*, este cumple funciones tan importantes como generar un conjunto de herramientas capaces de realizar la compilación cruzada, al cual se le denomina *toolchain*, además de 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 para sistemas embebidos con procesadores distintos a un X86, como PowerPC, MIPS, ARM, entre otros.

El *toolchain* es un conjunto de herramientas de programación que permiten compilar el código para un sistema determinado. En este sentido, un equipo Linux 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 la máquina en la cual está funcionando se llama "sistema huésped", tales herramientas de compilación son proporcionadas por la distribución de Linux con la cual se trabaje siendo *libc* de GNU la biblioteca estándar de *C*[41].

De acuerdo a 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 y procesador objetivo diferentes al sistema huésped. Para este caso particular el *toolchain* se compone de una Colección de Compiladores de GNU (GCC[42], GNU *Compiler Collection*) [27], utilidades binarias como *binutils* para ensamblar y enlazar, y una biblioteca estándar de *C* especial para sistemas embebidos llamada *uClibc*. De manera que, si el sistema huésped utiliza un procesador x86 y el sistema 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.

---

[40] Microprocesador sin Bloqueos en las Etapas de Segmentación (MIPS, *Microprocessor without Interlocked Pipeline Stages*) es en sí un microprocesador con un Conjunto Reducido de Instrucciones (RISC, *Reduced Instruction Set Computer*) desarrollado por MIPS Technology, utilizado en múltiples sistemas embebidos como TiVo, dispositivos Windows CE, enrutadores Cisco, etc.

[41] *C* es un lenguaje de programación diseñado para desarrollar sistemas operativos, como Unix, pero permite también la creación de aplicaciones.

[42] GCC es un compilador integrado del proyecto GNU para C, C++, Objective C y Fortran; es capaz de recibir un programa fuente en cualquiera de estos lenguajes y generar un programa ejecutable binario en el lenguaje máquina.

Es posible compilar directamente teniendo las herramientas adecuadas, GCC*, binutils*, *uClibc*, pero se presentan diversos problemas debido a que se deben configurar manualmente todas las opciones para la compilación y además existen incompatibilidades entre cada versión de GCC, *uClib* o *binutils*, debido a esto se creó la herramienta *buildroot* para OpenWrt que automatiza este proceso con el uso de *Makefiles*, y tiene una colección de parches para cada versión de GCC y *binutils* para que funcionen en la arquitectura MIPS de la mayoría de enrutadores inalámbricos.

2.3.1 Obtención del *Buildroot* de OpenWrt

Para desarrollar el *firmware* para el WRT54GL de Linksys, que incluye las modificaciones adecuadas para el cumplimiento de los objetivos propuestos en este trabajo de grado, se toma como base la versión de desarrollo de OpenWrt que se debe descargar directamente del servidor de OpenWrt mediante la herramienta SVN[43] digitando el siguiente comando en la consola Linux:

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

La versión estable recomendada de OpenWrt es la 8.09.1 también conocida como Kamikaze y también se puede descargar mediante SVN:

```
svn co svn://svn.openwrt.org/openwrt/branches/8.09
```

Dado que en la versión *trunk* (versión de desarrollo) contiene los *Makefiles* y la estructura de programa del *Firmware* Abierto para Redes WiFi (OpenFWWF, *Open FirmWare for WiFi Networks*) desarrollado por el grupo de Redes de Telecomunicaciones NTW de la *Università degli Studi di Brescia* en Italia, *software* con el cual se va a trabajar, se encontró la manera para no tener que escribir los *Makefiles* y demás archivos de configuración necesaria para incluir su última versión dentro de OpenWrt.

2.3.2 Manejo del *Buildroot* de OpenWrt

El *Buildroot* de OpenWrt tiene una herramienta de configuración similar a la que se puede encontrar en el *kernel* de Linux o en *busybox*[44] [28]. No es necesario ser usuario *root* para configurar y usar el *Buildroot*, un usuario sin privilegios especiales puede ejecutar el ayudante de configuración en el directorio en el cual se descomprime el OpenWrt, de la siguiente manera:

```
trunk ~# make menuconfig
```

Cada opción en la herramienta de configuración, tiene una ayuda asociada que describe el propósito de la entrada. Una vez que se configuran los paquetes, aplicaciones y utilidades, la herramienta genera un archivo *Config.in* que contiene la descripción de la configuración. Este archivo es utilizado por los *Makefiles* como guía en el proceso de construcción del *firmware.*

Tras configurar las opciones deseadas en *menuconfig*, se ejecuta la siguiente instrucción:

```
trunk ~# make
```

Este comando ejecuta el código referenciado en los *Makefiles*, tomando como guía el archivo de configuración *Config.in*, descarga las fuentes necesarias de Internet, configura y compila todas las herramientas seleccionadas, y finalmente genera la imagen[45] del *firmware* para el sistema objetivo y los

[43] Subversión (SVN, *Subversion*) es un *software* de control de versiones de código abierto. Se encarga de controlar los cambios que realizan los programadores sobre el código fuente y de almacenar la información de todo el desarrollo en sitios conocidos como repositorios.
[44] *Busybox* es un sistema operativo con aplicaciones típicas de Unix simplificadas, escrito pensando en la optimización del tamaño y destinado principalmente a sistemas empotrados e instalaciones de Linux.
[45] Imagen hace referencia a una compilación del firmware.

paquetes adicionales seleccionados en el paso previo. Todos los archivos binarios para la arquitectura objetivo se pueden encontrar en la carpeta *bin* dentro del directorio *trunk*.

Se pueden compilar las imágenes del *firmware* para las diversas arquitecturas en distintos tipos de sistemas de archivos. Los sistemas de archivos para este tipo de dispositivos deben estar comprimidos, pues, es necesario que ocupen el menor espacio posible. Pero la decisión de qué tipo de sistema de archivos depende del medio de almacenamiento que se va a usar. Un sistema basado en un disco RAM (*ramdisk*[46]), copia la información de la memoria *flash* al *ramdisk*, pero ya que la RAM es esencial para la operación del sistema, se prefiere un tamaño pequeño del *ramdisk*. Por otro lado, los sistemas basados en Discos Duros o *Compact Flash*[47], ofrecen mayor capacidad de almacenamiento pero aún puede ser poco para que todos los archivos se almacenen sin necesidad de compresión.

Los tipos más comunes de sistemas de archivos empleados en sistemas embebidos son JFFS2 y SquashFS:

- 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 de la mínima unidad borrable (*eraseblock*) del *chip* de memoria *flash* del sistema de archivos JFFS2.

- 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, con la ventaja de que no es posible dañar por error el sistema de archivos original, debido a que no se escribe directamente sobre él y en caso de error, se puede regresar el dispositivo a los valores por defecto deshaciendo los cambios realizados.

De acuerdo al proyecto OpenWrt, que usa tanto SquashFS como JFFS2, SquashFS provee mejor desempeño que JFFS2. Adicionalmente, al menos en el caso de los pocos archivos que necesitan ser actualizados en el proyecto, hay una pequeña ventaja en usar un sistema raíz de archivos comprimido de lectura/escritura como JFFS2 con respecto al impacto en el desempeño en un sistema raíz de solo lectura como SquashFS usado con un sistema JFFS2 para los archivos almacenados.

2.3.3 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:

- 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: `/trunk/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 reconstruye la imagen del sistema de archivos. Este método permite hacer todo en el sistema de

[46] Un disco RAM es una unidad de disco que usa una zona de la memoria RAM del sistema para almacenar datos como si se tratase de un disco duro.
[47] *CompactFlash* es un dispositivo de almacenamiento de datos, especialmente empleado en dispositivos portátiles. Para el almacenamiento, las *CompactFlash* emplean memoria *flash.*

archivos para la arquitectura objetivo, pero al reconstruir el *toolchain*, las herramientas o paquetes creados se pierden.

- Modificar el esquema del sistema de archivos para requisitos particulares que se encuentra disponible en: `/trunk/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 esquema del sistema de archivos, pero los cambios que se hagan persisten aún cuando se reconstruya totalmente el *toolchain* de la compilación cruzada y las herramientas.

2.3.4 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*. En el menú de configuración de *Busybox* "*Busybox Configuration"* que se encuentra en "*OpenWrt Package Selection*" se puede seleccionar qué elementos incluir dentro de la compilación.

2.3.5 Modificar la Configuración de uC*libc* para Requisitos Particulares

Al igual que *BusyBox, uClibc* ofrece varias opciones de configuración por lo que se permite seleccionar las funcionalidades que se desea tener a disposición 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: `/trunk/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` o `/toolchain/uClibc/uClibc.config-locale`.

2.3.6 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 se usa 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 siguientes pasos:

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

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

- Crea el directorio *build toolchain (*`/trunk/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 (*`/trunk/staging_dir_ARCH/` por defecto). Aquí es 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 `/trunk/staging_dir_ARCH/bin` a su respectivo PATH, y luego utilizar *arch-linux-gcc* para compilar. Para la construcción del directorio *staging*, primero se quita, y luego se crean varios subdirectorios y enlaces simbólicos dentro de él.

- Crea el directorio objetivo *(*`/trunk/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 los directorios SVN inútiles.

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

2.3.7 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 el *buildroot* por defecto se encuentra `en /trunk/staging_dir_ARCH`. La forma más simple para usarlo es añadir `/trunk/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.

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

Por defecto, la compilación cruzada se genera en el directorio `/trunk/staging_dir_ARCH/` pero se puede generar en otro directorio. Esta compilación puede ser usada para generar otros programas por cualquier usuario. Mover el directorio directamente es inútil debido a que hay código que está fuertemente ligado al *toolchain*, es necesario entonces configurar el *buildroot* para que lo genere usando la herramienta de configuración. En el menú "*Build options*" se selecciona "*Toolchain and header file location*" para modificar el directorio por defecto.

2.4 CONFIGURACIÓN Y COMPILACIÓN DEL FIRMWARE OPENWRT CON OPENFWWF

Tras descargar el código fuente descrito en la sección 2.2.1 que se compone de los directorios: *docs, include, package, scripts, target, toolchain y tools* y los archivos *BSDmakefile*, *Config.in*, *feeds.conf.default*, LICENSE, *Makefile*, README y *rules.mk*, se procede a configurarlo y compilarlo. Para iniciar el proceso de compilación es necesario tener instaladas las herramientas *gcc, g++, binutils, patch, bzip2, flex, bison, make, gettext, unzip, libz-dev y libc headers*, descritas en el archivo README

2.4.1 Interfaz Web

Para agregar soporte gráfico a OpenWrt es necesario instalar los paquetes de LuCI (*Lua*[48] *Configuration Interface*), que incluye varias herramientas de monitoreo y una interfaz web para dispositivos embebidos que hacen uso de OpenWrt [29].

Al ejecutar la siguiente instrucción se actualizan y descargan los complementos disponibles para el *firmware*:

```
trunk ~# ./scripts/feeds update
```

Después, se instala LuCI mediante la siguiente instrucción:

```
trunk ~# ./scripts/feeds install -a -p luci
```

Una vez se encuentra instalada, LuCI queda disponible en los paquetes de configuración.

2.4.2 Menú de Configuración

Como se mencionó anteriormente, es necesario configurar primero el *buildroot* con los paquetes y opciones deseadas para obtener un *firmware* adecuado a las necesidades del proyecto. Para realizar esto es necesario ingresar al directorio *trunk* desde una consola y ejecutar el siguiente comando:

```
/trunk ~# 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 varios 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 2.1.

Se escogen las opciones de manera que el enrutador no quede sobrecargado con recursos que no se van a utilizar, y solo ocupan al procesador y usan espacio en memoria. Esto con el fin de utilizarlos en el desarrollo del nuevo módulo.

[48] Lua es un lenguaje de programación rápido, ligero con énfasis en sistemas embebidos, que combina procedimientos de sintaxis simple con descripciones de datos construidos en arreglos asociativos y semánticas extensibles.

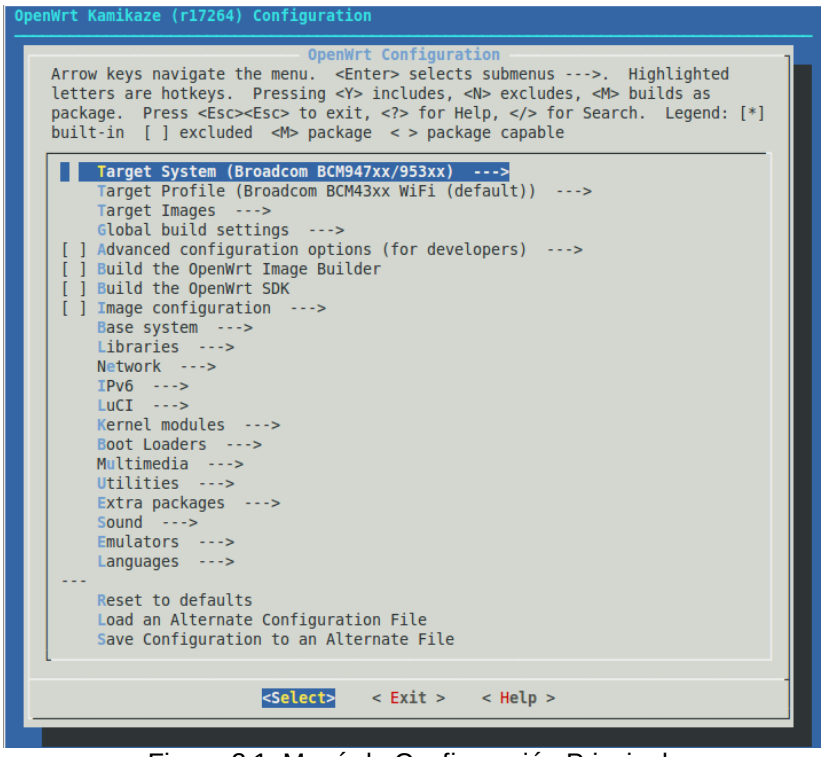


Figura 2.1. Menú de Configuración Principal.

Para compilar OpenFWWF es necesario trabajar con el *kernel* 2.6, para esto es necesario escoger dentro de "*Target System*" la opción BCM947xx/953xx que corresponde al *chipset* presente en el enrutador Linksys WRT54GL. Además de esto se debe escoger el perfil por defecto en "*Target Profile*" como "Broadcom BCM43xx WiFi (default)" [30].

El sistema de archivos que se escogió para usar es SquashFS, pues, los cambios realizados incorrectamente pueden ser corregidos al presionar el botón *reset* del enrutador y no transfiriendo de nuevo la imagen desde el PC. Al escoger "*Target Images*" como "SquashFS" queda definido el sistema de archivos.

Es necesario instalar los paquetes que habilitan al en modo AP y que manejan los procesos de autenticación de los usuarios. Con la instalación de los paquetes relacionados a *hostapd* (*hostapd, hostapd-mini, hostapd-utils*) dentro de la opción "*Network"* se asegura el correcto funcionamiento del enrutador, éste proceso se realiza como se ilustra en la Figura 2.2.

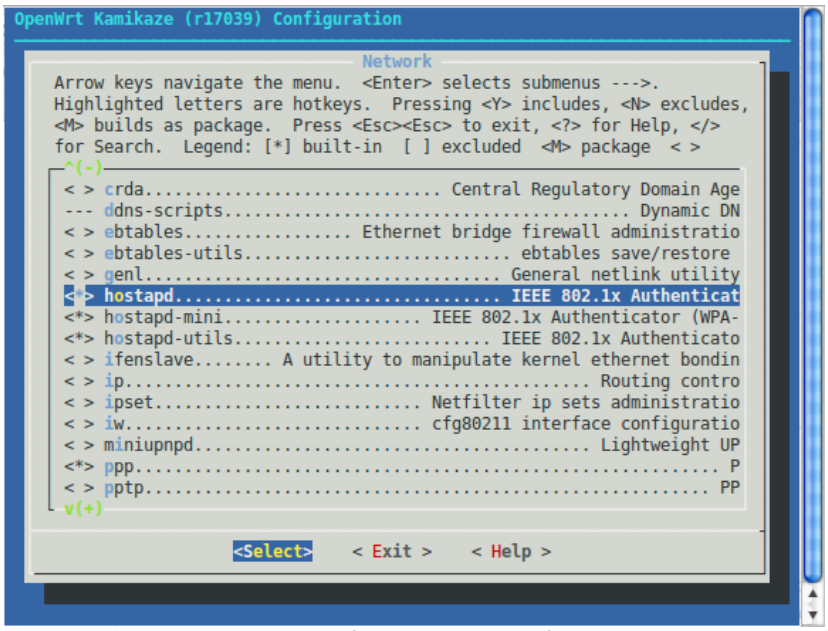


Figura 2.2. Menú de Configuración de Red.

En el menú principal se encuentra la opción "*Kernel modules*", en la cual se puede escoger los controladores que usa el sistema. Para seleccionar OpenFWWF es necesario agregar el módulo, pues, por defecto y al ser un módulo de desarrollo no viene predefinido para ser compilado. Para esto, en la sección "*Wireless Drivers*" se escoge kmod-b43 correspondiente a los drivers para los *chips* inalámbricos Broadcom de la familia 43XX como se muestra en la Figura 2.3.

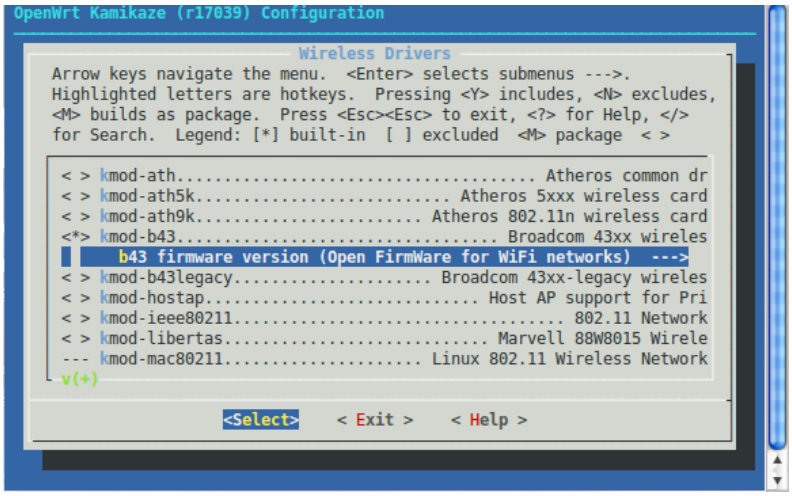


Figura 2.3. Menú de Controladores Inalámbricos.

Dentro de esta opción se encuentran tres candidatos, uno estable correspondiente a la última versión del driver b43 para Linux basado en los controladores originales de Broadcom, al igual que la versión 4.278.10.4 en desarrollo del mismo driver. La última opción corresponde a *Open FirmWare for WiFi Networks*, es decir, OpenFWWF, siendo éste la opción a elegir como lo muestra la Figura 2.4.

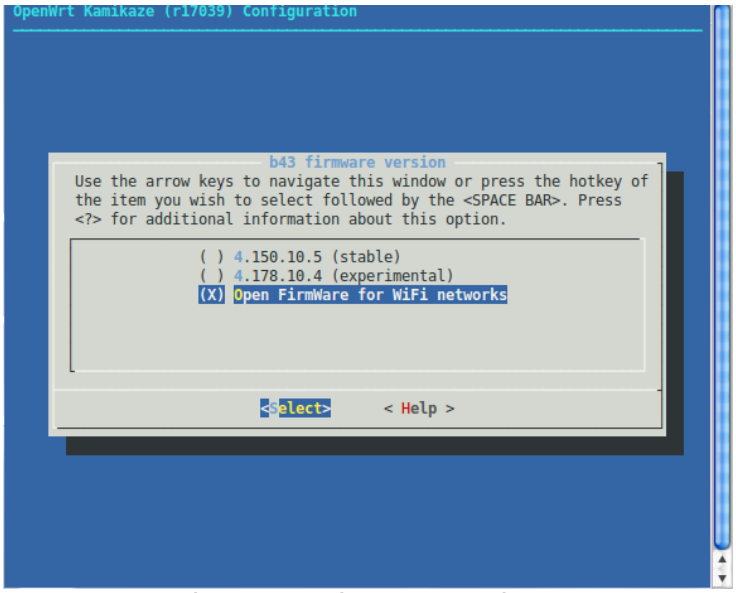


Figura 2.4. Menú de Selección de la Versión del *firmware* b43.

Para adicionar a la compilación los paquetes correspondientes a la interfaz web LuCI es necesario habilitarlos en el archivo Config.in, esto se realiza seleccionando en el menú principal la opción “LuCI” y dentro de ésta los componentes *luci-admin-core, luci-admin-full, luci-app-ddns, luci-app-firewall, luci-app-ntpc, luci-app-statistics,* como se muestra en la Figura 2.5.

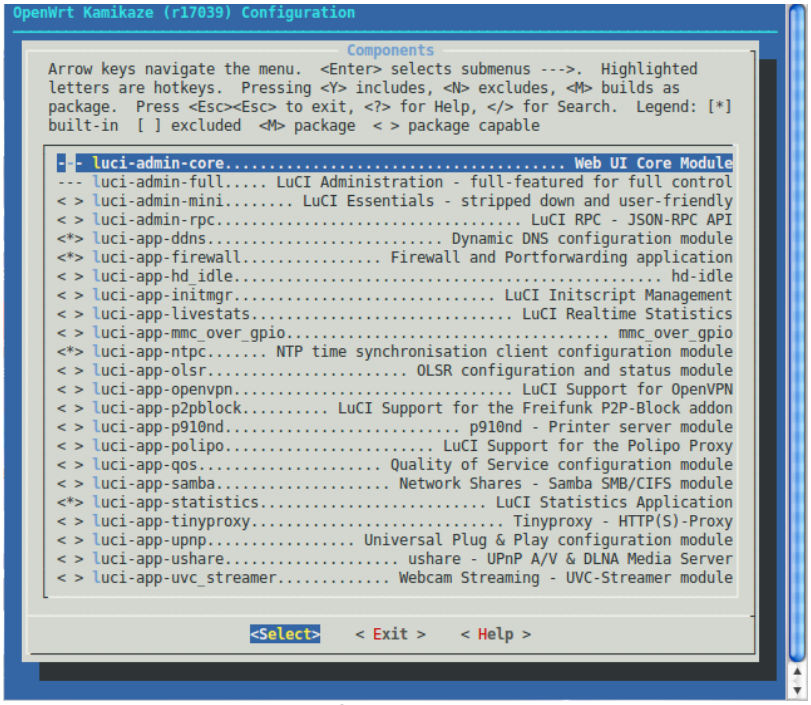


Figura 2.5. Menú de Componentes de LuCI.

Los componentes anteriores agregan una funcionalidad completa a la interfaz, además de ciertas aplicaciones útiles como recopilación de estadísticas y bloqueo de acceso para usuarios no deseados, entre otras.

Además, es necesario seleccionar el tema de presentación de LuCI, de manera que funcione la lógica de de la interfaz y se presenten los elementos gráficos en el navegador. La Figura 2.6 muestra la selección de *luci-theme-base y luci-theme-openwrt,*

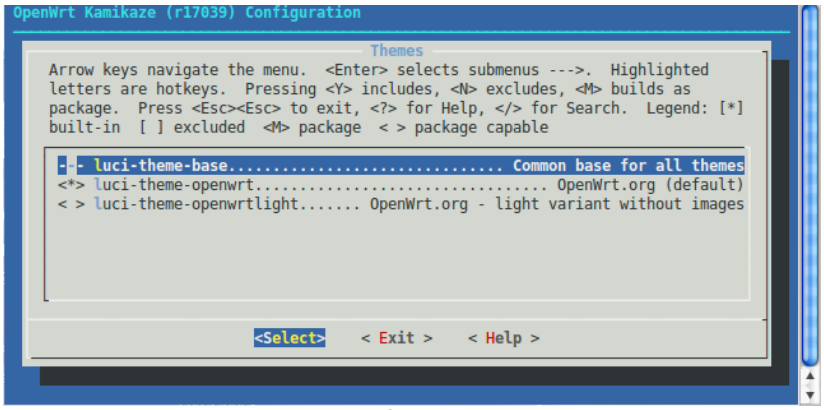


Figura 2.6. Menú de Temas de LuCI.

Tras ejecutar las instrucciones anteriores el *firmware* está listo para ser compilado, solo es necesario guardar la configuración y ejecutar desde la consola el comando *make*.

Si el tamaño de la imagen resultante es mayor al espacio disponible en el WRT54GL, es necesario buscar la manera de reducir el espacio. Para esto se ejecuta:

```
/trunk ~# make kernel_menuconfig
```

Ésta instrucción permite modificar aspectos directos del *kernel* seleccionado en *menuconfig.* En este caso se descarga automáticamente el *kernel 2.6* y se le aplican parches para que sea totalmente compatible con el dispositivo seleccionado. Tras terminar las tareas anteriores, se ejecuta el menú de configuración del *kernel* como lo muestra la Figura 2.7.

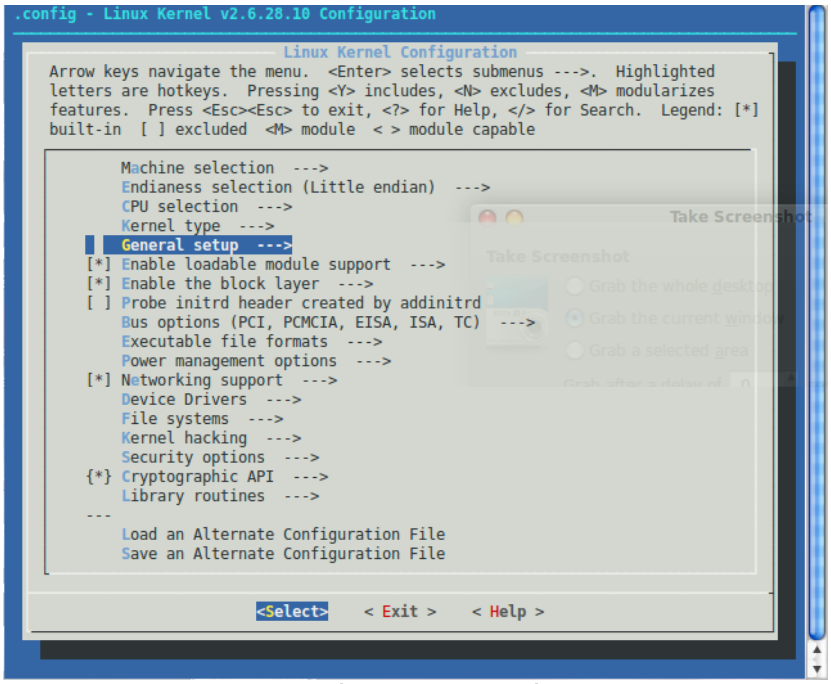


Figura 2.7. Menú de Configuración del *kernel*.

En éste menú es necesario ir a "*General setup"* y seleccionar "*Configure standard kernel features (for small systems")* en "*Optimize for size"* como se muestra en la Figura 2.8. Esta opción permite obtener suficiente espacio para crear una partición donde se puede almacenar los cambios que se realicen en el sistema raíz del *kernel* Linux.

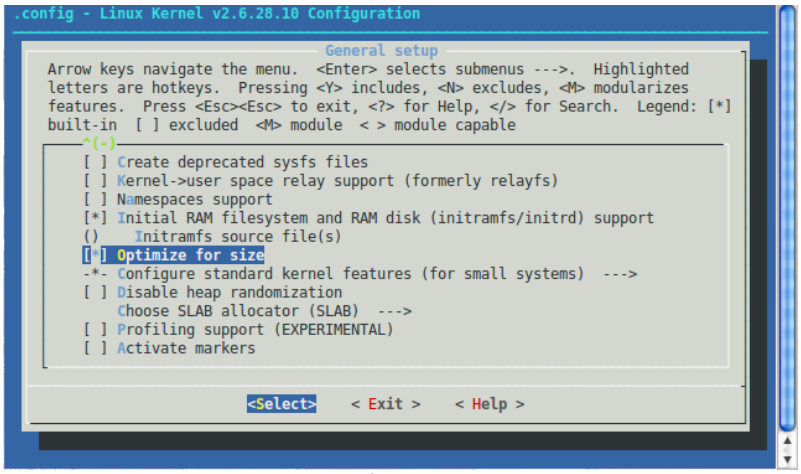


Figura 2.8. Selección de Optimizar Espacio.

Ahora sí, con espacio suficiente en el enrutador para albergar el sistema de archivos de OpenWrt se puede compilar el *firmware.*

```
/trunk ~# make
```

Con este comando se inicia la compilación y construcción del *Kernel*, se descargan las fuentes necesarias desde Internet y se realiza la compilación cruzada mediante el *toolchain.* Si no ocurren errores en la compilación debido a dependencias faltantes o mala configuración, el *firmware* OpenWrt con el controlador libre OpenFWWF para el *chip* Broadcom ha sido correctamente compilado.

2.4.3 Instalación de OpenWrt

Existen dos maneras sencillas de instalar OpenWrt, a través del Protocolo de Transferencia de Archivos Trivial (TFTP, *Trivial File Transfer Protocol*) o a través de la aplicación de Actualización de *Firmware* incluida en la instalación original del enrutador WRT54GL [31].

El método más sencillo de instalación es mediante la página de Actualización de *Firmware* en la interfaz web del *firmware* original del enrutador. Solo es necesario subir el nuevo *firmware* a través de la página web y seguir las instrucciones que allí se dan. El único problema con éste método es que si el suministro de energía falla o si se usa una imagen dañada, el enrutador quedaría inservible.

El otro método, a través de TFTP, es más usado para instalar imágenes compiladas por usuarios entusiastas con las cuales se pueda presentar algún problema, pues, si algo ocurre de manera no esperada, el mismo proceso sirve para regresar al *firmware* anterior. El proceso de instalación se describe a continuación.

Cuando el dispositivo inicia, éste corre un *bootloader* para realizar la inicialización básica del sistema además de validar y cargar el *firmware* actual. Durante la validación, se realiza una Comprobación de Redundancia Cíclica (CRC, *Cyclic Redundancy Check*), si se presume que el *firmware* instalado se encuentra dañado, el sistema esperará que un nuevo *firmware* sea cargado al dispositivo a través de alguna de sus interfaces de red. Aprovechando esta capacidad que brinda el *bootloader* es que se puede instalar un *firmware* por TFTP aun si el *firmware* actual no está corrupto. Los siguientes pasos describen el instructivo de instalación de un nuevo *firmware* a través de TFTP.

1. Es necesario desconectar el enrutador de la energía.
2. Iniciar el cliente TFTP en algún computador conectado al enrutador y configurarlo con alguna dirección IP de la red 192.168.1.0.
3. Fijar el modo octeto/binario en el cliente TFTP y la dirección IP de destino, es decir, la dirección por defecto del enrutador que es 192.168.1.1.
4. Indicar la imagen que se desea enviar y empezar su transmisión.
5. Conectar el enrutador a la energía mientras el cliente TFTP intenta establecer una conexión.
6. El cliente TFTP recibirá un reconocimiento del *bootloader* y empezará la transmisión de la imagen.

Si este proceso ha resultado exitoso, el enrutador se reiniciará automáticamente con el nuevo *firmware*. En caso contrario se puede volver a intentar el proceso sin riesgo de dejar el dispositivo inservible [32].

Una vez instalado OpenWrt, se puede acceder a él sin contraseña vía TELNET en la dirección 192.168.1.1, esto con el fin de establecer una contraseña de administrador. La Figura 2.9 muestra como ingresar a la consola del *firmware* y qué comando usar para establecer una contraseña. Al ejecutar el comando *passwd* se deshabilita el acceso a través de TELNET y se habilita únicamente el acceso a través de SSH o HTTPS si se ha instalado una interfaz web.

```
root@localhost:~$ telnet 192.168.1.1
 Trying 192.168.1.1...
Connected to 192.168.1.1.
 Escape character is '^]'.

BusyBox v1.11 Built-in shell (ash)
    Enter 'help' for a list of built-in commands.

  _______                     ________          __
 |       | .---- .---- .----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 WHITE RUSSIAN  -----------------------------------
  * 2 oz Vodka   Mix the Vodka and Kahlua together
  * 1 oz Kahlua  over ice, then float the cream or
  * 1/2oz cream  milk on the top.
-----------------------------------------------------
root@OpenWrt:~# passwd
```

Figura 2.9. Acceso a OpenWrt a través de telnet.

Es importante también habilitar la variable *boot_wait* que le indica al WRT54GL que debe retrasar el proceso de inicio del sistema, esto en caso de que sea necesario instalar un nuevo *firmware* mediante FTP [33]. La siguiente instrucción muestra como habilitar la espera:

```
root@OpenWrt:~# nvram set boot_wait=on
```

2.4.4 Adecuación de OpenFWWF

Con el fin de tener un controlador para el *chip* Broadcom del enrutador WRT54GL que permita realizar modificaciones a nivel MAC se utilizó el *firmware* OpenFWWF.

Para compilar el código es necesario contar con las herramientas necesarias como una distribución Linux y el paquete *b43-tools* que son un conjunto de herramientas para el *chip* WLAN de Broadcom de la serie 43XX. Una vez compilado OpenFWWF se cuenta con una versión ejecutable del *firmware* compatible con cualquier dispositivo que cuente con un *chip* Broadcom de la serie 43XX.

Para descargar el paquete *b43-tools* se ejecuta la siguiente orden en la consola de Linux:

```
~#git clone http://git.bu3sch.de/git/b43-tools.git
```

Con ésta se consigue una copia de la utilidad. Después es necesario ingresar al directorio que se acaba de descargar y dentro de la carpeta */b43-tools/assembler* ejecutar el comando make:

```
assembler/~# make
```

Esta instrucción instala las herramientas encargadas de compilar el código del controlador escrito en lenguaje ensamblador para un procesador MIPS en instrucciones de código máquina.

Al contar con la herramienta dentro de los repositorios del sistema se puede descargar el *firmware* OpenFWWF directamente de los servidores de los desarrolladores en el directorio deseado:

```
~# wget http://www.ing.unibs.it/openfwwf/firmware/openfwwf-5.2.tar.gz
```

Para tener acceso a la estructura de archivos que conforman el código fuente del controlador es necesario descomprimirlo con la siguiente instrucción:

```
~#tar -xzvf  openfwwf-5.2.tar.gz
```

El directorio *openfwwf-5.2* está compuesto por los siguientes archivos: *cond.inc*, COPYING, *initvals.asm, initvals.inc*, LICENSE, *Makefile, myreg.inc, shm.inc, spr.inc, ucode.asm.*

Para compilar el controlador es necesario estar dentro del directorio *openfwwf-5.2* y ejecutar *make*, como se ilustra en la figura 2.10. En caso de que no sea posible compilar el *firmware*, se deben descargar los repositorios faltantes, asegurarse de que *b43-tools* está instalado y que las modificaciones realizadas al código fuente sean coherentes y sigan la sintaxis adecuada.

```
~/openfwwf-5.2$ make

      b43-asm ucode5.asm ucode5.fw --cpp-args -DDEBUG=1 -- --
      ivalext .fw --psize
        Assembler info:
        Using start address 0
        ucode5.fw:  text = 1190 instructions (9520 bytes)
        b0g0initvals5.fw:  177 values (968 bytes)
        b0g0bsinitvals5.fw:  31 values (150 bytes)
```

Figura 2.10. Compilación del *firmware* OpenFWWF.

Una vez se obtenga una respuesta exitosa de la compilación se debe incluir el resultado en un archivo llamado `/trunk/dl/openfwwf-5.1.tar.gz`, de esta manera la compilación queda incluida en los paquetes a instalar de OpenWrt. La versión incluida en OpenWrt es la 5.1 y todos los *makefiles* y archivos relacionados para la compilación se encuentran relacionados con ese nombre de archivo, por esta razón, y para facilitar el proceso de inclusión de la última versión que brinda mayor estabilidad de funcionamiento en los dispositivos, es conveniente reemplazar el contenido del archivo comprimido ya existente.

2.4.5 Conjunto de Instrucciones de OpenFWWF

Al ser OpenFWWF un *firmware* para *chips* Broadcom basados en la arquitectura MIPS, trabaja con el tipo de instrucciones correspondientes a ese tipo de procesadores. Las instrucciones a las cuales responde el procesador se encuentran divididas según el tipo de operación que ejecuta, tales como: aritméticas y lógicas, de transferencia de datos, de rotación y desplazamiento, de ramificación, entre otras, con las

cuales se va a programar el nuevo mecanismo de reconocimiento. Éstas se muestran en la Tabla 2.1, Tabla 2.2, Tabla 2.3, Tabla 2.4, Tabla 2.5 y Tabla 2.6 respectivamente [34]. Es de resaltar que la complejidad de las instrucciones para programar un nuevo mecanismo de reconocimiento se suma a la dificultad de programar en lenguaje ensamblador.

Tabla 2.2. Conjunto de Instrucciones Aritméticas y Lógicas.

Comando	Operandos	Descripción	Operación
Add	A, B, rD	Suma	rD=A+B
add.	A, B, rD	Suma, guarda acarreo (*carry*)[49]	rD=A+B, almacena carry
Addc	A, B, rD	Suma con el *carry*	rD=A+B+*carry*
addc.	A, B, rD	Suma con el *carry, almacena carry*	rD=A+B+C, almacena *carry*
Sub	A, B, rD	Resta	rD=A-B
sub.	A, B, rD	Resta, guarda *carry*	rD=A-B
Subc	A, B, rD	Resta con el carry	rD=A-B-*carry*
subc.	A, B, rD	Resta con el carry, almacena carry	rD=A-B-C, almacena *carry*

Tabla 2.3. Conjunto de Instrucciones de Transferencia de Datos.

Comando	Operandos	Descripción	Operación
Mov	A,rD	Copia datos	rD=A
Tkiph	A,rD	TKIP S-Box búsqueda nivel alto	rD=SBOX[hi8(A)]
tkiphs	A,rD	TKIP S-Box búsqueda nivel alto con permutación de bytes	rD=byteswap(tkiph)
Tkipl	A,rD	TKIP S-Box búsqueda nivel bajo	rD=SBOX[lo8(A)]
Tkipls	A,rD	TKIP S-Box búsqueda nivel bajo con permutación de bytes	rD=byteswap(tkipl)

Tabla 2.4. Conjunto de Instrucciones de Rotación y Desplazamiento.

Comando	Operandos	Descripción	Operación
Sra	A, B,rD	Rotación Aritmética a la derecha	rD=A>>B
Or	A, B,rD	Operación OR	rD=A|B
And	A, B,rD	Operación AND	rD=A&B
Xor	A, B,rD	Operación OR	rD=A^B
Sr	A, B,rD	Rotación a la derecha	rD=A>>B
Sl	A, B,rD	Rotación a la izquierda	rD=A<<B
Srx	M, S, A, B,rD	Rotación a la derecha Extendida	
Rl	A, B,rD	Rotar a la izquierda	rD=lrot(A, B bits)
Rr	A, B,rD	Rotar a la derecha	rD=rrot(A, B bits)
Nand	A, B,rD	Limpiar bits	rD=A&(~B)
Orx	M, S, A, B,rD	OR exclusiva	

[49] *Carry* en matemática elemental es un dígito que ha sido transferido de una columna de dígitos a otra columna de dígitos de mayor potencia en un algoritmo de cálculo. Se acostumbra a usar el término en inglés.

Tabla 2.5. Conjunto de Instrucciones de Ramificación.

Comando	Operandos	Descripción	Operación
Jmp	L	Salto incondicional	PC=l
Jand	A, B, l	Salta si el binario AND	if (A&B)PC=l
Jnand	A, B, l	Salta si el binario NAND	if (!(A&B)) PC=l
js	A, B, l	Salta si todos los bits son 1s	if ((A&B)==A) PC=l
jns	A, B, l	Salta si no todos los bits son 1s	if ((A&B)!=A) PC=l
je	A, B, l	Salta si son iguales	if (A==B) PC=l
jne	A, B, l	Salta si no son iguales	if (A!=B) PC=l
jls	A, B, l	Salta si es menor *	if (A<B) PC=l
jges	A, B, l	Salta si es mayor o igual *	if (A>=B) PC=l
jgs	A, B, l	Salta si es mayor*	if (A>B) PC=l
jles	A, B, l	Salta si es menor o igual *	if (A<=B) PC=l
jl	A, B, l	Salta si es menor	if (A<B) PC=l
jge	A, B, l	Salta si es mayor o igual	if (A>=B) PC=l
jg	A, B, l	Salta si es mayor	if (A>B) PC=l
jle	A, B, l	Salta si es menor o igual	if (A<=B) PC=l
call	lrX, l	Almacena PC, invoca función	lrX=PC ; PC=l
ret	lrX,lrY	Almacena PC, retorna a función	lrX=PC ; PC=lrY
jzx	M, S, A, B, l	Salta si cero después se sumar S+M	
jnzx	M, S, A, B, l	Salta si no cero después de sumar S+M	
jext	E, A, B, l	Salta si la condición externa es verdadera	if (E) PC=l
jnext	E, A, B, l	Salta si la condición externa es falsa	if (!E) PC=l

* Operaciones en las cuales se tiene en cuenta el signo.

Tabla 2.6. Otras Instrucciones.

Comando	Operandos	Descripción	Operación
nap	Ninguno	No hace nada durante un evento	

Tabla 2.7. Descripción de Operandos.

rD	Registro de Destino
A	Registro fuente o inmediato
B	Registro fuente o inmediato
M	Mascara (*mask*)
S	Cambio (*shift*)
E	Condición externa (*External Condition*)
L	Nombre de rama o dirección
lrX	Registro de Enlace (lr0 – lr3)*Link Register*
lrY	Registro de Enlace(lr0 – lr3)*Link Register*

2.4.5.1 **Direccionamiento *Offset***

A continuación se presenta información útil sobre objetos de memoria y registros con los cuales es necesario trabajar para manejar la transmisión y recepción de tramas.

Para acceder a los registros es necesario usar los registros *offset* que se encuentran en la dirección 0x600 más el registro *offset* como se muestra en la Tabla 2.7 [35], éste se encuentra divido en dos partes de dos bytes de acuerdo a las especificaciones dadas en [25].

Tabla 2.8. Registros.

Offset	Nombre	Descripción
0x0490	Registro BRC	Este registro se usa para fijar o borrar las condiciones almacenadas en el Registro 4[50].
0x0500	Control de TXE	Controla el motor de transmisión (TXE, *Transmission Engine*). El valor de este registro cambia de acuerdo a la trama que debe ser enviada.
0x0506	TXETimeout	Este registro contiene la información del `ACKTimeout` cuando una trama transmitida necesita ser reconocida. El registro tiene la siguiente configuración: • Bit 0x8000: indica que el valor del *timeout* es válida, es decir, que actualmente se encuentra en espera de un ACK. • Bits 0x7ff8: contienen el valor del `ACKTimeout` en microsegundos.
0x0508	TXE WM0	Usado para habilitar el byte WM0 en el TXE.
0x050A	TXE WM1	Usado para habilitar el byte WM1 en el TXE.
0x0540	Comando TXFIFO	Este registro se usa para indicarle al TXE qué cola debe ser usada y cuál operación debe ser realizada.
0x0568	Selección de TX	Usado para copiar información en el TXE.
0x068E	SPR_IFS_0x0E	Contiene el tiempo que el canal ha estado libre durante dos transmisiones.
0x0690	Estado de IFS	Estado de los IFS

2.4.5.2 **Registro de Condición 4**

Este registro de condición es controlado por el registro BRC y revisa el estado de ciertas condiciones para establecer los procesos a seguir. La Tabla 2.8 [36] muestra la composición de dicho registro.

[50] El Registro 4 se describe en la siguiente sección.

Tabla 2.9. Bits de Configuración Registro de Condición 4.

Bit	Nombre	Descripción
0	COND_NEED_BEACON	Se fija cuando se debe transmitir una trama *Beacon.*
1	COND_NEED_BEACON	Se fija cuando una trama recibida necesita una trama de respuesta (ACK, CTS).
2	COND_NEED_PROBE_RESP	Se fija cuando se debe transmitir una trama *Probe Response.*
3	COND_CONTENTION_PARAM_ MODIFIED	Cuando los parámetros de contención han sido modificados con una trama *Beacon*. Estos deben ser restaurados antes de realizar cualquier otra transmisión.
4	COND_MORE_FRAGMENT	Indica que la trama actual está compuesta por múltiples fragmentos.
5	COND_FRAME_BURST	Indica que la transmisión actual usa *frameburstsing*[51].
6	COND_REC_IN_PROGRESS	Se fija durante la recepción de una trama y se deshabilita cuando ésta termina.
7	COND_FRAME_NEED_ACK	Indica que la trama transmitida necesita ser reconocida, por ende hay que habilitar el `ACKTimeout`.
8	COND_TX_ERROR	Indica que ocurrió algún error con la transmisión.
9	COND_RX_ERROR	Indica que ocurrió algún error con la recepción.
10	COND_TX_MULTICAST_FRAME	Indica que la trama actual es una trama multicast.
11	COND_NEED_RTS	Indica que es necesario enviar una trama RTS.
12	COND_NEED_PROBE_RESP	Indica que es necesario enviar una trama Probe Response.
13	No usado	
14	No usado	
15	No usado	

2.4.5.3 **Motor de Modificación de Transmisión**

El *firmware* usa el Motor de Modificación de Transmisión (TME, *Transmit Modify Engine*) para modificar las tramas pendientes que están listas para ser transmitidas dentro del serializador. Hasta 64 bytes de la trama original pueden ser modificados. Por ejemplo, si se quiere modificar el campo *Dirección 1* del encabezado de 802.11, es necesario escribir un nuevo valor en "*TemplatefillValue*", fijar el bit correspondiente de "*Templatefillmask*" y habilitar las modificaciones con TXE_WMx [37].

Es importante recordar que TXE_WM0 + TXE_WM1 son de 32 bits de longitud (16 + 16). De manera que si se usa TXE_WM0=0x1 se habilita la modificación del byte 0 y 1. Si se quiere modificar únicamente el byte 0 será necesario usar la máscara correspondiente 0x00FF. La modificación del campo *Dirección 1* a AA:BB:CC:DD:EE:FF quedaría así:

SPR_TME_VAL10 = 0xBBAA
SPR_TME_VAL12 = 0xDDCC
SPR_TME_VAL14 = 0xFFEE
SPR_TME_MASK10 = 0xFFFF
SPR_TME_MASK12 = 0xFFFF

[51] Frame-Bursting: técnica de transmisión de 802.11 que incrementa el *throughput* de la conexión bajo ciertas condiciones, disminuyendo el encabezado asociado a la sesión inalámbrica y transmitiendo hasta tres paquetes en el intervalo de espera IFS.

SPR_TME_MASK14 = 0xFFFF
SPR_TXE_WM0 = 0x00E0 -> Se fijan los bits 5,6 y 7.

2.4.5.4 **Control TXE0**

El registro SPR_TXE0_CTL contiene las instrucciones necesarias para el TXE, y tiene diferentes valores para diferentes tramas. Este registro se usa, entre otras cosas, para fijar los diferentes IFS en el mecanismo de conexión que controla el acceso al canal para transmisión [37]. Por ejemplo, SPR_TXE0_CTL puede tener los siguientes valores:

- 0x4d95 para tramas *Beacon*, indica al TXE que use un PIFS.
- 0x4021 para tramas ACK, indica al TXE que use un SIFS.
- 0x4c1d para tramas de Datos, indica al TXE que use un DIFS.

2.4.5.5 **Comando FIFO**

El TXE puede ejecutar diferentes operaciones sobre las tramas en las colas FIFO. El número de la cola FIFO de la cual la trama debe ser seleccionada y la operación que debe ser ejecutada en esa trama, están almacenados en el registro Comando FIFO [37]. Algunas de las operaciones principales que puede ejecutar son:

- SPR_TXE0_FIFO_CMD = SHM_TXFCUR | 0x2000 : elimina la trama en la cola indicada en SHM_TXFCUR.
- SPR_TXE0_FIFO_CMD = SHM_TXFCUR | 0x4000 : transmite la trama en la cola indicada en SHM_TXFCUR.

2.4.5.6 **Selección de Transmisión**

Este registro le indica al TXE qué fuente de datos debe usar y donde los debe ubicar. Le permite al *firmware* transferir tramas, o partes de ellas, de una memoria a otra. Es especialmente útil cuando el *firmware* necesita analizar el encabezado de la trama o modificar una parte de ella gracias a que es posible indicar la longitud de los datos que tomará el TXE [37]. A continuación se muestran algunos datos de configuración:

- `0x1F00`: Copiar fuente a
- `0x0060`: Longitud.
- `0x001c`: Destino

2.5 PROCESOS DE TRANSMISIÓN Y RECEPCIÓN

En esta sección se presenta el comportamiento natural del *firmware* OpenFWWF durante los procesos de Transmisión (TX) y Recepción (RX). Esta información es útil para observar los cambios que ocurren en el estándar al usar diferentes mecanismos de reconocimiento.

2.5.1 Transmisión

En la Figura 2.11 [38] se presenta el diagrama en bloques relacionado a la transmisión de tramas. Cada bloque representa una etiqueta en la fuente, una revisión o una toma de decisión.

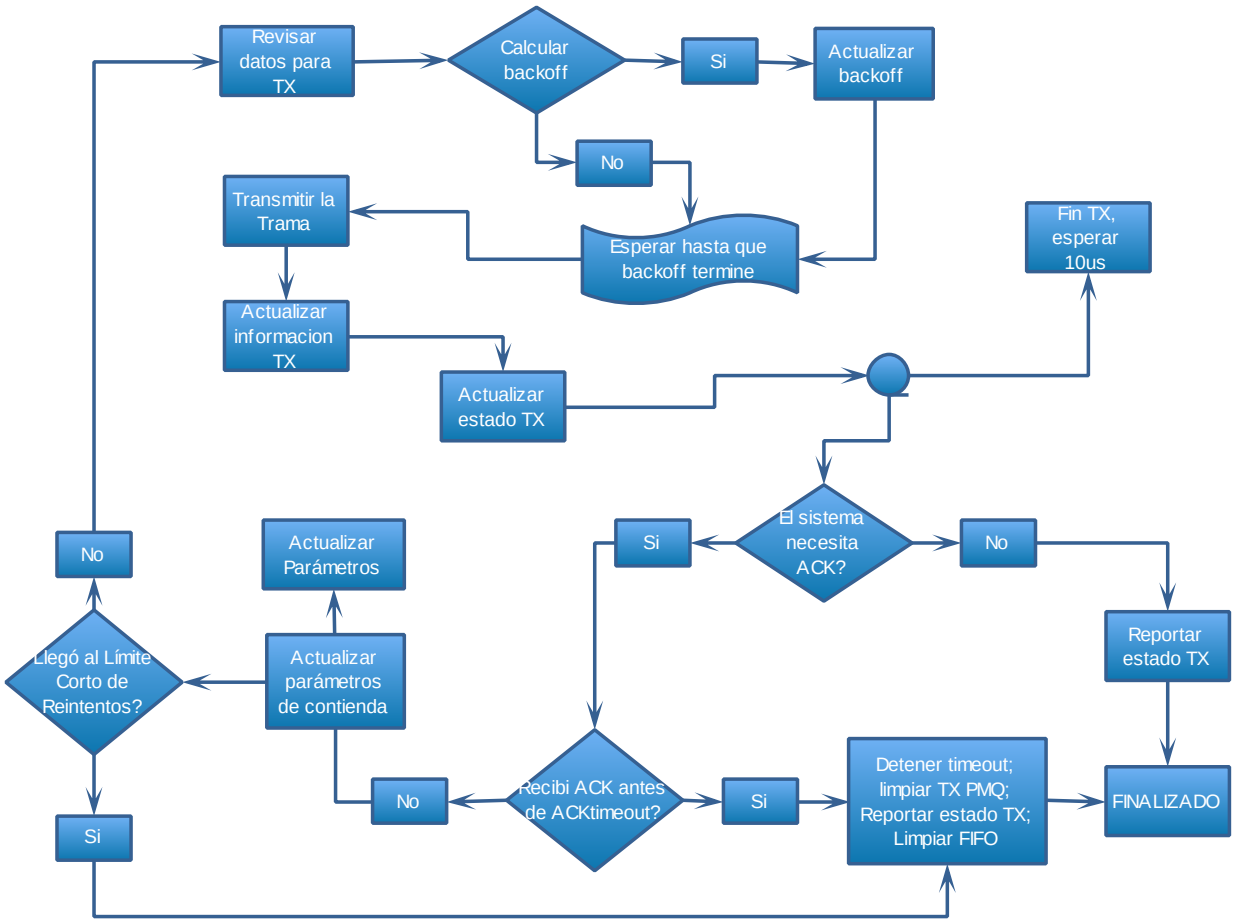


Figura 2.11. Proceso de Transmisión de OpenFWWF.

Comportamiento del *firmware*

La transmisión comienza cuando una trama se pone en la cola FIFO de OpenFWWF. Si el contador de tiempo *backoff* no ha empezado, se llamará a la función `set_backoff_time` para que calcule un nuevo *backoff*. El *firmware* esperará hasta que el tiempo *backoff* termine y llamará a la función `tx_frame_now`. Esta función toma la trama de la cola y la dirige al serializador, donde estará lista para ser transmitida por el canal si todas las condiciones indican que el canal está preparado para la transmisión. Si la trama necesita ser reconocida, entonces, se cargará el registro `SPR_TXE0_TIMEOUT` con el valor de *timeout* para la recepción de la trama ACK proveniente de la estación receptora, habilitando así el contador; si no se requiere reconocimiento, ninguna operación será llevada a cabo. Cuando la trama se envía, el procesador MAC configurará las condiciones para que la función `tx_infos_update` se ejecute, esta función actualiza la información de reintentos de transmisión. De aquí en adelante, se pueden presentar dos escenarios:

1. El primero se presenta cuando la trama no necesita reconocimiento: en este caso la función `report_tx_status_to_host` será llamada y el proceso de transmisión terminará.

2. La segunda posibilidad es que la trama recién enviada necesite reconocimiento: en este caso el *firmware* esperará hasta que el `ACKTimeout` termine o hasta que una trama ACK sea recibida. Si el `ACKTimeout` expira, se cancelará la función `contention_params_update`. Esta función fija el valor de la ventana actual y revisa si el *Short RetryLimit* ha sido alcanzado. Si el contador

Short Retry es igual al *Short RetryLimit*, entonces, se elimina la trama y el *firmware* le comunica al controlador que la transmisión falló. Si el contador *Short Retry* es menor que el *Short RetryLimit*, el *firmware* configurará la máquina de estados para realizar una retransmisión: se calcula un nuevo tiempo *backoff* y el proceso inicia otra vez. Si el reconocimiento se recibe dentro del intervalo del `ACKTimeout`, el *firmware* detiene el conteo y le informa al controlador que la transmisión fue exitosa. La función `tx_end_wait_10us` se ejecuta al terminar el proceso de transmisión para prevenir mediciones de ruido por cerca de 10µs (esta operación hace que el mecanismo CSMA/CA funcione apropiadamente) y retorna al estado inicial.

2.5.2 Proceso de Recepción

La figura 2.12 [39] presenta el diagrama en bloques de la parte de código concerniente a la recepción de tramas, cada bloque representa una etiqueta, un proceso de revisión o una toma de decisión.

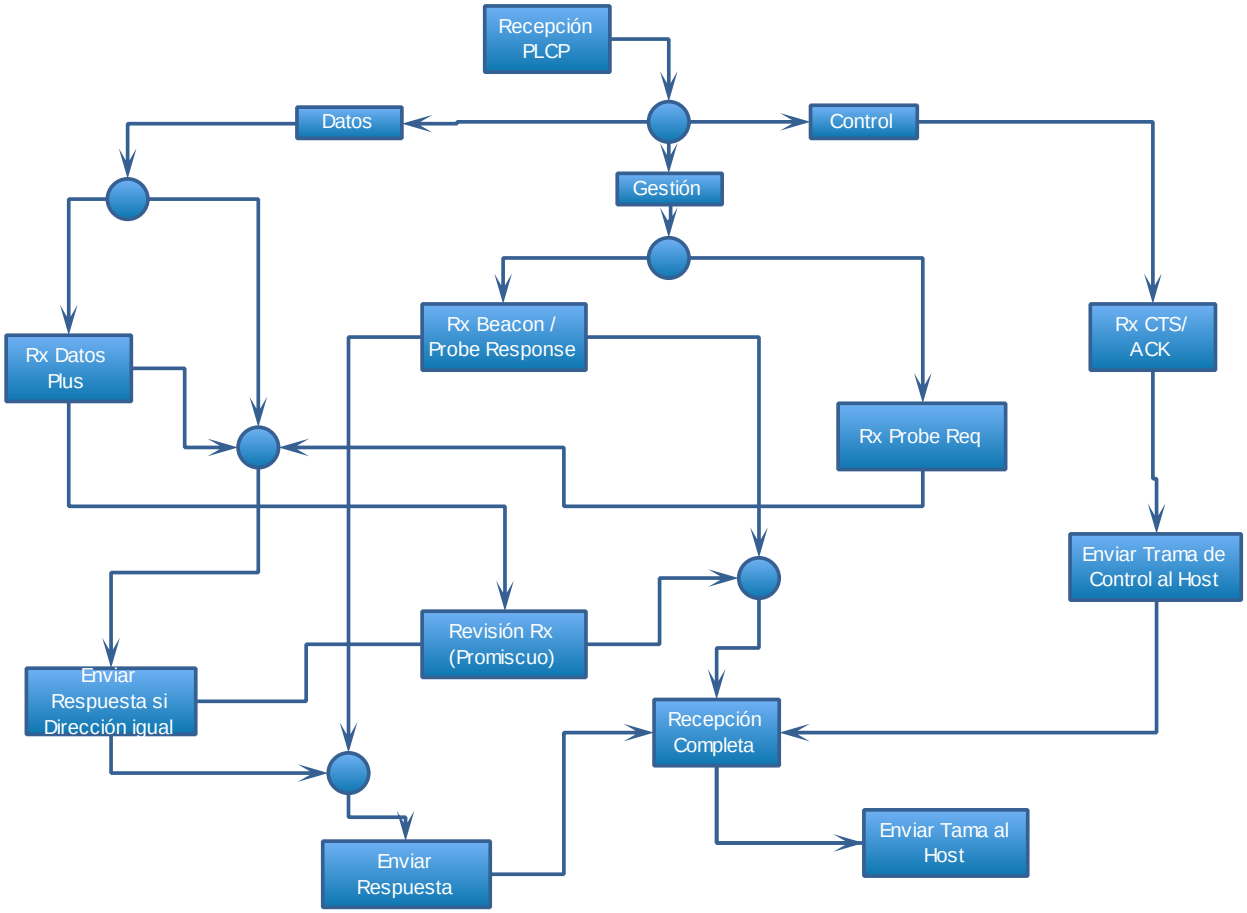


Figura 2.12. Proceso de Recepción de OpenFWWF.

El proceso de recepción comienza con la función `rx_plcp` que se ejecuta cuando se recibe una PLCP válida. Aquí, la trama se clasifica de acuerdo a su tipo: gestión, control o datos. Una vez el encabezado se recibe correctamente, es analizado: el *firmware* revisa que la longitud de la trama sea válida y que tenga una versión válida del protocolo, extrae información para actualizar el NAV, extrae el tipo de trama y busca carga útil.

Si la trama es una trama de control, se llama a la función `rx_ack` (actualmente el *firmware* no maneja tramas CTS o RTS). Esta es una función clave en el mecanismo de transmisión, `rx_ack`, de hecho

revisa la dirección MAC de destino de la trama ACK y si corresponde a la dirección de la estación que tiene el *firmware*, revisa si el sistema actualmente se encuentra esperando un reconocimiento. En este caso si el parámetro `EXPECTED_CTL_RESPONSE` corresponde al tipo de trama, entonces, la operación finalizara. El *firmware* detiene el `ACKTimeout` de manera que el sistema entiende que la transmisión terminó exitosamente; finalmente la condición `COND_TX_PMQ` se ignora para que `tx_contention_params_update` no sea llamado, a menos de que se haga explícitamente.

Otra función importante es `send_report`: cada vez que una trama que necesita reconocimiento se recibe, se ejecuta esta función. `send_response` prepara la trama de reconocimiento que será enviada de regreso a la estación origen.

`rx_beacon_probe_response` es fundamentalmente para sincronización, esta función, de hecho, extrae la información del tiempo de las tramas *Beacon* y *Probe Response* y ajusta el contador interno de la estación. `rx_complete` espera hasta que la recepción termina y, si todo ocurre satisfactoriamente, llama a las funciones `send_frame_to_host` si la trama necesita ser enviada al host[52] (al driver b43), o `discard_frame` si la trama no es válida. Finalmente, `send_frame_to_host` llama a `push_frame_into_fifo`, que copia la trama recibida a la cola de recepción para que el host procese la trama recibida a través del Acceso Directo a Memoria (DMA, *Direct Memory Access*).

2.6 PLATAFORMA DE DESARROLLO MAC

Una vez realizado el proceso descrito en las secciones anteriores y una vez comprobado que el *firmware* OpenFWWF compila en un PC con sistema operativo Linux, se dispone de una plataforma para desarrollar o modificar características del nivel de Control de Acceso al Medio MAC para dispositivos inalámbricos que posean una interfaz Broadcom de la familia 43XX, esto incluye PCs, portátiles, enrutadores y prácticamente cualquier dispositivo que cuente con un *chip* de esta familia y pueda correr una distribución Linux.

2.6.1 Puesta en Funcionamiento y Verificación

Con el proceso de compilación se generan 3 archivos: `code5.fw`, `b0g0bsinitvals5.fw` y `b0g0initvals5.fw`, los cuales contienen la información necesaria para ser introducida dentro del *chip* en su inicialización, pues todos los *chips* de Broadcom requieren que un *firmware* sea cargado dentro del controlador inalámbrico antes de empezar a funcionar. En este caso se usa OpenFWWF que maneja ahora gran parte de la funcionalidad de la interfaz inalámbrica en lugar del *software* propietario de Broadcom que viene por defecto.

Si se desea trabajar con el *firmware* original de Broadcom, éste debe ser extraído de los drivers propietarios de dicha empresa. Para esto, se debe descargar el controlador de un punto de distribución legal, seguidamente se debe extraer el *firmware* del driver de Broadcom[53] usando la herramienta *b43-fwcutter* o *bcm43xx-cutter* incluida en el paquete *b43-tools*, previamente descrito, e instalarlo en el directorio `/lib/firmware`. Pero si se desea modificar las capacidades del nivel MAC de 802.11 es necesario incluir el resultado de la compilación de OpenFWWF en este directorio, además de seguir los pasos que se describen a continuación según el tipo de dispositivo objetivo.

En un PC:

- Tras copiar el resultado de la compilación de OpenFWWF, es necesario desactivar el soporte de Calidad del Servicio, pues hasta ahora el *firmware* no tiene esas capacidades y simplemente bloquearía el sistema [40]:

```
modprobe qos=0
```

[52] Cada trama recibida se controla de manera que se pueda comprobar si la dirección de destino corresponde a la dirección de la estación.

[53] El firmware del driver binario de Broadcom Corporation tiene Copyright y no debe ser redistribuido.

En enrutadores [30]:

- Comprobar que el *Kernel* de Linux en el que se instaló el módulo es 2.6 con la instrucción `uname -a`.
- Verificar que el módulo se encuentre cargado, es decir, que al ejecutar la instrucción `lsmod` aparezca `b43`.
- Verificar si se tiene el *firmware* correcto con `ls -l /lib/firmware`. Si existe un directorio llamado `b43-open`, entonces, el *firmware* fue correctamente compilado, pero es necesario cambiarlo de directorio así:

```
cd /lib/firmware
ln -s b43-open b43
```

- Si es el caso de que se compila junto con `b43-legacy`, es necesario eliminar ese controlador con `rm /etc/modules.d/30-b43legacy` y sus archivos mediante `rm -rf b43-legacy`.
- Al igual que en un PC es necesario deshabilitar el soporte de QoS. En este caso, se modifica mediante `vi` el archivo `/etc/modules.d/30-b43` e incluir la sentencia `b43 qos=0`.
- Una vez realizados los procesos anteriores se puede inicializar la interfaz inalámbrica mediante `ifconfig wlan0 up`. Este proceso toma su tiempo debido a que el archivo `pcm5.fw` no se encuentra presente con el *firmware* abierto.
- Al reiniciar el dispositivo, se debe ejecutar `dmesg` para verificar que se inicializó openFWWF comprobando la sentencia `b43-phy0: Loading OpenSource firmware version 5.2.`

Tras verificar que el *firmware* abierto está correctamente instalado en el dispositivo a usar, se procede a configurar el comportamiento de sus interfaces y en general del enrutador o computador, para así comprobar que las características programadas sobre OpenWRT funcionen de manera adecuada.

2.6.2 Criterios de Diseño del Mecanismo Block ACK.

El mecanismo de reconocimiento Block ACK propuesto, se pensaba desarrollar bajo las siguientes consideraciones:

- Reconocimiento para un número fijo de tramas.
- No pretende implementar el mecanismo de Block ACK propuesto en 802.11e y tampoco brindar capacidades de QoS en un escenario de larga distancia.
- Se presume que brinda un mejor desempeño que el ACK estándar.
- Si algunas tramas no son correctamente recibidas es necesario enviar de nuevo todo el bloque.
- De acuerdo a los avances realizados, se intentaría identificar que trama o tramas son las faltantes para ser enviadas y reconocidas individualmente, de manera que no sea necesario enviar el bloque completo.

2.6.3 Diagramas en Bloque del Mecanismo Block ACK

Para un mejor entendimiento del mecanismo de reconocimiento que se quería implementar, se presenta a continuación el diagrama en bloques del funcionamiento general que se intentó llevar a la plataforma de desarrollo MAC. El funcionamiento presentado es básico. A partir de éste se realizarían las pruebas necesarias para determinar si era funcional y cómo determinar el número de tramas que conformarían un bloque de datos acorde a los escenarios de larga distancia.

2.6.3.1 **Transmisión con Block ACK**

La Figura 2.13 ilustra el proceso básico de transmisión haciendo uso del mecanismo de reconocimiento Block ACK.

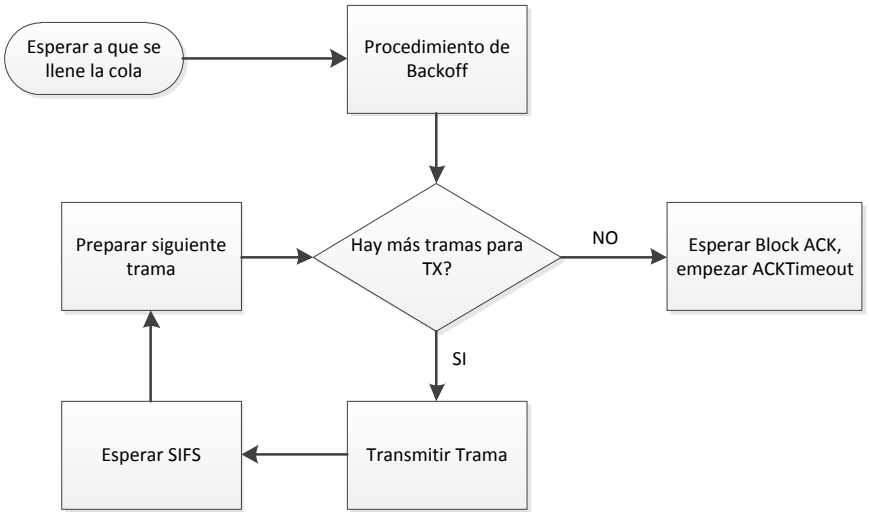


Figura 2.13. Diagrama de Bloques Proceso de Transmisión con Block ACK.

El proceso inicia en el momento en que se llena la cola de transmisión con X número de tramas, tras esto se inicia el procedimiento de *Backoff* para competir por el medio. Una vez se adquiere acceso al canal se transmiten trama a trama intercalado por un tiempo SIFS con el fin de dar tiempo suficiente en el transmisor para preparar la siguiente trama y en el receptor para procesar la trama actual. Cuando se transmite la última trama se entra en un estado de espera de Block ACK y se inicia el `ACKTimeout`[54], al igual que lo haría en el funcionamiento normal.

2.6.3.2 **Recepción con Block ACK**

La Figura 2.14 ilustra el proceso básico de recepción haciendo uso del mecanismo de reconocimiento Block ACK.

El proceso de recepción inicia al llegar la primera trama, a la cual se le verifica su integridad y pueden ocurrir dos cosas, si no ha sido correctamente recibida se descarta inmediatamente el bloque y se entra en un estado de espera de X periodos SIFS hasta que termine la transmisión del bloque de datos. Por otro lado, si la recepción fue exitosa, se guarda el estado de las tramas y se almacena en el *Buffer*, si no es la última trama se espera un tiempo SIFS para esperar la próxima trama y se repite el procedimiento. En el momento en que llega la última trama y todas han sido recibidas correctamente, se envía un mensaje Block ACK reconociéndole al transmisor la correcta recepción del bloque de datos recién enviados.

[54] Se esperaba que las pruebas realizadas con el mecanismo de reconocimiento ACK con `ACKTimeout` arrojara los valores adecuados a usar con el mecanismo Block ACK.

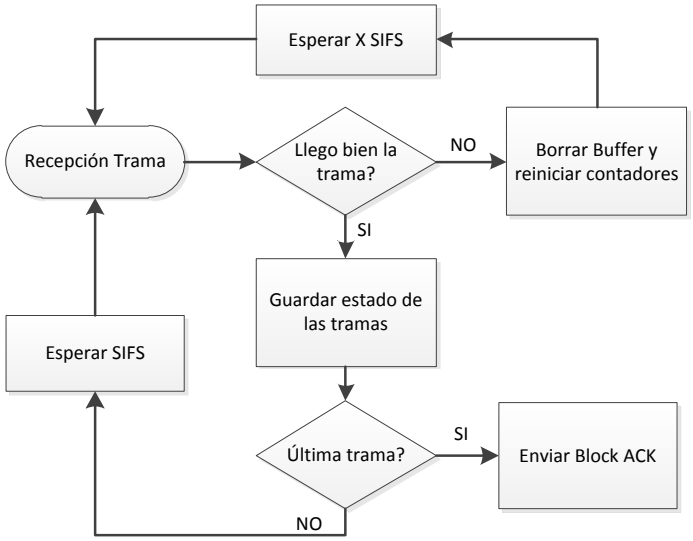


Figura 2.14. Diagrama de Bloques Proceso de Recepción con Block ACK.

2.6.4 Limitaciones.

Es importante tener en cuenta, que además de las limitaciones de OpenFWWF ya expuestas en 2.2.3, varios aspectos del comportamiento del nivel MAC no pueden ser alterados únicamente sobre este *firmware*, sino que deben trabajar en conjunto con los demás componentes del sistema en que trabaja. Tal es el caso del mecanismo de reconocimiento en bloque, `Block ACK`, propuesto en este Trabajo de Grado y cuyo desarrollo fue interrumpido tras un arduo trabajo investigativo. Dicho mecanismo debe ser desarrollado no solamente sobre el *firmware* controlador del *chip* Broadcom, sino que debe interactuar directamente con el *Kernel* del sistema Linux sobre el cual se implemente. Tal resultado se obtuvo tras la infructuosa realización de un módulo en código ensamblador sobre OpenFWWF y el posterior contacto con Francesco Gringoli, creador del firmware, quien informó de tal hecho. De manera que la continuación del trabajo implicaría el estudio y modificación del núcleo de Linux para que trabaje en conjunto con OpenFWWF y sea posible realizar un diseño detallado del mecanismo `Block ACK`.

También es importante resaltar el papel que tiene una plataforma de desarrollo tratada en el trascurso de este proyecto, ya que abre a la Facultad un nicho importante de estudio en las redes 802.11, permitiendo crear nuevos esquemas de reconocimiento, nuevas técnicas de acceso al medio, entre otros.

3 PARÁMETROS DE DESEMPEÑO Y HABILITACIÓN DE LOS MECANISMOS DE RECONOCIMIENTO

En la primera parte de este capítulo se presentan los principales parámetros que permiten evaluar el desempeño de un enlace y reflejan el comportamiento del mecanismo de reconocimiento de 802.11, dando cumplimiento al objetivo "*Determinar los parámetros necesarios para evaluar el desempeño de los mecanismos de reconocimiento",* propuesto en el anteproyecto de este Trabajo de Grado. También se exponen las herramientas que se usaron en el proceso de medición de desempeño y finalmente se presentan los procedimientos que se realizaron para habilitar los mecanismos de reconocimiento considerados.

3.1 PARÁMETROS DE DESEMPEÑO

El desempeño de una red WLAN es el indicador de qué tan productiva será la conexión inalámbrica de los usuarios, y por ende es la medida con la cual se determina si una solución es mejor que otra. Aspectos propios del estándar y aspectos típicos como el ambiente inalámbrico en el cual reside la red, el diseño del sistema y componentes de susbistemas como la antena, ganancia y potencia son características clave a la hora de evaluar el desempeño [41].

Los parámetros de desempeño que se evaluaron en este proyecto son *throughput*, *jitter*, porcentaje de paquetes perdidos, ACKs Duplicados y Retransmisiones.

3.1.1 Throughput

El *throughput* es definido como el número de bits que pueden ser transferidos entre el Punto de Acceso y el cliente en cierto periodo de tiempo. Típicamente se define como medida de la velocidad de transmisión de datos entregados exitosamente al receptor. Por lo tanto las unidades del *throughput* son bits/s o Bytes/s.

3.1.2 Jitter

El *jitter* es la medida de la variabilidad de la latencia de los paquetes a través de una red. Una red con latencia constante no tiene *jitter*. Éste se expresa en unidades de tiempo.

3.1.3 Porcentaje de Paquetes Perdidos

Es una relación entre el número de paquetes transmitidos y el número de paquetes perdidos.

La Tabla 3.1 [41] resume algunos de los factores clave que afectan el comportamiento de los parámetros anteriormente descritos. La mayoría de estos factores deben ser tomados en cuenta cuando se especifica el diseño de la WLAN cliente y la infraestructura del AP inalámbrico.

Tabla 3.1. Factores que Afectan el Desempeño.

Factor	Dependencias	Impacto
Interferencia	Otros dispositivos que ocupan el mismo espectro de frecuencia (hornos microondas, teléfonos inalámbricos, dispositivos Bluetooth, etc.)	Impacta la Relación Señal a Ruido (SNR, *Signal-to-Noise Ratio*). Mínima diferencia de potencia entre la señal que se desea recibir y el ruido.
Dispersión y Desvanecimiento Multitrayecto	Obstáculos	Incrementa los errores de transmisión debido a: • Potencia reducida en ciertas frecuencias (D*esvanecimiento Selectivo en Frecuencia*). • La diferencia de tiempo entre diferentes componentes recibidos hacen que la señal recibida se expanda en el dominio del tiempo (*Doppler Spread*).
Tipo de Antena	Ganancia de antena Diversidad de antena Longitud del cable de antena	La ganancia de antena es la relación de potencia con una antena isotrópica medida en dBi. La directividad de la antena concentra la potencia en cierta dirección. Disminuye las pérdidas producidas por el fenómeno por desvanecimiento multitrayecto especialmente para largas distancias. Las pérdidas en el cable dependen de la longitud y área del cable. Estas pérdidas degradan significativamente la señal y deben ser minimizadas.
Diseño del Sistema	Sensibilidad del receptor Potencia de transmisión Tiempo de cambio de modo	El Receptor tiene un umbral mínimo de potencia que la señal recibida debe tener para poder alcanzar cierta tasa de bits. Si la señal es menor que el umbral, la tasa máxima de bits debería ser disminuida, con un impacto directo en el rendimiento. Potencias más altas de transmisión provocan más interferencia, pero usualmente son limitadas por entidades regulatorias. Tiempo que toma cambiar entre los modos de transmisión y recepción.
Carga de la Red	Número de usuarios Mezcla de tráfico	El número de usuarios afecta el desempeño por ser el aire un medio compartido. Algunas aplicaciones pueden ser más demandantes en términos de ancho de banda que otras.
Ubicación	Distancia entre el AP y el cliente Orientación	A medida que la distancia entre el AP y el cliente aumenta, la velocidad de transmisión de datos disminuye. El desempeño puede ser afectado por la orientación del cliente dependiendo de la ubicación de la antena y también de la ubicación del AP.
Configuración AP y clientes	Aspectos propios de la tecnología	La configuración realizada sobre los dispositivos involucrados en el enlace varía el desempeño.

Dado que el escenario para cada usuario inalámbrico es diferente, no hay manera de predecir exactamente que desempeño obtendrá en determinado escenario.

La clave para desarrollar una prueba repetible es la definición a fondo del escenario de pruebas: Configuración del AP, ubicación relativa del AP a los clientes WLAN, aislamiento o efectos de la interferencia, etc. Adicionalmente, sería beneficioso poder variar la posición del cliente en situaciones controladas para establecer el impacto sobre las variables importantes del enlace.

Dado que los ejercicios de medición realizados no describen el comportamiento de un único sistema bajo diferentes condiciones sino que comparan varios sistemas bajo los mismos escenarios de transmisión, igual ganancia de antenas, potencia, distancia y otros, se caracterizaron los enlaces sobre los parámetros del sistema que determinaron aspectos variables entre uno y otro, es decir, los aspectos propios del funcionamiento de 802.11 dentro de sus variaciones consideradas en el desarrollo del proyecto, como los métodos de reconocimiento usados, las características de estos métodos y su influencia sobre la velocidad de transmisión de datos, *jitter,* y porcentaje de paquetes perdidos.

3.2 HERRAMIENTAS PARA EVALUAR EL DESEMPEÑO

Con el fin de medir los aspectos característicos de la red bajo las diferentes opciones de ACK, fue necesario usar las herramientas que se presentan a continuación:

3.2.1 Wireshark

Wireshark es un analizador de protocolos libre y de código abierto que captura directamente los paquetes de las interfaces de red e intenta desplegar su información lo más detalladamente posible. Está disponible para Windows y Linux y soporta los protocolos listados en [42], entre ellos IEEE802.11, permitiendo observar las características y contenido de los paquetes de WiFi, y así demostrar la presencia o no de los mensajes ACK del nivel de Enlace.

Además de esto, Wireshark permite analizar el tráfico de telefonía sobre una red, mostrando estadísticas sobre protocolos de señalización y flujo de datos. En este caso se usó para analizar el protocolo RTP con el cual se establecieron comunicaciones de voz y sobre las que se pudo observar el número de paquetes transmitidos, su número de secuencia, el *jitter* promedio y máximo, y el número de paquetes perdidos en cada conversación.

Wireshark puede ser usado en los siguientes casos:

- Para diagnosticar problemas de red.
- Para examinar problemas de seguridad.
- Para depurar implementaciones de protocolos.
- Para aprender y visualizar la estructura interna de los protocolos.

Un ejemplo de la interfaz y datos que despliega Wireshark se presenta en la Figura 3.1, donde se muestra una captura “en vivo”, es decir, directamente de la interfaz inalámbrica, en la que se puede observar flujos de paquetes en diversos protocolos, tales como TCP y HTTP, entre otros.

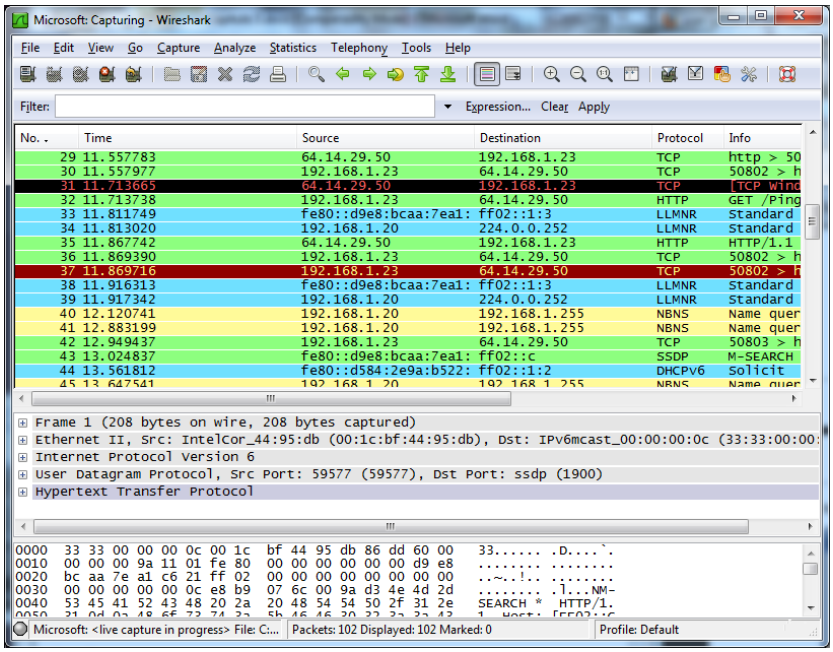


Figura 3.1. Captura de Wireshark.

3.2.2 Airmon-ng

Para capturar el tráfico perteneciente a una red inalámbrica es necesario habilitar la tarjeta inalámbrica en modo monitor o promiscuo, de manera que todos los paquetes 802.11 no correspondientes a la dirección física de la interfaz también sean capturados y así observar también el flujo propio del nivel de Enlace de Datos de WiFi, como son los mensajes *Beacon, Probe Request*, *Acknowledgment*, entre otros.

Con este fin existe la herramienta gratuita `Aircrack-ng`, que nace como un programa que permite recuperar claves WEP y WPA-PSK basado en los paquetes de datos de una WLAN, pero al pasar el tiempo se convirtió en un conjunto de herramientas para verificar redes inalámbricas [43]. Entre ellas se encuentra `Airmon-ng`, que es un script usado para habilitar el modo monitor en las interfaces de red inalámbricas. El modo de uso de este script en Linux es el siguiente [44]:

```
airmon-ng  < start | stop >  < interface >
```

La Figura 3.2 muestra el resultado que se obtuvo al ejecutar adecuadamente el comando `airmon-ng` durante el desarrollo del presente Trabajo de Grado.

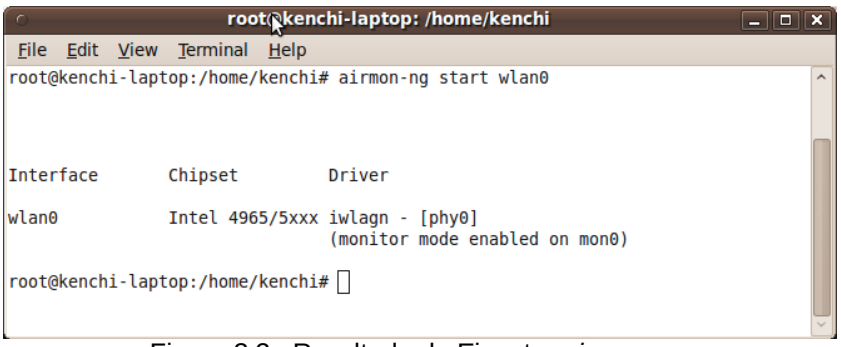


Figura 3.2. Resultado de Ejecutar *airmon-ng*.

3.2.3 Iperf

Iperf es una herramienta para Windows y Linux usada para probar el desempeño de una red mediante la emisión y recepción de flujos de datos TCP y UDP en el modelo cliente-servidor. En modo TCP permite medir el *throughput* de la red, mientras que en modo UDP permite evaluar aspectos de la calidad del enlace como *jitter* y pérdida de datagramas [45].

La Figura 3.3 muestra el funcionamiento general de Iperf, donde se encuentra un servidor a la espera de paquetes TCP/UDP y un cliente que transmite el flujo de datos TCP/UDP según los parámetros indicados en las instrucciones ejecutadas.

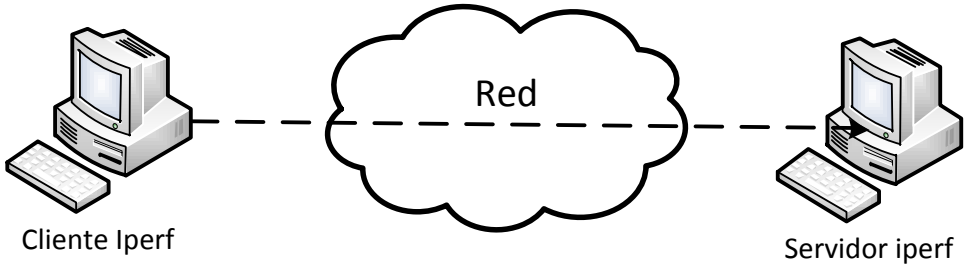


Figura 3.3. Esquema de Funcionamiento de *iperf*.

La configuración cliente/servidor más usual es la que se muestra en la Tabla 3.2 [45]:

Tabla 3.2. Configuración *iperf*.

MODO	Servidor	Cliente	Respuesta
TCP	iperf -s	iperf –c dir_ip_serv	*Throughput*
UDP	iperf -s	iperf –c dir_ip_serv -u	*Throughput, Jitter* y datagramas perdidos

Su operación se complementa con la información de la Tabla 3.3 [45]:

Tabla 3.3. Parámetros adicionales de *iperf*.

Argumento	Función
-b	Define el ancho de banda a usar en UDP.
-r	Realiza pruebas bidireccionales individuales.
-d	Realiza pruebas bidireccionales simultáneos.
-w	Define el tamaño de la ventana de TCP
-p, -t, -i	Define el puerto, intervalo y tiempo de la prueba
-f	Define el formato de presentación en Mbits, MBytes, Kbits, KBytes
-M	Define el tamaño máximo de los segmentos TCP
-P	Define el número de sesiones de cliente a ejecutar al mismo tiempo.

3.3 SOFTWARE PARA PRUEBAS

A continuación se describen las herramientas software que permitieron realizar pruebas de Voz sobre IP y de *streaming*[55] de video, con el fin de visualizar el desempeño del enlace bajo situaciones reales de uso.

[55] *Streaming*: es un término que denota contenido multimedia que es constantemente transmitido por un servidor multimedia.

3.3.1 Elastix

Elastix es una colección de productos y herramientas de código abierto que forman un Servidor de Comunicaciones Unificadas. Elastix integra las funciones de PBX (Asterisk), Fax (Hylafax), Mensajería instantánea (Openfire) y correo electrónico (Postfix) [46].

Este servidor funciona sobre CentOS pero su interfaz de control viene sobre WEB, de manera que el sistema de gestión es muy amigable. Para el desarrollo de este proyecto solo se usó como PBX para probar el funcionamiento de VoIP sobre enlaces de larga distancia bajo la influencia de diferentes configuraciones del mecanismo de reconocimiento de nivel de Enlace de 802.11. La Figura 3.4 muestra la página inicial de administración de Elastix.

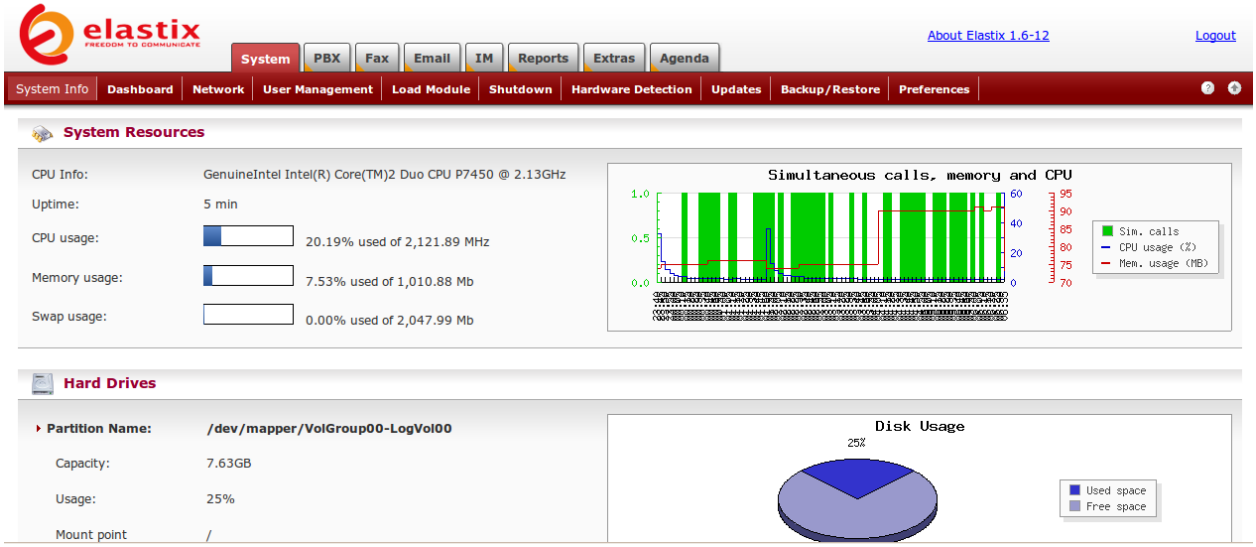


Figura 3.4. Página Inicial de Administración de Elastix.

3.3.2 X-Lite

Es un cliente *softphone*[56] perteneciente a la empresa CounterPath, que ofrece a los usuarios toda la productividad de un teléfono tradicional con las ventajas de una red IP. Fue diseñado para trabajar como un terminal VoIP para servidores de telefonía en una LAN empresarial o con proveedores de servicio de VoIP.

Características [47]:

- Señalización de las llamadas basada en Estándares Abiertos y en el Protocolo de Inicio de Sesión (SIP, *Session Initiation Protocol*).
- Teleconferencia y Videoconferencia.
- Mensajería instantánea.
- Cancelación acústica de eco, control automático de ganancia, detección de voz.
- Soporte de Marcación por Tonos (DTMF, *Dual-Tone Multi-Frequency*).
- Disponible para Windows, Linux y MAC.

[56] *Softphone*: término que se usa para describir al software usado para reemplazar a los tradicionales aparatos telefónicos en una red de Telefonía IP.

Aunque X-lite se encuentra disponible para Linux, su desarrollo para este sistema operativo se encuentra muy atrasado y los controladores de audio en equipos actuales no son los mejores. Por eso se recomienda usar este *softphone* preferiblemente en Windows.

En la Figura 3.5 se muestra la presentación general de X-lite 3.0 registrado como usuario 1 y listo para recibir y realizar llamadas.

Figura 3.5. Presentación general de X-lite.

3.3.3 Zoiper

Zoiper, al igual que X-Lite es un softphone gratuito, pero perteneciente a la compañía Attractel, que soporta tanto el protocolo de Intercambio entre Asterisk (IAX, *Inter-Asterisk eXchange*) como SIP. Zoiper es una aplicación de escritorio disponible para Windows, Mac OS X y Linux. Entre sus características se encuentra [48]:

- Soporte de protocolos SIP, IAX e IAX 2.
- Maneja hasta 6 líneas simultáneamente.
- Teleconferencia.
- Interfaz de Desarrollo de Programación.
- Integración con Thunderbird y Outlook.

Además, soporta el protocolo T.38 para Fax sobre IP, configuración de presencia en línea y completos ajustes de audio, que lo hacen ideal para trabajar sobre Linux, ya que este trae los controladores actualizados del sonido para ese sistema operativo. En la Figura 3.6 se muestra la imagen inicial de Zoiper para Linux registrado como usuario 2 y listo para realizar y recibir llamadas.

Figura 3.6. Zoiper 2.0 para Linux.

3.3.4 VLC

VLC Media Player es un reproductor multimedia de código abierto que funciona en Windows, MAC OS X, Linux, BSD, Solaris, entre otros. En su forma más básica, VLC es un reproductor liviano que puede reproducir casi cualquier formato de audio y video. Además de esto, puede trabajar como servidor de *streaming* y codificador de video, permitiendo emitir video a través de una red y si es necesario codificarlo en diferentes formatos de video. Una gran ventaja que tiene es la habilidad de reproducir contenido de video incompleto, con daños o aún en descarga [49].

La Figura 3.7 [49] muestra el comportamiento de VLC como servidor de *streaming* con o sin codificación y manejando muchas fuentes simultáneas, como servidor de Video en Demanda (VOD, *Video on Demand*), reproductor, y además presenta una interfaz web donde se puede configurar las listas de reproducción y otros parámetros que hacen de VLC una plataforma integral de video.

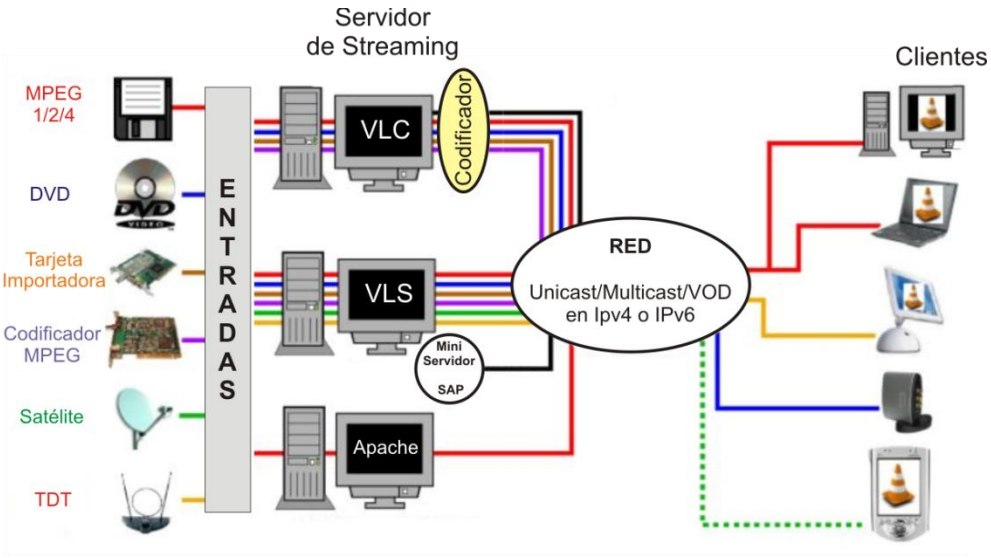


Figura 3.7. Esquema General de Funcionamiento de VLC.

Para finalizar esta sección, la tabla 3.4 presenta de forma resumida cómo estas herramientas permitieron evaluar el desempeño de los diferentes mecanismos de reconocimiento.

Tabla 3.4. Herramientas Software Utilizadas para Evaluar el Desempeño de los Mecanismos de Reconocimiento.

HERRAMIENTAS	APLICACIÓN
Wireshark	Permitió capturar y analizar los paquetes transmitidos desde y hacia la interfaz inalámbrica de los dispositivos 802.11.
Airmon-ng	Permitió habilitar la tarjeta inalámbrica (del PC) en modo monitor o promiscuo permitiéndole a Wireshark capturar todos los paquetes que viajan en el medio inalámbrico.
Iperf TCP	Permitió medir el *Throughput* del enlace para diferentes valores de `ACKTimeout` y para el caso de `No ACK`.
Iperf UDP	Permitió medir el *Jitter* y el Porcentaje de Datagramas Perdidos que presenta el enlace para diferentes valores de `ACKTimeout` y para el caso de `No ACK`.
Elastix	La PBX Elastix junto con los clientes *softphone* X-lite (Windows) y Zoiper (Linux) permitieron realizar una llamada telefónica entre los extremos del enlace. La herramienta Wireshark facilitó el análisis del Protocolo de Transporte en Tiempo Real (RTP, *Real-time Transport Protocol*), con el cual se establecieron las llamadas, debido a que muestra resultados de la conversación como el número de paquetes transmitidos, número de paquetes perdidos, número de secuencia, *Jitter* promedio y *Jitter* máximo.
X-Lite	
Zoiper	
VLC	La plataforma multimedia VLC permitió realizar un *streaming* de video a través del enlace. En este caso la herramienta Wireshark permitió medir el número de paquetes ACK duplicados, el número de retransmisiones y el número total de paquetes enviados.

3.4 CONFIGURACIÓN MECANISMOS DE RECONOCIMIENTO

En esta sección se presenta el proceso que se llevó a cabo para habilitar el mecanismo de `No ACK` y para variar el valor del `ACKTimeout`.

3.4.1 Ubiquiti PowerStation 2

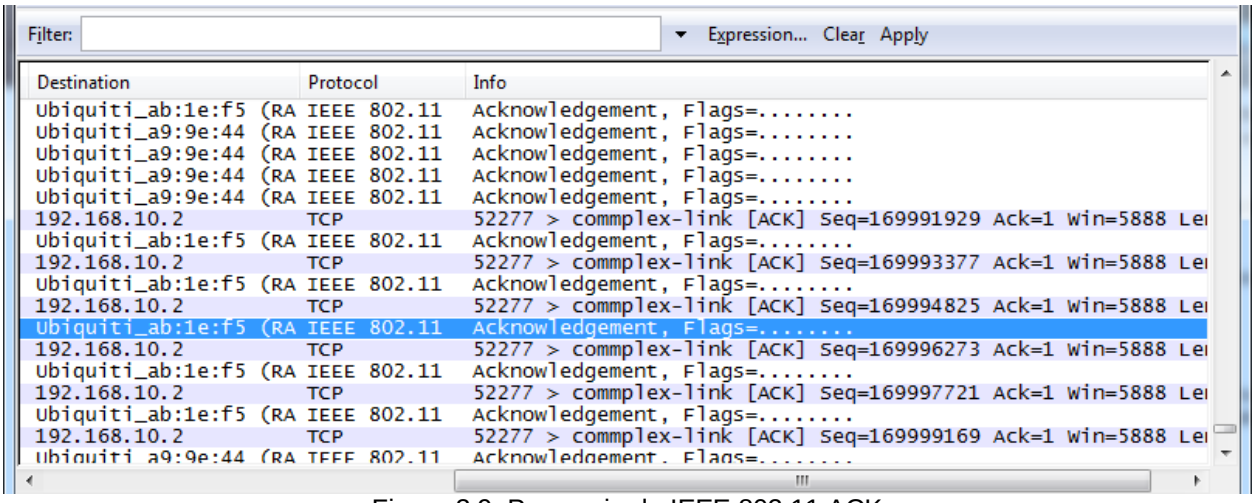
Dado que el mecanismo de reconocimiento de bloque se dejó de desarrollar, se optó por usar los equipos Ubiquiti PowerStation2 sobre los cuales se instaló el *firmware* OpenWrt. Dichos equipos facilitan por una parte, el proceso de instalación a la intemperie, pues, tanto radio como antena se encuentran sellados en un único armazón hermético y su alimentación es a través del puerto Ethernet (POE, *Power Over Ethernet*) [50]. Y por otra parte el proceso de configuración es más preciso que con los WRT54GL, además que son los equipos usados por el proyecto "*Análisis, Uso, Adecuación y Apropiación de Servicios sobre Tecnologías Inalámbricas en Zonas de Difícil Acceso de las Poblaciones Indígenas del Cauca Andino*".

Estos equipos están diseñados para realizar enlaces de larga distancia, pues, además de que la antena y el radio están integrados en un mismo dispositivo, la ganancia de la antena panel es de 17 dBi con la posibilidad de cambiar de polarización vía *software* si detecta gran cantidad de interferencia, característica que complementa con una potencia de transmisión de 400 mW y 2 interfaces Ethernet habilitadas con POE. El dispositivo PS2 se puede observar en la Figura 3.8 [50].

Figura 3.8. Ubiquiti PowerStation 2.

3.4.2 Activación de No ACK

La Figura 3.9 evidencia la presencia de mensajes ACK a nivel de Enlace de 802.11.



Filter: Expression... Clear Apply

Destination	Protocol	Info
Ubiquiti_ab:1e:f5 (RA	IEEE 802.11	Acknowledgement, Flags=.....
Ubiquiti_a9:9e:44 (RA	IEEE 802.11	Acknowledgement, Flags=.....
Ubiquiti_a9:9e:44 (RA	IEEE 802.11	Acknowledgement, Flags=.....
Ubiquiti_a9:9e:44 (RA	IEEE 802.11	Acknowledgement, Flags=.....
Ubiquiti_a9:9e:44 (RA	IEEE 802.11	Acknowledgement, Flags=.....
192.168.10.2	TCP	52277 > commplex-link [ACK] Seq=169991929 Ack=1 Win=5888 Le
Ubiquiti_ab:1e:f5 (RA	IEEE 802.11	Acknowledgement, Flags=.....
192.168.10.2	TCP	52277 > commplex-link [ACK] Seq=169993377 Ack=1 Win=5888 Le
Ubiquiti_ab:1e:f5 (RA	IEEE 802.11	Acknowledgement, Flags=.....
192.168.10.2	TCP	52277 > commplex-link [ACK] Seq=169994825 Ack=1 Win=5888 Le
Ubiquiti_ab:1e:f5 (RA	IEEE 802.11	Acknowledgement, Flags=.....
192.168.10.2	TCP	52277 > commplex-link [ACK] Seq=169996273 Ack=1 Win=5888 Le
Ubiquiti_ab:1e:f5 (RA	IEEE 802.11	Acknowledgement, Flags=.....
192.168.10.2	TCP	52277 > commplex-link [ACK] Seq=169997721 Ack=1 Win=5888 Le
Ubiquiti_ab:1e:f5 (RA	IEEE 802.11	Acknowledgement, Flags=.....
192.168.10.2	TCP	52277 > commplex-link [ACK] Seq=169999169 Ack=1 Win=5888 Le
Ubiquiti_a9:9e:44 (RA	IEEE 802.11	Acknowledgement, Flags=.....

Figura 3.9. Presencia de IEEE 802.11 ACK.

Para deshabilitar dichos mensajes de reconocimiento y anular las retransmisiones que se deberían generar por la no llegada de ACK, se llevó a cabo el siguiente proceso:

Se accedió a la consola de OpenWRT y se ejecutó la instrucción:

```
# sysctl -w /dev/ath0/maxretries=0
```
[51]

Este comando, deshabilita los reintentos de transmisión a nivel 2. Seguidamente se ejecutó:

```
# iwpriv ath0 noackpolicy X Y Z
```

Este comando fija la política de `No ACK` en los parámetros de WMM para el AP. El comando `noackpolicy` tiene 3 parámetros de configuración `X Y Z`. El primer parámetro es la Categoría de Acceso representado como se muestra en la Tabla 3.5 [52]. El segundo valor indica que el mecanismo de `No ACK` se usará para el conjunto de parámetros del AP. El tercer valor define si se habilita o no la regla de `No ACK`.

Tabla 3.5. Categorías de Acceso para WMM.

Categoría de Acceso	Descripción
0	Mejor Esfuerzo (BE - *Best Effort*)
1	Segundo Plano (BK - *Background*)
2	Video (VI - *Video*)
3	Voz (VO - *Voice*)

Por tanto, se ejecutó dicho parámetro 4 veces para habilitar `No ACK` en cada una de las categorías de acceso. Para asegurar la igualdad en el tratamiento de los datos, desde el software no se asignó ningún paquete a alguna categoría de acceso, así que todo el flujo de datos fue tratado como *Best Effort*, de la misma manera que se hace al no habilitar algún proceso de priorización de tráfico.

Una vez realizados los pasos anteriores, solo queda habilitar WMM:

```
# iwpriv ath0 wmm 1
```

Si se desea regresar al mecanismo de reconocimiento estándar simplemente se debe deshabilitar el soporte a WMM con `iwpriv ath0 wmm 0`.

La Figura 3.10 muestra el resultado de ejecutar el procedimiento anteriormente descrito, donde se evidencia la no presencia de los mensajes IEEE 802.11 ACK, pues, a pesar de que existen tramas IEEE 802.11 *Beacon*, ningún mensaje ACK responde a la llegada de las tramas de TCP.

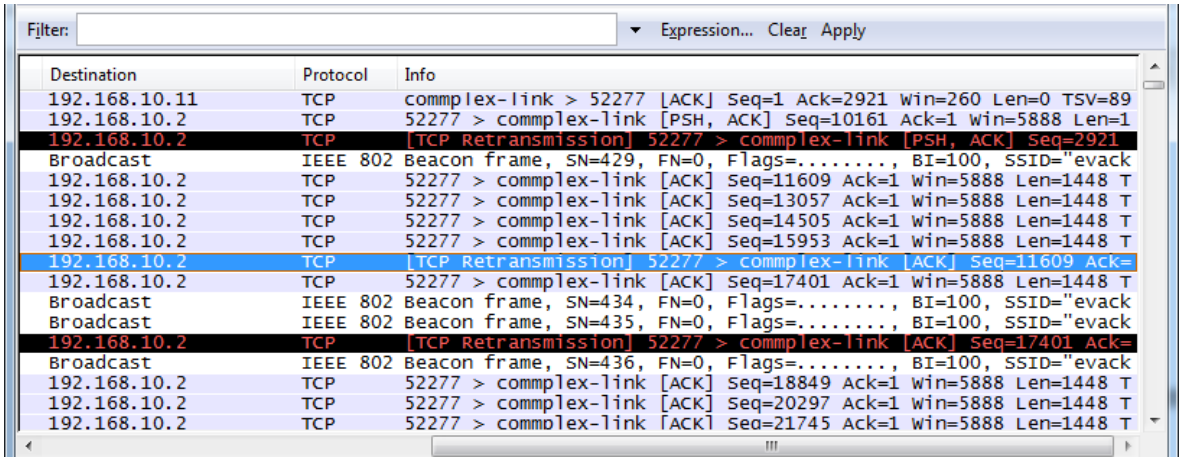


Figura 3.10. 802.11 con `No ACK`.

3.4.3 Variación de ACKTimeout

Para modificar el `ACKTimeout` fue necesario ejecutar la siguiente instrucción en la consola de OpenWRT:

```
#athctrl -d <distacia>
```

Donde la distancia se da en metros. Una vez ejecutada la instrucción, se pudo calcular el `ACKTimeout` correspondiente de la siguiente manera:

$$ACKTimeout = 21 + \frac{d}{150} \tag{4}$$

d es la distancia en metros y el resultado es en µs. El proceso anterior equivale a escribir el valor del `ACKTimeout` directamente en `/proc/sys/dev/wifi0/acktimeout`.

La ecuación (4) es una aproximación experimental del valor `ACKTimeout` dada por el *firmware*, ya que, dicho valor debe ser calculado teniendo en cuenta los intervalos de tiempo SIFS, `Slottime` y los retrasos generados por el procesamiento de las tramas junto con el tiempo de viaje de propagación en el aire.

Con los procedimientos anteriores fue posible cambiar la duración del `ACKTimeout` y también escoger el mecanismo `No ACK` permitiendo la evaluación de dichos mecanismos. De manera que el paso a seguir fue el correspondiente a realizar las pruebas de desempeño de los mecanismos de reconocimiento sobre los escenarios de prueba como se evidencia en el siguiente capítulo.

4 RESULTADOS, ANÁLISIS Y CONCLUSIONES DE LA EVALUACIÓN DE LOS MECANISMOS DE RECONOCIMIENTO COMPRENDIDOS EN EL TRABAJO DE GRADO

En la primera parte de este capítulo se detallan los escenarios donde se realizaron las pruebas al igual que la configuración de los equipos y de las herramientas. En la última sección de este capítulo se muestran todos los datos obtenidos durante la evaluación de los mecanismos de reconocimiento (ACK y `No ACK`) sobre dos enlaces punto a punto de larga distancia. Se analizan los resultados y las gráficas obtenidas para finalmente presentar una conclusión sobre el mejor mecanismo de reconocimiento para este tipo de enlaces y así dar culminación a este trabajo de grado.

En este capítulo se da cumplimiento al objetivo específico número 4 "*Determinar el mecanismo de reconocimiento más adecuado para enlaces de larga distancia mediante la comprobación práctica sobre enlaces pilotos"* propuesto en el anteproyecto de este trabajo de grado.

4.1 PLAN DE PRUEBAS

Después de desarrollar el contenido teórico de este documento, se plantea a continuación un plan de pruebas que encamina la evaluación de los mecanismos de reconocimiento propuestos en este trabajo:

- Determinación de los escenarios de pruebas.
- Pruebas de funcionamiento en enlaces de corta distancia.
- Ubicación de los equipos de red en el primer escenario de prueba.
- Toma de resultados para diferentes valores de `ACKTimeout` y para el mecanismo `No ACK` en el primer escenario de prueba.
- Ubicación de los equipos en el segundo escenario de prueba.
- Toma de resultados para diferentes valores de `ACKTimeout` y para el mecanismo `No ACK` en el segundo escenario de prueba.
- Análisis de resultados.
- Conclusiones.

4.2 CONFIGURACIÓN DE EQUIPOS Y ESCENARIOS DE PRUEBA

Para la evaluación de los mecanismos de reconocimiento fue necesario escoger unos escenarios de prueba. En esta sección del documento se presentan dichos escenarios junto con la configuración de red de los equipos.

4.2.1 Configuración de Equipos

La Figura 4.1 presenta la configuración de red que se utilizó en los escenarios de prueba.

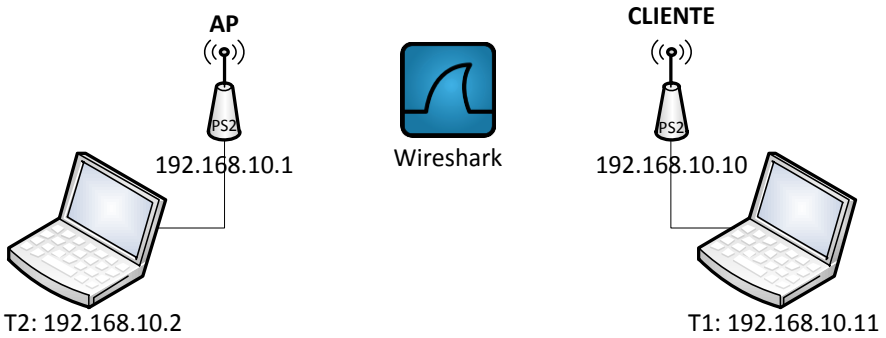


Figura 4.1. Configuración de Red de los Enlaces.

Con el fin de configurar los equipos según el esquema planteado, fue necesario seguir los siguientes pasos:

1. Configurar un PC con una dirección IP en la red `192.168.1.x` con máscara de red `255.255.255.0` para conectarse al enrutador.

2. Seguir los pasos descritos en la sección 2.2.3 para cambiar la contraseña por defecto y habilitar el acceso a través de SSH.

3. Acceder a la consola de configuración del enrutador con el comando `ssh -l root 192.168.1.1` e ingresar la contraseña seleccionada en el paso anterior.

Estos pasos se realizaron al iniciar el proceso de configuración tanto en el AP como en el cliente.

4.2.1.1 **Configuración del AP**

Para configurar el primer dispositivo como AP, se cambió la dirección IP de la interfaz *Ethernet* y seguidamente se configuró el comportamiento inalámbrico de la siguiente manera:

1. Para cambiar la dirección IP de la tarjeta *Ethernet* del dispositivo, se ejecutó `vi /etc/config/network` y se editó la configuración de la interfaz LAN como se muestra en la Figura 4.2.

```
#### LAN configuration
config interface lan
        option type     bridge
        option ifname   "eth0.0"
        option proto    static
        option ipaddr   192.168.10.1
         option netmask  255.255.255.0
```

Figura 4.2. Configuración Interfaz Ethernet del AP.

De manera que al AP se le asignó la dirección IP `192.168.10.1`.

2. Para cambiar la configuración inalámbrica se editó el archivo `/etc/config/wireless` de la manera que muestra la Figura 4.3.

```
config wifi-device ath0
        option type atheros
        option channel 11
        option txpower 26
        option disabled 0
        option agmode 11g

config wifi-iface
        option device   ath0
        option network  lan
        option mode     ap
        option ssid     evack
        option encryption none
```

Figura 4.3. Configuración Inalámbrica del AP.

Donde las opciones que se configuraron se describen en la Tabla 4.1.

Tabla 4.1. Configuración Inalámbrica del AP.

Parámetro (option)	Valor	Significado
Type	atheros	La tarjeta inalámbrica del PS2 es Atheros
Cannel	11	Canal escogido para la red
Txpower	26	Potencia de transmisión de 26 dBm
Disabled	0	Habilita la tarjeta inalámbrica
Agmode	11g	Fija el modo de operación del dispositivo en el modo 802.11g
Network	lan	Establece un puente entre la interfaz inalámbrica y la LAN
Mode	ap	Fija el enrutador en modo Punto de Acceso
SSID	evack	Fija el nombre de la red en “evack”
encryption	none	Establece la red como libre

Al crear el puente entre la interfaz inalámbrica y la LAN, la dirección IP se comparte entre los dos.

4.2.1.2 **Configuración del Cliente**

Para configurar el otro enrutador como cliente y con la dirección IP `192.168.10.10` fue necesario seguir los siguientes pasos:

1. Para cambiar la dirección IP de la tarjeta Ethernet, se ejecutó `vi /etc/config/network` y se editó la configuración de la interfaz LAN como se muestra en la Figura 4.4.

```
#### LAN configuration
config interface lan
        option type     bridge
        option ifname   "eth0.0"
        option proto    static
        option ipaddr   192.168.10.10
         option netmask  255.255.255.0
```

Figura 4.4. Configuración Interfaz Ethernet del Cliente.

2. Para cambiar la configuración inalámbrica y que el dispositivo funcione como estación, se editó el archivo `/etc/config/wireless` de la manera que se muestra en la Figura 4.5.

```
config wifi-device ath0
        option type atheros
        option channel 11
        option txpower 26
        option disabled 0
        option agmode 11g

config wifi-iface
        option device   ath0
        option network  wan
        option mode     sta
        option ssid     evack
        option encryption none
```

Figura 4.5. Configuración Inalámbrica del Cliente.

La opción ‘`mode sta`’ configura el enrutador como estación.

Tras reiniciar los equipos, la configuración que se acabó de realizar queda por defecto. Para verificar que los cambios han surtido efecto es necesario conectarse al cliente y ejecutar la instrucción `iwlist ath0 scan` en donde se debe visualizar la red que se acaba de crear como se muestra en la Figura 4.6.

```
ath0      Scan completed :
          Cell 01 - Address: 00:15:6D:A9:9E:44
                    ESSID:"evack"
                    Mode:Managed
                    Channel:11
                    Quality:5/5  Signal level:-42 dBm  Noise level:-91 dBm
                    Encryption key:off
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
                              9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s; 36 Mb/s
                              48 Mb/s; 54 Mb/s
```

Figura 4.6. Comprobación Existencia de la Red.

4.2.2 Escenarios de Prueba

Con el objetivo de evaluar el mecanismo de reconocimiento ACK y `No ACK` de manera práctica, se escogieron dos escenarios de prueba. Dado que la intención era evaluar dichos mecanismos en enlaces de larga distancia se tomó como punto de referencia la torre de telecomunicaciones de la Facultad de Ingeniería Electrónica y Telecomunicaciones (FIET) de la Universidad del Cauca y se buscaron dos puntos alejados de la FIET pero con línea de vista hacia la misma. El principal requerimiento para buscar estos lugares fue que la distancia entre el punto y la FIET superara en gran medida la distancia para la que el estándar 802.11 fue diseñado, es decir 300 m [53], y así crear un escenario de larga distancia.

Se encontró que lugares como el Tablazo (Zona de la Ciudad de Popayán) y el corregimiento Cajete del Municipio de Popayán eran puntos estratégicos para la realización de dichos enlaces.

4.2.2.1 **FIET – Tablazo**

Este enlace se realizó entre la torre de telecomunicaciones de la FIET y la antigua torre de la emisora de la Universidad del Cauca ubicada en la Zona del Tablazo de la ciudad de Popayán. Este escenario se escogió, después de comprobar que existe línea de vista entre los lugares involucrados y que la distancia entre los mismos genera un escenario de larga distancia. Ya que ambos sitios son propiedad de la Universidad del Cauca, se facilitó el acceso a los lugares y la seguridad de los dispositivos, factores que permitieron llevar a cabo las pruebas de manera satisfactoria. El perfil de este enlace de 2.7 Km realizado con Global Mapper se muestra en la Figura 4.7.

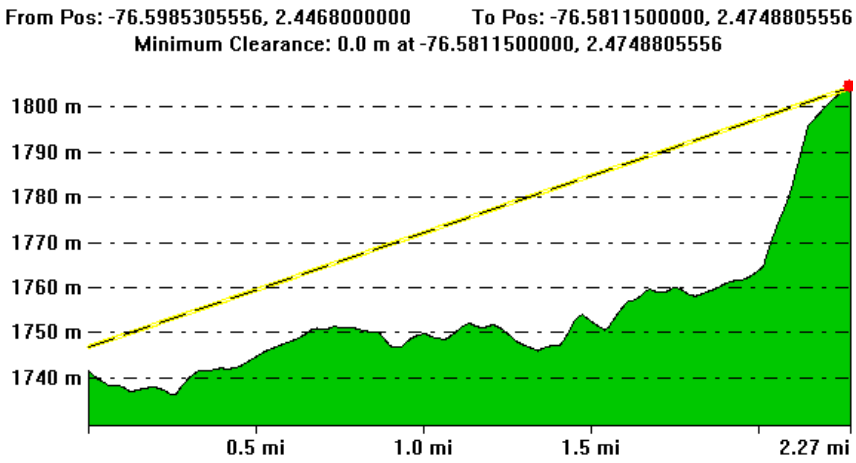


Figura 4.7. Perfil del Enlace FIET – Tablazo.

Para realizar este enlace, se decidió utilizar los dispositivos PowerStation 2 de Ubiquiti, los cuales se configuraron a su máxima potencia (26dBm) en el canal 11. Con la intención de que los enlaces fueran punto a punto se deshabilitó el DHCP de la interfaz inalámbrica y se activaron las restricciones de conexión a nivel MAC. Uno de los radios se instaló en la torre de telecomunicaciones de la FIET orientado hacia el Tablazo, se configuró como AP y se alimentó por POE desde el Centro de Cableado 1 (CC1) de la FIET ubicado en el SISE. El otro radio se configuró como cliente y se instaló en la torre del Tablazo. El enlace se ajustó hasta lograr un nivel de señal en el cliente de -46 dBm. La Tabla 4.2 especifica algunas características propias del enlace FIET – Tablazo.

Tabla 4.2. Características del Enlace FIET – Tablazo.

Distancia del Enlace	3.7Km
SSID del AP	Evack
Frecuencia	2462 MHz
Canal	11
Polarización	Vertical
Velocidad de Transmisión de Datos	54 Mbps
Velocidad de Recepción de Datos	54 Mbps
Nivel de Señal en el Cliente	-46 dBm

4.2.2.2 **FIET – Cajete**

Este enlace se realizó entre la torre de telecomunicaciones de la FIET y una casa del corregimiento Cajete del municipio de Popayán. Este escenario se escogió, después de comprobar que existe línea de vista entre los lugares involucrados y que la distancia entre los mismos genera un escenario de larga distancia, además, de que es uno de los puntos del Proyecto FRIDA. El perfil de este enlace de 6.6 Km realizado con Global Mapper se muestra en la Figura 4.8.

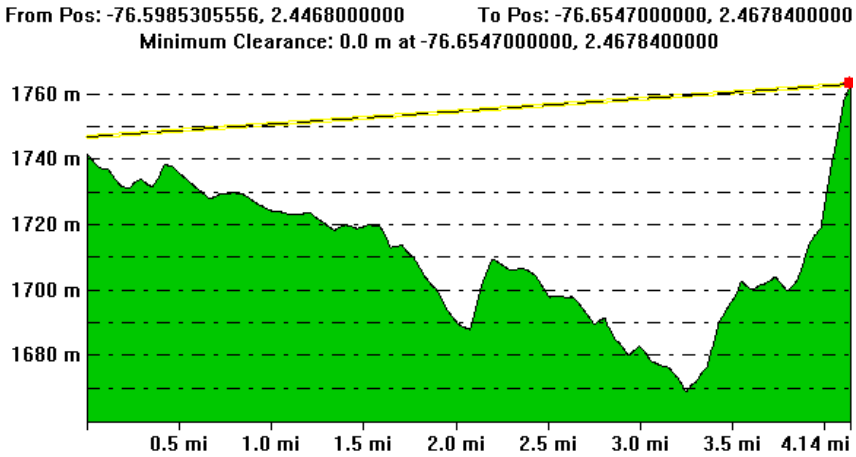


Figura 4.8. Perfil del Enlace FIET – Cajete.

Para realizar este enlace, se decidió utilizar los dispositivos PowerStation 2 de Ubiquiti, los cuales se configuraron a su máxima potencia (26dBm) en el canal 11. Con la intención de que los enlaces fueran punto a punto se deshabilitó el DHCP de la interfaz inalámbrica y se activaron las restricciones de conexión a nivel MAC. Para realizar este enlace, uno de los radios se instaló en la torre de telecomunicaciones de la FIET orientado hacia Cajete, se configuró como AP y se alimentó por POE desde el Centro de Cableado 1 (CC1) de la FIET ubicado en el SISE. El otro radio se configuró como cliente y se instaló en el techo de una casa en Cajete. El enlace se ajustó hasta lograr un nivel de señal en el cliente de -49dBm. La Tabla 4.3 especifica algunas características propias del enlace FIET – Cajete.

Tabla 4.3. Características del Enlace FIET – Cajete.

Distancia del Enlace	6.6Km
SSID del AP	evack
Frecuencia	2462 MHz
Canal	11
Polarización	Vertical
Velocidad de Transmisión de Datos	54 Mbps
Velocidad de Recepción de Datos	54 Mbps
Nivel de Señal en el Cliente	-49 dBm

Tras realizar las dos instalaciones, la torre de la FIET quedó con dos equipos Ubiquiti PowerStation2 como se muestra en la Figura 4.9. La antena inferior orientada hacia el Tablazo y la superior hacia Cajete.

Figura 4.9. Torre FIET.

4.3 CONFIGURACIÓN DE HERRAMIENTAS

Para medir el desempeño de los mecanismos de reconocimiento es necesario configurar las aplicaciones que se presentaron en la sección 3.2. Para ello es importante tener en cuenta que se decidió configurar la estación terminal T1 con el sistema operativo Microsoft Windows 7 y la dirección IP 192.168.10.2/24, la estación terminal T2 se configuró con Ubuntu 9.10 Karmic Koala y la dirección IP 192.168.10.11/24.

4.3.1 Iperf

Tanto en T1 como en T2 se instaló *iperf* de la siguiente manera:

- T1: se descargó la aplicación desde http://www.noc.ucf.edu/Tools/Iperf/iperf.exe, la cual se puede ejecutar sin necesidad de ser instalada. Es importante ejecutarla como administrador, pues si no se ejecuta de esta manera, el sistema operativo no permite que funcione en el modo servidor.

- T2: se instaló *iperf* mediante el comando sudo `apt-get install iperf`. Desde Linux no es necesario ejecutarla como *super usuario*.

La Tabla 4.4 muestra un ejemplo de la configuración que se usó durante las pruebas con *iperf*.

Tabla 4.4. Ejemplo Configuración *iperf*.

Servidor TCP	iperf -s -f m
Cliente TCP	iperf -c 192.198.10.2
Servidor UDP	iperf -s -u -f m
Cliente UDP	iperf -c 192.168.10.2 -u -f m -b 16

La Figura 4.10 muestra una captura de la ejecución de *iperf* como cliente TCP.

```
kenchi@kenchi-laptop:~$ iperf -c 192.168.10.2 -f m
------------------------------------------------------------
Client connecting to 192.168.10.2, TCP port 5001
TCP window size: 0.02 MByte (default)
------------------------------------------------------------
[3] local 192.168.10.11 port 43306 connected with 192.168.10.2 port 5001
[ID] Interval       Transfer     Bandwidth
[3]  0.0-10.0 sec  0.91 MBytes  0.76 Mbits/sec
```

Figura 4.10. Iperf TCP.

4.3.2 VLC

La Figura 4.11 muestra el esquema de red usado para las pruebas con VLC, donde se definió el equipo T2 como servidor de video *streaming,* para lo cual se ejecuta la siguiente instrucción desde la consola de Linux:

```
vlc -vvv ubicación_del_video --sout '#standard {access=http, mux=ogg,
dst=192.168.1.11:8080}'
```

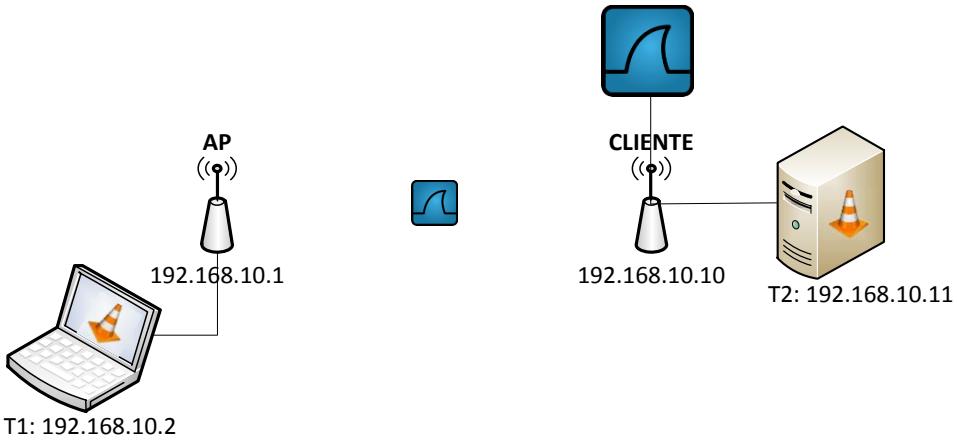


Figura 4.11. Esquema general de las pruebas con VLC.

Se escogió empaquetar los datos de video en tramas HTTP con el objetivo de realizar el flujo sobre TCP y no se hizo ninguna codificación de video para que la velocidad de transferencia de datos fuera alta.

Para reproducir el video, por parte de T1 solo fue necesario ejecutar VLC y seguir los siguientes menús: `Media> Open Network Stream`, en donde se escogió el protocolo HTTP y se escribió la dirección del servidor, 192.168.10.11 con el puerto 8080 así:

```
http://192.168.10.11:8080
```

Al presionar '*play*', empezó a cargar y reproducir el video de la red, como se evidencia en la Figura 4.12.

Figura 4.12. Reproducción video *streaming*.

4.3.3 Wireshark

El analizador de protocolos *wireshark* se usó en dos lugares de observación. El primero fue para capturar los paquetes de la red 802.11 y efectivamente comprobar el tráfico de paquetes de IEEE 802.11, especialmente los ACK y la ausencia de los mismos, y el segundo fue para capturar los paquetes de video y voz que intercambiaron los equipos terminales.

Con el fin de capturar los paquetes fue necesario habilitar el modo monitor de la interfaz inalámbrica desde Linux. Esto se realizó con el comando `sudo airmon-ng start wlan0`. Seguidamente se lanzó *wireshark* como *super usuario*: `sudo wireshark`, pues solo de esta manera el sistema permite usar todas las interfaces de red, entre ellas `mon0` como se muestra en la Figura 4.13, la cual debe ser seleccionada.

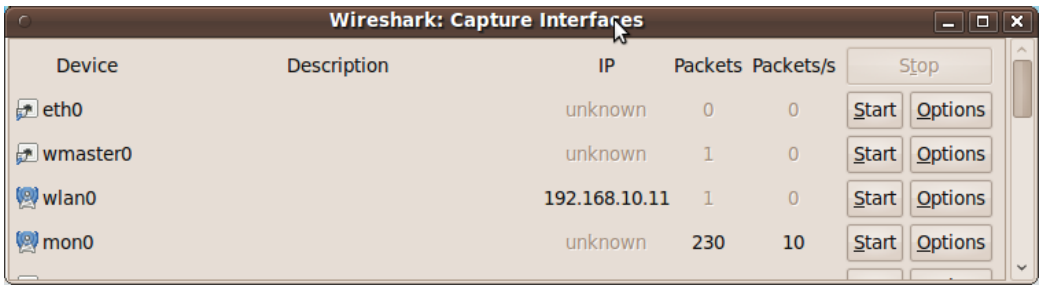


Figura 4.13. Interfaces de Wireshark.

Para capturar los paquetes que llegaban a la interfaz Ethernet solo fue necesario seleccionar como fuente dicha interfaz, la eth0.

4.3.4 Elastix

Elastix se usó como servidor de VoIP para establecer un flujo de llamadas entre los equipos terminales. Después de instalarlo se accedió a él por consola como *super usuario* con la contraseña seleccionada y se editó el archivo `/etc/sysconfig/network-scripts/ifcfg-eth0` con el fin de fijar la dirección IP en 192.168.10.20. Una vez hecho esto se accedió a su página web en la dirección http://192.168.10.20 donde se usó el usuario *admin* y la contraseña '*palosanto*' que está por defecto.

Desde la interfaz web se pueden observar todas las opciones de configuración que tiene este servidor de comunicaciones, y directamente en la pestaña PBX se pueden crear las extensiones SIP. Para esto se seleccionó agregar dispositivo SIP y se crearon las extensiones 1 y 2 como muestra la Figura 4.14.

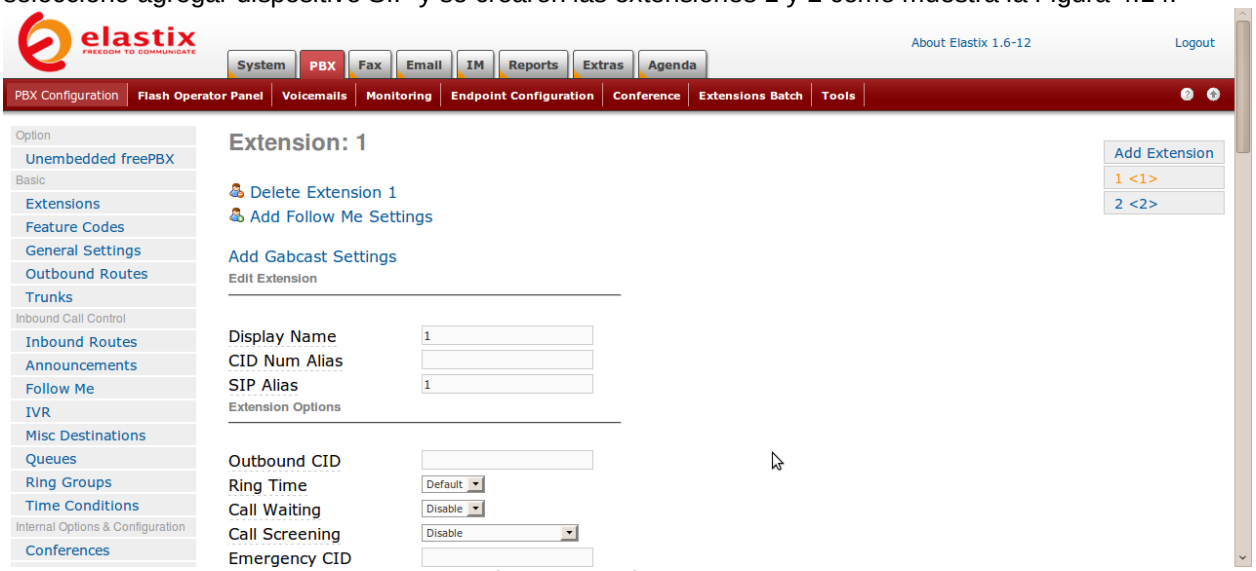


Figura 4.14. Menú de Creación de Extensiones en Elastix.

4.3.5 X-Lite y Zoiper

X-Lite en T1 y Zoiper en T2 funcionaron como las terminales VoIP de Elastix, para lo cual fue necesario configurar una cuenta SIP en ambos *softphones* con la información suministrada en el paso anterior y se resume en la Tabla 4.5.

Tabla 4.5. Configuración *Softphones*..

Parámetro SIP	Terminal 1	Terminal 2
Nombre desplegado	1	2
Usuario	1	2
Password	1	2
Dominio/Proxy	192.168.10.20	192.168.10.20

Tras la configuración de las herramientas, el escenario de pruebas quedó como se presenta en la Figura 54.

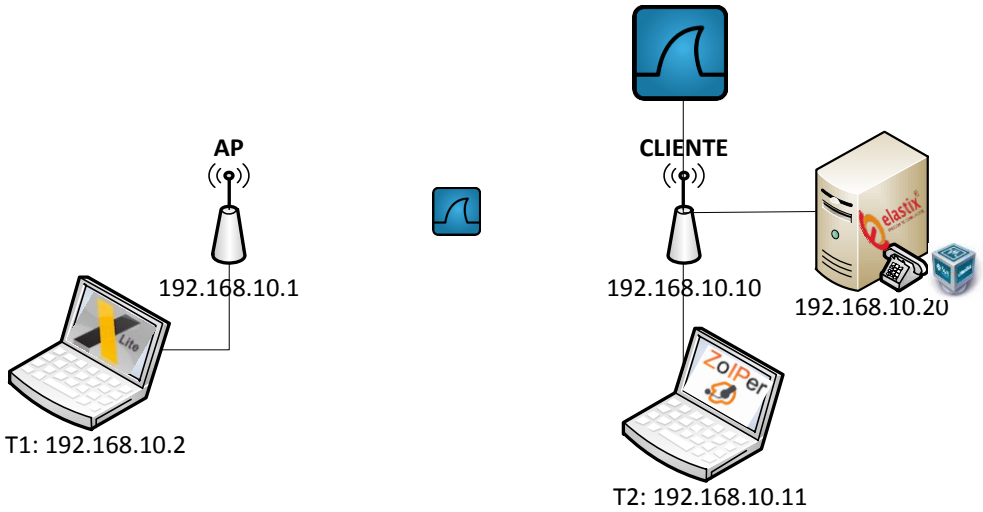


Figura 4.15. Esquema General de Pruebas con Elastix.

4.4 RESULTADOS Y ANÁLISIS DE LA EVALUACIÓN PRÁCTICA DE LOS MECANISMOS DE RECONOCIMIENTO

Para evaluar el desempeño del mecanismo de reconocimiento ACK primero se realizaron pruebas con *iperf* usando los protocolos TCP y UDP, se realizó una llamada telefónica a través de la PBX Elastix y se transmitió un video utilizando la herramienta VLC mientras se modificaba el `ACKTimeout` de los radios. Las mismas pruebas se realizaron para el mecanismo `No ACK`.

Con el objetivo de obtener datos más precisos se realizó 5 veces cada prueba bajo las mismas condiciones y luego se calculó el promedio de los datos obtenidos. Cabe resaltar que tanto las tablas como las gráficas fueron realizadas tomando los valores promedio y que en el anexo A se encuentran las tablas con la totalidad de resultados obtenidos.

4.4.1 Enlace FIET – Tablazo

Para el enlace FIET-Tablazo se analizaron los resultados de las pruebas realizadas para `ACKTimeouts` anteriores, cercanos y posteriores al correspondiente a la distancia entre los dos puntos involucrados (3.7 Km). Esta relación se encuentra en la Tabla 4.6 y se calculó a partir de la ecuación (4).

Tabla 4.6. Equivalencia entre distancia y `ACKTimeout`.

Distancia (m)	0	322	1609	3218	3379	3540	3701	3862	4827	6436	8045	9654
ACKTimeout (µs)	21	23	32	42	44	45	46	47	53	64	75	85

4.4.1.1 Evaluación del Desempeño del Mecanismo de Reconocimiento ACK.

- **Evaluación del Mecanismo ACK utilizando la herramienta IPERF sobre el protocolo TCP**

Las Tablas 4.7 y 4.8 contienen los valores promedio del *throughput* del enlace medido desde el AP hacia el cliente y desde el cliente hacia el AP respectivamente para los valores de `ACKTimeout` consignados en la Tabla 25.

Tabla 4.7. *Throughput* Promedio Medido desde el AP hacia el Cliente (Mbps).

ACKTimeout (µs)	21	23	32	42	44	45	46	47	53	64	75	85
Throughput (Mbps)	0.67	0.75	0.67	14.18	13.86	15.14	14.84	14.48	16	16.42	15.28	15.96

Tabla 4.8. *Throughput* Promedio Medido desde el Cliente hacia el AP (Mbps).

ACKTimeout (µs)	21	23	32	42	44	45	46	47	53	64	75	85
Throughput (Mbps)	0.55	0.74	0.67	16.44	19.06	19.22	14.98	18.08	18.62	16.66	15.32	14.96

Del resultado de esta prueba se observó que el *throughput* del enlace, tanto en un sentido como en el otro, aumenta al acercarse al valor del `ACKTimeout` calculado para la distancia del enlace hasta un máximo de 19.22 Mbps. A medida que sigue aumentando el `ACKTimeout`, el *throughput* disminuye ligeramente. Este comportamiento se evidencia en la Figura 4.16.

Es importante resaltar que con el valor de `ACKTimeout` correspondiente al estándar, 300 metros, el *throughput* es de apenas el 3.85 % del máximo alcanzado.

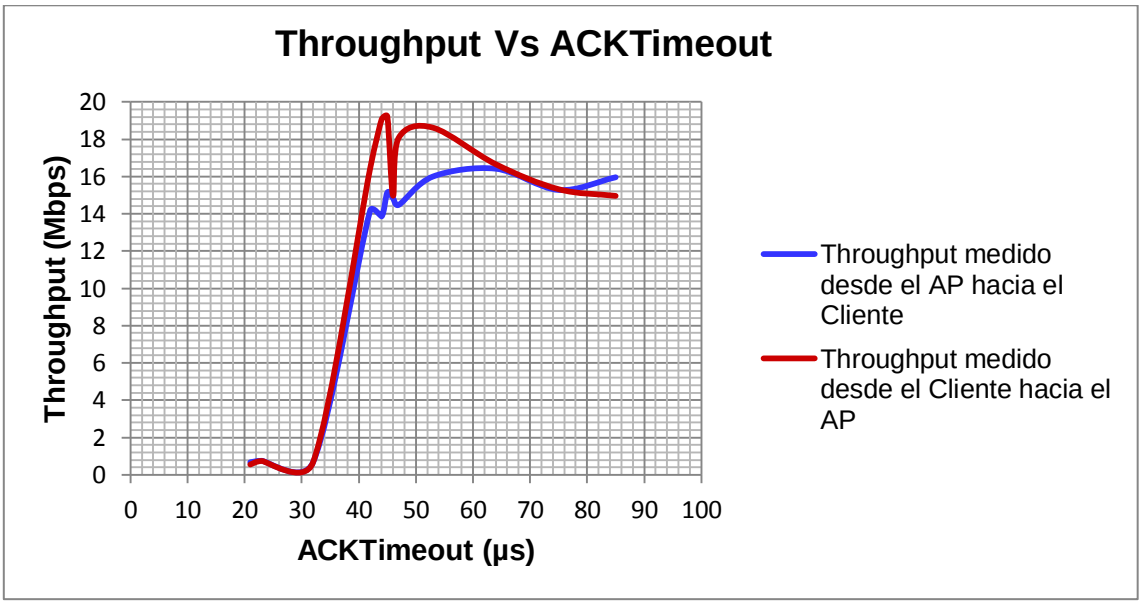


Figura 4.16. *Throughput* Vs `ACKTimeout`.

- **Evaluación del Mecanismo ACK utilizando la herramienta IPERF sobre el protocolo UDP**

La Tablas 4.9 y 4.10 contienen los valores promedio de *throughput*, *jitter* y el porcentaje de datagramas perdidos medidos desde el AP hacia el cliente y desde el cliente hacia el AP respectivamente.

Tabla 4.9. *Throughput, Jitter* y Porcentaje de Datagramas Perdidos Medidos Desde el AP Hacia el Cliente.

ACKTimeout (µs)	**21**	**23**	**32**	**42**	**44**	**45**	**46**	**47**	**53**	**64**	**75**	**85**
Throughput (Mbps)	0.8	0.8	0.8	18.9	20.3	21.5	21.2	20.7	21.1	20.9	21.6	21.9
***Jitter* (ms)**	90.46	71.28	88.04	2.74	1.59	1.55	1.48	1.64	1.62	1.50	1.54	1.45
% Datagramas Perdidos	95.24	94.64	95.23	5.09	7.13	1.57	3.05	4.70	3.01	3.24	1.13	0.02

Tabla 4.10. *Throughput, Jitter* y Porcentaje de Datagramas Perdidos Medidos Desde el Cliente Hacia el AP.

ACKTimeout (µs)	**21**	**23**	**32**	**42**	**44**	**45**	**46**	**47**	**53**	**64**	**75**	**85**
Throughput (Mbps)	1.57	0.79	0.78	20.9	21.74	21.58	22	21.96	22	22	21.86	21.9
***Jitter* (ms)**	96.52	95.56	93.77	1.17	0.90	0.99	0.89	0.95	0.95	1.22	1.14	1.24
%Datagramas Perdidos	95.6	95.6	95.60	3.49	1.05	1.76	0.01	0.15	0.02	0.02	0.40	0.01

De esta prueba se observó que al usar un valor de `ACKTimeout` cercano o mayor al calculado para la distancia del enlace, se alcanza el máximo valor de *throughput* y se mantiene prácticamente constante de ahí en adelante, por el contrario, valores inferiores representan alrededor de 4.75% de dicho de máximo. La Figura 4.17 muestra gráficamente este comportamiento.

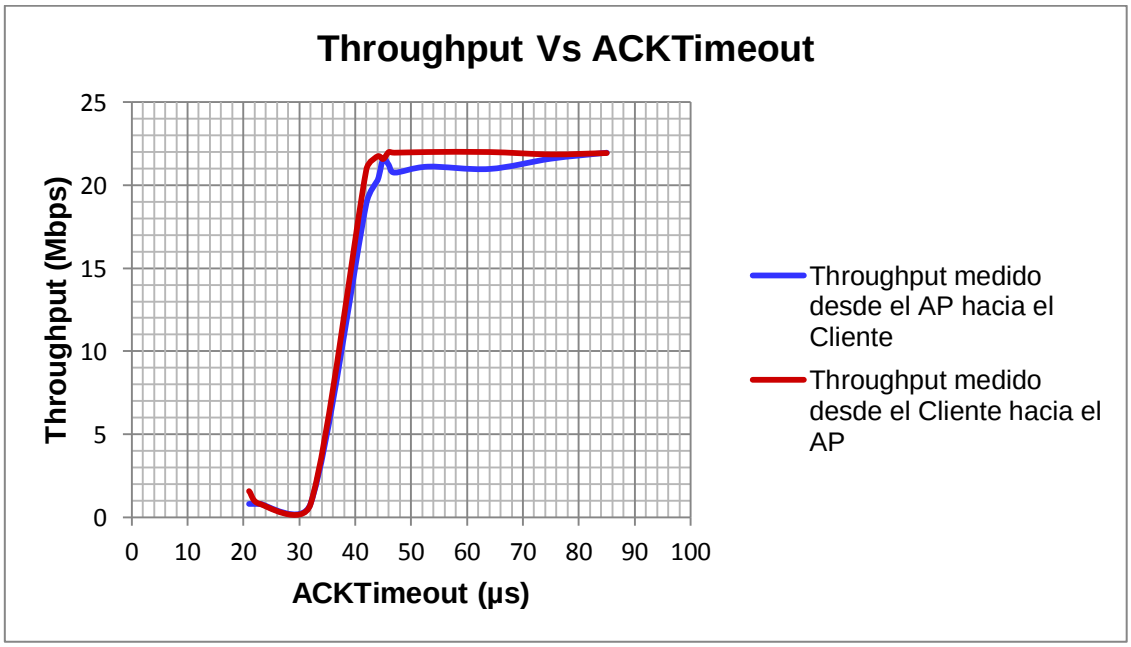


Figura 4.17. *Throughput* Vs `ACKTimeout`.

Respecto al *jitter,* en esta prueba se observó que a medida que se aumenta el `ACKTimeout`, la llegada de paquetes se hace más continua, es decir, a intervalos más regulares, lo que hace disminuir dicha

variable. Los valores iniciales de *jitter* medidos desde el AP hacia el cliente varían de forma distinta a los medidos en la dirección contraria, seguramente por alguna variación en los aspectos que interfieren en la red como puede ser la interferencia. La síntesis de esta información se presenta en la Figura 4.18.

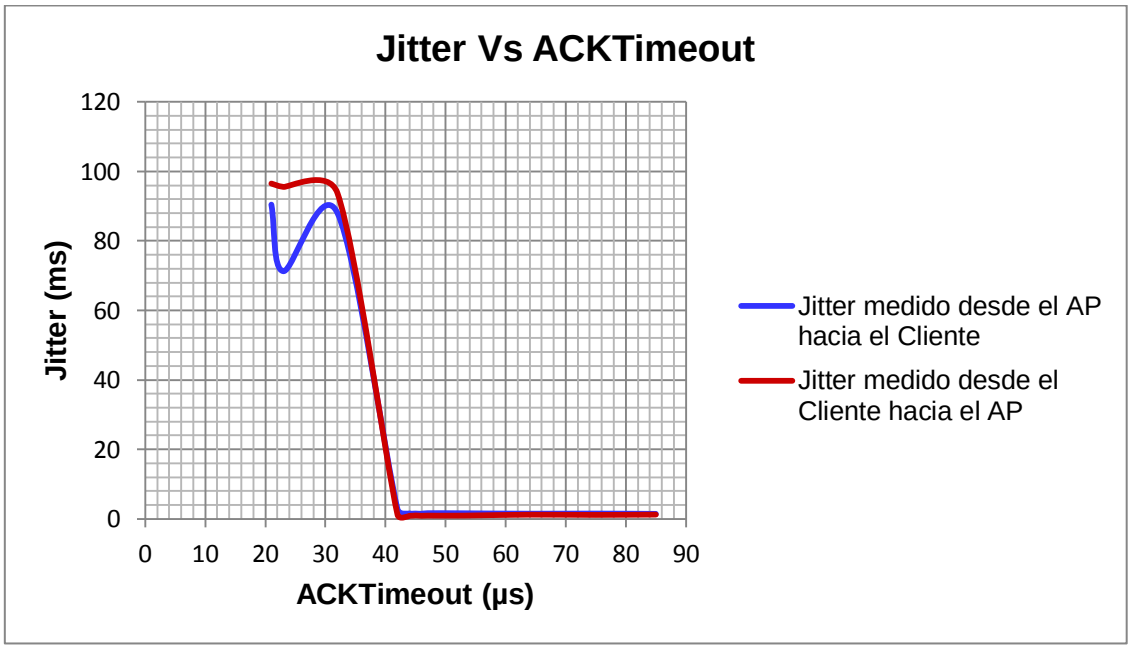


Figura 4.18. *Jitter* Vs `ACKTimeout`.

De igual manera que el *jitter*, el porcentaje de datagramas perdidos disminuye drásticamente al llegar a un `ACKTimeout` adecuado a la distancia y permanece casi nulo en adelante. Tanto la Figura 4.18 y 4.19 siguen el mismo tipo de curva.

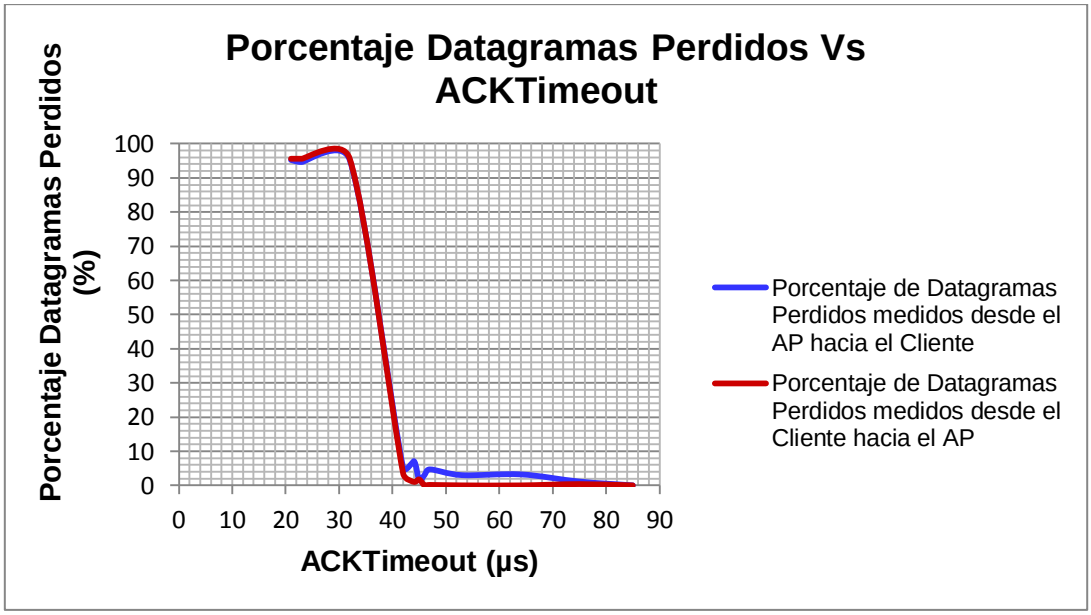


Figura 4.19. Datagramas Perdidos (de un total de 893) Vs `ACKTimeout`.

- **Evaluación del Mecanismo ACK durante una llamada Telefónica**

Las Tablas 4.11 y 4.12 exponen los resultados obtenidos después de realizar 5 llamadas de 30 segundos cada una, mediante la PBX Elastix y los *softphones* X-Lite y Zoiper. Los datos consignados en la Tabla 4.11 fueron obtenidos usando la ley de compresión U del códec G.711 y los datos consignados en la Tabla 4.12 fueron obtenidos usando la ley de compresión A del códec G.711.

Tabla 4.11. Porcentaje de Paquetes Perdidos, *Jitter* Máximo y *Jitter* Promedio Utilizando la Ley de Compresión U del Códec G.711.

ACKTimeout (µs)	**21**	**23**	**32**	**42**	**44**	**45**	**46**	**47**	**53**	**64**	**75**
% Paquetes Perdidos	0.06	0	0.37	0	0	0	0	0	0	0	0
***Jitter* Máximo(ms)**	16	3.95	14.20	3.35	1.25	2.54	1.94	3.46	1.56	2.73	2.33
***Jitter* Promedio(ms)**	4	0.52	4.92	0.64	0.32	0.42	0.41	0.44	0.34	0.56	0.43

Tabla 4.12. Porcentaje de Paquetes Perdidos, *Jitter* Máximo y *Jitter* Promedio Utilizando la Ley de Compresión A del Códec G.711.

ACKTimeout (µs)	**21**	**23**	**32**	**42**	**44**	**45**	**46**	**47**	**53**	**64**	**75**
% Paquetes Perdidos	0.04	0.04	0.11	0	0	0	0	0.01	0	0	0
***Jitter* Máximo (ms)**	2.41	2.93	4.21	3.43	3.16	2.58	1.51	2.81	3.65	2.34	2.34
***Jitter* Promedio (ms)**	0.54	0.57	1.35	0.64	0.52	0.47	0.39	0.42	0.45	0.43	0.41

Para facilitar el análisis de los resultados mostrados en las dos tablas anteriores, se realizó la Figura 4.20 con el porcentaje de paquetes perdidos Vs el `ACKTimeout`. La representación del *jitter* máximo Vs el `ACKTimeout` se presenta en la Figura 4.21 y la gráfica del *jitter* promedio Vs el `ACKTimeout` se presenta en la Figura 4.22. Estas gráficas fueron realizadas tomando los valores de cada uno de los parámetros presentados en las Tablas 4.11 y 4.12.

En dichas Figuras se muestra que a partir de un `ACKTimeout` de 42µs, el número de paquetes perdidos para ambas leyes es prácticamente cero, pero antes de llegar a este valor se presenta un pico más representativo para la ley A, aunque es un valor muy pequeño. Como era de esperarse, si la unidad de transmisión es más pequeña se obtiene un mejor desempeño[57], tal como ocurre en este caso en que prácticamente no se perdieron paquetes para las distintas configuraciones de `ACKTimeout`.

[57] Según la Teoría de la Transmisión.

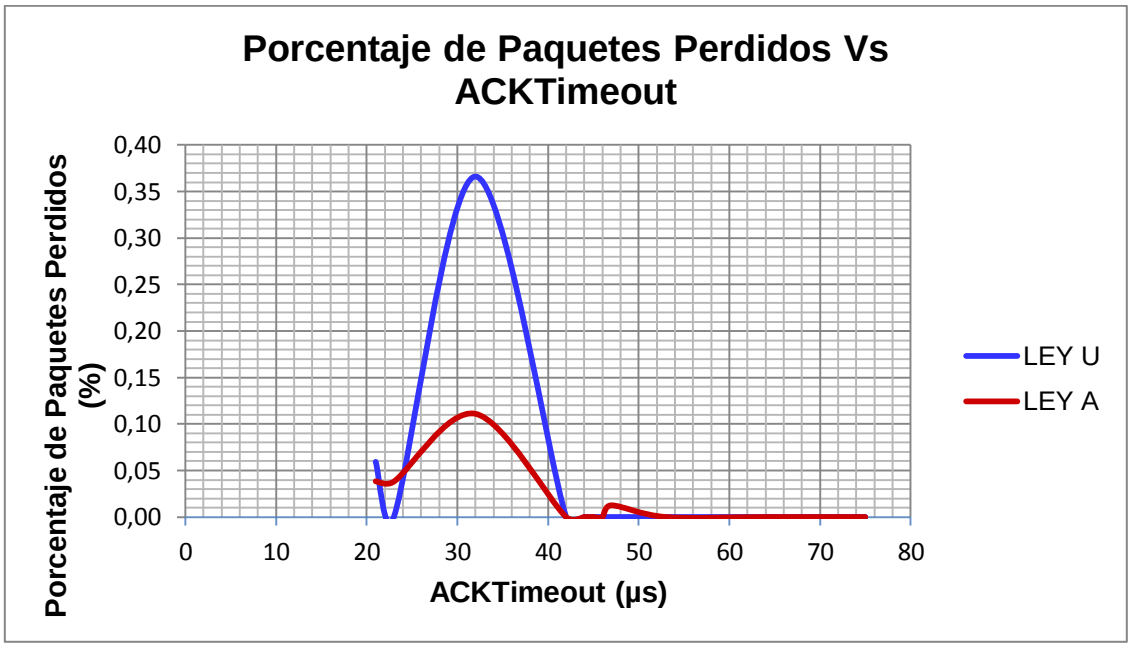


Figura 4.20. Porcentaje de Paquetes Perdidos Vs `ACKTimeout`.

Las Figuras 4.21 y 4.22 muestran el comportamiento del *jitter* máximo y *jitter* promedio durante las conversaciones. Se puede observar que aunque en algún momento el *jitter* llegó a un valor pico de 14.2 ms, el flujo continuo de paquetes de voz, y el uso de protocolos de tiempo real permitieron que a nivel promedio se regule esta diferencia de paquete a paquete a 4.92ms para el peor de los casos, pues para los demás valores de `ACKTimeout` el *jitter* promedio oscila en 0.5 ms.

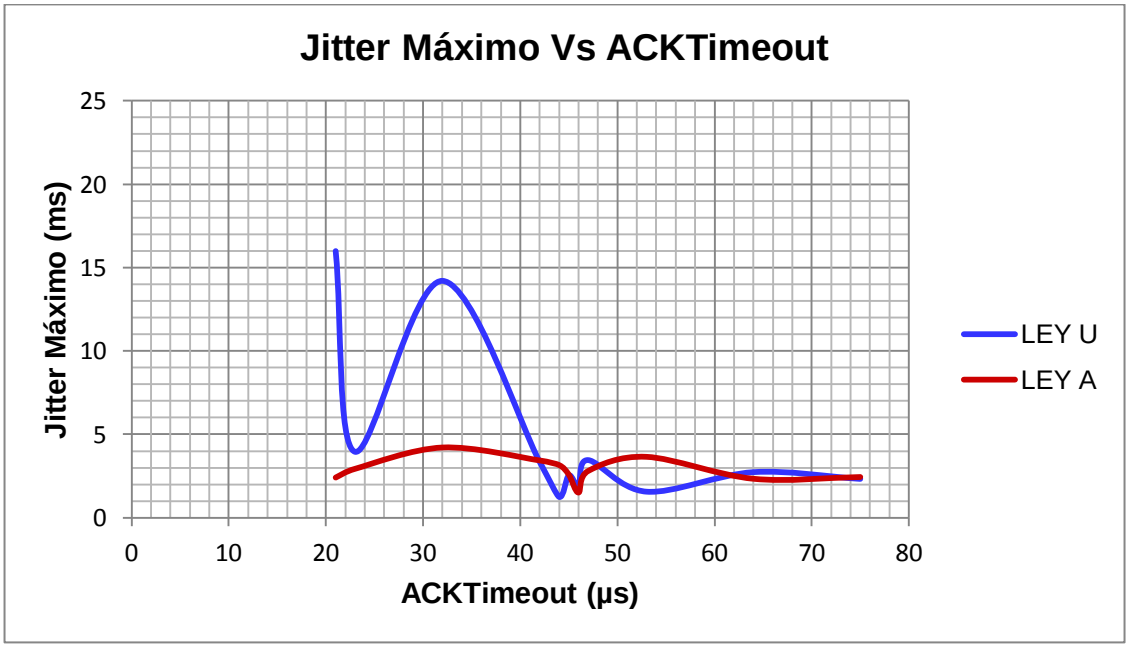


Figura 4.21. *Jitter* Máximo Vs `ACKTimeout`.

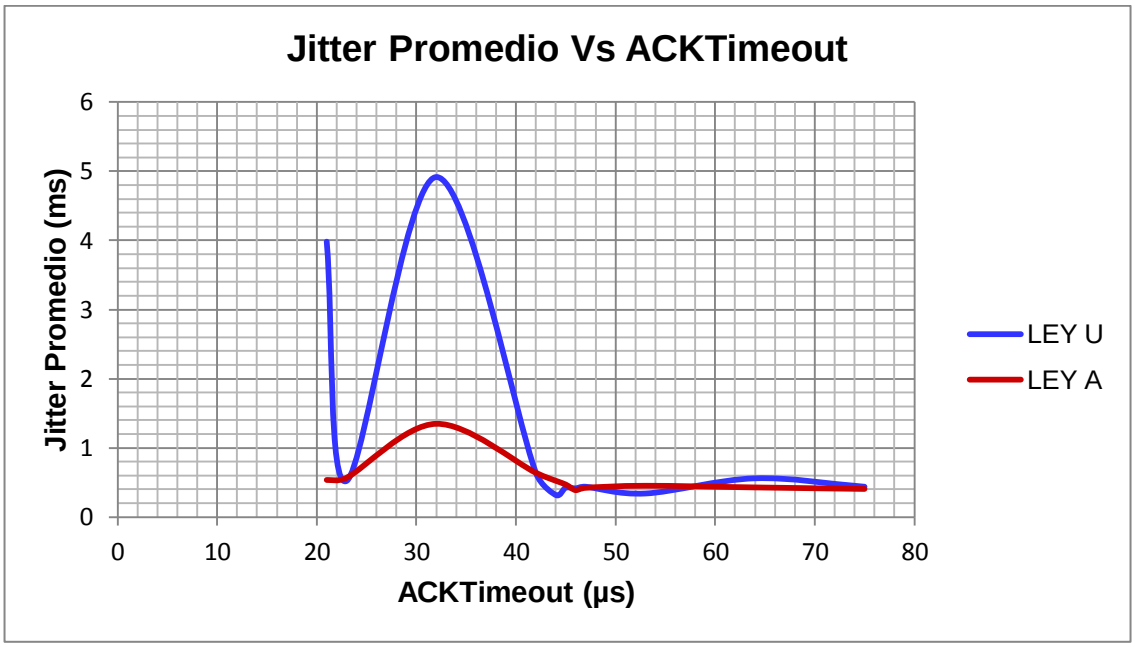


Figura 4.22. *Jitter* Promedio Vs `ACKTimeout`.

- **Evaluación del Mecanismo ACK durante un *streaming* de video.**

La Tabla 4.13 presenta la información obtenida al realizar un *streaming* de video por VLC. Para cada valor de `ACKTimeout` se tomaron muestras durante 60 segundos, obteniendo que para valores bajos de `ACKTimeout`, el número de ACK duplicados a nivel de transporte es realmente alto, mientras que el número de paquetes HTTP transmitidos es más bajo que en los otros casos, esto debido a que no se mantiene una correcta sesión TCP y el flujo de datos es mucho menor.

Tabla 4.13. ACK Duplicados, Retransmisiones y Paquetes HTTP Transmitidos.

ACKTimeout (µs)	**21**	**23**	**32**	**42**	**44**	**45**	**46**	**47**	**53**	**64**	**75**	**85**
ACK Duplicados	356	458	525	2	2	8	9	2	7	10	3	0
Retransmisiones	18	17	37	5	5	1	2	12	1	8	0	0
Paquetes HTTP Transmitidos	3817	3958	3010	5132	5534	4801	5055	7174	7349	6859	4466	5457

4.4.1.2 **Evaluación del desempeño del mecanismo No ACK.**

- **Evaluación de No ACK utilizando la herramienta IPERF sobre el protocolo TCP.**

La Tabla 4.14 muestra los datos obtenidos con la herramienta *iperf* usando el protocolo TCP. Los datos fueron medidos desde el AP hacia el cliente y desde el cliente hacia el AP donde en ambos casos se obtuvo un desempeño bajo de la red. Los valores de *throughput* obtenidos en esta prueba representan el 3.3% y el 0.77% del valor máximo obtenido en la Tabla 4.9. Estos valores son comparables con los obtenidos al realizar las pruebas con ACK para valores bajos de `ACKTimeout`.

Tabla 4.14. *Throughput* Medido desde el AP Hacia el Cliente y desde el Cliente hacia el AP (Mbps).

	Desde el AP hacia el Cliente	Desde el Cliente hacia el AP
Throughput Promedio	0.73	0.17

- **Evaluación de No ACK utilizando la herramienta IPERF sobre el protocolo UDP**

La Tabla 4.15 muestra los datos obtenidos con la herramienta *iperf* usando el protocolo UDP, donde se evidencia la tendencia del *throughput* hacia valores similares a los obtenidos con un `ACKTimeout` no adecuado para la distancia del enlace, pero con ciertas diferencias como lo son un *jitter* más bajo y una pérdida de datagramas ligeramente menor.

Tabla 4.15. *Throughput, Jitter* y Datagramas Perdidos Medidos desde el AP Hacia el cliente y desde el cliente Hacia el AP.

De AP hacia el Cliente			***De Cliente a AP***		
Throughput (Mbps)	***Jitter* (ms)**	**%Datagramas Perdidos**	**Throughput (Mbps)**	***Jitter* (ms)**	**%Datagramas Perdidos**
0.9	28.81	89.40	0.83	32.80	86.01

- **Evaluación de No ACK durante una llamada Telefónica**

La Tabla 4.16 expone los promedios de los resultados obtenidos después de realizar 5 llamadas de 30 segundos cada una, mediante la PBX Elastix. Los datos consignados en ella fueron obtenidos usando la ley de compresión U del códec G.711 y la ley de compresión A del mismo códec. En esta se encontró un *jitter* promedio y un porcentaje los paquetes perdidos bajo, permitiendo el correcto establecimiento de la comunicación de voz.

Tabla 4.16. Porcentaje de Paquetes Perdidos, *Jitter* Máximo y *Jitter* Promedio Utilizando la ley de compresión U y la ley de compresión A del Códec G.711.

Ley U del códec G.711			**Ley A del códec G.711**		
% Paquetes Perdidos	***Jitter* Máximo (ms)**	***Jitter* Promedio (ms)**	**% Paquetes Perdidos**	***Jitter* Máximo (ms)**	***Jitter* Promedio (ms)**
0.50	33.444	3.304	0.59	20.17	1.76

- **Evaluación de No ACK durante un streaming de video**

Para el *streaming* de video no fue posible establecer correctamente la sesión TCP, como resultado el único flujo de datos de HTTP fue de 29 paquetes como se evidencia en la Tabla 4.17.

Tabla 4.17. ACK Duplicados, Retransmisiones y Paquetes HTTP Transmitidos.

ACK Duplicados	**Retransmisiones**	**Paquetes HTTP**
0	0	29

4.4.2 Enlace FIET – Cajete

Para el enlace FIET-Cajete se analizaron los resultados de las pruebas realizadas para `ACKTimeouts` anteriores, cercanos y posteriores a los correspondientes a la distancia entre los dos puntos involucrados (6.6 Km). Esta relación se encuentra en la Tabla 4.18 y es calculada a partir de la ecuación (4).

Tabla 4.18. Equivalencia entre Distancia y `ACKTimeout`.

Distancia (m)	0	322	1609	3218	4827	6114	6275	6436	6597	6758	8045	9654
ACKTimeout (µs)	21	23	32	42	53	62	63	64	65	66	75	85

4.4.2.1 Evaluación del desempeño del Mecanismo de Reconocimiento ACK

- **Evaluación del Mecanismo ACK utilizando la herramienta IPERF sobre el protocolo TCP**

Las Tablas 38 y 39 contienen los valores del *throughput* del enlace medido desde el AP hacia el cliente y desde el cliente hacia el AP respectivamente para diferentes valores de `ACKTimeout`. Con el objetivo de obtener datos más precisos se realiza 5 veces cada prueba bajo las mismas condiciones y luego se calcula el *throughput* promedio.

Tabla 4.19. *Throughput* Medido desde el AP hacia el Cliente.

ACKTimeout (µs)	21	23	32	42	53	62	63	64	65	66	75	85
Throughput (Mbps)	0.75	0.75	0.75	0.75	0.73	9.09	12.46	14	10.8	9.35	14.44	13.02

Tabla 4.20. *Throughput* Medido desde el Cliente hacia el AP.

ACKTimeout (µs)	21	23	32	42	53	62	63	64	65	66	75	85
Throughput (Mbps)	0.76	0.76	0.77	0.76	0.73	13.28	10.54	11.51	11.56	5.94	8.13	8.4

Del resultado de esta prueba se puede observar que el *throughput* del enlace, tanto en un sentido como en el otro, aumenta al acercarse al valor del `ACKTimeout` calculado para la distancia del enlace hasta un máximo de 14.44 Mbps. A medida que sigue aumentando el `ACKTimeout`, el *throughput* disminuye notablemente. Nuevamente se aprecia que el *throughput* alcanzado para valores de `ACKTimeout` bajo, incluyendo el establecido en el estándar, es menor al 6% del valor máximo alcanzado. Este comportamiento se evidencia en la Figura 4.23.

Es de anotar, que en comparación con el enlace FIET-Tablazo, el *throughput* decrece más rápidamente después de alcanzar el valor máximo.

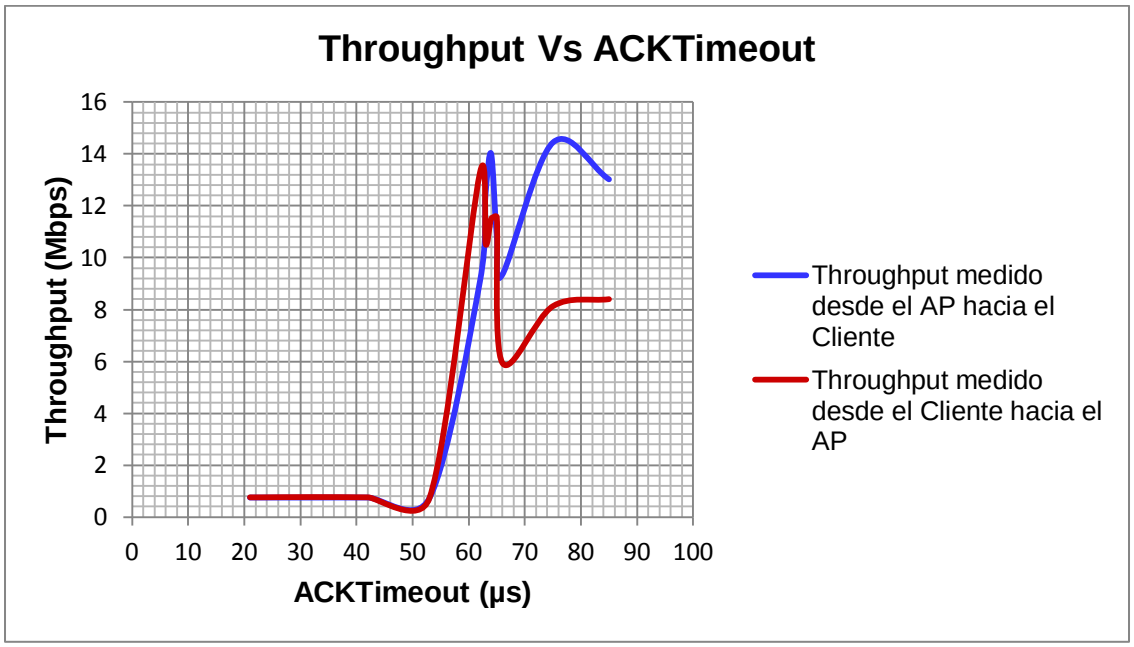


Figura 4.23. Throughput Vs `ACKTimeout`.

- **Evaluación del Mecanismo ACK utilizando la herramienta IPERF sobre el protocolo UDP.**

La Tablas 4.21 y 4.22 contienen los valores de *throughput*, *jitter* y el porcentaje de datagramas perdidos medidos desde el AP hacia el cliente y desde el cliente hacia el AP respectivamente. Con el objetivo de obtener datos más precisos se realiza 5 veces cada prueba bajo las mismas condiciones y luego se calcularon los valores promedio.

Tabla 4.21. *Throughput, Jitter* y Porcentaje de Datagramas Perdidos medidos desde el AP hacia el Cliente.

ACKTimeout (µs)	21	23	32	42	53	62	63	64	65	66	75	85
Throughput (Mbps)	0.80	0.8	0.79	0.73	0.71	14.7	14.8	14	15.4	14.7	14.1	15.5
***Jitter* Promedio (ms)**	21.29	21.10	20.99	62.03	25.51	0.77	1.73	0.76	0.55	0.69	0.75	0.96
% Datagramas Perdidos	93.06	93.46	93.05	93.82	93.77	7.21	5.97	9.57	4.04	7.55	10.2	2.72

Tabla 4.22. *Throughput, Jitter* y Porcentaje de Datagramas Perdidos medidos desde el Cliente hacia el AP.

ACKTimeout (µs)	21	23	32	42	53	62	63	64	65	66	75	85
Throughput (Mbps)	0.81	0.81	0.81	0.80	0.80	16	14.9	14.7	11.6	13.2	11.7	12.2
***Jitter* Promedio (ms)**	87.06	69.88	120.4	96.48	121.1	0.67	1.82	0.83	1.69	2.03	0.98	2.17
% Datagramas Perdidos	94.20	93.60	94.20	94.60	94.60	10.8	7.78	6.94	25.4	16.2	25.2	21.8

Para facilitar el análisis de los resultados mostrados en las tablas anteriores, se realizaron las Figuras 4.24, 4.25 y 4.26, donde se observó que al usar un valor de `ACKTimeout` cercano o mayor al calculado

para la distancia del enlace, se alcanzan los máximos valores de *throughput* y a medida que aumenta el `ACKTimeout`, el *throughput* decrece momentáneamente para volver a incrementarse.

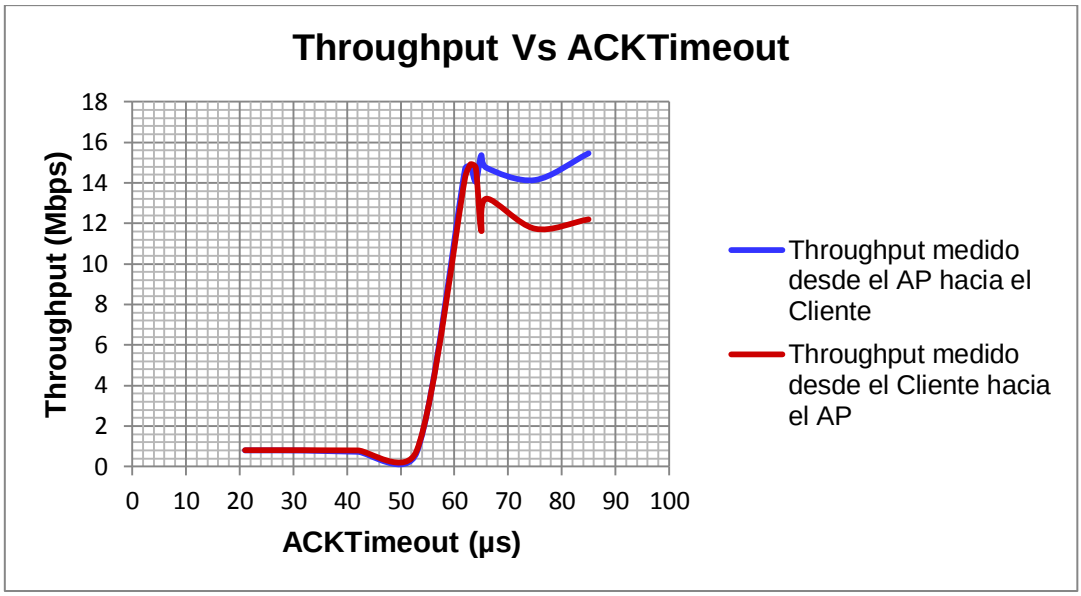


Figura 4.24. *Throughput* Vs `ACKTimeout`.

De acuerdo a la Figura 4.25, se observó que a medida que se aumenta el `ACKTimeout`, la llegada de paquetes se hace más continua, es decir, a intervalos más regulares, lo que hace disminuir el *jitter.* Los valores iniciales de *jitter* medidos desde el AP hacia el cliente varían de forma distinta a los medidos en la dirección contraria, seguramente por alguna variación en los aspectos que interfieren en la red como puede ser la interferencia debido a la presencia de múltiples redes inalámbricas cercanas.

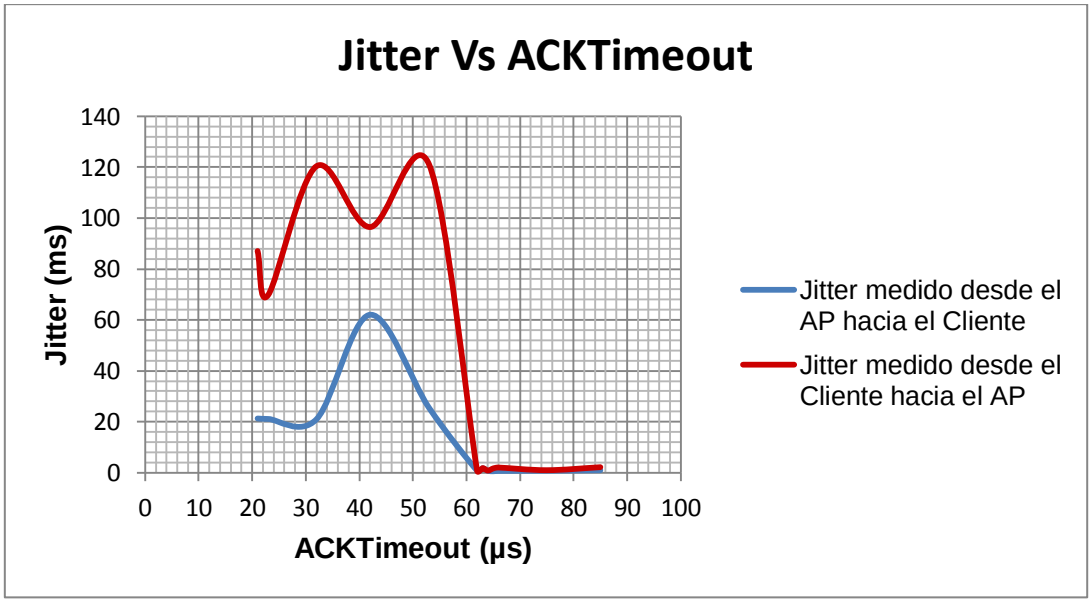


Figura 4.25. *Jitter* Vs `ACKTimeout`.

La disminución de *jitter* se puede asociar al menor porcentaje de paquetes perdidos a medida que se acerca el `ACKTimeout` al valor calculado para la distancia de este enlace, tal como se muestra en la Figura 4.26.

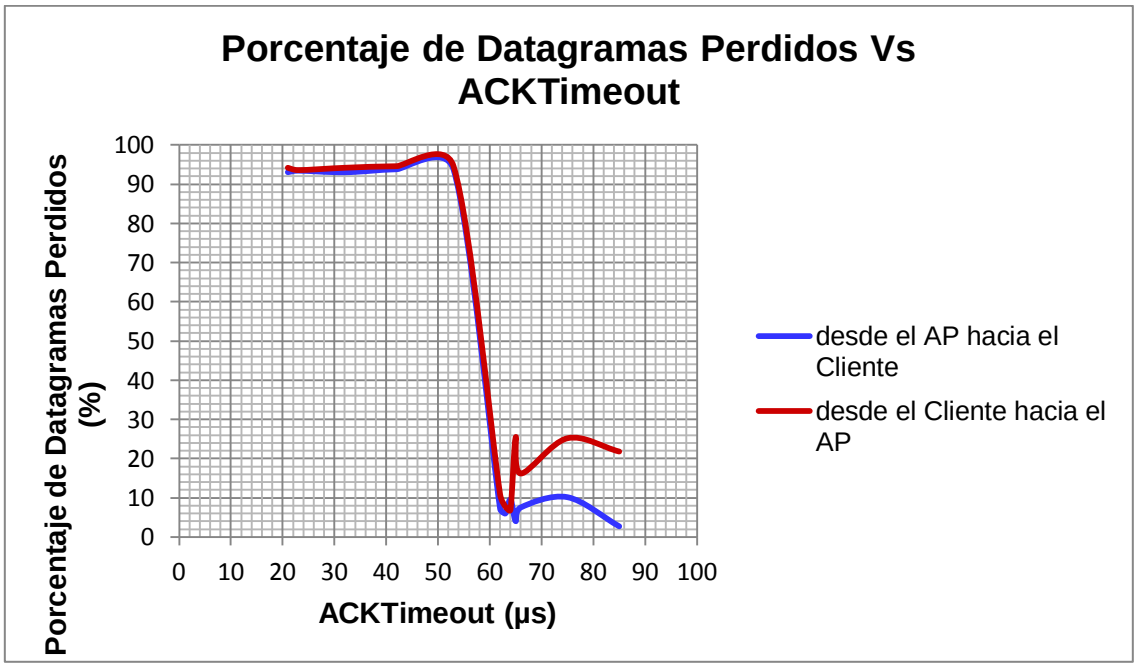


Figura 4.26. Porcentaje de Datagramas Perdidos Vs `ACKTimeout`.

- **Evaluación del Mecanismo ACK durante una llamada Telefónica**

Las Tablas 4.23 y 4.24 presentan los resultados obtenidos después de realizar 5 llamadas de 30 segundos cada una, mediante la PBX Elastix. Los datos consignados en la Tabla 42 fueron obtenidos usando "la ley de compresión U" del códec G.711 y los datos consignados en la Tabla 43 fueron obtenidos usando "la ley de compresión A" del códec G.711.

Tabla 4.23. Porcentaje de Paquetes Perdidos, *Jitter* Máximo y *Jitter* Promedio utilizando la ley de compresión U del Códec G.711.

ACKTimeout (µs)	**21**	**23**	**32**	**42**	**53**	**62**	**63**	**64**	**65**	**66**	**75**	**85**
% Paquetes Perdidos	0.05	0.04	0.01	0.03	0.09	0	0	0.01	0.01	0	0	0
***Jitter* Máximo(ms)**	5.83	4.46	5.06	3.48	2.25	3.11	2.83	3.11	3.90	2.81	3.92	3.77
Jitter Promedio(ms)	1.01	1.07	0.92	0.83	0.52	0.88	0.46	0.60	0.62	0.50	0.58	0.57

Tabla 4.24. Porcentaje de Paquetes Perdidos, *Jitter* Máximo y *Jitter* Promedio utilizando la ley de compresión A del Códec G.711.

ACKTimeout (µs)	**21**	**23**	**32**	**42**	**53**	**62**	**63**	**64**	**65**	**66**	**75**	**85**
% Paquetes Perdidos	0.12	0.03	0.01	0.04	0.03	0	0	0	0	0	0	0
***Jitter* Máximo(ms)**	7.75	4.67	4.54	6.13	9.24	4.14	4.82	4.49	3.71	3.96	4.77	6.78
Jitter Promedio(ms)	1.92	0.96	1.32	1.43	2.75	1.22	1.09	1.47	0.72	0.98	1.26	1.87

Dichas Tablas representadas gráficamente en las Figuras 4.27, 4.28 y 4.29, muestran valores bajos de pérdida de paquetes, y aunque se visualizan curvas de variación, éstas son muy cercanas a cero. Caso que no ocurre con el *jitter* máximo en donde se presenta un pico alto en el comportamiento de la ley de compresión A, pero si se observa el comportamiento del *jitter* promedio, éste no fue elevado significativamente a causa de ese fenómeno.

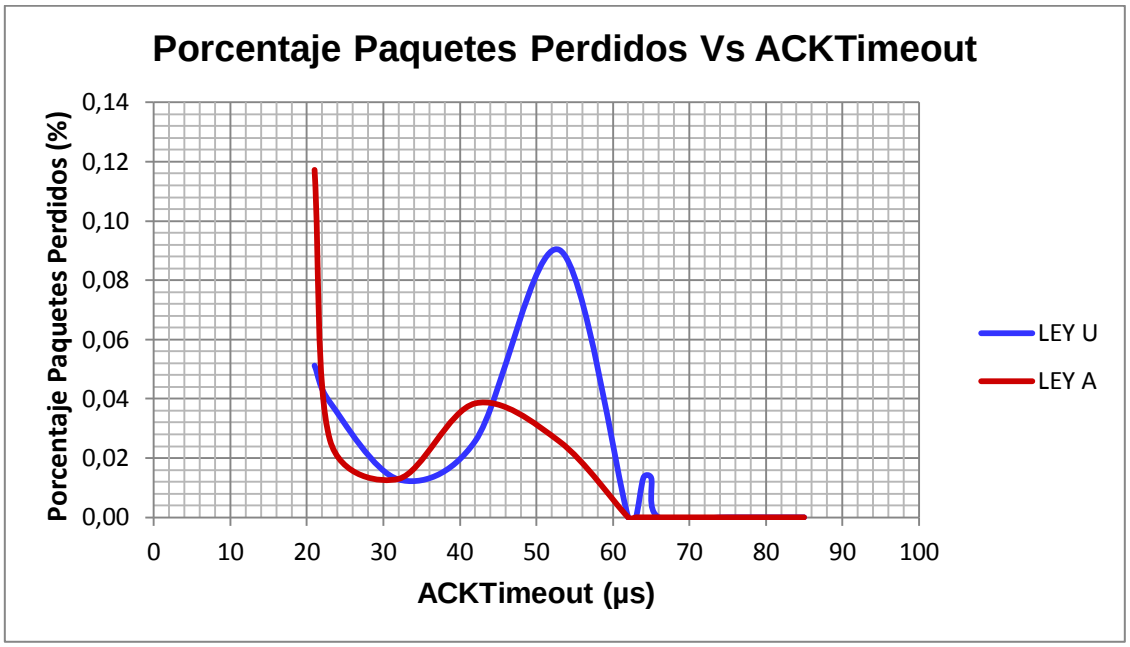


Figura 4.27. Porcentaje de Paquetes Perdidos Vs `ACKTimeout`.

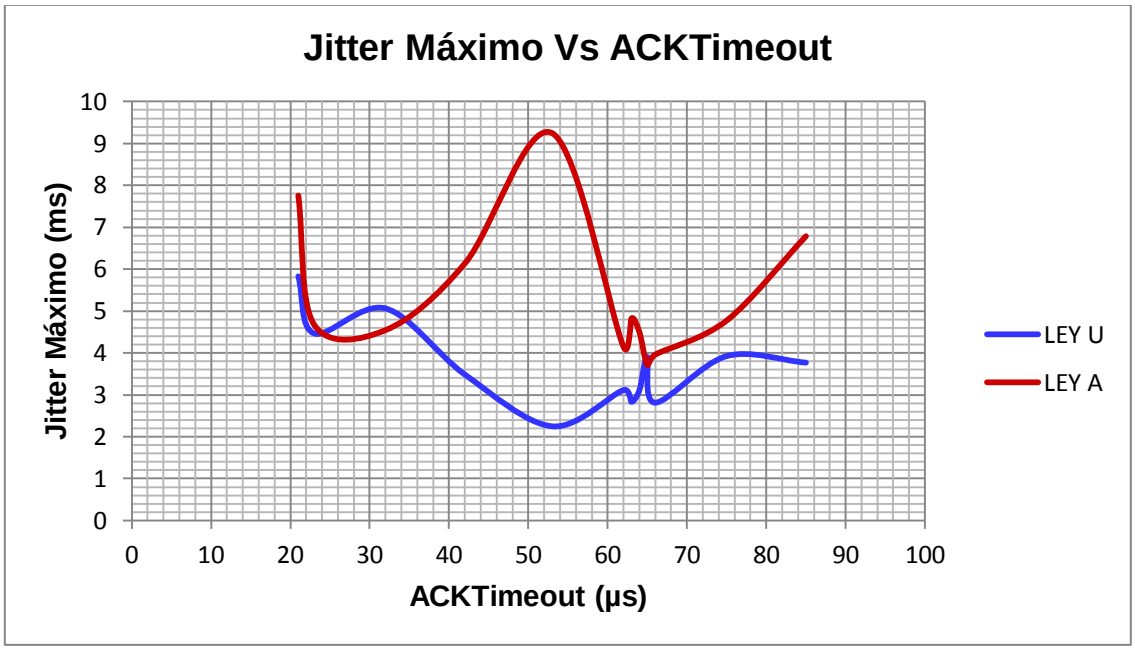


Figura 4.28. *Jitter* Máximo Vs `ACKTimeout`.

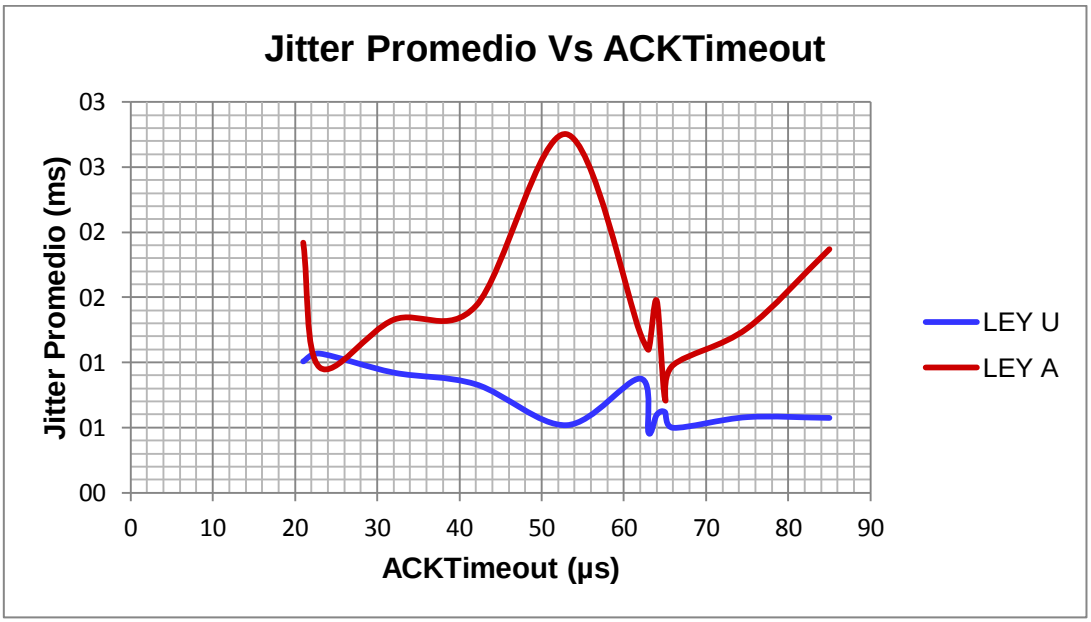


Figura 4.29. *Jitter* promedio Vs `ACKTimeout`.

- **Evaluación del Mecanismo ACK durante un *streaming* de video.**

La Tabla 4.25 despliega la información obtenida al realizar un streaming de video por VLC. Para cada valor de `ACKTimeout` se tomaron muestras durante 60 segundos, obteniendo que para valores bajos de `ACKTimeout`, el número de ACK duplicados a nivel de transporte es alto. Para los mismos, el número de paquetes HTTP transmitidos es muy alto al igual que el número de paquetes retransmitidos, esto debido a que no se estableció una correcta sesión TCP y el flujo de datos de video fue mucho menor, dicho comportamiento refuerza el obtenido con el enlace FIET-Tablazo.

Tabla 4.25. ACK Duplicados, Retransmisiones y Paquetes HTTP Transmitidos.

ACKTimeout (Mbps)	**21**	**23**	**32**	**42**	**53**	**62**	**63**	**64**	**65**	**66**	**75**	**85**
ACK Duplicados	201	196	183	177	171	0	0	3	3	0	2	0
Retransmisiones	4638	5491	5970	6124	6057	59	53	18	76	39	60	49
Paquetes HTTP Transmitidos	6188	6886	6245	8903	8716	2245	2164	2433	4278	4108	4951	4886

4.4.2.2 **Evaluación del desempeño del mecanismo No ACK.**

- **Evaluación de No ACK utilizando la herramienta IPERF sobre el protocolo TCP.**

La Tabla 4.26 muestra los datos obtenidos con la herramienta *iperf* usando el protocolo TCP. Los datos fueron medidos desde el AP hacia el cliente y desde el cliente hacia el AP. En esta se observa que el *throughput* promedio es inferior al alcanzado en las pruebas anteriores sobre este escenario, aunque refuerzan los resultados obtenidos con el primer enlace de prueba.

Tabla 4.26. *Throughput* medido desde el AP hacia el Cliente y desde el Cliente hacia el AP.

	Desde el AP hacia el Cliente	***Desde el Cliente hacia el AP***
Throughput (Mbps)	0.61	0.51

- **Evaluación de No ACK utilizando la herramienta IPERF sobre el protocolo UDP.**

La Tabla 4.27 muestra los datos obtenidos con la herramienta *iperf* usando el protocolo UDP. Los datos fueron medidos desde el AP hacia el cliente y desde el cliente hacia el AP. En estos se evidencia la limitación en *throughput, jitter* y datagramas perdidos que establece la falta de un mecanismo de reconocimiento, pues los valores obtenidos son muy bajos comparados a los presentados en el mejor de los casos en la Tabla 40.

Tabla 4.27. *Throughput, Jitter* y Porcentaje de Datagramas Perdidos medidos desde el AP hacia el Cliente y desde el Cliente hacia el AP.

Desde el AP hacia el Cliente			***Desde el Cliente hacia el AP***		
Throughput (Mbps)	***Jitter* (ms)**	**%Datagramas Perdidos**	**Throughput (Mbps)**	***Jitter* (ms)**	**%Datagramas Perdidos**
0.992	48.81	92.20	0.99	45.19	92.23

- **Evaluación de No ACK durante una llamada Telefónica.**

La Tabla 4.28 presenta los resultados obtenidos después de realizar 5 llamadas de 30 segundos cada una, mediante la PBX Elastix. Los datos consignados en esta tabla fueron obtenidos usando la ley de compresión U del códec G.711 y la ley de compresión A del códec G.711. Con tales datos se puede mostrar que para la ausencia de mensajes de reconocimiento, el mejor escenario en que puede funcionar este esquema es para el de voz, pues la tabla presenta valores muy adecuados para el establecimiento de la comunicación.

Tabla 4.28. Porcentaje de Paquetes Perdidos, Delta Máximo, *Jitter* Máximo y *Jitter* Promedio utilizando la ley de compresión U y la ley de compresión A del Códec G.711.

Ley U del códec G.711			**Ley A del códec G.711**		
% Paquetes Perdidos	***Jitter* Máximo (ms)**	***Jitter* Promedio (ms)**	**% Paquetes Perdidos**	***Jitter* Máximo (ms)**	***Jitter* Promedio (ms)**
0.27	3.828	0.27	0.22	4.234	0.22

- **Evaluación de ACK durante un streaming de video.**

Para el *streaming* de video no fue posible establecer correctamente la sesión TCP, como resultado el único flujo de datos de HTTP fue de 63 paquetes como se evidencia en la Tabla 4.29.

Tabla 4.29. ACK duplicados, Retransmisiones y Paquetes HTTP Transmitidos.

ACK Duplicados	**Retransmisiones**	**Paquetes HTTP**
0	0	63

Con el desarrollo del Trabajo de Grado se generó una base conceptual de los mecanismos de reconocimiento de 802.11 orientado a enlaces de larga distancia en donde se presentaron varios escenarios que permitieron evidenciar su funcionamiento observando adecuadamente los indicadores que permiten medir su desempeño. Así mismo se evidenciaron los problemas que presentan este tipo de redes y cómo los mecanismos de reconocimiento son parte de la cura de los mismos y se formuló una guía relacionada a cómo generar o modificar procedimientos de nivel MAC dentro de una plataforma de desarrollo de bajo costo.

CONCLUSIONES

Fue enriquecedor e importante participar del proyecto de investigación financiado por FRIDA y desarrollado por los grupos de investigación GNTT, IDIS y GESC denominado "*Análisis, Uso, Adecuación y Apropiación de Servicios sobre Tecnologías Inalámbricas en Zonas de Difícil Acceso de las Poblaciones Indígenas del Cauca Andino*", aportar conocimiento a las comunidades indígenas del Cauca y aparte de eso brindar mejoras al funcionamiento del enlace gracias al desarrollo de este Trabajo de Grado.

El proceso de investigación llevo a encontrar una plataforma libre de desarrollo del nivel de Control de Acceso al Medio de 802.11 que aunque tiene sus limitaciones por sí solo, permite estudiar el comportamiento y modificar aspectos básicos del comportamiento de la MAC.

Una vez establecidos los enlaces piloto, fue evidente la necesidad de tener un proceso organizado para el desarrollo de las pruebas, ya que la lejanía de los sitios impide el rápido desarrollo de las mismas y es necesario tener un canal de comunicación siempre abierto.

La opción de convertir los enrutadores inalámbricos en minicomputadores ofrece gran número de posibilidades de estudio de redes inalámbricas y amplía las posibilidades de funcionamiento de las mismas.

La configuración por defecto de los equipos para redes inalámbricas típicas, es decir, con un `ACKTimeout` no deben ser usados para establecer enlaces de larga distancia, pues aunque se aumente la potencia y ganancia de las antenas, no es suficiente para obtener un buen desempeño de la red.

Un AP estándar funciona en largas distancias casi igual a uno sin mecanismo de acceso, es decir, con `No ACK`.

A pesar de considerar el mecanismo `No ACK` como alternativa viable para enlaces de larga distancia, los resultados de las pruebas arrojaron que la probabilidad de que un paquete llegue bien y la estación transmisora no necesite ser notificada de ello es prácticamente del 8 %, ya que fue ese el desempeño promedio obtenido respecto al máximo valor.

El funcionamiento de una red inalámbrica de larga distancia sin mecanismo de reconocimiento es solo recomendable para comunicaciones de voz, pues, los valores de *jitter* y paquetes perdidos son muy tolerables.

Una correcta elección del `ACKTimeout` hace posible establecer conexiones de transmisión de datos mucho mayores que las obtenidas con un ACK por defecto. En este proceso se debe tener en cuenta que si se escoge uno mayor el desempeño será menor al obtenido con el `ACKTimeout` preciso.

No es estrictamente necesario conocer la distancia entre los puntos a interconectar, al realizar las pruebas de conectividad variando el `ACKTimeout` es posible determinar el mejor tiempo y así encontrar la distancia.

El `ACKTimeout` no es el único parámetro que afecta el correcto funcionamiento de una red inalámbrica de larga distancia. La interferencia, la frecuencia usada, la potencia, y los parámetros propios del protocolo como `slottime` afectan el buen funcionamiento.

La mejor manera de transmitir un video por *streaming* es mediante el uso de UDP o protocolos de tiempo real, en el desarrollo del trabajo se utilizó TCP únicamente con fines demostrativos.

Un valor de `ACKTimeout` por debajo del indicado para la distancia del enlace representa gran cantidad de retransmisiones y ACK duplicados al momento de hacer un streaming de video. Solo bajo una buena configuración del `ACKTimeout` se presenta un flujo continuo de video.

REFERENCIAS BIBLIOGRÁFICAS

[1] L. Subramanian, S. Surana, R. Patra, A Sheth. "*Rethinking Wireless for the Developing World"*. Grupo TIER (Technology and Infrastructure for Emerging Regions) de la Universidad de California en Berkley. 2006. Documento PDF disponible en: http://tier.cs.berkeley.edu/docs/wireless/large_wild.pdf .Recurso consultado por última vez en junio de 2008.

[2] "*ACK Timeouts and the effects on distance links".* Página disponible en: http://www.air-stream.org.au/ACK_Timeouts . Recurso consultado por última vez en noviembre de 2008.

[3] IEEE 802.11 Working Group, "*IEEE Standard for Information technology—Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Higher-Speed Physical Layer Extension in the 2.4 GHz Band*". IEEE Computer Society. Documento PDF disponible en: http://easy.intranet.gr/IEEE80211b.pdf. Recurso consultado por última vez en mayo de 2009.

[4] J. Geier. *Making the Choice: 802.11a or 802.11g*. WiFi Planet. 2002. Documento PDF disponible en: http://www.wi-fiplanet.com/tutorials/article.php/1009431. Recurso consultado por última vez en mayo de 2009.

[5] WiFi Alliance. *Wi-Fi CERTIFIED™ 802.11n draft 2.0: Longer-Range, Faster-Throughput, Multimedia-Grade Wi-Fi® Networks*. 2007. Documento PDF disponible en: http://www.wi-fi.org/files/kc/WFA_802_11n_Industry_June07.pdf. Recurso consultado por última vez en junio de 2009.

[6] WiFi Alliance. *What is Wi-Fi Certification?.* http://www.wi-fi.org/knowledge_center_overview.php?docid=3280. Recurso consultado por última vez en mayo de 2009.

[7] *Certified Wireless Network Administrator™ Study Guide.* Planet3 Wireless Inc. 2002.

[8] D. Coleman, D. Westcott, *CWNA® Certified Wireless Network Administrator™ Study Guide*. Wiley Publishing, Inc., Indianapolis, Indiana. 2006.

[9] T. Li, Q. Ni y T. Xiao. *Investigation of the block ACK scheme in wireless ad hoc networks*. Wireless Communications and Mobile Computing. Wiley InterSience. 2006.

[10] P. Roshan y J. Leary. *802.11 Wireless LAN Fundamentals*. Cisco Press. Cisco Systems, Inc. 2004.

[11] R. Andrade, P. Salas y D. Paredes. *Tecnología Wi-Fi*. Buenos Aires 2008.

[12] F. Simó. "*Modelado y optimización de IEEE 802.11 para su aplicación en el despliegue de redes extensas en zonas rurales aisladas en países en desarrollo.*" Depto. de Ingeniería de Sistemas Telemáticos. Universidad Politécnica de Madrid. Madrid, España. Enero de 2007.

[13] P. Bhagwat, B. Raman y D. Sanghi. "*Turning 802.11 Inside-Out*". ACM SIGCOMM Computer Communication Review, Vol. 34. 2004. Documento PDF disponible en: http://www.cse.iitk.ac.in/users/braman/papers/inside-out.pdf. Recurso consultado por última vez en agosto de 2009.

[14] IEEE 802.11 Working Group, "*IEEE Standard for Information technology—Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*". IEEE

Computer Society. Documento PDF disponible en: http://standards.ieee.org/getieee802/download/802.11-2007.pdf. Recurso consultado por última vez en junio de 2009.

[15] IEEE 802.11 Working Group, "*IEEE Standard for Information technology—Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*". IEEE Computer Society. Documento PDF disponible en: http://standards.ieee.org/getieee802/download/802.11e-2005.pdf. Recurso consultado por última vez en julio de 2009.

[16] V. Gite. "*What is Linux*?". 2009. Página disponible en: http://www.cyberciti.biz/faq/explain-linux/. Recurso consultado por última vez en julio de 2009.

[17] A. Weiss. "*The Open Source WRT54G Story*". Página disponible en: http://www.wi-fiplanet.com/tutorials/article.php/3562391. Recurso consultado por última vez en junio de 2009.

[18] "*What is DD-WRT?".* Página disponible en: http://www.dd-wrt.com/wiki/index.php/What_is_DD-WRT%3F. Recurso consultado por última vez en julio de 2009.

[19] "*OPENWRT*". 2005. Página disponible en: http://openwrt.org/. Recurso consultado por última vez en agosto de 2009.

[20] The madwifi Project. *MadWifi*. Página disponible en: http://madwifi-project.org/wiki/About/MadWifi. Recurso consultado por última vez en agosto de 2009.

[21] The MadWiFi Project. *HAL*. Página disponible en:http://madwifi-project.org/wiki/About/HAL. Recurso consultado por última vez en marzo de 2010.

[22] The MadWifi Project. *ath5k*. Página disponible en:http://madwifi-project.org/wiki/About/ath5k. Recurso consultado por última vez en marzo de 2010.

[23] The Network Simulator – ns – 2. Página disponible en:http://www.isi.edu/nsnam/ns. Recurso consultado por última vez en marzo de 2010.

[24] F. Gringoli y L. Nava. "OpenFWWF – Open FirmWare for WiFi networks". Página disponible en: http://www.ing.unibs.it/openfwwf/. Recurso consultado por última vez en septiembre de 2009.

[25] "*BCM-Specs, 802.11 Registers*". Página disponible en: http://bcm-v4.sipsolutions.net/802.11/Registers. Recurso consultado por última vez en sptiembre de 2009.

[26] Comando Make. Página disponible en: http://linux.about.com/library/cmd/blcmdl1_make.htm. Recurso consultado por última vez en julio de 2009.

[27] GCC, THE GNU COMPILER COLLECTION. Boston, MA 2006. Página disponible en: http://gcc.gnu.org/. Recurso consultado por última vez en julio de 2009.

[28] "*Busybox: The Swiss Army Knife Of Embedded Linux".* USA, 1999 - 2006. Página disponible en: http://www.busybox.net/. Recurso consultado por última vez en julio de 2009.

[29] "*LuCI*". Página disponible en: http://luci.subsignal.org/. Recurso consultado por última vez en julio de 2009.

[30] "How to Free your WRT54* or brcm47xx WiFi Router". Página disponible en: http://groups.fsf.org/wiki/Freest_hardware_comparisons/brcm47xx_HOWTO_free_your_wrt54*/brcm47xx_router. Recurso consultado por última vez en septiembre de 2009.

[31] G. Court. "*Installing OpenWRT on the Linksys WRT54GL*". Página disponible en: http://garycourt.com/blog/post/installing-openwrt-on-the-linksys-wrt54gl/. Recurso consultado por última vez en junio de 2009.

[32] "*Installing OpenWrt via TFTP*". Página disponible en: http://wiki.openwrt.org/oldwiki/openwrtdocs/installing/tftp. Recurso consultado por última vez en septiembre de 2009.

[33] "*boot_wait - What it is, and how it Works*". Página disponible en: http://wiki.openwrt.org/oldwiki/openwrtdocs/bootwait. Recurso consultado por última vez en septiembre de 2009.

[34] Instruction Set Summary for the core revision. Disponible en el archivo README de Assembler en b43-tools. Recurso consultado por última vez en diciembre de 2009.

[35] F. Gringoli y L. Nava. "*OpenFWWF – Specifications*". Página disponible en: http://www.ing.unibs.it/openfwwf/specification.php. Recurso consultado por última vez en diciembre de 2009.

[36] F. Gringoli y L. Nava. "*OpenFWWF – Registers*". Página disponible en: http://www.ing.unibs.it/openfwwf/condition_reg.php. Recurso consultado por última vez en diciembre de 2009.

[37] F. Gringoli y L. Nava. "*OpenFWWF – Internal Hardware Registers*". Página disponible en: http://www.ing.unibs.it/openfwwf/IHR.php. Recurso consultado por última vez en diciembre de 2009.

[38] F. Gringoli y L. Nava. "*OpenFWWF- Transmission*". Página disponible en: http://www.ing.unibs.it/openfwwf/transmission.php. Recurso consultado por última vez en diciembre de 2009.

[39] F. Gringoli y L. Nava. "*OpenFWWF – Reception*". Página disponible en: http://www.ing.unibs.it/openfwwf/reception.php. Recurso consultado por última vez en diciembre de 2009.

[40] Linux Wireless – b43. Página disponible en: http://linuxwireless.org/en/users/Drivers/b43. Recurso consultado por última vez en junio de 2009.

[41] IBM – Intel. "*Defining WLAN Performance Parameters*". 2004. Página disponible en: http://www.intel.com/network/connectivity/resources/doc_library/white_papers/304702.pdf. Recurso consultado por última vez en octubre de 2009.

[42] Wireshark Foundation. "*Display Filter Reference*". Página disponible en: http://www.wireshark.org/docs/dfref/. Recurso consultado por última vez en junio de 2009.

[43] T. d'Otreppe. "*Aircrack-ng*". Página disponible en: http://www.aircrack-ng.org/doku.php. Recurso consultado por última vez en octubre de 2009.

[44] T. d'Otreppe. "*Airmon-ng*". Página disponible en: http://www.aircrack-ng.org/doku.php?id=airmon-ng. Recurso consultado por última vez en octubre de 2009.

[45] B. Carrera. "*Iperf – The Easy Tutorial*". Página disponible en: http://openmaniak.com/iperf.php. Recurso consultado por última vez en enero de 2009.

[46] b. Sharif. "Elastix Without Tears". 2007. Recurso disponible en: http://elastixconnection.com/downloads/elastix_without_tears.pdf. Recurso consultado por última vez en diciembre de 2009.

[47] CounterPath Solutions, Inc. "*X-Lite 3.0 User Guide*". Recurso disponible en: http://www.counterpath.com/assets/files/191/X-Lite3.0_UserGuide.pdf. 2006. Recurso consultado por última vez en enero de 2009.

[48] Atractel. "*Zoiper. Userguide. Edition Version 2.0 for Windows*". 2008. Recurso disponible en: http://www.zoiper.com/downloads/Zoiper_2.0_Free_Manual.pdf. Recurso consultado por última vez en enero de 2009.

[49] A. de Lattre, M. Ariberti, A. Daoud y C. Stenac. "VideoLAN HOWTO". The VideoLAN Project. 2003 http://www.ibiblio.org/pub/linux/docs/HOWTO/other-formats/pdf/VideoLAN-HOWTO.pdf. Recurso consultado por última vez en diciembre de 2009.

[50] Ubiquiti Networks. ""PS2 Datasheet". Recurso disponible en: http://ubnt.com/downloads/ps2_datasheet.pdf. Recurso consultado por última vez en enero de 2009.

[51] TIER. "Wireless: DriverInfo". Grupo TIER Technology and Infrastructure for Emerging Regions de la Universidad de California en Berkley. página disponible en: http://tier.cs.berkeley.edu/wiki/Wireless:DriverInfo. Recurso consultado por última vez en diciembre de 2009.

[52] Protocols Group. "*Madwifi/Atheros Wireless Linux Driver Users Guide*". 2006. http://www.esiee.fr/~tds/Labo/PlateformeEtLabo/Labo2401/documents/Madwifi_Atheros.pdf. Recurso consultado por última vez en diciembre de 2009.

[53] "*The importance of the ACK Timeout*". Página disponible en: http://freenet-antennas.com/PHP-Nuke/modules.php?name=News&file=article&sid=22. Recurso consultado por última vez en diciembre de 2009.