

**PROTOTIPO DE UNA CENTRAL DE TELEFONÍA MÓVIL
PORTÁTIL PARA USO EN LOS LABORATORIOS DE LA FIET**



**Sonia Urbano Bolaños
Cesar Ricardo Maca García**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán, Septiembre de 2015**

**PROTOTIPO DE UNA CENTRAL DE TELEFONÍA MÓVIL
PORTÁTIL PARA USO EN LOS LABORATORIOS DE LA FIET**



Trabajo de Grado presentado como requisito para optar el título
de Ingeniero en Electrónica y Telecomunicaciones

**Sonia Urbano Bolaños
Cesar Ricardo Maca García**

**Director: Mag. Fernando Aparicio Urbano Molano
Codirector: Dr. Álvaro Rendón Gallón**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán, Septiembre de 2015**

Índice general

Lista de figuras	III
Lista de tablas	IV
1. Conceptos generales	1
1.1. Introducción	1
1.2. Marco regulatorio de las comunicaciones móviles en Colombia	2
1.3. Generalidades de GSM, SDR y VoIP	4
1.3.1. Sistema global de comunicaciones móviles (GSM)	4
1.3.2. Radio Definido por Software (SDR)	8
1.3.3. Voz sobre IP (VoIP)	10
2. Implementación de una celda de red GSM con OpenBTS	13
2.1. Elementos de software y hardware utilizados	13
2.1.1. GNU Radio	13
2.1.2. OpenBTS	14
2.1.3. USRP N210	17
2.1.4. Asterisk	22
2.2. Instalación y configuración	25
2.2.1. Instalación de GnuRadio	25
2.2.2. Instalación de Asterisk	26
2.2.3. Instalación de OpenBTS	27
2.2.4. Conexión con Equipo USRP N210	28
2.2.5. Habilitación del soporte para dispositivos UHD y creación de enlace simbólico	29
2.2.6. Configuración de OpenBTS	31
2.2.7. Instalación de sipauthserve y smqueue	33
2.2.8. Configuración de parámetros de OpenBTS	34
2.2.9. Registro de terminales en OpenBTS	35
3. Servicios de comunicación a la red de telefonía móvil a través de Asterisk	37
3.1. Mensajes y llamadas	37
3.2. Configuración del idioma español	38
3.3. voicemail	39

3.4. Servicio de transferencia de llamada atendida	41
3.5. IVR	42
3.6. Conexión Internet	44
3.7. Establecimiento de llamadas de un terminal Ip a un terminal móvil GSM	45
4. Practicas de Laboratorio	47
4.1. Descripción de las practicas propuestas	47
4.1.1. Practica 1. Central de telefonía móvil. Conceptos generales	47
4.1.2. Practica 2. Creación de una red GSM basada en el pro- yecto OpenBTS	48
4.1.3. Practica 3. Configuración de servicios de valor agregado .	48
4.2. Evaluación de las practicas propuestas	49
4.2.1. Análisis de la encuesta.	49
Trabajos futuros	59
Conclusiones	60
Bibliografía	62

Índice de figuras

1.1. Arquitectura GSM.	6
1.2. Arquitectura SDR, Tomado de [8]	9
1.3. Arquitectura VoIP, Tomado de [10]	11
2.1. Arquitectura OpenBTS, Tomado de [15]	16
2.2. Diagrama de bloque USRP tomado de [19]	18
2.3. USRP N210, Tomado de [21]	19
2.4. Funciones placa madre.	20
2.5. tarjeta madre USRP N210, Tomado de [21]	21
2.6. Interfaz grafica gnuradio-companion.	26
2.7. Verificación de la instalación del controlador.	30
2.8. Variables de configuración de OpenBTS.	32
3.1. Servicio de Voicemail	40
3.2. Transferencia de llamada atendida	41
3.3. Transferencia entre tres teléfonos.	42
3.4. Conexión a Internet con Red de prueba	45
3.5. Conexión terminales GSM-IP	46
4.1. Resultados primera pregunta	50
4.2. Resultados segunda pregunta	51
4.3. Resultados tercera pregunta	52
4.4. Resultados cuarta pregunta	53
4.5. Resultados quinta pregunta	54
4.6. Resultados sexta pregunta	55
4.7. Resultados séptima pregunta	56
4.8. Resultados octava se pregunta	57

Índice de tablas

1.1. Bandas de frecuencia GSM, Tomada de [2]	5
1.2. Frecuencia Offset y ARFCN de GSM, Tomado de [5]	6
1.3. Identificador internacional de abonados móviles.	7
2.1. Códecs y formatos de audio soportados por Asterisk. tomado de autores	25
2.2. Códigos de identificación de Operadores móviles en Colombia, Tomado de [30]	35
2.3. Inicialización de ejecutables para red OpenBTS.	35

Capítulo 1

Conceptos generales

1.1. Introducción

Los sistemas de comunicaciones digitales han avanzado gracias a la evolución de dispositivos electrónicos como procesadores DSP y hardware configurable (FPGA). Es así que con el transcurso del tiempo se han implementado sistemas de comunicaciones cada vez más eficientes y de menor complejidad. Una de las técnicas para la implementación de sistemas de comunicaciones digitales que ha ido ganando terreno a nivel mundial, especialmente en el ámbito académico, es la técnica SDR (*Software Defined Radio*, Radio Definido por Software), debido a su gran flexibilidad, versatilidad y eficiencia. SDR se define como radios en los cuales gran parte de las funciones de la capa física son implementadas por medio de software libre y hardware reconfigurable.

Una de las múltiples aplicaciones de la técnica SDR consiste en utilizarla como una opción para el estudio de las redes de telefonía móvil, buscando así otra alternativa de profundizar los conceptos alrededor de las redes de telecomunicaciones basadas en IP y las plataformas de servicio, que permitan construir algunos servicios de valor agregado, siendo esta característica una de las ventajas clave, permitiendo así que los conceptos teóricos sean verificados en práctica en un ambiente interactivo y realista.

Para la implementación de sistemas de comunicaciones SDR, existen diferentes arquitecturas, dentro de las cuales se tiene implementación directa en dispositivos FPGA. Para el desarrollo e implementación de la Central de Telefonía Móvil Portátil, se acude al proyecto OpenBTS¹ que genera una interfaz de aire GSM (*Global System for Mobile Communications*, Sistema Global para las Comunicaciones Móviles) «um», que es la interfaz que se usa para establecer la comunicación entre la MS (*Mobile Station*, Estación Móvil) y la BTS (*Base Transceiver Station*, Estación Base Transceptora) en una arquitectura de red GSM convencional. OpenBTS hace uso del hardware USRP (*Universal*

¹<http://openbts.org/>

Software Radio Peripheral, Periférico Universal de Radio por Software) y el software GNU Radio corriendo sobre un computador para construir una completa aplicación de software radio. Además utiliza el software Asterisk para realizar el control y conmutación de las llamadas.

Este proyecto se presenta como una alternativa de estudio para laboratorios de los programas de Tecnología en Telemática e Ingeniería Electrónica y Telecomunicaciones de la FIET (Facultad de ingeniería Electrónica y Telecomunicaciones) utilizando el proyecto OpenBTS y el dispositivo USRPN210, de manera que pueda ser integrado a ellos como una opción para afianzar la investigación de conceptos relacionados a comunicaciones móviles y sistemas de radio en el mundo real.

A lo largo del texto se presenta un marco teórico que fundamenta los principales conceptos relacionados con las redes de comunicación en telefonía móvil, se realiza una introducción al concepto de SDR y VoIP; descripción del dispositivo USRPN210, seguido de la presentación y configuración de las plataformas OpenBTS, GNU Radio y Asterisk, continuando con la implementación de algunos servicios de valor agregado. El montaje requerido para la implementación del laboratorio es descrito, acompañado de las prácticas de laboratorio propuestas. Finalmente, se realizan las conclusiones del trabajo y se proponen las respectivas recomendaciones y sugerencias para trabajos futuros.

1.2. Marco regulatorio de las comunicaciones móviles en Colombia

La utilización y apropiación de tecnologías de la información y las comunicaciones (TIC) tiene un impacto positivo en la salvaguardia de la paz y el desarrollo social y económico de los Estados. Las radiocomunicaciones son parte esencial de las TIC y como tal representan un elemento indispensable para promover el desarrollo de las Naciones. El espectro radioeléctrico es el recurso fundamental para el uso de las radiocomunicaciones que hoy nos permiten comunicarnos a distancia sin necesidad de cables o estar limitados a una única ubicación geográfica.

Producto de las crecientes necesidades en comunicaciones personales, y de entidades privadas o estatales, cada vez se requieren soluciones más universales, más robustas y complejas. El avance de las TIC se ha hecho vertiginoso y el despliegue de soluciones convergentes es cada vez mayor. Por esto es necesario actualizar permanentemente la reglamentación nacional e internacional sobre el uso del espectro radioeléctrico; para ello, los gobernantes de las naciones, reunidos durante las Conferencias Mundiales de Radiocomunicaciones (CMR)

de la Unión Internacional de Telecomunicaciones (UIT) ², revisan periódicamente el Reglamento de Radiocomunicaciones, en el cual se define, entre otros elementos, la atribución del espectro radioeléctrico en todo el planeta.

Producto de la participación de Colombia en estas labores de la UIT, así como del desarrollo de estudios locales, y del análisis de la reglamentación nacional, se creó la Agencia Nacional del Espectro (ANE) (Ley 1341 de 2009) ³ como una Unidad Administrativa Especial cuyo objetivo es brindar el soporte técnico para la gestión y la planeación del espectro radioeléctrico. Adicionalmente le corresponde la vigilancia y control en coordinación con las diferentes autoridades que tengan funciones o actividades relacionadas con el espectro radioeléctrico. Para el cumplimiento de su misión, la ANE realiza actividades de monitoreo del espectro radioeléctrico y tiene funciones sancionatorias por vía administrativas en los casos de infracción al régimen del espectro definido por el Ministerio de Tecnologías de la Información y las Comunicaciones.

la Agencia Nacional del Espectro (ANE) ha desarrollado el Cuadro Nacional de Atribución de Bandas de Frecuencias; un cuadro armonizado y actualizado con las últimas disposiciones internacionales en la materia y permite que los diferentes servicios de radiocomunicación del país, operen en bandas de frecuencias definidas previamente para cada uno de ellos, con el fin de asegurar su operatividad, minimizar la probabilidad de interferencias objetables y permitir la coexistencia de servicios de telecomunicaciones dentro de una misma banda de frecuencias, que le permitirá al país continuar en la senda del desarrollo constante [1].

Algunas frecuencias o bandas de frecuencias del espectro radioeléctrico son de uso libre por parte del público en general y le corresponde autorizarlas al Ministerio de Tecnologías de la Información y las Comunicaciones sin contraprestación o pago por esa utilización. El uso de estas frecuencias o bandas de frecuencias ha sido regulado mediante las resoluciones No. 797 de 2001, 2190 de 2003, 689 de 2004, 1201 de 2004, 1713 de 2004, 2544 de 2009 del Ministerio de Tecnologías de la Información y las Comunicaciones ⁴.

Usar el espectro radioeléctrico sin permiso previo y expreso otorgado por el Ministerio de Tecnologías de la Información y las Comunicaciones acarrea sanciones de carácter administrativo, sin perjuicio de las condenas penales a que haya lugar. En lo penal, el artículo 1° del Código Penal Colombiano establece lo siguiente:

²<http://www.itu.int/ITU-R/index.asp?category=conferences&rlink=wrc&lang=es>

³COLOMBIA, CONGRESO DE LA REPÚBLICA. Ley 1341. (30, julio, 2009). Por la cual se definen principios y conceptos sobre la sociedad de la información y la Organización de las TIC, se crea la ANE y se dictan otras disposiciones. Diario Oficial. Bogotá, 2009. No. 47426.

⁴<http://www.ane.gov.co/cnabf/index.php/referencias>

“ARTÍCULO 1°. El artículo 257 del Código Penal quedará así:”

“ARTICULO 257 (Modificado por el artículo 1 de la Ley 1032 de 2006): Del acceso ilegal o prestación ilegal de los servicios de telecomunicaciones. El que acceda o use el servicio de telefonía móvil celular u otro servicio de telecomunicaciones mediante la copia o reproducción no autorizada de señales de identificación de equipos terminales de estos servicios, derivaciones, o uso de líneas de telefonía pública básica conmutada local, local extendida o de larga distancia no autorizadas, o preste servicios o actividades de telecomunicaciones con ánimo de lucro no autorizados, incurrirá en prisión de cuatro (4) a diez (10) años y multa de quinientos (500) a mil (1.000) salarios mínimos legales mensuales. La pena anterior se aumentará de una tercera parte a la mitad, para quien hubiese explotado comercialmente por sí o por interpuesta persona, dicho acceso, uso o prestación de servicios de telecomunicaciones no autorizados. Igual aumento de pena sufrirá quien facilite a terceras personas el acceso, uso ilegítimo o prestación no autorizada del servicio de que trata este artículo.”

1.3. Generalidades de GSM, SDR y VoIP

1.3.1. Sistema global de comunicaciones móviles (GSM)

GSM es un estándar presentado por primera vez como *Groupe Special Mobile*, desarrollado en 1982 por un consorcio de países Europeos, en la conferencia de telecomunicaciones CEPT (Conferencia Europea de Correos y Telégrafos), a fin de solucionar inconvenientes presentados con la telefonía analógica, también denominado estándar “de segunda generación”(2G) ya que a diferencia de la primera generación de teléfonos portátiles, las comunicaciones se producen de un modo completamente digital[2].

En 1991 se convirtió en un estándar internacional llamado “Sistema global de comunicaciones móviles”. En la actualidad es el estándar más predominante en el mundo con más del 90 % de los terminales en uso según la Asociación GSM ⁵.

GSM basa su división de acceso al canal en combinar modelos de reparto del espectro disponible; tales como SDMA, FDMA, TDMA, FMMA. El primero es determinante a la hora de especificar la arquitectura de red, mientras que el resto se resuelve con circuitería en los terminales y antenas del operador [3].

⁵<http://www.gsma.com/publicpolicy/wp-content/uploads/2012/04/latammospa.pdf>, pp. 4

Particularidades de GSM

Bandas de frecuencia manejadas en la interfaz de radio de GSM:

Cabe resaltar la importancia que representa la frecuencia de operación que maneja la red GSM, en la tabla 1.1 se ilustra las principales bandas utilizadas por este estándar de comunicación.

Banda	Nombre	Canales	Uplink(Mhz)	Downlink(MHz)	Notas
GSM 850	GSM 850	128 - 251	824 - 849	869 - 894	Usada en los EE.UU., Sudamérica y Asia.
GSM 900	P-GSM 900	0-124	890 - 915	935 - 960	La banda con que nació GSM en Europa y la más extendida
	E-GSM 900	974 - 1023	880 - 890	925 - 935	E-GSM, extensión de GSM 900
	R-GSM 900	n/a	876 - 880	921 - 925	GSM ferroviario (GSM-R)
GSM 1800	GSM 1800	512 - 885	1710 - 1785	1805 - 1880	
GSM 1900	GSM 1900	512 - 810	1850 - 1910	1930 - 1990	Usada en Norteamérica, incompatible con GSM-1800 por solapamiento de bandas

Tabla 1.1: Bandas de frecuencia GSM, Tomada de [2]

Especificaciones GSM: Fueron elaboradas y editadas por el ETSI ⁶, divididas en series con base en su funcionamiento como se describe a continuación.

1. 01.XX Cuestiones generales.
2. 02.XX Aspectos de servicio.
3. 03.XX Aspectos de red.
4. 04.XX Interfaz MS-BS y protocolos.
5. 05.XX Capa física en el trayecto radioeléctrico.
6. 06.XX Codificación de voz.
7. 07.XX Adaptadores de terminal para MS.
8. 08.XX Interfaces BS-MSC.
9. 09.XX Interfuncionamiento de la red.
10. 10.XX Interfuncionamiento de los servicios.

⁶<http://www.etsi.org>

11. 11.XX Especificaciones y homologación de los equipos.
12. 12.XX Operación y mantenimiento.

Frecuencia Offset y ARFCN: La interfaz de radio GSM combina la técnica de acceso por división de frecuencia con la técnica de acceso por división de tiempo y realiza duplexado por frecuencia de manera que GSM utiliza la Técnica de Acceso FDD/FDMA/TDMA, el uso de FDD divide la banda de frecuencias en dos: Uplink(enlace de subida) y Downlink (enlace de bajada), estas bandas son separadas por una frecuencia offset y son identificadas por un numero llamado ARFCN(*Absolute Radio Frequency Channel Number*, Numero absoluto de canal RF) [5].

	GSM 450	GSM 850	GSM 900	GSM 1800	GSM 1900
Offset	10 Mhz	45 Mhz	45 Mhz	95 Mhz	80 Mhz
ARFCN	259-293	128-251	1-124	512-885	512-810

Tabla 1.2: Frecuencia Offset y ARFCN de GSM, Tomado de [5]

Arquitectura

Los elementos más importantes de la arquitectura GSM son la estación móvil, el subsistema de estación base, el subsistema de conmutación de red y el centro de manejo de redes (figura 1.1) [6].

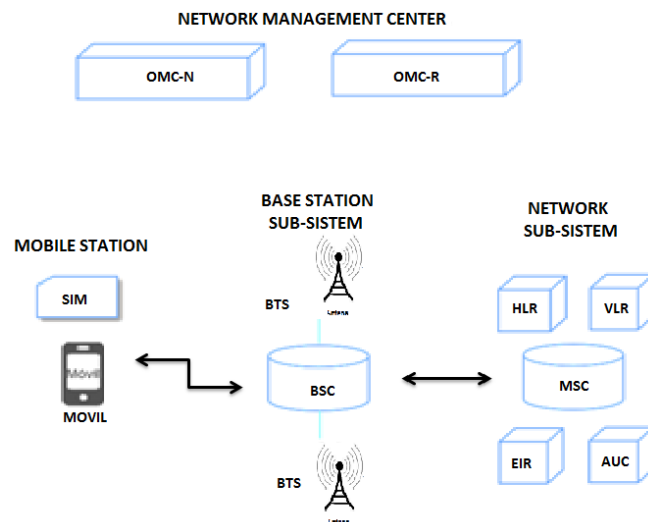


Figura 1.1: Arquitectura GSM.

Estos elementos de la red GSM se comunican por medio de interfaces:

1. Um: para establecer comunicación entre la MS Y BTS, también es llamada interfaz de aire.
2. A-bis: para establecer comunicación entre la BTS y BSC, intercambia información sobre la gestión de llamadas, la movilidad y los recursos de radio.
3. A: para establecer comunicación entre la BSC Y MSC.

Los elementos o componentes se describen a continuación.

Estación móvil: Dispositivo que permite acceder a los servicios que ofrece la red y está conformado por las terminales y la tarjeta SIM: Las terminales se identifican por medio del identificador internacional de dispositivos móviles (IMEI); la tarjeta SIM ó Modulo de identificación de abonado almacena el IMSI "identificador internacional de abonados móviles" que identifica de manera única a la SIM en la red y almacena las claves, cifrado y autenticación del usuario en el sistema.

IMSI			
MCC	Mobile Country Code	2,3 cifras	Identifica el código del país.
MNC	Mobile Network Code	2 cifras	Número de la red
MSIN	Mobile station Identification Number	13 cifras máximo	Identifica la estación móvil

Tabla 1.3: Identificador internacional de abonados móviles.

Subsistema de estación Base: Compuesta por el controlador de estaciones base y sus estaciones base conectadas.

1. Controlador de estaciones base, a este están conectados todas las estaciones base de la red y es el encargado de administrar la distribución de los registros.
2. Estaciones base (BTS), transceptor de estación base, encargado de la gestión de canales de radio.

Subsistema de conmutación de red: Gestiona las identidades de los usuarios, su ubicación y el establecimiento de comunicaciones con otros usuarios. Hacen parte de el:

1. Centro de conmutación móvil (MSC): A este están físicamente conectados los controladores de estación base; que los conecta con la red de telefonía pública y con Internet.
2. Home location Register (HLR): o registro de ubicación de origen, es la base de datos que contiene la información administrativa sobre el usuario del servicio y la localización actual del terminal.

3. Visitor location Register (VLC): o registro de ubicación del visitante; Es una base de datos con información de usuarios que no son abonados locales; el VLC bloquea a un usuario cuando este intenta acceder a un servicio al que tiene restricciones.
4. Authentication Center (AUC): o centro de autenticación; este verifica si un usuario que intenta acceder al servicio de GSM, tiene dicho acceso por medio del SIM ya que este le envía el IMSI, evitando que clonen las SIM.
5. Equipment Identity Register (EIR): o Registro de identificación del equipo; base de datos que contiene la lista de terminales móviles.

Network management center ó Operation and maintenance center: accesa remotamente los componentes de la red (BSS, MSC, VLR, HLR Y AUC), verifica que toda la red, este funcionando correctamente, además de recoger los datos de los usuarios para hacer la facturación.

1.3.2. Radio Definido por Software (SDR)

Radio Definido por Software fue creado por Joseph Mitola en 1991 para referirse a radios reconfigurables, integrada de una computadora o un dispositivo de computación embebida; cuyo objetivo se basa en controlar dispositivos de radio comunicaciones o radares a fin de permitir la integración e interoperabilidad de diversas arquitecturas de comunicación; además de servir como solución a problemas como el ruido y las interferencias generados por los equipos hardware utilizados en sistemas de radiocomunicaciones.

SDR es una tecnología compuesta de hardware y software, donde la mayoría de las funciones de la capa física son definidas mediante software, lo que permite la integración de diferentes arquitecturas dando un mejor uso de recursos, como los protocolos de las capas superiores del modelo OSI ⁷, haciéndolo más óptimo y reconfigurable. Este sistema se caracteriza principalmente por su capacidad de configuración, actualización y adaptación [7].

Arquitectura

Está conformado por bloques como se ilustra en la figura 1.2:

Bloque de Radiofrecuencia RF: El bloque de Radiofrecuencia RF, se encarga de recibir las señales a través de la antena para adecuarlas y convertirlas a frecuencia intermedia y la conforman: [8]

1. **La antena:** es un dispositivo o circuito eléctrico especial, que permite emitir ondas electromagnéticas al espacio libre.

⁷http://es.wikipedia.org/wiki/Modelo_OSI

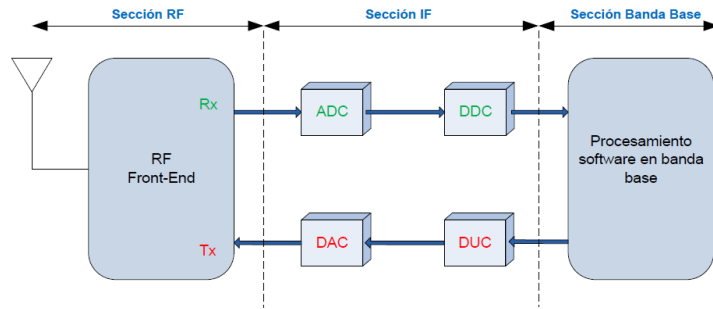


Figura 1.2: Arquitectura SDR, Tomado de [8]

2. **RF Frond-End:** Realiza una conversión o adaptación de frecuencias, es el encargado de la adaptación de nivel y compuesto por el transceptor que cuenta con dispositivos electrónicos que procesan el nivel de las señales para que sea adecuado a la función siguiente. Estos procesan las señales GSM a la frecuencia de transmisión/recepción del orden de 1.000 MHz. Cuando se trasmite se produce una amplificación de la señal entregada por las etapas anteriores hasta el nivel de potencia suficiente para su transmisión por el medio físico. El nivel de potencia en telefonía GSM es del orden de 2W como máximo (+ 33dBm).
3. **Oscilador Local:** es el encargado de generar las frecuencias apropiadas para convertir la frecuencia de RF en la frecuencia intermedia FI, mediante una mezcla no lineal que produce frecuencias de suma y resta. Se selecciona la frecuencia deseada mediante filtros analógicos para su amplificación en los amplificadores de frecuencia intermedia.

Bloque de Frecuencia Intermedia (FI): El bloque de Frecuencia Intermedia FI, digitaliza la señal y se traslada de forma digital a banda base gracias al módulo DDC. La mayor parte de la amplificación en recepción se produce a una frecuencia inferior, llamada frecuencia intermedia. El bloque de Frecuencia Intermedia cumple estas funciones:

1. **Conversión AD/DA:** Tomando en cuenta que la transmisión por el medio físico se realiza mediante señales analógicas, pero el procesamiento en el Transceiver es de índole digital, se hace imprescindible realizar una conversión analógica/digital en el receptor y digital/analógica en el transmisor. Está compuesta por:

ADC *Digital-to-analog converter*, el convertidor analógico a digital, es el encargado de entregar un valor binario de salida a partir de una entrada analógica de voltaje, e involucra los siguientes procesos de muestreo; Tasa de muestreo que consiste en toma muestras periódicas de la amplitud de la señal analógica, cuantificación que mide el nivel de voltaje de cada

muestra y le asigna un valor numérico de salida y codificación que traduce los valores obtenidos durante la cuantificación en código binario.

DAC *Analogo-to-digital Converter*, es el encargado de convertir la señal de FI digital a analógica.

Bloque Banda Base: Aquí se lleva a cabo todo el procesamiento digital de la señal, se encarga de todo el procesamiento en banda base de la señal como: modulación/demodulación, análisis espectral de la señal, salto de frecuencia, establecimiento de sesión, ecualización, manejo de tiempos de bit, entre otros [9].

DDC *Digital Down Converter*, es el encargado de transformar la señal de Frecuencia intermedia digital a Banda base y se compone de un oscilador local, un mezclador digital, y filtros pasábandas digitales.

DUC *Digital Up Converter*, es el encargado de transformar la señal de Banda base a Frecuencia intermedia digital y se compone de un oscilador local, un mezclador digital, y filtros pasábandas digitales.

Modulador/Demodulador: Esta parte se encarga de transmitir una información útil mediante una onda electromagnética que se propaga, es necesario imprimir de algún modo esta información sobre una señal portadora. Esto se logra modificando alguno de los parámetros que la definen de acuerdo con el valor de la información a transmitir. Este proceso se denomina modulación.

1.3.3. Voz sobre IP (VoIP)

Es una tecnología que permite la transmisión de la voz a través de redes IP en tiempo real en forma de paquetes de datos. Básicamente VoIP es un método por el cual tomando señales de audio analógicas del tipo de las que se escuchan cuando uno habla por teléfono se las transforma en datos digitales que pueden ser transmitidos sobre redes de internet hacia una dirección IP determinada utilizando un PC, gateways y teléfonos estándares. En general, servicios de comunicación voz, fax, aplicaciones de mensajes de voz que son transportados vía redes IP, Internet normalmente, en lugar de ser transportados vía la red telefónica convencional. La figura 1.3 muestra un diagrama del alcance al que puede llegar la tecnología de VoIP al integrar varios medios de comunicación [10].

Los elementos fundamentales de una red VoIP son:

1. Terminales: Teléfonos IP que pueden ser hardware o software.
2. Central de conmutación: Controla y gestiona los procesos de toda la comunicación de VoIP.

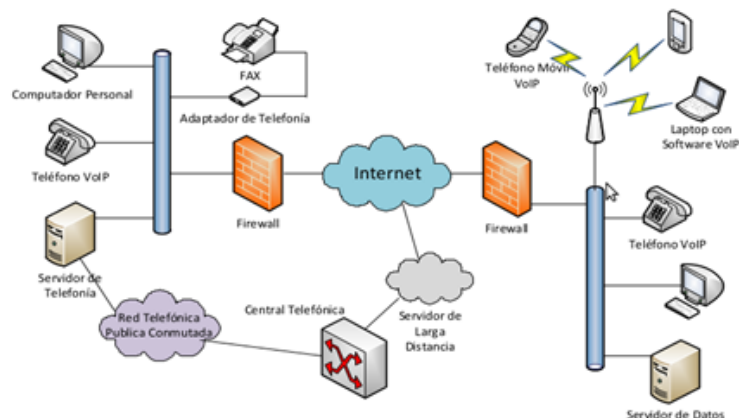


Figura 1.3: Arquitectura VoIP, Tomado de [10]

3. Gateway: Dispositivo que sirve de enlace con otras redes telefónicas. Actúa de forma transparente al usuario.

Durante el momento de una comunicación, se hacen necesarias ciertas reglas que permitan establecer, mantener, administrar y finalizar los elementos que están involucrados en ese momento en la red. Este conjunto de reglas se denominan protocolos y son directamente responsables del funcionamiento de la comunicación, existen varios protocolos comúnmente usados para VoIP. El protocolo más usado es el H.323, un standard creado por la International Telecommunication Union (ITU), H323 es un protocolo muy complejo que fue originalmente diseñado para videoconferencias y actualmente incorpora muchos protocolos individuales que fueron desarrollados para aplicaciones específicas, el problema con H323 es que no fue específicamente dirigido a VoIP, por lo cual surgió una alternativa con el desarrollo del Session Initiation Protocol (SIP)⁸.

Protocolo SIP: Fue desarrollado con el fin de tener una estandarización en la iniciación, modificación y finalización de sesiones interactivas entre usuarios donde interviene uno o más participantes con elementos multimedia como llamadas, vídeos, mensajería instantánea y juegos. El protocolo SIP se concentra en el establecimiento, modificación y terminación de las sesiones, y se complementa entre otros con el SDP, que describe el contenido multimedia de la sesión, por ejemplo qué direcciones IP, puertos y codecs se usarán durante la comunicación. Es un protocolo de señalización. Se complementa con el RTP *Real-time Transport Protocol*, portador del contenido de voz y vídeo que intercambian los participantes en una sesión establecida por SIP. SIP ofrece la posibilidad de programar nuevos servicios no definidos por la propia recomendación, esta ventaja de su flexibilidad es por lo que actualmente es más usado que otros protocolos. Algunas funciones de señalización son:

⁸<http://www.telefoniavoz.com/voip/protocolos-en-la-telefonía-ip.htm>

1. Establecer, modificar y finalizar llamadas
2. Registrar y localizar participantes.
3. Gestión del conjunto de participantes y componentes del sistema

Los clientes SIP llamados peers o agentes de usuario usan los protocolos TCP (Protocolo de Control Transmisión) y UDP (Protocolo de Datagramas de usuario) para conectar con los servidores SIP. SIP es usado simplemente para iniciar y terminar sesiones multimedia. Todas las comunicaciones en SIP usan el Protocolo de Transporte en Tiempo Real (RTP).

Codecs en telefonía IP: los codecs realizan la tarea de conversión digital tomando muestras de la señal de audio miles de veces por segundo. Por ejemplo, el codec que es el que mas se utiliza G.711 toma 64,000 muestras por segundo. Convierte cada pequeña muestra en información digital y lo comprime para su transmisión. Cuando las 64,000 muestras son reconstruidas, los pedacitos de audio que se pierden entre medio de estas son tan pequeños que es imposible para el oído humano notar esta perdida, esta suena como una sucesión continua de audio. Existen diferentes frecuencias de muestreo de la señal en VOIP, esto depende del codec que se este usando [11].

1. 64,000 veces por segundo
2. 32,000 veces por segundo
3. 8,000 veces por segundo

Capítulo 2

Implementación de una celda de red GSM con OpenBTS

2.1. Elementos de software y hardware utilizados

2.1.1. GNU Radio

Este marco de desarrollo de software de código abierto fue iniciado en 2001, por el Filántropo John Gilmore, quien ha sostenido el proyecto de Radio GNU con el financiamiento de \$ 320,000 (EE.UU.) de Eric Blossom para tareas de creación de código y gestión de proyectos [12].

GNU Radio es un software de desarrollo de herramientas de código libre y abierto que proporciona bloques de procesamiento de señales para implementar software radio. Este puede ser usado con hardware de RF externo de bajo costo para crear radios definidos por software o sin hardware utilizando el entorno de simulación. Es ampliamente utilizado en entornos de aficionados, académicos y comercial para contribuir tanto en la investigación de comunicaciones inalámbricas y sistemas de radio del mundo real.

GNU Radio está licenciado bajo la *General Public License* (GPL) GNU versión 3. Todo el código es propiedad de la Fundación del Software Libre. Esta licencia permite ejecutar este software con cualquier propósito y permite modificar el programa.

Las aplicaciones de GNU Radio son escritas a principio utilizando el lenguaje de programación Python, mientras que el suministro de herramientas críticas de procesamiento de señales que requieren alto rendimiento son implementados en C++ usando extensiones de procesamiento de punto flotante, cuando este esta disponible. Así, el desarrollador es capaz de implementar, de manera simple, sistemas de radio de alto rendimiento funcionando a tiempo

real aprovechando el ambiente de desarrollo de aplicaciones de manera inmediata. [13]

GNU Radio realiza todo el procesamiento de la señal. Se puede utilizar para escribir aplicaciones para recibir los datos de flujos digitales o para enviar datos en flujos digitales. GNU Radio tiene filtros, códigos de canal, elementos de sincronización, ecualizadores, demoduladores, vocoders, decodificadores, y muchos otros elementos (en la jerga GNU Radio, llamamos a estos elementos de bloques) que típicamente se encuentran en los sistemas de radio. Más importante aún, se incluye un método para conectar estos bloques y luego se administra la forma se hace pasar datos de un bloque a otro. Extendiendo GNU Radio también es bastante fácil, si usted encuentra un bloque específico que falta, puede crear de forma rápida y agregarlo.

GNU Radio, sólo puede manejar datos digitales. Por lo general, las muestras de banda de base complejas son el tipo de datos de entrada para los receptores y el tipo de datos de salida para los transmisores. Los tipos de datos que se puede pasar de un bloque a otro son: bit, byte, vectores o tipos de datos más complejos. El hardware analógico se usa para desplazar la señal a la frecuencia central deseada.

La empresa Ettus Research ¹ como parte del proyecto de GNU-Radio desarrollo un hardware de adquisición y transmisión de señales que permite trabajar en varias bandas de radiofrecuencias mediante módulos intercambiables que definen la banda de operación y las características de la sección de radio frecuencia, a fin de emular en un entorno real sistemas de comunicaciones y sistemas de radio.

2.1.2. OpenBTS

OpenBTS es un proyecto actualmente mantenido por la compañía Range Network, fundada por David Burgess y Harvind Samra, desarrolladores Originales del software OpenBTS, que busca crear un nuevo tipo de red inalámbrica que promete ampliar la cobertura a los mercados desatendidos, mientras desata una plataforma para la innovación [14].

El software OpenBTS *Open Base Transceiver Station* es una aplicación de unix que utiliza un radio definido por software para presentar una interfaz de aire GSM Um^2 a los dispositivos de los usuarios, la interfaz de aire es construida al utilizar el Hardware USRP *Universal Software Radio Peripheral*; También cabe resaltar que estos dispositivos son presentados como terminales SIP a internet para ello usa un softswitch SIP o PBX.

¹<http://www.ettus.com>

²http://en.wikipedia.org/wiki/Um_interface

La configuración de OpenBTS reside en un archivo maestro con el nombre de (OpenBTS.config), el cual se encuentra localizado en el directorio (/apps) de la raíz de la instalación de OpenBTS.

OpenBTS se distribuye en dos formas: [15]

“El lanzamiento público. El cual se distribuye bajo la licencia AGPLv3 (licencia pública general de Affero, tercera versión) con derechos de autor asignados a la FSF (Fundación para el software libre). El cual se utilizó para propósitos de experimentación, educación, evaluación y prueba de conceptos para proyectos de investigación.”

“El lanzamiento comercial. Es instalado en los productos de Range Networks bajo una mezcla de licencias de GPL y otras que no son GPL. El código fuente de los componentes de la instalación de OpenBTS licenciados bajo la licencia GPL está disponible para los clientes comerciales. El lanzamiento comercial proporciona características adicionales de seguridad, escalabilidad y la operación de redes multi-BTS⁴⁷. La versión comercial está destinada a los siguientes tipos de usuarios:

1. Que necesitan proporcionar servicios de telefonía móvil en aplicaciones industriales, gubernamentales o comerciales.
2. Cuando el modelo de negocio es incompatible con la licencia A/GPLv3, o
3. Cuando requieran soporte comercial, monitoreo de redes u otros servicios profesionales.”

Arquitectura OpenBTS

Los elementos más importantes de la arquitectura de OpenBTS son los bloques Transceiver, Sipauthserve, Smqueue y las bases de datos OpenBTS.db, Sqlite3.db, Sipauthserve.db y Smqueue.db, que interactúan entre sí, junto a una PBX para conformar un punto de acceso GSM. Mantiene conexiones SIP representadas con líneas de color negro, conexiones sqlite3 representadas con las líneas de color rojo y la línea azul que identifica la conexión ODBC (figura 2.1).

OpenBTS (Transceiver): El transceiver es un radiomodem basado en software que realiza las funciones de la especificación 05.05 de GSM, responsable del envío y recepción de la señal de radio *Um*, este cumple con dos funciones importantes que son la implementación de la capa física L1 con la plataforma TDMA a través de L3 de la capa de aplicación; a la vez que maneja la interfaz

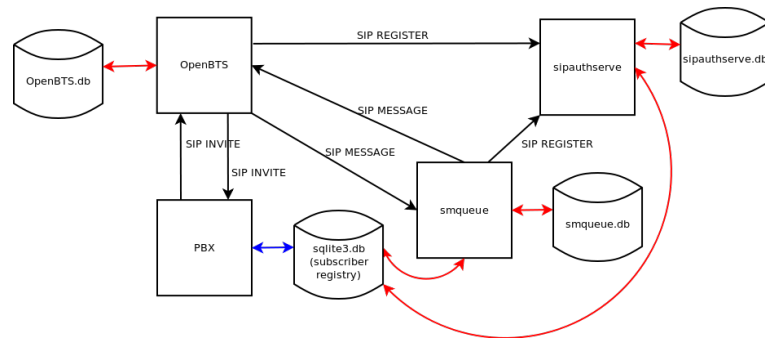


Figura 2.1: Arquitectura OpenBTS, Tomado de [15]

USB del hardware USRP. Su interfaz SIP está normalmente en el puerto 5062, su ubicación por defecto esta en OpenBTS/trunk.

Sipauthserve: Es una aplicación que administra la base de datos de información del suscriptor, cumpliendo de forma análoga con las funciones ejecutadas por el HLR *Home Location Register* de GSM, sus funciones involucran el registro y autorización de usuarios, esta aplicación actualiza la base de datos (sqlite3.db) de estos usuarios en respuesta a los cambios efectuados. Su interfaz SIP normalmente se ejecuta en el puerto 5064, su directorio de ubicación se encuentra en subscriberRegistry/trunk.

Smqueue: Servidor usado para prestar el servicio de mensajes cortos de texto (SMS) en sistemas OpenBTS, es necesario iniciar de forma independiente del transceiver de OpenBTS. Su interfaz SIP normalmente se ejecuta en el puerto 5063. Smqueue no es necesario en instalaciones que no sean compatibles con la mensajería de texto. Su directorio de ubicación esta en smqueue/trunk .

PBX (Asterisk): Referenciado en la sección 2.1.4 del presente trabajo de grado.

Bases de datos[16]

1. OpenBTS.db: Alberga los parámetros de configuración de OpenBTS, los cuales se pueden definir desde el CLI de OpenBTS. Entre los parámetros tenemos: potencia, Banda de frecuencia para GSM, número de canal de radiofrecuencia absoluto (ARFCN), puertos, entre otros. Su ubicación por defecto se encuentra en /etc/OpenBTS/OpenBTS.db.
2. Sqlite3.db Subscriber Registry (SR): Es la red de registro sip, como una red de registro SIP convencional, pero aumentada para dar soporte de movilidad y funciones de autenticación asociadas con GSM. Esta base de datos puede ser manipulada directamente usando sintaxis SQL en tiempo real. En ella se encuentra la información del Subscriber como el IMSI, el

número de extensión dado, puerto, códec, dirección IP entre otros. Su ubicación por defecto se encuentra en `/var/lib/asterisk/sqlite3dir/sqlite3.db`.

3. `Sipauthserve.db`: En ella se encuentran almacenados los parámetros de configuración del Subscriber Registry. Su ubicación por defecto se encuentra en `/etc/OpenBTS/sipauthserve.db`
4. `Smqueue.db`: Base de datos para servicio de mensajes. Su ubicación por defecto se encuentra en `/etc/OpenBTS/smqueue.db`

Funcionamiento de la arquitectura OpenBTS:

Su estructura se basa en la pila de protocolos de GSM, por capas, utilizando el modelo OSI (*Open System Interconnection* Sistema Abierto de interconexión)[16]. Las capas se describen a continuación en forma jerárquica [17].

Capa Física - *Physical Layer* (L1): representa las funciones necesarias para transferir cadenas de bits sobre el medio físico, transmite tramas de control y tráfico. Entre sus funciones está la de Radiomodem, TDM (*Time Division Multiplexing*, Multiplexación por División de Tiempo) y sincronización; y por último codificación, descritas en las series 04.04 y 05.xx de GSM.

Capa de Enlace de Datos - *Data Link Layer* (L2): Provee conexiones de enlace para intercambiar señalización entre las diferentes entidades como MS y BTS. Entre sus funciones está el direccionamiento, segmentación y retransmisión (*LAPDm Link Access Protocol for Dm-channel* Protocolo para Control de Enlace en Canales Dm), descritas en las series 04.05 y 04.06 de GSM y ITU-T Q.921.

Las funciones de la capa 3 están diseñadas como la capa de aplicación y no debe confundirse con las funciones de la capa 3 del Modelo OSI.

Capa de aplicación - *Layer 3* (L3): La capa de aplicación se compone de tres subcapas: RR *Radio Resource* Recursos de Radio: que se encarga de generar el enlace entre la MS y La BTS, MM *Movility Management* Gestión de movilidad que se encarga de administrar la actualización de localización y los procedimientos de registro y CM *Call Management* Administración de llamada.

Entre sus funciones está la administración de la conexión y señalización, descritas en las series 04.07, 04.08, 04.10, 04.11, 04.12 de GSM y ITU-T Q.93150.

2.1.3. USRP N210

Universal Software Radio Pheripheral, es una plataforma flexible de bajo costo para sistemas SDR *Software Defined Radio* desarrollada por Matt

Ettus como una propuesta de hardware libre, donde los microprocesadores convencionales pueden actuar como dispositivos de radio bajo un gran ancho de banda, diseñado para trabajar con un equipo externo (computador) con el procesamiento de señales específicas, que permita implementar de forma rápida, sistemas flexibles y potentes de Radio Definido por Software en tiempo real [18].

Las secciones RF y de IF se llevan a cabo en el dispositivo USRP. En la sección RF se adecuan las señales de frecuencias altas que se reciben del medio y las convierte en frecuencias intermedias para la salida y para que además los ADCs puedan procesar la señal de radiofrecuencia, todo lo anterior en el caso de la recepción. En la transmisión, las señales son amplificadas y moduladas adecuándolas en la frecuencia intermedia deseada para que puedan ser transmitidas.

En la sección IF se lleva a cabo la digitalización y paso a banda base de la señal por medio de ADC's y DDCs *Digital Down Converters*, Conversores Digitales de Bajada en el caso de la recepción, para la transmisión se cambia la señal de banda base a frecuencia intermedia a través de los DUCs (*Digital Up Converters*, Conversores Digitales de Subida),y posteriormente a través de los DACs para pasar a una frecuencia deseada de forma analógica [19].

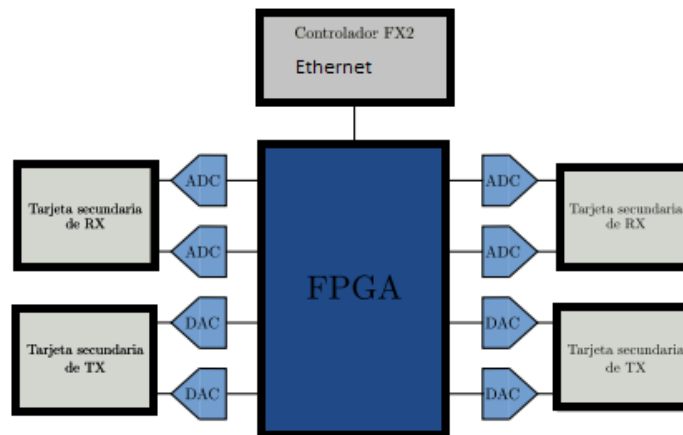


Figura 2.2: Diagrama de bloque USRP tomado de [19]

Es un equipo SDR de hardware libre que ofrece Ettus Research el cual está diseñado para trabajar en proyectos de un gran ancho de banda que en conjunto con un computador permite establecer sistemas de radiocomunicaciones potentes y flexibles con aplicaciones en tiempo real [20].

Su arquitectura cuenta con una FPGA Xilinx Spartan 3A - DSP 3400 que se conecta con un chip Gigabit Ethernet que sirve como interfaz de conexión al

computador con un ancho de banda de 50 MHz-100 MHz usando una cuantización de 8 bits. El FPGA realiza un procesamiento a alta velocidad y reduce la tasa de datos para que puedan ser enviados al computador. La configuración FPGA incluye un conversor DAC de doble canal de 16 bits de resolución y 400 MS/s de tasa de muestreo, con un ADC de doble canal de resolución 14bits y una tasa de muestreo de 100 MS/s, con DDC y DUC de resolución 25 MHz, soportando mediante el puerto de expansión la sincronización de varios USRP N210 y la configuración 2x2 MIMO.³

El panel frontal del USRP N210 contiene una serie de LEDs, los cuáles se presentan y enuncian a continuación [21].



Figura 2.3: USRP N210, Tomado de [21]

1. LED A: este led indica que se está transmitiendo.
2. LED B: este led se indica si se está utilizando MIMO.
3. LED C: este led indica que se está recibiendo.
4. LED D: este led indica que el firmware está cargado.
5. LED E: este led indica cuando la referencia está en estado locked.
6. LED F: este led indica cuando el Complex Programmable Logic Device está cargado

También es en el panel frontal donde se puede encontrar 4 conectores SMA:

1. REF1 y REF2: Estos son los encargados de interconectar la señal de RF con la daughterboard para actuar bien como transmisor o bien como receptor.
2. SMA: Es para utilizar una señal de reloj de referencia (REF IN) externa de 10 MHz.
3. PPS IN Pulse-Per-Second: Se utiliza para mejorar la sincronización de la señal.

³<http://es.wikipedia.org/wiki/MIMO>

Tarjeta madre (Motherboard): esta placa está constituida por el FPGA y los conversores de doble canal y es la encargada de comunicar la señal que es generada por el computador hacia una tarjeta secundaria que hace parte del módulo de RF y hace que la señal llegue intacta para poder realizar la comunicación (figura 2.4).

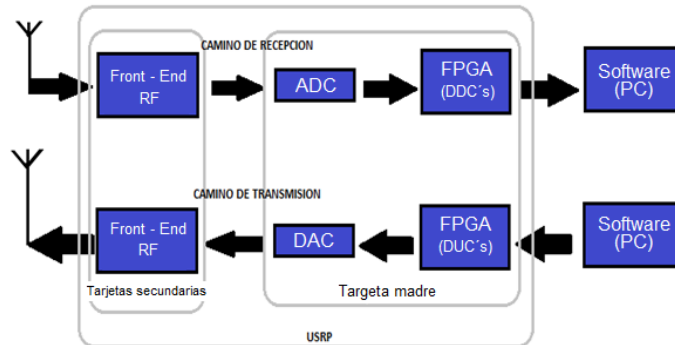


Figura 2.4: Funciones placa madre.

En el caso de la transmisión, la función de la tarjeta madre comienza cuando recibe la señal del computador, realiza el proceso de conversión de la señal de banda base a FI digital, posteriormente es conducida al DAC, quien entrega a la tarjeta secundaria una señal de FI analógica, por otra parte el caso de la recepción su función empieza cuando la señal sale de la tarjeta secundaria y termina cuando esa señal está convertida en forma digital y es dirigida al computador.

Para comprender el desempeño de la tarjeta madre del dispositivo USRP N210 se analizaran algunos detalles de la función que cumple dentro de la arquitectura del hardware (figura 2.5).

Cuando se desea transmitir alguna señal, esta es generada por el host a través del software y será enviado al USRP entrelazando los componentes de fase y cuadratura de la señal. Cuando la señal llega al puerto Ethernet es conducida a la FPGA que es la encargada de separar sus componentes y de realizar operaciones sobre el ancho de banda mediante el uso de filtros. Además realiza los desplazamientos en frecuencia de la señal a través de los DUC (*Digital Up Converters*), luego la señal es conducida a los conversores de digital a analógico para ser transmitida a la tarjeta secundaria.

En el caso de la recepción la señal llega a los ADC procedente de la tarjeta secundaria donde se hace la conversión de tipo analógico a tipo digital y es entregada al FPGA que se encarga de desplazar las frecuencias a través del

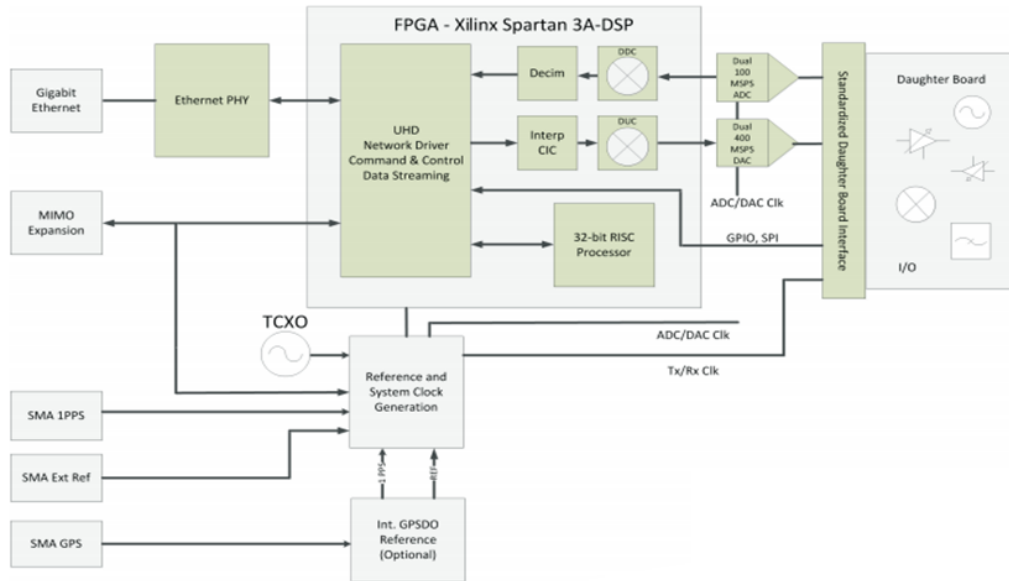


Figura 2.5: tarjeta madre USRP N210, Tomado de [21]

DDC *Digital Down Converters* y de realizar las distintas operaciones sobre el ancho de banda necesario para el funcionamiento y entrelazar los componentes requeridos.

El convertor analógico digital (ads62p44) presenta una ganancia programable de 6,5 dB, realizando el ajuste grueso en pasos de 0,5 dB (hasta 6 dB) y el ajuste fino en pasos de 0,1 dB (hasta 0,5 dB). Mientras que el convertor digital analógico no presenta ganancia configurable [22].

USRP N210 está diseñado para que el DUC lleve a cabo el traslado de frecuencia desde la banda base hasta la frecuencia intermedia y posteriormente en la tarjeta secundaria se lleva a cabo el traslado de la frecuencia intermedia hasta la radio frecuencia. Para el caso de la transmisión y, en la recepción, de forma contraria, la radio frecuencia es trasladada a la frecuencia intermedia y a través de DDC se obtiene la frecuencia base. El dispositivo GPSDO (Global Positioning System Disciplined Oscillator) [23], permite una sincronización precisa de muestras para transmisión Tx y recepción Rx en el dispositivo USRP, generando una frecuencia de reloj de 10MHz en su oscilador de salida.

Tarjeta secundaria: También conocidas como tarjetas hijas, son las encargadas de que las funciones del transmisor y el receptor se cumplan, de tal manera que su trabajo con respecto a la transmisión empieza desde que la señal del DAC y termina cuando esta señal es conducida hasta el conector SMA, el cual se utiliza para acoplar una antena o bien conectar señales de entrada. Respecto a la recepción, su trabajo empieza desde que la señal llega

al conector SMA y termina cuando es conducida al ADC. Estas tarjetas secundarias se conectan a la tarjeta madre ya que cuenta con cuatro entradas, dos para recepción y dos para transmisión.

Antenas: Debido a la proyección de este trabajo en el ámbito de la telefonía celular, se hizo uso de la antena VERT900 de Ettus Research, es una antena omnidireccional adecuada para transmitir la señal de datos al medio inalámbrico, tiene una ganancia de 3dBi y además trabaja en las bandas Celular/PCS e ISM (industrial, Scientific and Medical) operando en la banda de frecuencias de 824-960.

2.1.4. Asterisk

Asterisk es una plataforma de software libre y código abierto que fue inicialmente desarrollada en la empresa estadounidense Digium por Mark Spencer, bajo el sistema operativo GNU/Linux y liberada con la licencia GPL2 *General Public License 2*.

Asterisk es una plataforma para telefonía, la cual es implementada en un computador y trabaja como un servidor de comunicaciones para realizar control y conmutación de llamadas a la vez que permite implementar aplicaciones y servicios. Trabaja con casi todos los protocolos de señalización SIP como SIP, IAX2, MGCP y Cisco Skinny.

A pesar de que se trata de simplemente una aplicación software, esta plataforma proporciona todas las funcionalidades, desde las más básicas hasta las más avanzadas, y establece los servicios de una central secundaria privada automática. Asterisk soporta distintas tecnologías que permiten hacer o recibir llamadas telefónicas al igual que distintos protocolos de voz sobre IP, así como conectividad analógica y digital, dependiendo de las redes de telefonía de acuerdo a la necesidad del servicio. Asterisk al ser de código abierto tiene muchas características adicionales las cuales permiten personalizar el procesamiento de las distintas llamadas telefónicas.

Arquitectura Asterisk

A diferencia de las PBX tradicionales, asterisk es totalmente independiente de los dispositivos externos y todo entra al sistema como un canal genérico y aunque cada dispositivo tenga características diferentes siempre se gestionan de la misma forma.

La arquitectura de asterisk está basada en un sistema modular, que depende del núcleo principal de sistema, lo que permite que cada usuario pueda

seleccionar que partes de asterisk o módulos desea utilizar. Cada módulo posee una funcionalidad específica, de tal forma que puedan tratarse todos los aspectos del sistema, pasando por los tipos de canales (SIP, IAX, DAHDI) o conexiones a otros sistemas para interactuar con Asterisk (mail, bases de datos, web, etc.) que permiten a asterisk manejar un dispositivo de alguna determinada tecnología. Por ejemplo, para manejar dispositivos SIP se utiliza el módulo Chan_SIP, para IAX2 se utiliza Chan_IAX y para canales analógicos y digitales Chan_ZAP. El ejemplo más común es cuando se realiza una llamada de un teléfono a un sistema asterisk, esta conexión está representada por solo un canal (channel SIP). Estos módulos se clasifican en varias categorías, algunas de las más comunes:

1. Aplicaciones: son aquellas acciones aplicables al manejo de las llamadas dentro del Plan de Marcación en el archivo *extensión.conf* de tal forma que puedan definirse acciones que pueden utilizarse o escogerse dentro de una llamada. Por ejemplo, la más popular en todos los planes sería la aplicación Dial, la cual simplemente tiene como propósito lanzar una llamada a un canal en función de las propiedades que señalemos durante su ejecución. El número marcado es usado para encontrar la extensión en el plan de marcado que se usará para procesar la llamada. Las extensiones incluyen una lista de aplicaciones de plan de marcado las cuales serán ejecutadas sobre el canal. Existen otras aplicaciones comunes, como *VoiceMail* encargada de la gestión del Buzón de Voz, *Record* para grabar el sonido de un determinado canal, etc [24].

Algunas características en común de las aplicaciones son las siguientes:

- . Las acciones están exclusivamente enfocadas por y para los canales.
- . Se carga de forma dinámica
- . Se ejecutan de manera síncrona.
- . El formato clásico de este tipo de módulos es *app_ < nombre > .so*

2. Recursos: La función específica de los recursos es la de integrar Asterisk con los sistemas externos. Hablamos de bases de datos, servidores web, calendarios, etc. Aportan funcionalidades o recursos adicionales al núcleo. Uno de los más comunes, es el recurso para ofrecer servicios de Música en Espera (Music ion Hold), o para realizar interconexiones con bases de datos a través de ODBC. El formato clásico de este tipo de módulos es *res_ < nombre > .so*
3. Funciones de plan de marcado: La idea fundamental detrás de las Funciones es la capacidad de obtener o añadir, determinada información específica a cada canal. Suelen ser complementarias a las Aplicaciones proporcionando muchas mejoras útiles como el manejo de cadenas de

caracteres o la conectividad ODBC (Open DataBase Connectivity, Conectividad Abierta de Bases de Datos), haciendo que las aplicaciones de plan de marcado sean más dinámicas. Las funciones no pueden ser llamadas directamente en el plan de marcado, estas son llamadas dentro de las aplicaciones y con letras mayúsculas.

4. Controladores de canales: Son los drivers más importantes y específicos para cada tipo de canal disponible actualmente o en un futuro para Asterisk. Estos son los que aportan específicamente la posibilidad de volver totalmente independiente el sistema de los mismos, para así poder tratarlos de forma totalmente homogénea. Son exactamente una especie de interfaz entre el núcleo de Asterisk y la parte “lógica” dentro del sistema operativo. Sin estos módulos asterisk no tendría forma de realizar llamadas. Cada llamada entra al sistema a través de un controlador de canal, el cual verifica el plan de marcado y asigna un canal de asterisk a la llamada. Todos los canales más típicos, tienen su correspondiente módulo para el driver. Ejemplos típicos son el específico para DAHDI, SIP e IAX, que justamente son los más utilizados por la comunidad Asterisk. El formato clásico de este tipo de módulos es *chan_ < nombre > .so*

5. Traductores de codecs y formatos: Son la representación para los sistemas de audio y vídeo digitales de transmisión (codecs) y almacenamiento (formatos). Son los encargados de convertir vía software, entre un tipo y otro tipo de formato o códec de forma simultánea al curso de la llamada. Asterisk soporta muchos codecs diferentes y ejecuta su traducción cuando sea necesario, esto permite convertir formatos de audios entre llamadas. Por ejemplo, si una llamada, viene del canal DAHDI, con el códec G.711 (alaw o ulaw dependiendo del país) y quiere pasarse a una extensión SIP dentro de nuestro sistema Asterisk, el traductor correspondiente será el encargado de realizar esta conversión en tiempo real. Por otro lado, si lo que estamos tratando son archivos, los encargados de interpretar el contenido para pasarlo a través del audio del canal correspondiente serían los traductores de Formatos. En este ámbito los más populares son los estándares, GSM y WAV en los que están basados la mayor parte de los sonidos por defecto del sistema Asterisk. El formato clásico para los módulos de codecs es *codec_¡nombre¡.so* y para los de formatos *format_ < nombre > .so*

Estructura de Archivos

Los recursos que Asterisk utiliza en cada módulo, necesitan estar alojados en directorios específicos, estos recursos usan muchos directorios sobre el sistema de archivos de Linux con fin de almacenar y administrar distintos

Tipo	Soportados
Códecs audio	ulaw, alaw, gsm, ilbc, speex, g722, g723, g726, g729.
Códecs formato	GSM, PCM, WAV, OGG, SLINEAR,MP3.

Tabla 2.1: Códecs y formatos de audio soportados por Asterisk. tomado de autores

funcionamientos de todo el sistema como por ejemplo las grabaciones de voz, música en espera, archivos de configuración y contestación automática[25].

1. Archivos de configuración: Todos los archivos de configuración se encuentran en el directorio `/etc/asterisk` que se ha creado cuando se haya realizado en la instalación de Asterisk. Algunos archivos de configuración que se utilizarán en este trabajo son `extensions.conf`, `sip.conf`, `manager.conf`, además de otros archivos que definen parámetros para una funcionalidad específica como por ejemplo `voicemail.conf`.
2. Archivo de módulos: Estos archivos son de gran importancia para una actualización de la versión de asterisk ya que es importante conocer donde estas instalados los módulos que se quieren utilizar en algún momento. Usualmente estos módulos se instalan en el directorio `/usr/lib/asterisk/modules`.
3. Almacenamiento temporal: El mensaje de voz, grabaciones de llamadas, llamadas generadas por algunas aplicaciones y algunas otras son informaciones transitorias para las que Asterisk utiliza el almacenamiento temporal. El directorio utilizado para este almacenamientos es `/var/spool/asterisk`.

2.2. Instalación y configuración

Se debe tener en cuenta que para la instalación de estas herramientas se debe haber previamente instalado el sistema operativo Ubuntu 10 o 12.04 LTS para mejores resultados, ya que están mejor sustentadas. En nuestro caso se trabajo con Ubuntu 12.04 LTS.

La construcción de la celda de red GSM se realiza en el siguiente orden.

2.2.1. Instalación de GnuRadio

Para realizar la instalación de GnuRadio, ingresamos a su pagina oficial con el link <http://gnuradio.org/redmine/projects/gnuradio/wiki>, buscamos la opción *Installing GNURadio* en el que se recomienda hacer la instalación utilizando el `build-gnuradio` que ayuda a instalar de manera automática; el script se descarga en el computador.

Posteriormente se crea una carpeta llamada gnuradio para mayor organización, en la cual se copia el script build-gnuradio haciendo uso de las siguientes sentencias.

```
mkdir gnuradio
cp build-gnuradio gnuradio
```

Luego se procede a ejecutar el script. Para ello se cambia las propiedades al ejecutable y posteriormente se ejecuta, este paso toma aproximadamente una hora, dependiendo de la velocidad de red a la que se este conectado. Para ello se hace uso de las siguientes sentencias.

```
chmod a+x build-gnuradio
./build-gnuradio
```

El proceso se realiza a travez de un GIT que descarga los instaladores de gnuradio como los de UHD que es el hardware que maneja el dispositivo USRP 1, en nuestro caso se maneja el USRP N210, esta version hace parte de los nuevos productos presentados por Ettus los cuales usan sincronización mediante el software UDH *Universal Hardware Device* y no mediante driver del software Gnuradio, lo cual produce que el cambio de oscilador ya no sea necesario.

Para confirmar que la instalación se realizo con éxito abrimos la interfaz de usuario grafica GNU Radio Companion desde el directorio gnuradio [26].

```
gnuradio-companion
```

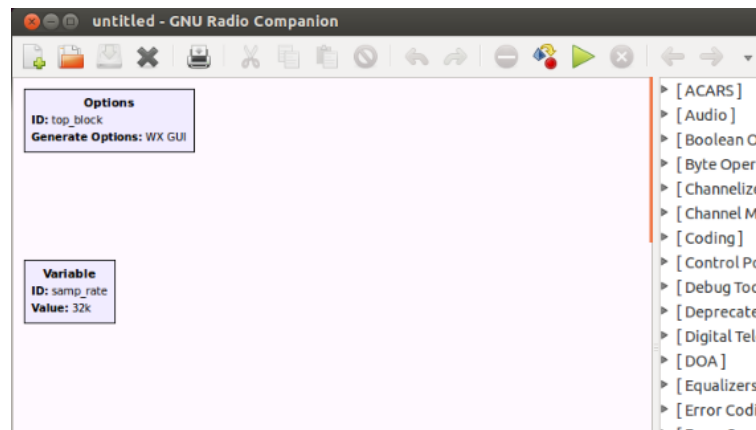


Figura 2.6: Interfaz grafica gnuradio-companion.

2.2.2. Instalación de Asterisk

Para la instalación de Asterisk, es necesario instalar los siguientes módulos, ya contenidos en la lista de repositorios.

1. Asterisk: nos proporciona los ficheros base del proyecto para implementar la PBX. Para instalar Asterisk se utiliza el comando:

```
Sudo apt-get install asterisk
```

2. Dahdi: Es un marco de dispositivos controladores usado por varios proveedores hardware para realizar interfaces entre aplicaciones con hardware de Telefonía, en otras palabras nos permite dar soporte para hardware y drivers de tarjetas. Para instalar Dahdi se utiliza los siguientes comandos:

```
Sudo apt-get install dahdi  
sudo apt-get install asterisk-dahdi
```

3. Libpri: permite dar soporte para conexiones digitales. Para instalar Libpri se utiliza el comando:

```
Sudo apt-get install libpri
```

Después de la instalación, se recomienda ejecutar el siguiente comando *update*. De manera que se modifiquen los archivos *sip.conf* y *extensions.conf* para configurar los clientes SIP y el plan de marcado.

```
sudo apt-get update
```

Para confirmar que la instalacion se realizo con exito se verifica abriendo la Interfaz de Línea de Comandos (CLI). Para abrir la línea de comandos CLI se ejecuta la siguiente línea:

```
sudo Asterisk -rvvv
```

2.2.3. Instalación de OpenBTS

Para realizar la instalación de OpenBTS se procede con la opción *InstallOpenBTSrP313* presentada en la pagina de *rangepublic* y se complementa con consultas adicionales, la razón por la que se toma esta opción obedece a que se desea instalar la ultima version etiquetada, en nuestro caso la *versión 3.1.3*[27].

A continuación se presenta la instalación en su orden.

1. Como primer paso se crea el directorio OpenBTS, ya que se busca una mejor organización al instalar los paquetes, este paso es opcional. Para esto se utiliza el comando:

```
sudo mkdir OpenBTS
```

2. Seguidamente, en el directorio OpenBTS creado se descargan los paquetes que conformaran el sistema a fin de compilar las diferentes partes de OpenBTS exceptuando la PBX, referidos en la pagina de rangepublic y listados a continuación.

```
. autoconf  
. libtool  
. libosip2  
. libortp  
. libusb-1.0  
. g+ +  
. sqlite3  
. libsqlite3-dev (sipauthserve solamente)  
. libreadline6-dev  
. libncurses5-dev
```

Para esto se utiliza el siguiente comando:

```
sudo apt-get install autoconf libtool libosip2-dev libortp-dev  
libusb-1.0-0-dev g++ sqlite3 libsqlite3-dev erlang libreadline6-dev  
libncurses5-dev
```

3. Si el paso anterior se realizo con éxito, debemos encontrar una carpeta llamada public dentro de nuestra carpeta creada OpenBTS, Se accede a esta carpeta y en este directorio se debe entrar a la raíz.

```
cd public  
cd a53/trunk
```

Luego para compilar e instalar liba53 se ejecuta el comando.

```
sudo make install
```

2.2.4. Conexión con Equipo USRP N210

Para la conexión del dispositivo USRP N210 con OpenBTS, es indispensable haber instalado liba53. Luego se ejecutan los pasos a continuación:

1. Se realiza la unión de las partes del dispositivo USRP N210 y se conecta al computador por medio de red alámbrica ethernet, su construcción se encuentra referenciada en el anexo A.

2. Luego se procede a probar que el equipo pueda conectarse a la USRP-N210, para ello se debe configurar en el computador un nuevo perfil de red alambrico en el cual la dirección IP asignada al computador es 192.168.10.3 con mascara de red 255.255.255.0 y sin gateway, debido a que así viene configurada la USRP-N210. Una vez realizado el paso anterior se debe dar ping a la dirección 192.168.10.2 asignada por defecto a la USRP [28].

Notas: se debe quitar el proxy y se debe contar con una tarjeta de red de minimo 1 GB de velocidad.

3. Una vez se tiene exito en esta prueba, se debe entrar al directorio `a53/trunk` e ingresar el comando `uhd_usrp_probe`:

```
cd /public/a53/trunk
uhd_usrp_probe
```

Esto permite realizar la verificación de la instalación del controlador, debe arrojar resultados como los de la fig 2.7, entonces sabremos que la USRP esta funcionando en óptimas condiciones. Este comando también permite apreciar los datos completos del USRP y sus daughterboards.

2.2.5. Habilitación del soporte para dispositivos UHD y creación de enlace simbólico

1. Para habilitar el soporte para dispositivos UHD, se debe ubicar en la raíz del directorio de OpenBTS e ingresar los comandos enunciados a continuación, este proceso toma unos minutos.

```
cd Public/OpenBTS/trunk
autoreconf -i
./configure -with-uhd
Make
```

2. Creado el soporte, se configura un enlace simbólico al ejecutar el `Transceiver52M`, para lo cual se debe ubicar en el directorio `apps` situado en la raíz de OpenBTS y en ella se ejecuta `ln -s ../Transceiver52M/Transceiver`, no debe reportar errores.

```
cd /public/OpenBTS/trunk/apps
ln -s ../Transceiver52M/Transceiver
```

```
root@projectgsm-pc /home/projectgsm/OpenBTS/public/a53
Antennas: TX/RX, RX2, CAL
Sensors: lo_locked
Freq range: 68.750 to 2200.000 MHz
Gain range PGA0: 0.0 to 31.5 step 0.5 dB
Connection Type: IQ
Uses LO offset: No

RX Codec: A
Name: ads62p44
Gain range digital: 0.0 to 6.0 step 0.5 dB
Gain range fine: 0.0 to 0.5 step 0.1 dB

TX DSP: 0
Freq range: -50.000 to 50.000 MHz

TX Dboard: A
ID: WBX v3 (0x0056)
Serial: EBR26SCXW
ID: WBX + Simple GDB, WBX v3 + Simple GDB, WBX v4 + Simple GDB, WBX-
120 + Simple GDB (0x004f)
Serial: EBR26RCGS

TX Frontend: 0
Name: WBXv3 TX+GDB
Antennas: TX/RX, CAL
Sensors: lo_locked
Freq range: 68.750 to 2200.000 MHz
Gain range PGA0: 0.0 to 31.0 step 1.0 dB
Connection Type: IQ
Uses LO offset: No

TX Codec: A
Name: ad9777
Gain Elements: None
root@projectgsm-pc /home/projectgsm/OpenBTS/public/a53#
```

Figura 2.7: Verificación de la instalación del controlador.

2.2.6. Configuración de OpenBTS

1. Para la configuración de OpenBTS se debe crear la base de datos para la configuración de OpenBTS *OpenBTS.db* que debe ser instalada en el directorio */etc/OpenBTS*; este aun no existe, por lo tanto se debe crear utilizando el comando:

```
sudo mkdir /etc/OpenBTS
```

Luego se ubica en el directorio *OpenBTS/trunk* donde se crea el archivo haciendo uso de la siguiente sentencia:

```
sudo sqlite3 -init ./apps/OpenBTS.example.sql  
/etc/OpenBTS/OpenBTS.db ".quit"
```

Si el archivo se ha creado satisfactoriamente, se procede a insertar la configuración por defecto de OpenBTS y además se prueba que el archivo se haya creado adecuadamente mediante el comando:

```
sqlite3 /etc/OpenBTS/OpenBTS.db .dump
```

Si al ejecutar el comando, se observan muchas variables de configuración como se observa en la figura 2.8, significa que el archivo se ha creado correctamente.

2. Para verificar y correr OpenBTS, debemos ubicarnos en el directorio *apps* de OpenBTS e introduce el comando *sudo ./OpenBTS*, para este paso se debe tener conectado el dispositivo USRP. Como respuesta presenta una autorización para usar el CLI.

```
cd /public/OpenBTS/trunk/apps  
sudo ./OpenBTS
```

3. Luego se abre una nueva pestaña en el terminal y nos ubicamos en el directorio *apps* de OpenBTS donde se introducirá el comando *sudo ./OpenBTSCLI*, que abrirá la interfaz de usuario de OpenBTS, se ingresa el comando *config*, donde usted puede realizar la configuración de los parámetros de openBTS.

```
cd /public/OpenBTS/trunk/apps  
sudo ./OpenBTSCLI  
config
```

```
root@projectgsm-pc: /home/projectgsm/OpenBTS/public/openbts/trunk 10:00 projectGSM
INSERT INTO "CONFIG" VALUES('SIP.Timer.A','2000',0,0,'SIP timer A, the INVITE retry period, RFC-3261 Section 17.1.1.2, in milliseconds.');
```

```
INSERT INTO "CONFIG" VALUES('SIP.Timer.B','10000',0,0,'INVITE transaction timeout in milliseconds. This value should usually match GSM.Timer.T3
113.');
```

```
INSERT INTO "CONFIG" VALUES('SIP.Timer.E','500',0,0,'Non-INVITE initial request retransmit period in milliseconds.');
```

```
INSERT INTO "CONFIG" VALUES('SIP.Timer.F','5000',0,0,'Non-INVITE initial request timeout in milliseconds.');
```

```
INSERT INTO "CONFIG" VALUES('SIP.Timer.H','5000',0,0,'ACK timeout period in milliseconds.');
```

```
INSERT INTO "CONFIG" VALUES('SMS.FakeSrcSMSC','0000',0,0,'Use this to fill in L4 SMSC address in SMS delivery.');
```

```
INSERT INTO "CONFIG" VALUES('SMS.MIMEType','application/vnd.3gpp.sms',0,0,'This is the MIME Type that OpenBTS will use for RFC-3428 SIP MESSAGE
payloads. Valid values are "application/vnd.3gpp.sms" and "text/plain.');
```

```
INSERT INTO "CONFIG" VALUES('SubscriberRegistry.A3A8','/OpenBTS/comp128',0,0,'Path to the program that implements the A3/A8 algorithm.');
```

```
INSERT INTO "CONFIG" VALUES('SubscriberRegistry.Manager.Title','Subscriber Registry',0,0,'Title text to be displayed on the subscriber registry
manager.');
```

```
INSERT INTO "CONFIG" VALUES('SubscriberRegistry.Port','5064',0,0,'Port used by the SIP Authentication Server. NOTE: In some older releases (pre-
2.8.1) this is called SIP.myPort.');
```

```
INSERT INTO "CONFIG" VALUES('SubscriberRegistry.UpstreamServer','',0,0,'URL of the subscriber registry HTTP interface on the upstream server. B
y default, this feature is disabled. To enable, specify a server URL eg: http://localhost/cgi/subreg.cgi. To disable again, execute "unconfig
SubscriberRegistry.UpstreamServer.');
```

```
INSERT INTO "CONFIG" VALUES('SubscriberRegistry.db','/var/lib/asterisk/sqlite3dir/sqlite3.db',0,0,'The location of the sqlite3 database holding
the subscriber registry.');
```

```
INSERT INTO "CONFIG" VALUES('TRX.IP','127.0.0.1',1,0,'IP address of the transceiver application. Static.');
```

```
INSERT INTO "CONFIG" VALUES('TRX.MinimumRxRSSI','-63',0,0,'Bursts received at the physical layer below this threshold are automatically ignored.
Values in dB. Set at the factory. Do not adjust without proper calibration.');
```

```
INSERT INTO "CONFIG" VALUES('TRX.Port','5700',1,0,'IP port of the transceiver application. Static.');
```

```
INSERT INTO "CONFIG" VALUES('TRX.RadioFrequencyOffset','128',1,0,'Fine-tuning adjustment for the transceiver master clock. Roughly 170 Hz/step.
Set at the factory. Do not adjust without proper calibration. Static.');
```

```
INSERT INTO "CONFIG" VALUES('TRX.Timeout.Clock','10',0,0,'How long to wait during a read operation from the transceiver before giving up.');
```

```
INSERT INTO "CONFIG" VALUES('TRX.Timeout.Start','2',0,0,'How long to wait during system startup before checking to see if the transceiver can be
reached.');
```

```
INSERT INTO "CONFIG" VALUES('TRX.TxAttenOffset','0',1,0,'Hardware-specific gain adjustment for transmitter, matched to the power amplifier, expe
ssed as an attenuation in dB. Set at the factory. Do not adjust without proper calibration. Static.');
```

```
INSERT INTO "CONFIG" VALUES('TRX.Args','',1,0,'Extra arguments for the Transceiver');
```

```
INSERT INTO "CONFIG" VALUES('Test.GSM.SimulatedFER.Downlink','0',1,0,'Probability (0-100) of dropping any downlink frame to test robustness. St
atic.');
```

```
INSERT INTO "CONFIG" VALUES('Test.GSM.SimulatedFER.Uplink','0',1,0,'Probability (0-100) of dropping any uplink frame to test robustness. Static
.');
```

```
INSERT INTO "CONFIG" VALUES('Test.GSM.UplinkFuzzingRate','0',1,0,'Probability (0-100) of flipping a bit in any uplink frame to test robustness.
Static.');
```

```
INSERT INTO "CONFIG" VALUES('Test.SIP.SimulatedPacketLoss','0',1,0,'Probability (0-100) of dropping any inbound or outbound SIP packet to test r
obustness. Static.');
```

```
COMMIT;
```

```
root@projectgsm-pc: /home/projectgsm/OpenBTS/public/openbts/trunk
```

Figura 2.8: Variables de configuración de OpenBTS.

2.2.7. Instalación de sipauthserve y smqueue

1. El siguiente paso es construir e instalar el registro de suscriptor y Sipauthserve a través del puerto 5062, Sipauthserve es muy importante ya que es el encargado de la autorización SIP para el tráfico de registro. Para ello se debe crear el directorio encargado del registro de suscriptores, este se crea con el comando.

```
sudo Mkdir -p /var/lib/asterisk/sqlite3dir
```

Luego se instala el servidor de autenticación SIP sipauthserve, para ello debemos ubicarnos en la raíz del directorio subscriberRegistry e ingresamos *make* para iniciar la instalación.

```
cd public/openbts/subscriberRegistry/trunk
make
```

Posteriormente se copia la base de datos de suscriptores de ejemplo; para modificar la información de los suscriptores, se utiliza la sentencia:

```
sudo sqlite3 -init subscriberRegistry.example.sql /etc/OpenBTS/
sipauthserve.db ".quit"
```

Se puede probar que haya quedado bien instalado sipauthserve si digitado el siguiente comando lanza un ALERT indicando que el sipauthserve se está iniciando o reiniciando. Cabe aclarar que el comando debe ser utilizado siempre que se quiera iniciar el servidor de autenticación SIP, estando en la carpeta trunk del sipauthserve y dejando quieta la pestaña en la que se ejecuta.

```
sudo ./sipauthserve
```

2. Ahora se procede con la instalación del servidor de mensajería instantánea Smqueue, para lo cual se ingresa al directorio smqueue y se inicia la instalación haciendo uso de los siguientes comandos.

```
cd /public/smqueue/trunk
autoreconf -i
./configure
make
```

3. Para realizar la configuración de smqueue se copia la base de datos de configuración con el uso de la sentencia.

```
sudo sqlite3 -init smqueue.example.sql /etc/OpenBTS/smqueue.db
".quit"
```

4. Se puede probar que haya quedado bien instalado smqueue si digitado el siguiente comando lanza un ALERT indicando que el smqueue se está iniciando o reiniciando. Cabe aclarar que los comandos deben ser utilizados siempre que se quiera iniciar smqueue, estando en la carpeta trunk del smqueue y dejando quieta la pestaña en la que se ejecuta.

```
cd smqueue
sudo ./smqueue
```

2.2.8. Configuración de parámetros de OpenBTS

En esta sección se definen los parámetros para el funcionamiento de la red, la configuración se realiza desde el CLI de OpenBTS [29].

1. La Operación de redes GSM se realiza en varias bandas de frecuencia como son 850Mhz, 900Mhz, 1800Mhz y 1900Mhz, en Colombia se utiliza las bandas GSM-850 y GSM-1900Mhz. Nosotros tomaremos la de 900 Mhz, para ello se digita el siguiente comando:

```
config GSM.Radio.Band #; identifica la banda que se elije.
```

2. Se define el número de canal de radio frecuencia absoluta ARFCN, según la tabla de ARFCN presentada en la sección 1.3.1 del presente trabajo de grado, para ello se digita el siguiente comando:

```
config GSM.Radio.CO #; # identifica el ARFCN seleccionado
```

3. Luego se configura los Codigos MCC y MNC haciendo uso de los siguientes comandos:

```
config GSM.Identity.MCC #; # identifica el código del país
                                (Colombia = 732), también se
                                puede dejar el valor por defecto
                                tiene configurado la USRP N210
                                que es 001
```

```
config GSM.Identity.MNC # ; # identifica al operador, se debe
                                tomar un código que no pertenezca
                                a los operadores locales, también se
                                puede dejar el valor por defecto
                                tiene configurado la USRP N210
                                que es 01.
```

Para el punto anterior se debe verificar teniendo en cuenta la tabla 2.2..

Codigos MNC Colombia	
001	GSM Colombia Telecomunicaciones S A. E.S.P
002	EDATEL S A. E.S.P
003	LLEIDA S.A.S
004	COMPATEL COLOMBIA S.A.S
020	UNE EPM TELECOMUNICACIONES S.A E.S.P. UNE EPM TELCO S.A.
099	EMPRESAS MUNICIPALES DE CALI E.I.C.E.E.S.P.
101	COMUNICACIÓN CELULAR S.A COMCEL S.A
103	COLOMBIA MOVIL S.A E.S.P (TIGO)
111	COLOMBIA MOVIL S.A E.S.P (TIGO)
123	COLOMBIA TELECOMUNICACIONES S.A E.S.P (MOVISTAR)
130	AVANTEL S.A.S

Tabla 2.2: Códigos de identificación de Operadores móviles en Colombia, Tomado de [30]

4. Por ultimo se configura un mensaje de Bienvenida, haciendo uso del siguiente comando:

```
config Control.LUR.OpenRegistration.Message
;Aqui se digita el mensaje de Bienvenida.
```

2.2.9. Registro de terminales en OpenBTS

1. Con la USRP N210 conectada al computador, se inicializa cada uno de los ejecutables en su orden como lo indica la tabla 2.2.

Ubicación del directorio	Comando a ejecutar
/OpenBTS/public/openbts/trunk/apps	./OpenBTS
OpenBTS/public/openbts/trunk/apps	./OpenBTSCLI
OpenBTS/public/subscriberRegistry/trunk	./sipauthserve
OpenBTS/public/smqueue/trunk/smqueue	./smqueue
/etc/asterisk	asterisk -rvvvv

Tabla 2.3: Inicialización de ejecutables para red OpenBTS.

Cuando se abre el CLI de OpenBTS se debe ingresar el siguiente comando.

```
config Control.LUR.OpenRegistration .*
```

Esto habilita la conexión de cualquier celular a la red, para verificar que fue habilitado correctamente se ingresará el comando *config* y se buscará el parámetro *OpenRegistration*.

2. Luego se conectan los Teléfonos a nuestra red, para ello se ingresa a las configuraciones del equipo y ubicas la red, puede mostrarse con el nombre (Range).
3. Para la realización de las pruebas se toma un sótano sin cobertura para que el teléfono no acapare las torres locales y la red no pueda causar interferencias.

Como respuesta de haber realizado la conexión, la red enviara un mensaje del numero 101 con el numero de IMSI *International Mobile Subscriber Identity* del celular, el cual sera necesario en el paso siguiente para configurar las llamadas de voz GSM a través de Asterisk. Luego se debe devolver el mensaje con el numero que daremos a nuestro abonado a este mismo numero.

El paso siguiente es llamar a 600 desde el teléfono, para llamar al eco de servicios, si estas dos pruebas se realizan con éxito significa que se tiene una red funcionando en optimas condiciones.

Capítulo 3

Servicios de comunicación a la red de telefonía móvil a través de Asterisk

3.1. Mensajes y llamadas

Para realizar llamadas es necesario conectar el dispositivo USRP N210 al computador e inicializar las herramientas anteriormente instaladas (OpenBTS, OpenBTSCLI, Asterisk, SipAuthServe, Smqueue), como se describe en la sección 2.2.9, de igual manera se debe conectar el móvil a la red configurada.

El numero IMSI dado por la red, se digita en los archivos de configuración *Sip.conf* y *Extensions.conf* de Asterisk para poder realizar llamadas de voz sobre el protocolo GSM.

En el archivo *Sip.conf* se debe editar la siguiente secuencia de lineas para cada abonado que se desee registrar a fin de modificarlo ubicándonos al final del archivo:

```
[IMSI732XXXXXXXXXX62]
callerid=1212
canreinvite=no
type=friend
allow=gsm
allow=ulaw
allow=alaw
context=sip-external
host=dynamic
dtmfmode=RFC2833
```

```
[IMSI73210XXXXXXXXX44]
callerid=1111
canreinvite=no
type=friend
allow=gsm
allow=ulaw
allow=alaw
context=sip-external
host=dynamic
dtmfmode=RFC2833
```

En *Extension.conf* se debe configurar de la siguiente forma:

```
[macro-dialGSM]
exten => s,1,Dial(SIP/${ARG1},20)
exten => s,2,Goto(s-${DIALSTATUS},1)
exten => s-CANCEL,1,Hangup
exten => s-NOANSWER,1,Hangup
exten => s-BUSY,1,Busy(30)
exten => s-CONGESTION,1,Congestion(30)
exten => s-CHANUNAVAIL,1,playback(ss-noservice)
exten => s-CANCEL,1,Hangup

[sip-external]
exten => 1212,1,Macro(dialGSM,IMSI73210XXXXXXXXX62@127.0.0.1:5062)
exten => 1111,1,Macro(dialGSM,IMSI73210XXXXXXXXX44@127.0.0.1:5062)
```

Después de la configuración de los usuarios, se guardan los cambios realizados en los archivos y se debe reiniciar el servicio de Asterisk para garantizar que los cambios hechos en los archivos surtan efecto en el servicio, para ello se ejecuta el siguiente comando:

```
sudo /etc/init.d/asterisk restart
```

Finalmente, se tendrá un servicio con el cual se podrán realizar llamadas.

Para la configuración de servicios Adicionales se toma como referencia algunas de las practicas presentadas en la asignatura de Laboratorio III de Sistemas de Telecomunicaciones de la Universidad del Cauca¹.

3.2. Configuración del idioma español

Cuando se realiza la instalación de Asterisk por defecto viene configurado con el paquete de idiomas en inglés, es necesario descargar el paquete de

¹<http://www.jano.unicauca.edu.co/lab3sistel>

idiomas en español ².

Luego se ubica el paquete de sonidos descargado en la carpeta de sonidos de Asterisk. Para ello nos ubicamos en el directorio: `usr/share/asterisk/sounds`.

Aquí se crea el directorio `es` y dentro de él los directorios `digits`, `letters` y `phonetic`. Para ello se utilizan las siguientes sentencias.

```
cd /usr/share/asterisk/sounds
sudo mkdir es
cd es
sudo mkdir digits
sudo mkdir letters
sudo mkdir phonetic
```

El paso siguiente es dar permisos ya que los permisos de escritura, lectura y ejecución solo los posee el super-usuario o root. Para dar los permisos se utilizan las sentencias.

```
sudo chmod 777 es
sudo chmod 777 digits
sudo chmod 777 letters
sudo chmod 777 phonetic
```

Luego en el directorio `es` que se creó, se copian los archivos del directorio `es` descargado.

En el directorio `digits` que se creó, se copian lo que está dentro de `digits/es` del paquete descargado.

En el directorio `letters` que se creó, se copian lo que está dentro de `letters/es` del paquete descargado.

En el directorio `phonetic` que se creó, se copian lo que está dentro de `phonetic/es` del paquete descargado.

Para finalizar nos ubicamos en el archivo `sip.conf` se busca la línea `defaultlanguage = en` y se la cambia por `defaultlanguage = es`.

3.3. voicemail

Este servicio es requerido cuando el abonado al que se llama no responde, de manera que permite grabar y almacenar un mensaje, para que luego se pueda acceder al mensaje, llamando a una extensión [31]. Para este servicio es necesario instalar `sendmail`, lo cual lo hacemos con el siguiente comando:

```
sudo apt-get install sendmail
```

²http://www.netsecuritysolutionsltda.com/spanish//index.php?option=com_content&task=view&id=113&Itemid=153

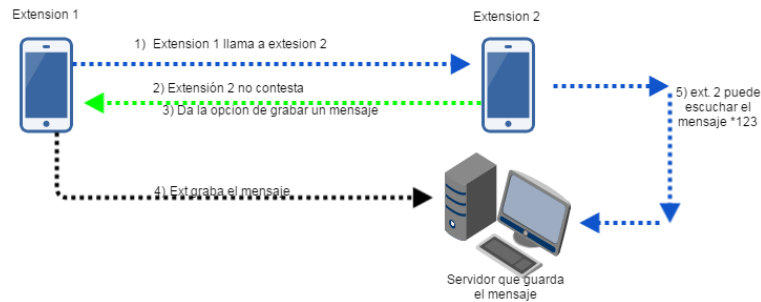


Figura 3.1: Servicio de Voicemail

Para la configuración de este servicio se editan los archivos *extensions.conf*, *voicemail.conf* y *sip.conf*, de manera cómo se ilustra a continuación.

En el archivo *voicemail.conf*, se busca el contexto [general] en el cual se habilitan los siguientes parámetros:

```
format=gsm
attach=yes
maxmsg=100
maxsecs=180
minsecs=3
maxsilence=10
moveheard=yes
```

Para finalizar se crea al final del archivo un contexto llamado en nuestro caso [default], a fin de asignar a cada abonado un buzón de voz.

```
[default]
111 => 111,1212,1212@default
222 => 222,1111,1111@default
```

En el archivo *extensions.conf*, se introduce el buzón y se configura *VoceMailMain* como se ilustra a continuación.

```
exten => 1212,1,Dial(SIP/IMSI73210XXXXXXXXX62@127.0.0.1:5062,15,tT)
exten => 1212,n,VoiceMail(111@default)
exten => 1212,n,Hangup

exten => *321,1,Answer()
exten => *321,n,VoiceMailMain(@buzon)
exten => *321,n,Hangup()
```

Para finalizar en el archivo *sip.conf*, solo agregamos una línea como se ilustra a continuación para habilitar el *voicemail*.

```

[IMSI73210XXXXXXXXX62]
callerid=1212
canreinvite=no
type=friend
allow=gsm
allow=ulaw
allow=alaw
context=sip-external
host=dynamic
dtmfmode=RFC2833
mailbox=111@defaul

```

Luego se guardan los cambios realizados en los archivos y se debe reiniciar el servicio de Asterisk para garantizar que los cambios hechos en los archivos surtan efecto en el servicio, para ello se ejecuta el siguiente comando desde el CLI de Asterisk:

```
reload
```

Finalmente se tendrá un servicio de *voicemail*.

3.4. Servicio de transferencia de llamada atendida



Figura 3.2: Transferencia de llamada atendida

La transferencia de llamadas se utiliza generalmente en empresas y es la encargada de transferir la llamada que actualmente se esté atendiendo a cualquiera de las extensiones que desea la persona que llama.

Existen dos métodos de transferir las llamadas el primero es una llamada atendida y el segundo una llamada desatendida o ciega. Para ello existen las

opciones de *atxfer* y *blinxfer*:

En el archivo de *Features.conf* es donde se configura o asigna una función en una determinada combinación de teclas las cuáles serán las claves para que los teléfonos puedan hacer dicha transferencia.

Las líneas que se deben agregar al código en el contexto de [featuremap] y la clave que representa cada tipo de transferencia es:

```
Blindxfer => #1
Atxfer => *2
disconnected => *0
automon => *1
automixmon => *3
(Set(DYNAMIC\_FEATURES=automon))
```

Para modificar el *Dial()* accedemos al archivo *extensions.conf* y dentro del contexto donde se encuentre la configuración anteriormente realizada colocamos las letras *tTwWxX* para que se ejecute la transferencia.

```
exten => 1212,1,Macro(dialGSM,IMSI73210XXXXXXXXX62@127.0.0.1:5062,
10,tTwWxX)
```

3.5. IVR

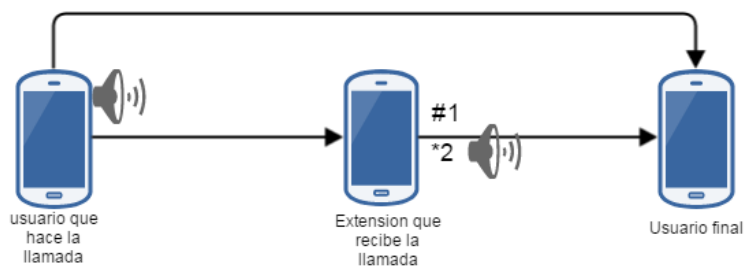


Figura 3.3: Transferencia entre tres teléfonos.

El servicio IVR representa un sistema centralizado y automatizado de respuesta interactiva de gestión de llamadas orientado a entregar o capturar información automatizada a través del teléfono el cual permite a los usuarios interactuar con un menú compuesto por mensajes de voz pregrabados. IVR es un sistema capaz de recibir una llamada e interactuar con el usuario a través grabaciones ya hechas y reconociendo respuestas del cliente según las teclas que se presionen [32].

Estos sistemas de respuesta de voz interactiva se utilizan en general en conjunto con los sistemas de distribución automáticas de llamadas como parte

de un esquema más amplio de flujo de llamadas para mejorar la experiencia en llamadas del cliente y brindar así un mejor servicio en cuanto a un gran número de llamadas entrantes.

Cuando el usuario realiza una llamada, el sistema de IVR contesta y presenta una serie de alternativas a través de mensajes grabados para que el usuario pueda escoger y seleccionar la opción que desea y lo hace presionando las teclas del teléfono según el menú que el sistema le ofrezca.

La configuración para este servicio se realiza reconfigurando el archivo *extensions.conf* y se inicia con la grabación de los archivos de audio, los que el usuario escuchará cuando realice la llamada y para obtener esta grabación se hace uso de la aplicación *record()* que permite crear un archivo de audio con alguna grabación que desee el usuario.

Ya ubicados en el archivo *extensions.conf* dentro de un contexto que puede ser creado por el programador, se crea el menú y se reproduce a través de un background que trae la grabación que se ha hecho para exponer tal menú y a través de la aplicación *WaitExten* asignamos el tiempo en segundos que se va a esperar para la respuesta del usuario.

```
[IVR-ejemplo]
Exten => s,1 Answer(): El orden de prioridad 1
Exten => s,n Wait(0.5)
Exten => s,n,Background(menu-redGSM); Es aquí donde se va a reproducir
el menú que se haya grabado, en este caso el mensaje que se ha grabado
es menu-redGSM.
Exten => s,n WaitExten(5); Es la aplicación que permite la cantidad de
tiempo para que el usuario envíe una respuesta a través del teclado.
```

Para poder reproducir los archivos anteriormente grabados se debe crear una extensión para cada uno de ellos.

```
Exten => 9992,1,Answer()
Exten => 9992,n, Playback(Recording /bienvenida-centralita.gsm)
Exten =$>$ 9992,n, Hangup()
```

Luego se crea una extensión que sirva de conexión al IVR que se creara mas adelante.

```
Exten =$>$ 9000,1,Goto(IVR centralita,s,1)
```

Finalmente se crea el IVR, para lo cual creamos un contexto llamado [IVR-centralita] y configuramos el menú.

```

Exten => s,1 Answer()
Exten => s,n Wait(0.5)
Exten => s,n,Background(menu-redGSM)
Exten => s,n WaitExten(5)
Exten => 1,1,Goto(sip-external,1111,1)
Exten => 2,1,Goto(sip-external,1212,1)
Exten => 4,1,Goto(aplicaciones,9992,1)
Exten => *,1,Goto(s,1)
Exten => t,Playback(goodbye)
Exten => t,Hangup()
Exten => i,Playback (pbx-invalid)

```

3.6. Conexión Internet

Para establecer una conexión a internet necesitamos configurar OpenBTS para manejar el tráfico GPRS. Para ello se realizan los siguientes pasos en su orden.

Configuración de OpenBTS: Para habilitar el servicio GPRS, ejecute el siguiente comando en la CLI de OpenBTS.

```
config GPRS.Enable 1
```

Luego se accede al archivo *iptables.rules* en aplicaciones y se adiciona la línea como se muestra a continuación, claro está que esto depende de la interfaz Ethernet o wlan con la que se cuente.

```
-A POSTROUTING -o eth0 -j MASQUERADE
```

A continuación se cargan las reglas desde el directorio de aplicaciones.

```
sudo iptables-restore <iptables.rules
```

Para cargar automáticamente la configuración con la configuración de red, se va al directorio */etc/network/interfaces* y se agrega la siguiente línea, donde */path/to* se refiere a la ubicación en la que tengas el archivo *iptables.rules*.

```
pre-iptables-restore </path/to/iptables.rules
```

Si al intentar acceder a internet se presenta un error con el DNS, se debe ingresar en el directorio */etc/resolv.conf* y se debe cambiar la dirección del *nameserver* a: 8.8.8.8 en lugares sin proxy, por lo contrario si te encuentras en un lugar con proxy a la dirección de este servidor.

Para comprobar que si estas recibiendo internet en tu móvil por la red se debe tener deshabilitado el WIFI del móvil

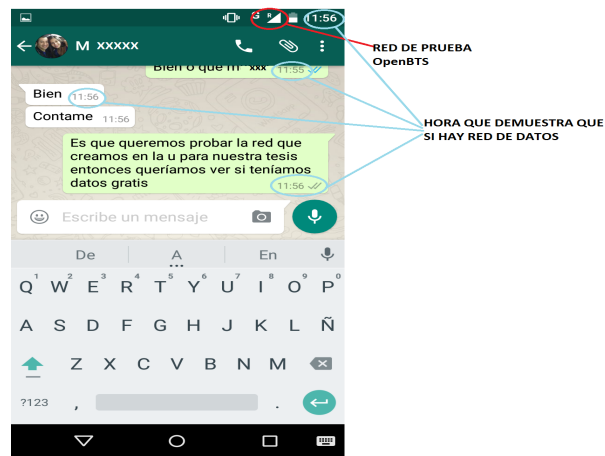


Figura 3.4: Conexion a Internet con Red de prueba

3.7. Establecimiento de llamadas de un terminal Ip a un terminal móvil GSM

Para emular una red y verificar el alcance cuando se tiene conexión a Internet se adicionan End-Points como son: teléfonos Ip, Softphones y Teléfonos móviles smartphones.

Su funcionamiento se describe a continuación:

1. Para realizar llamadas de un móvil GSM a un Softphones, o teléfonos Ip se configura las cuentas en los archivos sip.con y extensions.conf.

Los Softphones ³, ya instalados se configuran teniendo en cuenta que se debe crear un nuevo usuario y a cada uno de ellos darle un dominio, nombre, una contraseña y una identificación del usuario.

2. luego se realiza la configuración del teléfono IP *grandstream gxv3140* que es un teléfono de video de gran alcance IP multimedia, con la dirección del servidor el cual contiene a OpenBTS y Asterisk, se verifica la conexión haciendo *ping* en ambas direcciones.
3. Para comprobar que ha sido registrado se ejecuta el siguiente comando desde el CLI de Asterisk:

```
Sip show Peers
```

Si todo funcionó correctamente se procede a hacer llamadas y ejecutar los servicios anteriormente configurados desde terminales Ip a Móviles GSM.

³<http://www.zoiper.com/softphone/>, <http://www.3cx.com/VOIP/softphone.html>

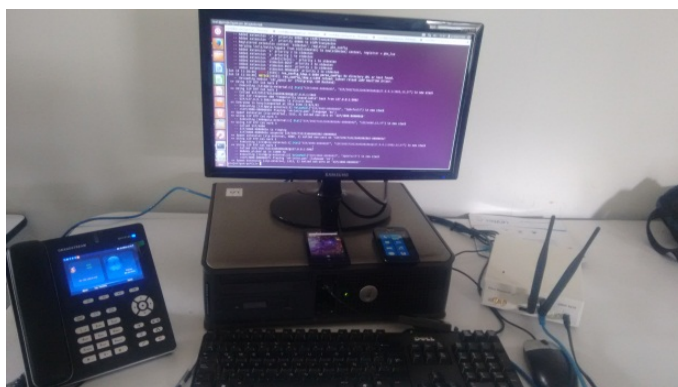


Figura 3.5: Conexión terminales GSM-IP

Cuando se pone en funcionamiento la red de prueba es fundamental el papel de la señalización, porque permite el intercambio de información entre los usuarios y la red, a fin de que la llamada pueda ser establecida y posteriormente, terminada.

Capítulo 4

Practicas de Laboratorio

Para evaluar las practicas propuestas, estas se realizan con dos grupos de alumnos de la asignatura “Laboratorio III de Sistemas de Telecomunicaciones” presentada en el programa de Ingeniería en Electrónica y Telecomunicaciones de la Universidad de Cauca, a los cuales se les realiza una encuesta al final de la sección. Las practicas y las encuestas están referenciadas en los anexos 1 y 2.

4.1. Descripción de las practicas propuestas

4.1.1. Practica 1. Central de telefonía móvil. Conceptos generales

Objetivo General de Formación

Adquirir un conocimiento teórico de las plataformas Asterisk, GNURadio, OpenBTS y el dispositivo USRP N210, elementos necesarios para implementar una red móvil GSM con OpenBTS.

Objetivos Específicos

1. Ampliar el campo de recursos con el que se cuenta al manipular redes móviles
2. Profundizar en los conceptos teóricos de las plataformas Asterisk, GNU-Radio y OpenBTS.
3. Profundizar en los conceptos teóricos del dispositivo USRP N210.

Descripción

En esta práctica se proporcionara la documentación y las referencias necesarias que permitan al alumno desarrollar bases de conocimiento para el análisis

sis, desarrollo e implementación de una red de telefonía móvil. La practica se encuentra en el Anexo 1 y se desarrollara en una sesión de 2 horas.

4.1.2. Practica 2. Creación de una red GSM basada en el proyecto OpenBTS

Objetivo General de Formación

Implementar una red de telefonía móvil GSM a través del dispositivo de hardware libre USRP N210 y herramientas de código abierto como OpenBTS y Asterisk, que en conjunto con un computador permite la implementación y el diseño de sistemas de radiocomunicaciones potentes.

Objetivos Específicos

1. Instalación y configuración de Plataformas GnuRadio, Asterisk y OpenBTS.
2. Diseñar y configurar planes de marcación con la plataforma Asterisk para una red de telefonía GSM basada en el proyecto OpenBTS.
3. Registro de terminales en la red GSM.

Descripción

En esta práctica se realiza la instalación y configuración de las herramientas necesarias para la implementación de una Celda GSM fundamentado en base al proyecto OpenBTS, donde el estudiante podrá interactuar con herramientas Open Source y software libre, idóneo para que experimente y desarrolle su capacidad investigativa en sistemas de comunicaciones móviles. La practica se encuentra en el Anexo 1 y se desarrollara en dos sesiones de 2 horas.

4.1.3. Practica 3. Configuración de servicios de valor agregado

Objetivo General de Formación

Configurar servicios de comunicación para la red de telefonía móvil empleando el programa de software de código abierto Asterisk.

Objetivos Específicos

1. Configuración de Llamadas y mensajes.
2. Configuración de idioma español.
3. Configuración de servicios básicos Asterisk (VoiceMail, Transferencia de llamada, IVR).

4. Conexión Internet
5. Establecimiento de llamadas de un terminal móvil GSM a cualquier terminal Ip conectado por internet.

Descripción

En esta práctica se realiza la configuración del servicio de llamadas y mensajes, de igual manera se implementan algunos servicios de valor agregado haciendo uso del software Asterisk y finalmente se realiza el establecimiento de llamadas desde nuestra red a un terminal Ip conectado mediante Internet. La practica se encuentra en el Anexo 1 y se desarrollara en cuatro sesiones de 2 horas.

4.2. Evaluación de las practicas propuestas

4.2.1. Análisis de la encuesta.

Como técnica de recolección de información se plantea la encuesta que será una herramienta fundamental para ver el estado y el comportamiento de los laboratorios, de tal forma que los estudiantes puedan participar y dar sus opiniones acerca del presente proyecto que está enfocado a los laboratorios de la FIET.

Esta encuesta escrita se aplicó a 18 estudiantes que están cursando el laboratorio III de sistemas de telecomunicaciones, quienes han aprendido conceptos básicos relacionados con este proyecto; además se realizaron 2 encuestas en dos horario diferentes lo que genera dos grupos distintos de estudiantes y diversas opiniones.

En la encuesta se consultó sobre propuestas para el mejoramiento del desempeño en redes para estos laboratorios, que trabajan sobre la plataforma asterisk de tal forma que combinándolo con otras tecnología o plataformas se pueda tener un laboratorio más interactivo con la redes de comunicación por radio en este caso redes GSM, que ayudará a los estudiantes a tener consultas sobre este tipo de redes y realizar trabajos futuros en ellos.

Se hace esta encuesta para analizar si el estudiante, que está cursando este tipo de laboratorio, está de acuerdo con lo planteado y si muestran interés por las investigaciones que se deben dar para este caso. Los autores esperan que el 100 % de los participantes de las practicas, que realizaron la encuesta, les haya parecido de gran importancia para su rendimiento académico y de miras al futuro, de tal forma que su calificación sea una de las dos mejores opciones expuestas en la encuesta.

A continuación se presenta un análisis de los datos obtenidos en la encuesta desarrollada en el marco de este proyecto, donde los estudiantes podían calificar las preguntas con 1, 2, 3 o 4, de acuerdo a su conformidad con el desempeño como estudiante. Estos valores se denotaron como 1: malo; 2: aceptable; 3: bueno; 4: muy bueno.

1. Primera pregunta: ¿En qué grado esta(s) práctica(s) de laboratorio cumple(n) con el proceso de planificación, implementación y evaluación?

La pregunta hecha se enfoca en el objetivo de como el estudiante ve este tipo de prácticas para los laboratorios, si se cree que puede mejorar o ayudar en los aspecto del trabajo para conseguir un propósito academico, poner en funcionamiento todas sus habilidades en los métodos o guías que se puedan presentar y de esta manera pueda verificar sus resultados de crecimiento como ingeniero en el conocimiento de estas plataformas.

La figura 4.1 resume las opiniones de los estudiantes respecto a la anterior pregunta, puede notarse que ninguno de los estudiantes que hicieron parte de esta práctica tomó las opciones 1 y 2, lo cual expresa que a los estudiantes de este tipo de laboratorios les parece que estas prácticas ayudan a mejorar en los aspectos de planificación dentro de los objetivos que se puedan realizar y también mejoran en las actividades de seguir métodos o guías que ayuden a dar más conocimiento.

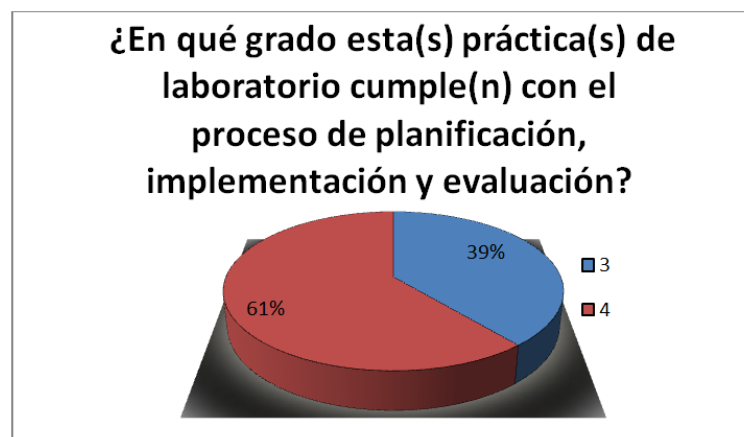


Figura 4.1: Resultados primera pregunta

Dentro de las personas encuestadas la figura 4.1 nos muestra que el 39% le dan un valor de bueno al objetivo de esta pregunta lo cual hace que el resultado esperado sea satisfactorio, y pensamos que este porcentaje se debe a que se debería haber dado una mejor información sobre el propósito de las practicas, por ejemplo incluyendo una introducción más

clara y amplia sobre las plataformas que se usaron. El 61 % le da una calificación de muy bueno y cumple con nuestras expectativas.

2. Segunda pregunta: ¿La práctica contribuyo a mejorar sus conocimientos en redes móviles?

Se considera que el conocimiento en redes móviles es de gran importancia en esta época, tanto en preparación universitaria como laboralmente para un profesional en ámbitos de las telecomunicaciones, por lo tanto esta pregunta tiene como objetivo analizar si este tipo de prácticas hacen que el estudiante mejore en sus capacidades de conocer e implementar el funcionamiento de redes móviles y si contribuye a un mejor profesionalismo en la materia.

La figura 4.2 nos muestra que ninguno de los estudiantes ve como malo (opción 1) el mejoramiento en los conocimientos acerca de redes móviles, por el contrario se muestra que la gran mayoría de los estudiantes encuestados están de acuerdo y le dan una calificación de muy bueno a que la práctica puede contribuir al conocimiento que ya tienen sobre redes móviles y de esta forma ayudar a mejorar las implementaciones en este tema.

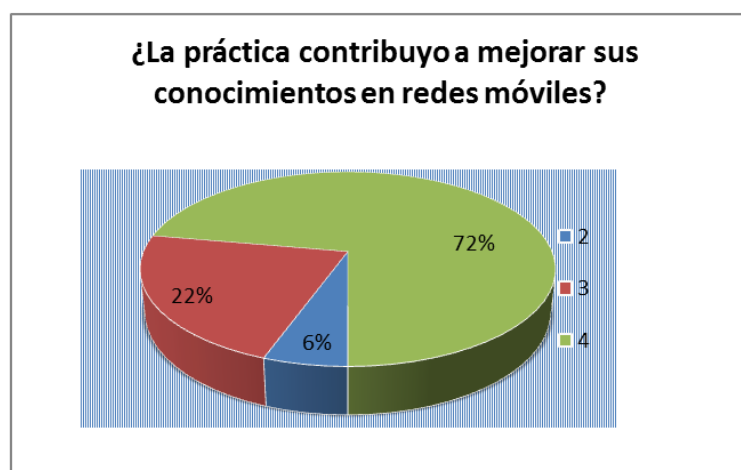


Figura 4.2: Resultados segunda pregunta

El 6 % de los participantes de la encuesta dan un valor aceptable al propósito de esta pregunta (figura 4.2), se cree que por falta de entendimiento o falta de interés de las personas con el tema o tal vez les falte un poco más de conocimiento en conceptos básicos sobre este tipo de redes; el 22 % participan dando un valor bueno al objetivo de proporcionar conocimiento y el 72 % están de acuerdo con lo planteado en la práctica y

dan una calificación de muy bueno a que fomenta e incrementa el conocimiento en la redes móviles, lo que muestra que es satisfactorio y que es muy pequeña la diferencia al resultado esperado.

3. Tercera pregunta: ¿La práctica motiva a tener conocimientos acerca de la plataforma OpenBTS?

Dentro de este proyecto la plataforma OpenBTS es de gran importancia ya que esta plataforma que ha sido descrita en capítulos anteriores es la tecnología que une a asterisk con el hardware USRP para que de esta forma se pueda realizar y manipular la red GSM que se ha creado. Por esto el objetivo de esta pregunta es saber si con este tipo de prácticas los estudiantes se interesan por la investigación y por adquirir un conocimiento sobre esta plataforma de tal manera que puedan construir cada uno de ellos su propia red y trabajar sobre posibles mejoras.

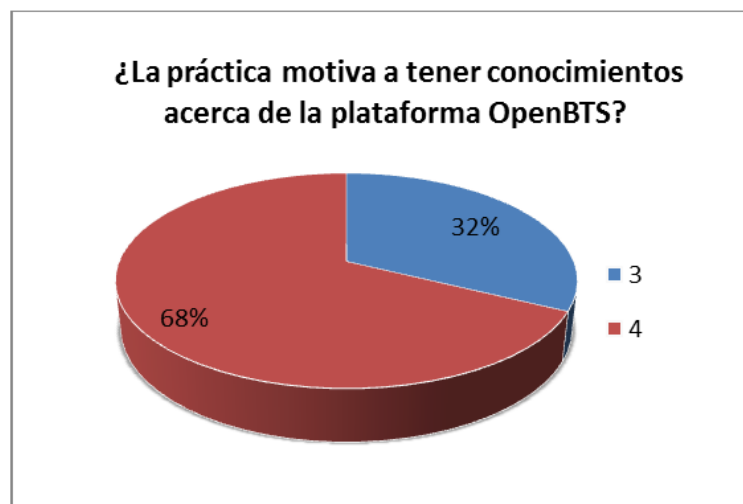


Figura 4.3: Resultados tercera pregunta

En la figura 4.3 se aprecia el resumen de las respuestas de las personas encuestadas y que al igual de la primera pregunta, ninguno de los estudiantes tomó las opciones 1 y 2, con lo que se puede concluir que a los estudiantes que hicieron parte de la práctica les parece motivante tener conocimientos sobre esta plataforma y de esta forma poder desarrollar y manipular todo lo referente a esta red que se ha elaborado en este proyecto.

El 68 % de los estudiantes encuestados dan una calificación de muy bueno frente al interés de obtener un conocimiento sobre esta plataforma, y el 32 % cree que es bueno (calificación de 3), lográndose el resultado que se

esperaba para el objetivo de esta pregunta. Adicionalmente se cree que dando una mejor introducción y motivación sobre esta plataforma a todas las personas les parecerá muy bueno fomentar este tipo de prácticas; pero como conclusión basándonos en estos resultados los estudiantes muestran gran interés y motivación por tener más conocimientos sobre el tema.

4. Cuarta pregunta: ¿La práctica contribuyo a mejorar sus conocimientos acerca de la configuración de servicios en Asterisk?

Desde hace varios años en los laboratorios ya nombrados anteriormente siempre se ha trabajado con asterisk, y de hecho todos los estudiantes que han pasado por estos laboratorios tienen un buen conocimiento acerca de esta plataforma pero hasta el momento no se ha trabajado en los laboratorios la combinación con otras plataformas o tecnologías que puedan ayudar a entender, practicar o simplemente aprender más conocimientos sobre asterisk. Es de aquí donde surge esta pregunta y que se plantea como objetivo si es interesante para los estudiantes conocer más sobre asterisk y si este tipo de prácticas ayudan a fomentar para adquirir este conocimiento.

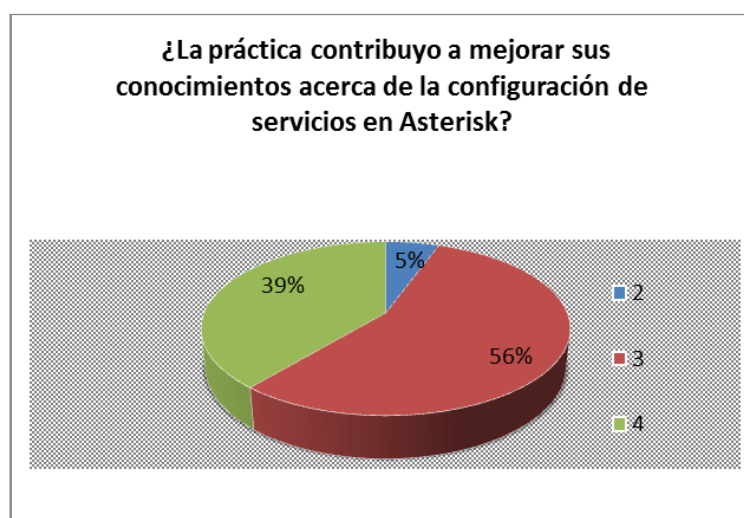


Figura 4.4: Resultados cuarta pregunta

En la figura 4.4 nos muestra que ninguna de las personas encuestadas selecciono malo la opción (1) frente a esta pregunta, con lo que se puede concluir que este tipo de prácticas incentivan que el estudiante se interese por tener más conocimientos acerca de las plataformas con las que practica, en este caso Asterisk y se investigue con que otro tipo de plataformas o tecnologías se puede combinar para tener los resultados que se plantee.

En la figura 4.4 se puede observar que el mayor número de personas le da un valor de bueno (3) a que esta práctica si contribuye a mejorar los conocimientos y configuración de esta plataforma equivalente, al 56 % de los participantes y el 39 % aprueba este objetivo con un valor de muy bueno (4). Al observar la figura, la diferencia con nuestro resultado deseado es de 5 %, lo cual nos da a entender que para llegar al resultado esperado se debe mejorar, tanto en los ponentes como en los estudiantes en ciertas características tales como: en una mejor atención y socialización hasta donde se puede llegar con Asterisk y que proyectos se pueden plantear y mostrar así las distintas formas de poder obtener mas conocimientos.

5. Quinta pregunta: ¿Cree usted que la manipulación y configuración de sistemas embebidos como la USRP N210 ayuda a su formación en servicios telemáticos?

Trabajar dentro de la universidad en la manipulación de cualquier sistema siempre ayuda a que el estudiante se interese más por conocer el funcionamiento, configuración y demás; por tanto, para conocer la determinación de cada uno de los estudiantes encuestados se crea esta pregunta con el objetivo de conocer si es productiva la manipulación del hardware utilizado en este proyecto para fomentar conocimiento en la formación de servicios telemáticos.

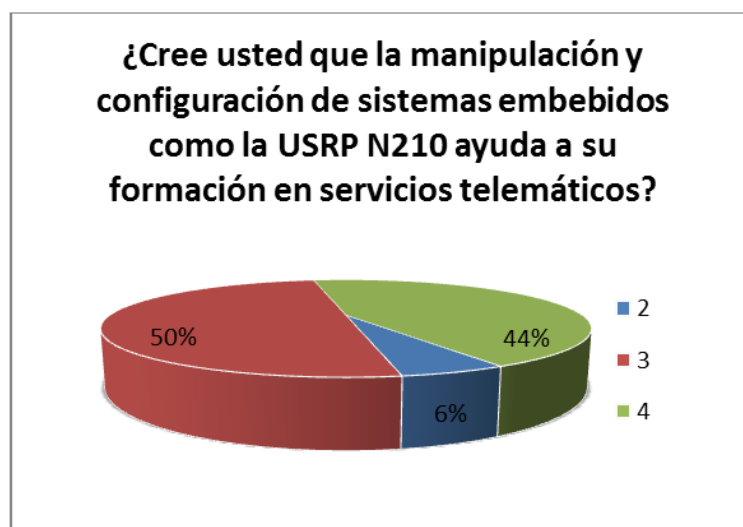


Figura 4.5: Resultados quinta pregunta

La figura 4.5 resume las respuestas participativas de los estudiantes encuestados, donde se observa que ninguno de los estudiantes califico el objetivo de esta pregunta como malo, por lo que se puede concluir que los estudiantes les parece que es de gran ayuda la manipulación de este

tipo de sistemas para obtener el conocimiento y la formación en los distintos servicios que se desean promover.

En la figura se observa que solo el 6 % de las personas que participaron en la encuesta dan un valor aceptable para la manipulación de los sistemas, pero la mitad de las personas encuestadas aseguran que esta actividad es buena para la formación en telemática y el 44 % lo ve como muy bueno en el mismo sentido de formación. Lo que muestra la diferencia con el resultado esperado y reitera que para poder llegar al 100 % se debe tener más manipulación de sistemas embebidos dentro de la universidad para que los estudiantes se familiaricen más y se tenga una buena formación y entendimiento en este tipo de dispositivos.

6. Sexta pregunta: ¿Las guías presentadas son coherentes con el objetivo planteado?

Las guías en un laboratorio son de gran importancia para poder entender y plantear bien el objetivo al que se quiere llegar, por lo tanto se presenta esta pregunta con el objetivo de conocer si las guías presentadas a los estudiantes son claras y coherentes con el fin que se busca en la creación de esta red, de acuerdo al objetivo principal de todas las practicas.

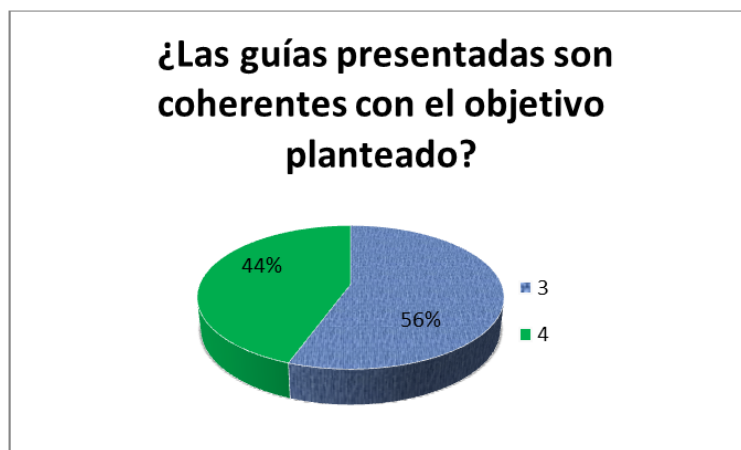


Figura 4.6: Resultados sexta pregunta

En la figura 4.6 se puede observar que los valores 1 y 2 que corresponden a malo y aceptable respectivamente no fueron calificados por los estudiantes lo que nos da a conocer que los documentos que se presentaron como guías para las prácticas de laboratorio cumplen con el objetivo que se pretendía y solo es necesario atender algunas observaciones que se recibieron.

Además la figura muestra que el 56 % del personal encuestado da un valor de bueno y aunque satisface que sea buena la calificación de los estudiantes, tal vez faltó explicar un poco más algunos detalles. El resto que es equivalente al 44 % denota estas guías de laboratorio con un valor de muy bueno y dan el valor esperado para que estas guías sean aceptadas dentro de los laboratorios.

7. Séptima pregunta: ¿Cree usted que las prácticas fomentan investigación e innovación en los estudiantes?

Para poder elaborar las practicas descritas anteriormente se tuvo que investigar y buscar la mejor forma de lograr los objetivos y una de las ideas más importantes en este proyecto es que el estudiante no se quede solo con las guías presentadas si no que se motive a realizar la investigación e innovación en trabajos futuros para poder expandir los conocimientos de las distintas plataformas. De esta manera nace el objetivo de esta pregunta para obtener los datos si los estudiantes creen que estas prácticas incentivan a tener investigación en este tipo de proyectos.



Figura 4.7: Resultados séptima pregunta

La figura 4.7 resume las respuestas de los estudiantes encuestados respecto a la séptima pregunta, puede notarse que ninguno calificó el objetivo de esta pregunta como malo o aceptable por el contrario, la gran mayoría de los estudiantes le da una calificación de muy bueno y están de acuerdo de que este tipo de prácticas ayuda a que los estudiantes se motiven por la investigación de nuevas tecnologías, manipulación y configuración de plataformas. De esta manera se concluye que el objetivo de esta pregunta es válido y aceptado por todas las personas que hicieron parte de las

prácticas.

Como se observa en la figura el 91 % de los estudiantes encuestados le da un valor de muy bueno a que este tipo de prácticas fomentan en los estudiantes investigación y motivan para tener innovación en este tipo de plataformas, el 9% restante dan la calificación de bueno, tal vez se pueda mejorar en algunos detalles pero se llega al resultado esperado en la participación de las personas encuestadas.

8. - Octava pregunta: ¿Cree que esta práctica cumple con sus expectativas?

Estas prácticas se hicieron con el objetivo de evaluar su diseño, con el fin de que el estudiante tenga la posibilidad de crear, modificar o manipular una red de telefonía móvil GSM y algunos servicios de valor agregado sobre ella, de esta forma se puede concluir si para las personas que hicieron parte de la práctica, esta experiencia cumple con los objetivos propuestos.



Figura 4.8: Resultados octava se pregunta

El resumen de las respuesta sobre esta pregunta se puede observar en la figura 4.8 y al igual que en gráficos expuestos anteriormente ninguno de los estudiantes califico con valores de 1 y 2 los que corresponden a malo y aceptable respectivamente; por el contrario se puede observar que el 64 % le da un valor de muy bueno a que las prácticas cumplen con las expectativas que se plantearon desde un inicio y solo el 33 % le dan un valor de bueno, quizás porque algunas de estas personas pueden dar más conocimiento y tengan las expectativas más altas, pero se concluye que este tipo de prácticas son de gran ayuda para incrementar los conocimientos, fomentar investigación y crear innovación en los estudiantes dentro de los laboratorios dictados en la FIET y cumple con los resultados esperados.

Las preguntas hechas en esta encuesta sirvieron para tener en cuenta en que puntos del diseño de las prácticas se puede mejorar y así mismo tener en cuenta las observaciones que se hicieron por algunos estudiantes y profesores. En las respuestas de las personas encuestadas ninguna de ellas calificó las preguntas con el valor de 1 que tiene como equivalencia un valor de malo con lo que se puede concluir que estas prácticas diseñadas para los laboratorio nombrados anteriormente son importante y de gran ayuda para aumentar conocimiento y fomentar investigación en los estudiantes en el tema de redes de comunicación y poder así tener un mejor profesionalismo dentro y fuera de la Universidad.

Trabajos futuros

1. Con el aporte de conocimientos necesarios se podría seguir este trabajo, dando más funcionalidades a este tipo de red en cuanto a servicios se trata. Además, con la combinación de asterisk, OpenBTS y USRP con otras tecnologías puede dar lugar a nuevos proyectos. La posibilidad de mejorar el alcance de la centralita modificando o cambiando componentes de la red como por ejemplo las antenas, las tarjetas secundarias o simplemente modificando la potencia con la cual se pueda transmitir este tipo de señales.
2. Utilizar un tipo de antenas con las que se pueda realizar un sistema de comunicaciones Wireless para poder trabajar con la tecnología MIMO, de tal forma que pueda dar conexiones de distintas redes, hechas con dispositivos periféricos y poder tener un mejor alcance.
3. La posibilidad de mejorar el alcance de la central con el uso de antenas mas potentes y acceder a otra USRP a fin de implementar sistemas MIMO.
4. Hacer un análisis al desempeño del funcionamiento de la red GSM creada.
5. Acoplar la red GSM creada a la red telefónica publica conmutada para lo cual es necesario un medio físico de conexión FXO que puede ser obtenido de una tarjeta Digium que brindan puertos físicos de estas características.
6. Desarrollar un script a fin de implementar de manera general la instalación de las herramientas necesarias para la implementación de la red móvil GSM basada en el proyecto OpenBTS.
7. En algunas partes del departamento del Cauca hay lugares donde la red de telefonía móvil de las empresas reconocidas en el país no tiene cobertura, por lo que se puede llevar este tipo de trabajos a utilizarlos como beneficio social dentro de estos territorios de tal forma que pueda ser un proyecto futuro ayudados por alguna agencia del gobierno.

Conclusiones

1. OpenBTS es la una herramienta que permite implementar una red de telefonía móvil a bajo costo, proporciona portabilidad, rápida restauración y reconfiguración de la red y permite interacción de tecnologías de comunicación como GSM e IP.
2. El proyecto OpenBTS proporciona un entorno idóneo para el desarrollo y construcción de una red de telefonía móvil GSM, a fin de permitir que estudiantes en un campo de laboratorio exploren el funcionamiento e interactúen con los componentes OpenBTS, Asterisk y USRP-N210 y establezca comunicación entre ellos.
3. La interacción con herramientas hardware Open Source como USRP-N210, y software libre y código abierto como (OpenBTS y Asterisk), permite en los estudiantes crear entornos de aprendizajes más dinámicos e interactivos para complementar el proceso de enseñanza y aprendizaje, facilita el trabajo en equipo y fomenta su capacidad investigativa para configurar servicios.
4. Mediante este proyecto se puede reforzar los conocimientos teóricos sobre redes GSM e IP y llevarlos a la práctica que ayudará a que el estudiante pueda comprender los parámetros y protocolos establecidos para este tipo de red.
5. Los servicios de valor agregado son de gran importancia para las centrales telefónicas y este proyecto ayuda a que la persona que trabaje sobre este tipo de redes, involucrando la plataforma Asterisk tenga más conocimiento sobre el tipo de configuración que se debe dar de acuerdo a la red con la que se trabaje.
6. La utilización de hardware Open Source configurable disminuye los costos en infraestructura, haciendo de OpenBTS una excelente opción para aumentar la competitividad y comodidad para variar parámetros a fin de generar sistemas mas potentes.
7. Se puede crear una red de telefonía móvil portátil a bajo costo con la utilización de hardware Open Source reconfigurable y software de código

abierto, a bajo costo y tomar las ventajas que ella ofrece como la configuración de aplicaciones, servicios y la conexión con dispositivos a grandes distancias fuera del alcance de nuestra red si se cuenta con Internet.

8. Con la creación de una red GSM basada en el proyecto OpenBTS permite poner en marcha una red con una celda en poco tiempo, ideal para implementar servicios de voz corporativos, permitiendo a pequeñas empresas competir en el ámbito de las telecomunicaciones.

Bibliografía

- [1] Gloria P. Avila Fajardo, Sandra C. Riascos Erazo. *Propuesta para la medición del impacto de las TIC en la enseñanza universitaria, volumen 14, pag 169-188, (2011). Universidad de la Sabana.*
- [2] Sigmund M. Redl, Matthis K. Weber, Malcolm W. Oliphant. *An Introduction to GSM, Artech House Publishers, (1995).*
- [3] Enric. Forner Clavijo, Carlos. Torrent Cuevas, Rubén. Martí Mateu, Francisco. Cordobés Gil, Pablo. Martínez Dimingo. *Tecnología GSM, pag 19-24, UNAD, (2010).*
- [4] GSM. *Sistema Global para las comunicaciones móviles*, Disponible en: https://es.m.wikipedia.org/wiki/Sistema_global_para_las_comunicaciones_m%C3%B3viles
- [5] Julián D. Vásquez Gutiérrez, Iván F. Santa Ramírez, José V. Restrepo Laverde. *Prototipo de una Estación Celular Portátil para Atención de Emergencias, (2012). Universidad Autónoma*, Disponible en: [http://kosmos.upb.edu.co/web/uploads/articulos/\(A\)_Prototipo_de_una_estacion_celular_portatil_para_atencion_de_emergencias_1fxQNS.pdf](http://kosmos.upb.edu.co/web/uploads/articulos/(A)_Prototipo_de_una_estacion_celular_portatil_para_atencion_de_emergencias_1fxQNS.pdf)
- [6] Alvaro Pachón. *Evolución de los sistemas móviles celulares GSM, Sistemas y Telemática, Volumen 2, (2006).*
- [7] HÜSEYİN ARSLAN. *Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems- 1ra Edición, pp. 43-105, (2007).*
- [8] Iván. Pinar Domínguez, Juan J. Murillo Fuentes. *Laboratorio de Comunicaciones Digitales Radio Definida por Software, Universidad de Sevilla - 1ra Edición, (2011).*
- [9] W. Tuttlebee. *Software Defined Radio: Origins, drivers and international perspectives, John Wiley and Sons, Inc. Nueva York. (2002).*
- [10] José M. Huidobro, David R. Martínez. *Integración de voz y datos: call centers: tecnología y aplicaciones, McGraw-Hill, Interamericana de España, (2003).* Disponible en: http://es.wikipedia.org/wiki/Voz_sobre_Protocolo_de_Internet

- [11] Jalife Salma. *VOZ sobre IP y temas afines, Presentation based on the curse “Voz sobre IP para América Latina”, (2004)*. Disonible en: <http://www.canieti.org/assets/files/58/salma%20jalife.pdf>
- [12] Eric Blossom. *GNU radio: tools for exploring the radio frequency spectrum, publisher for Belltown Media, pag 4, number 122, (2004), [en línea]*. Disonible en: http://es.wikipedia.org/wiki/GNU_Radio
- [13] Radio. GNU. *Welcome to GNU Radio, (2013), [en línea]*. Disonible en: <http://gnuradio.org/redmine/projects/gnuradio/wiki>
- [14] David A. Burgess, Harvind. Samra. *The Open BTS Project, an opensource GSM base station, (2008), [en línea]*, Disonible en: <http://openbts.org/>
- [15] David A. Burgess, Harvind. Samra. *OpenBTS Public Release, Accedido en Enero de 2014, [en línea]*, Disonible en: <http://wush.net/trac/rangepublic>
- [16] Axelle Apvrille, Fortinet. *OpenBTS for dummies, in GNU Radio, Accedido en Enero de 2014, [en línea]*, Disonible en: <http://gnuradio.org/redmine/attachments/420/fordummies.pdf>
- [17] Roig Oriol Sallent, Comes Ramon Agustí González & José Luis Valenzuela. *Principios de Comunicaciones Móviles, Universidad Politécnica de Catalunya, 1 Edición, (2003)*
- [18] Tong, Z. & Arifianto M.S.& Liau. *C. F. Wireless transmission using universal software radio peripheral; Space Science and Communication, International Conference on , vol., no., pp.19,23, 26-27 Oct. (2009)*
- [19] Germán Cuña, Rafael Durán & Mauricio Olivera *Redes cognitivas, estudio de la movilidad del espectro, Undergraduate capstone thesis from Instituto de Ingeniería Eléctrica, Facultad de Ingeniería, UdeLaR - (2014), [en línea]*, Disonible en: <http://iie.fing.edu.uy/publicaciones/2014/CDO14/>
- [20] George Fredric Eichinger & Kaushik Chowdhury & Miriam Leeser *Cognitive Radio Universal Software Hardware, Dynamic Spectrum Access Networks (DYSPAN), (2012) IEEE International Symposium on, pages 270-271, IEEE, [en línea]*, Disponible en: <http://www.coe.neu.edu/Research/rcl/theses/eichinger-msthesis2012.pdf>
- [21] Matt Ettus ,*Ettus Research, USRP N210, [en línea]*, Disponible en: <http://www.ettus.com/product/details/UN210-KIT>
- [22] Juan P. Montenegro Hidalgo. *Implementación de un sistema de comunicaciones basado en software radio, (2014), [en línea]*, Disponible en: https://repositorio.uam.es/bitstream/handle/10486/660544/montero_hidalgo_pablo_pfc.pdf?sequence=1

- [23] GPS disiplined oscillator, [en línea], Disponible en: http://en.wikipedia.org/wiki/GPS_disciplined_oscillator
- [24] Asterisk Arquitectura, [en línea], Disponible en: <http://www.wikiasterisk.com/index.php?title=Arquitectura>
- [25] Jim. Van Meggelen, Leif. Madsen, Jared. Smith. *Asterisk: The future of telephony, publisher for O´Reilly Media, pag 59-62 (2007)*
- [26] GNURadio. *Installing GNU Radio From Source*, [en línea], Disponible en: <http://gnuradio.org/redmine/projects/gnuradio/wiki/InstallingGRFromSource>
- [27] David A. Burgess, Harvind. Samra. *The Open BTS Project - an opensource GSM base station, (2008)*, [en línea], Disponible en: <https://wush.net/trac/rangepublic>
- [28] Setup USRP N210 *Conexion usrp*, [en línea], Disponible en: <https://sites.google.com/site/byungchulweb/setup-usrp2>
- [29] Juan P. Tene Castillo. *Sistema de conmutación y control de llamadas basado en OpenBTS y Asterisk, Universidad de Cuenca, (2014)*. [en línea], Disponible en: <http://dspace.ucuenca.edu.ec/handle/123456789/21413>
- [30] Comisión de Regulación de Comunicaciones. *Codigos MNC, Accedido en 2015*. [en línea], Disponible en: <http://www.pnn.gov.co/mapa/codigosMNC.xhtml;jsessionid=E89DE4225419AB45BDD3406B2>
- [31] Jim. Van Meggelen, Leif. Madsen, Jared. Smith. *Asterisk: The Future of Telephony, publisher for O´Reilly Media, pag 84-90, (2007)*.
- [32] Juan C. Valero. *Curso practico de Asterisk 1.4, pag 136-139, (2007)*. [en línea], Disponible en: <http://www.capatres.com/formacion-voip>