DESARROLLO DE PRÁCTICAS DE LABORATORIO APLICANDO RADIO DEFINIDO POR SOFTWARE PARA SISTEMAS DE TELEFONÍA MÓVIL CELULAR.

David Norberto Angulo Esguerra

Código: 2101723

Universidad Santo Tomás

Facultad de Ingeniería de Telecomunicaciones

Bogotá, D. C. Septiembre de 2015

DESARROLLO DE PRÁCTICAS DE LABORATORIO APLICANDO RADIO DEFINIDO POR SOFTWARE PARA SISTEMAS DE TELEFONÍA MÓVIL CELULAR.

David Norberto Angulo Esguerra

Código: 2101723

Director: Ingeniera

Mónica Espinosa Buitrago

Asesor: Ingeniero

Iván Díaz Pardo

Universidad Santo Tomás

Facultad de Ingeniería de Telecomunicaciones

Bogotá, D. C. Septiembre de 2015

CONTENIDO

1. INTRODUCCIÓN. 1

2. PLANTEAMIENTO DEL PROBLEMA. 4

OBJETIVO GENERAL. 6

2.1 Objetivos Específicos. 6

3. JUSTIFICACIÓN. 7

4. EVOLUCIÓN DE LA TELEFONÍA MÓVIL. 9

4.1 GSM (Global System for Mobile Telecommunications). 14

4.1.1 Arquitectura de red GSM. 15

4.1.2 Tecnología de acceso al medio (FDMA - TDMA). 17

4.2 LTE (Long Term Evolution). 18

4.2.1 Arquitectura de red LTE. 19

4.2.2 Tecnología de acceso al medio (OFDM – OFDMA – SC-FDMA). 21

5. RADIO DEFINIDO POR SOFTWARE ETTUS USRP N210. 23

5.1 GNURadio. 25

5.2 Herramienta SDR para GSM (OpenBTS). 26

5.3 Herramienta SDR para LTE. 27

6. METODOLOGÍA. 31

7. IMPLEMENTACIÓN DE UNA ESTACIÓN BASE GSM (2G) OPENBTS SOFTWARE. 34

7.1 Componentes de Hardware: 34

7.2 Componentes de software: 34

7.3 Proceso de actualización. 35

7.4 Proceso de descarga. 36

7.4.1 Requisitos previos para la instalación. 37

7.5 Proceso de instalación. 37

7.5.1 Instalación de Asterisk. 39

7.5.2 Instalación Sipauthserve. 39

7.5.3 Instalación de Smqueue. 39

7.5.4 Instalación de OpenBTS. 40

7.6 Proceso de verificación de los servicios. 40

7.6.1 Proceso Asterisk. 41

7.6.2 Proceso Sipauthserve. 41

7.6.3 Proceso Smqueue. 41

7.6.4 Proceso OpenBTS. 42

7.7 Verificación de conectividad vía puerto Gigabit Ethernet. 42

7.8 Prueba de conectividad. 44

7.9 Puesta en marcha de la estación base y configuración parámetros iniciales. 45

7.10 Capturas analizador de espectro. 49

7.11 Búsqueda de la red en los terminales. 52

7.11.1 Terminales sistema operativo iOS. 52

7.11.2 Terminales sistema operativo Android. 53

7.12 Adición de un terminal a la base de datos. 56

7.13 Servicio de mensajería de texto. 57

7.13.1 Mensaje de texto como prueba de conectividad desde el terminal hacia la estación base. 58

7.13.2 Mensaje de texto desde la estación base hacia el terminal. 58

7.14 Servicio de llamadas de voz. 60

7.14.1 Servicio de llamada de voz entre dos terminales. 61

7.15 Capturas de tráfico en el software Wireshark. 62

8. RECEPCIÓN DE SEÑALES LTE (4G) EN GNURADIO. 65

8.1 Componentes de Hardware: 65

8.2 Componentes de software: 65

8.3 Proceso de descarga bloques lte. 65

8.4 Proceso de instalación bloques lte. 67

8.5 Desarrollo de flujograma en GNURadio 70

8.5.1 Bloque de conexión del hardware USRP N210 con GNURadio. 70

8.5.2 Diagrama de bloques de sincronización. 72

8.5.3 Diagrama de bloques para la recepción OFDM. 77

8.5.4 Diagrama de flujo demultiplexación y decodificación PBCH. 79

8.5.5 Diagrama de flujo decodificación BCH y MIB. 81

9. CONCLUSIONES. 88

10. REFERENCIAS BIBLIOGRÁFICAS. 90

TABLA DE FIGURAS.

Figura 1. Tiempo en ejecución en HSDPA y HSUPA. 12

Figura 2. Arquitectura red GSM - GPRS. 15

Figura 3. Arquitectura red LTE....20

Figura 4. Ettus USRP N210. 23

Figura 5. Arquitectura OpenBTS. 26

Figura 6. Bloques de recepción OFDM – LTE. 28

Figura 7. Diagrama metodología. 31

Figura 8. Ubicación archivos de descarga. 38

Figura 9. Verificación Asterisk como proceso en el sistema. 41

Figura 10. Verificación Sipauthserve como proceso en el sistema. 41

Figura 11. Verificación Smqueue como proceso en el sistema. 42

Figura 12. Verificación OpenBTS como proceso en el sistema. 42

Figura 13. Dirección IP del USRP por defecto en consola. 43

Figura 14. Características USRP en consola. 44

Figura 15. Detección del USRP en el ordenador. 44

Figura 16. Cambio dirección IP puerto Gigabit Ethernet del ordenador. 45

Figura 17. Prueba de conectividad con el equipo USRP. 45

Figura 18. Ingreso a CLI de OpenBTS. 46

Figura 19. Configuración inicial de la estación base. 46

Figura 20. Configuración banda de frecuencia en consola. *47*

Figura 21. Configuración canal inalámbrico en consola. *48*

Figura 22. Proceso de reinicio de consola OpenBTS. *48*

Figura 23. Ruido en la interfaz de aire en consola. *48*

Figura 24. Configuración potencia de la portadora en consola. *49*

Figura 25. Portada en banda de frecuencia de 850 MHz y canal en 166. ..... *49*

Figura 26. Cambio de banda de frecuencia a 900 MHz en consola. *50*

Figura 27. Reinicio del servicio OpenBTS en consola. *50*

Figura 28. Portadora en banda de frecuencia de 900 MHz en analizador de espectro. *51*

Figura 29. Portadora en banda de frecuencia de 900 MHz con 0dB de potencia. *51*

Figura 30. Ancho de banda portadora en banda de frecuencia de 900 MHz. *52*

Figura 31. Búsqueda de la red Test PLMN en iOS. *53*

Figura 32. Búsqueda de la red Test PLMN en Android. *54*

Figura 33. Información del comando tmsis en el CLI de OpenBTS. *54*

Figura 34. IMEI del terminal iPhone. *56*

Figura 35. Comando para agregar un terminal a la base de datos en consola. *56*

Figura 36. Conexión establecida de terminal con red Test PLMN 1-1. ..... *57*

Figura 37. Mensaje como prueba de conectividad enviado desde el terminal. *58*

Figura 38. Mensaje de texto desde CLI de OpenBTS hacia terminal. *59*

Figura 39. Mensaje de texto satisfactorio desde CLI de OpenBTS hacia terminal. *59*

Figura 40. Eventos de mensajería de texto en el CLI de OpenBTS. *60*

Figura 41. Canal de tráfico activo en un llamado de voz. *60*

Figura 42. Canales de tráfico activos en llamada de voz. *61*

Figura 43. Eventos de llamadas de voz en el CLI de OpenBTS. *61*

Figura 44. Captura protocolo de señalización SIP. *62*

Figura 45. Captura protocolo de cabecera GSM gsmtap. *63*

Figura 46. Captura protocolo de transmisión UDP. *64*

Figura 47. Clonación del paquete lte en Linux. *66*

Figura 48. Archivos de descarga del paquete lte. *66*

Figura 49. Ubicación archivos de cabecera para los bloques lte. *67*

Figura 50. Ubicación archivos de ejemplo de bloques lte. *67*

Figura 51. Compilación de los bloques lte en consola Linux. *68*

Figura 52.Ejecución de los bloques lte en consola Linux. *69*

Figura 53. Instalación de los bloques lte en consola Linux. *69*

Figura 54. Ubicación del módulo lte. *70*

Figura 55.Bloque conexión USRP N210 con GNURadio. *70*

Figura 56. Configuración bloque UHD:USRP Source *71*

Figura 57. Configuración bloque WX GUI Slider. *71*

Figura 58.Configuración variable samp_rate. *72*

Figura 59. Diagrama de bloques de sincronización. *72*

Figura 60.Configuración bloque CP-based FFO Sync. *73*

Figura 61. Configuración bloque PSS Symbol Selector. *73*

Figura 62. Configuración bloque FFT. *73*

Figura 63.Configuración bloque Extract subcarriers. *74*

Figura 64.Configuración bloque PSS Calculator. *74*

Figura 65. Configuración SSS Frame Tagger. *75*

Figura 66. Configuración bloque Signal Source. *75*

Figura 67. Configuración bloque Multiply. *75*

Figura 68. Configuración bloques CP FFO Calculator. *76*

Figura 69.Configuración bloque SSS Symbol Selector. *76*

Figura 70.Configuración bloque SSS Calculator. *76*

Figura 71. Configuración bloque SSS Frame Tagger. *77*

Figura 72.Diagrama de bloques recepción OFDM *77*

Figura 73.Configuración bloque Remove CP. *78*

Figura 74. Configuración bloque Channel estimator generic *78*

Figura 75. Configuración del bloque RS map Generator. *78*

Figura 76. Configuración variable importación lte. *79*

Figura 77.Diagrama de bloques demultiplexación y decodificación PBCH. *79*

Figura 78. Configuración bloque PBCH Demux. *80*

Figura 79. Configuración bloque Pre Decoder. *80*

Figura 80. Configuración bloque Layer demapper. *80*

Figura 81.Configuración bloque Interleave. *81*

Figura 82. Configuración bloque QPSK soft demod. *81*

Figura 83.Diagrama de flujo decodificación BCH y MIB. *82*

Figura 84. Configuración bloque Repeat. *82*

Figura 85.Configuración bloque Stream to Vector. *83*

Figura 86. Configuración bloque Scrambling sequence Message *83*

Figura 87. Configuración bloque Repeat Message Source. *83*

Figura 88. Configuración bloque Multiply. *83*

Figura 89. Configuración bloque Vector to Stream. *84*

Figura 90. Configuración bloque Stream to Stream. *84*

Figura 91. Configuración bloque Add. *84*

Figura 92. Configuración bloque Multiply Const. *85*

Figura 93. Configuración Stream Mux. *85*

Figura 94.Configuración bloque Subblock deinterleaver. *85*

Figura 95. Configuración bloque BCH Viterbi. *86*

Figura 96. Configuración bloque Check CRC16. ..... *86*

Figura 97. Configuración bloque CRC antenna selector. *86*

Figura 98. Configuración bloque Unpack MIB. *87*

TABLA DE TABLAS.

Tabla 1. Comparación generaciones de la TMC. 13

1. INTRODUCCIÓN.

Actualmente el sector de las telecomunicaciones ha experimentado un crecimiento a gran escala, sólo en Colombia el número de líneas activas de telefonía móvil celular (TMC) según el boletín trimestral del MINTIC (ministerio de las tecnologías de la información y las comunicaciones), correspondiente al tercer periodo del año 2014, ha crecido en un 112,4% con respecto al mismo periodo del año 2013 donde la penetración de la TMC se encontraba en un 103,2%. De esta manera, se sumaron casi cinco millones de nuevos abonados a la telefonía móvil en un año [1]. Por su parte, el internet móvil ésta a punto de triplicar las cifras del internet fijo con aproximadamente veinte millones de suscriptores, contra algo más de cinco millones que según el MINTIC utilizan internet fijo. Estas cifras indican que los servicios móviles que utilizan el espectro radioeléctrico tienen mayor demanda a comparación de los servicios alámbricos, de allí, se identifica la necesidad de brindar cada vez mejores servicios en relación a temas de cobertura, velocidad e infraestructura [1].

En el marco de la telefonía móvil, se ha evolucionado en tecnología desde la creación de GSM (Global System for Movile Communications), llamada la segunda generación (2G), avanzando por UMTS (Universal Mobile Telecommunications System), tercera generación (3G), hasta llegar a LTE (Long Term Evolution), cuarta generación (4G). La constante evolución ha creado nuevas y novedosas aplicaciones entorno a las comunicaciones multimedia, pero los usuarios han experimentado falencias en la prestación del servicio debido a la insuficiencia de infraestructura [2].

La cuarta generación (4G), y la tecnología LTE nacen como un novedoso modelo de negocio, especialmente para los operadores móviles, por esta razón, en la

subasta realizada en Colombia, para el alquiler de una porción del espectro electromagnético comprendido en un total de ocho (8) bloques entre las bandas AWS y 2.5 GHz, se recogieron en total 770.539 millones de pesos; data la subasta del 26 de Junio del año 2013, en el cual participaron las principales compañías del sector de telecomunicaciones como Claro, Directv, Avantel, eTb en conjunto con Tigo y Movistar [3].

Estas nuevas incorporaciones a los modelos de negociación, derivan en una presión tecnológica y profesional, que impactan directamente a la población estudiantil que se encuentra en procesos de preparación académica en el área de la ingeniería.

Por esta razón, fue importante realizar actividades complementarias en la adquisición y fortalecimiento del conocimiento en la facultad de ingeniería de telecomunicaciones de la Universidad Santo Tomás, ya que de esta manera el egresado perteneciente a dicha facultad, tendrá las capacidades necesarias para enfrentarse a las complejidades que presenta el sector de la telefonía móvil y el sector TIC en general.

Se proyectó utilizar las herramientas disponibles en la Universidad Santo Tomás para desarrollar prácticas de laboratorio enmarcadas en los sistemas de radio definido por software SDR (Software Defined Radio), donde se identifique la configuración de una estación base en una red de segunda generación (2G) y recepción de señales en redes de cuarta generación (4G).

Para las prácticas de laboratorio, se utilizó como dispositivo esencial un USRP (Universal Software Radio Peripheral) de referencia N210, que posee gran capacidad de procesamiento de señal, con variabilidad de frecuencia y compatibilidad al software libre GNURadio. Al igual que un ordenador con puerto

GigaEthernet para la conexión con el USRP y demás herramientas complementarias de software y hardware.

2. PLANTEAMIENTO DEL PROBLEMA.

La asignatura de sistemas de telecomunicaciones móviles que pertenece a la facultad de ingeniería de telecomunicaciones de la Universidad Santo Tomás, actualmente no cuenta con todas las herramientas en relación a infraestructura de telecomunicaciones que permita realizar prácticas de laboratorio por parte del estudiante, siendo un espacio de gran importancia para el aprendizaje y fortalecimiento de las competencias de la población estudiantil. Por esta razón, se plantea utilizar radio definido por software como método de ayuda en el fortalecimiento de la infraestructura de los sistemas de comunicaciones móviles, teniendo grandes ventajas para los estudiantes de ingeniería de telecomunicaciones, puesto que trabajarían con tecnologías emergentes en el mercado y realizarían prácticas de laboratorio que permitirían cumplir con los requerimientos del sector TIC.

La universidad y su formación pertinente son el enlace adecuado donde el estudiante tomasino recolecta las condiciones necesarias que lo llevarán a estar contextualizado. Además, el manejo del SDR será un espacio adecuado para el fortalecimiento de las capacidades intelectuales de los estudiantes de ingeniería de telecomunicaciones.

El alcance de las prácticas de laboratorio a desarrollar, se basó en la implementación de una estación base funcional de una red de segunda generación 2G y la recepción de señales de una red de cuarta generación 4G, teniendo en cuenta los estándares TMC y PCS (Servicio de Comunicación Personal). En cuanto a la estación base 2G-GSM se analizaron las funciones principales, mientras que, en relación con la señal recibida 4G-LTE se analizaron los parámetros de recepción.

De acuerdo a la problemática existente, se plantea la siguiente pregunta:

¿Cómo desarrollar prácticas de laboratorio para la implementación de una estación base GSM (2G) y recepción de señales en redes LTE (4G), aplicando SDR?

OBJETIVO GENERAL.

Desarrollar prácticas de laboratorio para los estudiantes de la facultad de ingeniería de telecomunicaciones de la Universidad Santo Tomás con el fin de implementar una estación base de una red GSM (2G) y recibir señales de una red LTE (4G) utilizando radio definido por software.

2.1 Objetivos Específicos.

- Identificar los principales conceptos y características de la telefonía móvil celular, en relación a la evolución en tecnología e infraestructura.

- Establecer el funcionamiento necesario en el dispositivo USRP Ettus N210 para la recepción y transmisión de señales en una red GSM (2G) y recepción de señales LTE (4G).

- Realizar prácticas de laboratorio que permitan evidenciar los conceptos necesarios para el entendimiento de las funciones de una estación base GSM (2G) y la recepción de señal LTE (4G), mediante la implementación de las prácticas de laboratorio.

- Documentar guías de laboratorio con la utilización de radio definido por software.

3. JUSTIFICACIÓN.

Es vital para el ingeniero de telecomunicaciones conocer a profundidad los inconvenientes en infraestructura y tecnología que actualmente se presentan en el servicio de telefonía móvil, es necesario entonces, desarrollar estrategias donde el estudiante encuentre espacios de aprendizaje que lo llevarán a dar soluciones a los inconvenientes manifestados hoy en día a nivel de infraestructura y de esta forma, contribuir en el desarrollo de los servicios de telecomunicaciones.

La telefonía móvil ha experimentado algunas falencias en el servicio durante los últimos años, en sitios de escasa cobertura las señales que provienen de los operadores de telefonía móvil, son de baja potencia o en su defecto no existe nivel de recepción alguno y la conectividad se pierde en su totalidad, esto se debe a procesos de interferencia, capacidad límite de la celda o como se ha evidenciado en el documento inexistencia de infraestructura.

La arquitectura del operador de servicios móviles no cuenta con las herramientas suficientes para llegar a sitios como zonas rurales y explotaciones mineras, que se caracterizan por ser territorios alejados de las instalaciones de un operador de telefonía móvil, además de otros sitios como parqueaderos subterráneos, ascensores y en general, edificaciones robustas (indoor), donde las pérdidas de la potencia de la señal son bastante altas y la conectividad con la red se pierde [4].

Estas limitaciones en tecnología hacen parte de los profesionales en ingeniería de telecomunicaciones, que deben crear nuevas formas de comunicación o mejorar las existentes, sin embargo, es difícil hacerlo sin estar en un marco que permita la vinculación entre la academia y la profesión.

Para establecer en el estudiante de telecomunicaciones conceptos en el área de sistemas de telecomunicaciones móviles, se planteó trabajar en el desarrollo de prácticas de laboratorio con el uso de radio definido por software, donde se implementó una estación base para la tecnología GSM y una práctica en la recepción de señales de la tecnología LTE, y de esta forma el estudiante será capaz de asimilar de forma óptima las arquitecturas comprendidas en un operador de telefonía móvil.

En un entorno académico, las prácticas de laboratorio son el complemento capaz de fortalecer los conocimientos aprendidos en la teoría por parte del estudiante y de esta forma, adquirir nuevas capacidades intelectuales. La realización de las prácticas de laboratorio están proyectadas a la asignatura sistemas de telecomunicaciones móviles y se impactó sobre la preparación del estudiante de ingeniería que curse la asignatura.

4. EVOLUCIÓN DE LA TELEFONÍA MÓVIL.

En este capítulo se presenta una breve descripción de los avances en el campo de la telefonía móvil durante los últimos años. Para luego abarcar los conceptos necesarios en GSM y LTE que dan soporte a la implementación de las prácticas de laboratorio.

En las tres últimas décadas, los servicios de telecomunicaciones han tenido un constante auge. Esto gracias a la masificación de nuevos servicios, con el fin de cubrir la necesidad de la población en comunicación multimedia. En algunos casos, dichos servicios requieren tanto cambios en infraestructuras como la generación de nuevas tecnologías.

De modo que, la telefonía móvil se ha visto forzada a evolucionar desde su creación con lo que denominamos 1G, pasando por 2G y 3G hasta la actual 4G. Esta última generación ha incorporado una serie de tecnologías que permiten que el usuario disfrute de la convergencia de servicios en su dispositivo móvil [5].

El origen de la primera generación 1G se dio en los laboratorios Bell ubicados en Norteamérica, en la década de los 70's. Donde se creó una tecnología basada en celdas utilizando radio para su comunicación, dando lugar a la patente de la primera generación de la telefonía móvil analógica, aunque no fue expandida ni comercializada [6].

Los países desarrollados emprendieron una carrera en la búsqueda del logro de su propia tecnología de comunicación móvil, basada en investigaciones. De este modo para la segunda generación 2G, Estados Unidos adoptó el modelo CDMA mientras el resto del mundo adoptó el modelo GSM [6].

GSM se creó en el año de 1982, en principio como el Grupo Especial Móvil (de sus siglas en ingles GSM), para posteriormente llamarse Global System for Mobile Telecommunications, esta tecnología se explicara en profundidad en 4.1.

La red evolucionó para incorporar el servicio de datos móviles, apareciendo nuevas versiones que eran adaptaciones de GSM, primero GPRS (General Packet Radio Service) y luego EDGE (Enhanced Data Rates for GSM Evolution). La evolución permitía el traspaso de una red con orientación de circuitos, a la modernidad que brinda una red orientada a paquetes. Las velocidades de descarga aumentaron de 9.6 Kbps en GSM a 160 Kbps para GPRS y con este, el servicio de internet era ofrecido a los usuarios [6].

Rápidamente el servicio de datos tuvo acogida en los abonados de los operadores móviles y la tecnología EDGE nació como una alternativa tecnológica para mejorar aún más las velocidades de descarga, ofreciendo 384 Kbps. La ampliación de la velocidad radicó en el sistema de modulación (8PSK) del canal de transmisión y por ende ocurrió una mejora en la eficiencia espectral [6].

Con el objetivo de tener una red globalizada y aceptada mundialmente, se llega al acuerdo de la creación de una nueva generación con un estándar común. Esta tercera generación (3G), fue gestada por el 3GPP (Third Generation Partnership Project) mediante el release (estándar para el avance de la tecnología) 99 (R99). Este release introdujo los conceptos necesarios para el avance a una red de tercera generación, vinculaba la tecnología UMTS (Universal Mobile Telecommunications System) a las redes de telefonía móvil celular. Algunas de las mejoras con respecto a una red GPRS fueron: [7]

- La creación de una infraestructura capaz de ofrecer a los usuarios conexión con los servicios multimedia (Audio, Música y Video).

- El Aumento de la velocidad del enlace descendente a (2 Mbps).
- La adaptación de la tecnología (WCDMA) como método de acceso a la red, de hecho, el cambio de método de acceso al medio ha sido una constante característica en la evolución de las generaciones móviles.
- La interoperabilidad con una red GSM, es decir, los terminales o dispositivos son capaces de acceder e interactuar con las dos redes.
- La incorporación total del servicio de datos móviles y con este internet, al igual que distribuir el servicio a menor costo a comparación de GPRS.

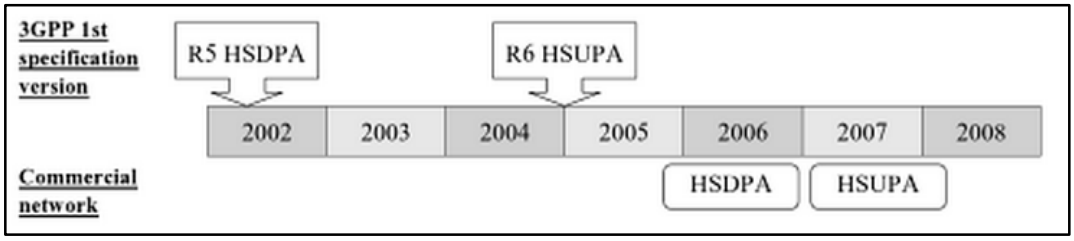
Una serie de releases determinaron los avances tecnológicos que debían ser implementados en las arquitecturas de red, para llegar hasta la actual cuarta generación (4G), estas versiones se describen en los párrafos siguientes.

Por parte de los usuarios, la demanda en relación a la conexión a internet crecía con grandes porcentajes de penetración, de esta manera, los usuarios descargan más información a comparación de la que suministran a la red y este consumo de recursos deriva en latencias en el servicio, por lo que era sumamente importante la mejora de la tecnología en el sentido descendente.

El release cinco (R5) realizó una introducción a una red IMS (IP Multimedia Subsystem) al igual, estandarizó HSDPA (High Speed Downlink Packet Access), donde la velocidad de descarga, es decir, el sentido descendente mejoró de 5 Mbps que ofrece la tecnología UMTS convencional a 10 Mbps en HSPDA [7].

Luego de realizar una mejora al servicio de descarga, se lanzó un release para mejorar la velocidad de subida, el release seis (R6), incorporó dos tecnologías: IMS fase II y HSUPA (High Speed Uplink Packet Access), de esta manera el usuario experimenta velocidades de subida de hasta 5.7 Mbps [7].

En la Figura 1, se observa una línea de tiempo desde la estandarización hasta la ejecución de los releases cinco y seis que incorporaban las tecnologías HSDPA y HSUPA a la tercera generación (3G) de la telefonía móvil.



Fuente: Harri Holma, Antti Toskala.

Figura 1. Tiempo en ejecución en HSDPA y HSUPA.

El release 7 (R7) añade HSPA+, considerado como la introducción a una red de cuarta generación (4G), ya que, las características que fueron estandarizadas en este release son utilizadas en la red LTE, las tecnologías descritas en esta versión son: [8]

- MIMO (Multiple Input Multiple Output): es un gran avance para las telecomunicaciones en general, porque se realizó el procedimiento adecuado para estandarizar un sistema que permite mejorar la velocidad de transmisión utilizando diversidad de espacio, es decir, que se utilizan 2, 4 o 8 antenas según la versión MIMO (MIMO2x2 MIMO 4x4 MIMO8x8) en vez de utilizar solo una antena, lo que deriva en mejorar la velocidad de recepción y permite además la corrección de errores ocurridos en el canal de transmisión. MIMO también es una característica clave para la disminución de retardos en las transmisiones inalámbricas.

- 16QAM (Quadrature Amplitude Modulation) y 64QAM, se estandarizó dos tipos de modulaciones, para la mejora de la eficiencia espectral.

Estas tecnologías permiten que las velocidades se eleven, para downlink se llegó a 42 Mbps y para uplink 12 Mbps [8].

En la Tabla 1**,** se observa la comparación de algunas de las características más importantes en la evolución de las generaciones pertenecientes a la telefonía móvil.

Tabla 1. Comparación generaciones de la TMC.

Estándar	ANCHO DE BANDA (Portadora)	TECNOLOGÍA DE ACCESO	VELOCIDAD DE DATOS (downlink)	GENERACIÓN
GSM	200 KHz	TDMA / FDMA	9.6 Kbps	2G
GPRS	200 KHz	TDMA / FDMA	160 Kbps	2G
EDGE	200 KHz	TDMA / FDMA	384 Kbps	2G
UMTS	5 MHz	WCDMA	2 Mbps	3G
HSDPA	5 MHz	WCDMA	10 Mbps	3G
HSPA+	5 MHz	WCDMA	42 Mbps	3G
LTE	1,4 MHz 3 MHz 5 MHz 10 MHz 15 MHz 20 MHz	OFDMA / SC-FDMA	Hasta 100 Mbps	4G

Es importante esclarecer dos posibles confusiones que pueden generar la Tabla 1:

- En la tecnología de acceso para la segunda generación se presentan dos tecnologías TDMA (Time Division Multiple Access) y FDMA (Frequency Division Multiple Access), en realidad TDMA es la tecnología definida para el acceso al medio en 2G, pero muchos autores convergen en que tanto TDMA como FDMA son utilizadas para este fin. TDMA permite dividir el tiempo en fracciones iguales que son llamados “slots” para asignarle a un usuario el ancho de banda necesario en GSM y FDMA divide el espectro de frecuencias y los asigna a una serie de usuarios para excluir en lo posible la interferencia entre celdas [9].

- La tecnología de acceso al medio para LTE es OFDM (Orthogonal Frequency Division Multiplexing), pero esta tecnología fue dividida en dos clases dependiendo del tipo de enlace en la comunicación, para el enlace ascendente “uplink” la tecnología utilizada es SC-FDMA (Single Carrier Frequency Division Multiple Access), mientras que, en el enlace descendente se encuentra OFDMA (Orthogonal Frequency Division Multiple Access) [10].

Con este marco general, es importante entrar en detalle a las tecnologías objeto, al igual que describir la contextualización que las prácticas de laboratorio van a tener en las arquitecturas de red.

4.1 GSM (Global System for Mobile Telecommunications).

Es un sistema de telefonía móvil diseñado en su totalidad para brindar el servicio de voz, lo que implica estar orientado hacia la conmutación de circuitos. También permite una transmisión de datos, aunque el uso de este servicio es secundario,

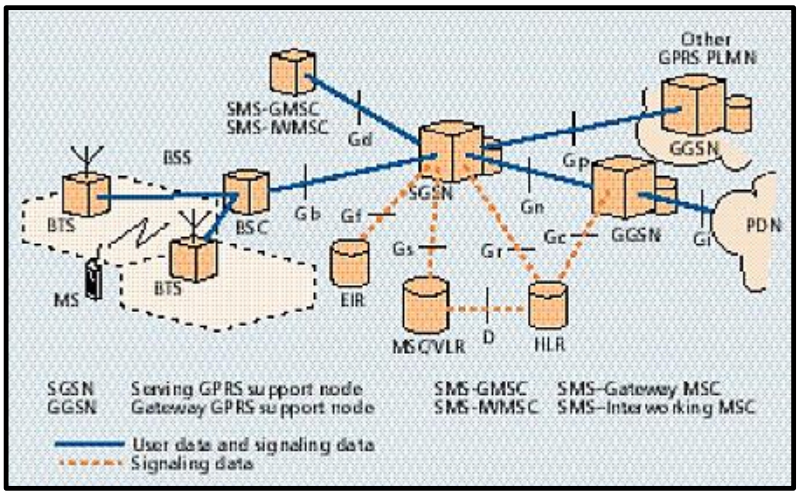
por la baja velocidad que ofrece GSM (9.6 Kbps). De esta manera únicamente se comercializó el servicio de mensajería corta, SMS (Short Message Service), ya que es eficiente por el bajo consumo de ancho de banda para su funcionamiento [11].

Los principales puntos a tener en cuenta para el adecuado desarrollo de las prácticas de laboratorio y su posterior documentación, son la arquitectura de red y el método de acceso al medio.

4.1.1 Arquitectura de red GSM.

En la

Figura 2**,** se evidencia la arquitectura básica de un sistema de telefonía móvil GSM – GPRS [12].



Fuente: C. Bettstetter.

Figura 2. Arquitectura red GSM - GPRS.

La arquitectura ésta conformada en ordenación jerárquica vista desde el usuario hacia la conexión con otras redes por:

Entorno red de acceso, Base Station Subsystem (BSS) [12].

- **Mobile Station (Mb):** es el nombre que recibe el dispositivo móvil para la comunicación vía RF (Radio Frequency).

- **Base Transceiver Station (BTS):** son las estaciones o celdas que brindan la conectividad con un Mb mediante la interfaz *um.*

- **Base Station Controller (BSC):** en este nodo encontramos la controladora de un grupo de estaciones base (BTS), además del control de potencia de los equipos de usuario.

Entorno núcleo de la red, Network Switching Subsystem (NSS) [12].

- **Mobile Services Switching Center (MSC):** es el primer nodo del núcleo de toda la red, se encarga de la asignación de los canales de tráfico (TCH) a los abonados, sincronizando el time slot adecuado para cada comunicación. Además, realiza los procedimientos de enrutamiento.

- **Equipment identity Register (EIR):** es una base de datos que posee interconexión con la MSC, allí se almacena toda la información perteneciente a cada uno de los abonados sobre el estado de sus dispositivos móviles.

- **Home Location Register (HLR):** es una base de datos que al igual que el EIR sostiene conexión con la MSC, la diferencia es que el HLR posee información más concentra acerca de la ubicación de los dispositivos en la red y a que servicios puede acceder el usuario, es decir, datos sobre la capacidad de consumo.

- **Visited Location Register (VLR):** base de datos encargada de registrar el LAC (Location Area Call) de un grupo de usuarios, es decir, la ubicación de cobertura que poseen los usuarios, además, mantiene una copia de la información del HLR mientras que el dispositivo este activo, entre otras funciones.

- **Authentication Center (AuC):** es una base de datos que funciona como un centro de autenticación, la MSC y el HLR consultan permanentemente para conocer los permisos de un usuario cuando este quiere acceder a la red, este procedimiento se realiza mediante la SIM (Subscriber Identification Module), que posee cada dispositivo móvil.
- **(SGSN):** es una puerta de enlace que permite la comunicación entre la BSC y el GGSN, las funciones principales son: la conexión de los dispositivos móviles con la red de datos y realizar el enrutamiento del tráfico de datos.

- **(GGSN):** es el último nodo perteneciente a la red, aquí encontramos una puerta de enlace para la conexión con otro tipo de redes, en especial internet. También posee características de firewall para no permitir el acceso desde otros puntos externos a la red y culmina con algunos procesos de facturación.

4.1.2 Tecnología de acceso al medio (FDMA - TDMA).

Es un tipo de multiplexación, considerada una de las más utilizadas en gran número de las comunicaciones digitales. Como se vio en la sección *evolución de la telefonía móvil,* una portadora para GSM posee un ancho de banda de 200 KHz. FDMA consiste en dividir el espectro asignado a la red GSM en estas portadoras

todas equivalentes a 200 KHz, luego TDMA realiza un proceso similar de división de recursos, esta vez no en frecuencia sino en tiempo, así TDMA es una tecnología que permite fraccionar las portadoras en 8 partes iguales en tiempo con duración de 577 µs y de esta manera cada fracción de tiempo se convierte en un canal de comunicación para ser utilizado por un usuario, estas fracciones de tiempo se conocen con el nombre de time slots [6].

El primer time slot (Ts0), generalmente es utilizado para mensajes de difusión constantes para todos los usuarios BCCH (Broadcast Control Channel), donde viaja información acerca de la red, como el operador al que pertenece, frecuencia utilizada, área de localización del usuario, entre otras. Los demás time slots son utilizados para diversos canales lógicos de comunicación, entre ellos se puede mencionar los más importantes: FCCH (Frequency Correction Channel) canal muy utilizado en procesos de enganche a la red y en handoff, RACH (Random Access Channel) es un canal bidireccional para la autenticación del móvil, PCH (Paging Channel) canal utilizado como alerta de llamada o notificación, SDCCH (Standalone Dedicated Control Channel) es un canal de cuantiosa importancia, ya que, es utilizado para la señalización entre el MS y la BTS y por último, el canal de tráfico TCH (Traffic Channel) [13].

4.2 LTE (Long Term Evolution).

Aproximadamente en el año 2008 se genera el release ocho (R8) por parte del 3GPP, con el cual se estandariza para el mercado la tecnología LTE, que genera un impacto directamente a las tecnologías móviles existentes hasta ese momento.

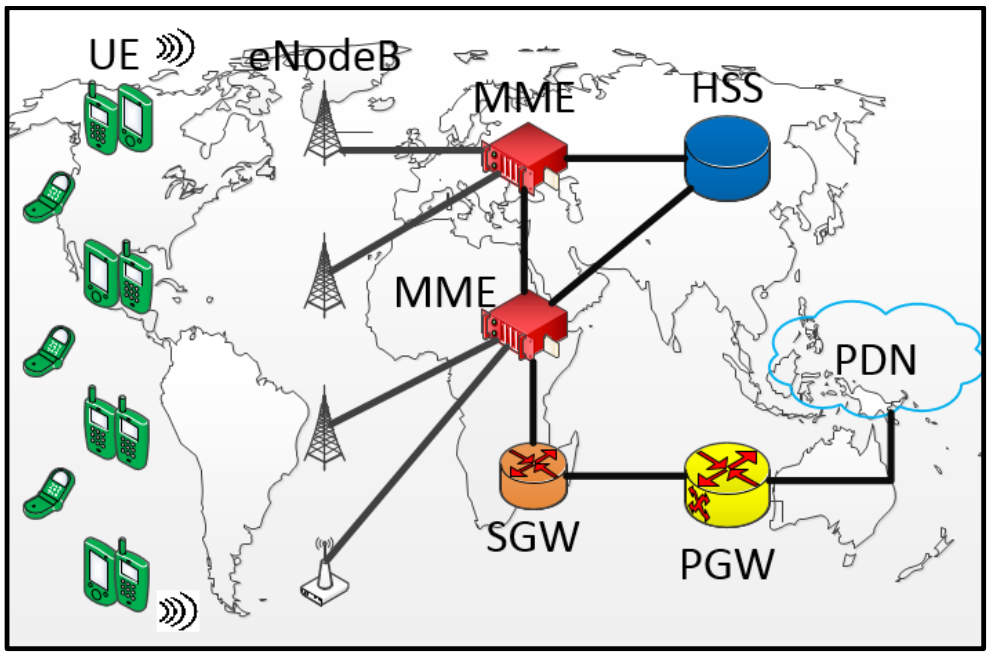
En cuestión de espectro radioeléctrico la nueva estandarización revolucionó la industria de telefonía móvil, ya que, la tecnología de acceso al medio de LTE permite un uso más adecuado de este recurso limitado, utiliza portadoras con

mucho más ancho de banda que además tienen la ventaja de ser variables y de hasta de 20 MHz, incorpora también términos relevantes para el avance de las redes de siguiente generación (NGN), como all-IP y la convergencia de servicios [14].

4.2.1 Arquitectura de red LTE.

En la arquitectura general del estándar LTE, se realizó una reducción en los nodos en la red a comparación de 2G/3G. La red se encuentra fragmentada en dos subsistemas de trabajo, el primero llamado Eutran (Evolved Universal Terrestrial Radio Access Network), que es la subdivisión de la red encargada del acceso y la segunda que hace énfasis en el núcleo de la misma, llamada EPC (Evolved Packet Core). En la

Figura 3, se observa la arquitectura de red de cuarta generación 4G.



Fuente: Propia.

Figura 3. Arquitectura red LTE.

Entorno red de acceso, Eutran [15].

- **User Equipment (UE):** nombre adoptado para el dispositivo móvil en una red LTE.

- **Evolution Node B (eNB):** son las estaciones base que se encargan del manejo de los protocolos para la interfaz de aire (Uu), además unen las funciones de las estaciones base y controladoras diseñadas en la tercera generación (3G), Node B y RNC (Radio Network Controller) respectivamente. Las funciones principales son la asignación de recursos, control de potencia, handoff, señalización, entre otras.

Entorno núcleo de la red, EPC [16].

- **Mobility Management Entity (MME):** es el nodo evolucionado para un SGSN de segunda generación, aunque se agregaron funciones de listado para el seguimiento de la gestión en los abonados y la selección de la puerta de enlace en la conexión con otras redes.

- **Serving Gateway (Serving GW):** es un nodo de gran importancia, ya que, se encarga de enrutador todos los paquetes entre el Eutran y el EPC. Además, al poseer conectividad directa por medio de la interfaz S1-U con los nodos de ENodeB una de sus funciones adicionales es el anclaje del handoff, también cuando este procedimiento ocurre con otra infraestructura distinta a un ENodeB.

Los nodos anteriores poseen la flexibilidad de instalarse de manera conjunta o por separado según la demanda a soportar. En caso de escoger la segunda opción el MME se comunica con la Eutran por medio de la interfaz S1-MME.

- **Packet Data Network Gateway (PDN GW):** es un nodo de mucha importancia para la protección de información interna de la red del operador, ya que, aunque su función principal es la interconexión con redes externas PDN (Packet Data Network), también, posee sistemas como firewall de alta seguridad para la protección de la red.

- **Home Subscriber Server (HSS):** básicamente es la única base de datos en la nueva arquitectura de red, allí se unieron todas las funciones de las bases de datos (HLR, VLR, EiR, AuC), tales como identificación del usuario, estado del dispositivo móvil, información del usuario, entre otras.

4.2.2 Tecnología de acceso al medio (OFDM – OFDMA – SC-FDMA).

Multiplexación utilizada con el fin de reducir los fenómenos de desvanecimiento y atenuación en un canal de comunicaciones. OFDM es una tecnología orientada a los sistemas de comunicaciones inalámbricos y su funcionamiento se basa en el ensanchamiento de la señal en el espectro electromagnético, este procedimiento permite reducir los picos que presenta la señal transmitida sin alterar la potencia de la misma [17].

Aunque OFDM posee varias derivaciones tecnológicas las diferencias son notables en la utilización del espectro, para la cuarta generación (4G-LTE), los métodos de acceso a la red son distribuidos en dos tecnologías OFDMA y SC-FDMA, para el enlace descendente (Downlink) y ascendente (Uplink) respectivamente.

En LTE a diferencia de las otras generaciones, se utilizan dos tecnologías para el acceso al medio. En un principio se utilizó OFDMA tanto para downlink como para uplink, pero se demostró que aunque OFDMA maneja una gran eficiencia espectral, en uplink la utilización de esta tecnología derivaba en el alto consumo de la batería del dispositivo celular, entonces se adecuó SC-FDMA con la característica de reducir el drenaje de la batería y reducir interferencias [18].

OFDMA a diferencia de OFDM realiza una modificación de la señal tanto en tiempo como en frecuencia mientras que OFDM trabaja solo con portadoras en frecuencia, es decir, OFDMA utiliza subportadoras en frecuencia y el acceso a la red en tiempo, un total de doce (12) portadoras con ancho de banda de 15 KHz igual para cada una, cada una de estas portadoras son multiplicadas por el ancho de banda correspondiente y de esta manera se compone un resource block (180 KHz) [19].

Mientras que, SC-FDMA es la tecnología utilizada para la comunicación entre los usuarios y el eNB (Evolution Node B o estación base en LTE) que se conoce como el enlace ascendente, en este caso es utilizada una única portadora que varía conforme el paso del tiempo y su ancho de banda es de 60 KHz [19], allí se utiliza solo una portadora en uplink entre el transmisor y el receptor, por trabajarse como enlace único y por consiguiente no ocurren derivaciones en el espectro.

5. RADIO DEFINIDO POR SOFTWARE ETTUS USRP N210.

El núcleo del proyecto es el dispositivo USRP N210, que puede usarse para finalidades académicas, investigativas, industriales o de defensa [20]. Su fabricante, Ettus Research, se diferencia por la comercialización de sistemas de radio definidos por software de alta calidad. En la Figura 4, se observa el dispositivo en mención y las interfaces que la componen.

Fuente: Propia.

Figura 4. Ettus USRP N210.

Las características principales del Ettus USRP N210, son [20] [21]:

- Permite recibir y transmitir en diferentes tecnologías, ya que tiene la capacidad de variar la banda de frecuencia de trabajo, gracias al cambio de la tarjeta insertada en una de sus ranuras.

- Compatibilidad con otros softwares de simulación como Simulink (de la gama de productos Matlab), LabVIEW y GNURadio.

- Variación de potencia de salida y ancho de banda.

- Capacidad 2x2 MIMO (Multiple Input Multiple Output), lo cual es de vital importancia en recepción a la hora de realizar mediciones en el espectro radioeléctrico y en sistemas multicanales.

- Gran capacidad de procesamiento por la utilización de una FPGA (Field Programmable Gate Array) Xilinx Spartan 3A en su unidad de procesamiento DSP (Digital Signal Processor).

- Posee unidades de conversión digital-analógica (16 bits) y analógica-digital (14 bits).

- Conectividad al ordenador por medio de la interfaz Gigabit Ethernet.

GNURadio es un software libre, que además trabaja con versiones embebidas de Ubuntu (Distribución Linux). Para trabajos que incumban el dispositivo Ettus USRP N210, el proceso de instalación es bastante ágil y eficaz, ya que el sistema operativo y el software pueden ser instalados sobre un mismo ordenador, de esta manera se completa la arquitectura básica para la realización de las prácticas de laboratorio en la Universidad Santo Tomás. Igualmente el dispositivo se encuentra disponible para trabajo independiente dentro de las instalaciones de la Universidad.

5.1 GNURadio.

Esta herramienta de código abierto ha generado un nuevo concepto en el área de sistemas de radio definido por software, el cual es la capacidad de variación hardware-software, es decir, el hardware de GNURadio puede variar sus funciones acorde al paso de tiempo y a la aplicación que se esté utilizando [22].

Para desarrollar aplicaciones en GNURadio se utilizan dos lenguajes de programación, C++ y Phyton, la diferencia radica en el procesamiento que sea necesario para el funcionamiento de la aplicación, mientras que Phyton es utilizado generalmente para aplicaciones de bajo procesamiento que necesiten igualmente de bajos recursos, C++ se utiliza cuando el procesamiento de señal es bastante alto, como en tecnologías en tiempo real [22].

Generalmente GNURadio, se utiliza en ambientes de estudio para el espectro electromagnético en su parte de radio, permite además de recibir estas señales, analizarlas mediante bloques de procesamiento, las señales también pueden ser generadas y transmitidas. Algunos de los principales bloques de hardware disponibles en GNURadio son: filtros, moduladores, demoduladores, codificadores, decodificadores, capas de sincronización, transmisores y receptores, entre otros elementos que componen un radio [22].

Otras de las capacidades de recepción para trabajo con GNURadio, excluyendo GSM y LTE, son [22]:

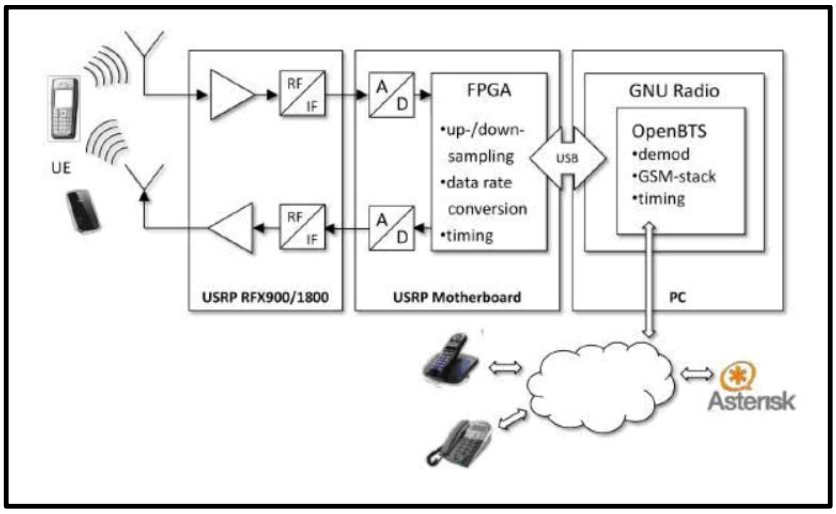
- Televisión Digital.
- Señales de teledetección por satélite.
- Radio.
- GPS (Global Positioning System).

Es importante mencionar la utilización de Asterisk dentro de la arquitectura, ya que esta herramienta en conjunto con protocolo SIP (Session Initiation Protocol), son los encargados de realizar la conmutación y la gestión de las llamadas, para que un usuario GSM sea visto por el sistema como un cliente SIP [23].

SIP es un protocolo utilizado para la señalización en una arquitectura de red telefónica, igualmente para servicios de voz sobre IP (VoIP). Además permite la conexión bidireccional entre los usuarios y la red de telefonía que este en uso [24].

5.2 Herramienta SDR para GSM (OpenBTS).

La primera práctica de laboratorio consistió en implementar una estación base del sistema de telefonía móvil celular de segunda generación (2G), la arquitectura a desarrollar se muestra en la Figura 5.



Fuente: A. Azad.

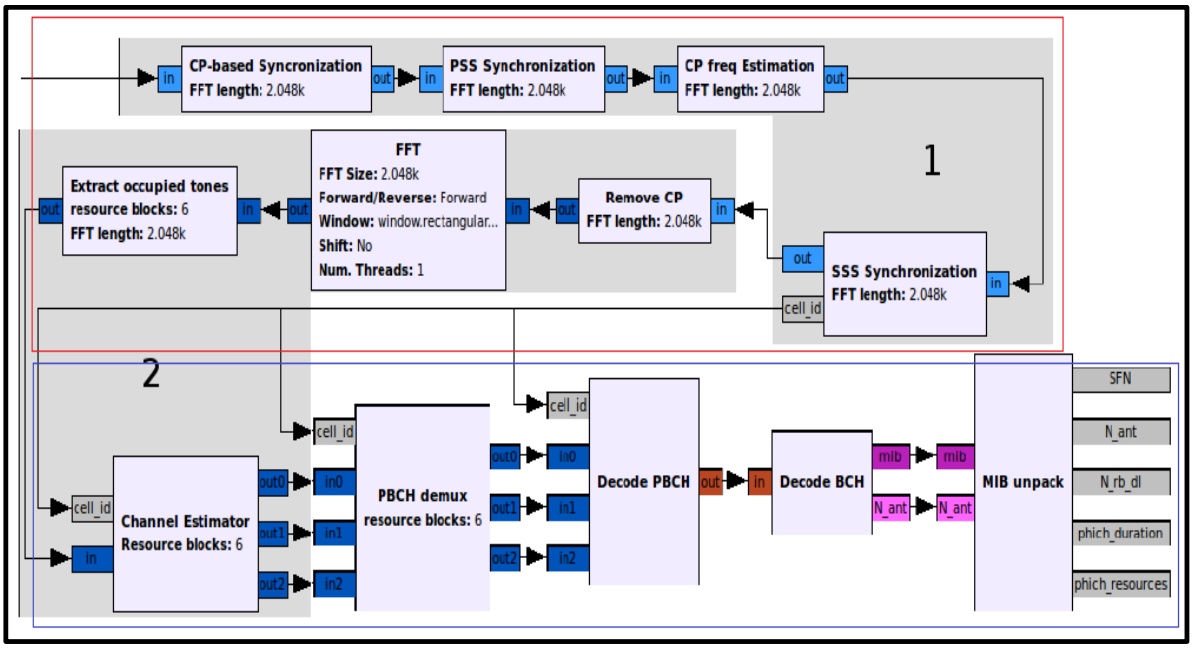
Figura 5. Arquitectura OpenBTS.

Como se puede observar, la arquitectura consiste en: [24]

- Un terminal celular llamado por el estándar, Mobile Station.
- Una interfaz de aire *um.* El dispositivo encargado de recepción dicha interfaz es el USRP N210.
- Una tarjeta hija que se le adhiere al USRP, para que este trabaje en la banda de frecuencia GSM dada para Colombia.
- El dispositivo USRP, con cada una de sus bloques internos previamente descritos.
- En el ordenador se realiza una conexión por cable Gigabit Ethernet para tener conectividad con el dispositivo USRP N210, además de tener el soporte adecuado que brinda la plataforma Asterisk para el control de las llamadas.

5.3 Herramienta SDR para LTE.

La segunda práctica de laboratorio se fundamentó en la recepción de señales de la tecnología LTE de cuarta generación (4G). Mediante unos bloques de procesamiento en GNURadio, se obtendrá la trama LTE para su posterior análisis, la arquitectura a desarrollar se modela en la Figura 6.



Fuente: J. Demel, S. Koslowski, and F. K. Jondral.

Figura 6. Bloques de recepción OFDM – LTE.

En GNURadio, la arquitectura se divide en dos bloques principales de trabajo, el primer bloque, identificado por el recuadro de color rojo que hace referencia a la etapa de sincronización y que además está compuesta por siete bloques de procesamiento. Mientras que, el recuadro de color azul describe los bloques de barrido de la señal, en total cinco [25]. A continuación, se presenta una descripción de la función de los bloques más influyentes de la arquitectura a implementar:

- **CP (Cyclic Prefix)-based Synchronization:** Este bloque permite sincronizar la trama recepcionada con un reloj principal de funcionamiento, este proceso se realiza mediante la detección de un símbolo del reloj, que es correlacionado mediante la siguiente expresión:

$$R_{xx}^{(\mathrm{CP})}[\tau] = \frac{1}{KN_s}\left|\sum_{k=0}^{K-1}\sum_{n=\tau-kN_s}^{\tau-kN_s-N_{\mathrm{CP}}+1} x^*[n]x[n-N_{FFT}]\right|.$$

Donde N_s, es un símbolo OFDM, que consiste en la sumatoria de el componente NCP (número de muestras del bloque CP) y del componente N_{FFT} que hace referencia a la tasa de muestreo [26].

- **PSS (Primary Synchronization Symbol) Synchronization:** Es un bloque de gran importancia, ya que, es el encargado de detectar la señal en el espectro radio eléctrico, además de identificar la estación base transmisora (NID) en conjunto con el SSS.

- **SSS (Second Synchronization Symbol) Synchronization:** Este bloque se encarga de enviar el NID a la segunda etapa de procesamiento, es importante conocer el NID porque así se extrae el código de scrambling, además este bloque termina la sincronización de la trama y ya puede ser enviada.

- **Channel Estimator:** Este bloque realiza una estimación de los parámetros del canal inalámbrico de la señal recibida, de acuerdo a la cantidad de resource blocks de la trama OFDM configurada y así, se puede conocer el ancho de banda de la portadora transmitida.

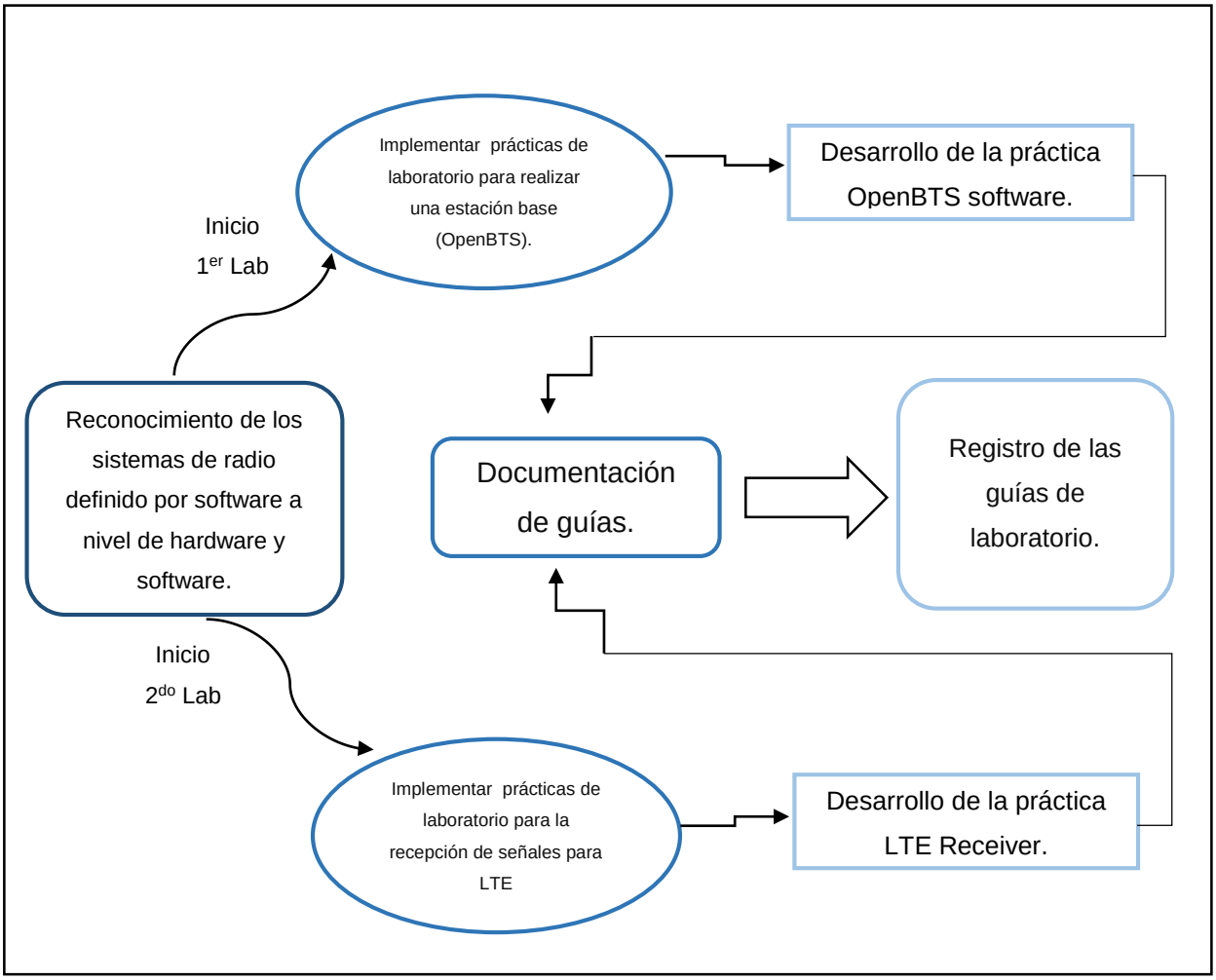
- **PBCH (Physical Broadcast Channel) demux resource blocks, Decode PBCH y Decode BCH (Broadcast Channel):** Es un canal físico PBCH, que transporta los datos del enlace descendente, como información acerca del

estado del enlace. Luego los siguientes bloques organizan la trama y el BCH que es un canal de transporte se encarga de entregar la carga útil.

- **MIB (Master Information Block) unpack.** Es el último bloque de procesamiento y entrega para su análisis la siguiente información: ancho de banda del enlace en Resource blocks y el indicador del estado del enlace en downlink, PHICH (Physical Hybrid ARQ Indicator Channel).

6. METODOLOGÍA.

Para realizar el proyecto se ha diseñado un esquema metodológico ilustrado en la Figura 7, se observa además, las fases que componen el proyecto, así se pueden determinar los pasos a ejecutar a lo largo del desarrollo del mismo con su respectiva retroalimentación cuando sea el caso.



Fuente: Propia.

Figura 7. Diagrama metodología.

- **Reconocimiento de los sistemas de radio definido por software a nivel de hardware y software:** al iniciar el proyecto se realizó un proceso de conocimiento de las herramientas que brinda el radio definido por software**.** Lo que permitió la interacción con el equipo y con todos los sistemas que acompañan el software, enfocando dicho proceso a los sistemas de telefonía móvil 2G.

- **Implementar prácticas de laboratorio para realizar una estación base (OpenBTS):** posterior a la etapa preliminar, se crea un esquema de los elementos físicos que componen la práctica de laboratorio. Así, como un diseño lógico, que permita evidenciar el proceso a llevar a cabo en la implementación de una estación base de segunda generación de telefonía móvil celular.

- **Desarrollo de la práctica OpenBTS software:** finalizado el proceso relacionado con el conocimiento de las herramientas SDR y completado el diseño del esquema a implementar, se procedió a desarrollar las prácticas OpenBTS, con la ayuda de los equipos disponibles en la Universidad Santo Tomás. De esta manera se evaluaron los servicios y la funcionalidad de la estación base.

- **Reconocimiento de los sistemas de radio definido por software a nivel de hardware y software:** finalizada la implementación y el desarrollo de la estación base, se retoma el proceso de conocimiento de las herramientas propias del radio definido por software, esta vez, para la cuarta generación de la telefonía móvil.

- **Implementar prácticas de laboratorio para la recepción de señales LTE:** luego de la culminación de la primera práctica de laboratorio, se procedió a diseñar un esquema que contiene el proceso a seguir para la práctica 4G. Fue de gran ayuda estructurar este proceso para la recepción adecuada de señales LTE y el uso del radio definido por software.

- **Desarrollo de la práctica LTE Receiver:** esta etapa se compone de una serie de procesos para la recepción de señales LTE, que implican el desarrollo del esquema creado anteriormente y puesto en marcha sobre el software GNURadio. Con la gestión del equipo USRP N210, como herramienta de recepción.

- **Documentación de guías:** finalizadas las prácticas de laboratorio que componen el proyecto, se procedió con la documentación de las guías de laboratorio, donde se describió detalladamente paso a paso el proceso que se llevó a cabo para el desarrollo adecuado de las mismas. Sobre formatos modelados según los lineamientos de la facultad de ingeniería de telecomunicaciones.

- **Registro de las guías de laboratorio:** por último, se anexaron las guías de laboratorio en el programa de ingeniería de telecomunicaciones, como método de ayuda académico para la asignatura sistemas de telecomunicaciones móviles y con el fin de apoyar los procesos que se adelantan en el semillero de investigación telesoft.

7. IMPLEMENTACIÓN DE UNA ESTACIÓN BASE GSM (2G) OPENBTS SOFTWARE.

Para la implementación de una estación base GSM (2G), se utilizaron los siguientes componentes a nivel de hardware y software.

7.1 Componentes de Hardware:

- Radio Definido por Software Ettus Research USRP N210.
- Dos antenas VERT900 de la empresa Ettus Research con capacidad de detección dualband en los rangos de 824 a 960 MHz y 1710 a 1990 MHz con una ganancia de 3 dBi.
- Ordenador Samsung Notebook NP530U3C, memoria RAM de 4 GB y procesador Intel(R) Core(TM) i5-3317U CPU 1.7GHZ.
- Terminal iPhone 5s.
- Terminal Motorola moto g.
- Cable Gigabit Ethernet.
- SIM Card Claro en terminal iPhone 5s y SIM Card Movistar en terminal Motorola.

7.2 Componentes de software:

- Sistema Operativo Ubuntu 14.04 LTS instalado en ordenador Samsung.
- Sistema Operativo iOS 8.4 en terminal iPhone 5s.
- Sistema Operativo Android 4.4.4 KitKat en terminal Motorola.
- OpenBTS en sistema operativo Ubuntu 14.04 LTS.
- Asterisk en sistema operativo Ubuntu 14.04 LTS.
- Sipauthserve en sistema operativo Ubuntu 14.04 LTS.

- Smqueue en sistema operativo Ubuntu 14.04 LTS.

Se utilizaron los comandos de color azul en modo administrador para habilitar los permisos en el proceso de descarga e instalación. Para habilitar dichos permisos se digitó el comando **sudo su**, seguidamente se proporcionó la contraseña del usuario.

7.3 Proceso de actualización.

En primera instancia se realizó una actualización y seguido a esto, la descarga de los códigos para su posterior instalación, se utilizaron los siguientes comandos en la consola de Linux/Ubuntu.

Para actualizar el sistema operativo:

- apt-get update
- apt-get upgrade

Luego de actualizar el sistema operativo, se utilizaron los siguientes comandos para la instalación del software **git** que permite generar compatibilidad con el sistema, al igual que con posteriores repositorios y dependencias. Además, de necesitar el software **git** para la descarga del código a instalar posteriormente [27].

- apt-get install software-properties-common python-software-properties
- add-apt-repository ppa:git-core/ppa
- apt-get update
- apt-get install git

7.4 Proceso de descarga.

Posterior a la correcta instalación del software git, se procedió con la descarga del conjunto de códigos que hacen parte de la practica OpenBTS, para esto se utilizó el software git y de esta manera, se clonaron los paquetes del enlace https://github.com/RangeNetworks/dev.git.

- git clone https://github.com/RangeNetworks/dev.git

Este proceso toma algunos minutos y crea una serie de directorios, se ingresa al directorio dev donde se encuentran los archivos necesarios para continuar con el proceso de descarga, se continuó así con los siguientes comandos:

- ./clone.sh
- ./switchto.sh v4.0.0
- ./state.sh
- ./switchto.sh 5.0

Ya que en las instalaciones de la Universidad Santo Tomas se encuentra disponible el equipo USRP N210, la instalación estuvo orientada a la compatibilidad con dicho dispositivo, sin embargo, es posible realizar la práctica con otra gama de productos que ofrece la empresa Ettus Research.

- ./build.sh N210
- ./build.sh N210 openbts
- ./build.sh B100 openbts

El primer comando digitado permite la total descarga del código para su posterior

instalación por lo que toma un tiempo considerable que oscila entre 40 a 60 minutos.

En el proceso de descarga y compatibilidad se exigen los siguientes paquetes previamente instalados (uhd binary, ntp, bind9), se realizó la instalación con los siguientes comandos y se retornó desde el comando ./build.sh N210.

7.4.1 Requisitos previos para la instalación.

- Paquete uhd binary: Para la instalación del paquete uhd binary, se utilizó el siguiente comando, apt-get install libuhd-dev libuhd003 uhd-host y se obtuvo la no compatibilidad con la versión del sistema operativo, sin embargo, fue posible la instalación de este paquete insertando los siguientes comandos: bash -c 'echo "deb http://files.ettus.com/binaries/uhd/repo/uhd/ubuntu/`lsb_release -cs` `lsb_release -cs` main" > /etc/apt/sources.list.d/ettus.list', apt-get update y por último, apt-get install -t `lsb_release -cs` uhd.

- Paquete ntp (Network Time Protocol): Para la instalación de este paquete su utilizó el comando: apt-get install ntp.

- Paquete bind9: Para la instalación de este paquete su utilizó el comando apt-get install bind9.

7.5 Proceso de instalación.

Luego de tener todos los paquetes correctamente instalados en el sistema operativo, se continuó con el proceso de instalación de OpenBTS, Asterisk, Sipauthserve y Smqueue, al igual que para todas las dependencias que hacen

parte de la instalación de los softwares en mención.

Como se muestra en la Figura 8, en el directorio Vídeos/dev/BUILDS/2015-07-08-13-11-53, se encuentran todos los archivos instaladores.

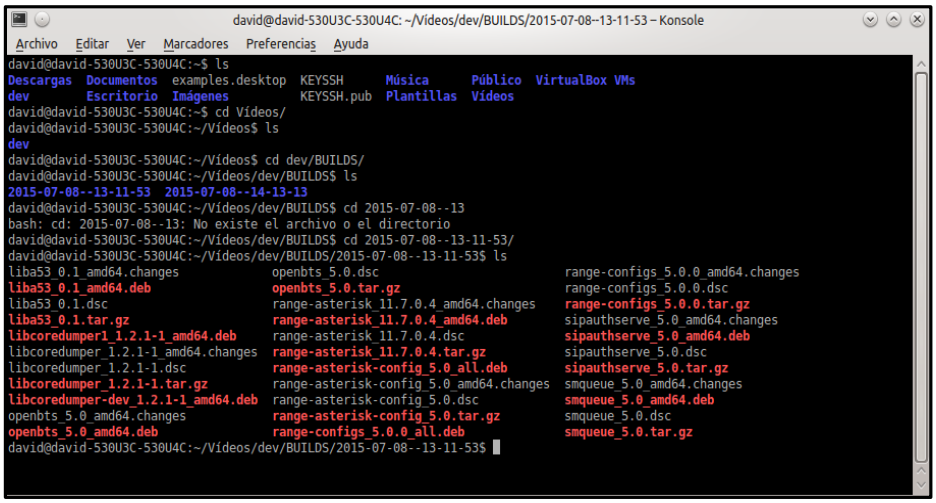


Figura 8. Ubicación archivos de descarga.

Para la instalación de las dependencias se utilizaron los siguientes comandos:

- apt-get install software-properties-common python-software-properties
- apt-get repository ppa:chris-lea/zeromq
- apt-get update
- dpkg -i libcoredumper1_1.2.1-1_i386.deb
- dpkg -i liba53_0.1_i386.deb

De igual forma para la instalación de componentes adicionales se utilizó el siguiente comando:

- dpkg -i range-configs_5.0_all.deb

Finalizado el procedimiento que abarcan las dependencias y los componentes, ya se encuentran habilitados e instalados todos los requerimientos para continuar con la implementación de la estación base.

7.5.1 Instalación de Asterisk.

Asterisk es una herramienta diseñada para el manejo y control de las llamadas en los servicios de VoIP, de esta manera, realiza el cambio del tráfico existente en la red GSM a tráfico de llamadas IP. Para la instalación de esta herramienta se utilizaron los siguientes comandos:

- dpkg -i range-asterisk*.deb
- apt-get install –f

7.5.2 Instalación Sipauthserve.

Servidor encargado de la señalización y el registro de los clientes SIP (Session Initiation Protocol) en la base de datos de la estación base. Se Utilizaron los siguientes comandos para su instalación.

- dpkg -i sipauthserve_5.0_i386.deb
- apt-get install -f

7.5.3 Instalación de Smqueue.

Esta herramienta se encarga del tráfico de mensajes en la red, utiliza el servidor SIP para almacenar y entregar los mensajes al terminal previamente registrado en la red. Los siguientes comandos permitieron la instalación de dicha aplicación.

- dpkg -i smqueue_5.0_i386.deb

- apt-get install -f

7.5.4 Instalación de OpenBTS.

Por último, se procede a instalar la herramienta que genera la interfaz de aire (um) en la estación base, para comunicar la red con los terminales registrados en la misma. Para la instalación de la herramienta madre del proyecto se utilizaron los siguientes comandos.

- dpkg -i openbts_5.0_i386.deb
- apt-get install –f

Así, se finalizó con la instalación de los componentes en software necesarios para la implementación de una estación base GSM (2G). Para verificar que todos los servicios se encontraban habilitados, se pueden activar como procesos en el sistema operativo mediante los siguientes comandos:

- start asterisk
- start sipauthserve
- start smqueue
- start openbts

7.6 Proceso de verificación de los servicios.

Una vez los servicios son activados, de manera gráfica se puede verificar su estado, insertando el comando ps –e en la consola, ya que estos servicios son procesos activos en el sistema operativo Ubuntu.

7.6.1 Proceso Asterisk.

En la Figura 9, se observa que el servicio Asterisk se encuentra dentro de los procesos activos en el sistema, como el proceso número 2792.

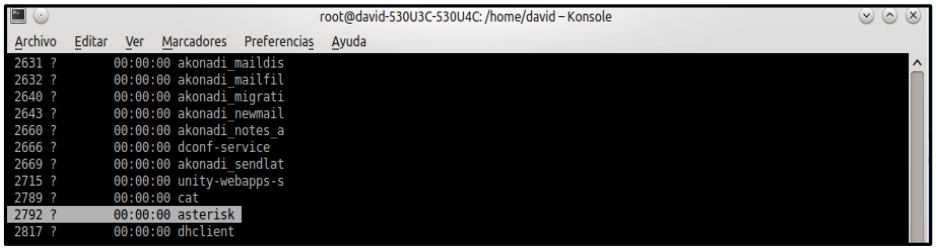


Figura 9. Verificación Asterisk como proceso en el sistema.

7.6.2 Proceso Sipauthserve.

En la Figura 10, se observa que el servicio Sipauthserve se encuentra dentro de los procesos activos en el sistema, como el proceso 1551.

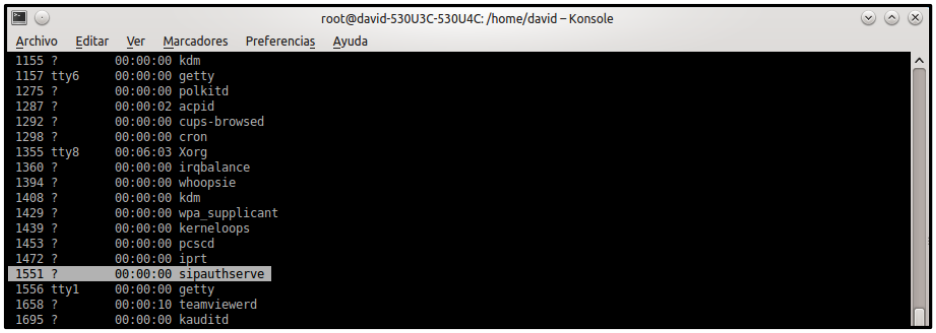


Figura 10. Verificación Sipauthserve como proceso en el sistema.

7.6.3 Proceso Smqueue.

En la Figura 11, se observa que el servicio Smqueue se encuentra dentro de los procesos activos en el sistema, como el proceso 2877.

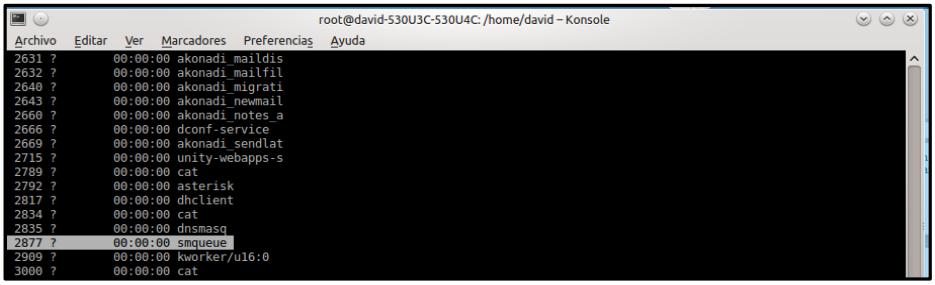


Figura 11. Verificación Smqueue como proceso en el sistema.

7.6.4 Proceso OpenBTS.

En la Figura 12, se observa que el servicio OpenBTS se encuentra dentro de los procesos activos en el sistema, como el proceso 3150.

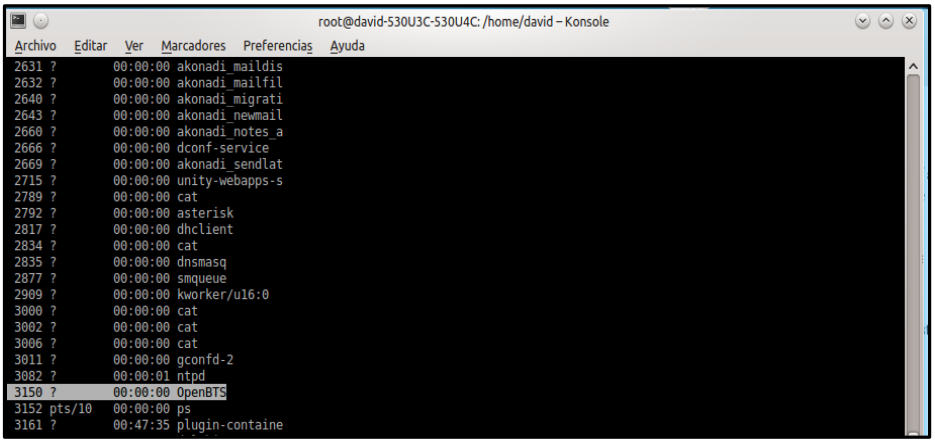


Figura 12. Verificación OpenBTS como proceso en el sistema.

7.7 Verificación de conectividad vía puerto Gigabit Ethernet.

Fue necesario detener todos los procesos para verificar la conectividad, se utilizaron los siguientes comandos:

- stop asterisk
- stop sipauthserve

- stop smqueue
- stop openbts

Una vez las herramientas de software fueron correctamente instaladas, es posible realizar la conexión del ordenador con el radio mediante el puerto Gigabit Ethernet, fue necesario digitar el paquete uhd binary que se instaló anteriormente y utilizar los siguientes comandos para verificar la conectividad:

- uhd_find_devices

Este comando permite identificar la dirección IP por defecto y el serial del equipo desde su fabricación, tal como se observa en la Figura 13.

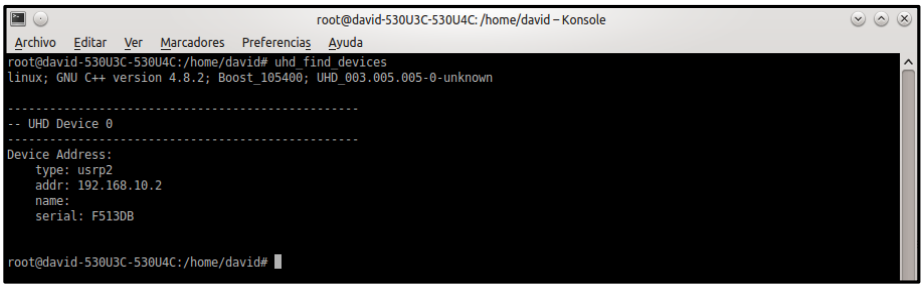


Figura 13. Dirección IP del USRP por defecto en consola.

- uhd_usrp_probe

Este comando es utilizado para consultar información más específica y detallada de las características que posee el equipo, tal como se observa en la Figura 14.

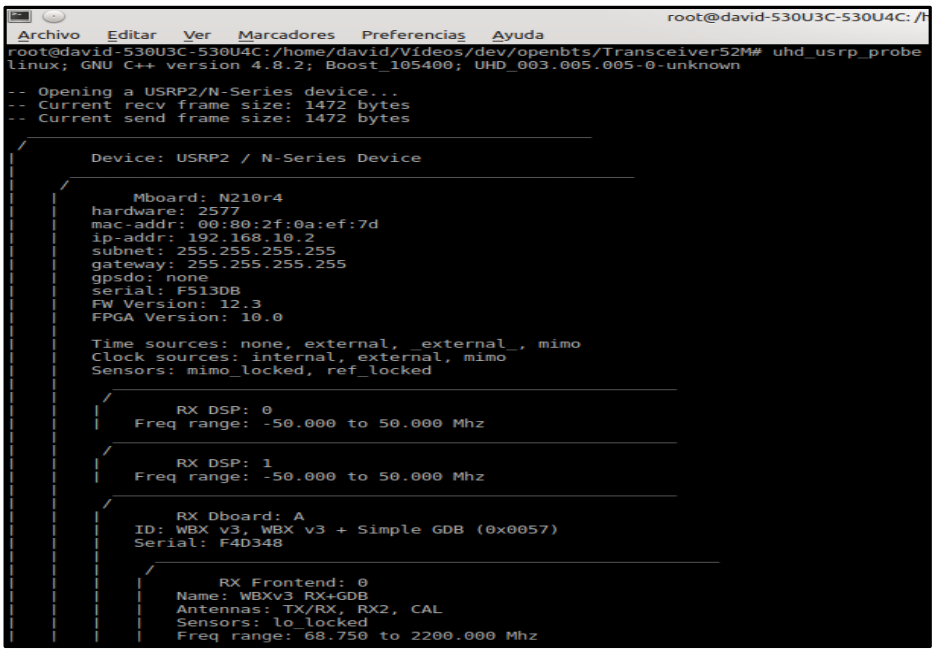


Figura 14. Características USRP en consola.

Finalmente en la Figura 15, se utilizó el comando (./transceiver) para verificar la detección del radio en el ordenador. Este comando se puede utilizar únicamente al ingresar al directorio /dev/openbts/Transceiver52M.

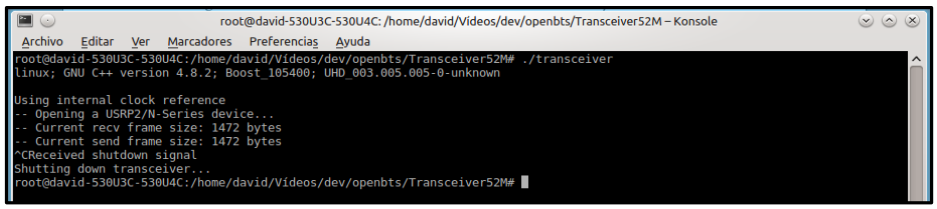


Figura 15. Detección del USRP en el ordenador.

7.8 Prueba de conectividad.

Utilizando el comando ifconfig eth0 192.168.10.1/24, se cambió la dirección IP del ordenador para añadirlo a la subred que tiene el dispositivo Ettus por defecto.

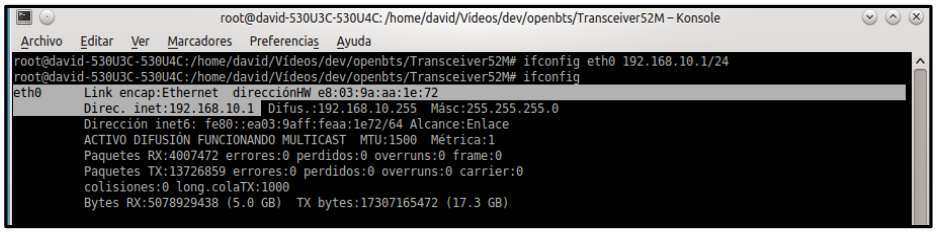


Figura 16. Cambio dirección IP puerto Gigabit Ethernet del ordenador.

Se debe tener en cuenta que el puerto eth0 puede variar según el ordenador utilizado, para verificar el puerto habilitado se utilizó el comando ifconfig. En la Figura 16, se muestra el cambio de dirección IP en el puerto eth0.

Finalmente en la Figura 17, se observa que para efecto de comprobación de conectividad con el equipo, un ping fue enviado de forma satisfactoria a la dirección IP 192.168.10.2 que posee el radio por defecto, con el comando ping 192.168.10.2.

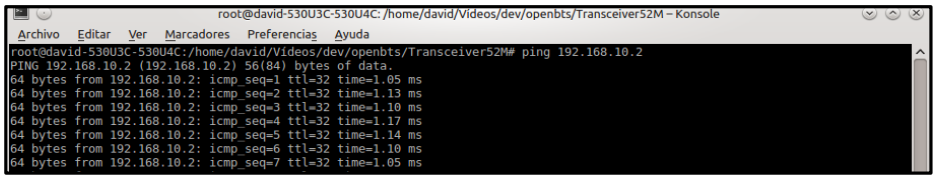


Figura 17. Prueba de conectividad con el equipo USRP.

7.9 Puesta en marcha de la estación base y configuración parámetros iniciales.

Luego que las pruebas de conectividad son exitosas, todos los elementos de hardware y software fueron correctamente instalados y conectados entre sí. El primer paso para la puesta en marcha de la red es activar las herramientas previamente instaladas (asterisk, sipauthserve, smqueue y openbts).

En el directorio raíz se encuentra añadido otro directorio llamado /OpenBTS, se ingresó a este directorio y se digitó el siguiente comando (./OpenBTSCLI).

OpenBTS es una herramienta que posee su propia consola predeterminada (CLI) Command Line Interface, donde es posible configurar los parámetros de la red. En la Figura 18, se observa dicho proceso.

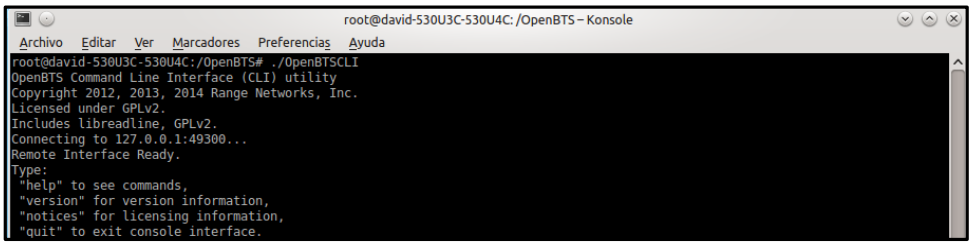


Figura 18. Ingreso a CLI de OpenBTS.

Los comandos de color verde son utilizados en el CLI de OpenBTS.

Para consultar la información de la red por defecto se ingresó el comando (config GSM.Radio), En la Figura 19, se observa dicho proceso.

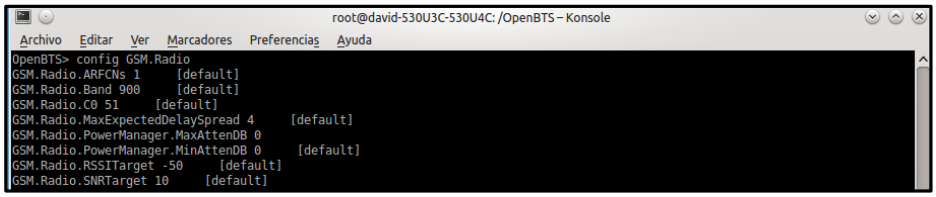


Figura 19. Configuración inicial de la estación base.

Igualmente en la Figura 19, se observa los parámetros básicos que limitan la red, los dos de mayor importancia son:

- Banda de frecuencia: En Colombia las bandas de frecuencia disponibles para el servicio de telefonía móvil celular GSM (2G) son 850 MHz, 900

MHz, 1800 MHz y 1900 MHz. Fue importante seleccionar una adecuada banda de frecuencia para disminuir el ruido en el canal de transmisión.

- ARFCN (Absolute Radio Frequency Channel Number) y C0: es el canal inalámbrico de la banda seleccionada que será ocupado en el momento de realizar las transmisiones.

Para cambiar la configuración de estos parámetros no es necesario salir del CLI de OpenBTS, pero si es necesario reiniciar el proceso, en la Figura 20 y Figura 21, se observa la aplicación de los comandos en la configuración del radio para el cambio de banda de frecuencia y para el cambio del canal inalámbrico a utilizar respectivamente.

- Para realizar el cambio de la banda de frecuencia se utilizó el comando (config GSM.Radio.Band) seguido de la banda a la que se quiere migrar, (config GSM.Radio.Band 850).

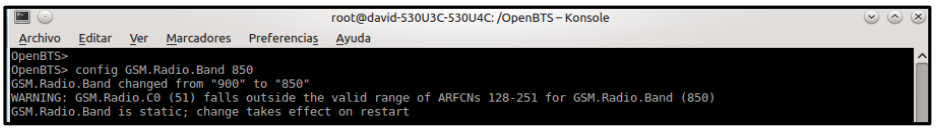


Figura 20. Configuración banda de frecuencia en consola.

La configuración cambia satisfactoriamente, pero informa al usuario que hay dos peticiones pendientes para el cambio de banda de frecuencia. La primera radica en la utilización de un canal comprendido entre 128 a 251 (ver próxima viñeta) y la segunda el reinicio de la consola openbts para que el cambio sea exitoso.

- El cambio del canal inalámbrico de transmisión, se realizó mediante el comando (config GSM.Radio.C0) seguido del canal que se pretende utilizar, (config GSM.Radio.C0 166).

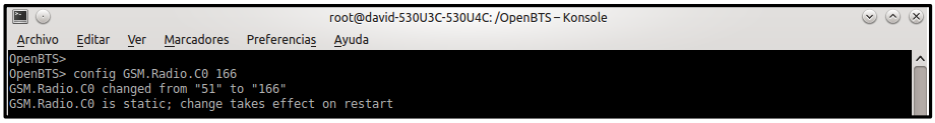


Figura 21. Configuración canal inalámbrico en consola.

Igualmente en el cambio del canal, para subir la configuración al proceso Openbts, la consola realiza petición de reinicio.

Para reiniciar la consola y subir las configuraciones establecidas al sistema se utilizaron los siguientes comandos: quit, stop openbts, start openbts y finalmente, ./OpenBTSCLI. El proceso se observa en la Figura 22.

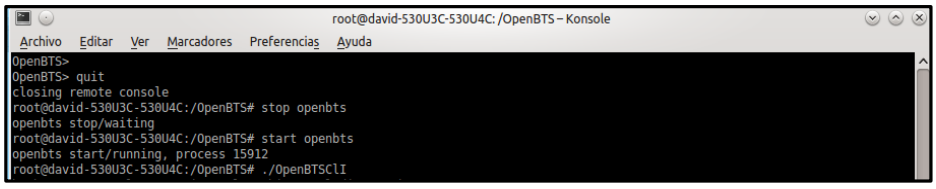


Figura 22. Proceso de reinicio de consola OpenBTS.

A continuación, se debe verificar el ruido existente en la interfaz de aire para esto se utilizó el comando (noise), si los niveles de ruido no son aceptables, es posible que las antenas del radio no se encuentren en posición adecuada. Para corregir el posicionamiento de las antenas y sostener niveles adecuados en la transmisión, fue necesario que las antenas formen 90° grados de ángulo entre sí. En la Figura 23, se observa los niveles de ruido en la interfaz de aire.

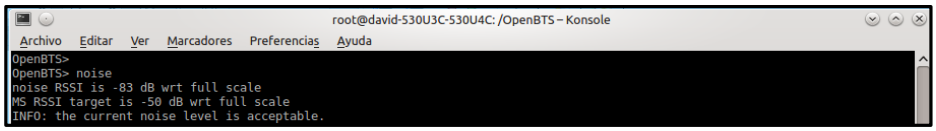


Figura 23. Ruido en la interfaz de aire en consola.

Para incrementar la potencia de la portadora o consultar su estado, se utilizó el comando (power) seguido del nivel que se considere sea necesario, (power 40). El

nivel de potencia posee un rango de 0 a 80 dB. En la Figura24, se observa la configuración de la potencia de la señal.

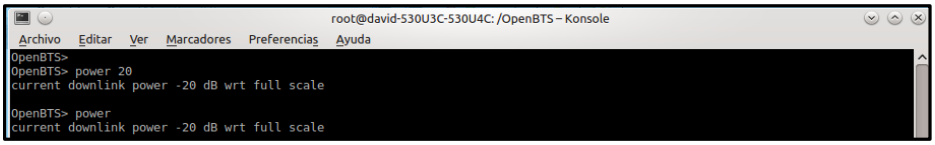


Figura 24. Configuración potencia de la portadora en consola.

7.10 Capturas analizador de espectro.

De acuerdo a la configuración observada en la Figura 20 y en la Figura 21, la banda de frecuencia utilizada para la comunicación entre los terminales y la estación base es 850 MHz, con portadora ubicada en canal 166, para esta configuración la portadora equivalente se observa a continuación en la Figura 25.

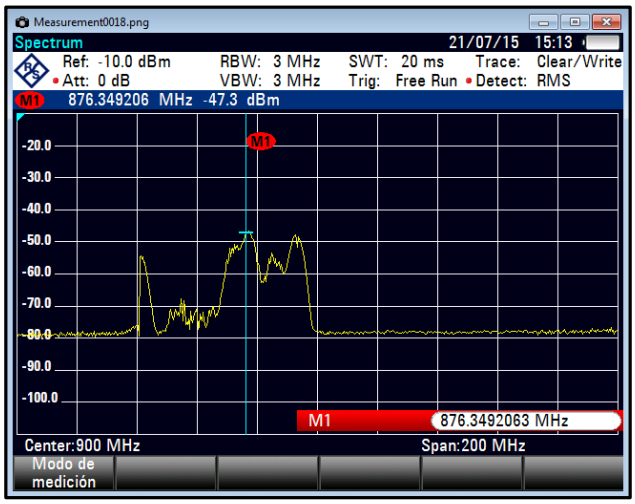


Figura 25. Portada en banda de frecuencia de 850 MHz y canal en 166.

Igualmente en la Figura 25, se observa que en la banda de 850 MHz existen otras portadoras de igual o menor potencia, sin embargo, afectan la señal derivando en atenuaciones a la portadora que genera la estación base, por esta razón, se

realizó un cambio en la banda de frecuencia a 900 MHz, como se observa en la Figura 26.

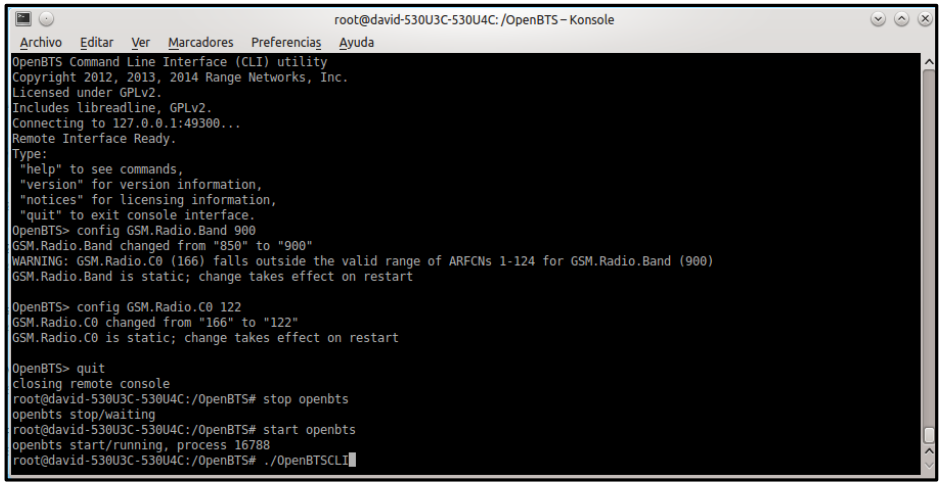


Figura 26. Cambio de banda de frecuencia a 900 MHz en consola.

El cambio de banda de frecuencia, implica también un cambio en la canalización que está en el rango de 1 a 124 para la banda seleccionada, el cambio de canal se puede observar en la Figura 26.

Como se observa en la Figura 27, se reinicia el servicio OpenBTS para guardar la configuración anterior.

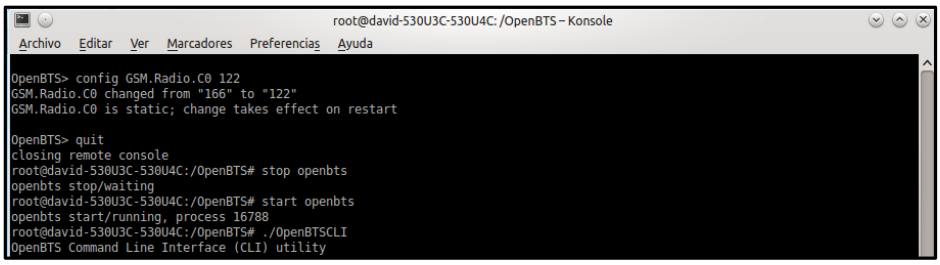


Figura 27. Reinicio del servicio OpenBTS en consola.

Los cambios anteriores en la consola OpenBTS generan un traslado a una banda más limpia, este concepto puede ser evidenciado en la Figura 28.

Figura 28. Portadora en banda de frecuencia de 900 MHz en analizador de espectro.

A continuación se aumenta la potencia de la portadora de un nivel de 20 dB a 0dB, en la Figura 29 se observa el cambio de potencia.

Figura 29. Portadora en banda de frecuencia de 900 MHz con 0dB de potencia.

Finalmente en la Figura 30, se observa la ubicación de la portadora en canal 122 y banda de frecuencia 950 MHz, es decir con un corrimiento de señal con frecuencia central de 959.4 MHz y potencia de 0 dB. Igualmente se puede apreciar el ancho de banda de la señal.

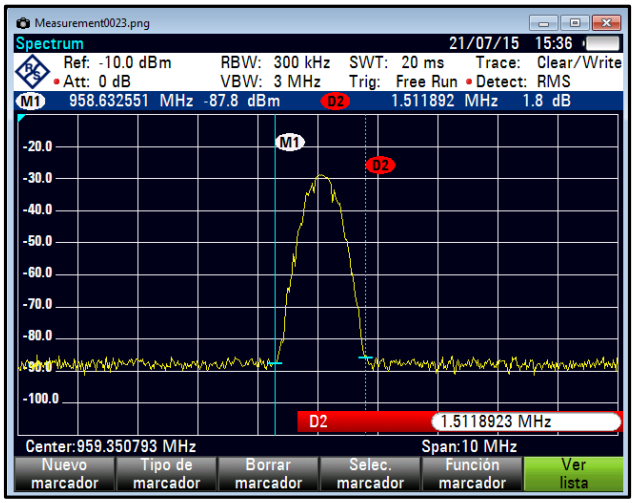


Figura 30. Ancho de banda portadora en banda de frecuencia de 900 MHz.

7.11 Búsqueda de la red en los terminales.

Posterior a la configuración de la banda de frecuencia y la canalización de la portadora, es posible buscar la portadora como red en los terminales, los pasos para el logro de este objetivo difieren según el sistema operativo del terminal, a continuación se hace la demostración del proceso de búsqueda para terminales iOS y Android.

7.11.1 Terminales sistema operativo iOS.

En la Figura 31, se observa la manera adecuada para la selección de red. Los pasos para encontrar la red **Test PLMN 1-1** en un terminal con sistema operativo iOS son:

- Ingresar a ajustes.
- Seleccionar la barra de operadores.
- Deshabilitar la opción de operador automático, inmediatamente el terminal buscará las redes disponibles.
- Seleccionar la red **Test PLMN 1-1.**

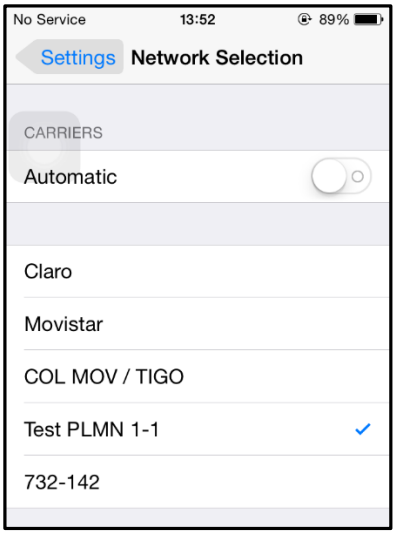


Figura 31. Búsqueda de la red Test PLMN en iOS.

7.11.2 Terminales sistema operativo Android.

En la Figura 32, se observa la manera adecuada para la selección de red. Los pasos para encontrar la red **Test PLMN 1-1** en un terminal con sistema operativo Android son:

- Ingresar a ajustes o configuraciones.
- Seleccionar la barra mas.
- Seleccionar la barra de redes móviles.
- Seleccionar la barra de operador de red.

- Automáticamente el terminal realiza la búsqueda de las redes disponibles, en caso contrario seleccionar la barra buscar redes.
- Seleccionar la red **Test PLMN 0001-01.**

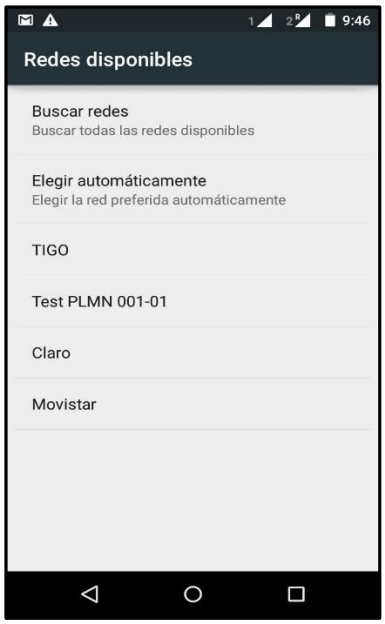


Figura 32. Búsqueda de la red Test PLMN en Android.

Una vez los terminales seleccionen la red, con el comando (tmsis) en el CLI de OpenBTS, estos pueden ser vistos por el usuario de la estación base, como se observa en la Figura 33.

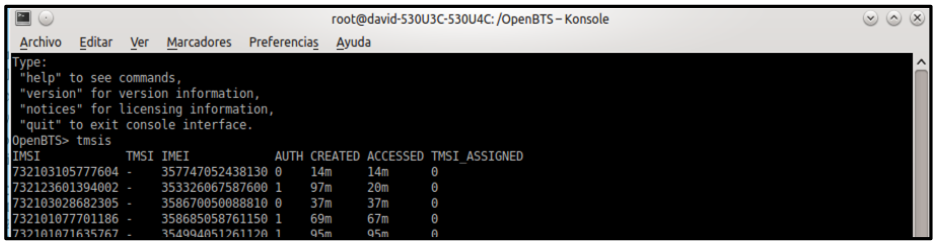


Figura 33. Información del comando tmsis en el CLI de OpenBTS.

La información suministrada por el comando tmsis en el CLI de OpenBTS, índica

las siguientes características del terminal.

- **IMSI (International Mobile Subscriber Identity):** es un número que posee una extensión de 14 dígitos asociado a la SIM Card insertada en el terminal. Esta numeración indica el código del país, código del operador y el número de identificación de suscripción del móvil (por sus siglas en inglés MSIN).

- **IMEI (International Mobile System Equipment Identity):** es un número único internacional que identifica el terminal, posee una longitud de 15 dígitos. El último dígito es cambiado por un cero en la configuración predeterminada de OpenBTS.

- **Authentication:** es un número binario, índica en caso de ser '0' que el terminal no está registrado en la base de datos, en caso contrario '1' el terminal ya está registrado y conectado exitosamente a la red.

Luego de que el terminal seleccione la red intentará registrarse en este, pero solo hasta hacer el proceso de registro en la base de datos de OpenBTS, tendrá conexión satisfactoria con la red. Es decir, para que un terminal sea aceptado por la estación base, este debe estar previamente registrado y se puede identificar cuando aparezca con '1' en la columna de Authentication. Este proceso se puede observar detalladamente en el siguiente apartado Adición de un terminal a la base de datos.

Para verificar que el terminal conectado con la estación base sea el que el usuario está intentando registrar en la red, se digitó en el teclado del terminal +#06#, así se conoce el IMEI del teléfono y debe coincidir con el IMEI del comando tmsis

insertado en el CLI de OpenBTS. En la Figura 34, se observa el IMEI corresponde al IMSI 732101077701186 de la Figura 33.

Figura 34. IMEI del terminal iPhone.

7.12 Adición de un terminal a la base de datos.

Este proceso es sencillo, bastó con conocer el IMSI del terminal y asociarle un número telefónico que puede ser el que comúnmente utiliza. Así con el comando ./nmcli.py sipauthserve subscribers créate “nombre” IMSI ”imsi del terminal” \”número telefónico”.

Para que el comando tenga efecto se debe ingresar al directorio /dev/openbts/NodeManager. En la Figura 35, se observa la aplicación del comando en consola, de igual manera el mensaje que retorna cuando el terminal se registra satisfactoriamente.

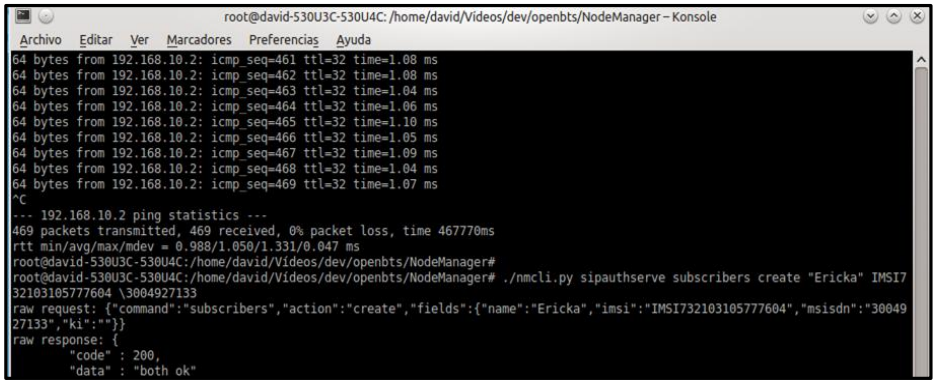


Figura 35. Comando para agregar un terminal a la base de datos en consola.

Luego de registrar el usuario en la base de datos, se realizó de nuevo los pasos de la sección Búsqueda de la red en los terminales. Adición de un terminal a la base de datos. Así, el terminal conectó definitivamente con la estación base y obtuvo disposición para recibir y enviar mensajes de texto y llamadas de voz.

En la Figura 36. Conexión establecida de terminal con red Test PLMN 1-1., se observa en la parte superior izquierda, en la barra operador que el terminal está conectado y tiene plena cobertura con la red Test PLMN 1-1.

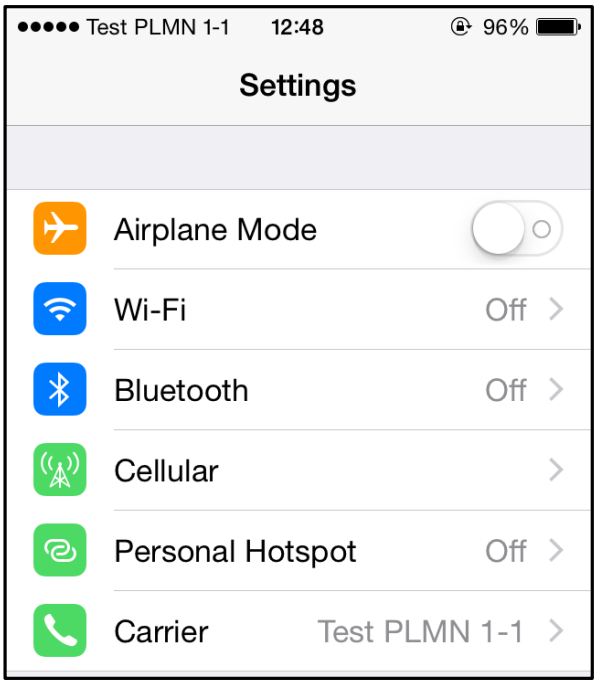


Figura 36. Conexión establecida de terminal con red Test PLMN 1-1.

7.13 Servicio de mensajería de texto.

Después de todos los procesos de instalación y conectividad entre los terminales y la red, finalmente se pueden aplicar servicios de telecomunicaciones. El primer servicio es el envío de mensajería de texto entre la estación base y un terminal.

7.13.1 Mensaje de texto como prueba de conectividad desde el terminal hacia la estación base.

Para verificar que el terminal obtuvo el servicio de mensajería de texto activo, se realizó un envío de mensaje al número 411, la estación base retorno un mensaje de acuso que permite hacer veraz la comunicación, esta prueba se puede observar en la Figura 37.

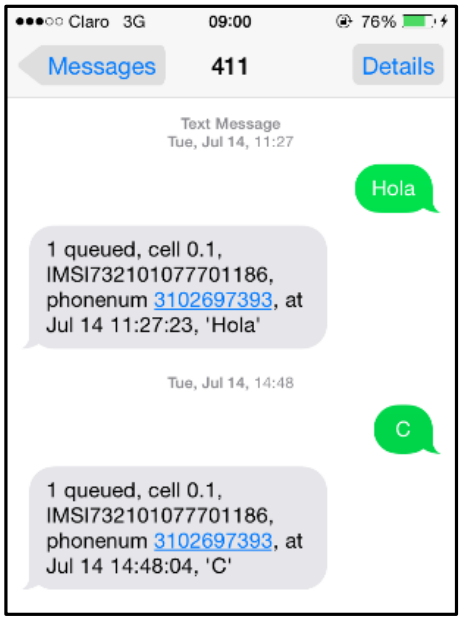


Figura 37. Mensaje como prueba de conectividad enviado desde el terminal.

7.13.2 Mensaje de texto desde la estación base hacia el terminal.

En el CLI de OpenBTS, es posible digitar el siguiente comando sendsms “IMSI del terminal” “número del remitente” “Mensaje a enviar”. Para enviar un mensaje de texto corto al terminal que se desee, desde que este esté registrado previamente en la base de datos de la estación base.

En la Figura 38, se observa la aplicación correcta del comando sendsms 732101077701186 1234567 Mensaje de texto desde la consola al terminal.

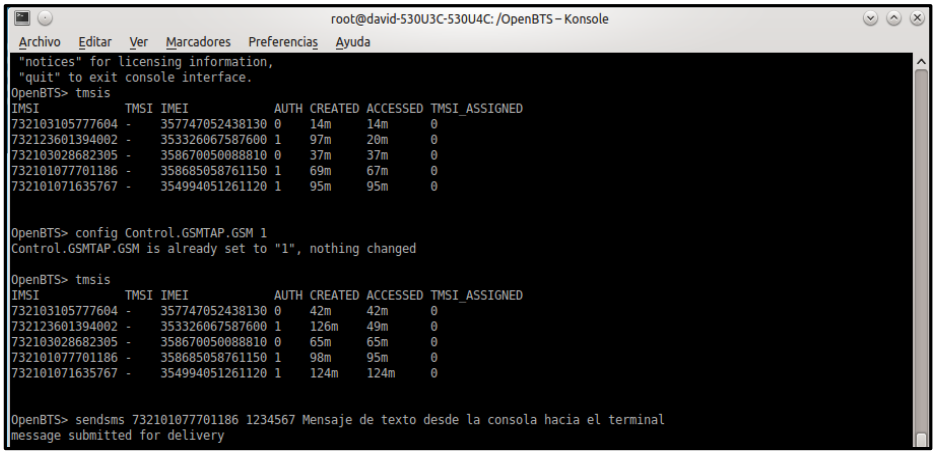


Figura 38. Mensaje de texto desde CLI de OpenBTS hacia terminal.

En la Figura 39. Mensaje de texto satisfactorio desde CLI de OpenBTS hacia terminal., se observa la entrega satisfactoria del mensaje de texto al terminal asociado al IMSI 732101077701186 desde el número escogido como remitente. Igualmente se observa que el número máximo de caracteres permitidos es de 40.

Figura 39. Mensaje de texto satisfactorio desde CLI de OpenBTS hacia terminal.

Finalmente en la Figura 40, se observa los eventos a razón de mensajería de texto que ocurrieron con éxito en el CLI de OpenBTS, para activar esta función se insertó el comando stas SMS.

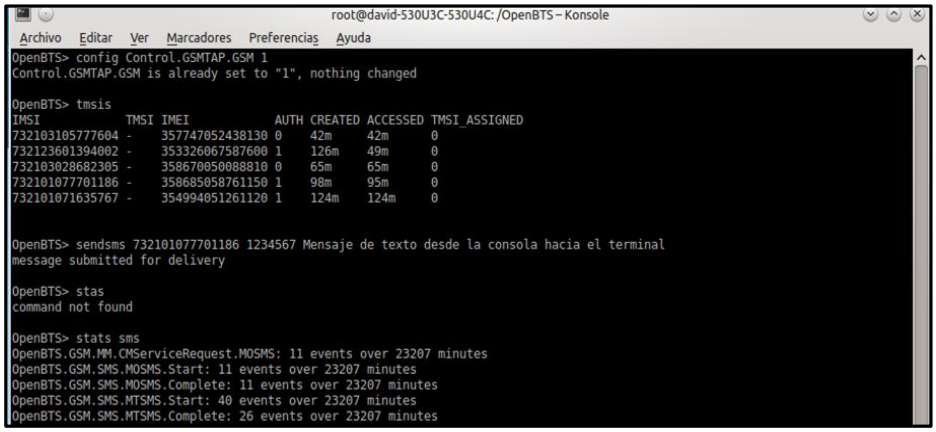


Figura 40. Eventos de mensajería de texto en el CLI de OpenBTS.

7.14 Servicio de llamadas de voz.

Para realizar una llamada de un terminal hacía la estación base, es necesario ingresar al teclado de llamadas y digitar el número 2602, así se obtiene comunicación con la estación base y esta retorna al terminal el mismo tráfico enviado. También se puede llamar al número 2600 y la estación base esta vez retorna un sonido continuo como prueba del servicio de llamadas de voz activo.

En la Figura 41, se observa que hay un canal de tráfico TCH (Traffic Channel) que se encuentra en uso, que es ocupado al momento de realizar un llamado al número 2602, igualmente se observan parámetros de transmisión y finalmente el IMSI del terminal que efectuó la llamada.

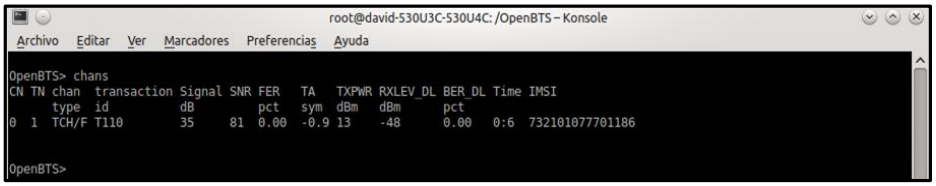


Figura 41. Canal de tráfico activo en un llamado de voz.

7.14.1 Servicio de llamada de voz entre dos terminales.

Una vez mínimo dos terminales se encuentren registrados en la base de datos de OpenBTS y estén conectados a la red, pueden soportar el servicio de llamadas de voz entre sí. Digitando el número con el cual fue registrado en la base de datos en el panel de llamada, es posible efectuar la comunicación entre ellos.

En la Figura 42, se observa que hay dos canales de tráfico ocupados en la estación base, de esta manera se comprueba que la llamada ha sido satisfactoria.

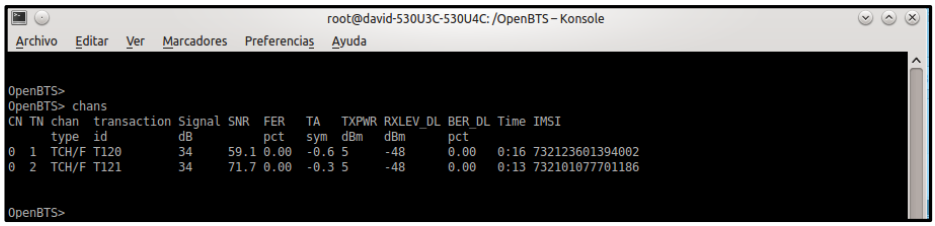


Figura 42. Canales de tráfico activos en llamada de voz.

Finalmente en la Figura 43, se utilizó el comando (stats GMS) con el fin de observar los eventos a nivel de llamadas de voz ocurridos en la estación base.

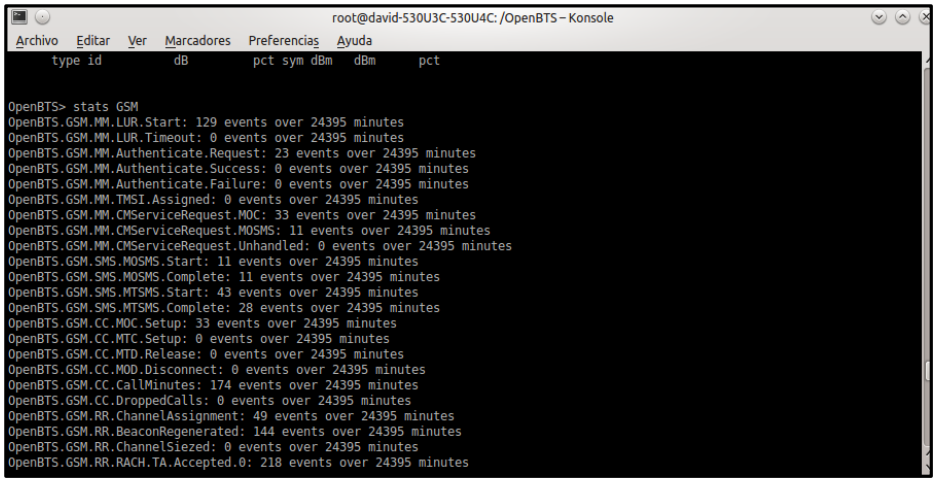


Figura 43. Eventos de llamadas de voz en el CLI de OpenBTS.

7.15 Capturas de tráfico en el software Wireshark.

Al momento de realizar la práctica de laboratorio paralelamente, se corrió el software Wireshark, el cual permite capturar el tráfico que fluye en el puerto Gigabit Ethernet del ordenador donde se encuentra corriendo el proyecto OpenBTS. A continuación se muestran los protocolos abarcados en cada uno de los procesos de señalización y tráfico.

En la Figura 44, se observa el proceso de señalización que se establece con la utilización del protocolo SIP, además, se observa el terminal fuente que corresponde al IMSI 732101077701186 también observado en la Figura 42, utilizando el canal de tráfico.

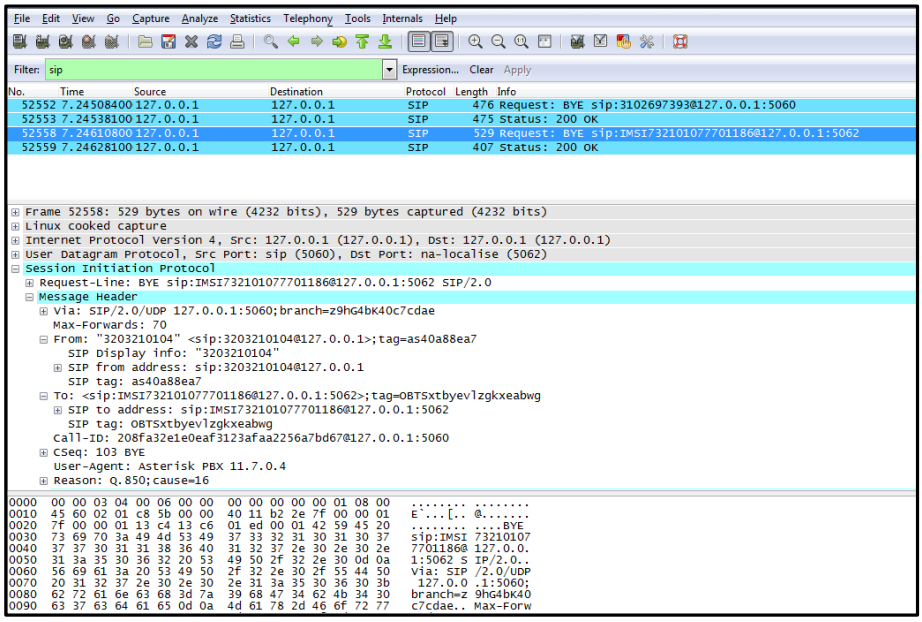


Figura 44. Captura protocolo de señalización SIP.

A continuación, se observa la verificación del canal seleccionado para la transmisión de datos en la estación base, el cual fue el canal 121 con frecuencia

central de 959.2 MHz en enlace descendente y 914.2 MHz para enlace ascendente. Así, la Figura 45, permite observar igualmente la utilización del protocolo gsmtap en la estación base. El protocolo gsmtap encapsula tramas GSM procedentes de la interfaz de aire (um).

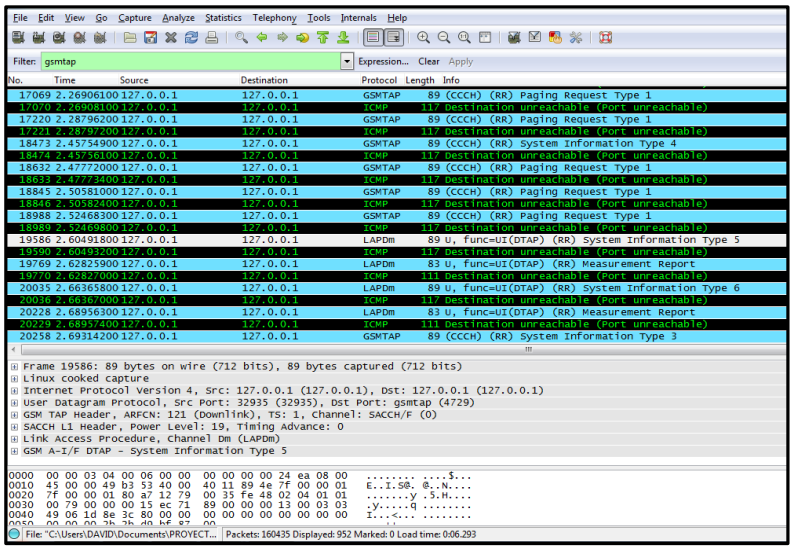


Figura 45. Captura protocolo de cabecera GSM gsmtap.

Por último, en la Figura 46, se observa el uso del protocolo de transmisión UDP (User Datagram Protocol) y las direcciones Ipv4 correspondientes al ordenador y al equipo USRP N210.

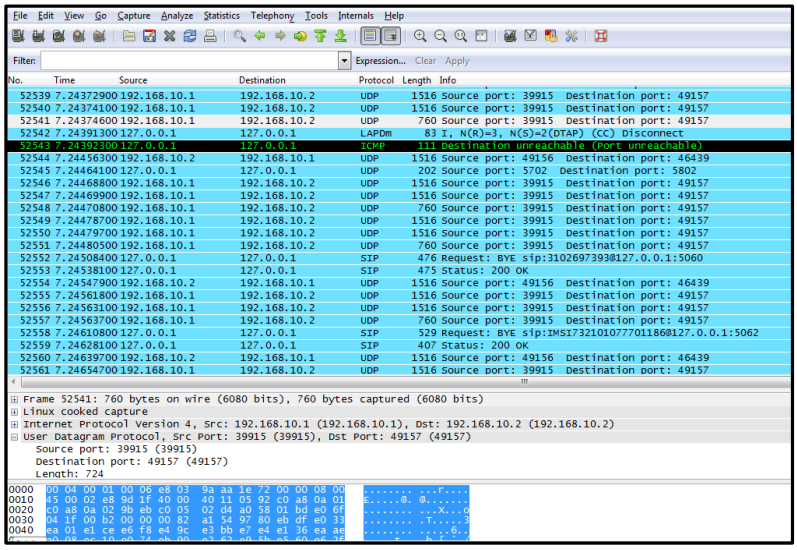


Figura 46. Captura protocolo de transmisión UDP.

8. RECEPCIÓN DE SEÑALES LTE (4G) EN GNURADIO.

Para la implementación de un sistema receptor de señales LTE (4G), se utilizaron los siguientes componentes a nivel de hardware y software.

8.1 Componentes de Hardware:

- Radio Definido por Software Ettus Research USRP N210.
- Dos antenas VERT900 de la empresa Ettus Research con capacidad de detección dualband en los rangos de 824 a 960 MHz y 1710 a 1990 MHz con una ganancia de 3 dBi.
- Ordenador Samsung Notebook NP530U3C, memoria RAM de 4 GB y procesador Intel(R) Core(TM) i5-3317U CPU 1.7GHZ.
- Cable Gigabit Ethernet.

8.2 Componentes de software:

- Sistema Operativo Ubuntu 14.04 LTS instalado en ordenador Samsung.
- GNURadio v.3.7.6.

A continuación, se observa el proceso de descarga y clonación del módulo lte sobre una consola de distribución Linux.

8.3 Proceso de descarga bloques lte.

En la implementación del sistema de diagrama de bloques para la recepción de señales LTE, fue necesario realizar un proceso de descarga de librerías y bloques adicionales. En la descarga del módulo **lte**, que se encuentra disponible en el

enlace https://github.com/kit-cel/gr-lte, fue posible recopilar todos los bloques necesarios para el desarrollo del flujograma en GNURadio [25].

En la Figura 47, se observa el proceso de descarga del paquete lte, al igual que, los comandos utilizados. La clonación de este paquete no contiene dependencias de directorio, es suficiente con digitar el comando con los permisos de administrador.

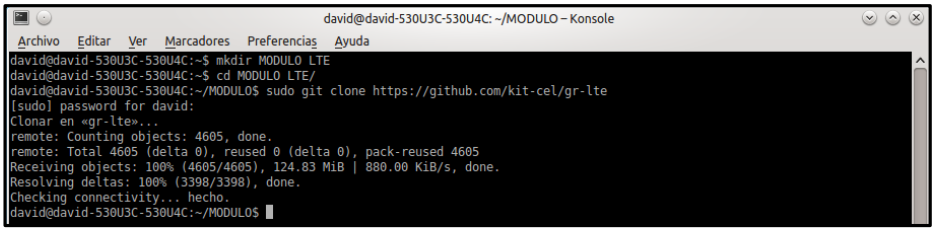


Figura 47. Clonación del paquete lte en Linux**.**

Luego de la finalización del proceso de clonación, se ingresó al directorio gr-lte para comprobar la correcta descarga de los archivos. Así, en la Figura 48, se observa los archivos que componen todo el paquete lte.

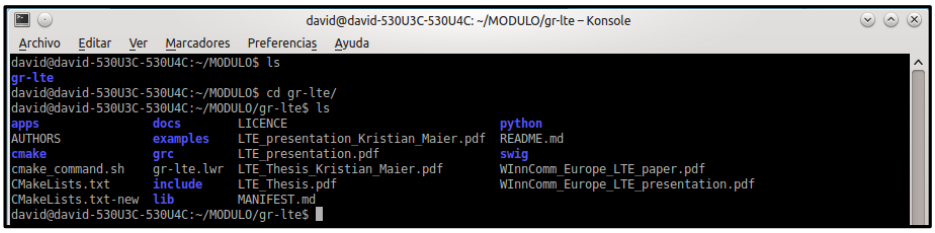


Figura 48. Archivos de descarga del paquete lte.

Los principales archivos a trabajar para el desarrollo del flujograma, fueron ubicados en los directorios /grc/ y /examples/. Todos los componentes de estos directorios se observan en la Figura 49 y Figura 50. De esta manera, se comprobó

la descarga satisfactoria de los bloques que hacen parte del sistema de recepción de señales LTE.

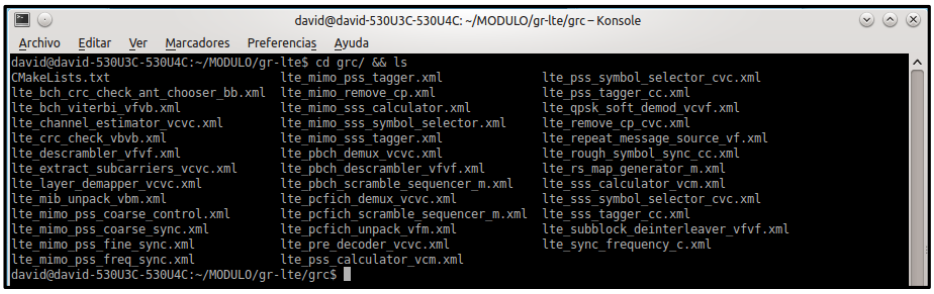


Figura 49. Ubicación archivos de cabecera para los bloques lte.

Es importante mencionar que los archivos que se observan en la Figura 50, son componentes fundamentales para la composición total del flujograma, ya que en estos archivos se alojan bloques de procesamiento únicos que no fueron desarrollados como bloques independientes en la barra de búsqueda de GNURadio.

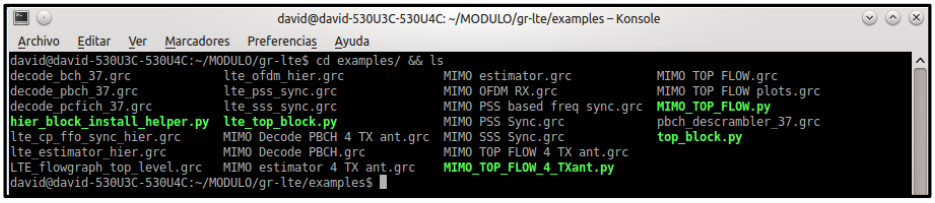


Figura 50. Ubicación archivos de ejemplo de bloques lte.

8.4 Proceso de instalación bloques lte.

Una vez realizado el proceso de descarga del paquete lte, se procedió con la instalación de todos los bloques. Con el fin de incorporar dichos bloques a la barra de búsqueda de GNURadio.

Para la compilación y ejecución de los bloques incorporados en el paquete lte, se utilizó la herramienta make en la consola de Linux. En la Figura 51, se observa la compilación de los bloques con el comando (sudo cmake ../).

Para utilizar la herramienta make, fue necesario crear el directorio /build/ e ingresar a este, para desde allí continuar con la compilación y, copiar el archivo ejecutable cmake_command.sh.

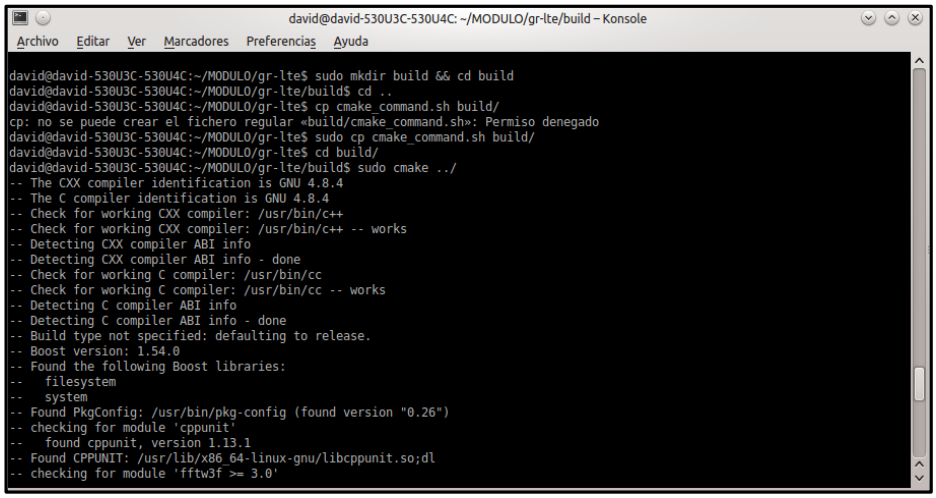


Figura 51. Compilación de los bloques lte en consola Linux.

Posterior a la compilación exitosa, se continuó con la ejecución del archivo Cmake. En la Figura 52, se observa el comando utilizado y la ejecución continua, este proceso puede tardar varios minutos por la verificación de los archivos de cabecera de cada bloque.

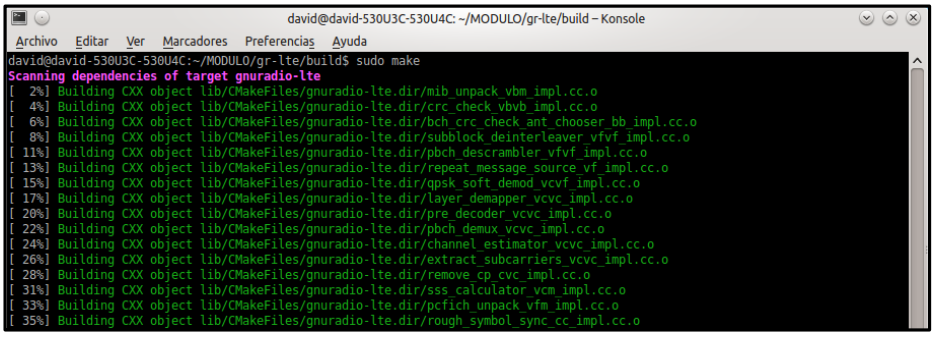


Figura 52.Ejecución de los bloques lte en consola Linux.

Finalmente se procedió con la instalación de los bloques lte sobre el software GNURadio. Para este proceso basta con digitar el comando sudo make install sobre el directorio /build/. En la Figura 53, se observa la ejecución del comando descrito anteriormente sobre la consola Linux.

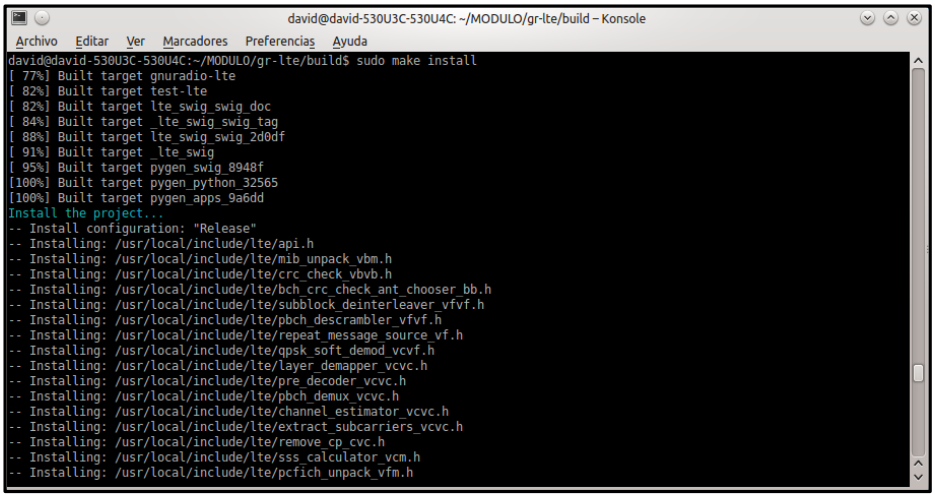


Figura 53. Instalación de los bloques lte en consola Linux.

Para comprobar que todos los bloques fueron instalados correctamente, se ingresó al directorio /usr/local/include/. Allí se sitúa el módulo lte que contiene todos los bloques de GNURadio, como se observa en la Figura 54.

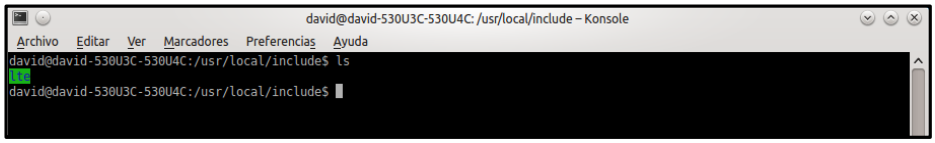


Figura 54. Ubicación del módulo lte.

8.5 Desarrollo de flujograma en GNURadio.

En este apartado se describe el desarrollo del flujograma según su función en conjuntos de bloques y bloques dependientes. Además, de registrar la utilidad de cada bloque. Es importante mencionar que el diagrama de bloques principal se observa en la Figura 6, a continuación se presenta entonces las derivaciones correspondientes de cada bloque según sea el caso.

8.5.1 Bloque de conexión del hardware USRP N210 con GNURadio.

Para establecer la conexión hardware – software, se utilizó el bloque UHD: USRP Source. Así, con las configuraciones adecuadas el dispositivo USRP N210 obtuvo compatibilidad con el software GNURadio. A continuación, en la Figura 55, se observa el bloque en mención y dos bloques dependientes de este.

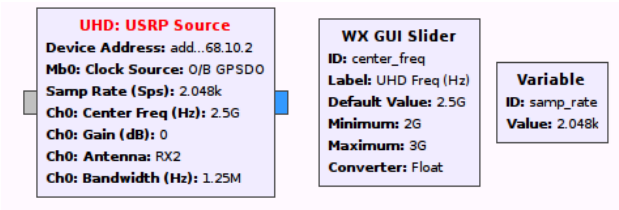


Figura 55.Bloque conexión USRP N210 con GNURadio.

Los parámetros de configuración del bloque UHD: USRP Source se observan en la Figura 56. Donde fue posible cambiar la dirección IP del hardware a conectar, como la tasa de muestreo, ancho de banda de trabajo y demás parámetros.

El bloque WX GUI Slider se utilizó como parámetro secundario del bloque UHD: USRP Source. Realizó la configuración de la frecuencia central de operación, y guardas de frecuencia mínima y máxima para la banda de trabajo del USRP N210. Estas configuraciones se observan en la Figura 57.

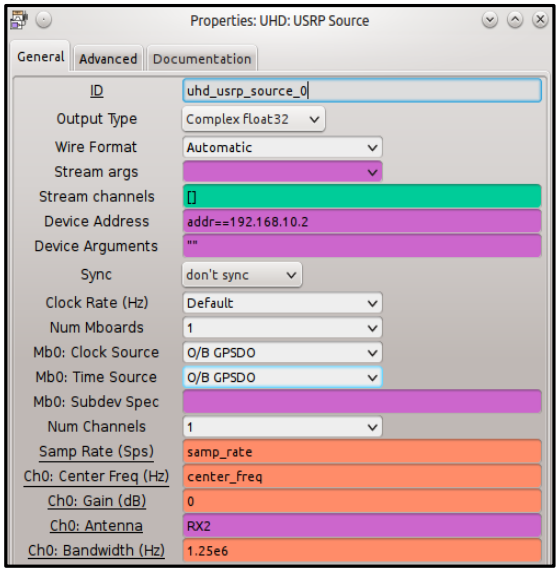


Figura 56. Configuración bloque UHD:USRP Source

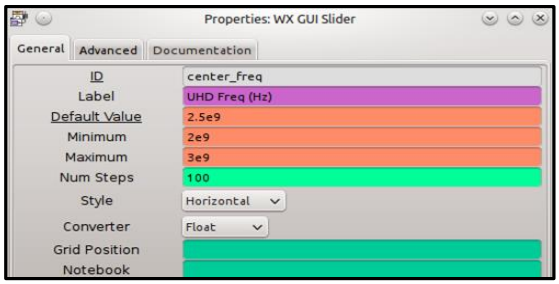


Figura 57. Configuración bloque WX GUI Slider.

Por último, en la configuración del bloque UHD_ USRP N210 se utilizó una variable de parametrización. La configuración de la variable samp_rate se puede observar en la Figura 58.

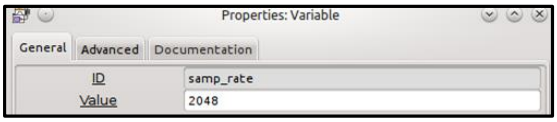


Figura 58.Configuración variable samp_rate.

8.5.2 Diagrama de bloques de sincronización.

En la Figura 59 se observa el diagrama de bloques correspondiente a la sincronización de la señal recibida. Además de variables y parámetros de configuración que hacen parte de los bloques allí descritos.

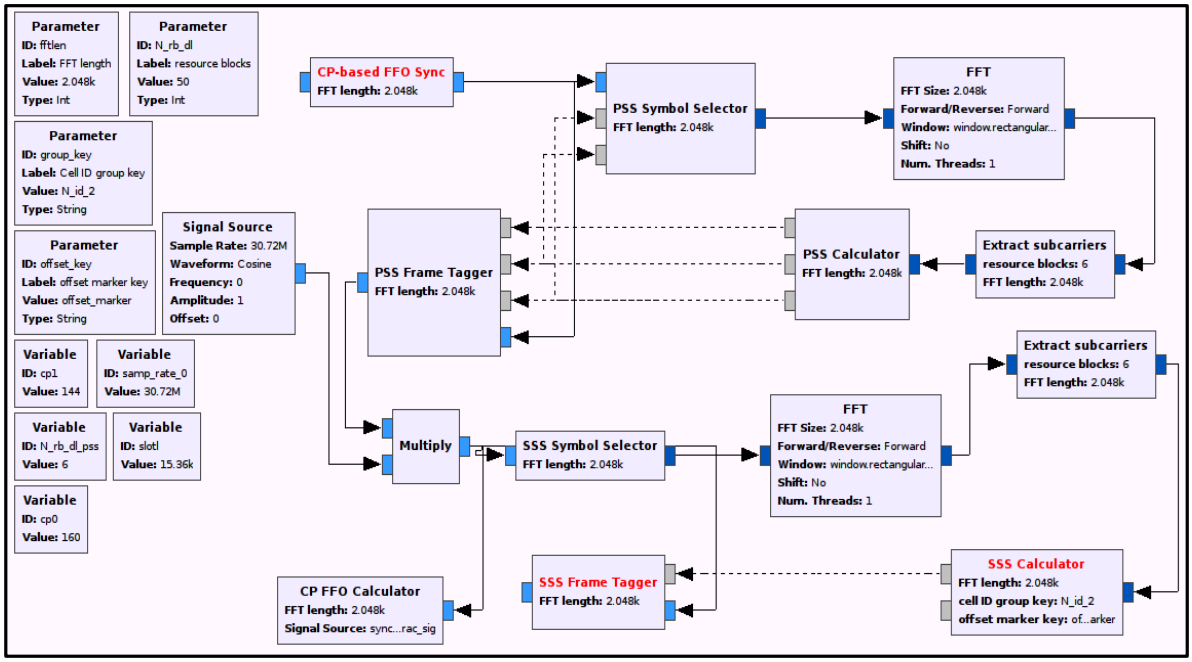


Figura 59. Diagrama de bloques de sincronización.

Los bloques que conforman la capa de sincronización y su respectiva descripción son:

- **CP-based FFO Sync:** este bloque tiene la funcionalidad de sincronización del prefijo cíclico para diferenciarla con los símbolos de principio y fin de la

frecuencia receptora seleccionada. La configuración de este bloque puede ser observada en la Figura 60.

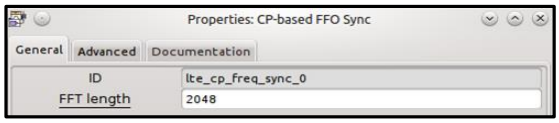


Figura 60.Configuración bloque CP-based FFO Sync.

- **PSS Symbol Selector:** este bloque permitió seleccionar el símbolo de inicio extraído del CP, su configuración su puede observar en la Figura 61.

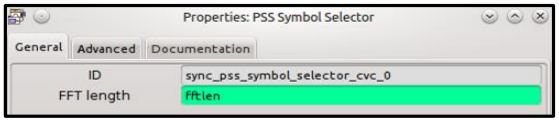


Figura 61. Configuración bloque PSS Symbol Selector.

- **FFT:** se realizó la transformada rápida de Fourier para cambiar el dominio de los componentes de tiempo a frecuencia y así, extraer los símbolos CP en función del tiempo. La configuración de este bloque se observa en la Figura 62.

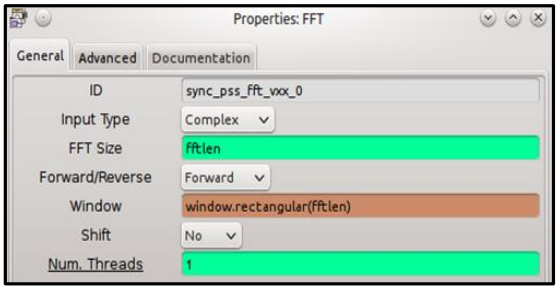


Figura 62. Configuración bloque FFT.

- **Extract subcarriers:** este bloque permitió la selección de la cantidad de resources blocks(6) y de esta manera, determinar el ancho de banda y las

subportadoras a trabajar que para el caso observado en la Figura 63, es de 72 subportadoras.

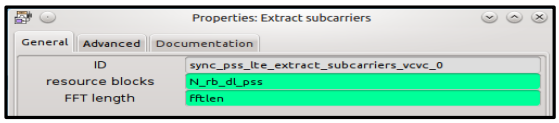


Figura 63.Configuración bloque Extract subcarriers.

- **PSS Calculator:** utilizando el cálculo resource blocks contra subportadoras, este bloque permitió diferenciar por secuencia de símbolos la cantidad de los mismos y organizarlos en tres distintas fuentes para evitar la interferencia inter simbólica. Además, tiene la función de retroalimentar el conjunto de bloques para la sincronización PSS. La configuración de este bloque se observa en la Figura 64.

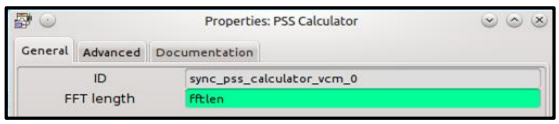


Figura 64.Configuración bloque PSS Calculator.

- **SSS Frame Tagger:** una vez los símbolos primarios son procesados, se implementaron los bloques para la detección de los símbolos secundarios.

  Utilizando una fuente de señal compleja para sincronizar la señal recibida, se realizó nuevamente la transformada rápida de Fourier y la extracción de las subportadoras, esta vez para la selección del símbolo secundario. Este proceso tuvo cabida ya que el bloque Multiplicador entregó una salida en frecuencia.

Un buffer de entrega permitió guardar los símbolos primarios extraídos en los procedimientos anteriores y generar el proceso una vez más para los símbolos secundarios, como se observa en la Figura 59, el bloque SSS Frame Tagger tiene conexión con la entrada de todo el conjunto de bloques que forman la sincronización PSS. La configuración del bloque descrito anteriormente se observa a continuación en la Figura 65.

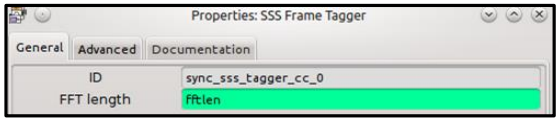


Figura 65. Configuración SSS Frame Tagger.

- **Signal Source:** se implementó este bloque para la sincronización de la señal recibida, con el fin de extraer el símbolo secundario. La configuración puede observarse en la Figura 66.

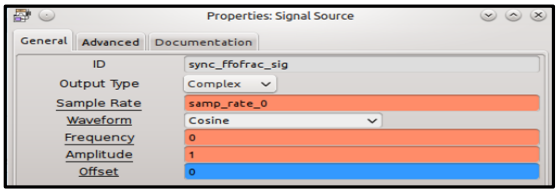


Figura 66. Configuración bloque Signal Source.

- **Multiply:** Este bloque permite multiplicar las dos señales de entrada y una única señal de salida, su configuración se observa en la Figura 67.

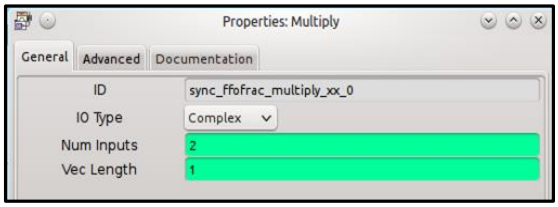


Figura 67. Configuración bloque Multiply.

- **CP FFO Calculator:** este bloque permite la visualización del CP y no contiene salidas. Su configuración se observa en la Figura 68.

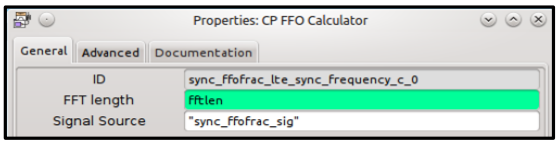


Figura 68. Configuración bloques CP FFO Calculator.

- **SSS Symbol Selector:** este bloque permitió seleccionar el símbolo de inicio secundario extraído del CP, la configuración se observa en la Figura 69.

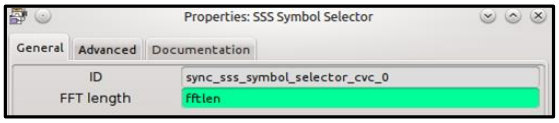


Figura 69.Configuración bloque SSS Symbol Selector.

- **SSS Calculator:** la extracción del NID, es de suma importancia para los procesos de demultiplexación y decodificación de la señal recibida. Esta es una de las funciones del bloque SSS Calculator, además de entregar los símbolos secundarios al bloque SSS Frame Tagger. No posee retroalimentación porque no existen terceros símbolos para extraer. Su configuración se observa en la Figura 70.

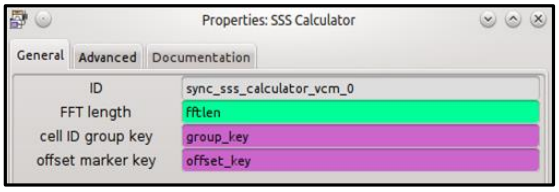


Figura 70.Configuración bloque SSS Calculator.

- **SSS Frame Tagger:** el último de los bloques comprendidos en la etapa de sincronización tiene la función de entregar a los bloques de recepción de señal OFDM los símbolos extraídos en forma continua y no en trama. Además, de la señal recepcionada. La configuración del bloque SSS Frame Tagger se observa en la Figura 71.

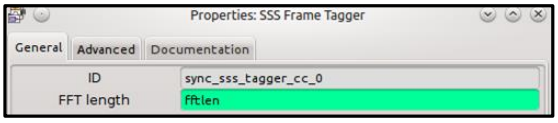


Figura 71. Configuración bloque SSS Frame Tagger.

8.5.3 Diagrama de bloques para la recepción OFDM.

En la Figura 72, se observa el diagrama de bloques que corresponde a la etapa de recepción de la señal OFDM que se realizó luego de la correcta sincronización del sistema.

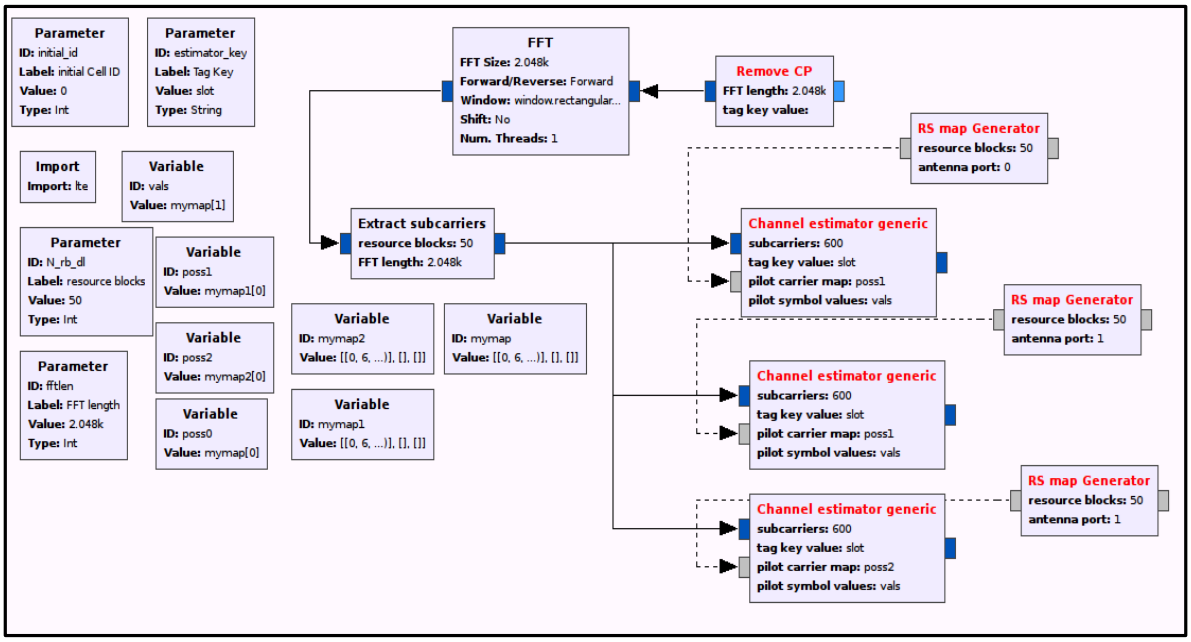


Figura 72.Diagrama de bloques recepción OFDM

- **Remove CP:** este bloque remueve el prefijo cíclico de la señal recibida, ya que no es necesaria cuando el proceso de sincronización termina. La configuración de este bloque se puede observar en la Figura 73.

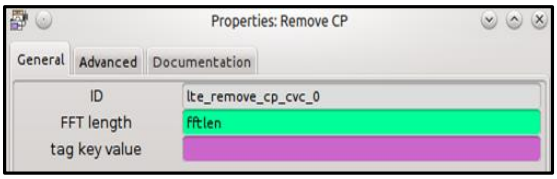


Figura 73.Configuración bloque Remove CP.

- **Channel estimator generic:** es necesario realizar una estimación del canal inalámbrico para conocer variables que afectan la comunicación (Channel Idenficacion Quality) y entregar la señal sin alteraciones a los procesos de demultiplexación y decodificación. Lo anterior define la función principal de este bloque y su configuración se observa en la Figura 74.

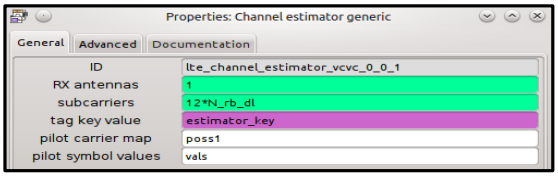


Figura 74. Configuración bloque Channel estimator generic

- **RS map Generator:** este bloque posee comunicación con el NID que corresponde a la identificación de la estación base transmisora, por lo que realizó un mapeo del canal según la cantidad de resource blocks asignados y por último, ser entregados al bloque de estimación de canal. Su configuración es observada en la Figura 75.

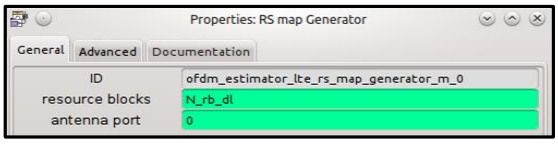


Figura 75. Configuración del bloque RS map Generator.

- **Variable importación lte:** esta variable posee fue caso atípico, ya que depende de la importación del módulo lte, sin embargo, fue fácil conocer su funcionamiento ya que entrega un vector de cuatro posiciones correspondientes a las variables N_rb_dl, initial_id,1,0. La configuración de este bloque se observa en la Figura 76.

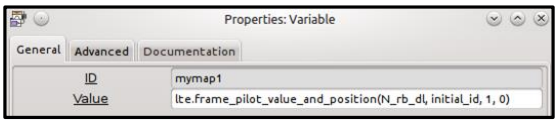


Figura 76. Configuración variable importación lte.

8.5.4 Diagrama de flujo demultiplexación y decodificación PBCH.

Se observa en la Figura 77, el diagrama de bloques que corresponde a la etapa de demultiplexación y decodificación de la señal recepcionada.

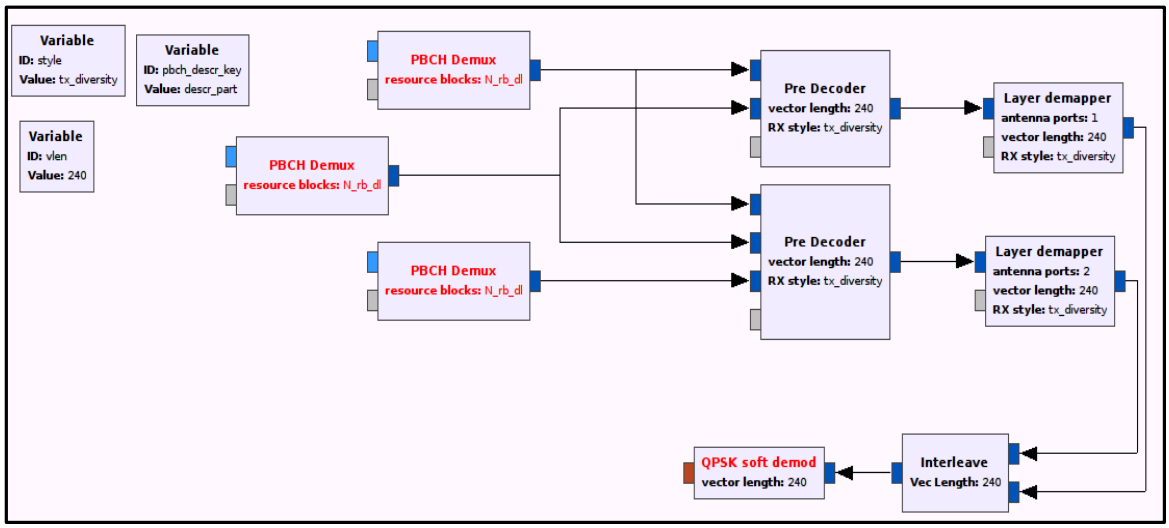


Figura 77.Diagrama de bloques demultiplexación y decodificación PBCH.

- **PBCH Demux:** este bloque realizó la demultiplexación del canal físico de difusión, que es transmitido junto con el MIB por el eNodeB. La configuración del bloque se observa en la Figura 78.

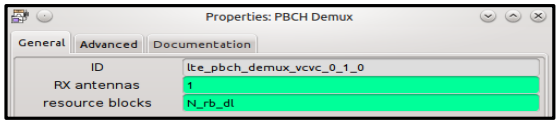


Figura 78. Configuración bloque PBCH Demux.

- **Pre Decoder:** la decodificación del canal PBCH comprendió una etapa previa de simultaneidad con la señal recepcionada y se realiza en dos fases para tener posterior redundancia en la demodulación QPSK. La configuración de este bloque se observa en la Figura 79.

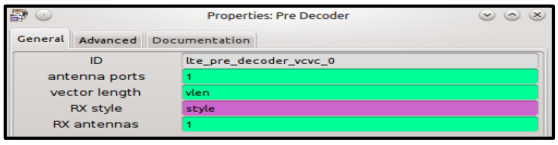


Figura 79. Configuración bloque Pre Decoder.

- **Layer demapper:** este bloque retiró las etiquetas de los símbolos que fueron extraídos en etapas anteriores. La configuración se puede observar en la Figura 80.

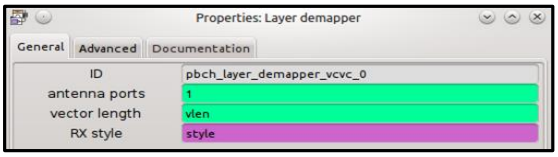


Figura 80. Configuración bloque Layer demapper.

- **Interleave:** este bloque posee la funcionalidad de un código de línea, ir intercambiando los valores de cada entrada y confinarlas en la única salida que posee el bloque. De esta manera, el primer valor de la primera entrada y el segundo de la segunda serán los dos primeros valores de la salida. La configuración del bloque se puede observar en la Figura 81.

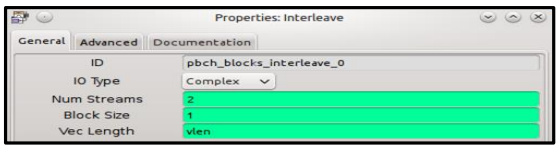


Figura 81.Configuración bloque Interleave.

- **QPSK soft demod:** las señales transmitidas mediante el sistema de acceso al medio OFDM tienen modulaciones en QPSK o QAM y este boque se encarga de demodular la señal recibida para que, terminado esta etapa se extraiga la carga útil de la señal. La configuración de este importante módulo se observa en la Figura 82.

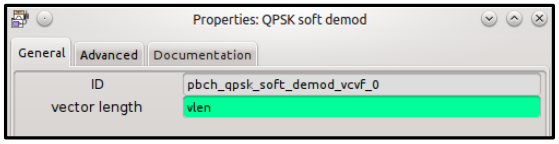


Figura 82. Configuración bloque QPSK soft demod.

8.5.5 Diagrama de flujo decodificación BCH y MIB.

En la Figura 83 se observa el diagrama de bloques que corresponde a la etapa de decodificación BCH y entrega del MIB de la señal recepcionada.

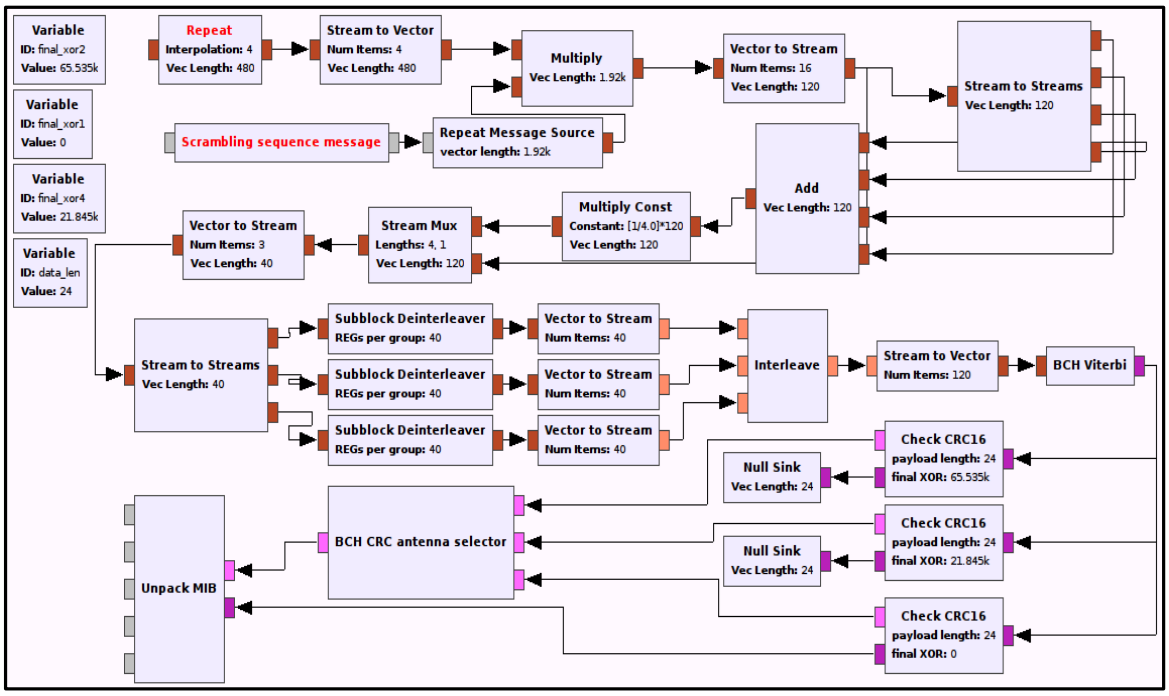


Figura 83.Diagrama de flujo decodificación BCH y MIB.

- **Repeat:** este bloque realiza una interpolación por un factor entero a la señal para reducir los efectos de muestreo. La configuración se observa en la Figura 84.

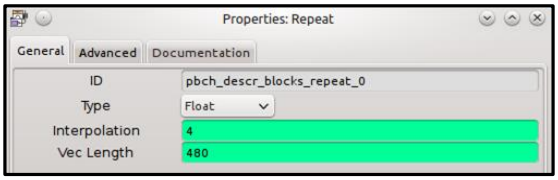


Figura 84. Configuración bloque Repeat.

- **Stream to Vector:** este bloque realizó una conversión de un flujo de valores a un vector. Se observa su configuración en la Figura 85.

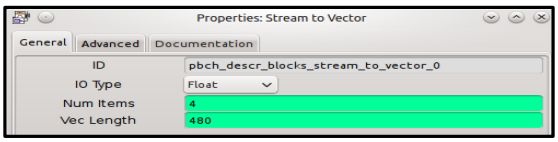


Figura 85.Configuración bloque Stream to Vector.

- **Scrambling sequence Message:** este bloque utiliza el NID para multiplicar posteriormente este dato con la señal y realizar el proceso de descrambling. La configuración del bloque se puede observar en la Figura 86.

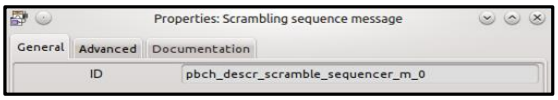


Figura 86. Configuración bloque Scrambling sequence Message

- **Repeat Message Source:** se realiza una continuación del dato NID esta vez descrito como mensaje. La configuración del bloque se observa en la Figura 87.

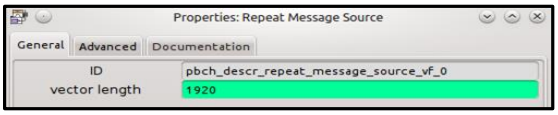


Figura 87. Configuración bloque Repeat Message Source.

- **Multiply:** este bloque como se describió anteriormente, une las dos señales bajo un solo parámetro de salida. Su configuración puede ser vista en la Figura 88.

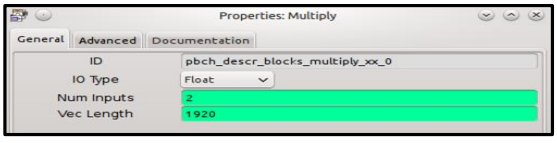


Figura 88. Configuración bloque Multiply.

- **Vector to Stream:** este bloque permite organizar y distribuir los valores provenientes de un vector en flujos de datos. La configuración de este bloque se observa en la Figura 89.

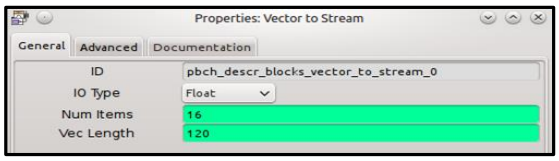


Figura 89. Configuración bloque Vector to Stream.

- **Stream to Stream:** la cadena de valores se dividen para tener varias salidas, sin afectar la misma. La configuración de este bloque se observa en la Figura 90.

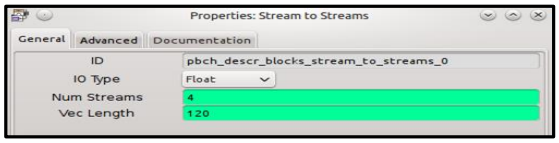


Figura 90. Configuración bloque Stream to Stream.

- **Add:** este bloque tiene la funcionalidad de caracterizar los datos entregados por el bloque anterior para unirlos sobre una misma salida. La configuración de este bloque se observa en Figura 91.

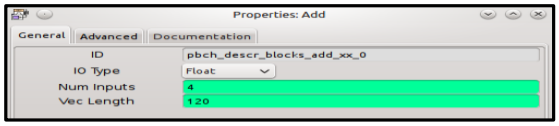


Figura 91. Configuración bloque Add.

- **Multiply Const:** este bloque simplemente multiplica el valor ingresado a este por una constante, el valor que fue multiplicado se puede observar en la Figura 92.

.

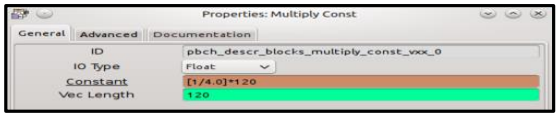


Figura 92. Configuración bloque Multiply Const.

- **Stream Mux:** este bloque se utilizó para darle flujo a la cadena de valores mediante un multiplexor. La configuración puede observarse en la Figura 93.

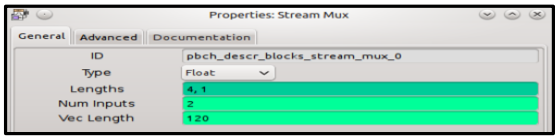


Figura 93. Configuración Stream Mux.

- **Subblock deinterleaver:** los bloques anteriores a este, son procesos de transformación y adecuación de señal. En el diagrama de bloques para la decodificación del canal BCH se realizará un Interleave para intercalar los datos de los valores de sus entradas, como se describió para el Interleave del diagrama para la recepción OFDM.

  En esta ocasión se trabajó un subproceso anterior, ya que ingresan flujos de datos al código de línea. La configuración del bloque se observa en la Figura 94.

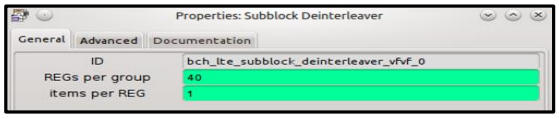


Figura 94.Configuración bloque Subblock deinterleaver.

- **BCH Viterbi:** se realizó a la señal el procesado del algoritmo Viterbi para terminar todos los procesos de envoltura y obtener la señal más pura. El bloque con su respectiva configuración se observa en Figura 95.

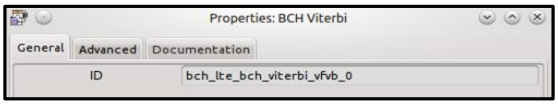


Figura 95. Configuración bloque BCH Viterbi.

- **Check CRC16:** para realizar un proceso de verificación se utilizó el bloque Check CRC16, así los datos del proceso de descrambling son comprobados en conjunto con el siguiente bloque. La configuración de este bloque se observa en la Figura 96.

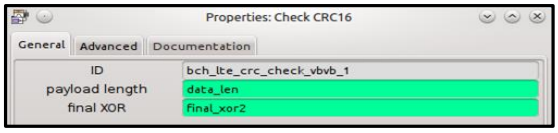


Figura 96. Configuración bloque Check CRC16.

- **BCH CRC antenna selector:** este bloque se utilizó para terminar con el proceso de descrambling, que se puede realizar conociendo la identificación de la estación base transmisora. La configuración del bloque se observa en la Figura 97.

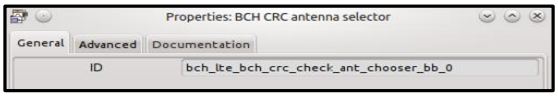


Figura 97. Configuración bloque CRC antenna selector.

- **Unpack MIB:** este bloque permite visualizar la información que se transporta en la carga útil de la señal. La configuración del bloque se observa en la Figura 98.

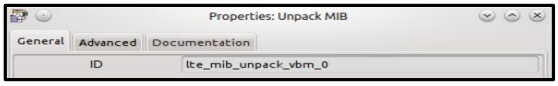


Figura 98. Configuración bloque Unpack MIB.

9. CONCLUSIONES.

La evolución de los sistemas de telefonía móvil, se ha visto enmarcada por el cambio de tecnología en cada una de sus generaciones. Siendo el método de acceso al medio el principal factor a influir. De esta manera, la innovación en tecnología deriva en un constante ajuste de infraestructura, para la adecuada prestación del servicio, y se debe reemplazar desde el terminal hasta software de gestión en la red.

El dispositivo Ettus USRP N210 posee gran capacidad de procesamiento y ofrece compatibilidad con herramientas de software libre. Sin embargo, se observó que para la interacción software-hardware, previamente se deben cumplir con una serie de dependencias que realizan el proceso de sincronía, para así obtener la compatibilidad exitosa. Las dependencias son de acceso libre y se instalan en consola de distribución Linux, entre ella están los servidores ntp y bind9, y para el manejo exclusivo de los sistemas SDR se utilizó la dependencia UHD binary. A nivel de hardware solo es necesario contar con antenas receptoras en las bandas a trabajar y conexión física con el ordenador, ya sea por medio del puerto Gigabit Ethernet o puerto USB.

La práctica de laboratorio comprendida en la implementación de una estación base de segunda generación, fue determinante para evidenciar los factores tecnológicos necesarios para el adecuado servicio de la misma. Por lo tanto, se determinó que las herramientas básicas para la puesta en marcha son; un servidor funcional en el manejo y control de las bases de datos, otro para el servicio de mensajería de texto corta, otro para la canalización del tráfico de voz y una última herramienta para generar y realizar configuraciones de la portadora en el canal descendente.

En la recepción de señales LTE de cuarta generación, la práctica de laboratorio se trabajó mediante la jerarquización de bloques, y se observó que fue funcional para completar cada una de las etapas de sustracción de cabeceras y demás códigos que hacen parte de la señal pura transmitida por un eNodeB. Además de visualizar los componentes que son incorporados en una señal 4G de forma práctica.

La documentación de guías de laboratorio, fue un proceso encaminado al ambiente académico, donde se resaltó la importancia de las competencias que hacen parte del programa de ingeniería de telecomunicaciones. Además, fue un espacio para complementar detalladamente los procesos que hicieron parte de las prácticas de laboratorio, en la búsqueda de futuros estudiantes que las desarrollen.

Los futuros procesos de investigación, desarrollados por otros integrantes del semillero **telesoft**, tendrán como apoyo el trabajo realizado anteriormente en proyectos que conciernen las temáticas de los sistemas de comunicaciones móviles y de radio definido por software. Como auxiliar de investigación fue fundamental para mi formación la participación dentro del semillero, ya que me permite fortalecer mis conocimientos en el área investigativa.

10. REFERENCIAS BIBLIOGRÁFICAS.

[1] Ministerio de las tecnologías de la información y las comunicaciones, “Boletín trimestral de las Tic, cifras tercer trimestre de 2014,” *Bogotá Colombia, Diciembre de 2014*.

[2] E. Martínez, “La evolución de la telefonía móvil,” *articulo publicado en la revista RED*, 2001.

[3] Ministerio de las tecnologías de la información y las comunicaciones, “Adjudicación licencias 4G.” 03-Jul-2013.

[4] M. Tolstrup, *Indoor Radio Planning: A Practical Guide for Gsm, Dcs, Umts, Hspa and Lte*. John Wiley & Sons, 2011.

[5] B. A. Series, “Mobile network evolution: a revolution on the move,” *IEEE Communications magazine*, p. 104, 2002.

[6] C. P. Hugo, *Redes móviles celulares*, Editorial@usantotomas.edu.co. Bogota, D. C., Colombia: Universidad Santo Tomás, 2009.

[7] H. Holma and A. Toskala, *HSDPA/HSUPA for UMTS: high speed radio access for mobile communications*. John Wiley & Sons, 2007.

[8] Q. Li, G. Li, W. Lee, M. Lee, D. Mazzarese, B. Clerckx, and Z. Li, “MIMO techniques in WiMAX and LTE: a feature overview,” *Communications Magazine, IEEE*, vol. 48, no. 5, pp. 86–92, 2010.

[9] A. Furuskar, S. Mazur, F. Muller, and H. Olofsson, “EDGE: Enhanced data rates for GSM and TDMA/136 evolution,” *Personal Communications, IEEE*, vol. 6, no. 3, pp. 56–66, 1999.

[10] K. Raghunath and A. Chockalingam, “SC-FDMA versus OFDMA: Sensitivity to large carrier frequency and timing offsets on the uplink,” in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, 2009, pp. 1–6.

[11] A. Pachón, “Evolución de los sistemas móviles celulares GSM,” *Sistemas & Telemática*, vol. 2, no. 4, 2006.

[12] C. Bettstetter, H.-J. Vogel, and J. Eberspacher, “GSM phase 2+ general packet radio service GPRS: Architecture, protocols, and air interface,” *Communications Surveys & Tutorials, IEEE*, vol. 2, no. 3, pp. 2–14, 1999.

[13] J. Cai and D. Goodman, “General packet radio service in GSM,” *Communications Magazine, IEEE*, vol. 35, no. 10, pp. 122–131, 1997.

[14] E. A. C. Pedraza and R. F. Escobar, “SISTEMAS MULTIMEDIA BASADOS EN PROTOCOLO IP IMS APLICADOS A SERVICIOS LTE DE 4G,” *REDES DE INGENIERÍA*, vol. 3, no. 2, pp. 100–108, 2013.

[15] C. P. Hugo, *Comunicaciones Móviles de última generación*, María Paula Godoy Casasbuenas. Bogota, D. C., Colombia: Universidad Santo Tomás, 2012.

[16] J. Z. Arunabha Ghosh, *Fundamentals of LTE*. Boston: Prentice Hall, 2010.

[17] M. A. D. V. Diego, R. E. Caldera, J. M. R. Cortés, and J. M. Carballido, “MODELADO DE CANAL INALÁMBRICO OFDM, HACIA LAS COMUNICACIONES 4G.,” *e-Gnosis*, vol. 7, 2009.

[18] G. Berardinelli, L. Ruiz de Temino, S. Frattasi, M. I. Rahman, and P. Mogensen, “OFDMA vs. SC-FDMA: performance comparison in local area IMT-A scenarios,” *Wireless Communications, IEEE*, vol. 15, no. 5, pp. 64–72, 2008.

[19] H. Holma and A. Toskala, *LTE for UMTS-OFDMA and SC-FDMA based radio access*. John Wiley & Sons, 2009.

[20] “07495 Ettus N200-210 DS Flyer.indd - 07495_Ettus_N200-210_DS_Flyer_HR_1.pdf.” .

[21] J. de J. R. Uribe, T. M. Bojacá, and C. H. C. Sánchez, “Caracterización de la plataforma de radio definido por software USRP N210-WBX,” *Gerencia Tecnológica Informática*, vol. 12, no. 34, pp. 91–102, 2013.

[22] R. Anand, G. Xavier, V. Hariharan, N. Prasannan, R. Peter, and K. Soman, “GNU Radio Based Control System,” in *Advances in Computing and Communications (ICACC), 2012 International Conference on*, 2012, pp. 259–262.

[23] J. D. VÁSQUEZ, I. F. SANTA, and J. V. RESTREPO, “Prototipo de una Estación Celular Portátil para Atención de Emergencias.”

[24] A. Azad, “Open BTS Implementation with Universal Software Radio Peripheral,” *Department of Electrical and Computer Engineering, Virginia Polytechnic and State University: Virginia, VA, USA*, 2011.

[25] J. Demel, S. Koslowski, and F. K. Jondral, “A LTE Receiver Framework Implementation in GNU Radio.”

[26] M.-G. Di Benedetto, A. Cattoni, J. Fiorina, F. Bader, and L. De Nardis, *Cognitive Radio and Networking for Heterogeneous Wireless Networks: Recent Advances and Visions for the Future*. Springer, 2014.

[27] M. Iedema, *Getting Started with OpenBTS*. O'Reilly Media, Inc., 2014.