

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previa a la obtención del título de:
INGENIERO ELECTRÓNICO**

**TEMA:
ANÁLISIS COMPARATIVO DEL DESEMPEÑO DEL ESTÁNDAR IEEE
802.11ac RESPECTO AL IEEE 802.11n A TRAVÉS DE SIMULACIÓN
NUMÉRICA APOYADA POR SOFTWARE.**

**AUTOR:
DANNY ALEJANDRO ORTIZ PUGA**

**DIRECTOR:
CARLOS AUGUSTO CUICHÁN MORALES**

Quito, abril de 2015

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO
DEL TRABAJO DE TITULACIÓN**

Yo, autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaro que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del autor.

Quito, abril de 2015

Danny Alejandro Ortiz Puga
C.I. 172301187-8

DEDICATORIA

Dedico el presente trabajo a mis padres, que con su apoyo incondicional y abnegación, fueron quienes me incentivaron a terminar mis estudios y supieron sembrar en mí el anhelo de superación para hacer posible la obtención de este título.

AGRADECIMIENTO

Agradezco a mis distinguidos maestros y guías que con su conocimiento y apoyo hicieron posible la realización del presente trabajo de grado.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1.....	2
MARCO TEÓRICO	2
1.1 Análisis del problema	2
1.2 Historia de la transmisión inalámbrica	4
1.3 Redes inalámbricas	5
1.3.1 Red de área local inalámbrica (WLAN).....	5
1.3.2 Arquitectura de WLAN.....	6
1.3.2.1 Capa física	7
1.3.2.2 Capa enlace.....	9
1.3.3 Tipos de WLAN	11
1.3.3.1 Modo Ad-hoc (IBSS) o Punto a Punto	11
1.3.3.2 Modo Infraestructura (BSS) o Punto Multipunto	12
1.3.3.2.1 Tipos de topologías en modo infraestructura:	13
1.3.3.3 Redes por el grado de autenticación	15
1.3.3.3.1 Red de acceso público.....	16
1.3.3.3.2 Red privada	16
1.4 Características de las redes inalámbricas	16
1.4.1 Modulación	16
1.4.1.1 Tipos de Modulación.....	16
1.4.2 Distancia.....	18
1.4.3 Cobertura.....	18
1.5 Estructura de los enlaces	20
1.5.1 Transmisores	20
1.5.2 Receptores	21
1.5.3 Seguridad	21
1.5.3.1 Proceso de Asociación y seguridad	21
1.5.3.2 Tipos de seguridades	23
1.6 Parámetros de transmisión.....	24
1.6.1 Bit error rate (BER).....	24
1.6.2 Retardo	25
1.6.3 Jitter.....	26
1.6.3.1 Jitter de Fase y Amplitud.....	27

CAPÍTULO 2.....	31
ESTÁNDARES IEEE 802.11ac y IEEE 802.11n.....	31
2.1 Descripción del estándar 802.11n	31
2.1.1 Modulación y Velocidad del estándar 802.11n.....	32
2.1.2 Transmisor y Receptor del estándar 802.11n.....	36
2.2 Descripción del estándar 802.11ac	40
2.2.1 Modulación y Velocidad 802.11ac	41
2.2.2 Transmisor y receptor 802.11ac.....	43
2.3 Migración de 802.11n a 802.11ac (FLUKE networks)	48
2.4 Sustentación Legal	49
2.4.1 IEEE (Instituto de Ingenieros Eléctricos y Electrónicos)	49
2.4.2 Normativa Internacional.....	50
2.4.2.1 Estándar IEEE 802.11n y IEEE 802.11ac	50
2.4.3 Normativa nacional	50
2.4.3.1 Uso del estándar IEEE 802.11n y IEEE 802.11ac en el Ecuador.....	51
2.5 Parámetros de diseño.....	52
2.6 Programación orientada a objetos	52
2.6.1 Lenguaje C++.....	53
2.6.2 Lenguaje NED.....	53
2.6.2.1 Módulos simples.....	54
2.6.2.2 Módulos compuestos.....	55
2.6.2.3 Redes	56
2.6.3 Pasos para ejecutar una aplicación tipo NED (Martínez, 2010)	56
2.7 OMNET++	56
2.8 NS3.....	59
2.8.1 Conceptos generales.....	60
2.8.1.1 Nodo	60
2.8.1.2 Aplicación	60
2.8.1.3 Canal.....	61
2.8.1.4 Dispositivo net.....	61
2.8.1.5 Topologías de ayuda.....	62
2.9 Selección del lenguaje de programación.	62

CAPÍTULO 3.....	64
SIMULACIÓN DEL DESEMPEÑO.....	64
3.1 Esquema de la simulación	64
3.2 Caracterización de redes ideales.....	65
3.2.1 Redes Ideales IEEE 802.11n	66
3.2.2 Redes Ideales IEEE 802.11ac	67
3.3 Caracterización de redes reales	67
3.3.1 Redes Reales IEEE 802.11n.....	67
3.3.2 Redes Reales IEEE 802.11ac	69
3.4 Entorno de simulación.....	71
3.4.1 Esquema de desarrollo de redes ideales	72
3.4.2 Esquema de desarrollo de redes reales	73
3.5 Red de simulación con el estándar IEEE 802.11n ideal.....	74
3.5.1 Programación del nodo	76
3.5.2 Transmisor o AP	78
3.5.3 Módulo Wireless	79
3.5.4 Módulo de red	81
3.5.5 Programación de módulos en C++.....	83
3.5.6 Implementación de la clase módulo Nodo.....	84
3.5.7 Programación del módulo AP	86
3.6 Red de simulación con estándar IEEE 802.11ac ideal	88
3.7 Red de simulación con estándar IEEE 802.11n real.....	88
3.7.1 Módulo Canal.....	89
3.7.2. Programación C++ del módulo canal	89
3.8 Red de simulación con estándar IEEE 802.11ac real	91
 CAPÍTULO 4.....	 93
ANÁLISIS DE RESULTADOS	93
4.1 Adquisición de datos en redes ideales	93
4.1.1 Proceso para ejecutar las simulaciones	94
4.1.2 Adquisición de datos de IEEE 802.11n ideal.....	96
4.1.3 Simulación IEEE 802.11ac ideal.....	98
4.2 Adquisición de datos redes reales.....	101
4.2.1 Simulación IEEE 802.11n real	102

4.2.2	Simulación IEEE 802.11ac real	105
CONCLUSIONES.....		108
RECOMENDACIONES.....		110

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Estándar 802.11n vs 802.11ac.	3
<i>Figura 2.</i> Red Ad-hoc.	12
<i>Figura 3.</i> Estructura BSS.	13
<i>Figura 4.</i> Topología estrella.	13
<i>Figura 5.</i> Red Punto a Punto.	14
<i>Figura 6.</i> Red enlazada por repetidores.	14
<i>Figura 7.</i> Red de tipología malla.	15
<i>Figura 8.</i> Cobertura antena omnidireccional.	19
<i>Figura 9.</i> Cobertura antena direccional o sectorial.	19
<i>Figura 10.</i> Relación de velocidades de un estándar combinado.	20
<i>Figura 11.</i> Relación de velocidades de dos estándares combinados.	20
<i>Figura 12.</i> Jitter de fase.	27
<i>Figura 13.</i> Jitter de amplitud.	27
<i>Figura 14.</i> Diagrama de bloques de transmisor 802.11n.	36
<i>Figura 15.</i> Transmisor 802.11ac por bloques.	44
<i>Figura 16.</i> Receptor 802.11ac por bloques.	44
<i>Figura 17.</i> Bloque codificador BCC.	45
<i>Figura 18.</i> Pantalla de inicio de OMNet++ 4.1	58
<i>Figura 19.</i> Estructura de módulos OMNet++	58
<i>Figura 20.</i> Esquema de módulos .NED	71
<i>Figura 21.</i> Esquema de módulos .NED con parámetros fundamentales.	73
<i>Figura 22.</i> Esquema de módulos .NED con parámetros fundamentales.	74
<i>Figura 23.</i> Desarrollo de un programa OMNet.	75
<i>Figura 24.</i> Módulo y submódulo .NED	76
<i>Figura 25.</i> Módulo de red Wireless.	82
<i>Figura 26.</i> Interfaz de una red real con el canal.	88
<i>Figura 27.</i> Carpeta de sistema msys.	94
<i>Figura 28.</i> Archivo ejecutable msys.	95
<i>Figura 29.</i> Interfaz de Omnet++	95
<i>Figura 30.</i> Proyectos para la simulación.	96
<i>Figura 31.</i> Simulación de red 802.11n ideal.	96
<i>Figura 32.</i> Simulación de red 802.11ac ideal.	99

<i>Figura 33.</i> Simulación de red 802.11n real.....	102
<i>Figura 34.</i> Simulación de red 802.11ac real.	105
<i>Figura 35.</i> Locación del software OMNet++	119
<i>Figura 36.</i> Terminal cmd.	120
<i>Figura 37.</i> Terminal administrativa de OMNet++.....	120
<i>Figura 38.</i> Ventana de inicio de Interfaz Gráfico de OMNet++	121
<i>Figura 39.</i> Ventana de trabajo de OMNet++.	121
<i>Figura 40.</i> Selección de la carpeta inet, examples.....	122
<i>Figura 41.</i> Selección de la carpeta red 80211.....	122
<i>Figura 42.</i> Selección de carpeta y archivo de simulación.	123
<i>Figura 43.</i> Botones de activación de proyecto.....	124
<i>Figura 44.</i> Ventana de simulación GNED.....	124
<i>Figura 45.</i> Dirección del archivo de adquisición de datos.....	125

ÍNDICE DE TABLAS

Tabla 1. Índice MCS de la norma 802.11n	34
Tabla 2. Indicador Bit y Modulación	39
Tabla 3. Tasa de datos según el ancho de banda en 802.11ac	42
Tabla 4. Tasa de datos según MCS en 802.11ac	43
Tabla 5. Indicador Bit y Modulación	47
Tabla 6. Parámetros de redes ideales	70
Tabla 7. Parámetros de redes reales	70
Tabla 8. Parámetros de simulación de red 802.11n ideal.....	87
Tabla 9. Parámetros de simulación de red 802.11ac ideal	88
Tabla 10. Parámetros de simulación de red 802.11n real	91
Tabla 11. Valores de evaluación de la red 802.11n real	91
Tabla 12. Parámetros de simulación de la red 802.11ac real	92
Tabla 13. Valores de evaluación de la red 802.11ac real.....	92
Tabla 14. Datos obtenidos de la simulación 802.11n ideal.....	98
Tabla 15. Datos obtenidos de la simulación 802.11ac ideal	100
Tabla 16. Datos obtenidos de la simulación 802.11n real.....	104
Tabla 17. Datos de retardos en 802.11n real.....	104
Tabla 18. Datos obtenidos de la simulación 802.11ac real	106
Tabla 19. Datos de retardos en 802.11ac real	107

ÍNDICE DE ANEXOS

Anexo 1. Programación Gned de los módulos de interfaz gráfica.....	111
Anexo 2. Programación C++ de los módulos.....	114
Anexo 3. Creación del módulo canal.....	116
Anexo 4. Manual de ejecución de programa.....	119

RESUMEN

El vertiginoso desarrollo tecnológico que existe en la actualidad se da debido a que se vive en la denominada “era de la información”, por lo tanto, así como existe tanta información, también se la debe transmitir de forma real, fidedigna y segura con técnicas y métodos de codificación que aceleren los procesos de transmisión y recepción.

El desarrollo de tanta información hace que exista más y más estudios de como transmitir esta información con rapidez, seguridad y fidelidad, por este motivo se ha desarrollado el estándar IEEE 802.11ac el cual es el sucesor de 802.11n. Pero el problema se centra en realizar un estudio comparativo de los estándares para verificar las mejoras del estándar sucesor en relación al 802.11n.

Para efectuar el análisis comparativo entre las normas se usará un paquete informático de simulación de redes por medio del software OMnet++, que es un simulador programable de redes. Esto permitirá que se tomen datos de las redes de transmisión en los dos estándares para efectuar la comparación de los mismos.

Por lo tanto es importante realizar el análisis de la simulación de las redes en los distintos estándares ya que se puede llegar a conclusiones sobre cómo y cuándo usar cada normativa y bajo qué condiciones, ya que en la actualidad existe una gran potencialidad de los equipos de transmisión y recepción. También al dar mejor información sobre el desempeño y las condiciones de trabajo de los estándares se puede dar una mejor visión a los usuarios, para saber qué tipo de estándar usar cuando requieran dispositivos de red inalámbricos.

ABSTRACT

The rapid technological development that exists today occurs because we live in so-called "age of information", therefore, just as there is so much information, also it must be transmitted in real, reliable and safe techniques and coding methods to accelerate the processes of transmission and reception.

The development of such information means that there are more and more studies on how to transfer this information quickly, reliably and fidelity, which is why we have developed the IEEE 802.11ac standard which is the successor of 802.11n. But the problem focuses on a comparative study of the standards to verify the successor improvements in relation to the 802.11n standard.

To perform a comparative analysis between the standards will be used a computer network simulation package by software OMnet++, which is a programmable network simulator. This will allow data transmission networks are taken in the two standards for comparison thereof.

Therefore it is important to perform simulation analysis of networks in different standards as it can draw conclusions about how and when to use each standard and under what conditions, as currently there are great potential devices of transmission and reception. Also to provide better information on performance and working conditions standards can give better vision for users to know what standard to use when requiring wireless network devices.

INTRODUCCIÓN

Los sistemas telemáticos de comunicación a nivel mundial están basados en protocolos y estándares con el objetivo de establecer las normas y estatutos para una correcta y eficiente funcionalidad de los mismos. El estándar IEEE 802.11ac se considera la evolución lógica de su homólogo IEEE 802.11n dada por la revisión y mejora tecnológica de este último. Por esta causa en la actualidad varios países comienzan a dar pasos hacia la sustitución tecnológica del estándar IEEE 802.11n por el nuevo IEEE 802.11ac, en función de una comunicación más eficiente.

Las razones por las que se plantea la propuesta de investigación son por el uso de los nuevos dispositivos basados en el nuevo estándar IEEE 802.11ac. Muchas empresas al conocer del nuevo estándar es muy obvio que migrarán sus redes y aplicaciones al nuevo protocolo por sus mejoras, pero es necesario tener mayor información sobre el desempeño en varias condiciones de trabajo de este nuevo estándar aprobado por la IEEE. Otra razón para esta investigación es la necesidad de priorizar recursos ya que como se ha mencionado, actualmente se usan en su mayoría dispositivos con el estándar IEEE 802.11n, más al comprobar su desempeño en distintas condiciones se puede verificar que dispositivos son apropiados a los requerimientos de los usuarios y condicionamientos dados.

En la actualidad existen muy pocos estudios comparativos entre estas tecnologías debido a la reciente incorporación de la tecnología IEEE 802.11ac, con este análisis se pretende comparar el desempeño de ambos estándares para largas distancias en ambientes outdoor y en función de los resultados obtenidos mediante la simulación apoyada por software al final de este estudio poder determinar en qué aspectos uno es superior a otro respecto al Bit Error Rate (BER), y retardos en las condiciones mencionadas anteriormente. El fin a conseguir es dar una guía a los usuarios de cuál es el mejor estándar y bajo qué condiciones de trabajo se los puede utilizar.

CAPÍTULO 1

MARCO TEÓRICO

1.1 Análisis del problema

Existe una alta demanda de velocidades de transmisión de datos en las redes informáticas y una gran demanda de equipos que se integran a redes inalámbricas mundialmente. Como la norma 802.11ac es reciente, Los usuarios de redes inalámbricas no tienen el conocimiento adecuado sobre la funcionalidad, la conveniencia y las desventajas de esta norma, con el fin de tener como perspectiva efectuar o no una migración de equipos o continuar con los que se tenían anteriormente, como los que trabajan con la norma 802.11n.

Para realizar el análisis del desempeño de la norma 802.11ac, en comparación de la norma 802.11n, se puede efectuar de forma real y virtual. Las dos formas tienen ventajas y desventajas. La primera se centra en la adquisición de datos por medio de transmisión y recepción con equipos reales bajo este estándar. Esta forma es la más segura ya que permite verificar realmente los datos bajo condiciones reales, pues como se trata de enlaces outdoor, se debe tomar en cuenta que las señales se envían por medio de cierta geografía, donde en algunos lugares habrá muchos obstáculos y en otros no. En el mejor de los casos donde exista línea de vista, otros parámetros como distancia, potencia y modulación serán factores imprescindibles para poder receptar el 100% de la señal. La desventaja en esta forma de adquirir los datos es que los equipos que trabajan con esta norma, en el mercado todavía no existen o los pocos que hay son muy costosos.

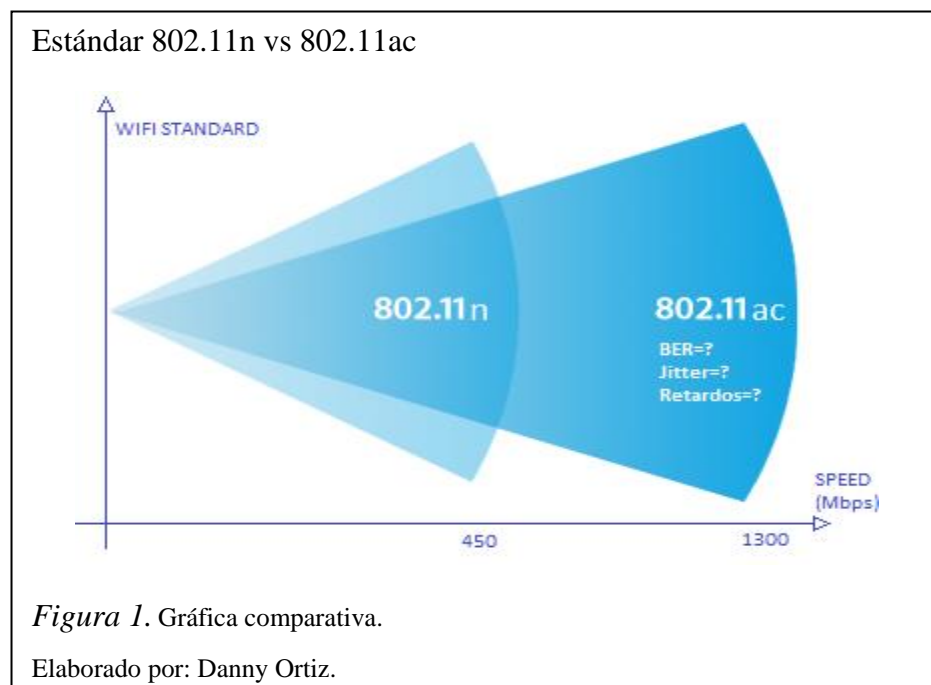
La otra forma es la virtual, que se realiza por medio de simulación con un paquete informático, la desventaja es que los condicionamientos de las redes como interferencia, ruido, etc, son simulados y solo son una aproximación de la realidad, además para efectuar la simulación se debe entender muy claramente la programación de los objetos y la interacción que tendrán dentro de la simulación.

Al realizar el análisis comparativo de la norma 802.11ac con la norma 802.11n se desea tener y brindar mayor información sobre el trabajo y desempeño del estándar

802.11ac a todos los usuarios que quieran trabajar bajo esta nueva norma aprobada por la IEEE. En el mercado local existen algunos equipos que ya trabajan con esta normativa IEEE 802.11ac, pero la información es muy limitada, así que la gran mayoría de usuarios tomarán un tiempo más para migrar a esta nueva tecnología por lo que esto retrasa en el país la actualización tecnológica.

Otro problema que se deriva de la falta de información, es que no se conoce las ventajas de esta normativa y bajo qué condiciones se puede usar en conjunto con la normativa IEEE 802.11n, con el fin de aprovechar los recursos y evitar la eliminación innecesaria de equipos con el anterior estándar sino permitir la escalabilidad en procesos de transmisión y recepción.

La información que se requiere exclusivamente está basada en la funcionalidad de la norma 802.11ac en comparación con la norma 802.11n. El análisis debe basarse en una comparación porque esto permitirá que se tenga una mejor idea del trabajo de esta nueva norma y no solo enfocarse en las altas velocidades que ofrece 802.11ac sino también comparar otros factores como se observa en la figura 1.



Además se la compara con su antecesora 802.11n, porque esta tiene parámetros parecidos a la norma 802.11ac, como modulación, canales, ancho de banda y otros más.

1.2 Historia de la transmisión inalámbrica

La comunicación inalámbrica se basa en el principio del electromagnetismo. A fines del siglo XIX el físico escocés James Clerk Maxwell, definió un conjunto de ecuaciones que representaban sus ideas básicas sobre la electricidad y el magnetismo. Después en 1884 el físico alemán Heinrich Rudolf Hertz usando las leyes de Maxwell, descubrió que las ondas de radio reaccionaban de manera similar a la luz, pero con una longitud de onda muchísimo mayor que el de la luz (Murillo, 2007).

Tras corto tiempo en 1893, el ingeniero Nikola Tesla, desarrolló el motor de inducción, con esto descubrió que se podía usar la inducción electromagnética para emitir mensajes. Guillermo Marconi invento la forma de enviar mensajes a más de una milla por medio de antenas y para ello utilizó el alfabeto Morse desarrollado por Samuel Morse. Estos descubrimientos dieron lugar al desarrollo de la radio. (VSAT, 2014)

Posteriormente se descubrió la forma de emitir señales continuas por medio de antenas, logrando obtener transmisiones continuas como audio y video. Después se ve la necesidad de transmitir a mayores distancias con lo que aparece la modulación. Surge la necesidad de ancho de banda y por lo tanto los arreglos de modulación y frecuencia para las transmisiones, entonces aparece AM (amplitud modulada) y más adelante FM (frecuencia modulada) (Murillo, 2007).

Tras descubrir los semiconductores, en 1956 aparece el transistor, con lo que la electrónica empieza su desarrollo y aparece la miniaturización de los circuitos, por medio de los circuito integrados, los micro circuitos y ahora los nano circuitos, reduciendo espacio físico y energía, logrando maximizar las aplicaciones tras lo cual se han alcanzado más aplicaciones y altas potencias de transmisión.

A la par del desarrollo electrónico aparecen los conceptos como la comunicación digital, con el concepto de programación de computadores y a su vez la conversión de datos, analógico – digital. Con esto se crean nuevas formas de aplicaciones para las transmisiones de datos y ya no solo por los cables sino de forma inalámbrica, así

aparecen parámetros relacionados a las transmisiones como seguridad, modulación digital, velocidades, calidad de la transmisión, etc. (VSAT, 2014)

Tras la aparición del internet en la década de los 80s se hace necesario usar nuevas formas de transmisión, ya que inicialmente era solo por cable coaxial, luego fue el cable UTP, la fibra óptica y después también por medio de antenas. Desde ese momento la IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) ve la necesidad del desarrollo de estándares que rigen las velocidades, modulación, anchos de banda, con el fin de normalizar las transmisiones de todo tipo en lo relacionado a datos, voz y video, por medio de la red.

Estos antecedentes dan lugar a que la IEEE LAN/MAN Comité de Normas, pongan su empeño en el desarrollo de normas para transmisiones en las bandas de frecuencias 2,4; 3,6; 5,8 y 60 GHz, a este estándar se lo denomina 802.11. Esta tecnología tiene sus orígenes en 1985, tras un fallo de la Comisión Federal de Comunicaciones de los Estados Unidos que aprobaron las bandas de frecuencias mencionadas anteriormente como bandas ISM (Industriales, científicas y médicas), para su uso libre y sin licencia. (E-CENTRO, 2014)

La versión base del estándar 802.11 fue lanzado en 1997. Tras varios estudios el estándar 802.11 ha ido cambiando y mejorando sus aplicaciones, a tal punto que se han generalizado las versiones de 802.11a, 802.11b, 802.11g, 802.11n y 802.11ac en la actualidad, las que trabajan en 2,4 GHz y 5,8 GHz. Estas normas constituyen la base de los productos de red inalámbricos (E-CENTRO, 2014).

1.3 Redes inalámbricas

1.3.1 Red de área local inalámbrica (WLAN)

WLAN es el acrónimo que se da a una red de área local pero inalámbrica (Wireless Local Area Network), en las que se aprovecha las ondas radioeléctricas para conectar redes de computadoras e Internet.

Este tipo de redes en su inicio eran de tipo doméstico pues se usaban routers de baja potencia que como máximo llegaban a conectar a dispositivos wireless a una

distancia entre 15 y 30 metros como es el caso de 802.11a con frecuencia de 5Ghz, pero con la optimización de recursos y dispositivos muchos routers que eran indoor, ahora son capaces de alcanzar distancias hasta de 100 metros sin dificultad ya que operan a una frecuencia de 2,4Ghz la cual tiene mucha más penetración que la frecuencia de 5Ghz dando así 3 veces más alcance con esta frecuencia, además la implementación de la tecnología MIMO (Múltiples Entradas Múltiples Salidas) permite hoy en día que los routers indoor operen satisfactoriamente dentro de hogares y oficinas, a pesar que la banda de 2,4Ghz ya está algo saturada, sigue actualmente en funcionamiento gracias a sus 11 canales, que permiten eliminar posibles interferencias con otros routers ubicados en los alrededores.

Tal es el desarrollo de la comunicación en redes de área local inalámbrica, que ya no son solo de área local sino también metropolitanas, pues, existen APs (Access Point) capaces de potencializar la transmisión y enviar información a distancias de 5 a 10Km a la redonda con ayuda de antenas de alta capacidad con potencias entre 600mW, 1W y 2W o 16Dbi a 17Dbi, en frecuencias de 2,4Ghz y 5,8Ghz. O dispositivos que se comunican punto a punto en distancias de 50Km ya que usando antenas directivas se puede focalizar toda la energía en una sola dirección y de esta forma lograr mucho más alcance que con las antenas omnidireccionales que generalmente se utilizan, además se utilizan potencias de 1W en frecuencias de 2,4Ghz y 5,8Ghz y con tecnología MU-MIMO (Múltiples Usuarios Múltiples Entradas Múltiples Salidas) ya que esta tecnología aumenta el alcance y la penetración de la señal.

Entonces el gran desarrollo de las WLAN ha permitido el desarrollo tecnológico y empresarial de algunas sociedades, pues esta ha fomentado el desarrollo de ISPs (Internet Service Provider, 2014).

1.3.2 Arquitectura de WLAN

Las transmisiones de todos los sistemas de comunicación se centran en el modelo OSI y por supuesto 802.11 no debería ser la excepción, por lo tanto WLAN también tiene su modelo basado en OSI. Es decir, a diferencia de las comunicaciones por cable en

las redes WLAN solo se usan dos capas, la primera, la capa física que define la modulación de las ondas de radio y las características de señalización para la transmisión de datos mientras que la segunda, la capa de enlace de datos que define la interfaz entre el bus del equipo y la capa física.

1.3.2.1 Capa física

En la capa física, WLAN usa el acceso inalámbrico, de tal manera que se definen tres tipos de capas físicas. Originalmente en 802.11 se definen dos espectros de radio y uno de infrarrojos. Estos espectros trabajan en las bandas de radio frecuencia de 2.4GHz ISM, reconocidas por reguladores internacionales, como FCC (USA), ETSI (Europa), y la MKK (Japón), para operaciones sin necesidad de licencia, para usos científicos, militares e industriales.

Bajo el estándar original 802.11 se usaba FHSS (Frequency hopping spread spectrum) o DSSS (Direct sequence spread spectrum) que son técnicas de modulación para la transmisión q trabajan a velocidades de 1 y 2 Mbps. Pero un problema que se daba en estos tipos de transmisión era la inoperancia de los mecanismos de modulación que ya no pueden interoperar entre ellos.

En el caso de la técnica FHSS o de espectro ensanchado por salto en frecuencia, en la banda de 2.4 GHz se dividía en 75 subcanales de 1MHz. Para efectuar la transmisión y recepción, los dos equipos tanto el emisor como el receptor definen los subcanales sobre los que se enviará la información y también el tipo de salto de frecuencia que se usará. En la red 802.11 para distintas transmisiones ocurrían un distinto salto de frecuencia, esto garantizaba la probabilidad de que dos emisores no usen el mismo subcanal simultáneamente, así evitar las interferencias por sobre posición de frecuencias.

Algo positivo de la técnica FHSS es que permite un enlace simple, por otra parte, tiene la desventaja que la velocidad es muy baja (máximo de 2Mbps). Por esta razón se restringe los canales a un ancho de banda máximo de 1MHz. Así que esta técnica obliga a que se use la banda de 2.4GHz por completo, para lo que la trama debe tener

una cabecera bastante grande para efectuar los saltos necesarios evitando interferencia de canales.

Por otro lado la técnica DSSS o de espectro ensanchado por frecuencia directa, divide la banda de 2.4GHz en 14 subcanales. Para ciertos países solo se dispone 11 canales, con un ancho de banda de 25MHz, con esto se garantiza que no existan interferencias. De esta manera al menos 3 subcanales pueden trabajar sin problema dentro de la misma transmisión. Para evitar la interferencia de un canal, se usa una técnica llamada chipping. Cada bit de datos es convertido en una serie de patrones redundantes de bit llamados chips. La redundancia inherente de cada chip combinado con el envío de la señal a través del canal da como resultado una forma de detección y corrección de errores; incluso si parte de la señal está mal, se puede recuperar en muchos casos, minimizando la necesidad de retransmisión.

El estándar original 802.11 DSSS especifica una secuencia de 11 bits para la secuencia chipping que es también conocida como secuencia Barker, y sirve para codificar todos los datos enviados a través del aire. Cada secuencia de 11 bits representa un solo bit de datos (1 o 0), y se convierte en una forma de onda, llamada símbolo, que puede ser enviada a través del aire. Estos símbolos son transmitidos con una tasa de símbolos de 1MSps (1 millón de símbolos por segundo) usando la técnica BPSK (Binary Phase Shift Keying). En el caso de 2 Mbps, se usa una implementación más sofisticada llamada QPSK (Quadrature Phase Shift Keying), que dobla la tasa de datos que soporta BPSK mejorando la eficiencia en el uso del ancho de banda.

Para incrementar la tasa de datos, se desarrollaron técnicas de codificación avanzadas. Mejor que las dos secuencias Barker de 11 bits, el 802.11b especifica la modulación Complementary Code Keying (CCK), que consiste en un conjunto de 64 palabras código de 8 bits. Como conjunto, estas palabras código tienen propiedades matemáticas únicas que les permiten distinguirse correctamente uno de otra por un receptor incluso en presencia de un ruido importante e interferencia multicamino. La tasa de 5.5 Mbps usa el CCK para codificar 4 bits por portadora, mientras que la tasa de 11Mbps codifica 8 bits por portadora. Ambas velocidades usan la técnica de modulación QPSK y señal a 1.375MSps. Así es como se obtiene altas tasas de datos.

1.3.2.2 Capa enlace

La capa de enlace del estándar 802.11 consiste en dos subcapas: Logical Link Control (LLC) y Media Access Control (MAC).

El estándar 802.11 utiliza el mismo LLC (control de enlace de datos) que el 802.2, pero el nivel MAC (Control de Acceso al Medio) es diferente.

En las redes inalámbricas se tienen 3 tipos de tramas:

- Tramas de administración: estas tramas sirven para transmitir información de administración.
- Tramas de datos: sirven para transmitir datos.
- Tramas de control: tramas que se usan para controlar el acceso al medio (RTS, CTS y ACK).

El 802.11 usa un protocolo un poco modificado del CSMA/CD (Protocolo que regula el acceso al medio, además detectan y tratan las colisiones) al que se le conoce como CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), o el DCF (Distributed Coordination Function). CSMA/CA permite evitar las colisiones usando ACK explícitos, o así la estación que recibe los datos envía un paquete ACK si estos llegan correctamente.

El protocolo CSMA/CA funciona de la siguiente manera:

1. Una estación que desee transmitir escucha si el medio está ocupado, si el canal está libre, la estación puede transmitir. Luego espera un tiempo aleatorio y entonces transmite al medio si este todavía está libre. Si por el contrario el canal está ocupado, la estación deja la transmisión para más adelante.
2. Si el paquete transmitido se recibe correctamente o efectúa la comprobación del CRC de la trama de administración, la estación receptora envía un ACK.
3. Si la estación emisora recibe el ACK se completa el proceso. Caso contrario si el ACK no es detectado por la estación emisora, porque el paquete original no ha sido recibido correctamente o el ACK se ha perdido, se asume que se ha producido una colisión y el paquete de datos se retransmite en otro tiempo aleatorio.

Si no se utilizaran ACKs explícitos cabría la posibilidad de que varias estaciones estuvieran sondeando el canal y al detectar que está libre simultáneamente intentarán transmitir al mismo tiempo, provocando colisión.

Otro problema que se ha de resolver es el problema conocido como “ruido escondido o nodo oculto”, que consiste en que dos estaciones en lados opuestos del punto de acceso pueden escuchar actividad del punto de acceso pero no de ellas, debido principalmente a la distancia o alguna obstrucción, lo que provocaría colisión. Para resolver este problema, 802.11 especifica un protocolo RTS/CTS (Request To Send, Petición de envío y Clear To Send, permiso para enviar) opcional en la capa MAC.

Este mecanismo funciona de la siguiente manera:

Una estación esperando para transmitir envía un paquete de control RTS (que contiene el ACK) y espera que el punto de acceso le conteste con un CTS si el medio está libre. Como todas las estaciones de la red pueden escuchar al AP, el CTS retrasa cualquier intento de transmisión.

Para servicios con límite de tiempo, por ejemplo vídeo y voz, existe una funcionalidad opcional llamada Función de Punto de Coordinación (PCF), que utiliza un acceso de mayor prioridad, así el AP controla el acceso al medio y emite peticiones de sondeo a las estaciones para transmitir datos. El punto de acceso sondeará cada estación en busca de datos, y después de un tiempo cambia a la siguiente estación. Ninguna estación puede transmitir hasta que sea elegida, y las estaciones reciben datos del punto de acceso solamente cuando sean elegidas. Por lo que el PCF da a cada estación un turno para transmitir en un momento predeterminado, garantizando un retraso o latencia máxima. El hecho de un AP tenga el control de acceso al medio y sondee las estaciones hace que no sea eficaz para redes grandes.

Si el modo de operación es el modo infraestructura con el PCF habilitado el tiempo se divide y el sistema puede estar en modo PCF y en el que está en el modo DCF.

El mecanismo de fragmentación de paquetes permite hacer control de congestión. El tamaño de los paquetes es importante en una LAN inalámbrica ya que la

probabilidad de que un paquete se deteriore aumenta con su tamaño, si se deteriora su retransmisión cuesta menos y en un sistema con salto de frecuencia, el medio se interrumpe por el salto por lo que los paquetes pequeños tienen menos posibilidad de que la retransmisión se posponga. Esta fragmentación reduce la necesidad de retransmisión en muchos casos y así mejora el funcionamiento radio de la red completa.

La capa MAC es la responsable de reconstruir los fragmentos recibidos, siendo el proceso transparente a los protocolos de nivel superior.

1.3.3 Tipos de WLAN

Una red de computadores de forma inalámbrica se puede denominar como Wireless Local Area Network o WLAN, y puede ser clasificada en tres modos fundamentales de acuerdo a su forma de conectar sus dispositivos y son:

- Ad-Hoc (IBSS o Conjunto de servicios básicos independientes)
- Modo Infraestructura (BSS o conjunto de servicios básicos)
- Por su grado de autenticación.

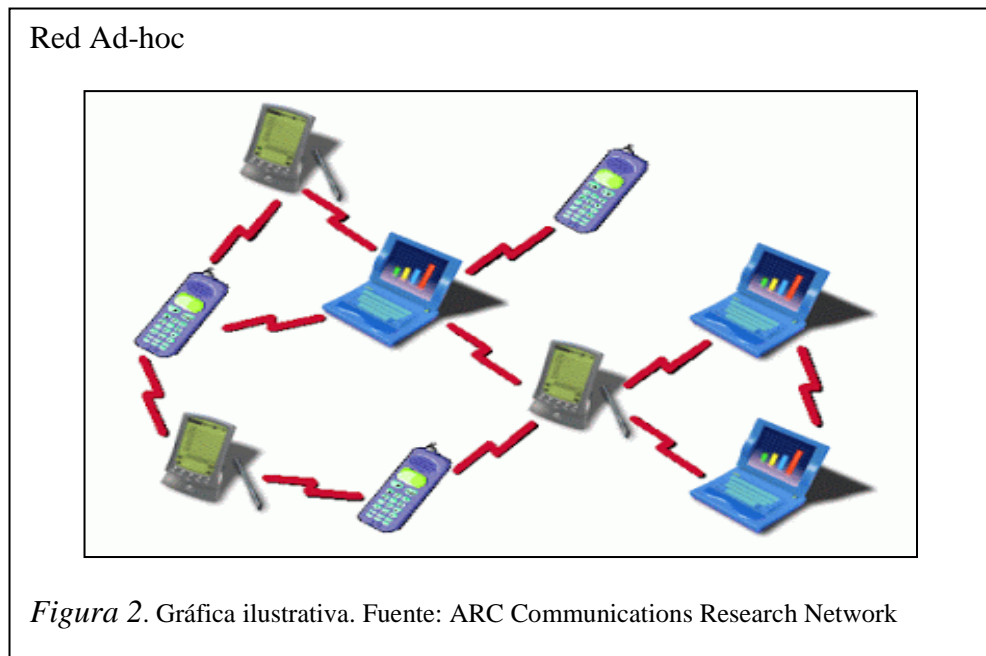
1.3.3.1 Modo Ad-hoc (IBSS) o Punto a Punto

Esta forma de conexión de redes consiste en un grupo de computadores que se comunican cada uno directamente unos con otros por medio de las señales de radio sin usar un punto de acceso.

Las configuraciones Ad-hoc son comunicaciones de tipo punto a punto. Las estaciones trabajan dentro de un rango de transmisión definido por lo que la cobertura es limitada a pocos usuarios. Normalmente este tipo de configuración es usada en campos como el ejército, celulares y juegos de vídeo (Conalep Tlalpan, 2011).

Con el avance de los medios y equipos tecnológicos también ha impulsado el avance de redes más eficientes como las MANET (Mobile Ad-hoc Networks). Las MANET actualmente son redes de usuarios de equipos móviles como celulares, consolas de juegos y tablets con el fin de compartir archivos y así proporcionar flexibilidad y autonomía como se observa en la figura 2, aprovechando los principios de auto-organización (Cano, 2004).

Los dispositivos Ad-hoc pueden también retransmitir tráfico entre dispositivos (computadoras, Móviles, Tablets) que estén dentro de su alcance. Dispone de un identificador único para cada uno de esas conversaciones con una dirección MAC de 48 bits.



1.3.3.2 Modo Infraestructura (BSS) o Punto Multipunto

Infraestructura (BSS Basic Service Set) o conjunto de servicios básicos. Contrario al modo Ad-hoc donde no hay un elemento central, en este modo hay un elemento de coordinación denominado un punto de acceso (AP) o estación base.

Si el AP se conecta a una red Ethernet cableado los clientes inalámbricos pueden acceder a la red fija a través del punto de acceso. Para interconectar muchos puntos de acceso y clientes inalámbricos, todos deben configurarse con el mismo SSID (Service Set Identifier).

En redes IEEE 802.11 el modo de infraestructura es conocido como conjunto de servicios básicos (BSS Basic Service Set) o maestro y cliente, tal como se aprecia en la figura 3.

Estructura BSS

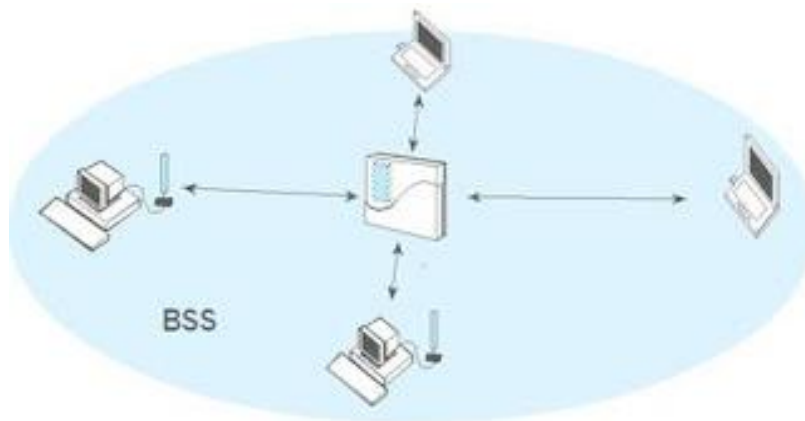


Figura 3. Gráfica ilustrativa BSS. Fuente: Dolpho Houses Network Architecture.

1.3.3.2.1 Tipos de topologías en modo infraestructura:

Caso 1. Estrella: es la infraestructura más común en redes inalámbricas, usada para un Hostpot (punto de conexión a Internet) como se muestra en la figura 4.

Topología estrella

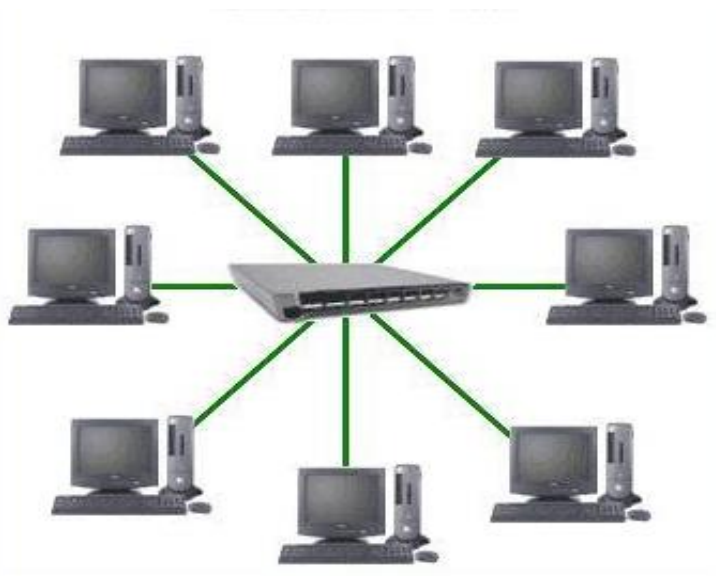
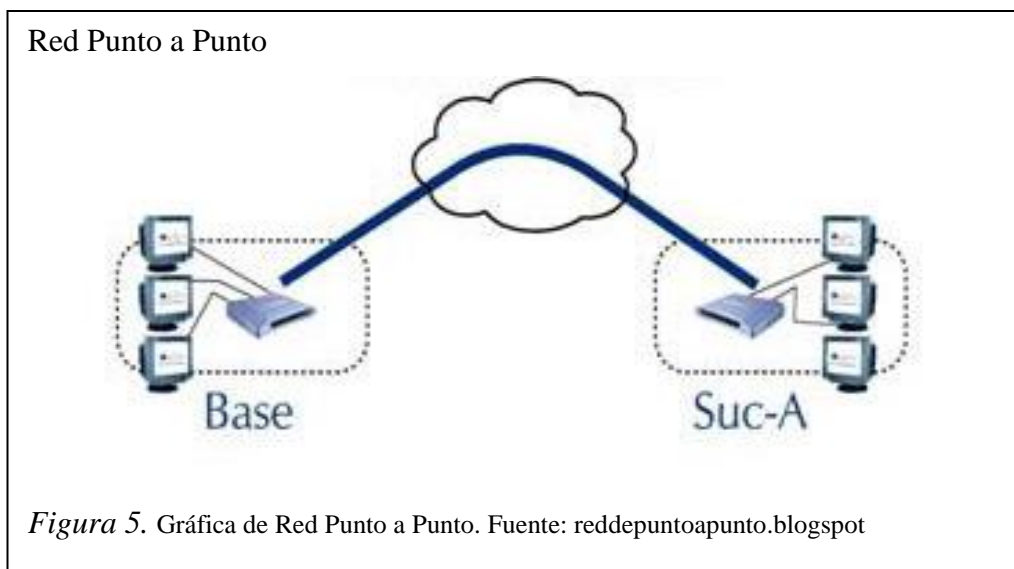
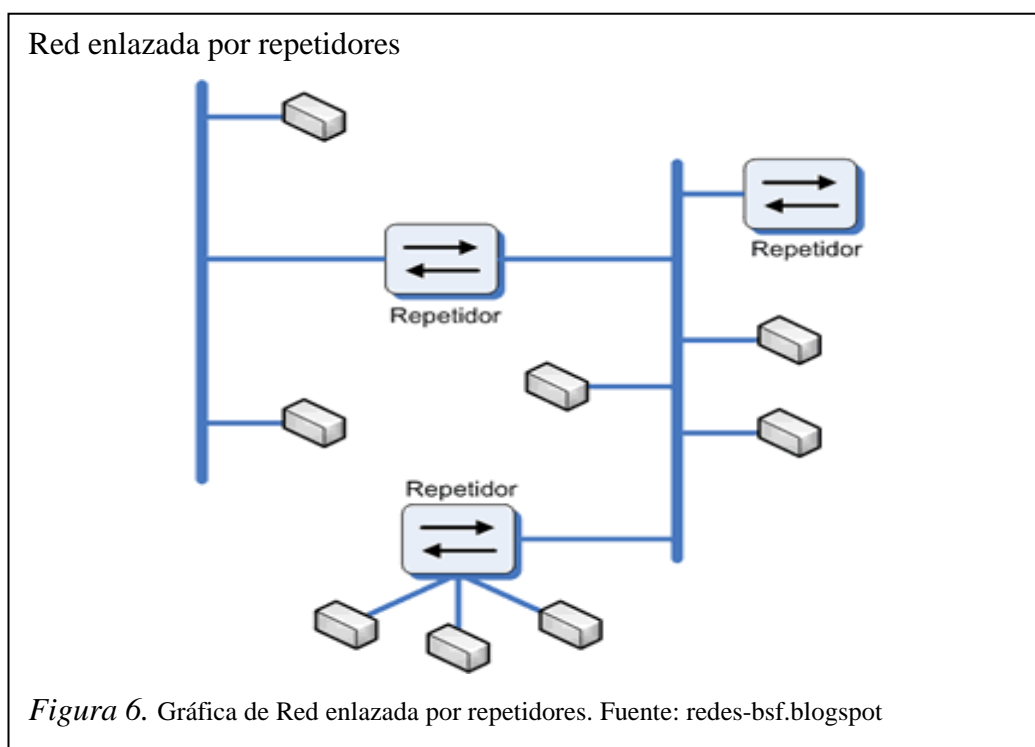


Figura 4. Gráfica de Topología estrella. Fuente: Blogspot tkmpm2010.

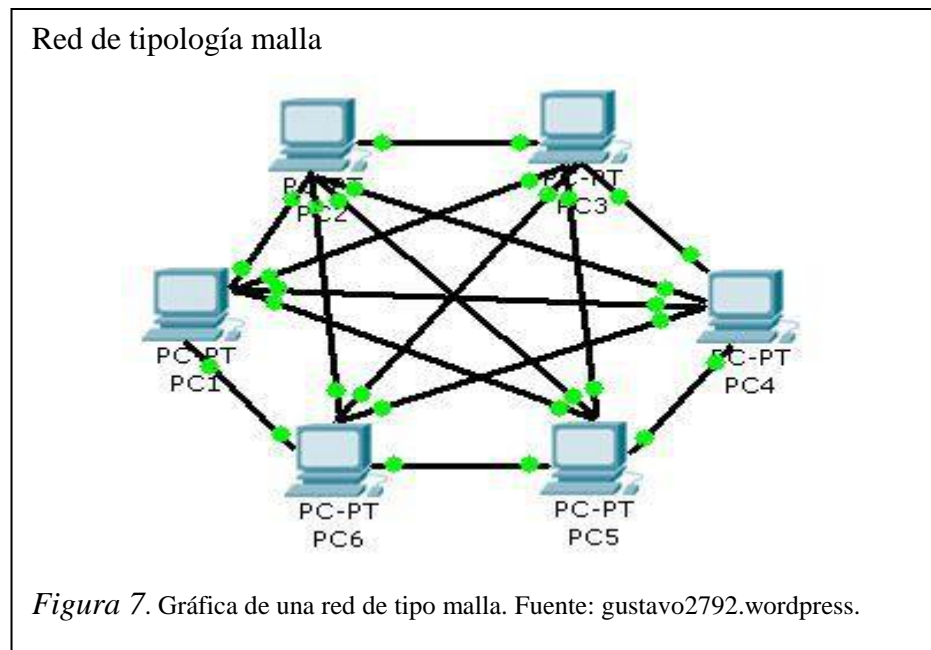
Caso 2. Punto a Punto: los alcances punto a punto permiten que la interconexión sea segura y equilibrada. Esta infraestructura inalámbrica permite que el AP y el cliente se conecten directamente sin intermediarios (Manet's, 2004).



Caso 3. Repetidoras: estas redes permiten restablecer la señal que se emite del AP, que quizás por la distancia o interferencias se ha disminuido o por obstrucciones en la línea de vista directa o hay distancia muy larga para cada enlace.



Caso 4. Malla: es utilizada en ámbitos urbanos, esta topología se encuentra en redes nacionales, campus universitarios y vecindario un dispositivo se encarga de centralizar las comunicaciones (Manet's, 2004).



1.3.3.3 Redes por el grado de autenticación

Las redes de tipo estructura se pueden definir como redes por autenticación. La autenticación se requiere por el uso y el tipo de clientes que deben conectarse a la red. Muchas redes requieren privacidad por el tipo y la cantidad de información que manejan. Por citar el caso, se tiene una red de gobierno, por el tipo de información que contiene esta red no se puede permitir que cualquier usuario use la red.

Tomando en cuenta el otro caso, la demanda de velocidad y ancho de banda de los usuarios puede hacer lento el tráfico de datos. Por ejemplo en una red de internet con una velocidad de 1 Mbps simétrico y sin seguridad, si muchos usuarios se conectan libremente, la velocidad de la red y el ancho de banda empleado se reparte para todos los usuarios de forma asimétrica causando lentitud y problemas en la recepción de información.

Siendo estas las razones más relevantes se presentan dos tipos de redes por el grado de autenticación, las cuales son públicas y privadas.

1.3.3.3.1 Red de acceso público

Estas redes como su nombre lo dice, su acceso es público, por lo que no se requiere usar seguridades. En este ejemplo no se requiere autenticación, es decir el acceso es libre a la red, pero puede tener problemas de dimensionamiento, de esta forma la red se ve limitada por la cantidad de IPs de las que se disponga. Por lo tanto si existe limitación de IPs la cantidad de usuarios también es limitada.

1.3.3.3.2 Red privada

La red privada, como su nombre lo indica solo es usada por determinados clientes por lo tanto debe tener seguridades internas. Usualmente las seguridades son basadas en WEP, WPA o WPA2, sin embargo existen otros medios de validación de usuarios, como portal cautivo, etc. En resumen una red de acceso privado es selectiva con sus usuarios y no se puede ingresar a ella libremente.

1.4 Características de las redes inalámbricas

1.4.1 Modulación

Consiste en variar determinado aspecto de una señal, denominada portadora con respecto a una segunda señal, denominada señal moduladora, generando finalmente una señal u onda modulada.

En el proceso de modulación, la señal de alta frecuencia (portadora) quedará modificada en alguno de sus parámetros como su amplitud, frecuencia, fase, etc. de manera proporcional a la amplitud de la señal de baja frecuencia o moduladora (ecuared, 2014).

1.4.1.1 Tipos de Modulación

- Cuando la señal portadora y la moduladora son señales analógicas.
 - Modulación de amplitud, AM
 - Modulación de frecuencia, FM
 - Modulación de fase, PM

- Cuando la señal portadora es analógica y la señal moduladora es digital.
 - Desplazamiento de amplitud, ASK
 - Desplazamiento de frecuencia, FSK
 - Desplazamiento de fase, PSK
 - Modulación de modulación en cuadratura, QAM

- Cuando la señal portadora es digital y la señal moduladora es analógica.
 - Modulación por amplitud de pulsos (PAM).
 - Modulación de pulsos en duración (PDM).
 - Modulación de pulsos en posición (PPM).
 - Modulación por codificación de pulsos PCM.
 - Modulación por anchura de pulso (PWM).
 - Modulación Delta.

En las transmisiones wireless se usan las modulaciones digitales de portadora analógica y moduladora digital específicamente la PSK, o desplazamiento de fase. Las variaciones de este tipo de modulación son BPSK (Binary Phase Shift Keying) la que usa 2 símbolos de desplazamiento, QPSK (Quadrature Phase Shift Keying) la que usa 4 símbolos en el desplazamiento y QAM (Quadrature Amplitude Modulation) la que usa 8 símbolos en el desplazamiento.

El estándar 802.11n usa para su desempeño el esquema OFDM, que consiste en modular en QAM o PSK, para enviar un conjunto de portadoras de varias frecuencias (Serrano).

El símbolo OFDM pasa por un codificador de canal con el objetivo de corregir los errores producidos en la transmisión, luego de este paso se lo denomina COFDM, del inglés Coded OFDM. Para evitar la interferencia de las miles de portadoras espaciadas equitativamente que forman una modulación OFDM, el proceso de modulación y demodulación se realiza en tiempo discreto mediante la IDFT (Discretización por la Transformada Inversa de Fourier) y la DFT (Discretización por la Transformada de Fourier) respectivamente.

La característica principal de la modulación OFDM es su fortaleza a la multitrayectoria, y al desvanecimiento debido a las condiciones ambientales meteorológicas y las interferencias de radio frecuencia. Además, en esta modulación como las señales tienen distintos retardos y amplitudes que llegan al receptor, estas contribuyen de forma positiva en el resultado evitando que existan problemas de interferencia (Serrano).

1.4.2 Distancia

La distancia se define como la máxima longitud de un enlace y depende de dos datos principales, las antenas y la potencia del transmisor.

En el caso de dispositivos indoor, la potencia es de 200 mW o hasta 250 mW con antenas de 2 dbi o 5dbi, lo que permite que se obtenga una distancia máxima de 30m a 50m con antenas direccionales.

En el caso de dispositivos outdoor, los APs consisten en la unión de un transmisor y un transceiver o router 802.11. Estos dispositivos permiten que se tenga potencias de hasta 1 W o 1,5 W con antena de 12 dbi a 35 dbi, aparte del procesamiento del router que emite los datos a las velocidades designadas a los estándares 802.11. Con estos parámetros se alcanzan distancias punto a punto de hasta 50 Km o más. (inforactiva.net, 2014).

1.4.3 Cobertura

La cobertura de transmisión depende del tipo de antena que se use, ya que existen dos tipos de antenas relacionadas con la cobertura.

La primera es la omnidireccional, la que permite tener una cobertura teórica de 360°, aunque realmente como se observa en la figura 8 depende de la polarización de la antena para lograr lo esperado (Cachorro wireless, 2013).

Cobertura antenna omnidireccional

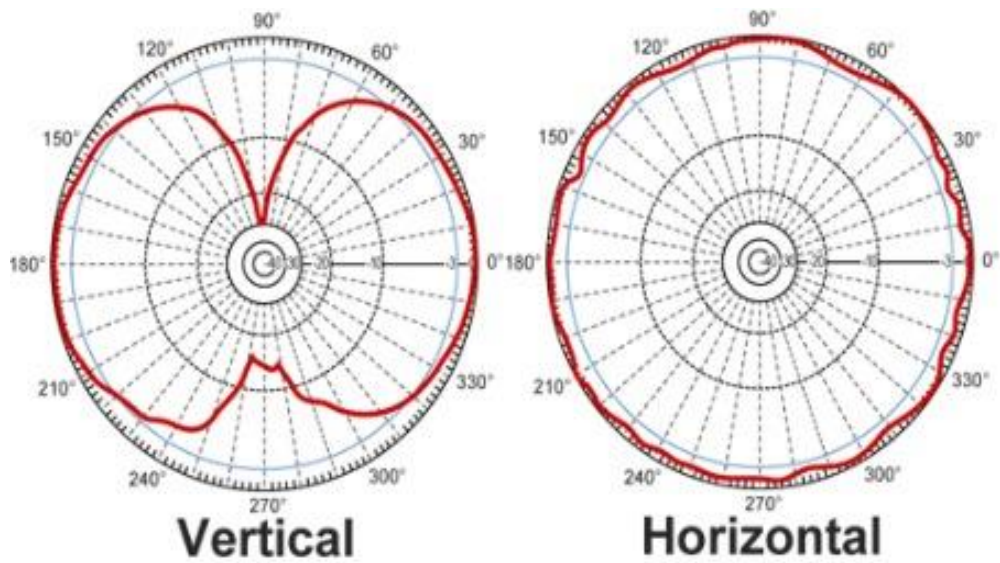


Figura 8. Gráfica de Cobertura omnidireccional. Fuente: cacharreopuroyduro.blogspot.

El segundo tipo, son las antenas direccionales las que enfocan su cobertura en zonas específicas con un ángulo de cobertura. Existen de 30°, 60°, 90°, 120°, tal como se muestra en la figura 9.

Cobertura antenna direccional o sectorial

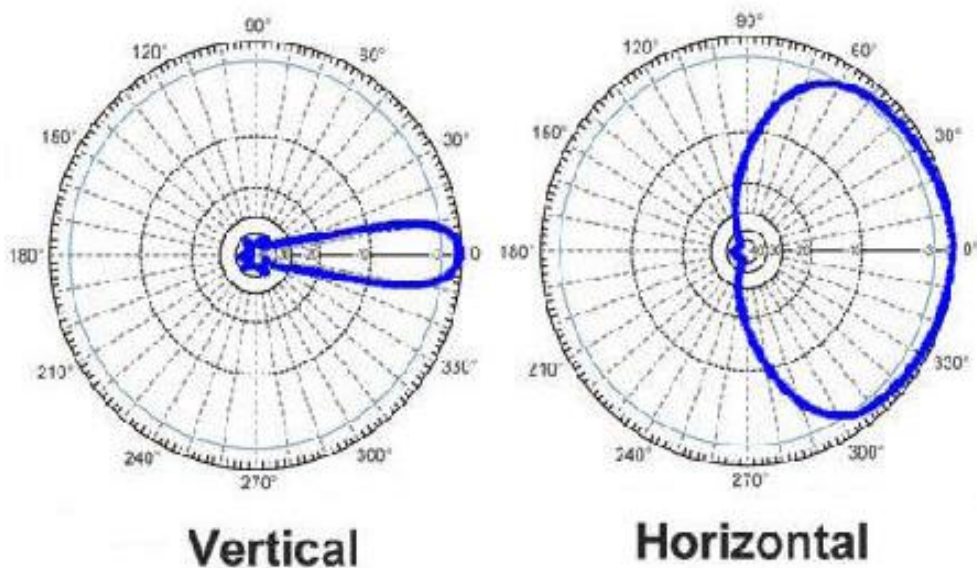


Figura 9. gonzalonazareno.org

1.5 Estructura de los enlaces

1.5.1 Transmisores

Los transmisores tanto en 802.11n como en 802.11ac, pueden ser usados como Access Point (AP) o como transmisores simples dependiendo de la red a la que pertenezcan. Para elegir un transmisor se debe tomar en cuenta lo siguiente:

- La capacidad del AP para soportar el tráfico, esta depende de la infraestructura de construcción del AP y la asignación del canal.
- La cobertura para determinar el área de tráfico, este parámetro depende de una ubicación adecuada del AP para cubrir un área específica.
- Las estaciones deben estar bajo el mismo estándar, ya que se puede tener redes con escalabilidad y soporte para 802.11ac, pero si las redes están conformadas por estaciones 802.11n, no se aprovechará el potencial de 802.11ac ya que con diferentes estándares, siempre la conexión se dará bajo el estándar de menor categoría. Por ejemplo, tal como se muestra en la figura 10 y figura 11, se puede observar el funcionamiento con los estándares 802.11g y 802.11n (TP-LINK, 2014).

Relación de velocidades de un estándar combinado



Figura 10. tp-link.com

Relación de velocidades de dos estándares combinados

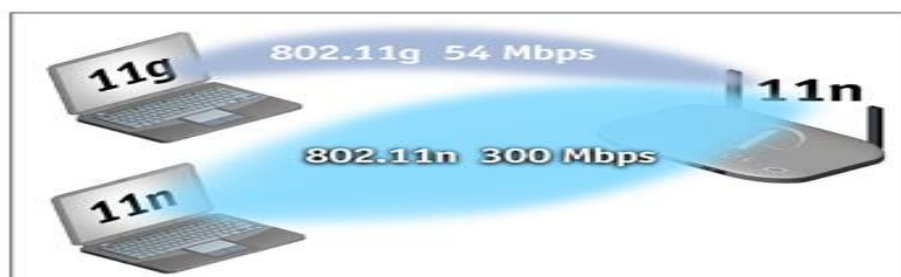


Figura 11. tp-link.com

1.5.2 Receptores

Los receptores son dispositivos importantísimos en el desempeño de la transmisión 802.11, ya sea 802.11n y 802.11ac.

Como se observa en la figura 10 citada anteriormente, en una red 802.11ac si se usan receptores 802.11n, todo el potencial de la red 802.11ac se ve limitado al desempeño máximo que puede aceptar el receptor 802.11n, que sería hasta 300Mbps realmente. Esto desde el punto de vista económico sería un desperdicio y pérdidas de dinero. Ahora si se mira desde el punto de vista más técnico en una migración se deben tomar en cuenta varios aspectos de las redes, como el uso al que son destinados los clientes o receptores. Por ejemplo, si las transmisiones son de video o teleconferencias, las transmisiones requieren que su ancho de banda y velocidad sean más altas. Entonces se admitiría que los receptores trabajen bajo la norma 802.11n o estándares anteriores pero con velocidades más reducidas que la red 802.11ac.

1.5.3 Seguridad

1.5.3.1 Proceso de Asociación y seguridad

En las redes WLAN, por su estructura se requiere tener seguridades basadas en algoritmos ya diseñados como los que se citan a continuación. (Gutierrez, 2012)

Antes de pasar al análisis de los métodos de seguridad que se utilizan, se debe resaltar que este proceso se efectúa en la capa de enlace especialmente en la MAC o Capa de Acceso al Medio. El proceso se inicia como se describe a continuación:

- **Sincronización**

En el caso de las redes con infraestructura, las estaciones pueden sincronizarse escaneando el medio de dos maneras diferentes:

- **Escaneo pasivo del medio:** la estación escucha el medio y recibe unas tramas llamadas balizas o Beacon desde el AP. Los Beacon se envían periódicamente e incorporan una marca de tiempo (para la sincronización), el nombre de la red (SSID), si se usa cifrado, las velocidades soportadas, etc.

- **Escaneo activo del medio:** la estación conoce el nombre de la red y es la propia estación la que busca la red determinada transmitiendo una trama llamada Probe Request. El AP contestará con el Probe Response dándole la misma información que da el Beacon. Este sistema se utiliza para evitar que el AP tenga que transmitir continuamente el Beacon y pueda ser ocultado.

La marca de tiempo de los Beacon es básica ya que será la que usarán todas las estaciones cliente de ese AP para sincronizarse.

- **Autenticación**

Este es el siguiente paso, una vez sincronizadas ambas estaciones, se realiza la autenticación. La autenticación 802.11 se aplica en la capa MAC del estándar y su proceso se realiza en una sola dirección (sólo el cliente debe autenticarse). El cliente debe demostrar que conoce la contraseña de autenticación, como puede ser mediante el cifrado con esa contraseña de una serie de bits conocidos por ambos extremos de la comunicación.

- **Asociación**

Finalmente, una vez autenticada la estación cliente, se realiza el proceso de Asociación. Este proceso sirve al AP para tener un registro de clientes que tiene conectados, así, una vez autenticados, el sistema de distribución (DS), sabrá dónde encontrar un cliente que esté conectado a la red inalámbrica. En este proceso se intercambia la IP actualizando la tabla ARP (Protocolo de resolución de direcciones) del AP y del cliente.

- **Seguridad**

El estándar 802.11 incluyó un método de seguridad con el que pretendía obtener el mismo nivel de privacidad que aporta una red cableada. Este método de seguridad tiene como fin permitir sólo el acceso de usuarios legítimos a la red (control de acceso). Y también evitar que los usuarios ilegítimos sean capaces de leer los datos transmitidos por la red inalámbrica.

1.5.3.2 Tipos de seguridades

Wep (Wireless Equivalent Privacy)

Este método cifra las transmisiones mediante una clave compartida (que los usuarios y el punto de acceso han de conocer) usando el algoritmo RC4. Además, como se puede observar, el uso de estas claves compartidas también puede servir para realizar la autenticación de usuario por método Shared-Key Authentication (Gutierrez, 2012).

Este sistema propuesto por el estándar inicial quedo totalmente obsoleto y desfasado a causa de la aparición de métodos de rotura del algoritmo RC4 y del incremento de la potencia de los procesadores. El IEEE propuso una revisión de seguridad y, en 2004, publicó el 802.11i. Hay que destacar que algunas de las mejoras propuestas por el 802.11i son incompatibles con el hardware anterior, como puede ser, cifrado mediante AES (Advanced Encryption Standard) que necesita de tarjetas inalámbricas con hardware especializado. Esto propicia que aún se use el sistema WEP por motivos de compatibilidad.

WPA (Wi-Fi Protected Access)

Surge con el fin de corregir las limitaciones del WEP. Introduce mejoras de seguridad como el TKIP (Temporal Key Integrity Protocol), que varía por sí solo la contraseña Wi-Fi cada cierto tiempo. Su variante más normal es la WPA-Personal. Usa el sistema PSK, o de clave pre compartida. En él, todos los usuarios de la red inalámbrica tienen una misma contraseña Wi-Fi, que el propio usuario define. También hay una versión WPA empresarial (WPA-Enterprise). Ofrece seguridad adicional al obligar al usuario a identificarse con un nombre y contraseña en sistemas de autenticación especiales, como RADIUS o 802.1X (Gutierrez, 2012)

WPA2 (Wi-Fi Protected Access 2)

Es el estándar más moderno para proteger redes inalámbricas y el que recomienda la Wi-Fi Alliance. Existe también una versión personal, WPA2-Personal y empresarial, WPA2-Enterprise.

WPA2 es compatible con WPA, lo que significa que en tu red Wi-Fi se puede usar PCs o dispositivos, como routers, adaptadores de red, etc. Que admitan uno u otro sistema.

WPA2 no es compatible con sistemas WEP, así que no se puede juntar en una misma red WiFi dispositivos que sólo admitan WEP con otros válidos para WPA2.

TKIP o AES

El cifrado AES (Advanced Encryption Standar) es sin duda el más fuerte y capaz de proteger la red Wi-Fi. Ya que ofrece un mayor nivel de seguridad, pero requiere un hardware específico que no es compatible con los dispositivos que sólo funcionaban con WEP y con WPA. Utiliza bloques de cifrado de 128, 192 o 256 bits y se lo considera el sistema de cifrado estrella. AES necesita más potencia de cálculo y eso repercute en el consumo energético de algunos dispositivos móviles. Pero AES no sólo es más seguro, sino también es más eficiente ya que utiliza menos ancho de banda. (Angel, 2014).

En el caso del TKIP se han descubierto ciertas vulnerabilidades que hacen que ya no ofrezca una buena seguridad (Gutierrez, 2012).

1.6 Parámetros de transmisión.

1.6.1 Bit error rate (BER)

En español, Tasa de Error Binario, este error sucede cuando en el transcurso de la transmisión por situaciones fuera de lo previsto en lugar de llegar un determinado bit llega otro, por ejemplo en lugar de ser 1 llega un 0. Siendo esta situación el BER o Bit error rate, es la relación entre la cantidad de bits errados contra la cantidad de bits transmitidos.

$$BER = \frac{\text{cantidad de bits errados}}{\text{cantidad de bits transmitidos}} = \frac{N_{Err}}{N_{bits}}$$

De esta comparación, el BER es la tasa de error de bit medida, donde:

N_{Err} es el número de bits errados y N_{bits} es el número total de bits comparados. Esto requiere de un equipo de medición muy sofisticado que permita la comparación de la amplitud y tiempo de transición de cada bit y ofrezca una capacidad de ajuste del retardo de la propagación de la señal. (Damm, 2010).

Por lo tanto se determina que el valor del BER mientras más cercano sea a 1, más eficiente será el sistema de transmisión, permitiendo que se evalúe su capacidad de eficiencia en el trabajo real.

1.6.2 Retardo

El retardo o latencia se define técnicamente como la diferencia que existe entre el momento en que una señal es transmitida y el momento que una señal llega a su destino.

Existen cuatro tipos de retardos cuando se transmite por una red de comunicaciones, éstos son los siguientes:

- **Retardo de transmisión:** es el tiempo que invierte el emisor en poner la información en la línea de transmisión. Depende del equipo encargado de poner la información en la línea.
- **Retardo de propagación:** es el tiempo que tarda la información en viajar por la línea de transmisión desde el emisor hasta el receptor. Depende de la distancia y del medio.
- **Retardo de conmutación:** es el tiempo que un nodo necesita para decidir hacia qué nodo debe reenviar la información recibida. Depende de la velocidad de proceso informático del nodo y del tráfico de la red.
- **Tiempo de aceptación:** es el tiempo que transcurre desde que llega la información hasta que el receptor reconoce la llamada. (Hernando, 2009)

El retardo en general es un problema de las redes de telecomunicaciones que disminuye la calidad de servicio principalmente en comunicaciones en tiempo real como voz y video, que son las más afectadas. Como en este estudio se realiza con transmisión de datos solo se dará importancia al retardo más importante que es el Jitter.

1.6.3 Jitter

El Jitter es un término usado en el dominio de frecuencia digital y se trata de la desviación no deseada de la verdadera periodicidad de una señal periódica, a menudo en relación con una fuente de reloj de referencia.

La fluctuación de fase puede observarse en las características tales como la frecuencia de los impulsos sucesivos, la amplitud de la señal, o fase de las señales periódicas. El Jitter es un factor significativo, y por lo general no deseado, en el diseño de casi todos los enlaces de comunicaciones. En las aplicaciones de recuperación de reloj se llama saltos temporales (E-CENTRO, 2014).

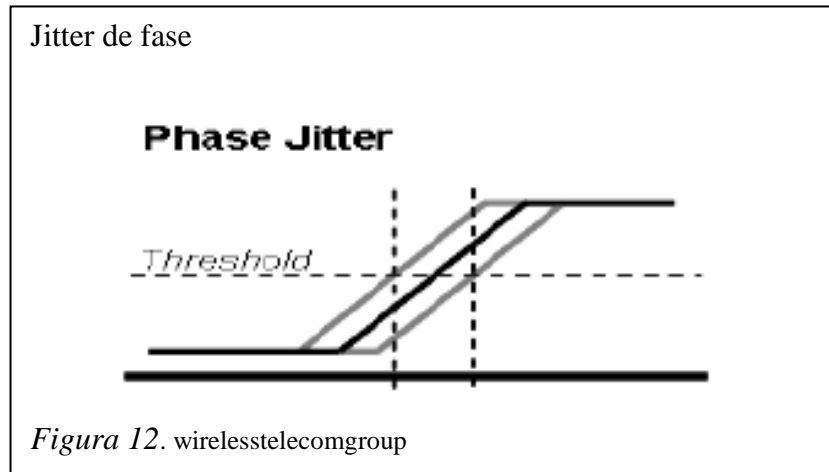
En una transmisión ideal la información binaria es transmitida como bits en un flujo de datos de unos y ceros aleatorios. Estos bits estarán estrictamente a disposición en un cierto momento y estarán presentes exactamente para un periodo predeterminado. Pero hablando del caso real está muy lejos de esperarlo idealmente, pues se debe trabajar contra una variedad de factores influyentes y la calidad de la señal de datos transmitidos que son una fuente de Jitter.

Comúnmente el Jitter es reconocido como una cantidad de alta frecuencia. El comportamiento del Jitter a frecuencias por debajo de 10 Hz es llamado “wander” y “drift” aún a frecuencia más bajas. (Damm, 2010)

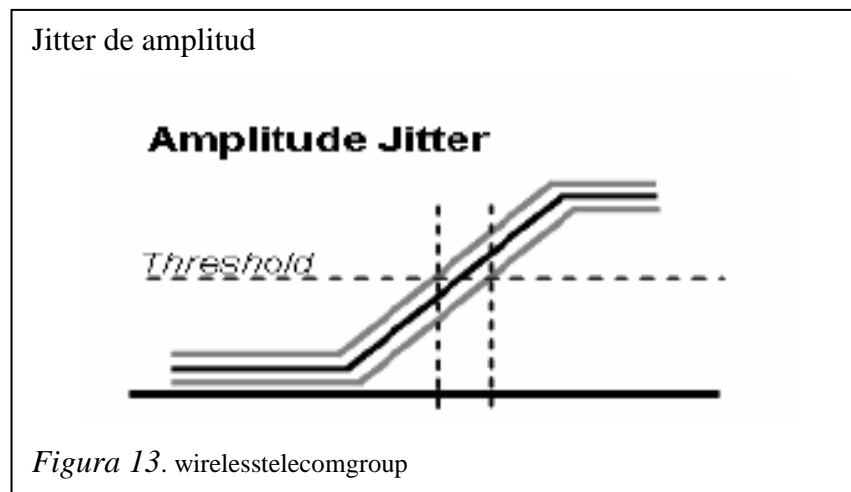
La frecuencia de reloj necesaria en el receptor serial se extrae de la trama de ingreso a la entrada de recepción del flujo de datos, donde las transiciones de bit son utilizadas para sincronizar el oscilador controlado por voltaje (VCO) por medio del lazo de enganche de fase de la unidad de recuperación de información del reloj (CDR).

1.6.3.1 Jitter de Fase y Amplitud

Las variaciones de frecuencia o desfase en el tiempo de transición ideal son llamadas Jitter de fase, tal como se muestra en la figura 12 (Damm, 2010).



En cambio las variaciones de nivel de señal también ocurren en sistemas digitales; son llamadas Jitter de amplitud, como se puede apreciar en la figura 13.



Debido a los tiempos finitos de transición de la señal, las variaciones del nivel de estado tienen un impacto en la determinación del estado binario de los sistemas. Dependiendo de su nivel de origen justo antes de la transición, la pendiente de la señal alcanza el umbral determinando tarde o temprano el real estado del bit. Esto nuevamente tiene un efecto en el periodo de decisión de “1” o “0” disponible en el sistema.

Dado que siempre es dependiente del estándar tecnológico, no existe una definición universal del Jitter. La Unión Internacional de Telecomunicaciones (ITU-I) define el Jitter como la variación en cortos periodos de los instantes significantes de una señal digital de su posición ideal en el tiempo (Damm, 2010).

El Jitter puede ser limitado o no limitado. El primero está relacionado con la frecuencia y magnitud de los eventos del sistema; por lo tanto, el Jitter limitado es determinístico. Esto significa que al deshabilitarse la fuente se detendrá el Jitter limitado también. El Jitter limitado siempre tiene una magnitud limitada. A continuación se menciona algunos tipos de Jitter limitado.

- **El Jitter Determinístico (JD)**

Este tipo de Jitter es una función de densidad de probabilidad no Gaussiana, está siempre limitado en amplitud y con causas específicas. Las fuentes son imperfecciones de los dispositivos, crosstalk y problemas de aterramiento (Wolfgang, 2014).

- **El Jitter por interferencia de símbolos (JIS)**

La Interferencia Entre Símbolos (IES) es un ejemplo, de otro tipo de Jitter. La Interferencia Entre Símbolos es la forma más común de JDI (Jitter dependiente de la información). Es causada generalmente por limitaciones del ancho de banda en las líneas de transmisión.

Este tipo de Jitter afecta a bits individuales rodeados por bits del estado opuesto, las transiciones de señal originan interferencia en los canales vecinos, pero si las líneas de datos del origen están inactivas no ocurrirá interferencia alguna. Debido a la energía finita de los eventos del origen, el Jitter resultante es también finito y siempre es atenuado (E-CENTRO, 2014).

- **El Jitter dependiente de la información (JDI)**

También es parte de la Interferencia Entre Símbolos (IES), Distorsión del Ciclo de Trabajo (DCT), y el Jitter Eco (JEC). Los errores de tiempo varían con el patrón de datos. La fuente primaria son los componentes y las limitaciones del ancho de banda. Las señales de alta frecuencia tienen menor tiempo para establecerse que las señales de más bajas frecuencias. Esto lleva a cambios en las condiciones iniciales para transiciones a diferentes frecuencias y produce errores de temporización dependientes del patrón de datos aplicado.

La distorsión de tiempo de trabajo o DCT es un tipo de Jitter causado cuando ciertos estados de bit tienen diferente duración. “1” siempre es más largo que “0” o viceversa y es causado por la configuración del BIOS e insuficiente alimentación continua de un componente.

- **El Jitter Eco (JEC)**

Es causado por incompatibilidad entre el componente transmisor y la línea de transmisión, depende del patrón o trama de datos. La longitud de la línea de transmisión influye también en la magnitud del JEC.

En las trasmisiones se puede dar también otro caso de Jitter, el llamado Jitter periódico o también conocido como Jitter Senoidal debido a su forma. La fuente es generalmente señales en forma de interferencia relacionadas con el patrón de datos, variaciones de aterramiento o variaciones en la fuente de alimentación. (Damm, 2010).

- **Jitter Aleatorio (JA)**

El Jitter ilimitado no depende de los eventos. Puede ser causado por los componentes del sistema o influencias externas. El principal es el Jitter aleatorio (JA) el cual es causado por el ruido blanco común en todos los componentes activos y pasivos. Los amplificadores y conductores de línea multiplican la energía del ruido. Por su

naturaleza, la distribución de energía del ruido blanco es Gaussiana, por lo tanto el (JA) puede describirse por su función de densidad de probabilidad que es:

$$FDP_{JA}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

Donde x es la variable independiente y σ (sigma) el valor cuadrático medio y μ (miu) la media de distribución. El Jitter se define como una desviación del estado ideal, lo que significa que tiene variaciones positivas y negativas relativas a su estado ideal. Por ello, en el análisis de Jitter, μ siempre es 0 y puede omitirse.

La ecuación de FDP (Factor de probabilidad) indica que independientemente de cuan largo pueda ser x, la probabilidad de que un Jitter causante de eventos puede ser muy pequeña, pero nunca alcanzará el eje x. Debido a su amplio espectro de frecuencias el ruido blanco es muy difícil de suprimir o atenuar sin impacto sobre la señal real (Damm, 2010).

Como observación el Jitter siempre estará presente en cualquier sistema de comunicación digital, por lo que lo ideal sería hacer lo posible para eliminar la mayor cantidad de fuentes de Jitter y especialmente el Jitter limitado, ya que este tiene que ver con la sincronización de los dispositivos, aterramientos, deficiencias de los dispositivos transmisor y receptor, incompatibilidad de estándares o trabajo de distintos estándares como en el caso de usar 802.11n y 802.11ac en una misma red. El efecto es la demora y pérdida de información haciendo más lenta la red que se haya implementado.

CAPÍTULO 2

ESTÁNDARES IEEE 802.11ac y IEEE 802.11n

2.1 Descripción del estándar 802.11n

La IEEE desde el año 2004 inicia el estudio del estándar 802.11n, ya que la demanda de mayor cantidad de información desplegada por Internet y la capacidad de mejoramiento del audio y video en varios sistemas de la época eran cada vez más notables. Por esta razón el estándar 802.11n se desarrolla usando algunas características de otras versiones de WIFI, como 802.11a, 802.11b, 802.11g, las cuales trabajan sobre aspectos como ancho de banda y velocidad, pero todavía bajo el tratamiento de una sola señal delegada a una sola antena.

Tras el aparecimiento de estándar 802.11n también aparece una técnica muy notable de transmisión, denominada MIMO (Multiple Input, Multiple Output). Esta tecnología se basa en la utilización de varias antenas para transportar múltiples ráfagas de datos usando el rebote de un lugar a otro, lo que permite la transmisión de mayor cantidad de datos en el mismo período de tiempo, es decir hay un aumento de velocidad. MIMO también constituye la base para aumentar el rango de cobertura y distancia.

Otra tecnología incorporada en 802.11n y directamente ligada también al aumento del rendimiento es el channel bonding que es la unión o emparejamiento de canales, desarrollado por la técnica de multiplexación denominada OFDM (Orthogonal Frequency Division Multiplexing). Este sistema permite utilizar simultáneamente dos canales no superpuestos como si se tratara de uno solo, pero con el doble de capacidad. Estos canales deben ser adyacentes o contiguos. Al usar esta técnica se puede sumar el ancho de banda de dos canales de 20 MHz para conseguir emitir datos sobre un canal de 40 MHz.

Un tercer aspecto importante sobre el que se destaca 802.11n de sus antecesoras es la implementación de una tecnología denominada agregación de paquete o (payload optimization), que en términos sencillos, permite meter más datos en cada paquete transmitido con el fin de optimizar las tramas emitidas (chemutal wireless, 2014).

2.1.1 Modulación y Velocidad del estándar 802.11n

Uno de los objetivos primordiales por los que existe 802.11n es el aumento de velocidad y para lograrlo fue indispensable mejorar las técnicas de modulación, por tanto se definen nuevas técnicas o mejoras a las ya existentes. A las técnicas de modulación se las denomina Esquemas de Modulación y Codificación, con sus siglas en inglés MCS (Modulation and Codification Scheme).

Estos esquemas muy necesarios porque en las redes WLAN o de comunicación inalámbrica los dispositivos tienen el Data Rate que significa Velocidad de transmisión o Tasa de Transferencia lo que genera una amplia forma de envío de datos, así que esto hace que aparezcan varios esquemas de modulación y codificación especialmente para el estándar 802.11n.

El MCS es un valor que determina la modulación, la codificación y el número de canales espaciales, dado por la permutación de los distintos factores involucrados. Al identificar los MCS soportados por los dispositivos, se puede determinar la velocidad de datos de la capa física real, que van desde un mínimo de 6,5 Mbps a un máximo de 600 Mbps, esto tomando en cuenta las velocidades teóricas.

Existen 77 MCS para la norma IEEE 802.11n, de los que los 16 primeros, son obligatorios para los dispositivos Access Points o puntos de acceso, encargados de transmitir.

Todos los puntos de acceso 802.11n deben soportar como mínimo desde MCS0 hasta MCS15. En cambio para los clientes de 802.11n es obligatorio usar los MCS desde MCS0 hasta MCS7 y los MCS8 al MCS15 son opcionales.

Cada MCS es una combinación de una modulación determinada, por ejemplo, BPSK, QPSK, 64-QAM.

La tasa de codificación o Coding Rate es por ejemplo, $\frac{1}{2}$, $\frac{3}{4}$, etc. Esta tasa de codificación es la velocidad de información no redundante que llega al receptor o sea la información que se repite en el proceso de recepción, de donde k/n es el valor de

redundancia. Así que k es la información útil recibida y n la cantidad de bits enviados. Entonces $5/6$ se refiere a 5 bits no redundantes de 6 recibidos.

El intervalo de guarda o Guard Interval (800ns o 400ns), que es el intervalo de tiempo que se añade entre tramas al salir del transmisor, con el fin de evitar la superposición de señales y sincronizar las cabeceras de las tramas en el receptor, este intervalo se mide en nano segundos (ns).

El intervalo de guarda extiende la duración del símbolo transmitido y por consecuencia, reduce ligeramente el caudal binario efectivo. Cuanto mayor sea el intervalo de guarda menor será la interferencia causada por los efectos multicamino.

El número de secuencias espaciales o Spatial Streams. En un entorno MIMO (Múltiples Entradas y Múltiples Salidas) las tramas de datos son multiplexados o enviados, por el uso de diferentes espacios dentro del mismo canal espectral a estos espacios se les llama secuencias espaciales.

Actualmente hay 128 tasas de transferencia diferentes, pero el estándar establece sólo un subconjunto de ellas como obligatorias para los productos comerciales los cuales son los siguientes:

Para todo AP 802.11n se debe, como mínimo multiplexar utilizando una o dos cadenas de bits, con canales de 20 MHz sin SGI (Short Guard Interval o intervalo de guarda corto). Como consecuencia, el AP (Punto de acceso) como mínimo debe ofrecer, tasas de transferencia para los clientes de 13, 26, 39, 52, 78, 104, 117 y 130 Mbps.

Para todo cliente 802.11n se debe, como mínimo, multiplexar utilizando OFDM (Orthogonal Frequency Division Multiplexing) en una única cadena de bits, utilizando canales de 20 MHz sin SGI. Esto da como resultado que las tasas de transferencia obligatorias para clientes 802.11n son: 6.5, 13, 19.50, 26, 39, 52, 58.5 y 65 Mbps. (redalyc.org, 2006)

A continuación, se cita la Tabla 1 con valores para cada índice MCS del 0 al 31. (La cueva WIFI, 2014)

Tabla 1

Índice MCS de la norma 802.11n

Índice MCS	Spatial Streams	Tipo de Modulación	Tasa de Codificación	Data Rate (Mbits/s)			
				Canal 20 MHz		Canal 40 MHz	
				800 ns GI	40 ns GI	800 ns GI	40 ns GI
0	1	BPSK	1/2	6,50	7,20	13,50	15,00
1	1	QPSK	1/2	13,00	14,40	27,00	30,00
2	1	QPSK	3/4	19,50	21,70	40,50	45,00
3	1	16-QAM	1/2	26,00	28,90	54,00	60,00
4	1	16-QAM	3/4	39,00	43,30	81,00	90,00
5	1	64-QAM	2/3	52,00	57,80	108,00	120,00
6	1	64-QAM	3/4	58,00	65,00	121,50	135,00
7	1	64-QAM	5/6	65,00	72,20	135,00	150,00
8	2	BPSK	1/2	13,00	14,40	27,00	30,00
9	2	QPSK	1/2	26,00	28,90	54,00	60,00
10	2	QPSK	3/4	39,00	43,30	81,00	90,00
11	2	16-QAM	1/2	52,00	57,80	108,00	120,00
12	2	16-QAM	3/4	78,00	86,70	162,00	180,00
13	2	64-QAM	2/3	104,00	115,60	216,00	240,00
14	2	64-QAM	3/4	117,00	130,00	243,00	270,00
15	2	64-QAM	5/6	130,00	144,40	270,00	300,00
16	3	BPSK	1/2	19,50	21,70	40,50	45,00
17	3	QPSK	1/2	39,00	43,30	81,00	90,00
18	3	QPSK	3/4	58,50	65,00	121,50	135,00
19	3	16-QAM	1/2	78,00	86,70	162,00	180,00
20	3	16-QAM	3/4	117,50	130,70	243,00	270,00
21	3	64-QAM	2/3	156,00	173,30	324,00	360,00
22	3	64-QAM	3/4	175,00	195,00	364,00	405,00
23	3	64-QAM	5/6	195,00	216,70	405,00	450,00
24	4	BPSK	1/2	26,00	28,80	54,00	60,00
25	4	QPSK	1/2	52,00	57,60	108,00	120,00
26	4	QPSK	3/4	78,00	86,80	162,00	180,00
27	4	16-QAM	1/2	104,00	115,60	216,00	240,00
28	4	16-QAM	3/4	156,00	173,20	324,00	360,00
29	4	64-QAM	2/3	208,00	231,20	432,00	480,00
30	4	64-QAM	3/4	234,00	260,00	486,00	540,00
31	4	64-QAM	5/6	260,00	288,80	540,00	600,00

Nota. MCS= Sistema de Modulación y Codificación. Fuente: lacuevawifi.com

Elaborado por: Danny Ortiz

En la tabla 1 se puede observar la distribución de los MCS para 802.11n, con sus respectivas secuencias de datos y codificación con relación a la modulación y velocidad tanto reales como teóricas en los canales de transmisión para este estándar. Para alcanzar los valores tanto de modulación como de tasa de datos transmitidos en los transmisores y receptores es necesario usar la tecnología MIMO (Multiple Input Multiple Output).

El objetivo de la tecnología MIMO es realizar la transmisión de las cadenas de datos por medio del uso de múltiples antenas tanto en el transmisor como en el receptor, además de varios flujos de datos los que se envían a la misma frecuencia pero en diferentes canales espaciales. La tecnología MIMO está diseñada para funcionar tanto en la banda de 5 GHz y la banda de 2,4 GHz (CETECOM, 2014).

La cantidad de cadenas de bits usando MIMO, se multiplexan simultáneamente dependiendo inicialmente del número mínimo de antenas en el transmisor como en el receptor. Esto se expresa en la fórmula $A \times B:C$ (antenas transmisoras \times antenas receptoras : máximo de cadenas de bits). (TELEQUISMO, 2012).

Las configuraciones habituales en los productos comerciales actualmente disponibles son $2 \times 2:2$; $2 \times 3:2$ y $3 \times 3:2$, lo que significa:

$2 \times 2:2 \Rightarrow$ 2 antenas para Tx, 2 antenas para Rx y 2 cadenas de datos.

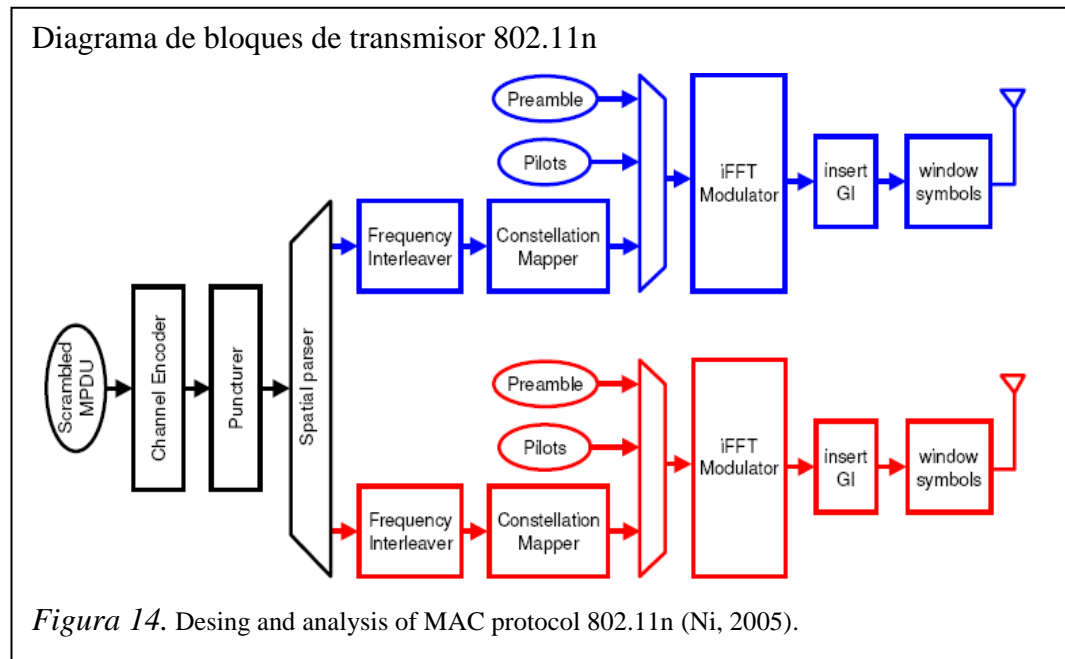
$2 \times 3:2 \Rightarrow$ 2 antenas para Tx, 3 antenas para Rx y 2 cadenas de datos.

$3 \times 3:2 \Rightarrow$ 3 antenas para Tx, 3 antenas para Rx y 2 cadenas de datos.

Por lo tanto con esta cantidad de antenas y el Coding Rate, se puede determinar que no es posible tener actualmente en el mercado dispositivos que brinden soporte para alcanzar la más alta velocidad estipulada en 802.11n. La tasa máxima provista por los productos actualmente en el mercado es de 300 Mbps. Esto es tasa de transferencia de bits por segundo. Los test publicados hasta el momento indican un throughput o tasa máxima de bits por segundo en el orden de los 100 Mbps para conexiones IEEE 802.11n (Mis libros de Networking, 2009).

2.1.2 Transmisor y Receptor del estándar 802.11n

El proceso de transmisión inicia con el circuito mostrado en la figura 14, donde se observan los bloques que se pueden utilizar para transmitir tramas mono usuario y multiusuario, El presente trabajo se centra en el caso de transmisión de un solo usuario, a pesar de que se trata de un sistema MIMO. (CHIMERA labs, 2014)



Para fines de explicación básica, solo se citan los bloques más pertinentes en el proceso de entramado y transmisión, ya que el fin de este proyecto es analizar lo resultante de la transmisión y recepción de los dispositivos bajo el estándar 802.11ac, es decir, la verificación de las tramas o palabras que se envían y las que llegan para medir el desempeño de la norma.

- **Enrutador de capa física (PHY Padding)**

Este bloque se encarga de añadir un número de bits (ceros) al final de la trama con el objeto de integrar el número de símbolos OFDM (Orthogonal Frequency Division Multiplexing). Inicialmente la MAC (Control de Acceso al Medio) crea una cadena y la PHY (Capa Física) añade de 0 - 7 bits a cada trama por usuario (Ni, 2005).

- **Scrambler o Transponder Inversor**

Después de la PHY (capa física), el scrambler tiene como propósito la manipulación de las tramas para realizar la tarea de destinar de forma aleatoria las secuencias largas de bits con similares valores añadiéndoles propiedades de transmisión de tramas. Los datos compuestos por el servicio, PSDU (Unidad de datos de servicio de la capa física), y demás partes alcanza un total de 127 tramas sincronizadas. Para la realización de este trabajo el Scrambler usa un generador polinomial.

$$S(x) = x^7 + x^4 + 1$$

El Scrambler es igual en el transmisor como en el receptor con el propósito de evitar errores de tramas, por lo tanto debe siempre iniciar en un estado bajo (Ni, 2005).

- **Espaciador Parser**

El espaciador Parser es un bloque que permite designar el codificador que se usará para transmitir los datos. En el caso de 802.11n tiene cuatro codificadores BCC (Codificador Convolutivo Binario), con el fin de aumentar la tasa de datos a ser transmitidos. La cantidad de codificadores define el número de datos a ser transmitidos, el ancho de banda y las salidas MIMO que tiene el transmisor. En el caso de 802.11n como se mencionó son 4 codificadores.

Las cadenas de bits que salen de los codificadores FEC (Forward Error Correction) se los divide en pequeños bloques de bits con el fin de reorganizarlos espacialmente para la transmisión MIMO, de esta manera se puede esperar que los bloques sean asegurados en la recepción mientras se los analiza para rearmar la trama recibida.

- **Frecuency Interleaver o Reorganizador de Frecuencia**

La función de este bloque, es la de separar los símbolos OFDM y enviar los datos dentro de los bloques asignados según se ha realizado la debida permutación de datos, por lo que se debe efectuar los siguientes tres pasos.

1. Primero los pequeños bloques deben estar correctamente organizados espacialmente antes de ser enviados.
2. Segundo es asegurar que las permutaciones de bloques estén correctamente ordenadas desde los más pequeños hasta los más grandes, desarrollando constelaciones de bits.
3. Y finalmente asignar los datos a la frecuencia que se usará según la constelación que se tenga para ser enviados.

- **Preamble y Pilots (Preámbulo y pilotos)**

Estos bloques se encargan de generar cadenas de bits que servirán de preámbulo de identificación de conexión y enlace con los dispositivos receptores.

Las cadenas pilotos son las encargadas de realizar el tanteo de la existencia de dispositivos de red 802.11, por medio del envío de tramas de verificación las que son reenviadas al AP en el caso de que existiera un receptor 802.11.

En el caso de las cadenas de preámbulo en cambio son las cadenas de cabecera de identificación de las tramas en cada codificador y mapeador de trama en 802.11. Esta cadena de bits se envía antes de cualquier transmisión de datos.

- **Modulador**

El modulador permite que la señal sea modulada de acuerdo al índice del esquema de Modulación y Codificación (MCS). Como se muestra en la tabla 2. La modulación puede ser BPSK, QPSK, 16-QAM y 64-QAM. El índice de codificación puede ser 1, 2, 3, 4 y determina el indicador y la modulación usada (Conalep Tlalpan, 2011).

Tabla 2

Indicador Bit y Modulación

Modulación	Bit
BPSK	1
QPSK	2
16-QAM	3
64-QAM	4

Nota. Desing and analysis of MAC protocol 802.11n (Ni, 2005)

Elaborado por: Danny Ortiz

La tabla 2 presenta el índice de codificación definido por un bit el que permite identificar la modulación que se debe usar en la trama de 802.11n.

- **Bloque de Intervalo de Guarda (IG)**

En este bloque el entramado se separa por medio de un Intervalo de Guarda para evitar las colisiones. En 802.11n el valor de la guarda es de 400ns (nanosegundos) en el ancho de banda de 20 MHz y 800ns (nanosegundos) en 40 MHz.

Al tener una separación de tiempo se garantiza que las tramas tengan un tiempo de separación (IG) a fin de que el entramado y mapeado, no tenga interferencia, desvanecimiento o Jitter, así se espera que los datos lleguen de forma segura al receptor (Ni, 2005).

Finalmente el resto de bloques se encargan de enviar secuencias pilotos para iniciar la transmisión y darse a conocer espacialmente como transmisor. También se efectúa una nueva revisión de los bloques de bits a mapeados espacialmente. Y después de la comprobación del mapeo bajo la modulación correcta y el símbolo requerido OFDM, se realiza el aventanamiento, lo que significa en realidad, el envío de cadenas de datos junto con la secuencia de identificación del dispositivo y los recursos de enlace con el receptor.

2.2 Descripción del estándar 802.11ac

El incremento de grandes grupos de información y la existencia de estándares de mayor calidad en audio y video, demandan mayores tasas de transmisión, así que aparece 802.11ac, en sucesión al estándar 802.11n.

El estándar 802.11ac al que también se le conoce como Gigabit Wi-Fi, es una especificación de la familia 802.11 aplicable a las redes de área local inalámbricas (WLAN). El estándar 802.11ac representa una extensión o actualización de la norma 802.11n actualmente activa. (Al-ghazu, 2013) .

Las redes que utilizan 802.11ac operarán exclusivamente sobre la banda de 5.8 GHz usando la modulación OFDM (orthogonal frequency division multiplexing) de forma similar a como lo hace 802.11a. Las mejoras soportadas por la 802.11ac posibilitan la transmisión simultánea de video HD (alta definición) a varios clientes, así como la sincronización inalámbrica más rápida y compartimiento de archivos de gran tamaño. Las características que tiene la norma 802.11ac, además de aquellas existentes en las normas 802.11a y 802.11n, son:

- Ancho de canal de hasta 160 MHz (megahertz).
- Single-link (enlace sencillo) con salida de 500 Mbps o más.
- Multi-station WLAN con salida de 1 Gbps (gigabit por segundo) o más.
- 400 ns (nanosegundos) intervalo de guarda corto.
- Código de chequeo de paridad de baja-densidad.
- Codificación de bloqueo espacio-tiempo.
- Hasta 8 flujos espaciales.
- Transmisión de formación de haz.

2.2.1 Modulación y Velocidad 802.11ac

Mientras que el resultado más evidente de las velocidades más altas son las transferencias de datos más rápidas para los usuarios individuales, la mayor ventaja para los administradores de redes puede ser la habilidad de manejar un mayor número de dispositivos sin degradaciones inaceptables en el rendimiento.

Dado el hecho de que más y más usuarios están accediendo a la red inalámbrica con varios dispositivos (teléfonos, tabletas y otros dispositivos smart), la actualización del AP a 802.11ac puede proporcionar alivio a un costo razonable. La ventaja acumulada de las características de 802.11ac permitirá que las soluciones Wi-Fi cumplan con la demanda actual de las aplicaciones en tiempo real de teléfonos de alta calidad y capacidad, tales como vídeo y voz (Avila, 2014).

El estándar 802.11ac logra su aumento de velocidad pura de varias maneras:

- Más unión de canales, incrementado del máximo de 40MHz en 802.11n y ahora hasta 80 y pronto 160 MHz.
- Una modulación más densa, ahora usando modulación de amplitud en cuadratura (QAM) de 256, comparada con 64 QAM de 802.11n.
- Un número más elevado de flujos espaciales. 802.11ac es compatible con ocho flujos espaciales, más que los cuatro de 802.11n.
- La simplificación de la capacidad de formación de haces de transmisión, que primero fue introducida con 802.11n.

La nueva tecnología MU-MIMO o MIMO de varios usuarios, permite a un AP enviar tramas a varios usuarios al mismo tiempo en la misma frecuencia. Así que por primera vez, un AP puede actuar algo así como un switch de Ethernet en lugar de un concentrador, asistiendo a más usuarios.

Tal como en el caso de 802.11n en 802.11ac el MU MIMO, da la factibilidad de enviar la información por varias antenas y la recepción igualmente por medio de

varias antenas, por lo que las antenas de MU MIMO son hasta en grupos de 3x3:3 como máximo.

El estándar 802.11ac solo funciona en la banda de 5 GHz menos congestionada, así que los clientes de banda doble y AP podrán obtener acceso a 802.11n en la banda de 2.4 GHz. El diseño de 802.11ac destaca la compatibilidad con los estándares existentes de Wi-Fi y puede coexistir con dispositivos de 802.11a/n. El diseño y la implementación de la red pueden maximizar los beneficios de la implementación de 802.11ac sin dejar de ser compatibles con los dispositivos más antiguos. (Fluke Networks, 2014).

Una forma fácil de ver el acondicionamiento del ancho de banda, la velocidad, el tipo de antenas usadas y la distribución de los MSC en 802.11ac es la Tabla 3 y Tabla 4 (Thornycroft, 2013) .

Tabla 3

Tasa de datos según el ancho de banda en 802.11ac.

Ancho de Banda (MHz)	Antenas transmisor - receptor	Escenario	velocidad máxima (Mbps)
40	3x3	pc	606
80	1x1	smartphone	433
80	2x2	tablet, pc	867
80	3x3	pc	1300
160	1x1	smartphone	867
160	2x2	tablet, pc	1,73
160	4x AP; 4 clientes de 1	múltiple smartphone	867 por cliente
160	8x AP; 4 clientes de 2	digital Tv, set-top box, tablet, pc, Smartphone	867 a 2 clientes de 1; 1730 a 1 cliente de 2; 3470 a 1 cliente de 4

Nota. Gigabit WiFi, 802.11ac defn (Thornycroft, 2013)

Elaborado por: Danny Ortiz

Tabla 4

Tasa de datos según MCS en 802.11ac.

MCS	Tasa de datos (Mbps) 40MHz		Tasa de datos(Mbps) 160 MHz		Ancho de canal	Tramas espaciales
	40 ns	80 ns	40 ns	80 ns		
0	6,5	7,2	468	520	x2,1 por 40 MHz; x4,5 por 80 MHz; x9 por 160MHz	x2 para 2; x3 para 3; x4 para 4; x5 para 5; x6 para 6; x7 para 7; x8 para 8
1	13	14,4	939	1040		
2	19,5	21,7	1404	1560		
3	26	28,9	1872	2080		
4	39	43,3	2808	3120		
5	52	57,8	3744	4160		
6	58	65	4212	4680		
7	6	72,2	4680	5200		
8	78	86,7	5616	6240		
9	86,7	96,3	6240	6933,3		

Nota. Gigabit WiFi, 802.11ac deft (Thornycorf, 2013)

Elaborado por: Danny Ortiz

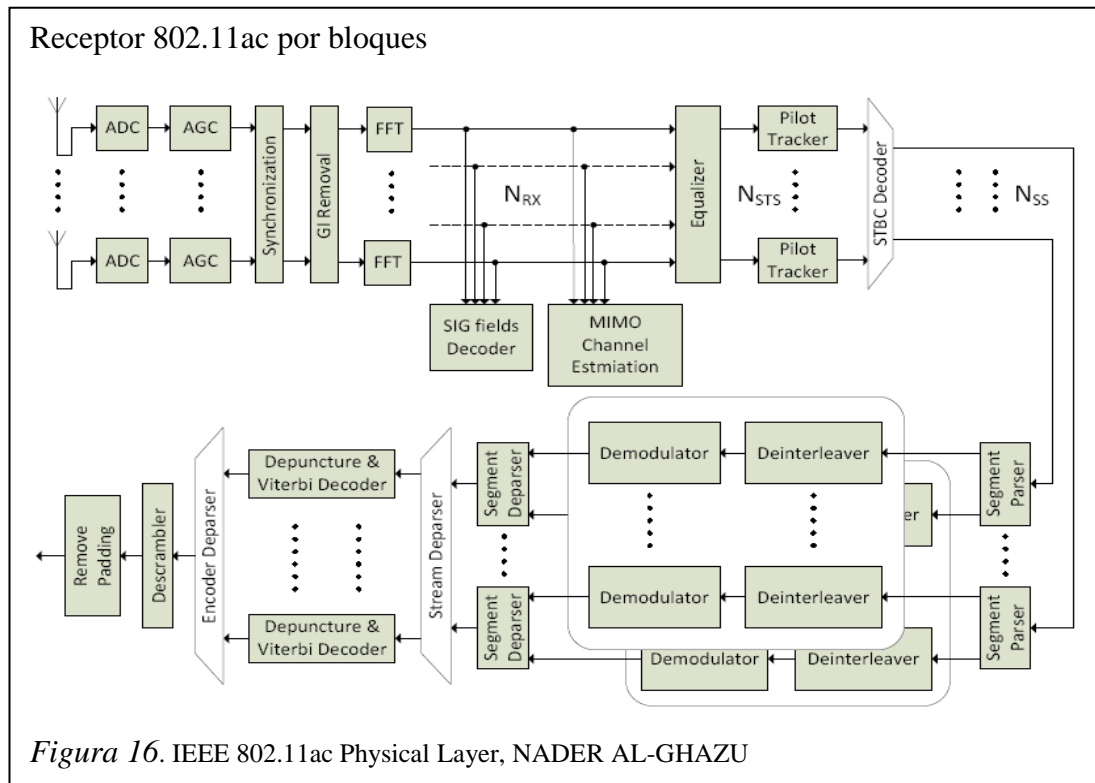
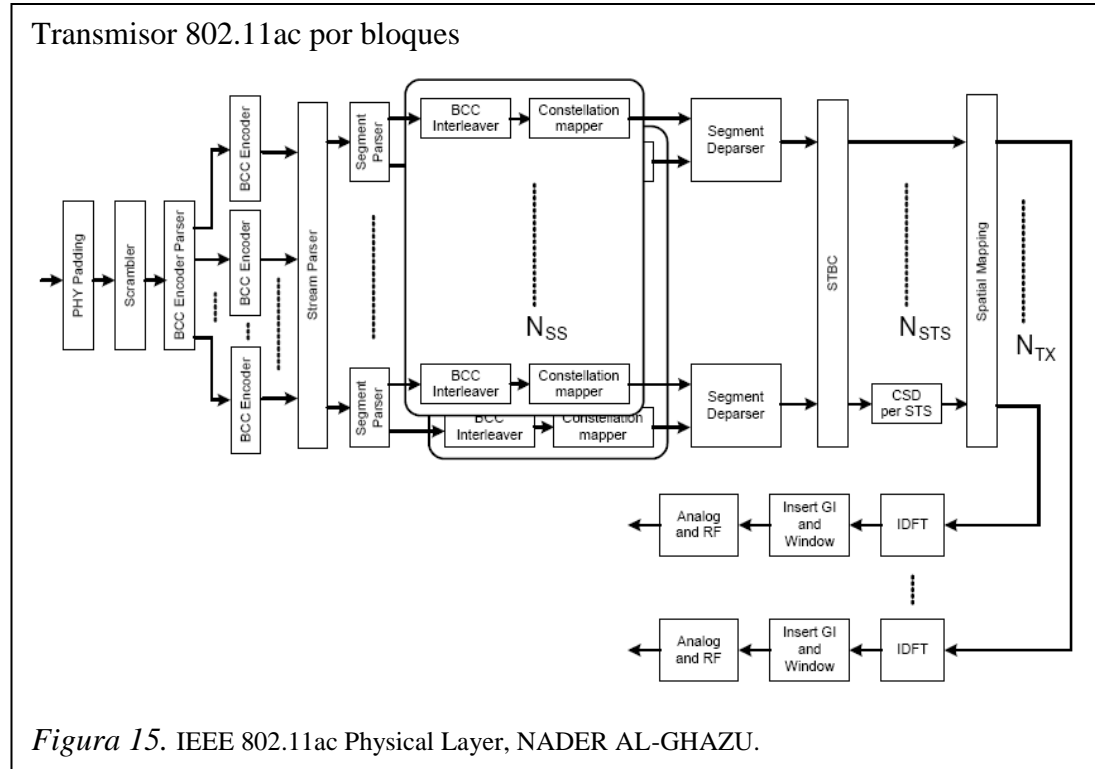
Estas dos tablas permiten revisar las tasas de datos que se requieren para las transmisiones actuales entre dispositivos wireless. Además se puede observar las velocidades reales y teóricas de acuerdo con el MCS.

2.2.2 Transmisor y receptor 802.11ac

El transmisor 802.11ac al ser una mejora del 802.11n, en realidad consta del mismo proceso de convolución, entramado, mapeado y de intervalo de guarda; por lo que el análisis de bloques es similar al anterior pero con la diferencia que el número de codificadores BCC aumenta a 6 en vez de 4 como en 802.11n, con esta cantidad de codificadores se obtiene una modulación diferente.

La cantidad de bits que se usan para definir el tipo de modulación, ahora se añade el dígito 8 por lo que las modulaciones que intervienen en el proceso de transmisión son BPSK, QPSK, 16-QAM, 64-QAM y 256-QAM. Por lo demás el polinomio generador, los codificadores de errores, las permutaciones de los bloques, el mapeo y

el proceso final de comprobación, mapeo y envío son exactamente igual al de 802.11n. Para verificar esta afirmación se presenta la figura 15 y figura 16, donde se destacan los bloques que forman parte esencial del transmisor 802.11ac.



- **PHY Padding o Enrutador de Capa Física**

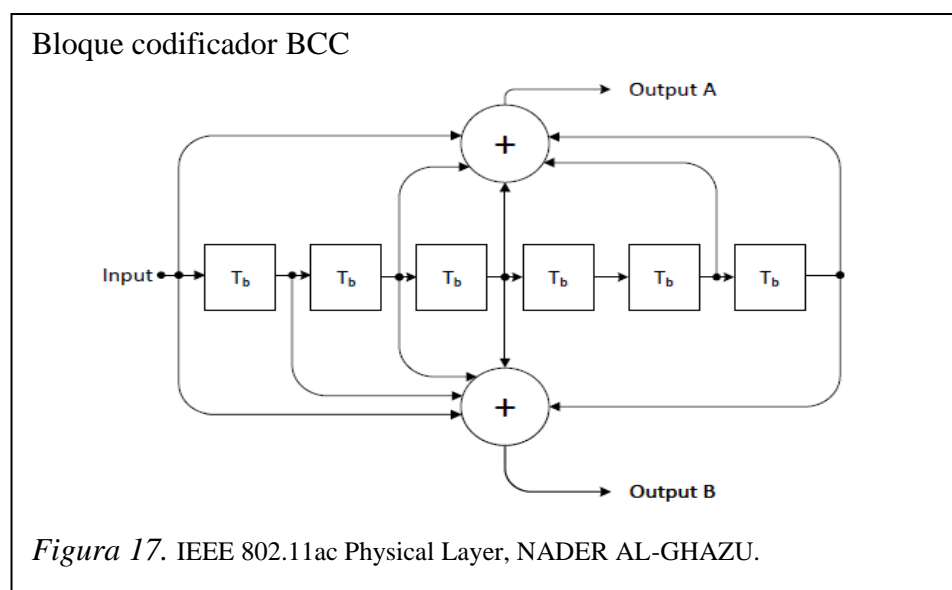
Este bloque añade tal como en 802.11n, un número de bits (ceros) al final de la trama con el fin de integrar el número de símbolos OFDM. Inicialmente la MAC crea una cadena y la PHY añade de 0 - 7 bits a la trama por usuario. (AL-GHAZU, 2013).

- **Scrambler**

Este bloque tiene como propósito la manipulación de las tramas, para realizar la tarea de destinar de forma aleatoria las secuencias largas de bits con similares valores añadiéndoles propiedades de transmisión de tramas. Los datos compuestos por el servicio, PSDU (Unidad de datos de servicio de la capa física), y demás partes alcanza un total de 127 tramas sincronizadas. El scrambler es igual en el transmisor como en el receptor con el propósito de evitar errores de tramas, por lo tanto debe siempre iniciar en un estado bajo. (AL-GHAZU, 2013).

- **Realimentador de Corrección de error de código (FEC)**

La trama de datos pasa por uno de los dos codificadores FEC, del transmisor 802.11ac. Para realizar este cometido la BCC codifica con el polinomio generador, dentro de cualquier longitud de 7 dígitos, para ello usa el proceso de realimentación y corrección de error en el código. Como se observa en la figura 17 de forma similar a 802.11n.



- **Codificador Parser**

En el caso de 802.11n son cuatro los codificadores BCC, sin embargo en 802.11ac son 6 con el fin de aumentar la tasa de datos a ser transmitidos. Con esto se define que el número de codificadores BCC está definido por el número de datos a ser transmitidos, el ancho de banda y las salidas MIMO (2x2; 2x3; 2x4; 3X3 y 4x4) que tiene el transmisor (AL-GHAZU, 2013).

- **Encadenador Parser**

Las cadenas de bits que salen de los codificadores FEC se dividen en pequeños bloques de bits con el fin de reorganizarlos espacialmente para la transmisión MU-MIMO, de esta forma se puede esperar que los bloques transmitidos sean asegurados en el receptor mientras se los analiza para rearmar la trama recibida.

- **BCC Interleaver o Codificador convolucional binario**

La función de este bloque circuital al igual que en la norma 802.11n es la de separar los símbolos OFDM y enviar los datos dentro de bloques asignados según se haya realizado la debida permutación de datos, por lo que se debe efectuar los siguientes tres pasos. Primero los pequeños bloques deben estar correctamente organizados espacialmente antes de ser enviados. Segundo asegurar que las permutaciones de bloques estén correctamente ordenadas desde los más pequeños hasta los más grandes, desarrollando constelaciones de bits. Y finalmente asignar los datos a la frecuencia que se usará, según la constelación que se tenga para ser enviados.

- **Mapeador**

En este proceso se usa los bits como indicadores para saber qué tipo de modulación se usará al enviar las tramas. Para eso se usa 1, 2, 4, 6, 8 teniendo así modulaciones BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM (AL-GHAZU, 2013). En la tabla 5 se muestra el indicador y la modulación usada, para la modulación OFDM.

Tabla 5

Indicador Bit y Modulación

Modulación	Bit
BPSK	1
QPSK	2
16-QAM	4
64-QAM	6
256-QAM	8

Nota. A Study of the Next WLAN Standard IEEE 802.11ac Physical Layer (Al-ghazu, 2013)

Elaborado por: Danny Ortiz

Igual que en el caso de 802.11n, en 802.11ac el índice de codificación dado por el bit de cabecera dentro de la trama 802.11ac, define el tipo de modulación que se usará al enviar las tramas por el aire.

- **Bloque de Intervalo de Guarda (IG)**

Este bloque tiene como objeto dar un espacio temporal al entramado para evitar las colisiones entre tramas enviadas. Tal como sucede en 802.11n los IG, dependen del ancho de canal por lo tanto el valor será de 400ns en el caso de un ancho de banda de 40MHz y 800ns en el caso de 80 y 160 MHz. De esta forma se garantiza que las tramas con los datos de información tengan un tiempo de separación a fin de que el entramado y mapeado, no causen interferencia, desvanecimiento o Jitter, entre sí; de tal forma que los datos lleguen de forma segura al receptor (AL-GHAZU, 2013).

Finalmente el resto de bloques se encargan de enviar secuencias pilotos para iniciar la transmisión y darse a conocer espacialmente como transmisor. También se encargan de efectuar una nueva revisión de los bloques de bits a mapeados espacialmente. Y después de la comprobación del mapeo bajo la modulación correcta y el símbolo requerido OFDM, se realiza el aventanamiento que significa en realidad el envío de cadenas de datos junto con la secuencia de identificación del dispositivo y los recursos de enlace con el receptor.

2.3 Migración de 802.11n a 802.11ac (FLUKE networks)

Después de haber hecho un reconocimiento y estudio de los sistemas de transmisión 802.11n y 802.11ac; se ha podido establecer que 802.11ac es mucho más rápido que 802.11n.

Siendo esta la situación sería muy fácil para alguien sin conocimientos profundos sobre el tratamiento de redes, definir una migración instantánea de las redes 802.11n a 802.11ac. Por lo tanto es importante resistir la tentación de realizar un sencillo remplazo 1 a 1 de una red 802.11n existente o incluso tecnologías anteriores como APs 802.11a/b/g/n con nuevos APs 802.11ac.

Algunos usuarios creen que la sustitución 1 a 1 de los APs es el método más barato y rápido de desarrollar la nueva tecnología y que los esfuerzos pasados para las implementaciones pueden ser simplemente reutilizados o reconfigurados para todas las tecnologías futuras sin ninguna planificación real. Por esta razón la falsa creencia del simple cambio 1 a 1 de dispositivos con lleva a un diseño de red pobre y costoso, ya que los dispositivos receptores siempre trabajarán a un nivel máximo aun teniendo un AP de alta velocidad como 802.11ac, por lo que a futuro implicara costos.

Los usuarios deberían definir la estrategia de migración en vez de implementaciones a ciegas como reemplazar APs. Esto incluye hacer coincidir las instalaciones tecnológicas con los requisitos empresariales o de hogar, estimar presupuestos e imaginar la ubicación y configuración de los APs para maximizar la cobertura y rendimiento antes de cualquier instalación de APs.

Igual que 802.11n, además de planear la ruta de migración, también es importante realizar estudios del mundo real para implementar la red más precisa, puesto que se debe tomar en cuenta todas las condiciones ambientales y de red.

La medición de la capacidad de transmisión es el único indicador real del rendimiento 802.11ac y se basa en tecnologías introducidas en 802.11n como MIMO, formación de haces de tramas, canales más amplios y flujos espaciales adicionales. Debido a todas estas capacidades, la intensidad de la señal no es un verdadero indicador del rendimiento WLAN (Fluke Networks, 2014).

Como se indicó anteriormente, 802.11ac es compatible con versiones anteriores de 802.11n y 802.11a y funciona en un entorno de modo mixto en la banda de 5G Hz.

El rendimiento para los clientes de 802.11ac puede verse afectado negativamente debido a velocidades de transmisión más lentas por los clientes de 802.11a/n.

El estándar 802.11ac presenta la función amplia de canales de 80MHz y 160 MHz que habilita una capacidad de transmisión más alta. El uso de canales más anchos en 802.11ac aumenta la probabilidad de interferencia de canales contiguos y esto afecta negativamente al rendimiento.

2.4 Sustentación Legal

Las normativas que permiten el desarrollo de las tecnologías deben estar sustentadas en marcos regulatorios legales como las constituciones de cada estado de forma nacional y de forma internacional por instituciones internacionales como la IEEE.

2.4.1 IEEE (Instituto de Ingenieros Eléctricos y Electrónicos)

Con los avances tecnológicos también se ha visto necesario la regulación y estandarización de algunos elementos de las comunicaciones, por esta razón se han desarrollado varias instituciones con el fin de brindar una mejor funcionalidad de los dispositivos garantizando mejores servicios a los usuarios. Una de estas instituciones es la IEEE (Institute of Electrical and Electronics Engineers).

Esta institución mundial tiene como propósito principal potencializar la innovación y excelencia tecnológica para el beneficio de la humanidad (IEEE, 2014). Siendo esta la causa principal de su creación se ha visto identificada con la estandarización y universalización de muchas herramientas, dispositivos y aparatos eléctricos y electrónicos, así como el acondicionamiento de las contribuciones tecnológicas de profesionales de la industria eléctrica y electrónica (IEEE, 2014).

Los estándares más estudiados por la IEEE son los siguientes:

- 802, normativas sobre estándares de redes.
- 802.11, normativas sobre estándares de redes inalámbricas.
- 1394, normativas aplicada a conexiones de alta velocidad.
- 488, normativa para buses de testeo y medida.
- 754, Normativa para cálculos de coma flotante.
- POSIX, normativa de interfaz y entorno de sistema operativo.
- VHDL, normativa para circuitos integrados digitales.

De estas normativas, el campo de estudio relativo a esta investigación es el 802.11n y 802.11ac, para efectuar el análisis comparativo del desempeño general.

2.4.2 Normativa Internacional

2.4.2.1 Estándar IEEE 802.11n y IEEE 802.11ac

La IEEE es la encargada de regular el uso, la propiedad intelectual, la seguridad, el medio ambiente, la interferencia de leyes y regulaciones que existan en la manipulación de herramientas y dispositivos que trabajen en frecuencias de 2,4 Ghz y 5,8 Ghz, como es el caso de los estándares 802.11, con el fin de fomentar la igualdad de condiciones de trabajo en todo el mundo, permitiendo así que exista un orden y una equitativa competencia industrial.

La norma IEEE 802.11n es desarrollada y presentada mundialmente en un borrador en noviembre de 2007, para ser aprobada en septiembre de 2009. Y la norma IEEE 802.11ac que es una actualización de 802.11n fue aprobada mundialmente en enero del 2013. (IEEE, 2013)

2.4.3 Normativa nacional

Las normativas nacionales están reguladas por Instituciones como el Consejo nacional de Telecomunicaciones CONATEL, que se encarga de establecer las políticas y normas de regulación de servicios de telecomunicaciones. La Secretaria

Nacional de Telecomunicaciones SENATEL, la que se encarga de ejecutar las políticas y normas establecidas por el CONATEL y la Superintendencia Telecomunicaciones SUPERTEL, la que tiene como fin la supervisión de personas naturales y jurídicas, tanto públicas como privadas a fin de que cumplan con lo establecido en las políticas que rigen el país en función de las telecomunicaciones. (TELECOMUNICACIONES, 2000).

La ley ecuatoriana para la regularización de las telecomunicaciones es muy extensa y tiene algunos reglamentos como los que a continuación se menciona (Cando, 2007):

- Reglamento para el servicio de telefonía celular móvil.
- Reglamento para la prestación de servicio móvil avanzado.
- Reglamento de interconexión.
- Reglamento para la prestaciones servicios portadores.
- Reglamento para la homologación de equipos de telecomunicaciones.
- Reglamento de radio comunicaciones.
- Norma para la implementación y operación de servicios de banda ancha
- Otros.

2.4.3.1 Uso del estándar IEEE 802.11n y IEEE 802.11ac en el Ecuador

Para la creación de WLAN bajo este estándar se debe tomar muy en cuenta lo que dice el Art. 15 y 16 de la Ley especial de Telecomunicaciones, donde menciona que para la implementación de este tipo de redes no se necesita ningún permiso habilitante, pero con la condicionante que solo sea privada, no para uso de prestación a terceros (TELECOMUNICACIONES, 2000).

Por lo tanto si se desea implementar una red bajo la norma 802.11n o la norma 802.11ac, no existe ningún impedimento, si se desea efectuar pruebas físicas con equipos reales.

2.5 Parámetros de diseño

Para efectuar el diseño de redes se requiere que existan algunos parámetros imprescindibles los que serán analizados a continuación:

- **Dispositivos**

- a. Estación móvil bajo el estándar requerido, Pc, Tablet, etc.
- b. Access Point o AP con el estándar establecido, Router 802.11x.
- c. Dispositivo de capa 2, en este caso Bridge que une la red 802.11 y la red Ethernet, Switch.

- **Problemas en la transmisión**

- a. La transmisión inalámbrica sufre pérdidas significativas de potencia y calidad al atravesar obstáculos como paredes de ladrillos y concreto. Pero puede pasar sin problemas por tableros de madera, plástico y vidrio.
- b. Desvanecimiento, este problema se presenta en las redes inalámbricas por la cantidad de señales dentro del mismo canal de transmisión. Otro factor que puede generar el desvanecimiento la distancia entre la estación y el AP.
- c. La aparición del BER.
- d. La desviación por el Jitter.

2.6 Programación orientada a objetos

Después de haber efectuado el debido estudio de los parámetros que intervienen en una red 802.11n y 802.11ac, por medio de la revisión de los datos más relevantes de cada estándar, se procede a analizar el software de simulación. Este software es la base fundamental para el desarrollo de este estudio, es la parte medular de este proyecto. Pero antes de optar por algún software de simulación se debe efectuar un estudio de los parámetros incluidos en una simulación, como, el tipo de lenguaje de programación que se va a usar, el formato de cada programa, la forma de esquematizar las redes, entre otros factores.

Por lo tanto desde este punto en adelante se va a analizar lo referente a la programación de redes simuladas en dos software, OMNet++ y NS3. Entonces se prosigue al estudio de estas dos plataformas de simulación.

2.6.1 Lenguaje C++

Este lenguaje desarrollado como base primordial de sistemas de programación, es la base de aplicaciones como OMNet++. Este lenguaje de programación permite desarrollar clases o estructuras como módulos que forman bloques de aplicaciones conjuntas para cierto fin.

Los bloques desarrollados en C++, permiten tener varias aplicaciones que como en el caso de OMNet++ sean usados para la simulación de situaciones de la vida real, como en este caso de redes de información.

Dado que los módulos simples son considerados en OMNet++ como los componentes activos de la red, son los únicos elementos programados en C++ a través de archivos .cpp o .cc, mediante la librería de OMNet++ (omnetpp.h), lugar donde ocurren los eventos para ejecutar una simulación. Mediante el uso del lenguaje C++ donde se programarán los módulos o elementos que conformarán la red de comunicación para la simulación.

Es importante resaltar que OMNet++ se basa en la plataforma de Visual Studio, desarrollado por Microsoft, la cual se compone de un entorno de desarrollo integrado (IDE) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic.NET (Sánchez, 2011). Como se trata de un lenguaje especializado tiene una estructura propia la misma que tiene sus propios archivos ejecución. El tipo de archivos tienen una terminación .NED.

2.6.2 Lenguaje NED

Una de las partes primordiales en el desarrollo de un programa bajo OMNet++ es definir el diseño en lenguaje .NED. Este lenguaje es una mezcla de C++ y Visual Studio, por lo que se debe usar algoritmos que son compatibles con estos programas

de simulación. Pero la estructura de la programación .NED es algo propio, por lo que se citan a continuación como se distinguen los módulos que corresponden al lenguaje .NED (Sánchez, 2011).

- Definición de módulos simples.
- Definición de módulos compuestos.
- Definición de redes.

El usuario define la estructura del modelo a través de los módulos y sus conexiones por medio del lenguaje NED, en OMNET++. Básicamente los elementos para este lenguaje son la declaración de los módulos y la definición de las características de la red.

La interfaz de los módulos se compone de compuertas y parámetros. Mientras que la definición de un módulo compuesto se compone de la declaración de la interface externa de módulos con sus respectivos parámetros y compuertas, así como la definición de las conexiones.

NED establece el comportamiento para los módulos declarados en la red. Habiendo hecho esto y después de haber creado un archivo .ini para la configuración de la simulación, se presenta el GUI (Tkenv) donde se corre la simulación y se observa el comportamiento de la red. Por otra parte se cuenta con las herramientas de graficación utilizadas para graficar variables escalares y vectoriales que emplea OMNET++ para registrar el desempeño de la red (Sánchez, 2011).

2.6.2.1 Módulos simples.

La estructura esquemática de un módulo NED simple se define de la siguiente forma:

- Parámetros: numeric, numeric const, bool, string, xml.
- Puertas: in u out, pueden ser arrays.

Ejemplo de un módulo simple.

simple Modulo

parameters:

p1 : tipo;

p2 : tipo;

gates:

in: g1;

out: g2;

out: g3[];

endsimple;

2.6.2.2 Módulos compuestos

La unión de módulos simples permite componer módulos complejos a estos nuevos módulos se los denomina compuestos. A los módulos compuestos se los usan para interconectar módulos. Con los módulos compuestos se pueden realizar múltiples niveles de composición.

Ejemplo de la estructura de un módulo compuesto:

module nombre

parameters:

//...

gates:

//...

submodules:

//...

connections:

//...

endmodule

2.6.2.3 Redes

En Omnet++ una red es un prompt gráfico sobre la que se asientan los módulos simples o compuestos, simples y compuestos para el desarrollo de un ejercicio de lenguaje NED a este proceso se le denomina Instanciación de la red. Las redes pueden contener asignación de parámetros, esto permite que los procesos de programación sean en orden eventual.

Ejemplo de un esquema de red

```
network red : Red  
endnetwork;
```

2.6.3 Pasos para ejecutar una aplicación tipo NED (Martínez, 2010)

1. Definir la estructura de la red. En este caso se usa el lenguaje NED.
2. Completar el comportamiento. La complementación del diseño de un prototipo se la hace bajo el lenguaje C++.
3. Configurar la simulación. Para configurar una simulación se debe efectuar una compilación y ejecución del prototipo diseñado, a lo cual se genera un archivo .ini.

2.7 OMNET++

OMNet++ (Objective Modular Network Testbed in C++) es un simulador de red basado en C++ que permite modelar redes de comunicaciones cableadas e inalámbricas disponible desde 1997, la ventaja de éste simulador sobre otros es la gran cantidad de módulos independientes creados para dar soporte a funciones específicas y que se integran de manera natural (OMNETT.ORG, 2014).

La meta de esta aplicación es simular redes de comunicaciones pero en vez de ser construido como un simulador especializado, se diseñó para ser tan general como fuera posible. Como resultado, se ha implementado exitosamente en varias áreas que van desde el modelado de redes de telecomunicaciones hasta la validación de arquitectura de hardware.

El modelado con OMNet++ es posible en cualquier sistema donde se pueda usar la aproximación a eventos discretos o donde las variables de estado cambien instantáneamente en momentos separados en el tiempo (Martínez, 2010). Un modelo en OMNet++ está compuesto por las siguientes partes:

- **Archivos de descripción de red (.ned):**

Estos describen la estructura del modelo con parámetros, conexiones, submódulos, etc. Los archivos NED pueden ser escritos directamente mediante líneas de texto o pueden ser generados usando la utilidad de diseño (GNED).

El lenguaje NED facilita la descripción modular de una red, la descripción de una red puede consistir en una serie de componentes. Los canales, módulos simples y módulos compuestos de una red pueden ser reusados en otra red.

- **Definiciones de mensaje (.msg):**

Los tipos de mensaje y su estructura deben estar especificados en estos archivos, pueden modelar varias cosas: eventos, mensajes, paquetes, tramas, celdas, bits o señales viajando a través de una red. Después de compilar el proyecto, OMNet++ traducirá estas definiciones a clases en C++.

- **Código fuente (.h/.cc):**

Estos son los archivos en C++ que definen las acciones de los módulos simples, los componentes activos en el modelo. Los módulos simples generan eventos y reaccionan a ciertos eventos, es decir, implementan el comportamiento del modelo.

- **Archivos de configuración (.ini):**

Para ejecutar una simulación, OMNet++ requiere un archivo de configuración. El archivo de configuración está dividido en secciones y contiene información

tal como: parámetros generales, parámetros específicos de la ejecución, almacenamiento de vectores de salida, especificación de valores para ciertos módulos. En la figura 18 se muestra una vista típica de la interfaz gráfica de OMNet++ (Sánchez, 2011).

Pantalla de inicio de OMNet++ 4.1

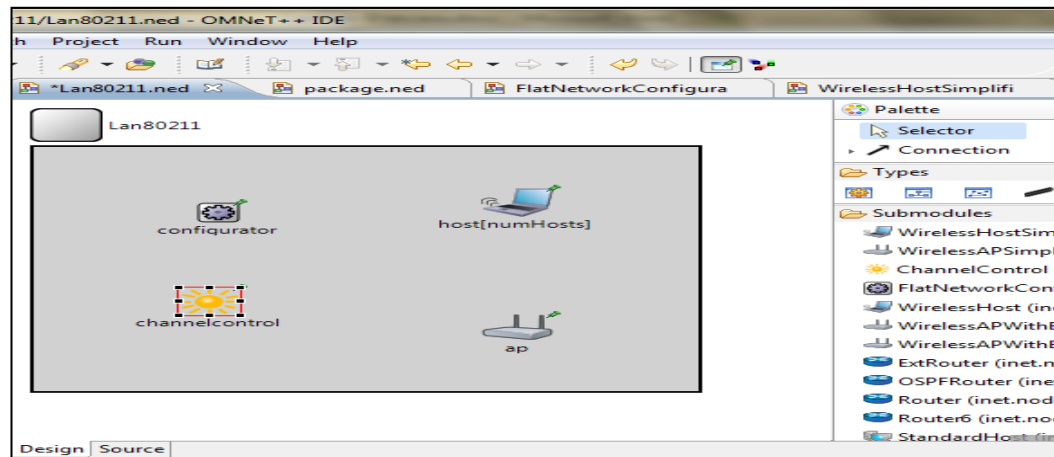


Figura 18. Captura de pantalla.

Elaborado por: Danny Ortiz.

Los modelos en OMNet++ consisten en módulos que se encuentran anidados jerárquicamente, estos módulos se comunican pasando mensajes entre ellos. El nivel más alto se denomina módulo de sistema, este contiene los submódulos que a su vez pueden contener submódulos. Los módulos simples se encuentran en el nivel más bajo del modelo y contienen los algoritmos del modelo y están implementados en C++, esto se puede apreciar en la figura 19 (Martínez, 2010).

Estructura de módulos OMNet++

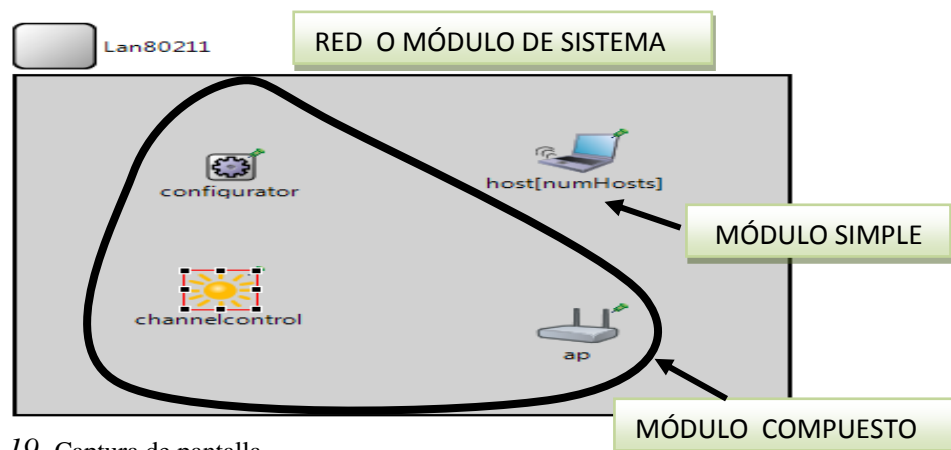


Figura 19. Captura de pantalla.

Elaborado por: Danny Ortiz

Los módulos pueden tener parámetros, pueden tomar cadenas de texto, valores numéricos o booleanos así como árboles de datos en XML. Los valores numéricos incluyen expresiones que utilizan otros parámetros o llaman a funciones en C, variables aleatorias de diferentes distribuciones y valores ingresados interactivamente por el usuario (Martínez, 2010).

Un módulo es una instancia de un tipo de módulo. Los módulos pueden estar anidados y la profundidad de este anidamiento no está limitada, este método de modelado hace posible reflejar la estructura lógica del sistema real en la estructura del modelo.

2.8 NS3

NS3 (Network Simulator 3) fue desarrollado para proporcionar una plataforma de simulación de red abierta y extensible, para la creación de redes de investigación y educación. NS3 proporciona modelos de cómo realizar paquetes de datos de redes de trabajo, y proporciona un motor de simulación para los usuarios para llevar a cabo experimentos de simulación.

Algunas de las razones para utilizar NS3 incluyen la realización de estudios que son más difíciles o imposibles de realizar con los sistemas reales, para estudiar el comportamiento del sistema en un entorno altamente reproducible.

Existen muchas herramientas de simulación para estudios de simulación de redes. A continuación se presentan algunas de las características distintivas de NS3 en contraste con otras herramientas (NS3, 2014).

- NS3 está diseñado como un conjunto de bibliotecas que se pueden combinar entre sí y también con otras bibliotecas de software externos. Mientras que algunas plataformas de simulación proporcionan a los usuarios con un único entorno de interfaz con el usuario una interfaz gráfica en la que todas las tareas se llevan a cabo ahí, NS3 es más modular en este sentido. Ya que tiene varios animadores externos y el análisis de datos y herramientas de visualización también se pueden utilizar con NS3. Sin embargo, los usuarios deben esperar para trabajar en la línea de comandos y con las herramientas C++ o Python.

- NS3 se utiliza principalmente en los sistemas Linux, aunque existe soporte para FreeBSD, Cygwin (para Windows), y el apoyo de Visual Studio nativo de Windows se encuentra en el proceso de ser desarrollado.

2.8.1 Conceptos generales

2.8.1.1 Nodo

Un dispositivo informático que se conecta a una red se denomina un host o en ocasiones es también llamado un sistema final. En NS3 la abstracción básica de un dispositivo informático se llama nodo. Esta abstracción está representada en C++ por la clase Node.

La clase Node proporciona métodos para la gestión de las representaciones de los dispositivos informáticos en las simulaciones. Se debe pensar en un nodo como un equipo al que va a agregar funcionalidad. Se puede agregar cosas como aplicaciones, pilas de protocolos y tarjetas de periféricos con sus controladores asociados para que el equipo realice un trabajo útil.

2.8.1.2 Aplicación

Un usuario normalmente puede ejecutar una aplicación que adquiere y utiliza los recursos controlados por el software del sistema para lograr algún objetivo (NS3, 2014). A menudo, la línea de separación entre el sistema y el software de la aplicación se realiza en el cambio de nivel de privilegio que sucede en las trampas del sistema operativo.

En NS3 no existe un concepto real de sistema operativo y sobre todo el concepto de niveles de privilegio o llamadas al sistema. Al igual que las aplicaciones de software se ejecutan en los computadores para realizar tareas en el "mundo real", Las aplicaciones como NS3 realizan lo mismo en un mundo simulado.

2.8.1.3 Canal

En el mundo real, se conoce como canal al medio de transmisión por el que viajan las señales portadoras de información entre un emisor y un receptor. Es decir en los medios de comunicación sobre los que hay flujos de datos se denominan canales. Al conectar el cable Ethernet al Jack en la pared, se está conectando el equipo a un canal de comunicación Ethernet. En cambio en el mundo simulado de NS3, uno se conecta a un nodo o un objeto que representa un canal de comunicación. Aquí la abstracción básica de subred de comunicación se llama el canal y está representada en C++ por la Clase Channel.

La clase Channel proporciona métodos para la gestión de los objetos de comunicación de subred y la conexión de los nodos a ellos. Los canales también pueden ser especializados por los desarrolladores en el sentido de la programación orientada a objetos. Una especialización Channel puede modelar algo tan simple como un alambre hasta cosas tan complicadas como un conmutador Ethernet grande, o el espacio tridimensional lleno de obstrucciones en el caso de las redes inalámbricas.

Se utiliza las versiones especializadas del canal llamado CsmaChannel, PointToPointChannel y WifiNetChannel. El CsmaChannel por ejemplo, utiliza los modelos de una versión de una subred de comunicación que implementa un medio de comunicación de acceso múltiple por detección de portadora. Esto da la funcionalidad de Ethernet similares.

2.8.1.4 Dispositivo net

En épocas pasadas para conectar computadores entre sí, se tenía que comprar un tipo específico de cable de red y un dispositivo de hardware el cual era una tarjeta periférica que se necesitaba para ser instalada en su computador. Si la tarjeta periférica implementaba algunas funciones de redes, se la llamaba tarjeta de interfaz de red o NIC. Hoy en día las computadoras vienen con el hardware de interfaz de red incorporada y los usuarios no ven estos bloques de construcción.

En NS3 hay que construir estos bloques de red que cubre tanto el controlador de software y el hardware simulado. Un dispositivo de red está "instalado" en

un nodo con el fin de posibilitar que el nodo pueda comunicarse con otros nodos en la simulación a través de canales. Al igual que en un computador real, un nodo puede ser conectado a más de un canal a través de múltiples NetDevices.

La abstracción neta del dispositivo se representa en C++ por la clase Netdevice. Esta clase proporciona métodos para la gestión de las conexiones a los objetos de nodo y del canal y puede ser especializado por los desarrolladores en el sentido de la programación orientada a objetos.

Algunas clases que se pueden utilizar es la NetDevice con llamados CsmaNetDevice, PointToPointNetDevice y WifiNetDevice.

2.8.1.5 Topologías de ayuda

En una red real, se encuentran computadores o host con agregados o integrados NIC. En el mundo simulado de NS3 se encuentran Nodos con NetDevice adjuntos que viene a ser lo mismo que en una red real. En una red simulada grande se tendrá que arreglar muchas conexiones entre nodos, NetDevice y Canales.

Desde la conexión Netdevice a Nodos y NetDevice a Canales, la asignación de direcciones IP, etc., son tareas tan comunes en NS3, que se ofrece lo que se llama topología de ayuda para hacer esto tan fácil como sea posible.

Por ejemplo, se puede tomar muchas operaciones básicas para crear un netdevice, como añadir una dirección MAC, instalar el dispositivo en una red de nodos, configurar pila de protocolos del nodo, o conectar el netdevice a un canal. Incluso se necesitarían más operaciones para conectar múltiples dispositivos en los canales multipunto y luego conectar las redes individuales en Internet.

2.9 Selección del lenguaje de programación.

Después de haber efectuado el análisis de los dos programas de simulación de redes, OMNet++ y NS3, desarrollados bajo el esquema del lenguaje C++. Se ha seleccionado uno de ellos para el desarrollo de este proyecto, que trata sobre un análisis comparativo del desempeño del estándar IEEE 802.11ac respecto al IEEE 802.11n a través de simulación numérica apoyada por software.

En conclusión se ha optado por usar OMNet++; por la factibilidad de los procesos y su interfaz gráfica que es amigable con el usuario ya que es similar a la de Visual Basic. Otro factor para el uso de esta plataforma es la similitud de programación con Java y C++.

Tomando en cuenta el caso de aplicabilidad OMNet++ puede trabajar en relación a otros software como INet++ y SUMO, que son plataformas de simulación de redes y eventos aleatorios con esquemas de estadística como redes telefónicas, viales, neurales y otras redes en las que se requiere un tratamiento matemático y estadístico más exacto.

CAPÍTULO 3

SIMULACIÓN DEL DESEMPEÑO

3.1 Esquema de la simulación

El esquema de simulación es la parte medular de este proyecto, define los parámetros que se deben usar para realizar el desarrollo de la simulación de las redes 802.11n y 802.11ac.

Los esquemas de simulación son los campos virtuales donde se ejecutan las acciones de transmisión y recepción de las redes 802.11n y 802.11ac, con el fin de realizar una comparación de datos para definir la eficiencia de los estándares usados.

Antes de pasar a efectuar una simulación, se debe crear un ambiente similar a los que usualmente tendrán las redes bajo los estándares que se estudian. Estos ambientes en los simuladores virtuales son ideales, lo que quiere decir que no existe ruido, ni Jitter, interferencias y otros problemas más que se tienen en los sistemas reales. Tomando esto en cuenta para fines de este proyecto se definirá dos tipos de ambientes para la simulación, la primera ideal y la segunda real.

La primera forma de simular es la ideal, está relacionada con el estudio de las redes 802.11n y 802.11ac pero en un entorno directo donde los parámetros negativos que puedan influir en la transmisión no existen. La razón por la que se realiza la simulación ideal es porque, así, se puede medir de manera óptima la transmisión en los estándares deseados, en este caso el 802.11n y 802.11ac.

La segunda forma trata de acercarse por medio de parámetros programados a una red real. Este tipo de simulación permite que se observe los datos de la red con parámetros de transmisión y recepción más cercanos a la realidad. Esta forma de simulación permitirá que se haga un análisis más real de los datos estudiados en las redes 802.11n y 802.11ac.

En el caso de la técnica de modulación OFDM para 802.11ac se ha proporcionado un rendimiento a velocidades gigabit. Utilizando los conceptos de la tecnología desarrollada con 802.11n, aumenta la velocidad máxima de datos de 600 Mbps a 1300 Mbps y eventualmente a 6.93 Gbps (Mis libros de Networking, 2009).

3.2 Caracterización de redes ideales

Las redes ideales permiten que tanto los dispositivos como los instrumentos actúen de manera ideal, por lo que las mediciones de parámetros son ideales o carentes de problemas relacionados, como distorsiones, pérdidas e interferencias. Dentro del diseño de redes ideales 802.11 se deben tomar en cuenta ciertos parámetros como ideales, por ejemplo, la distancia, la cobertura, la potencia.

Hablando con relación a la distancia de un enlace 802.11, este parámetro es primordial ya que sin importar el tipo de enlace sea este indoor u outdoor, si la distancia es muy extensa el enlace entre estaciones puede ser nulo. En el caso ideal, el estudio puede realizarse tomando en cuenta una distancia infinita, esto permitirá que a todo momento las estaciones, tanto transmisoras como receptoras se puedan enlazar sin problemas de interrupciones.

Otro parámetro indiscutible es la cobertura del AP, como se analizó anteriormente la cobertura depende del tipo de antena que se use en el enlace. Si se trata de una antena sectorial el enlace solo será para cierta zona de cobertura, mientras que si es omnidireccional el enlace será dentro de 360° de cobertura. En el caso de una red ideal la cobertura del enlace debe ser de 360°, por lo tanto la antena siempre será una omnidireccional, esto para fines de este estudio.

Finalmente la potencia del enlace es un factor importante en la transmisión, ya que para que exista un enlace entre estaciones la potencia del transmisor y el receptor debe ser similar, caso contrario la comunicación sería nula. Por ejemplo, si el transmisor tiene una potencia de 600mw y el receptor una potencia de 100 mw, con una distancia de 10 km. el receptor puede ver y recibir señales del transmisor, pero en cambio el transmisor no puede detectar ningún dato del receptor, debido a la distancia que existe entre ellos y la potencia tan reducida del receptor. La

comunicación solo sería half dúplex, más de lo que se trata es tener una transmisión full dúplex.

Con este criterio tomado en cuenta, se asume que en el caso ideal la potencia debe ser infinita tanto para el transmisor y como el receptor, con el fin de tener comunicación full duplex.

3.2.1 Redes Ideales IEEE 802.11n

Como hasta este momento es evidente para el caso de este estudio se realiza el trabajo con una red ideal con el fin de analizar los principales parámetros en máxima disposición. Esta red ideal 802.11n, por el tema principal debe ser outdoor. Tomando en cuenta esta disposición la red debe cumplir con algunos parámetros además de potencia, direccionalidad, distancia, y línea de vista. Este último es importante en una red ideal ya que a pesar de usarse una red 802.11n, donde el principal parámetro es la tecnología MIMO, se puede potencializar los parámetros a medirse con línea de vista explícita.

Para el diseño de una red 802.11n se debe tomar en cuenta algunos parámetros como:

- La cantidad de datos que se reciben en un segundo. En el caso ideal del estándar 802.11n hasta 600 Mbps.
- El ancho de banda del canal de transmisión 20 MHz y 40 Mhz.
- El tipo de modulación que se usa OFDM.
- Antenas MIMO.

Al tomar en cuenta estos parámetros se espera que en una red ideal los valores ingresados por la red simulada sean cercanos o iguales a los datos expuestos. Por esta razón usando el software de simulación OMNet++ se diseña una red con el estándar 802.11n.

3.2.2 Redes Ideales IEEE 802.11ac

Algo parecido al caso de 802.11n es el 802.11ac, ya que está por demás mencionar que este estándar es una mejora del mismo 802.11n, claro con la implementación de algunos aspectos que ya anteriormente se han estudiado.

En la norma 802.11ac se usa la tecnología mejorada de MIMO que es MU-MIMO. La línea de vista es un factor importante, pero con la tecnología MU-MIMO, por medio de los rebotes se puede recuperar la señal en el receptor, mas como se trata de efectivizar los parámetros en la recepción se debe tener línea de vista, adicionalmente a los parámetros como potencia, distancia y cobertura.

Para el caso de las redes 802.11ac los parámetros que se debe alcanzar en la simulación son:

- Cantidad de datos que se reciben en un segundo que es 1,3 Gbps.
- Ancho de banda del canal que es 40 MHz y 80 MHz.
- Modulación OFDM.
- Antenas MU-MIMO

3.3 Caracterización de redes reales

Para efectuar el desarrollo del rendimiento de 802.11ac con relación a 802.11n, es necesario realizar mediciones en locaciones reales, como es el caso de rebotes para efectuar la recuperación MIMO. En la realidad una red 802.11ac u 802.11n deben usar sus características de transmisión, como OFDM y MIMO para garantizar que son efectivas ante los distintos espacios físicos a los que son sometidos. Por esta razón se requiere que para la simulación se diseñen maneras de arreglar la actividad de MIMO o MU-MIMO.

3.3.1 Redes Reales IEEE 802.11n

Después de analizar las redes 802.11 ideales, con las que se han realizado la revisión de parámetros ideales como velocidad, ancho de banda, técnicas de modulación y transmisión, se procede a analizar las redes 802.11n de forma real donde se puede

simular situaciones cercanas a lo real como distorsiones, interferencias, retardos, BER y Jitter, además de otros parámetros negativos que pueden causar la pérdida de la señal y el comportamiento general de la red.

El estudio de las redes reales permitirá que se analice el comportamiento de los parámetros antes mencionados frente a diversos factores como señales externas de la misma frecuencia, sobre posición de la misma señal, desvanecimiento, difracción o rebotes.

Para realizar el análisis real de una red 802.11n, por medio del software de simulación OMNet++, se debe diseñar objetos o clases que simulen los factores de influencias negativas mencionadas anteriormente, con el fin de comparar los datos de forma ideal con los reflejados de forma real. Al realizar esta comparación se podrá definir el desempeño de la red frente a parámetros negativos y definir la mejor opción de trabajo, como distancias, enfoque de las antenas, potencias de transmisión y otros parámetros importantes que define el uso de una red 802.11n

Para realizar la simulación de una red real se debe tomar en cuenta ciertos parámetros como los que se citan a continuación de manera real.

La distancia, está relacionada con la potencia del transmisor y receptor por lo tanto deben correlacionarse. Es decir a mayor distancia mayor potencia, tanto en el transmisor como en el receptor. Para el caso real la distancia máxima que se puede usar para fines de estudio es 1 km o menos. La razón para usar esta distancia es simplemente por la facilidad al efectuar cálculos de tramas y datos recibidos y enviados.

De igual forma la potencia que se estima usar para el caso real es 450mW o 600mW para el transmisor y 250mw en el receptor, ya que con estas potencias se garantiza una comunicación entre AP y estación.

Con relación a la cobertura se usara antenas omnidireccionales para tratar de tener una cobertura de 360°.

Por lo demás, en el caso de 802.11n se estima que las velocidades alcancen valores de 150 o 145 Mbps, en un ancho de banda de 20Mhz y máximo hasta 300 Mbps en un ancho de banda de 40 Mhz.

3.3.2 Redes Reales IEEE 802.11ac

Como 802.11ac es el sucesor de 802.11n se deben tomar en cuenta los mismos parámetros en la parte real, de tal forma que para fines de estudio no exista discrepancias sobre los cálculos de tramas o bits, por tener diferentes valores de alcance, potencia y antenas. Los parámetros que se usarán son:

La distancia máxima que se usará para fines de estudio es 1 km o menos.

La potencia que se estima usar para el caso real es 450mW o 600mW para el transmisor y 250mW en el receptor, ya que con estas potencias se garantiza una comunicación entre AP y estación.

La cobertura al igual que en 802.11n se usará antenas omnidireccionales para tratar de tener una cobertura de 360°.

Por lo demás, en el caso de 802.11ac se estima que las velocidades alcancen valores de 450 Mbps, en un ancho de banda de 40Mhz y máximo hasta 900 Mbps en un ancho de banda de 80 Mhz, este último solo para fines de simulación ya que en la realidad el ancho de banda de 80 MHz, todavía no se lo usa.

De lo expuesto anteriormente se puede resumir que los parámetros de importancia que intervienen en la comparación de redes 802.11n y 802.11ac, son los que justamente se citan en la tabla 6 los parámetros de redes ideales y en la tabla 7 los parámetros reales.

Con estos datos se podrá realizar la simulación que permita el análisis comparativo entre los dos estándares.

Tabla 6

Parámetros de redes ideales

PARÁMETROS DE RED	802.11n	802.11ac
Modulación	OFDM	OFDM
Tecnología	MIMO	MU-MIMO
Frecuencia de trabajo	5,8 GHz	5,8 GHz
Ranura de tiempo (slotTime)	9 us	20 us
Velocidad de transmisión	600Mbps(ideal)	1300Mbps(ideal)
Ventana	20 y 40 MHz	40 y 80 MHz
Número de índice OFDM	1-4	1-8
Potencia	infinita	infinita
Distancia	infinita	infinita

Nota. Mbps=Megabits por segundo.

Elaborado por: Danny Ortiz

Tabla 7

Parámetros de redes reales

PARÁMETROS DE RED	802.11n	802.11ac
Modulación	OFDM	OFDM
Tecnología	MIMO	MU-MIMO
Frecuencia de trabajo	5,8 GHz	5,8 GHz
Ranura de tiempo (slotTime)	9 us	20 us
Velocidad de transmisión	300Mbps	600Mbps
Ventana	20 y 40 MHz	40 y 80 MHz
Número de índice OFDM	1-4	1-8
Potencia	450mW	450mW
Distancia	600m	600m

Nota. mW=mili watios, GHz=Giga Hertz.

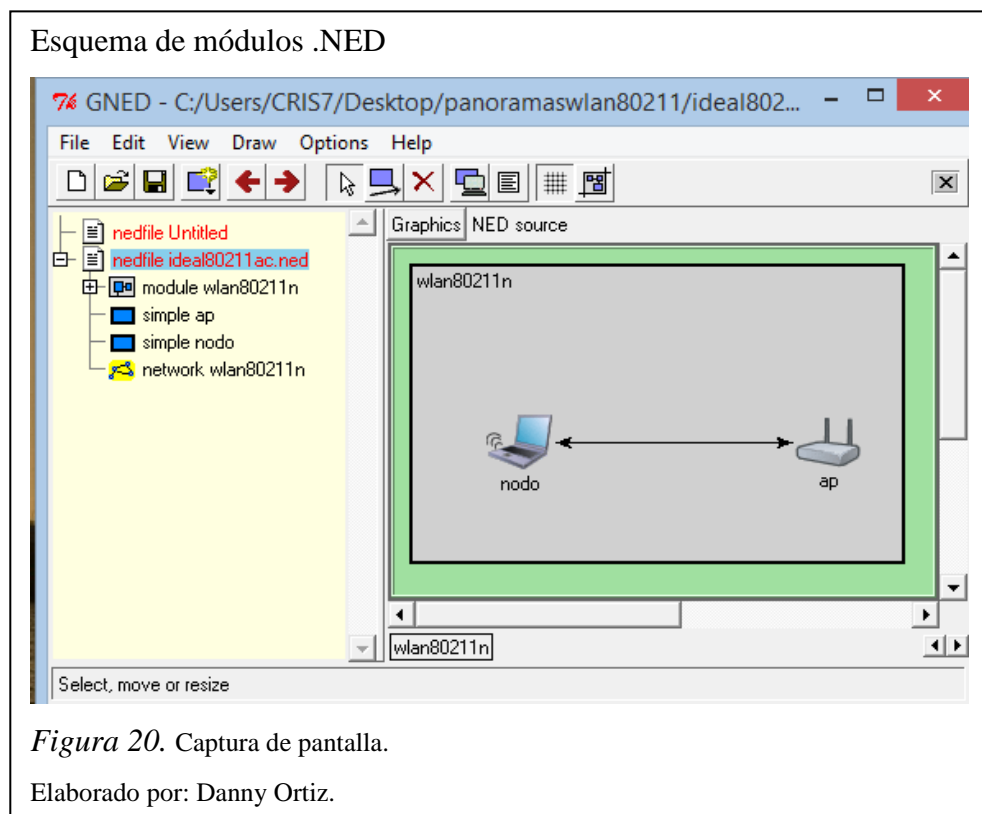
Elaborado por: Danny Ortiz

3.4 Entorno de simulación

Para implementar una red en OMNet++, se debe seguir los pasos mencionados en el capítulo II, que son los procedimientos básicos y esenciales para realizar una simulación. Desde esta parte del desarrollo escrito del “Proyecto de análisis de rendimiento de redes 802.11ac con relación a 802.11n”, en adelante, se centrará en describir paso a paso los archivos de simulación de las redes tanto ideales como reales.

Para iniciar el desarrollo de la simulación se sigue un orden, el mismo que a continuación se cita el proceso de programación para efectuar una simulación en OMNet++.

1. Desarrollar la interfaz gráfica en la herramienta GNED. Con esta interfaz se pretende que el usuario pueda visualizar los objetos que intervienen en la red a simular, en este caso el equipo Wireless bajo el estándar deseado y un Access Point (AP) bajo la misma norma, como se observa en la figura 20. El archivo tendrá una terminación .ned.



2. El segundo paso es programar el diseño de un archivo .cpp, bajo el lenguaje de C++, con el objetivo de enlazar los objetos de la interfaz gráfica y poder ejecutarlos para que funcionen de la manera correcta con el fin de que funcionen como en la vida real.
3. En tercer lugar se debe generar un archivo de compilación con el fin de encadenar los archivos .NED y los .cpp.
4. Finalmente se debe desarrollar un archivo OMNetpp.ini, el que sirve de archivo inicio para que la simulación funcione.

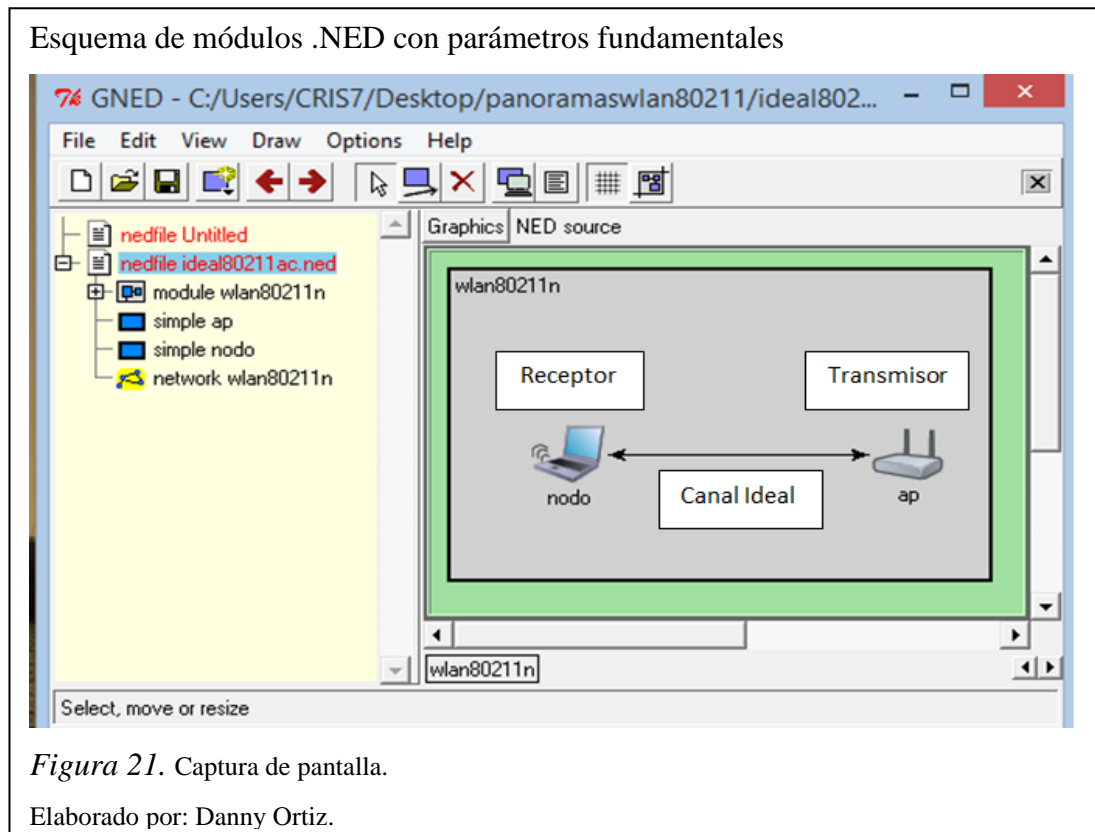
Para efectuar la simulación de redes se debe efectuar una diferenciación entre las redes que se simularán. Las primeras son redes ideales en los dos protocolos y las segundas redes reales, en las que se deben resaltar los aspectos más importantes de los esquemas 802.11, como BER, jitter y retrasos. Al desarrollar las redes tanto ideales como reales es necesario y estricto cumplir con los pasos antes mencionados.

3.4.1 Esquema de desarrollo de redes ideales

Para el desarrollo de las redes de simulación ideal, se debe tomar en cuenta que por su funcionamiento ideal se caracterizan por tener básicamente tres componentes.

- El nodo o dispositivo Wireless en cualquiera de los dos estándares en 802.11ac u 802.11n.
- Un dispositivo Transmisor, Access Point o AP, bajo el mismo estándar a los nodos.
- El canal de transmisión ideal es libre de interferencias, desvanecimiento, BER = 1, ósea ideal, y jitter. Se puede asegurar esto pues se dispone de un sistema simulador que permite asumir que el espacio libre es el vacío, por lo que se define que no existe pérdidas. Además como se mencionó en el Capítulo II, en el sistema de simulación se asume parámetros ideales como potencia de transmisión infinita, distancia de alcance infinita y el tipo de antena es omnidireccional, entonces así se asegura de evitar que se pierdan datos, por falta de alcance o posición espacial.

En la figura 21 se demuestra en OMNet++ cómo será la interfaz de simulación, donde constan los tres parámetros fundamentales.



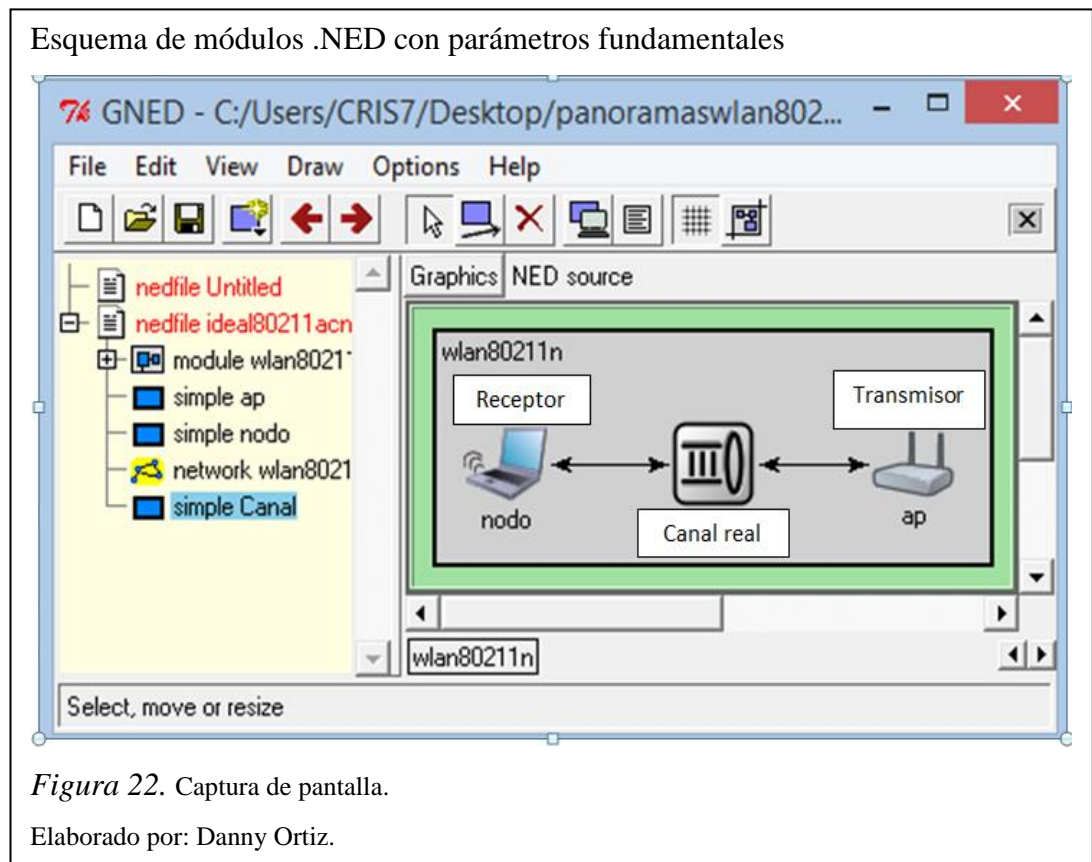
3.4.2 Esquema de desarrollo de redes reales

De forma similar de lo que sucede en las redes ideales, se tiene tres parámetros fundamentales, pero la característica principal de este tipo de redes es que el canal es distinto al de las redes ideales, pues se debe simular parámetros como desvanecimientos, rebotes, errores de recepción entre otros. Los tres principales componentes son:

- El dispositivo Wireless o receptor, bajo cualquiera de los dos protocolos 802.11ac u 802.11n.
- El dispositivo transmisor, Access Point o AP, bajo el mismo estándar a los nodos receptores. Este dispositivo debe enviar ráfagas de tramas de tipo OFDM, para cumplir con el proceso de modulación e implementar MIMO, enviado por dos canales o más.

- El canal de transmisión, pero como se mencionó antes este canal debe ser real. Este canal se distingue por tener colisiones y rebotes de forma aleatoria como sucedería en la vida real, los rebotes y colisiones de las tramas generarían jitter y/o BER real. Entonces el canal debe internamente por medio de algoritmos matemáticos generar sus propias colisiones e interferencias.

En la figura 22 se muestra la interfaz de simulación, donde constan los tres parámetros fundamentales, especialmente el canal real.

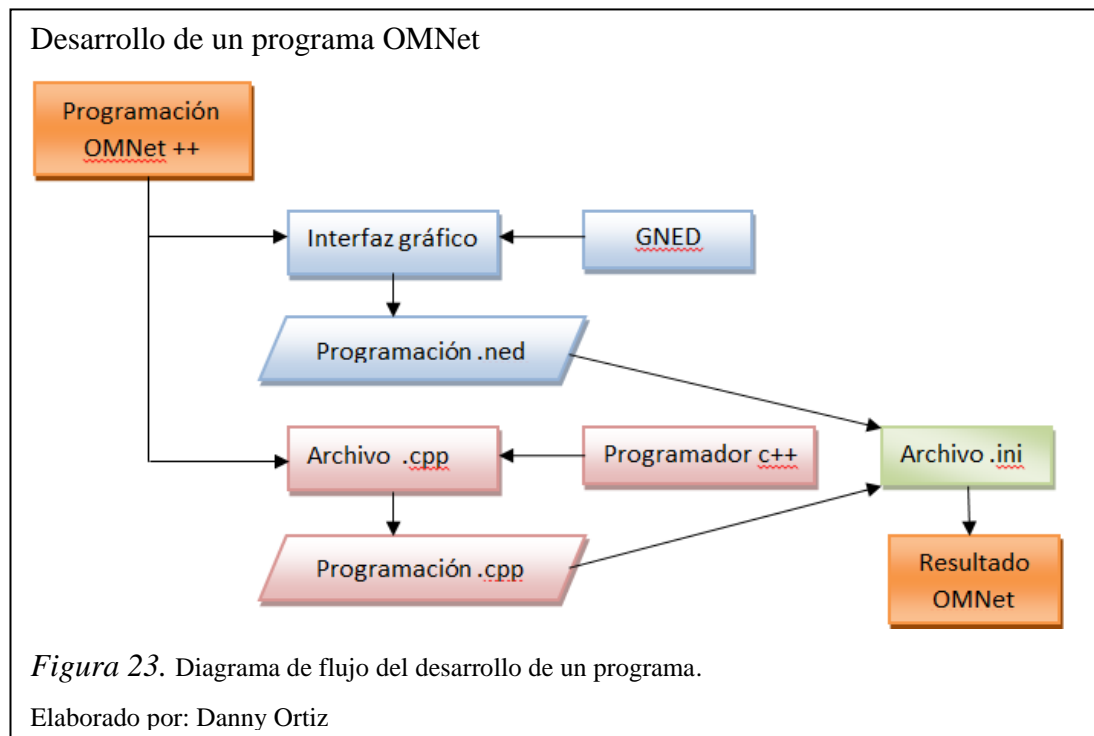


3.5 Red de simulación con el estándar IEEE 802.11n ideal


Con el fin de iniciar el desarrollo este proyecto se realiza un esquema del procedimiento a seguir para la realización de la simulación por medio de un flujograma de trabajo.


El objeto de mostrar por medio del flujograma el procedimiento es permitir al usuario lector observar el proceso de forma sencilla y clara, así como es la secuencia de procesos lógicos y digitales que se debe seguir para efectuar la simulación.

El desarrollo de un esquema de simulación en OMNet ++, requiere que se realice el debido procedimiento, expuesto en el diagrama de flujo de la figura 23.

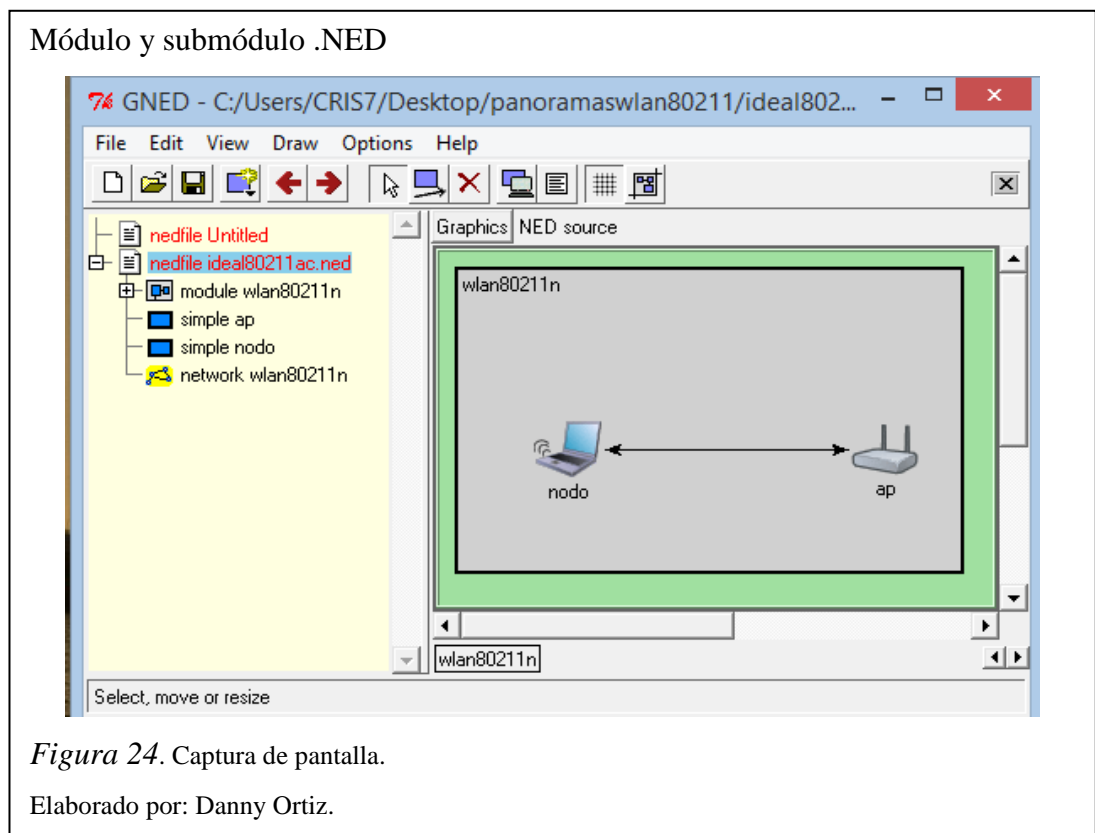


Como se mencionó anteriormente para iniciar el desarrollo de un programa de OMNet, se requiere implementar la interfaz gráfica por medio de la herramienta GNED.

En primer lugar se abre por medio de GNED, un nuevo archivo en New File NED, al presionar el icono . Al presionar New File aparece por defecto un módulo de trabajo, donde se ejecutarán los procesos de programación.

Después presionar el icono  (Add new component), que permite que se añada un nuevo componente o bloque de trabajo. Cumpliendo lo expuesto en el capítulo II. Add new component. Enseguida generados los submódulos se les proporciona los valores gráficos que deben tener como la gráfica de una laptop y/o un AP.

Finalmente luego de generar los submódulos, se crean conexiones o el canal, para el caso de la red ideal, el canal es directo, como se muestra en la figura 24.



Siguiendo el proceso de programación se debe crear un archivo con la programación bajo lenguaje C, en la pestaña NED source.

3.5.1 Programación del nodo

El primer archivo en generar es el módulo de Nodo o Laptop, que es un módulo simple y a este se le añade los valores de programación como los parámetros de distinción de la red, necesarios en lenguaje C++.

Programación del Nodo:

El primer paso para configurar el Nodo o Receptor se debe desarrollar el módulo de un interfaz gráfico. Para explicar el desarrollo del módulo se inicia la programación declarando las cabeceras del módulo, con las sentencias

```

simple nodo                //Iniciación del módulo simple.
.....                  //Programa.
endsimple                 //Fin del módulo.

```

El siguiente paso es declarar las variables que se van a utilizar en la programación de la interfaz gráfica, usando la directiva “parameters”.

```

parameters:              //parámetros del módulo
.....                  //parámetros

```

Finalmente, se termina la programación del módulo realizando las conexiones de información de entrada y salida. Estas conexiones se las declara con los comandos gates y con formato:

```

gates:                   //compuertas de conexión con otros módulos
.....                  //compuestas de entrada y salidas

```

Ahora se pasa a describir los parámetros que intervienen en la programación del módulo AP, dentro de la interfaz de GNED, como:

- **txRate**; velocidad de transmisión, propia de cada estándar, para este caso es como ideal 600 Mbps, ya que se trata de 802.11n.
- **radioDelay**: retardo de propagación, permite tener un valor de comparación entre paquetes y está relacionado con el intervalo de guarda.
- **pkLenBits**: longitud del paquete de bits.
- **igTime**: tiempo de separación de paquetes, cada paquete debe estar separado de acuerdo a cada estándar.
- **numnodos**: número de nodos a trabajar, está determinado por la cantidad aleatoria de usuarios de la red, lo que generará las colisiones de paquetes, esto permite simular los eventos de rebotes y revisar el jitter.

- **ventana:** ancho de banda que se usa de acuerdo al tipo de estándar, en este caso 802.11n entonces el ancho de banda es de 20 o 40 MHz.
- **slotTime:** ranura de tiempo de backoff o de memoria.
- **Gates:** in u out, son las compuertas de enlace o comunicación entre el nodo y el ap.

3.5.2 Transmisor o AP

El esquema de programación para el módulo simple de interfaz gráfica para el módulo AP es similar a la programación del nodo. Excepto en lo que respecta a los parámetros y funcionalidad.

Programación del punto de acceso (AP):

Como primer paso es declarar el inicio y fin del módulo con las sentencias:

```
simple AP                //Iniciación del módulo simple
.....                //Programa
endsimple              //Fin del módulo
```

El siguiente paso es declarar las variables o parámetros que se usarán en el módulo y la programación de la interfaz gráfica, usando la directiva “parameters”.

```
parameters:            //parámetros del módulo
.....                //parámetros
```

Para finalizar la programación del módulo se usa las órdenes de conexiones de entrada y salida. Estas conexiones se las declara con los comandos gates y con el siguiente formato:

```
gates:                 //compuertas de conexión con otros módulos
.....                //compuertas de entradas y salidas
```

Los parámetros mencionados en el programa del AP a desarrollar, son:

- **txRate**: velocidad de transmisión, propia de cada estándar, para este caso es como ideal 600 Mbps, ya que se trata de 802.11n.
- **radioDelay**: retardo de propagación, permite tener un valor de comparación entre paquetes y está relacionado con el intervalo de guarda.
- **pkLenBits**: longitud del paquete de bits
- **igTime**: tiempo de separación de paquetes, cada paquete debe estar separado de acuerdo a cada estándar.
- **ofdm**: número de nodos tramas por slot a trabajar, determina la cantidad de datos que contiene un slot de tiempo según sea los bits determinados en la simulación, este es un número aleatorio entre 0 y 8g.
- **ventana**: ancho de banda que se usa de acuerdo al tipo de estándar, en este caso 802.11n entonces el ancho de banda es de 20 o 40 MHz.
- **slotTime**: ranura de tiempo de backoff o de memoria, almacena una cantidad determinada de datos del paquete original.
- **Gates**: in y out, son las compuertas de enlace o comunicación entre el nodo y el ap.

Como paso final para el desarrollo de la interfaz de la red 802.11n se debe desarrollar el canal y como se ha mencionado antes el canal es ideal, lo que en realidad será solo una conexión simple de unión entre los dos dispositivos, el nodo Wireless y el AP.

3.5.3 Módulo Wireless

Después del desarrollo de los módulos de 802.11n, ahora se debe programar un módulo general o módulo compuesto, el mismo que contiene los tres módulos

simples. Este módulo contiene tres elementos fundamentales, parámetros, submódulos y conexiones.

Programación del módulo Wireless:

El módulo compuesto Wireless se compone de los dos módulos simples AP y NODO, estos son tratados como submódulos. A continuación se presenta el método de programación de este módulo compuesto.

La programación se empieza con el encabezado y finalización por medio de las órdenes del programa.

```
module wireless          //Inicialización del programa
.....                  //Programa
endmodule                //Finalización del módulo
```

La declaración de variables o parámetros internos del módulo compuesto, se declara con la codificación presentada adelante.

```
Parameters:             //inicio para declarar los parámetros
.....                  //declarar parámetros
```

En este punto del programa se declara los submódulos, que son los módulos simples que se han creado anteriormente, por medio de las sentencias. Además dentro de este espacio se ubica la posición espacial del nodo simple en la interfaz gráfica del módulo compuesto.

```
submodules:             //Inicio de las declaración de módulos simples implicados.
```

```
modulo(n);              //Declaración de los módulos simples.
```

```
parameters:             //Declaración de cada uno de los parámetros.
```

Además de los submódulos existen las compuertas de entrada y salida, las conexiones dentro del módulo compuesto y se las define con el siguiente código.

```
gatesizes:                                //Compuertas
.....
connections:                              //Conexiones
.....
```

Los parámetros usados dentro del módulo compuesto son:

- **txRate:** la velocidad de transmisión.
- **ventana:** el ancho de banda de 20 y/o 40 MHz.
- **p:** la probabilidad de estado bueno a malo, en 802.11n ideal es 1.
- **P:** la probabilidad de malo a bueno, en 802.11n es 1 por ser ideal.
- **ofdm:** la cantidad de datos aleatoria que se genera en el AP.
- **BER:** la relación de datos entre los llegados y los enviados.
- **numnodos:** cantidad de nodos aleatorios entre el AP y el nodo.

Posteriormente se desarrollan los submódulos, AP, Nodo y Canal. Con la generación adicional de que en el submódulo nodo, se realiza un arreglo de vector el cual es numnodos, y su patrón de programación son nodo: nodo [numnodos] el cual es el que representa el número de nodos que posee la red.

3.5.4 Módulo de red

Para terminar se desarrolla la red donde se declaran los módulos creados tanto simples como compuestos. El nombre de la red será wireless, por lo general se usa el módulo compuesto donde ya están incluidos los módulos simples creados inicialmente. En este último paso solo se añaden los datos como la cantidad de nodos, probabilidad de malo a bueno y probabilidad de bueno a malo para las colisiones y de esta forma poder verificar el jitter, además de la cantidad de bits para la modulación OFDM y el ancho de banda.

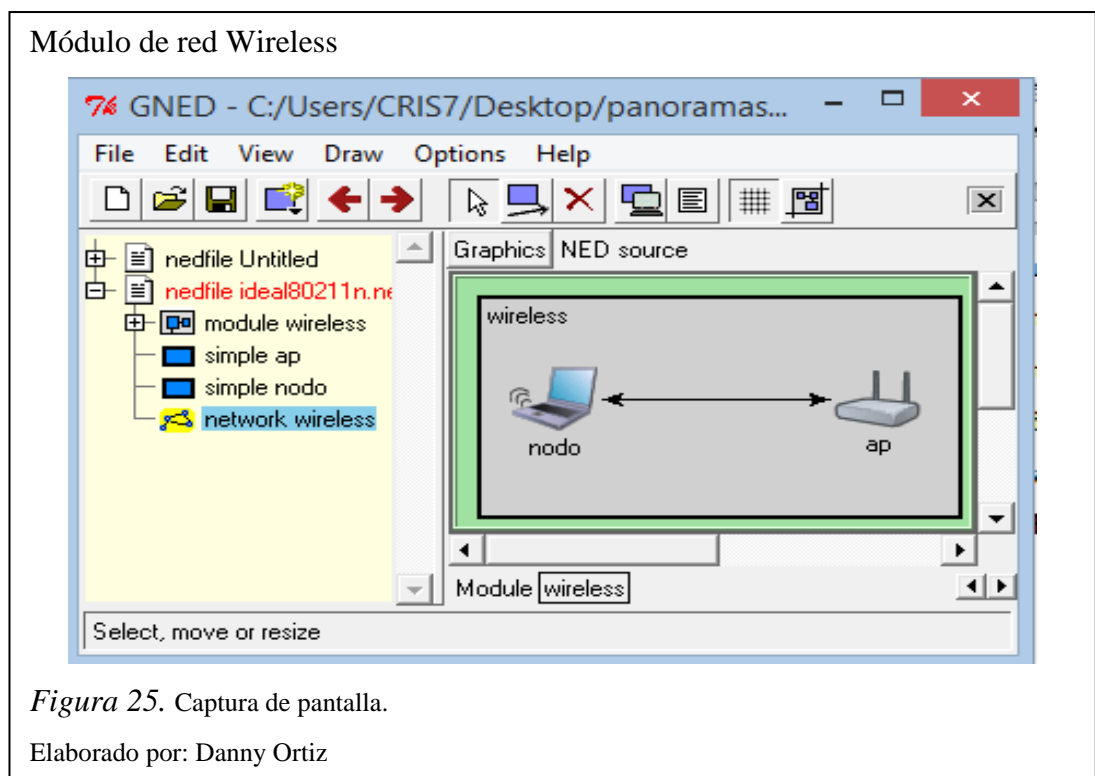
Dentro del módulo Wireless se declara el formato de red, como parte de las normativas de conformación de la interfaz de GNED. Dentro de esta ventana de programación se declara el objeto Parameters, el que contendrá los datos de entrada y/o salida a los usuarios, para interacción con el medio externo.

```
Parameters:          //declaración de los parámetros de red
.....              //parámetros de red
```

Para solicitar la entrada de datos ingresados por el usuario se usa el comando input permitiendo manipular y controlar, los datos que se requieran para lograr el fin esperado.

Los códigos de programación Gned de los módulos de interfaz gráfica mostrada en el proceso anterior se lo adjunta en el Anexo 1.

En caso de esperar datos de salida de la interfaz se usa el comando out, dejando que el usuario permita recibir u observar los datos obtenidos por el programa realizado. Entonces la interfaz debe quedar tal como se muestra en la figura 25.



3.5.5 Programación de módulos en C++

Ahora después de haber efectuado el desarrollo de las interfaces se inicia con el desarrollo de la programación con base en C++. Con el fin de iniciar el desarrollo de la programación C++ se debe seguir una programación secuencial de procesos. En primer lugar se desarrollan módulos de tipo clase en C++.

Las clases son estructuras de programación secuencial que permiten efectuar procesos ordenados y secuenciales con el objetivo de enlazar las interfaces gráficas y permitir el desempeño del programa de orientación visual.

En C++ y para el desarrollo de este programa se define cuatro clases (OMNet++, 2013):

1. **cSimpleModule**; esta clase es la de módulo simple. Dentro de esta clase se definen las funciones:
 - void initialize(), es la función que permite iniciar un proceso de clase en C++.
 - void handleMessage(cMessage *msg); es la función que permite dar acceso a datos de entrada y salida, por medio de mensajes.
 - void activity(); la función define el tipo de actividad de la clase y es similar a la anterior, pues también permite trabajar con mensajes o datos de entrada y salida.
 - void finish(); esta función termina un proceso de clase.
2. **cModule**; es la base común para la programación de cSimpleModule y cCompoudModule. Proporciona compuertas, parámetros y mapeo del generador de números aleatorios (RGN Random Number Generator), mostrar cadenas y un conjunto de métodos virtuales.
3. **cDefaultList**; esta clase usa como base la clase cModule y actúa como un software propietario. No subclasifica sus propias clases.

4. **cObject**; clase base para casi todas las clases en las librerías de OMNet++.

Las principales áreas cubiertas por cObject son:

- Almacenamiento de nombres de cadena a través de Name() y setName().
- Proporciona un mecanismo para recorrer recursivamente todos los objetos de simulación. Este mecanismo constituye el fundamento TKEnv, la simulación GUI.
- Administración de la propiedad, de protección contra errores de programación comunes.

3.5.6 Implementación de la clase módulo Nodo.

En primer lugar se desarrolla un archivo nodo.cpp. Donde se determina dos funciones:

- virtual void initialize()
- virtual void handleMessage(cMessage *msg)

Dentro de este primer módulo se debe declarar Define_Modulo(nodo), el que permite acoplar al archivo .cpp y el módulo de GNED.

Luego de la definición del módulo se procede a iniciar la función void nodo::initialize() que permite iniciar el proceso o la activación de la simulación, enlazando el interfaz gráfico del módulo NODO con la función de trabajo, haciendo que los datos de ingreso por interfaz gráfico sean tratados internamente por la función de iniciación.

Dentro de la función nodo::initialize se declaran las variables internas que se maneja en esta función. La función tiene la estructura que se presenta a continuación.

```
void nodo::initialize()
{  Declaración de variables
  Programa: .....}
```

En la función mencionada se declaran algunas variables pero a continuación se presentan las más importantes, especialmente las que se relacionan con los datos de ingreso de la interfaz gráfica.

- **Back:** recoge un numero aleatorio entre 0 y 1 con la función `dblrand()`; en la variable `z`, a este valor se le multiplica por la ventana.
- **slotTime:** define la ranura de tiempo de 9 us y obtiene en valor de tiempo de backoff del nodo.

Para continuar con el proceso de programación se prosigue con la función `handlemessage`, por medio de la que se trabaja con los mensajes de entrada y salida de las funciones y los módulos ya programados.

Los mensajes son datos que permiten enviar y recibir información que son con los que se debe realizar los algoritmos de funcionamiento del programa.

La función tiene la siguiente estructura:

```
void nodo::handleMessage(cMessage *msg)
{
    Declaración de variables
    Programa: .....}
```

Dentro de la estructura de la función se usan variables las cuales son las siguientes:

Para el primer paquete enviado, el nodo sabe que llega su tiempo de backoff o de memoria, enviando un mensaje de fin de evento; `endTxEvent`.

Más adelante se ubica la función `scheduleAt()`, es la encargada de hacer que el módulo espere el tiempo `Time1` hasta que llegue el mensaje `endTxEvent`.

Después del envío del primer paquete en la función `handleMessege` se genera las siguientes transmisiones.

```
double DIFS=0.00005;
time1=slotTime+DIFS+txt+simTime();
ev.printf("tiempo real de llegada: %4.10f nodo: %d \n",time1,id()-2);
scheduleAt(time1, endTxEvent);
```

Donde DIFS es el tiempo de espera para el reenvío de otro paquete, simTime es una función de acumulación de todo el tiempo del proceso hasta que llegue el mensaje endTxEvent.

La programación C++ de la simulación se la adjunta en el Anexo 2.

Hasta aquí los valores de programación cpp en el nodo.

Para el caso de la programación del canal no se efectúa ninguna acción ya que el canal es ideal, sin solapamiento, interrupciones y BER=1, por lo tanto se describe ahora la programación del AP o Transmisor. La razón para que se asuma estos datos es porque se asume el caso ideal, tal como se expresa en el apartado 3.4.1.

3.5.7 Programación del módulo AP

En el módulo AP se desarrolla un archivo ap.cc y de la misma forma como el módulo simple nodo consta de dos funciones, la initialize() y handleMessage(cMessage *msg). La función initialize permite la activación de la función. En la función handleMessage se debe saber a qué momento llegan los datos y cuando terminan, por lo tanto se debe recoger y enviar datos como mensajes, de esta forma se implementa el mensaje finRxEvent=newcMessage("end-reception").

La función scheduleAt() permite que llegue el mensaje en un tiempo de recepción endReception. Ese tiempo endReception es el tiempo que ha transcurrido desde que se genera el backoff hasta el momento que se recibe el mensaje en el AP, esto incluye el tiempo de pérdidas en el canal pero como el canal es ideal el tiempo tiende a ser igual a cero o no hay retardos, por lo tanto el jitter por desfase y solapamiento es cero. La función simTime() cumple la misma función descrita en el caso del módulo nodo.

La programación de este módulo permite que se enlace la red principal del interfaz gráfico diseñado antes y la programación del archivo cpp, por medio del comando Define_Module(ap), bajo la estructura.

```
Define_Module(ap),
void ap::initialize()
{
    Declaración de variables
    Programa: .....}
void ap::handleMessage(cMessage *msg)
{
    Declaración de variables
    Programa: ..... }
```

Como parte final de esta programación se debe verificar si hay error en las variables por medio de efectuar la ejecución del programa por medio del simulador, eso permite detectar errores en la programación de los mismos.

Después de esta ejecución se procede a realizar la simulación para la detección de los parámetros que se analizarán, los mismos que se muestran en la tabla 8 que se muestra a continuación.

Tabla 8

Parámetros de simulación de red 802.11n ideal

Ranura de tiempo (slotTime)	9us
Velocidad de transmisión 802.11n	150Mbps, 300Mbps, 600Mbps
DIFS	100us
Ventana	20 y 40 MHz
Número de nodos	2-10
Número de índice OFDM	1-4

Nota. us=micro segundos.

Elaborado por: Danny Ortiz

3.6 Red de simulación con estándar IEEE 802.11ac ideal

En el caso de la red 802.11ac se cumple todo igual que 802.11n, pero cambia los parámetros de red. Entonces existen los mismos módulos AP y nodo, además de las funciones de inicio y final de mensajes. A continuación se muestra en la tabla 9 los parámetros que se utilizarán en la simulación.

Tabla 9

Parámetros de simulación de red 802.11ac ideal

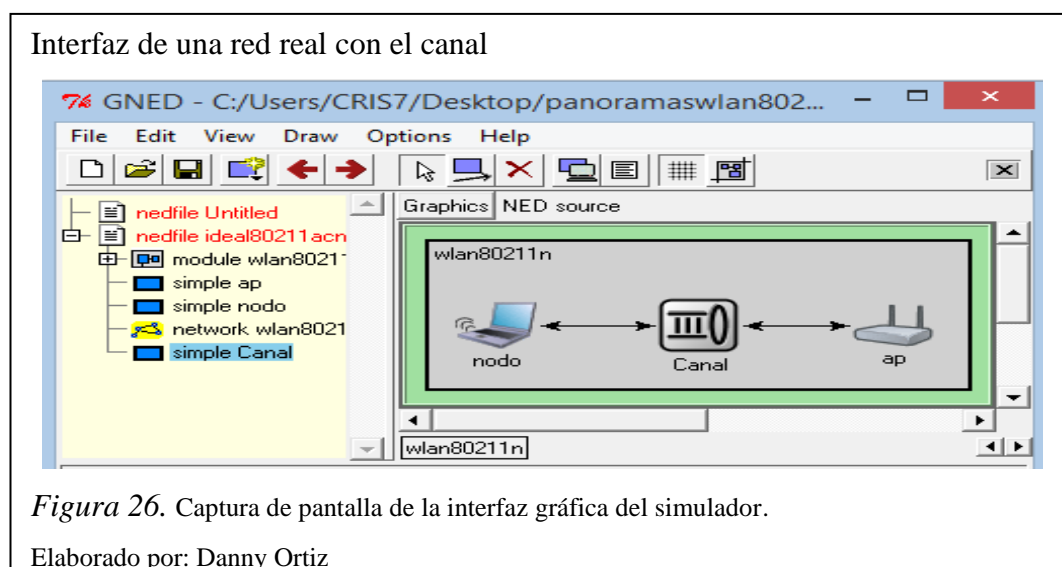
Ranura de tiempo (slotTime)	20us
Velocidad de transmisión	350Mbps, 600Mbps,1300 Mbps
DIFS	100us
Ventana	40 y 80 MHz y 160 MHz
Número de nodos	2-10
Número de índice OFDM	1-8

Nota. us=microsegundos

Elaborado por: Danny Ortiz

3.7 Red de simulación con estándar IEEE 802.11n real

En el diseño de las redes reales 802.11n, la programación es similar a la situación de la red 802.11n ideal, solo que en esta parte se añade un canal con parámetros de interrupciones, colisiones, retardos, jitter y por lo tanto el valor del BER es más real. A este canal se le crea una interfaz gráfica como se muestra en la figura 26.



3.7.1 Módulo Canal

Para el desarrollo de la interfaz se procede a ubicar los siguientes valores dentro del dispositivo de la interfaz del canal, esto por medio de GNED.

Creación del módulo canal se define por medio de los códigos que se presentan a continuación:

```
simple canal                //inicia el módulo
    parameters:            //parámetros de dispositivo
    .....                //declaración de variables
    gates:                 //compuertas de conexión
    .....                //declaración de entradas y salidas
```

3.7.2. Programación C++ del módulo canal

Como siguiente paso se programa el archivo cpp, el que será el enlace de la parte gráfica con la interna del programa y se lo hace por medio de la programación del archivo colisión.

La programación C++ del bloque de canal se muestra a continuación, definiendo el siguiente esquema.

```
#include <string.h>                //definición de librerías del programa
#include <OMNetpp.h>

class colision : public cSimpleModule //definición de la clase del módulo
{
    ..... };                        //declaración de funciones

Define_Module(colision);            //declaración del módulo

void colision::initialize()          //inicia la función
{
    .....                          //programa
}

void colision::activity()            //determina la actividad de la función
{
    .....                          //programa
}

void colision::finish()              //termina la función
{
    ..... }                        //programa
```

Donde aparecen las funciones void activity(), es donde se envía y recibe los mensajes de los dispositivos AP y nodo.

La función void finish() es la función de finalización de este módulo. Como primer paso recibe los mensajes de los dos lados del canal con el siguiente comando.

```
CMessage *msg;  
Msg =receive();
```

Se define las variables que definen las colisiones, el canal OFDM y otros parámetros de funcionalidad en el canal los cuales son:

- Kb estado bueno
- Km estado malo
- Kh error de colisión
- r variable de distribución de error por medio de la función uniform();

Se usará como base un dato de archivo de audio de un valor aleatoriamente elegido de 6384 bits, este archivo es enviado desde el AP, recibido por el canal y reenviado al nodo y de esta forma se tiene este valor para poder comparar los datos enviados y los datos recibidos exactamente.

Si $k_h > k_m$ no se genera el error.

Si $k_h < k_m$ se genera el error

Si los paquetes recibidos en el nodo sobrepasan el valor de 6384 bits, entonces no existen errores, mientras que si son menores a dicho valor si existen errores.

Para el desarrollo de OFDM, se incluye la parte de continuar, donde se recoge el algoritmo de envío de canales varios desde 1 a 8 de forma randómica según se defina en el AP, y finalmente la función finish(), define los valores que se mostrarán al usuario.

La programación del módulo canal se encuentra en el Anexo 3.

Como BER, en base a su formulación definida en el capítulo II se basa en el número de tramas recibidas con relación al número de tramas enviadas se tomará en cuenta estos datos.

Igualmente con el número de tramas colisionadas se puede definir el Jitter tomando en cuenta el número de canales que está definido por OFDM.

Entonces una vez implementado el canal real, se procederá a recoger los datos mencionados anteriormente, para ser analizados más adelante.

Tabla 10

Parámetros de simulación de red 802.11n real

Ranura de tiempo (slotTime)	9us
Velocidad de transmisión 802.11n	150Mbps, 300Mbps, 600Mbps
DIFS	100us
Ventana	20 y 40 MHz
Número de nodos	2-10
Número de índice OFDM	1-4

Nota. Datos para la simulación.

Elaborado por: Danny Ortiz

Tabla 11

Valores de evaluación de la red 802.11n real

Canal OFDM	1-6
Colisiones (P) estado bueno a malo	1, 0.001, 0.000001
Colisiones (p) estado malo a bueno	0.000001, 0.001, 1
BER	Por calcular

Nota. Datos para la simulación.

Elaborado por: Danny Ortiz

3.8 Red de simulación con estándar IEEE 802.11ac real

El diseño de la red 802.11ac es similar a la red 802.11n, la diferencia está en la velocidad del canal o txRate.

Cambian también el número de índice de canal que va desde 1 hasta 8, pues con eso determinará el tipo de modulación.

El BER, dependerá de los resultados por medio de conocer los valores enviados y los recibidos. Por lo demás el canal actúa de forma similar al caso de la red 802.11n.

Tabla 12

Parámetros de simulación de la red 802.11ac real

Ranura de tiempo (slotTime)	20us
Velocidad de transmisión 802.11n	300Mbps, 600Mbps, 1300Mbps
DIFS	150us
Ventana	40 y 80 MHz
Numero de nodos	2-10
Numero de índice OFDM	1-8

Nota. Datos para la simulación.

Elaborado por: Danny Ortiz

Tabla 13

Valores de evaluación de la red 802.11ac real

Canal OFDM	1-8
Colisiones (P) estado bueno a malo	1, 0.001, 0.000001
Colisiones (p) estado malo a bueno	0.000001, 0.001, 1
BER	Por calcular

Nota. Datos para la simulación.

Elaborado por: Danny Ortiz

Después de haber efectuado todo este análisis se procede a realizar la adquisición de datos para realizar un análisis paramétrico de datos, con el fin de efectuar el rendimiento de los parámetros de cada red.

CAPÍTULO 4

ANÁLISIS DE RESULTADOS

4.1 Adquisición de datos en redes ideales

El proceso de adquisición de datos es un evento de suma importancia para determinar la eficiencia de los estándares presentados en la simulación efectuada en OMNet++.

La adquisición de datos se realiza en base a cada red diseñada, las primeras redes para este proceso son las ideales con el formato 802.11n y 802.11ac, en segundo lugar se procede a tomar datos de las redes reales con las normas 802.11n y 802.11ac. Los datos que se tomarán para el proceso son:

- Cantidad de datos enviados.
- Cantidad de datos recibidos.
- Tiempos de retardo en los datos recibidos.

Luego esos datos se los debe comparar bit a bit para verificar que con estos datos se puede calcular el valor del BER, con el que se puede definir la eficacia de la transmisión y por supuesto la factibilidad de implementar una red 802.11n o una red 802.11ac.

A fin de estimar la eficiencia del estándar se debe usar el BER como un esquema cuantitativo de datos. Así, si el valor del BER es más aproximado a 1, el esquema será más eficiente y si es más bajo o más aproximado a cero, el esquema que se usa será menos eficiente.

La relación que se usa para calcular el BER es:

$$BER = \frac{\text{datos perdidos}}{\text{total de datos enviados}}$$

En el caso de la simulación se usa la relación entre los datos perdidos con relación a los datos enviados. De esa forma se puede verificar realmente si los datos enviados desde el transmisor llegan completamente al receptor o no.

En el caso de las simulaciones de redes ideales se estima que el BER siempre tiende a 1, pues la pérdida de datos es casi nula. La razón para que esto suceda es que los datos como se trata de un canal sin interferencias y con la potencia de transmisión y recepción infinita; la distancia entre el RX y TX es un factor estimado también como infinito, entonces el enlace no se ve afectado por pérdidas, pero se puede dar el caso de pérdidas despreciables por la misma programación o por el tiempo de procesamiento del simulador, aun así el valor es totalmente mínimo.

Mientras lo anterior sucede en el caso de las simulaciones ideales, en cambio en las reales se debe tomar en cuenta que si existirán pérdidas, puesto que la red simulada tendrá como parte de su trabajo un canal, mismo que hace las veces de interferencias, rebotes, selección de canal, retrasos, colisiones y todo para acercarlo más a lo real. Esto para saber en realidad cómo funciona el estándar que se usa.

4.1.1 Proceso para ejecutar las simulaciones

Para realizar las simulaciones de las redes ya programadas anteriormente en el capítulo 3 se debe seguir los siguientes pasos:

Primero se inicia el programa OMNet++. Después de haber sido instalado en el lugar deseado por el usuario, en el caso de este proyecto en la carpeta C:/; se procede a buscar la carpeta msys, tal como se observa en la figura 27 y el archivo msys ejecutable de la figura 28.

Carpeta de sistema msys

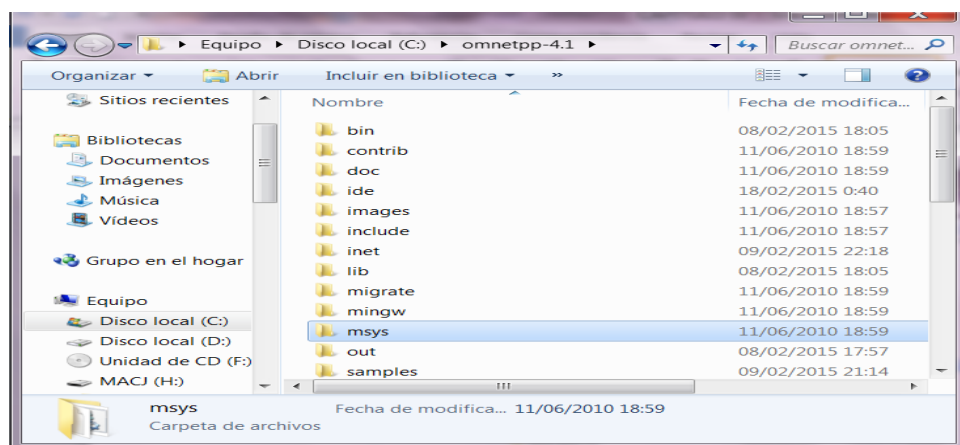


Figura 27. Captura de pantalla de la carpeta de sistema msys.

Elaborado por: Danny Ortiz

Archivo ejecutable msys

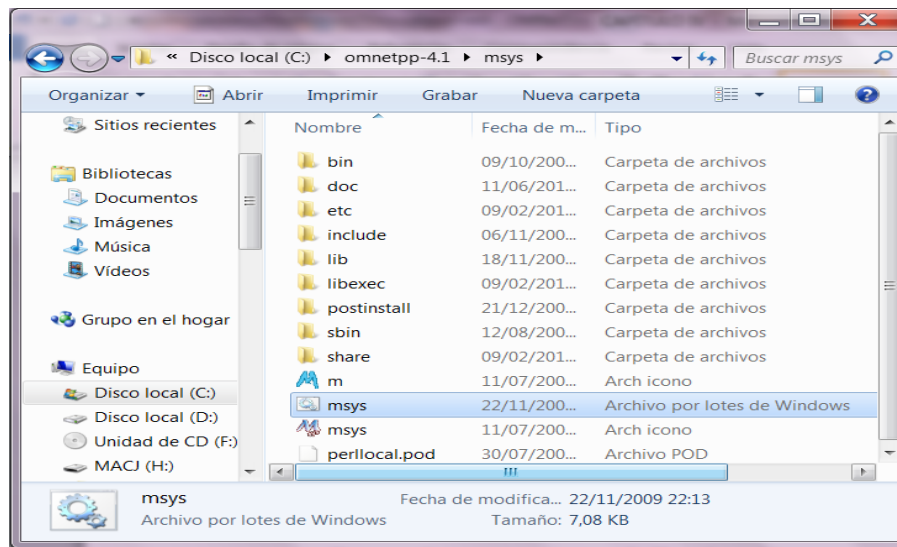


Figura 28. Captura de pantalla de la ubicación del archivo ejecutable msys.

Elaborado por: Danny Ortiz

Dentro de esa carpeta se halla el archivo msys, el que es el archivo ejecutable de mingwenv.cmd. En segundo paso después de haber abierto el programa OMNet++, se presenta la interfaz gráfica de OMNet++, la cual se muestra en la figura 29.

Interfaz de Omnet++

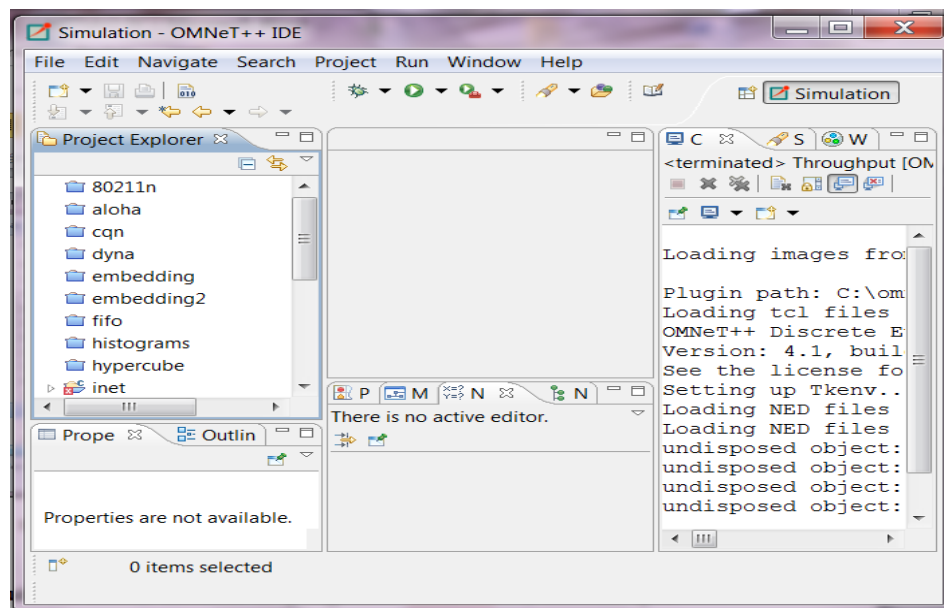
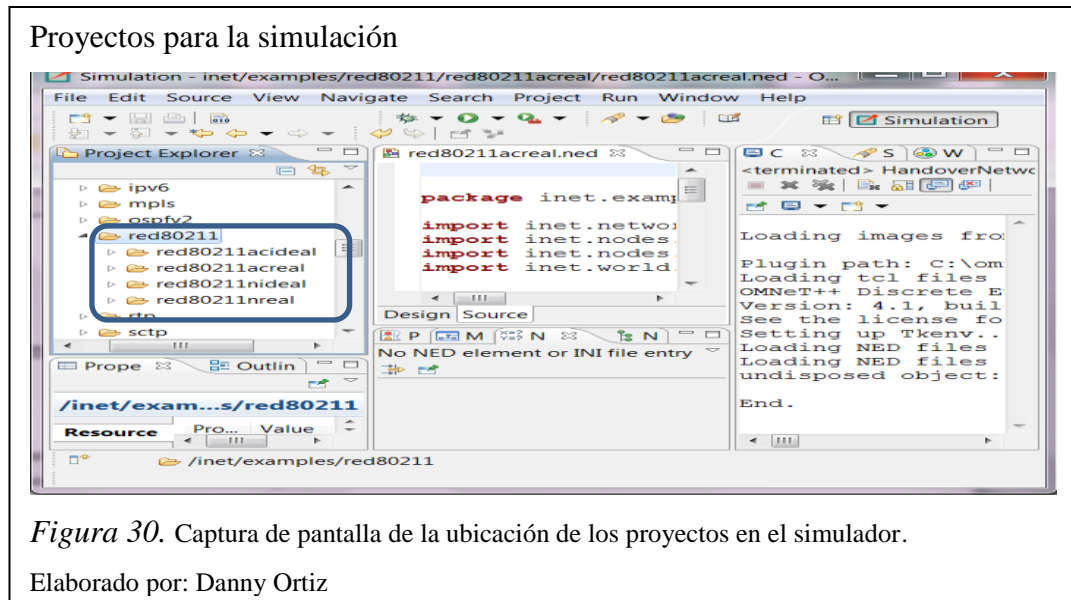


Figura 29. Captura de pantalla de la interfaz gráfica del simulador.

Elaborado por: Danny Ortiz

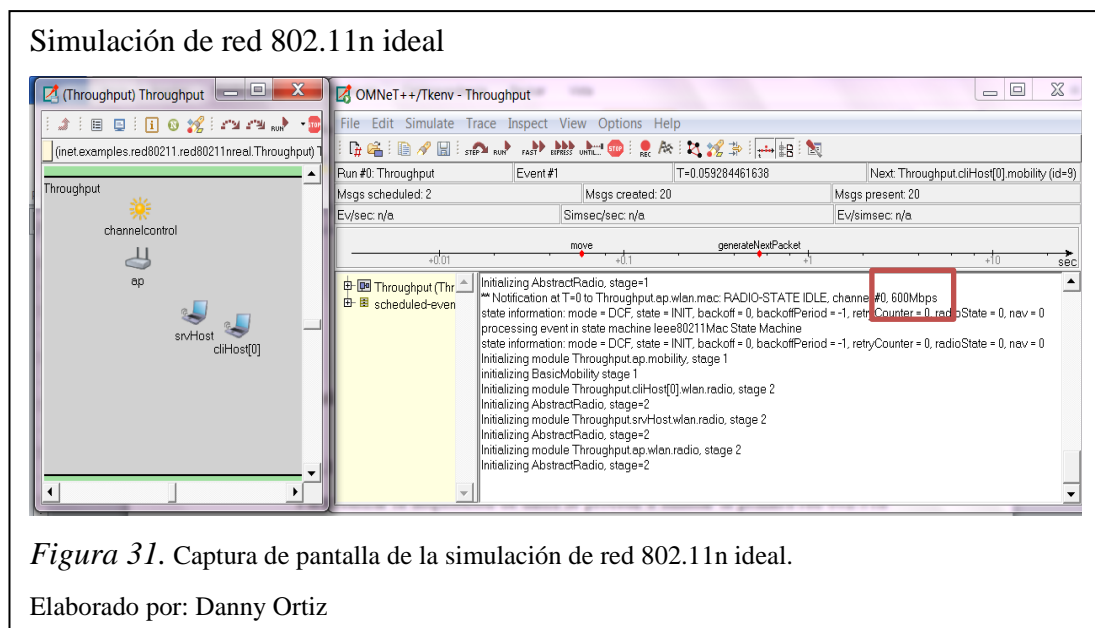
El tercer paso es abrir un nuevo proyecto OMNet++. En el caso de este proyecto se define cuatro proyectos. Dentro de la carpeta red80211, se definen las cuatro redes 80211n ideal, 80211ac ideal, 80211n real y 80211ac real. En la figura 30 se muestra los proyectos creados para el propósito de este trabajo.



Finalmente se abren los archivos .ned, .ini y .cc, para iniciar cada proyecto.

4.1.2 Adquisición de datos de IEEE 802.11n ideal

Para realizar la adquisición de datos se procede a simular la primera red 802.11n ideal. La figura 31 muestra el proceso de adquisición de datos. Para realizar este proceso, en la Gned del proyecto se presiona RUN, para que inicie la simulación.



Número de eventos y resultados de la simulación:

Datos enviados

Scalar *.ap.wlan.mac numRecibidos 0

Scalar *.ap.wlan.mac numAckEnviados 23724

Scalar *.ap.wlan.mac numRetra 0

Scalar *.ap.sink numPaketes 23724

Scalar *.ap.sink numBits 584688000 #velocidad de transmisión

Scalar *.ap.sink throughput 7117447.6025124 #datos enviados

Scalar *.ap.sink packetSec 593.1206335427

Scalar *.ap.wlan.mac numEnviados 0

Scalar *.ap.wlan.mac numEnviadosRetra 0

Datos recibidos

Scalar *.PC.sink numPaketes 23721

Scalar *.PC.sink numBits 584652000 #velocidad de recepción

Scalar *.PC.sink throughput 7116595.2734584 #datos recibidos

Scalar *.PC.sink packetSec 693.04960612153

Scalar *.PC.wlan.mgmt "paketes recibidos por CW" 0

Scalar *.PC.wlan.mgmt "paketes rebotados por CW" 0

De estos datos recolectados se procede a identificar el BER mediante la fórmula:

Total de datos perdidos = datos enviados – datos recibidos.

Dperdidos = 7117447.6025124 – 7116595.2734584 = 63290540

El BER permitido para una red 802.11n es del orden de 10^{-7} , entonces se procede al cálculo del Ber con los datos simulados.

$$BER = \frac{\text{datos perdidos}}{\text{total de datos enviados}}$$

$$BER = \frac{63290540}{71174476025124}$$

$$BER = 8,89231E^{-07}$$

A pesar que el valor del BER en el modo ideal debe ser igual a 1 en este caso presenta un valor muy pequeño ya que también influyen aspectos como la velocidad de procesamiento del simulador sin embargo este valor del BER es despreciable y está dentro de lo establecido por la norma 802.11n.

Para definir el Jitter aleatorio o blanco se debe tomar en cuenta las variaciones de tiempo que existen en las transmisiones, pero como se trata de una red ideal las potencias de transmisión y las distancias son ideales, por cuanto no hay retrasos ni pérdidas, entonces no existe Jitter. Los datos de retardos se presentan a continuación:

Tabla 14

Datos obtenidos de la simulación 802.11n ideal

ENUM	BACKOFF=3	DEFER=1	IDLE=0	RECEIVE=8	WAITACK=4	WAITBROADCAST=5	WAITCTS=6	WAITDIFS=2	WAITPIFS=7
3	1100	18	0	0	0	0	2	0	0
2	1118	18	0	0	0	0	2	0	0
7	1136	18	0	0	0	0	2	0	0
6	1154	18	0	0	0	0	2	0	0

Nota. Datos de retardos.

Elaborado por: Danny Ortiz

Las columnas finales, de color verde, muestran que no tienen ninguna variación por lo tanto no puede haber retrasos, por lo que el Jitter seria nulo.

4.1.3 Simulación IEEE 802.11ac ideal

Para realizar la adquisición de datos se procede a simular la primera red 802.11ac ideal. La figura 32 muestra el proceso de adquisición de datos. Para realizar este proceso, en la Gned del proyecto se presiona RUN, para que inicie la simulación.

Simulación de red 802.11ac ideal

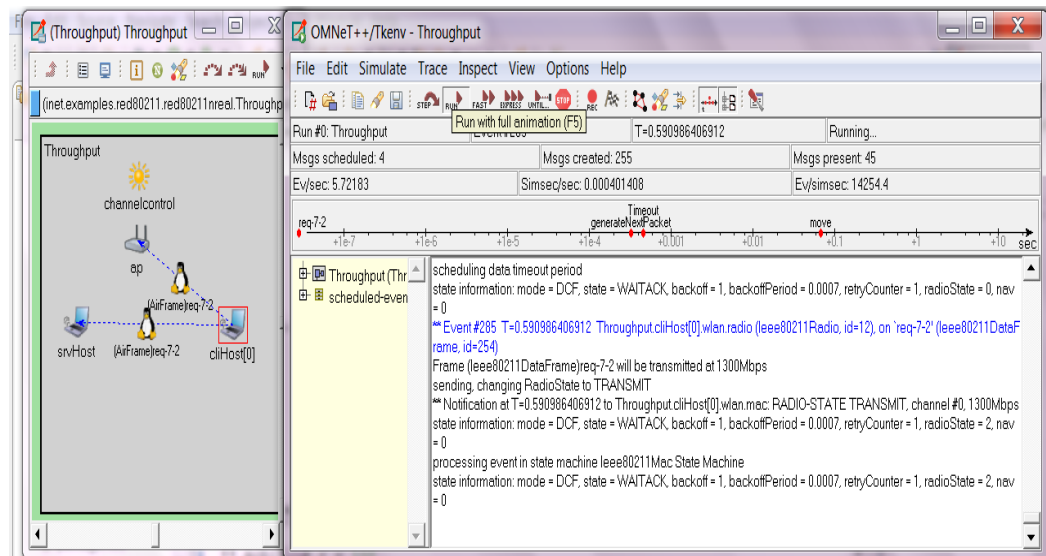


Figura 32. Ejecución de la Simulación de red 802.11ac ideal.

Elaborado por: Danny Ortiz

Número de eventos y resultados de la simulación:

Datos enviados

Scalar *.ap.wlan.mac numRecibidos 0

Scalar *.ap.wlan.mac numAckEnviados 23724

Scalar *.ap.wlan.mac numRetra 0

Scalar *.ap.sink numPaketes 23724

Scalar *.ap.sink numBits 987868000 #velocidad de transmisión

Scalar *.ap.sink throughput 72357576.025124 #datos enviados

Scalar *.ap.sink packeteSec 593.1206335427

Scalar *.ap.wlan.mac numEnviados 0

Scalar *.ap.wlan.mac numEnviadosRetra 0

Dato recibidos

Scalar *.PC.sink numPaketes 23721

Scalar *.PC.sink numBits 984652000 #velocidad de recepción

Scalar *.PC.sink throughput 72357555.273458 #datos recibidos

Scalar *.PC.sink packeteSec 693.04960612153

Scalar *.PC.wlan.mgmt "packetes recibidos por CW" 0

Scalar *.PC.wlan.mgmt "packetes rebotados por CW" 0

De estos datos recolectados se procede a calcular el BER:

Total de datos perdidos= datos enviados – datos recibidos

$$D_{\text{perdidos}} = 72357576.025124 - 72357555.273458 = 20751666$$

El BER permitido para una red 802.11ac es del orden de 10^{-7} . Entonces se procede al cálculo del BER con los datos simulados.

$$BER = \frac{\text{datos perdidos}}{\text{total de datos enviados}}$$

$$BER = \frac{20.751666}{72357576.025124}$$

$$BER = 2,86793E^{-07}$$

Se puede observar que el BER está dentro de lo establecido por la norma 802.11ac, pero menor al BER resultante de la norma 802.11n.

Para definir el Jitter aleatorio o blanco se debe tomar en cuenta las variaciones de tiempo que existen en las transmisiones, pero como se trata de una red ideal las potencias de transmisión y las distancias son ideales, por cuanto no hay retrasos ni pérdidas, entonces no existe Jitter.

Los datos de retardos en los bits de enlace y reconocimiento, se presentan a continuación.

Tabla 15

Datos obtenidos de la simulación 802.11ac ideal

ENUM	BACKOFF=3	DEFER=1	IDLE=0	RECEIVE=8	WAITACK=4	WAITBROADCAST=5	WAITCTS=6	WAITDIFS=2	WAITIFS=7
5	1071	18	0	0	0	0	2	0	0
4	1089	18	0	0	0	0	2	0	0
9	1107	18	0	0	0	0	2	0	0
8	1125	18	0	0	0	0	2	0	0

Nota. Datos de retardos.

Elaborado por: Danny Ortiz

Las columnas finales, de color verde, muestran que no tienen ninguna variación en los bits enviados y recibidos, por lo tanto no existen retrasos, por lo que el Jitter no existe o es nulo.

4.2 Adquisición de datos redes reales

El proceso de adquisición de datos es un evento de suma importancia para determinar la eficiencia de los estándares presentados en la simulación efectuada en OMNet++.

La adquisición de datos se realiza en base a cada red diseñada, las primeras redes para este proceso son las reales con el formato 802.11n y en segundo lugar las que tienen el formato 802.11ac.

Los datos que se tomarán para el proceso son:

- Cantidad de datos enviados
- Cantidad de datos llegados
- Tiempos de retardo en los datos recibidos

Con estos datos se puede calcular el valor del BER, con el que se puede definir la eficacia de la transmisión y por supuesto la factibilidad de implementar una red 802.11n o 802.11ac.

Otra situación que se debe considerar es que el Jitter en las redes reales está definido por varias situaciones entre ellas los rebotes de la señal, la modulación usada, los canales usados para la transmisión, la conversión de datos, etc.

Como no se puede controlar los parámetros citados antes, el Jitter es aleatorio, por lo que lo único que se puede definir es la medición de retardo en las transmisiones de las redes, así se puede notar cuál de las redes es más irregular en los retardos por lo que estos permite definir de forma indirecta el mayor o menor Jitter.

La potencia definida para cada red es constante según por lo que se ha programado esto dentro del archivo .ini de cada red y es de 450mW.

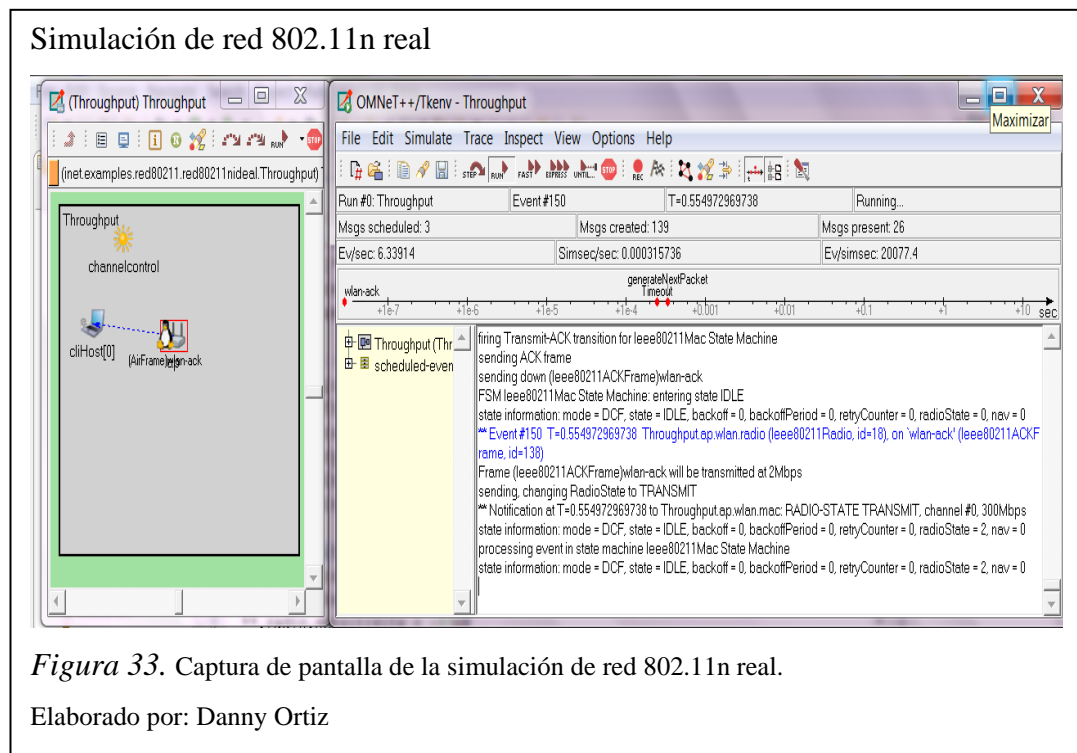
Otro factor que se toma en cuenta es la distancia que como máximo se ha considerado 600 m y 400m para la radiación.

La sensibilidad es máxima entre -85dB y -110dB. Se asume estos datos son los más cercanos a lo real para antenas inteligentes de tipo 802.11n y 802.11ac. Algunos de estos datos se los toma como base de manuales UBIQUITI, MIKROTIK, MERARI y MIMOSA, las antenas más recientes en transmisiones outdoor de tipo 802.11ac

4.2.1 Simulación IEEE 802.11n real

Para recolectar los valores pertinentes a la red real en 802.11n se procede a simular el proyecto red80211n real y se procede a simular el archivo .ned. La figura 33 muestra el proceso de adquisición de datos.

Para realizar este proceso, en la Gned del proyecto se presiona RUN, para que inicie la simulación.



Los datos adquiridos para revisar los valores del BER y el Jitter son los que se presentan a continuación.

Datos enviados

Scalar *.ap.wlan.mac numRecibidos 0

Scalar *.ap.wlan.mac numAckEnviados 23724

Scalar *.ap.wlan.mac numRetra 0

Scalar *.ap.sink numPaketes 23724

Scalar *.ap.sink numBits 276866000 #velocidad de transmisión

Scalar *.ap.sink throughput 67235956.025124 #datos enviados

Scalar *.ap.sink paqueteSec 593.1206335427

Scalar *.ap.wlan.mac numEnviados 0

Scalar *.ap.wlan.mac numEnviadosRetra 0

Dato recibidos

Scalar *.PC.sink numPaketes 23721

Scalar *.PC.sink numBits 269886000 #velocidad de recepción

Scalar *.PC.sink throughput 67235743025124 #datos recibidos

Scalar *.PC.sink paqueteSec 693.04960612153

Scalar *.PC.wlan.mgmt "packetes recibidos por CW" 0

Scalar *.PC.wlan.mgmt "packetes rebotados por CW" 0

De estos datos recolectados se procede a identificar el BER.

Total de datos perdidos= datos enviados – datos recibidos

$$Dt = 67235956025124 - 67235743025124 = 213000000$$

El BER permitido para una red 802.11n es del orden de 10^{-7} , entonces se procede al cálculo del ver con los datos simulados.

$$BER = \frac{\text{datos perdidos}}{\text{total de datos enviados}}$$

$$BER = \frac{213,000000}{67235956025124}$$

$$BER = 3,16795E^{-06}$$

Se puede observar que el BER está dentro de lo establecido por la norma 802.11n, como se nota en el apartado 4.1.1.

A fin de realizar el cálculo del BER se procede tomar en cuenta los valores de retraso en los datos de la simulación esto genera Jitter aleatorio o blanco.

Tabla 16

Datos obtenidos de la simulación 802.11n real

enum	BACKOFF=3	DEFER=1	IDLE=0	RECEIVE=8	WAITACK=4	WAITBROADCAST=5	WAITCTS=6	WAITDIFS=2	WAITIFS=7
1	1232	2475	327	3181	0.562813502304	0.686313502304	100	100	100
0	3707	8513	37	3214	0.550313502304	0.687313502304	340	3044	28354
3	12220	32718	0	3218	0	0.687792543291	1334	1000	1668
2	44938	57266	0	3218	0	0.687792543291	2334	8324	30964
7	102204	32760	0	3224	0	0.687803002009	1336	1000	1666
6	134964	49158	0	3224	0	0.687803002009	2004	2338	16366

Nota. Datos de retardos.

Elaborado por: Danny Ortiz

De los datos marcados con amarillo y verde se aprecian los retrasos de los mismos. Para calcular los retardos en cada caso se usa una simple sustracción de cada valor.

Tabla 17

Datos de retardos en 802.11n real

T0(μs)	T1(μs)	Retardos(μs)
0,562814	0,550314	0,0125
0,686314	0,687314	0,001
0,687314	0,687793	0,00047904

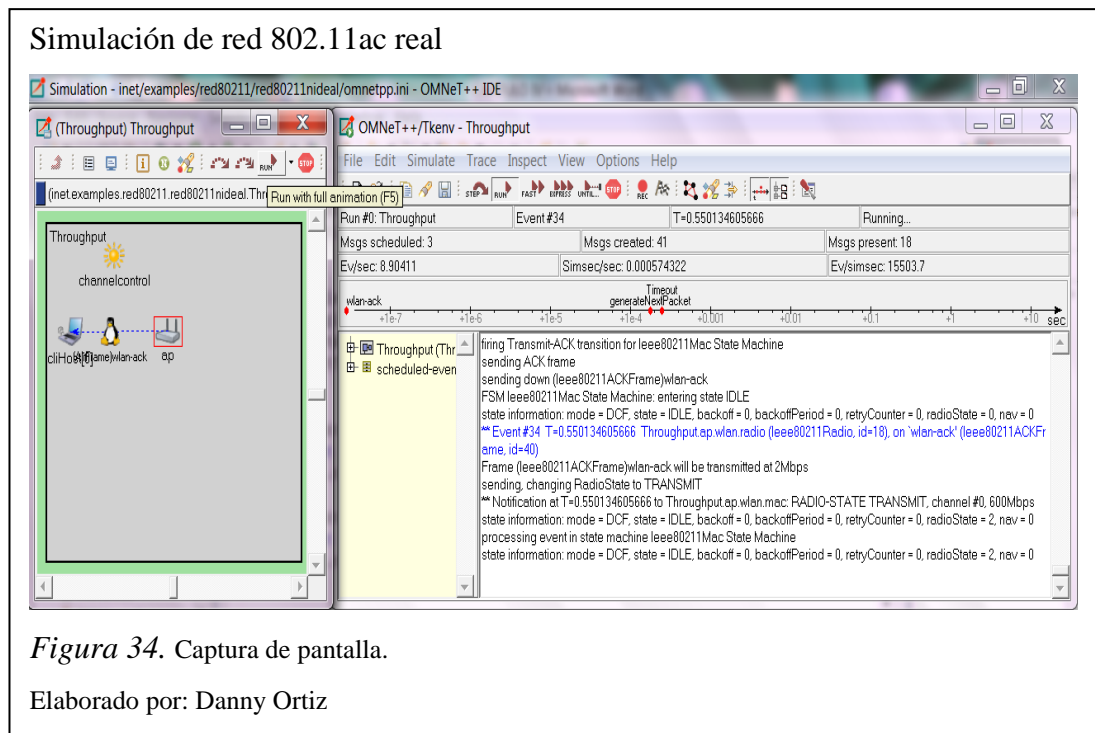
Nota. Calculo de retardos.

Elaborado por: Danny Ortiz

Estos valores de retrasos son aleatorios por lo que el Jitter es aleatorio. Además se puede apreciar que los retardos son bastante irregulares, esto quiere decir que hay más errores en el Jitter.

4.2.2 Simulación IEEE 802.11ac real

Para recolectar los valores pertinentes a la red real en 802.11ac se procede a simular el proyecto red80211ac real y se procede a simular el archivo .ned. La figura 34 muestra el proceso de adquisición de datos. Para realizar este proceso, en la Gned del proyecto se presiona RUN, para que inicie la simulación.



Los datos adquiridos para revisar los valores del BER y el Jitter son los que se presentan a continuación.

Número de eventos y resultados de la simulación:

Datos enviados

Scalar *.ap.wlan.mac numRecibidos 0

Scalar *.ap.wlan.mac numAckEnviados 23724

Scalar *.ap.wlan.mac numRetra 0

Scalar *.ap.sink numPaketes 23724

Scalar *.ap.sink numBits 587286000 #velocidad de transmisión

Scalar *.ap.sink throughput 68358598.009124 #datos enviados

Scalar *.ap.sink paqueteSec 593.1206335427

Scalar *.ap.wlan.mac numEnviados 0

Dato recibidos

Scalar *.PC.sink numPaketes 23721

Scalar *.PC.sink numBits 587224000 #velocidad de recepción

Scalar *.PC.sink throughput 68358555.273458 #datos recibidos

Scalar *.PC.sink packetSec 693.04960612153

Scalar *.PC.wlan.mgmt "packetEs recidos por CW" 0

Scalar *.PC.wlan.mgmt "packetes rebotados por CW" 0

De estos datos recolectados se procede a identificar el BER.

Total de datos perdidos= datos enviados – datos recibidos

Dp= 68358598.009124 – 68358555.273458= 42735666

El BER permitido para una red 802.11n es del orden de 10^{-7} , entonces se procede al cálculo del BER con los datos simulados.

$$BER = \frac{\text{datos perdidos}}{\text{total de datos enviados}}$$

$$BER = \frac{42,735666}{68358598.009124}$$

$$BER = 6,25169E^{-07}$$

Se puede observar que el BER está dentro de lo establecido por la norma 802.11ac, pero menor al BER resultante de la norma 802.11n.

Tabla 18

Datos obtenidos de la simulación 802.11ac real

enum	BACKOFF=3	DEFER=1	IDLE=0	RECEIVE=8	WAITACK=4	WAITBROADCAST=5	WAITCTS=6	WAITDIFS=2	WAITIFS=7
1	1232	601	355	1005	0.563813502304	0.592313502304	25	25	25
0	1833	2705	37	1015	0.550313502304	0.592813502304	112	868	7616
3	4538	9960	0	1016	0	0.592899236642	422	316	528
2	14498	17438	0	1016	0	0.592899236642	738	2624	9760
7	31936	9962	0	1017	0	0.592899746652	22	314	522
6	41898	14986	0	1021	0	0.592909746652	634	742	5194

Nota. Datos de retardos.

Elaborado por: Danny Ortiz

De estos datos se observan los retrasos en las transmisiones tomando en cuenta que eso genera el ruido gaussiano y este el Jitter.

Tabla 19

Datos de retardos en 802.11ac real

t1(μs)	t0(μs)	retardos(μs)
0,592314	0,563814	0,0285
0,5928135	0,550314	0,0425
0,592899	0,552314	0,04058573

Nota. Calculo de retardos.

Elaborado por: Danny Ortiz

Como se observa los valores no son constantes por lo que se demuestra que son aleatorios y por lo tanto el Jitter también es aleatorio. También se puede apreciar que los datos resultantes del retardo son las estables por lo tanto el Jitter es menor que en el caso de la red real 802.11n.

CONCLUSIONES

Tomando en cuenta todo lo realizado en este documento y los resultados de las simulaciones se llega a las siguientes conclusiones:

- En condiciones ideales lo que concierne en tener un espacio vacío y libre de interferencias, algo que es casi imposible, el BER es lo bastante insignificante que tiende a cero, mucho menor que el orden establecido para los formatos de 802.11 que es de 10^{-7} . Pero las condiciones ideales son muy pocas, esto solo serviría para poder establecer los parámetros ideales de las redes. Por ejemplo. La norma 802.11n debe alcanzar velocidades máximas de 600 Mbps más lo que como máximo se puede llegar a 300Mbps, por distintas razones. Por eso la simulación ideal permite que los parámetros ideales o sean casi exactos y esos es justamente lo que se demuestra en el apartado 4.1.1. Como se puede observar los valores números de bits o sea la velocidad es de 584688000 lo que es equivalente a 584,68 Mbps; cercano a los 600 Mbps de lo real.
- La norma 802.11ac debe alcanzar las velocidades máximas de 1,3Gbps, pero en el caso ideal, algo que realmente no sucede por las diversas condiciones que deben pasar las ondas que transmiten los datos, por lo tanto en la simulación ideal se ha demostrado que el máximo valor que se puede alcanzar es de un valor cercano al mencionado, para el caso de este estudio el valor es 987,56 Mbps, tal como se menciona en el apartado 4.1.2.
- La simulación ideal permite definir los parámetros más significativos de las normas 802.11n y 802.11ac, o sea, se puede verificar que las velocidades prescritas por el fabricante se cumplen en realidad, así para 802.11n la velocidad máxima de 600 Mbps, lo que a su vez en la simulación se pudo obtener el parámetro de 584,68 Mbps cerca de 600 Mbps. Además se verifica que la simulación ideal demuestra la velocidad ideal de 802.11ac que en teoría es 1,3Gbps mientras en la simulación llega a 987,86Mbps, cercano a 1,3Gbps.

- En la práctica las simulaciones reales se acercan más a lo que realmente sería una transmisión del protocolo 802.11n o 802.11ac. Para el esquema de simulación 802.11n de forma real se llega a una velocidad máxima de 300Mbps y para la norma 802.11ac una tasa de transmisión de 600Mbps. Para citar el caso específico de las simulaciones hechas, se puede notar que para la norma 802.11n se logró alcanzar una velocidad de 276866000 bps. En el caso de la norma 802.11ac se llegó hasta una velocidad de 587286000 bps. Entonces se está cumpliendo lo esperado.
- En el caso de eficiencia en la transmisión de datos se debe tomar en cuenta que el BER es solo uno de los parámetros que permite estimar lo conveniente o no de una norma para wlan. Haciendo hincapié en eso se puede revisar los datos del BER para verificar en cuál de ellos se perdió más datos.
 - BER ideal de 802.11n es: $8,89231E^{-07}$
 - BER ideal de 802.11ac es: $2,86793E^{-07}$
 - BER real de 802.11n es: $3,16795E^{-06}$
 - BER real de 802.11ac es: $6,25169E^{-07}$
- De estos datos se pueden estimar cuál de las normas es más eficiente en la transmisión y recepción de datos para wlan, para este estudio se nota que en los dos casos tanto ideal como real, la norma 802.11ac es más eficiente que 802.11n.
- Se observó que al transmitirse los datos a través del medio existen retardos por los rebotes de la señal, por la conversión de datos, por la forma de modulación, y otros factores. Entonces como se generan retardos por situaciones poco predecibles el Jitter es aleatorio, por lo tanto lo que se puede analizar es que los retardos en la red 802.11n son mucho más irregulares que en el caso de la red 802.11ac, por lo que se concluye que la red 802.11ac es mucho más estable que la red 802.11n.

RECOMENDACIONES

- Se recomienda la ejecución del programa de simulación dentro de una máquina virtual, ya que se tuvo conflictos con otros programas que utilizan java cuando se ejecutó en una pc normal.
- Se recomienda no utilizar Windows 8 ya que por sus seguridades y su reciente diseño, no se tiene suficiente compatibilidad con los módulos y subprogramas de OMnet++.
- Se recomienda ir verificando el correcto funcionamiento de cada módulo simple desarrollado, para que cuando se los junte para formar el módulo compuesto no tener posibles errores de funcionamiento.
- Se recomienda actualizar a las últimas versiones de Java, OMnet++, VMware Workstation e Inet, para evitar futuros conflictos por falta de librerías o desactualizaciones.
- Se recomienda que el programa de simulación realizado para este análisis comparativo sea utilizado en un futuro para realizar nuevos estudios entre estos dos estándares, como por ejemplo cuál de los dos estándares es más efectivo en cuanto a penetración, alcance, etc.

ANEXOS

Anexo 1.

Programación Gned de los módulos de interfaz gráfica

Programación de Interfaz gráfica del Nodo

```
simple nodo                                //iniciación del módulo simple
    parameters:                          //parámetros del módulo
    txRate: numeric const;               //velocidad de transmisión
    radioDelay: numeric const;           //retardo de propagación
    pkLenBits: numeric const;            //longitud del paquete de bits
    igTime: numeric const;               //tiempo de separación de paquetes
    numnodos: numeric const;             //número de nodos a trabajar
    ventana: numeric const;              //ancho de banda
    slotTime: numeric const;             //ranura de tiempo de backoff o de memoria
    gates:                               //compuertas de conexión con otros módulos
    in: in;                             //compuerta de recepción de mensajes
    out: out;                            //compuerta de envío de mensajes
endsimple                                //fin del módulo
```

Programación del módulo AP.

```
simple ap                                //iniciación del módulo simple
    parameters:                          //parámetros del módulo
    txRate: numeric const;               //velocidad de transmisión
    radioDelay: numeric const;           //retardo de propagación
    pkLenBits: numeric const;            //longitud del paquete de bits
    igTime: numeric const;               //tiempo de separación de paquetes
    ofdm: numeric const;                 //número de tramas por slot de tiempo
    ventana: numeric const;              //ancho de banda
    slotTime: numeric const;             //ranura de tiempo de backoff o de memoria
    gates:                               //compuertas de conexión con otros módulos
    in: in;                             //compuerta de recepción de mensajes
    out: out;                            //compuerta de envío de mensajes
endsimple                                //fin del módulo
```

Programación del módulo compuesto Wireless

```
module wireless                                //declaración del módulo compuesto
parameters:                                   //declaración de parámetros
    txRate: numeric const;
    ventana: numeric const;
    p: numeric const;
    P: numeric const;
    ofdm: numeric const;
    BER: numeric const;
    numnodos: numeric const;
submodules:                                   // declaración de los módulos simples
    nodo: nodo(numnodos);                     // módulo nodo
    parameters:
        txRate=txRate;
        ventana=ventana;
        numnodos=numnodos;
        display: "p=44,64;i=device/wifilaptop"; // posición espacial del
módulo
    ap: ap;                                    //módulo AP
    parameters:
        txRate=txRate;
        display: "p=192,64;i=device/accesspoint"; //posición especial del
módulo
    canal: canal;                             // declaración del módulo canal
    parameters:
        p=1;
        P=1;
        BER=1;
        Numnodos = numnodos;
    gatesizes:                                //entradas y salidas
        in[numnodos+1];
        out[numnodos+1];
        display: "p=192,64;i=block/queue";//posición especial del módulo
        display: "b=220,117";
```

```

        connections:           //conexiones de comunicación de módulos
for i=0..numnodos-1 do      //lazo de comunicación de conexiones
    nodo[i].out --> canal.in[numnodos];
    nodo[i].in <-- canal.out[numnodos];
endfor
    ap[i].out --> canal.in[numnodos];  //declaración de variables
entrada y salida
    ap[i].in <-- canal.out[numnodos];
endmodule

```

Programación de wlan Wireless:

```

Parameters:           //declaración de parámetros en la red
                      //código de datos de entrada de usuario.

numnodos=input(2,"numero de nodos: ");
p=input(0.000001,"probabilidad de cambio de estado bueno a malo");
P= input(0.002,"probabilidad de cambio de estado malo a bueno");
ventana= input(31,"Ingrese el valor de la ancho banda entre 20 y 40:");

```

Anexo 2.

Programación C++ de los módulos

```
void nodo::initialize()
{
    double z;                                //variable de contención
    double back;
    back=dblrand();                          //número aleatorio para el envío de datos
    z=abs(back*ventana);
    slotTime=z*0.000002;                     // ranura de tiempo 9us
    ev.print("el valor del backoff del nodo%d es: %4,10f \n",id()-2,slotTime);

    if (strcmp("nodo", getName()) == 0)      //condicionante recepción de mensajes
    {                                          por datos ingresados.
        cMessage *msg = new cMessage("wlan802.11Msg");
        send(msg, "out");
    }
}
```

```
void nodo::handleMessage(cMessage *msg)      //función de tratamiento de mensajes
{
    char pname[40];                          //declaración de variables
    sprintf(pname,"pk-%d-#%d",id()-2,pkCounter++);
    ev <<"generating packet" <<pname << endl;

    state = TRANSMIT;                        //define el comando de envío de datos
    sendDelayed(msg, radioDelay, "out");

    endTxEvent=cMessage("endTxEvent"); //finalización de envío de datos
    state=IDLE;
    double txt=0.009384;                     //limitante de datos
    double DIFS=0.00005;                     //valor de tiempo entre trama
    time1=slotTime+DIFS+txt;                  //tamaño de la trama enviada
    ev.printf("tiempo de atrazo: %4.10f nodo: %d \n",time1,id()-2);// el total de la
trama
    scheduleAt(time1, endTxEvent);

    double z;                                //declaración de datos
    double back;
    back=dblrand();
    z=abs(back*ventana);
    slotTime=z*0.0001;                       //tamaño de la ventana OFDM
    double txt=0.009384;                     //valor limitante de datos
    double DIFS=0.00005;                     //valor del tiempo entre trama
    time1=slotTime+DIFS+txt+simTime;          //tamaño de la trama recibida
}
```



```

ev.printf("tiempo real de llegada: %4.10f nodo: %d \n",time1,id()-2);
                                     //el total de la trama
    scheduleAt(time1, endTxEvent);
    send(msg, "out");
}
Define_Module(ap);

void ap::initialize()
{
    double z;
    double back;
    back=dblrand();
    z=abs(back*ventana);
    slotTime=z*0.000002; /// ranura de tiempo 20us
    ev.print("el valor del backoff del nodo%d es: %4,10f \n",id()-2,slotTime);

    if (strcmp("ap", getName()) == 0)
    {
        cMessage *msg = new cMessage("wlan802.11Msg");
        send(msg, "out");
    }
}

void ap::handleMessage(cMessage *msg){
    totalTramas++;
    double endreception = simTime();
    if(!canalOcupado)
    {
        ev<<"iniciado recepcion\n";
        grabTiempoIni=simTime();
        canalOcupado=true;
        scheduleAt(endReceotion,finRxEvento);
    } else
    { ev<<"otra trama llego mientras recibia - colision\n";
      tramasColi++;
      if(numeroTramasCol==0)
          numeroTramasCol==2;
      else
          numeroTramasCol++;
      if(endReception>finRxEvento->arrivalTime())
      {
          cancelEvent(finRxEvento);
          scheduleAt(endReception,finRxEvento);
      }
    }
    send(msg, "out");}

```

Anexo 3.

Creación del módulo canal:

```
simple canal                                //inicia el módulo
    parameters:                            //parámetros de dispositivo
    ventana:numeric const;                 //tamaño del ancho de banda
    BER:numeric const;                     //tasa de error bit
    P:numeric const;                       //probabilidad de colisión estado malo a bueno
    p:numeric const;                       //probabilidad de colisión estado bueno a malo
    numnodos:numeric const;                //número de nodos en la red
    gates:                                 //compuertas de conexión
        in:in[];                           //vector de mensajes entrada
        out:out[];                          //vector de mensajes salida
```

Programación C++ del módulo canal

```
#include <string.h>                        //declaración de librerías
#include <OMNetpp.h>

class colision : public cSimpleModule
{ protected:                             //declaración de las funciones
    virtual void initialize();
    virtual void activity();
    virtual void finish();
};

Define_Module(colision);

void colision::initialize(){
if (strcmp("nodo", getName()) == 0) {
    cMessage *msg = new cMessage("wlan802.11Msg");
    send(msg, "out");
}}
}}
```

```

void colision::activity(){
    cMessage *msg;
    msg receive();
    double r=uniform(0.0001,1);    //uso de la variable de probabilidad UNiforme
    double a=log(r);                //definición de variables para la probabilidad de
    choques de tramas
    double b=log(1-p);
    double c=log(1-P);
    double d=log(BER);
    double kb=a/b;
    double km=a/c;
    double kh=a/d;

    nc:
    if(kb>=6384)                    //condicionante valor de tramas
    { ev << "paquete sin errores\n";
      send(msg,"out",numnodos); }
    else {                          //tratamiento de valor de la trama
      double r1=uniform(0.0001,1);
      double a1=log(r1);
      double kb1=a1/b;
      kb=kb+kb1;
      double km1=a1/c;
      kh=a1/d;
      ev<<"kb=["<<kb<<"] km=["<<km<<"] kh=["<<kh<<"]\n";
      if(kh>km1)
      { kb=kb+km;
        ev<<"kb=["<<kb<<"]\n";
        goto nc; }
      else { goto nuevoerror;}
    }
    if(kb>6384){                    // condicionante de valor de trama
      ev << "paquete sin errores\n";
      send(msg,"out",numnodos);}
    else { goto nuevoerror; }

```

```

    send(msg, "out");
continuar:                                // desarrollo de las ventanas OFDM
    double r2=uniform(0.0001,1);
    double a2=log(r2);
    double kh2=a2/b;
    kh=kh+kh2;

    if(kh<km) { goto continuar;}
    else { kmt=km+kmt;
        if((kb+kmt)>=6384) {
            ev<<"kh=["<<kh<<" kb=["<<kb<<" km=["<<km<<"
kmt=["<<kmt<<"]\n";
            ev<<"paquete con errores en posiciones:\n";
            send(msg, "out", nodos);}
        else {
            kh=0;
            double r3=uniform(0.0001,1);
            double a3=log(r3);
            double kb2=a3/b;
            kb=kb+kb2+kmt;
            goto continuar;}
        }}

void colision::finish() {                //finalización de las funciones y parámetros presentados
    ev<<"P"<<P<<endl;
    ev<<"p"<<p<<endl;
    ev<<"BER"<<BER<<endl;
    ev<<"throughput"<<sum/ld<<endl;
    ev<<"numero de tramas transmitidas desde TX"<<ld<<endl;
    ev<<"numero de tramas que llegan al AP"<<ld-cont1<<endl;
    ev<<"numero de paquetes validosque llegan errados al AP"<<numpqterr<<endl;
    ev<<"número de canales definidos por AP"<<numcanal<<endl
}

```

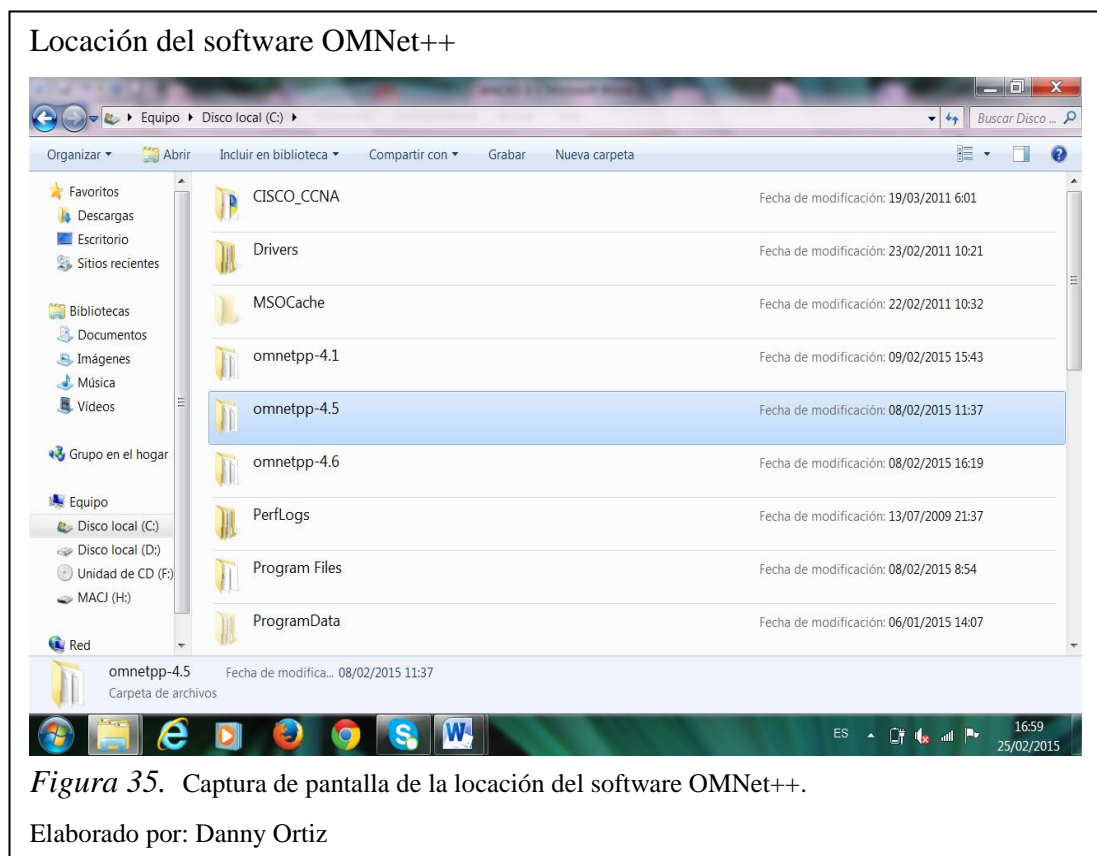
Anexo 4.

Manual de ejecución del programa

Con el objetivo de realizar la ejecución de la simulación de cada uno de las redes diseñadas para la verificación del presente proyecto se procede como se indica a continuación.

Paso 1. Ubicación del software de simulación OMNet++.

Como se analizó en el Capítulo I, el software debe estar instalado previamente. Para este proyecto específicamente se ha usado la versión OMNet 4.5. La instalación se lo ha hecho en la localidad C:/omnetpp 4.5; tal como se observa en la figura 35.

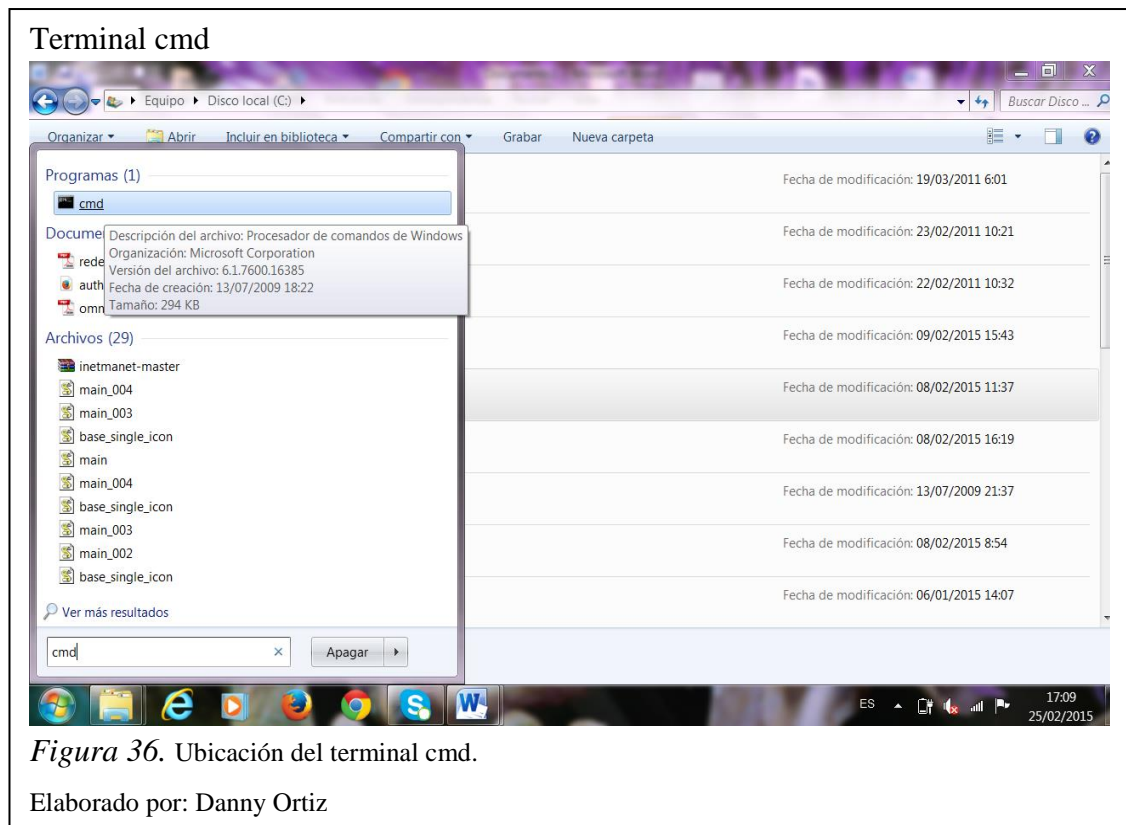


Paso 2. Ejecución del Software OMNet++.

Existen dos formas de ejecutar OMNet++.

En el sistema operativo Windows, la primera; es por la línea de comando CMD, por medio de usar las siguientes órdenes.

Ejecutar la terminal cmd, tal como se observa en la figura 36.



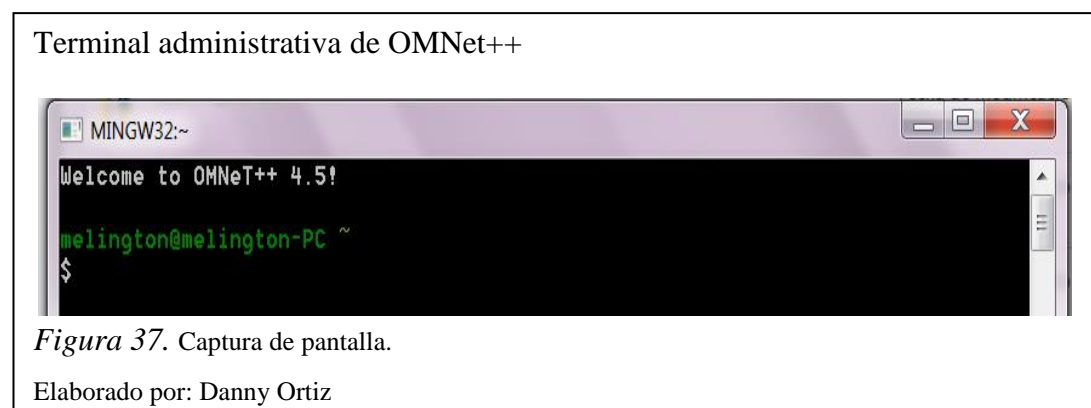
Se procede a escribir las siguientes órdenes.

```
C:/cd /
```

```
C:/>cd omnetpp -4.5
```

```
C:/omnetpp -4.5>mingwenv.cmd
```

Después de este último comando se genera la ventana de activación del software OMNet++. Como se observa en la figura 37.



Dentro de esta ventana de activación se procede a escribir el comando: **omnetpp**

Se espera un momento y se puede observar la ventana de interfaz gráfico de OMNet++.

Ventana de inicio de Interfaz Gráfico de OMNet++

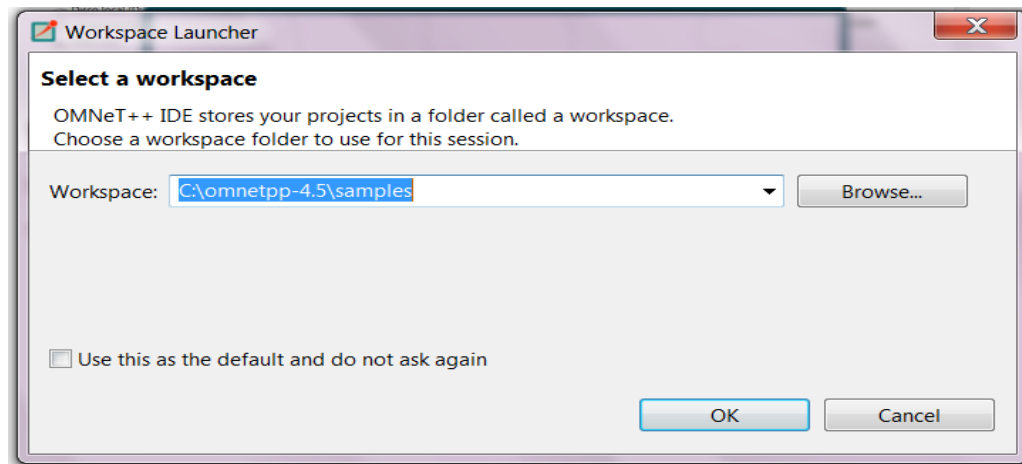


Figura 38. Interfaz de la ventana de inicio de OMNet++.

Elaborado por: Danny Ortiz

Como se observa en la figura 38, la ventana de interfaz gráfico permite seleccionar por default la carpeta de trabajo de OMNet++, lugar donde se pueden guardar las simulaciones realizadas en este trabajo. Se presiona OK y se espera un momento. En seguida aparece la ventana de trabajo de OMNet++. Como en la Figura 39.

Ventana de trabajo de OMNet++

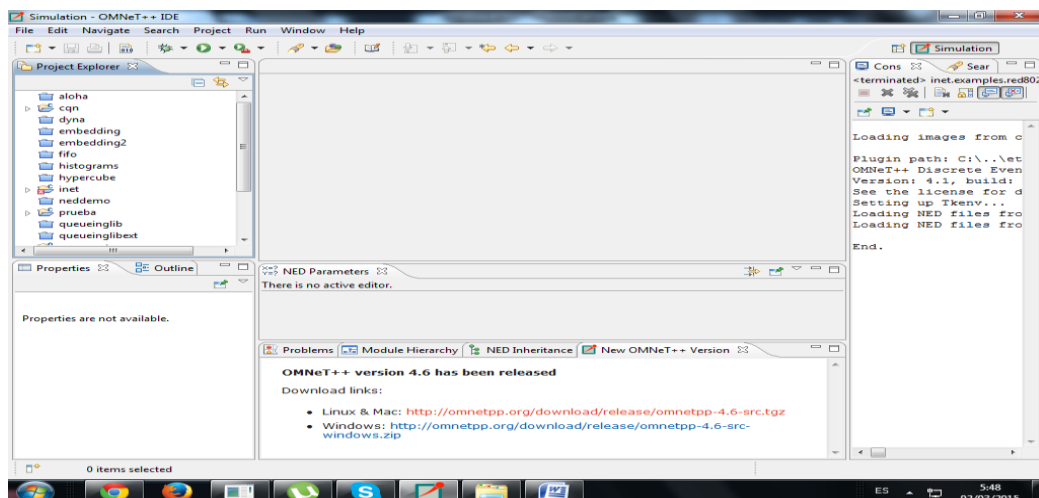


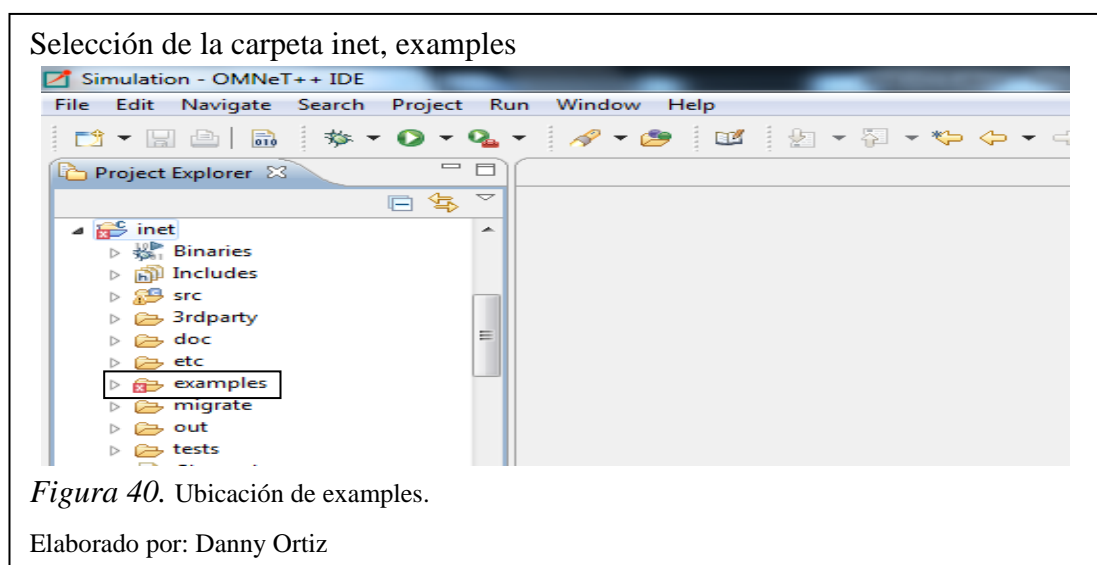
Figura 39. Captura de pantalla de la ventana de trabajo de OMNet++.

Elaborado por: Danny Ortiz

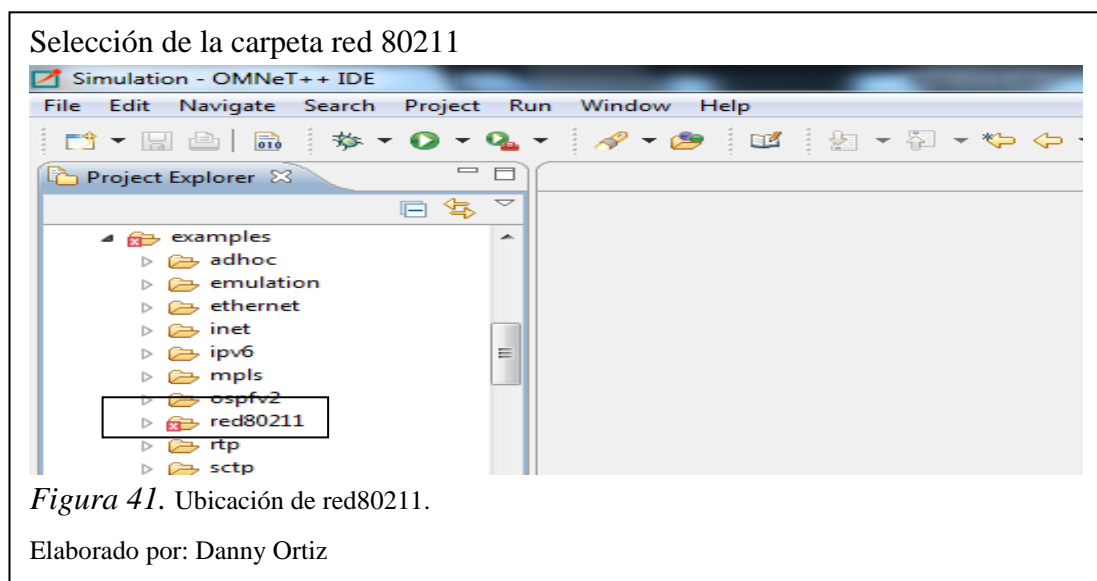
Paso 3. Activar proyecto

Para realizar algún trabajo en OMNet++ se deben respetar el orden de trabajo estudiado en el capítulo II. Entonces al realizar algún proyecto de simulación diseñado, se procede a crearlo desde esta ventana y guardarlo en la carpeta por defecto.

El contenido de la carpeta por defecto aparece en el lado izquierdo de la ventana de trabajo y se encuentra un árbol de selección. En ese lugar se selecciona la carpeta inet y dentro de esta la carpeta examples como se puede observar en la figura 40.

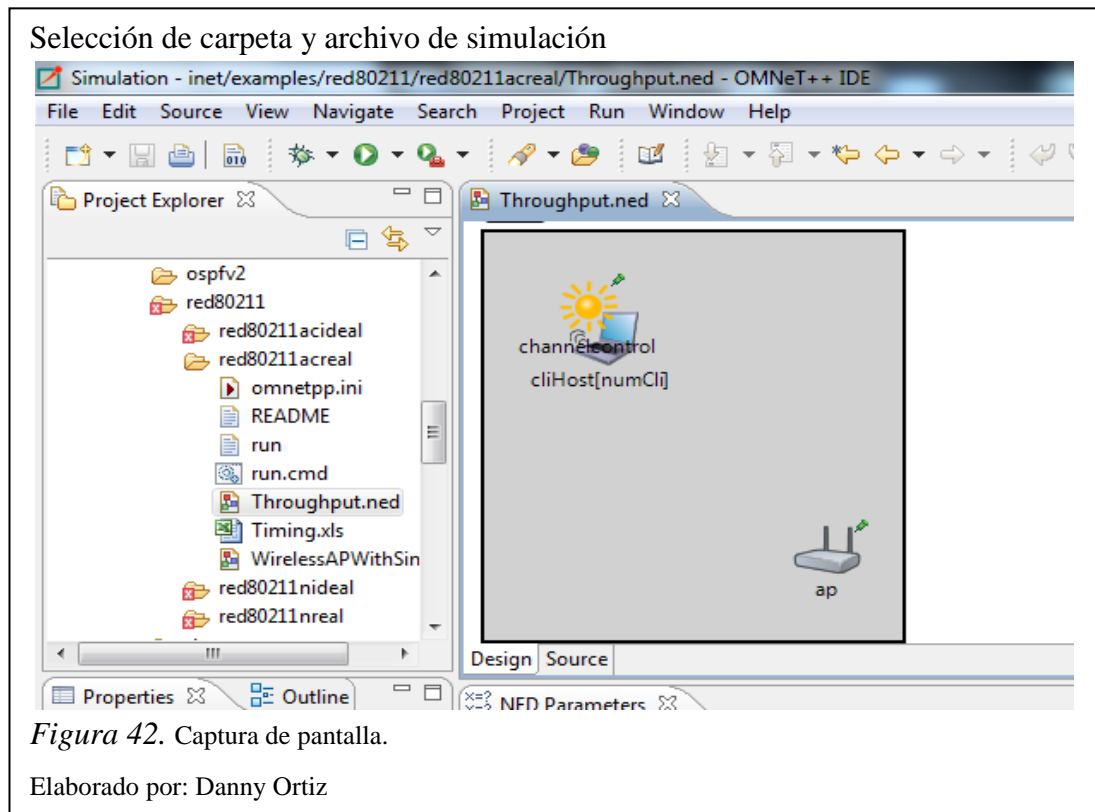


En la carpeta examples, se elige la carpeta red80211, la misma que contiene de las redes diseñadas, esto en la figura 41.



Dentro de la carpeta red80211 se encuentran las carpetas del diseño: ed80211nideal, red80211nreal, red80211acideal, red80211acideal.

Se abre cada carpeta dando doble clic sobre la misma, tal como se mira en la figura 42.



Dentro de cada carpeta se despliegan varios elementos como archivos de control y ejecución. Entre los más importantes están:

- Archivo .ned
- Archivo.ini

Paso 4. Simulación de la red diseñada

Luego de haber seleccionado la carpeta donde se encuentran las redes de simulación del proyecto, se escoge el archivo .ned. Se da doble clic sobre él y aparece en el prompt gráfico de la derecha en la imagen, de los dispositivos a simular, tal como se observa en la figura 42.

Ahora se hace uso de los botones de simulación que se pueden ver en la parte superior, bajo el tablero de Menú, como se observa en la figura 43.

Botones de activación de proyecto

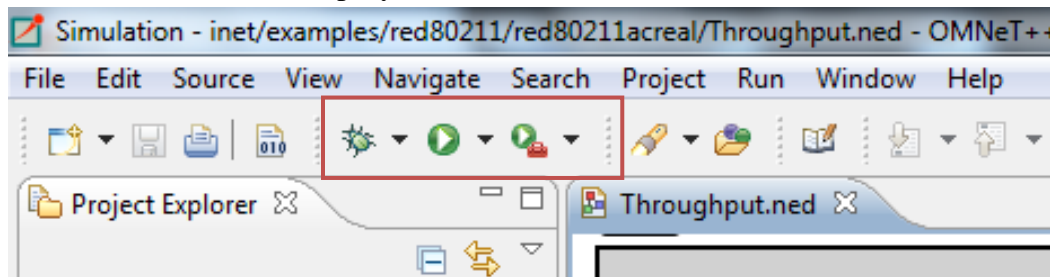


Figura 43. Ubicación de los Botones de activación.

Elaborado por: Danny Ortiz

Cuando se haya presionado el botón de RUN, se puede observar que aparece la ventana de GNED, para generar la simulación del programa diseñado, esto se observa en la figura 44. Además dentro de esta ventana se debe nuevamente presionar el botón RUN para que el programa empiece a funcionar.

Ventana de simulación GNED

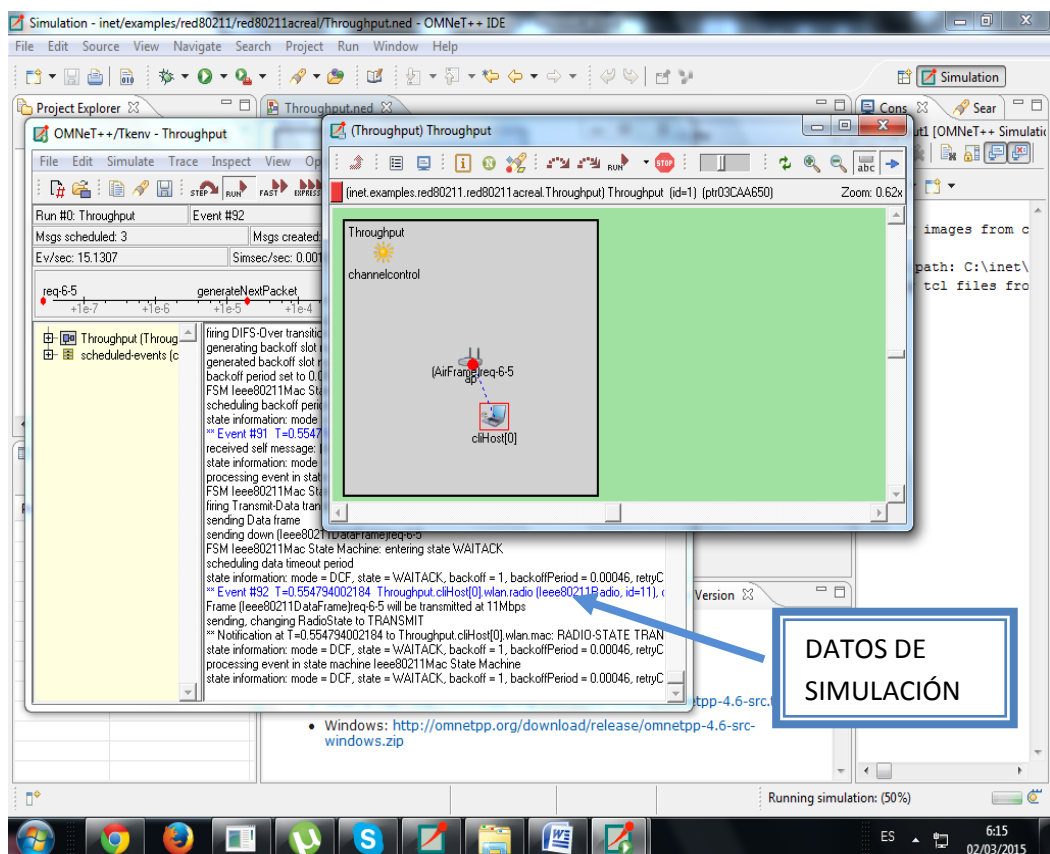


Figura 44. Ejecución de la simulación GNED.

Elaborado por: Danny Ortiz

Tal como se puede verificar en la figura 44, la generación de datos del programa se realiza en la ventana detrás del interfaz gráfico de simulación. Esos datos son los datos de simulación desde donde el usuario puede medir las condiciones propuestas en el programa. Para el caso de este proyecto, tiempos de espera, cantidad de datos de entrada y salida, valores de canal, entre otros.

Estos datos como se encuentran en la simulación no se los puede tomar a tiempo real, por lo tanto se deben guardar en el archivo de terminación, VCI, dentro de la carpeta que se muestra en la figura 45.

