

Universidad Acción Pro Educación y Cultura



Decanato de Ingeniería e Informática

Escuela de Ingeniería

Tesis de Grado para Optar por el Título de:

Ingeniero Electrónico

Mención Comunicaciones

DISEÑO DE UNA CELDA CELULAR BASADA EN OPEN BASE TRANSCEIVER
STATION (OPENBTS) Y ASTERISK PARA PEQUEÑAS COMUNIDADES EN LA
REPUBLICA DOMINICANA

Sustentantes:

Br.Jorge Silva

2007-2045

Br.Aldwin Polanco

2007-0484

Asesor:

Prof. Porfirio Sánchez

Distrito Nacional

República Dominicana

Agosto, 2015

RESUMEN

En los países en vía de desarrollo es común que existan comunidades de bajos recursos, con poca densidad poblacional, la virtualización de las BTS hace que sea posible llevar comunicación móvil a estas comunidades por su bajo costo. Dichas BTS están conformadas con por un computador, softwares y un hardware para hacer la difusión de la red y reemplaza todos los equipos que conlleva implementar una BTS convencional.

Se instaló una máquina virtual con Ubuntu 12.04 donde posteriormente se instalaron y configuraron los softwares Asterisk y OpenBTS, se seleccionó el USRP1 de Ettus Research como hardware de difusión GSM y se configuró con OpenBTS para realizar las pruebas de funcionamiento y conectividad.

Aunque no se pudo implementar físicamente el sistema, se realizó un planteamiento de sistema, tomando en cuenta los modelos de propagación de Okmura-Hata, Walfisch-Ikegami y el modelo Friis, para tener el plan de cobertura así como también los cálculos necesarios para cumplir con la calidad de servicio y tener un presupuesto de potencia.

AGRADECIMIENTOS

En primer lugar quiero agradecer a Dios, por permitirme llegar a este punto de la carrera, iluminar mis caminos y hacer esto posible. A mis padres, Raul Silva y Digna Hazin, quienes siempre han confiado en mí dándome todo su apoyo, su confianza y animándome a siempre seguir hacia adelante. A mi hermana Lisbeth Silva por siempre estar dispuesta a ayudarme en todas mis necesidades. A mi hermano Ángel Silva por siempre mantenerme de buen humor con sus ocurrencias. A mi esposa Dorothy Toribio, quien siempre estuvo dispuesta a darlo todo para hacer posible este proyecto.

A todas mis amistades, familiares, profesores, compañeros de estudios, en especial a Luisanyel Perdomo, Raffy Marizan, German Vargas, Porfirio Sánchez, Luis Perez Méndez, quienes de una manera u otra potenciaron y aportaron su granito de arena para ayudarme a completar cada etapa de esta carrera.

Jorge Raul Silva Hazim

AGRADECIMIENTOS

A Dios en primer lugar, por permitirme llegar al final de esta etapa de mi vida luego de mucho esfuerzo, sacrificio y aprendizaje. A mi familia, por el sacrificio y la dedicación en inculcar valores y los conocimientos de toda una vida para lograr ser lo que soy hoy, los cuales son los principales motivadores a seguir a delante.

A los maestros, que en todo este tiempo dedicaron un momento en enseñarme todas las herramientas principales de mi carrera y las experiencias de vida.

A mis amigos, que siempre han estado conmigo en todo momento compartiendo las buenas y malas y apoyando todas las decisiones. A mis compañeros de clase, que a lo largo de todo este proceso hicieron más ameno y placentero ciertos momentos.

A personas que sin ellos no hubiese sido posible este logro: Ada Cabrera, Reynaldo Polanco, Reynaldo Polanco (hijo), Rita Polanco, Héctor Zapata, Keyla Encarnación.

Gracias a todos!

Aldwin Polanco

TABLA DE CONTENIDOS

RESUMEN.....	I
AGRADECIMIENTOS.....	II
TABLA DE CONTENIDO.....	IV
LISTA DE FIGURAS.....	VII
LISTA DE TABLAS.....	VIII
INTRODUCCION.....	IX

CAPITULO I. ESTRUCTURA DE UNA ESTACIÓN BASE TRANSMISORA VIRTUAL: CONCEPTOS GENERALES.....	10
--	----

1.1 Estructura de una red celular.....	10
1.2 Operación de la celda celular.....	11
1.2.1 Subsistema de red y conmutación.....	12
1.2.2 Centro de operaciones y mantenimiento.....	13
1.2.3 Registro de Localización.....	14
1.2.4 Registro de Posiciones de visitantes.....	14
1.2.5 Centro de Autenticación.....	14
1.2.6 La reutilización de frecuencias.....	17
1.3 Estructura de una celda basada en OpenBTS y PBX virtual.....	18
1.4 Asterisk.....	20
1.4.1 Origen.....	20
1.4.2 Arquitectura.....	22
1.4.3 Gestion de modulos.....	24
1.4.4 Recursos y aplicaciones.....	25
1.4.5 Funciones del Dialplan.....	26
1.4.6 Drivers de Canals.....	26
1.4.7 Traductores de Codec y Formatos.....	27
1.4.8 Interfaces.....	27
1.4.9 Archivos de Configuracion.....	28

1.4.10 Asterisk Manager Interface.....	29
1.4.11 Comand Line Interface.....	29
1.4.12 Asterisk Gateway Interface.....	30
1.4.13 Plan de Marcacion (Dialplan).....	30
1.4.14 Concepto de Marcacion.....	32
1.5 OpenBTS.....	32
1.5.1 Open BTS se distribuye en dos formas.....	33
1.6 Tipos de interface (Front End).....	35
1.6.1 Receptor superheterodino.....	36
1.6.2 Receptor de Frecuencia Intermedia cero.....	38
1.6.3 Receptor de Baja Frecuencia Intermedia.....	39
 CAPITULO II. VIRTUALIZACIÓN DE CENTRALES TELEFÓNICAS: HERRAMIENTAS Y TECNOLOGÍAS.....	 40
2.1 Softwares.....	40
2.1.1 Configuración de Asterisk.....	40
2.1.2 Configuración de OpenBTS.....	47
2.2 Hardware y Dongles.....	51
2.2.1 USRP (Periférico de radio de softwares universal).....	51
2.2.2 SDR (Radio definido por Software o Software Defined Radio)....	55
2.3 Integración de las partes.....	58
2.3.1 Interacción de OpenBTS y USRP1.....	58
2.3.2 Conexión y reconocimiento de USRP1.....	62
 CAPÍTULO III. PLANTEAMIENTO DE LA RED UTILIZANDO OPENBTS Y ASTERISK.....	 64
3.1 Criterios de Planificación de la red.....	64
3.1.1 Calidad de Servicio.....	64
3.1.2 Tasa de Llamadas Exitosas.....	65

3.1.3 El modelo Erlang B.....	68
3.2 Calculo del presupuesto de potencia.....	70
3.2.1 Cálculos para la Estación Móvil.....	71
3.2.2 Cálculos para la BTS.....	71
3.2.3 Antena.....	73
3.3 Calculo de cobertura.....	73
3.3.1 Modelo de Okmura-Hata.....	74
3.3.2 Modelo Walfisch-Ikegami.....	76
3.3.3 Modelo Friis.....	77
DIFICULTADES DE IMPLEMENTACIÓN.....	78
CONCLUSIÓN.....	79
RECOMENDACIONES.....	80
REFERENCIAS BIBLIOGRÁFICAS.....	81
ANEXOS.....	83

LISTA DE FIGURAS

Figura 1. Diagrama básico de un sistema GSM.....	11
Figura 2. Diagrama Reutilización de frecuencias.....	17
Figura 3. Arquitectura de una celda basada en Asterisk y OpenBTS.....	19
Figura 4. Arquitectura de Asterisk.....	23
Figura 5. Arquitectura de OpenBTS.....	35
Figura 6. Arquitectura de un receptor superheterodino.....	37
Figura 7. Arquitectura de un receptor de frecuencia intermedia cero.....	38
Figura 8. Arquitectura de un receptor de Baja Frecuencia Intermedia.....	39
Figura 9. Instalación de Asterisk en Ubuntu 12.04.05 LTS.....	45
Figura 10. Indicación de instalación de Asterisk lista para configurar.....	46
Figura 11. Indicación de que Asterisk está correctamente instalado.....	47
Figura 12. Diagrama de IMSI.....	49
Figura 13. Diagrama de bloques de USRP1.....	55
Figura 14. Arquitectura del SDR.....	57
Figura 15. Modificación del archivo ld.so.conf.....	61

LISTA DE TABLAS

Tabla 1. Listado de MCC y MNC de Republica Dominicana.....	50
Tabla 2. Parámetros de medida de Calidad de Servicio.....	65
Tabla 3. Descripción de los parámetros del presupuesto de potencia.....	70
Tabla 4. Parámetros máximos y mínimos para el modelo de Okmura-Hata.....	75
Tabla 5. Parámetros máximos y mínimos para el Modelo Walfisch-Ikegami.....	76

INTRODUCCIÓN

Las telecomunicaciones en la actualidad son, más que un lujo, una necesidad para el ser humano, debido a que la comunicación a distancia es indispensable y facilita el desarrollo de la humanidad. Sin embargo existen brechas digitales en muchas regiones de la Republica Dominicana, donde existen pequeñas comunidades con baja densidad poblacional.

Las empresas de telecomunicaciones invierten según la demanda que tengan y como las comunidades pequeñas no prometen una recuperación total de la inversión a corto plazo las mismas optan por no invertir para no comprometer su capital de inversión.

Para que se pueda invertir en este tipo de proyectos, el presupuesto de implementación debe ser a bajo costo para que su inversión sea recuperable. La virtualización de los sistemas de comunicación es una de las innovaciones que hace esto posible, reduciendo los costos y espacio, optimizando la utilización de los recursos.

CAPÍTULO I. ESTRUCTURA DE UNA ESTACIÓN BASE TRANSMISORA VIRTUAL:

CONCEPTOS GENERALES

1.1 Estructura de una red celular

La estructura de una red celular, también conocida como red móvil, está basada en una serie de regiones llamadas celdas, las cuales cuando están unidas forman una especie de panal de avispas, conformando un área de cobertura. Cada celda cuenta con una estación base que contiene su propio transmisor / receptor que puede operar a diferentes tipos de frecuencia. Las bandas de frecuencia que utilizan las celdas móviles son:

- GSM (850 / 900 / 1800 / 1900 MHz)
- UMTS (850 / 900 / 1800 / 1900 / 2100MHz)

Cada celda está conectada entre si vía radioenlace o por cable y estas a su vez están enlazadas al Sistema de Comunicación por la misma vía. Además, cada celda contiene su propia torre de antena, su propia energía eléctrica: cuenta con energía eléctrica comercial, inversor y generador eléctrico independiente, de esa manera se evita que haya una interrupción de energía eléctrica.

Su arquitectura está basada en la Figura 1, donde se muestra un flujograma básico desde que la llamada sale del equipo hasta que ingresa en el sistema de comunicaciones.

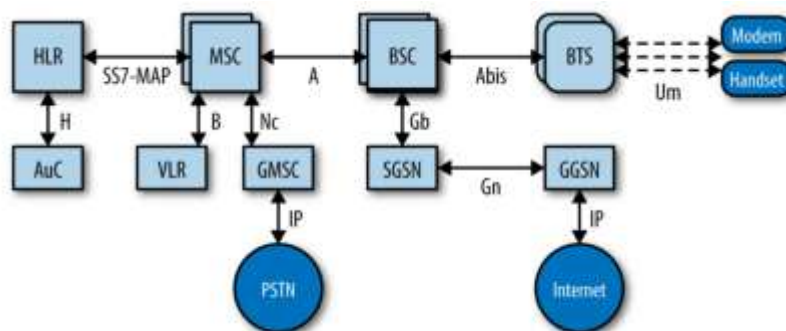


Figura 1. Diagrama basico de un sistema GSM. Michael Iedema. (2015).

Getting Started with OpenBTS. CA, Estados Unidos: O'Reilly Media, Inc.

Esta es la estructura que se utiliza hoy en día para crear un sistema de comunicación GSM de voz.

1.2 Operación de la celda celular

A continuación se van describir las funcionalidades de una celda celular, los componentes que esta posee y el modo de operación de cada uno. Comenzando desde un extremo, están las Estaciones Móviles, que son las terminales que tiene el usuario del servicio, en este caso, los celulares o teléfonos GSM, estos se comunican con la red vía radio UM, a esto también se le conoce como interfaz aérea.

Luego se encuentra el Subsistema de Estación Base (BSS o Base Station Subsystem), quien se encarga de gestionar los recursos para hacer posible la transmisión vía radiofrecuencia.

“Cada BSS dispone de un controlador de estación base (BSC, Base Station Controller) que se ocupa de un grupo formado por una o más estaciones transceptoras base (BTS, Base Transceiver Station). El controlador de esta estación base es responsable de gestionar localmente los recursos de radio de varias BTS, concentrando asimismo otras funciones comunes a múltiples BTS (ej. Codificación de las señales de voz).

El transceptor proporciona esencialmente, la conexión vía radio al terminal móvil. La interfaz entre una BTS y su correspondiente BSC se denomina interfaz Abis.” María Carmen España Boquera. (2003). Servicios avanzados de telecomunicación. Juan Bravo, España: Ediciones Díaz de Santos, S.A.

1.2.1 Subsistema de red y conmutación (Network Switching Subsystem o NSS)

Este subsistema incluye las principales funciones de conmutación GSM y se encarga de gestionar la comunicación entre los abonados de otras redes de telecomunicaciones. El NSS constituye el componente de la red fija de la red de servicios de radio móvil en un extremo y entre la red de servicio telefónico móvil de radio y otras redes públicas en la otra mano. Es la composición entre el HLR, el VLR y el MSC.

1.2.2 Centro de operaciones y mantenimiento (Operating and Maintenance Center o OMC)

El OMC está conectado al NSS y al BSC. Este se encarga de monitorear y controlar los elementos de la red, garantizando la calidad de servicio. Además, se mantiene verificando si existe una actualización y si algún equipo requiere mantenimiento o genera una alerta de falla preventiva.

El Centro de comunicación de Servicios Móviles (MSC, Mobile Switching Center), quien se comporta como el sistema neurálgico del sistema, este se encarga de gestionar la comunicación entre las BTS, está enlazado a cada subsistema mediante una conexión punto a punto. También, se encarga de la señalización y procesamiento de la llamada coordinando el traspaso entre células sin necesidad de cortar la llamada, lo que en términos de comunicación móvil se llama Handover.

La mayoría de los MSC están conectados con otras redes, incluso redes de telefonía fija, lo que permite que se hagan conexiones entre otras operadoras. A estos MSC se les conoce como Central de Conmutación Móvil de cabecera o Gateway (MSCC).

Registros de Localizaciones, los mismos se dividen en tres tipos: Registro de localización local (Home Location Register o HLR), Registro de Posiciones de Visitantes (Visitor Location Register o VLR) y Centro de Autenticación (Authentication Center o AuC).

1.2.3 Registro de Localización Local (HLR)

“Cuando se establece una relación contractual, los datos de usuario y las autorizaciones y claves asociadas se almacenan en una base de datos denominada Registro de Localización Local (HLR). Una referencia se almacena en el HLR indicando en qué parte de la red de radio móvil un usuario está operando de manera que una llamada entrante se transmitirá en consecuencia.” Bernhard H. Walke, P. Seidenberg, M. P. Althof. (2003). UMTS: The Fundamentals. Schlemmbach Verlag, Alemania: Jhon Wiley & Sons, Ltd.

1.2.4 Registro de Posiciones de Visitantes (VLR)

Esta base de datos es similar al HLR, la misma registra la posición del equipo terminal, inmediatamente el equipo terminal sale de la zona de cobertura de una BTS y entra a otra zona de cobertura el VLR registra automáticamente el equipo terminal para saber exactamente su ubicación.

1.2.5 Centro de Autenticación (AuC)

Esta base de datos contiene la información de autenticación, dicha información es utilizada para validar el equipo terminal y en temas relacionados con la seguridad, tales como fraudes, tarjetas SIM robadas, o equipos que tienen una mora en su factura. Aunque en ocasiones se suele representar el AuC como parte del HLR. Cabe notar que la interfaz que entre el HLR y el AuC es llamada “H-Interface”.

Existe otro registro llamado Registro de identidad del equipo (Equipment Identity Register o EIR), es el que guarda la información de los equipos móviles, tal como el IMEI (International Mobile Equipment Identity).

Cuando el dispositivo intenta realizar una llamada, el MSC consulta al EIR para validar este código y verificar que no tenga ningún tipo de restricción por parte de la operadora y que no haya sido reportado como perdido o robado.

Dentro de este registro también unas listas que son consultadas por el MSC al momento del móvil realizar una llamada. Estas listas son: Blanca, Negra y Gris.

- **Lista blanca:** Es aquella donde están registrados todos los IMEI de los equipos autorizados a operar con libre acceso.
- **Lista Negra:** Es aquella donde están registrados todos los IMEI que tienen restricciones o tienen el acceso prohibido. Estos pueden ser aquellos que han sido reportados como robados o perdidos, que son utilizados de forma ilícita o que pueden provocar alguna falla técnica si acceden a la red.
- **Lista Gris:** En esta se encuentran los equipos que están en bajo observación porque se ha detectado alguna falla en el mismo.

Otro elemento esencial en los sistemas GSM es el SIM (subscriber identity module, en español módulo de identificación de abonado), este es el modulo que contiene la información necesaria para que se pueda realizar una identificación y autenticación del usuario al que pertenece el equipo terminal y contiene algunas otras informaciones que requiere el sistema al momento de hacer la gestión del servicio.

Este módulo debe contener las siguientes informaciones:

- Número de Serie del modulo
- Estado (si está bloqueado o no)
- Algoritmo de autenticación
- Identificación internacional de usuario móvil (MSI)
- Identificación temporal de usuario móvil (TMSI)
- Algoritmo de generación de claves de cifrado
- Clave del algoritmo de cifrado de señalización y datos
- Numero de secuencia de la clave del algoritmo de cifrado
- Clase de control de acceso del usuario

1.2.6 La reutilización de frecuencias

Para poder aumentar la capacidad de cobertura y evitar interferencia entre las celdas, se utiliza el método de reutilización de frecuencias. En este método, a cada celda se le asigna una frecuencia diferente a las celdas adyacentes y así se evita la interferencia entre la difusión entre celdas, tal como lo muestra la figura 2.

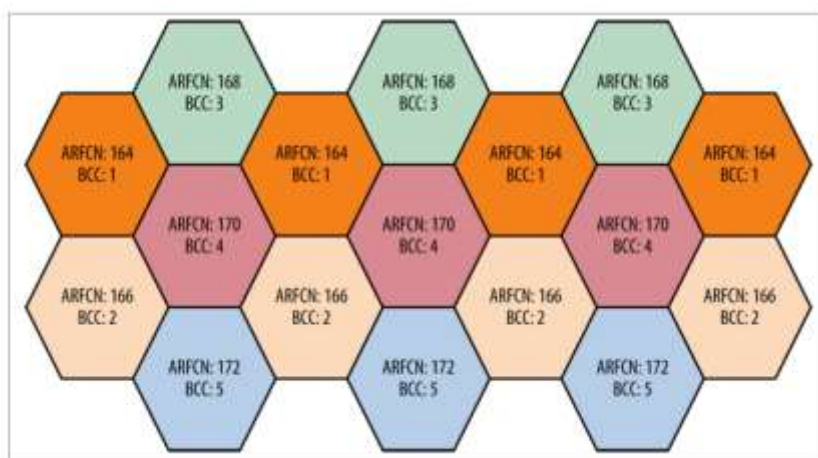


Figura 2. Diagrama Reutilización de frecuencias. Cada color representa una celda con una frecuencia específica diferente a las celdas adyacentes. Michael Iedema. (2015). Getting Started with OpenBTS. CA, Estados Unidos: O'Reilly Media, Inc.

1.3 Estructura de una celda basada en OpenBTS y PBX virtuales

En una celda basada en OpenBTS y una PBX virtual, se reemplazan los equipos físicos por un ordenador que contiene los softwares de que virtualizan el entorno de una celda y lo único que necesita es hardware que sirve como medio de transmisión.

En un sistema de telefonía virtual, todo el procesamiento, señalización, registros e identificación son realizados por el procesador del computador y gestionados por los softwares quienes generas instrucciones paso a paso para simular exactamente cada etapa de una celda celular

La estructura de una celda que se basa en OpenBTS y PBX virtual es como se muestra en la figura 3.

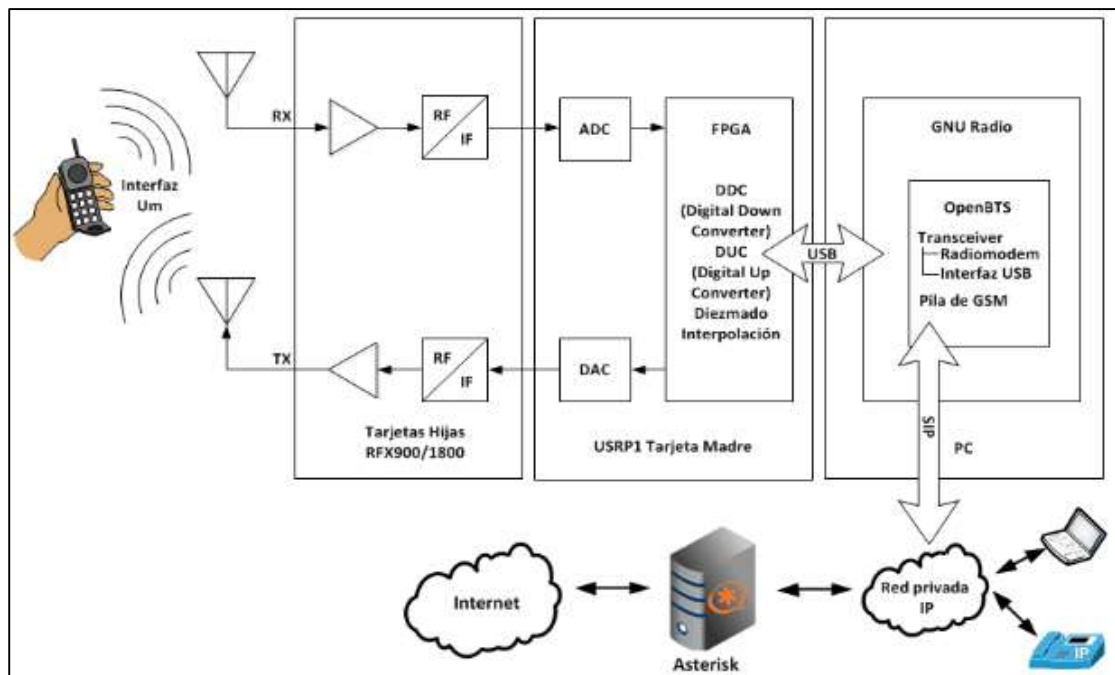


Figura 3. Arquitectura de una celda basada en asterisk y openBTS. Julián D Vásquez, Iván F Santa, José V Restrepo. (2012). Prototipo de una Estación Celular Portátil para Atención de Emergencias. 29/5/2015, de Universidad Pontificia Bolivariana.

1.4 Asterisk

1.4.1 Origen

El desarrollo de Asterisk fue iniciado en el año 1999 por Mark Spencer. En aquel momento ya existían otros proyectos de telefonía basados en la ideología Open Source. Pero esos proyectos estaban realmente enfocados a la telefonía a nivel software, basados en la búsqueda de establecer comunicaciones entre ordenadores a través de los sistemas de audio.

La diferencia de Asterisk con estos proyectos, era la idea preconcebida, de entrelazar por primera vez la red de telefonía común (public switched telephone network) (PSTN) o red telefónica pública conmutada con un ordenador. Para ello se formó una alianza con un proyecto de telefonía llamado Zapata iniciado por Jim Dixon.

La idea del proyecto Zapata, era la posibilidad de diseñar tarjetas específicas para convertir la señal analógica que provenía de la PSTN a una señal digital, y ahorrar costos en la construcción de dispositivos de telefonía y audio avanzados gracias a la tremenda reducción de costes que sufrían año tras año los procesadores. A través de estos, cada vez más potentes, y baratos, se podrían procesar una o varias señales de audio digital (DSP) sin gran dificultad y poder paralelamente construir teléfonos con ese coste reducido.

Hoy en día el proyecto Zapata, fue integrado totalmente en Asterisk, y Asterisk es patrocinado por una empresa que construye dispositivos de telefonía digital, llamada Digium, y se ha acomodado como una parte más de Asterisk con un nuevo nombre: DAHDI (son las siglas de Digium Asterisk Hardware Device Interface).

Después de 5 años de desarrollo sale al público la primera versión de Asterisk esta se llamó Asterisk 1.0, desde esta se ha venido desarrollando el sistema a ser una herramienta muy estable y confiable en la actualidad.

Asterisk es un proyecto de código abierto que agrupa la funcionalidad de una central telefónica privada PBX (Private Branch Exchange) dentro de un paquete de software, permitiendo realizar el registro, conexión y comunicación de terminales telefónicos conectados al servidor Asterisk.

A pesar de que se trata de una aplicación software ofrece las mismas características y servicios que un equipo de hardware PBX. Soporta una variedad de tecnologías para ejecutar y recibir llamadas telefónicas, muchos protocolos VoIP, así como también conectividad analógica y digital a las redes de telefonía tradicional, o a una red digital de servicios integrada (Vázquez & Restrepo, 2012).

Asterisk se encuentra formado por un conjunto de módulos que brindan una gran capacidad de abstracción de los protocolos, códecs e interfaces utilizados en cada conexión, esto es lo

que hace de Asterisk una herramienta totalmente flexible y adaptable. Existe una variedad de módulos en función de la aplicación que se vaya a desarrollar.

1.4.2 Arquitectura

Asterisk es un sistema modular que fue diseñado para que cada usuario pueda seleccionar las partes que va a utilizar y desactivar las que no desea utilizar, lo que lo hace un sistema escalable y extensible ya que se pueden añadir módulos para cada necesidad.

Asterisk es capaz de en cada módulo que usted carga, de proporciona capacidades en el sistema. Por ejemplo, un módulo podría permitir a su sistema Asterisk comunicarse con líneas telefónicas análogas, mientras que otro puede agregar reportes de llamadas.

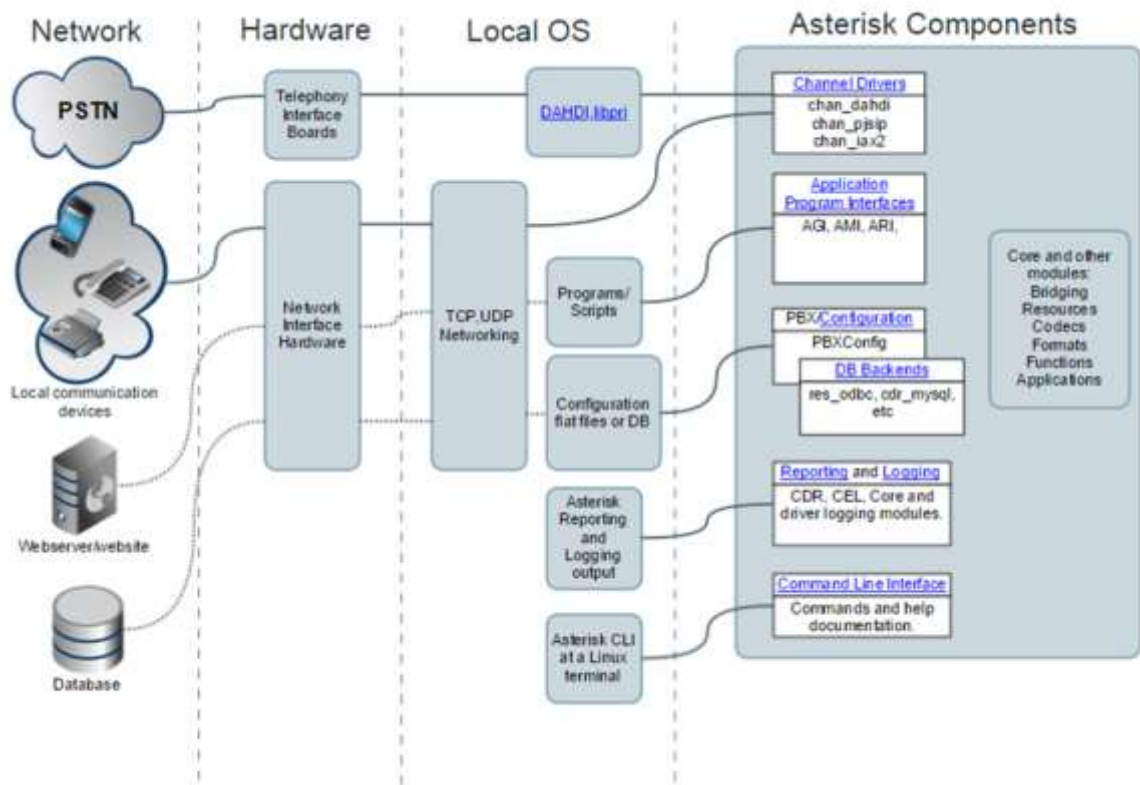


Figura 4. Arquitectura de Asterisk. Newton R. (Agosto 06, 2014). Asterisk Architecture. Junio 03, 2015, de Asterisk Sitio web: <https://wiki.asterisk.org/wiki/x/R4JJ>

Asterisk es una estructura totalmente independiente de los dispositivos, troncales, y demás externos al mismo. Gran parte de las PBX tradicionales, comúnmente podemos observar que existen estructuras lógicas para diferenciar entre los teléfonos y otros dispositivos de telefonía como faxes, y por otro lado líneas entrantes como troncales primarios, líneas RDSI, etc.

En Asterisk esto no existe, todo entra al sistema como un canal genérico y luego es gestionado de forma integral. Aunque sean diferenciados, todos se gestionan de la misma forma por eso, incluso recursos externos pueden llegar a ser manejados dentro de las posibilidades, de la misma forma, que una extensión SIP interna.

1.4.3 Gestión de Módulos

Como hablábamos anteriormente, Asterisk está basado en módulos independientes, los cuales pueden ser cargados y descargados a voluntad, en función de las necesidades que deseemos proveer al sistema.

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.).

Dentro del sistema de módulos, vemos a continuación los más populares:

- Aplicaciones
- Recursos
- Funciones del Dialplan
- Registro del detalle llamadas
- Registro de eventos de canales

- Drivers de canales
- Traductores de Codecs y Formatos
- Add-ons

1.4.3 Recursos y Aplicaciones

Las aplicaciones son aquellas acciones aplicables al manejo de las llamadas dentro del Plan de Marcación. 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. Existen otras aplicaciones comunes, como VoiceMail (encargada de la gestión del Buzón de Voz), Record (para grabar el sonido de un determinado canal), etc.

La función específica de los recursos es la de integrar Asterisk con los sistemas externos. Hablamos de bases de datos, servidores web, calendarios, etc. Tienen la capacidad de utilizar por sí mismos, Aplicaciones del sistema como veíamos antes. Pero una de las diferencias con respecto a estas, es que se cargan de manera estática, y pueden operar simultáneamente en múltiples canales, en vez de crearse dinámicamente para cada canal en curso.

1.4.5 Funciones del Dialplan

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 y son capaces de ofrecer mejoras para determinados aspectos del sistema que de por si pudieran ser limitados.

1.4.6 Drivers de canales

Son los drivers 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.

Todos los canales más típicos, tienen su correspondiente modulo para el driver. Ejemplos típicos son el específico para DAHDI, SIP e IAX, que justamente son los más utilizados por la comunidad Asterisk. El resto no está lo suficientemente difundidos, aunque tienen soporte y van mejorando con el tiempo, como el específico para tratar con estaciones Cisco, llamado como su protocolo propietario, Skinny (SCCP).

1.4.7 Traductores de Codecs y Formatos

Son la representación para los sistemas de audio y vídeo digitales de transmisión (codecs) y almacenamiento (formatos). De alguna forma, 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. 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.

1.4.8 Interfaces

Asterisk se despliega y comunica con el entorno a través de múltiples interfaces que provee, tanto para trabajos de gestión y mantenimiento manuales, como para su autogestión de forma estática, y sistemas externos que puedan recabar datos de forma dinámica y automática, y finalmente pasando a ofrecer interfaces para poder trabajar con lenguajes de programación que provean a su vez al sistema de nuevas funcionalidades extra que no estén construidas en su interior.

Se describen en cuatro estructuras básicas:

- Archivos de Configuración
- Interfaz de Comandos (CLI)
- Interfaz Pasarela (AGI)
- Interfaz de Gestión (AMI)

1.4.9 Archivos de Configuración

Todos se encuentran en la ruta de instalación por defecto de Asterisk. Suele existir un archivo de configuración que “soporte” la funcionalidad específica que puede aportar al sistema un módulo específico, como podría ser el caso de `chan_dahdi.conf`, que como veríamos antes, sería el archivo encargado de ofrecer toda la configuración específica para el módulo encargado de gestionar el canal dedicado a DAHDI.

Una parte de los archivos de configuración es posible que sean editados de forma dinámica a través de una Base de Datos. A esta funcionalidad se le ha llamado Asterisk Realtime, y por ejemplo es muy popular para la configuración del archivo específico para el módulo dedicado al canal SIP (`sip.conf`). Pero la mayor parte del resto de los archivos, son editados de forma manual.

1.4.10 Asterisk Manager Interface

En la mayor parte de las PBX más populares del mundo, existe un componente para poder ofrecer nuevas funcionalidades al sistema original a través de sistemas de procesamiento autónomos y ordenadores. Este recurso suele llamarse CTI (Computer Telephony Integration), y por regla general suele ser bastante limitado, ya que su potencial suele radicar en la cantidad de información que seamos capaces de extraer del mismo.

En Asterisk este CTI es llamado AMI, y su potencial es prácticamente ilimitado ya que podemos recoger prácticamente cualquier elemento interno en función de la configuración que apliquemos al mismo. Principalmente recibe todos aquellos comandos que sean capaces los módulos del sistema de aportar a la interfaz. Sistemas como marcadores predictivos, gestores de llamadas, sistemas de facturación, etc., se fundamentan en esta interfaz.

1.4.11 Comand Line Interface (CLI)

En términos generales, podría considerarse la consola de administración del sistema. Es muy importante y fundamental saber utilizarlo para la correcta gestión y mantenimiento del sistema de forma eficaz.

1.4.12 Asterisk Gateway Interface

Muy parecido al uso de Aplicaciones, disponemos de esta pasarela para poder ejecutar servicios en los canales, pero basados en otros lenguajes de programación diferentes a la estructura de Asterisk en sí.

Existen librerías AGI prácticamente para todos los lenguajes más populares como pueden ser, PHP, Python, Java, Ruby on Rails, etc.

Con esto tenemos un potencial ilimitado para crear o entrelazar aplicaciones ya existentes con nuestro sistema Asterisk, y obtener aún más funcionalidad de la que el sistema provee de por sí.

1.4.12 Plan de Marcación (Dialplan)

Es una parte fundamental de la arquitectura de Asterisk la podemos considerar como la espina vertebral de Asterisk. Si pudiéramos considerar el flujo de cualquier interacción de Asterisk y a su vez este flujo como un lenguaje de programación en si (o un script), de como por ejemplo las llamadas son gestionadas, este concepto se llamaría en Asterisk Plan de Marcación (o más popularmente conocido como Dialplan).

Existen tres formas de diseñar el dialplan:

- A través de un archivo de configuración simple
- Utilizando AEL (Asterisk Extension Logic)
- Mediante el lenguaje de programación LUA

El Plan de Marcación, llamando Dialplan en Asterisk, podría considerarse la columna vertebral del sistema. Como característica principal, podríamos decir que es algo parecido a un lenguaje de script, en el que funciones, aplicaciones y recursos se van intercalando para formar algo parecido a "procedimientos" y "programas" dentro de lo que pudiera considerarse.

Este paradigma de programación orientado a los canales de telefonía es exclusivo en el mundo y aporta todo el potencial y flexibilidad, que las PBX tradicionales no pueden ofrecer, o lo hacen mediante mecanismos de licenciamiento y estructuras en ocasiones demasiado complejas. Algo tan sencillo como crear un calendario de disponibilidad, suele ser una función que las centrales más nuevas y sofisticadas del mercado solo consiguen aplicando estrategias que en la mayor parte de los casos quedan limitadas a una serie de plantillas que no cubren la totalidad de la posible intención que requiera el cliente. Por ejemplo cuando hablamos de horarios por departamentos, se considera un proceso clásico en Asterisk, que en cambio, en algunas PBX, puede convertirse en un mundo.

1.4.13 Concepto de Marcación

Cuando hablamos de Plan de Marcación, hacemos referencia literalmente a esto: La Marcación en un teléfono, o dispositivo de cualquier tipo (no tiene que ser necesariamente los clásicos 10 números (del 0 al 9), sino también empezamos a incluir el concepto de carácter alfanumérico). Una vez realizado el marcado, este, entraría como una "entrada" dentro de nuestro plan de marcación por el contexto que corresponda, en función de cómo hayamos definido el canal al que hace referencia nuestro dispositivo SIP, IAX, etc. Dentro de este contexto, la entrada (la marcación que hicimos), empieza a recorrer el script secuencialmente y a partir de ahí pueden surgir una serie de eventos asociados a esa entrada que en cualquier caso siempre suelen tener algo que ver con el mundo de la comunicación.

1.5 OpenBTS

OpenBTS fue comenzado por Harvind Samra¹ y David A. Burgess. Un objetivo del proyecto es reducir el costo de la prestación de servicio GSM en áreas de zonas rurales y del mundo en desarrollo por menos de un dólar por mes por suscriptor.

“OpenBTS es una implementación de UNIX que utiliza un software de radio para presentar una interfaz GSM para un teléfono estándar 2G GSM y usa un softswitch SIP o PBX para

conectar llamadas. (Usted podría incluso decir que OpenBTS es una forma simplificada de IMS que funciona con teléfonos 2G características del teléfono.)

La combinación de la interfaz aérea GSM-mundial estándar con backhaul VoIP de bajo costo es la base de un nuevo tipo de red celular se pueden implementar y operado a un costo sustancialmente menor que las tecnologías existentes en muchas aplicaciones, incluyendo despliegues celulares rurales y redes celulares privados en zonas remotas.

1.5.1 OpenBTS se distribuye en dos formas:

1. La ("P") lanzamiento público. El lanzamiento público se distribuye bajo la licencia GPLv3 con derechos de autor asignados a Cuando la Free Software Foundation. El lanzamiento público es un subconjunto de la distribución Destinado a proyectos de prueba de concepto experimentación negocios, la educación y la evaluación.

2. El ("C") la liberación comercial. El lanzamiento comercial se instala en productos gama Networks bajo una mezcla de GPL y las licencias no-GPL. Rango Redes Bélgica Bosnia aussi un portal de clientes para los clientes de negocios donde el código fuente está disponible para los componentes de la instalación GPL OpenBTS.

La versión "C" está dirigido a usuarios que:

- Quienes necesitan proveer servicio celular en industrias, aplicaciones gubernamentales o comerciales.
- Cuyas políticas o modelos de negocio de la propiedad intelectual son compatibles con una / GPLv3
- Que requieren soporte comercial, monitoreo de red u otros servicios profesionales.” Michael Iedema, 2011

Los desarrolladores originales trabajando en el proyecto tienen acceso a licencias de espectro de pruebas, pero previamente han hecho frente a disputas legales (ahora resueltas) sobre el trabajo relacionado anterior significando que algo del código GSM de bajo nivel se está reescribiendo.

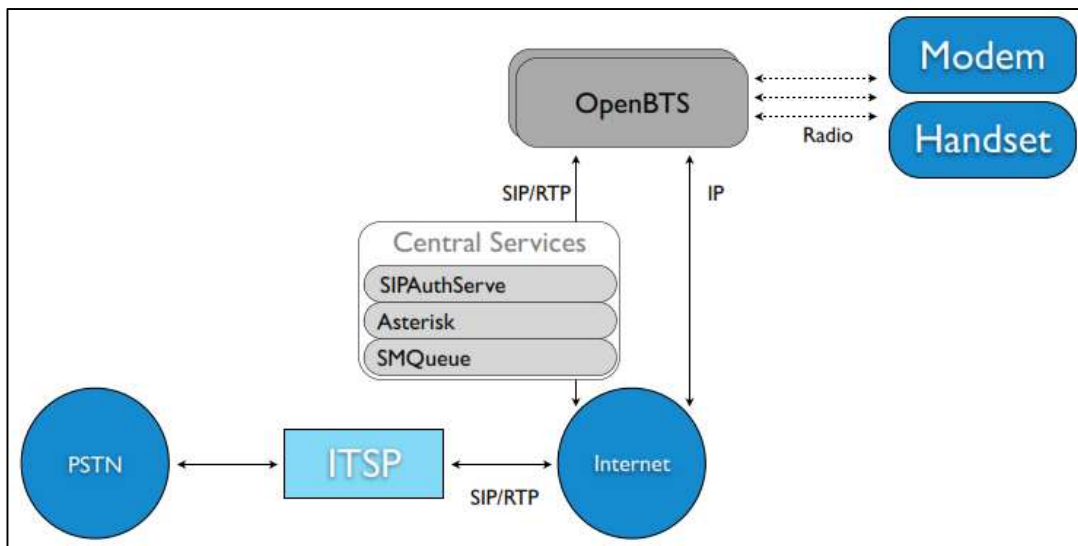


Figura 5. Arquitectura de OpenBTS. Michael Iedema. (2011). Welcome to OpenBTS. 2015, de Range Networks Sitio web: <https://wush.net/trac/rangepublic>

1.6 Tipos de Interfaz (front end)

El Front End es la interfaz que tendrá el sistema para comunicar el sistema virtual con los equipos terminales, básicamente es donde se conectará la antena que difundirá el servicio.

El mismo es parte de un sistema general de radio receptor-transmisor o transceptor.

En general, es definido como todo entre la antena y el sistema. Para un receptor, este "entre" incluye todos los filtros, amplificadores de ruidos bajos (Low-Noise Amplifiers o LNA) y mezclador descendente, necesarios para procesar las señales moduladas recibidas en la antena en señales adecuadas para la entrada en el conversor análogo - digital (Analog-

to.Digital Converter o ADC). Por esta razón, el extremo frontal RF se llama a menudo el análogo - digital o una porción de RF a banda base de un receptor.

Las arquitecturas de Front end más comunes son:

- Receptor superheterodino
- Receptor de Frecuencia Intermedia cero
- Receptor de Baja Frecuencia Intermedia
- Receptor Muestreo Bandpass

1.6.1 Receptor superheterodino

Esta es la arquitectura más común que puede encontrarse en los receptores o sistemas de Radio Frecuencia. La misma está basada en dos etapas de conversión: la señal RF recibida se desmodula primero a una frecuencia intermedia (IF) y luego se convierte a la señal de banda base. Su arquitectura se muestra en la figura 6.

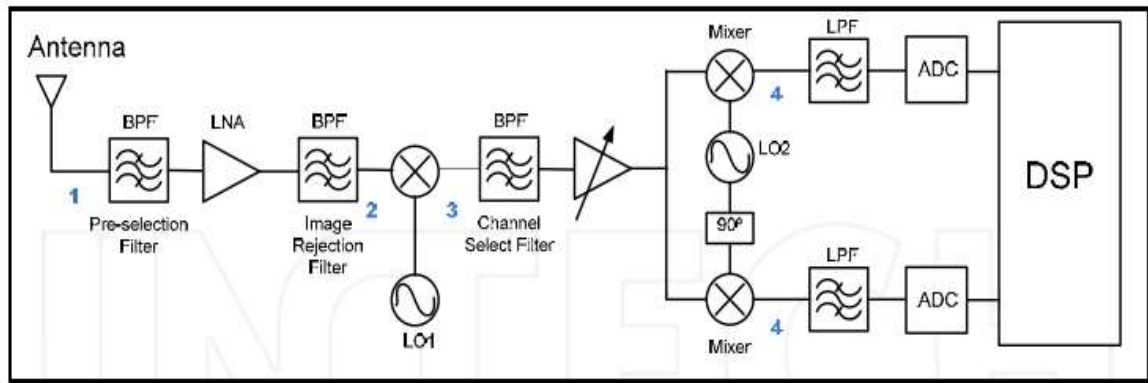


Figura 6. Arquitectura de un receptor superheterodino. Otto Limann. (1989).

Fundamentos de radio. España: MARCOMBO, S.A

En esta arquitectura la señal recibida primero se filtra por un filtro preseleccionado y luego de que es amplificada por amplificador de bajo ruido (LNA) pasa por un segundo filtro llamado filtro de rechazo de imagen, el cual reduce los efectos de frecuencia imagen antes de la primera conversión de RF a IF.

La señal se filtra de nuevo y es desmodulada a la banda base, donde es convertida al dominio digital para que se pueda procesar. Como antes se menciona, esta arquitectura es la más adoptada en los receptores de radio debido a su disponibilidad de bajo costo de banda estrecha de componentes de RF e IF con bajo consumo de energía.

Conjuntamente, esta arquitectura puede garantizar altos niveles de sensibilidad, es decir que permite una potencia de señal baja en la entrada para el que no es suficiente relación señal-ruido en la salida del receptor.

1.6.2 Receptor de Frecuencia Intermedia cero

Otra de las arquitecturas más típicas es el Receptor Zero-IF, que también se conoce como receptor homodino. Esta es una arquitectura simplificada del Receptor superheterodino, sin embargo en vez de tener dos etapas de conversión, esta convierte la señal RF directamente a banda base. En la figura 7, se muestra la arquitectura del receptor Zero-IF.

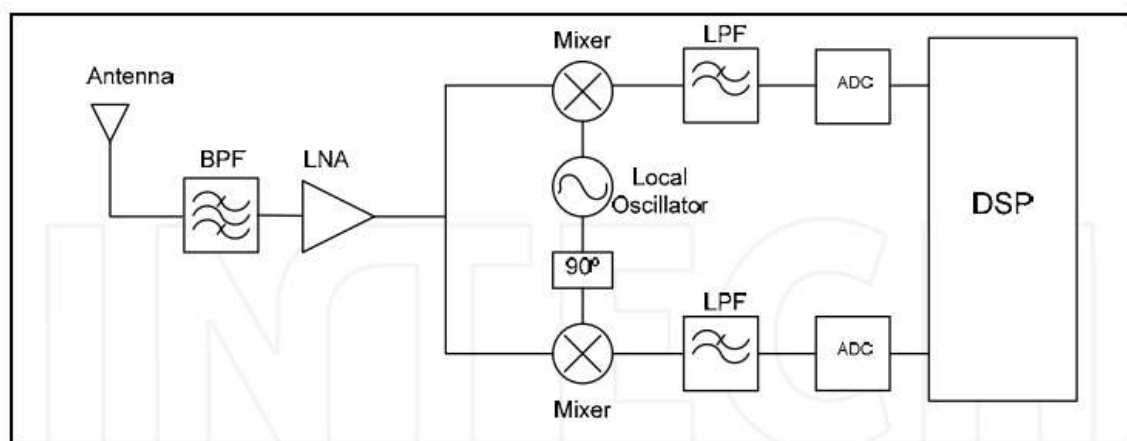


Figura 7. Arquitectura de un receptor de frecuencia intermedia cero. Otto Limann. (1989). Fundamentos de radio. España: MARCOMBO, S.A

La señal recibida pasa por el filtro pasabanda y luego es amplificado por el LNA, luego finalmente, se hace una baja conversión de la señal para ser entregada a un convertidor Analógica-Digital. En comparación con la arquitectura del receptor superheterodino, esta tiene una clara reducción en el número de componentes analógicos y garantiza un alto nivel de integración gracias a su simplicidad.

1.6.3 Receptor de Baja Frecuencia Intermedia

Una configuración similar a las antes mencionadas es la del receptor de baja frecuencia intermedia, en donde la señal se mezcla hasta un no cero bajo o moderada frecuencia intermedia en vez de hacer la baja conversión directa, usando la baja conversión de Radio Frecuencia en cuadratura. Tal como lo muestra la figura 8. En esta solución se intentan combinar la arquitectura del receptor superheterodino y el de baja frecuencia.

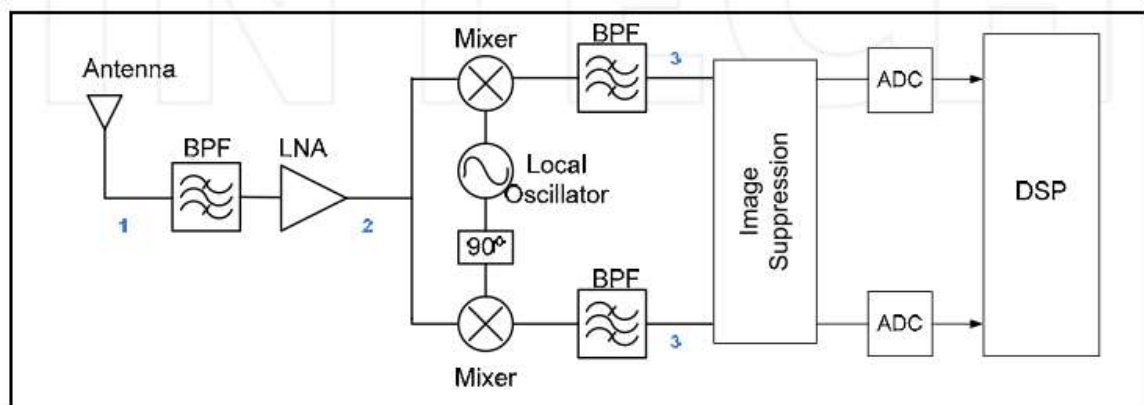


Figura 8. Arquitectura de un receptor de Baja Frecuencia Intermedia. Otto Limann. (1989). Fundamentos de radio. España: MARCOMBO, S.A

Esta arquitectura mantiene la permisividad de la integración de alto nivel, lo que es una ventaja al receptor de frecuencia intermedia cero. Sin embargo, esta continua sufriendo de la frecuencia imagen y problemas de desajuste con la modulación I/Q y el consumo de energía del convertidor Análogo-Digital se incrementa ya que se requiere ahora una alta tasa de conversión.

CAPÍTULO II. VIRTUALIZACIÓN DE CENTRALES TELEFÓNICAS: HERRAMIENTAS Y TECNOLOGÍAS.

2.1 Softwares

2.1.1 Configuración de Asterisk

Existen múltiples maneras de instalar asterisk, mediante repositorios o mediante la descarga directa, en esta ocasión instalaremos por descarga directa. Para poder instalar Asterisk lo primero que debemos tener en cuenta es que debe ser instalada en un sistema operativo Linux, en nuestro caso vamos a utilizar Ubuntu porque es una plataforma muy parecida a la interfaz de Windows y es más fácil de manejar debido a que maneja un entorno gráfico. Además es preciso instalar las dependencias a las cuales está ligada Asterisk, las cuales son:

- DAHDI
- LibPRI
- Asterisk-addons

Para la instalación de Asterisk en Ubuntu 12.4.05 LTS, abriremos la terminal y lo primero que tenemos que hacer es asegurarnos de que el sistema operativo este actualizado y para esto utilizaremos los siguientes comandos:

```
Sudo apt-get update && apt-get upgrade -y && reboot
```

Luego de que se terminen de descargar las actualizaciones y se apliquen al Sistema, el equipo se va a reiniciar y procedemos nuevamente a abrir la terminal de Ubuntu para instalar las dependencias básicas, para esto ejecutamos el comando:

```
Sudo apt-get install build-essential wget libssl-dev libncurses5-dev libnewt-dev libxml2-dev linux-headers-$(uname -r) libsqlite3-dev uuid-dev
```

Una vez resueltas las librerías básicas, descargamos las dependencias que Asterisk va a utilizar, primero las descargamos y luego procederemos a descomprimirlas para poder instalarlas, vamos a comenzar con DAHDI, para ello utilizamos el comando:

```
Sudo wget http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-current.tar.gz
```

Con este comando descargaremos la dependencia DAHDI. El Digium Asterisk Hardware Device Interface, o DAHDI, es el software que usa Asterisk para interactuar con el hardware de telefonía. Se recomienda instalarlo porque DAHDI es una dependencia requerida para usar aplicaciones de Asterisk como Meetme(). DAHDI es actualmente una combinación de dos códigos base separados: DAHDI-tools, el cual proporciona varias herramientas de administrador y DAHDI-linux, el cual proporciona los controladores (drivers) del kernel.

El siguiente paso a realizar es descargar LibPri, esto lo descargamos con el siguiente comando:

```
sudo wget http://downloads.asterisk.org/pub/telephony/libpri/libpri-1.4-current.tar.gz
```

El LibPri es una librería que añade soporte para la RDSI (Integrated Services Digital Network, Red Digital de Servicios Integrados (PRI y BRI). El uso de LibPRI es opcional, toma muy poco tiempo en instalar, no interfiere en el funcionamiento básico de Asterisk y será muy útil si alguna vez desea agregar tarjetas a un sistema en un momento posterior.

Los paquetes de Asterisk-addons proveen controladores para la conexión a servidores de mysql y manejo de bases de datos, además de proveer de controladores para el manejo de archivos en mp3, entre otros. Su instalación es opcional. Esto básicamente se utiliza para poner canciones como llamadas en espera o si la grabación del contestador se graba en un archivo .mp3 se puede cargar para ser reproducido por Asterisk. Para descargarlo se ejecuta el siguiente comando:

```
sudo wget http://downloads.asterisk.org/pub/telephony/asterisk/old-releases/asterisk-addons-1.6.2.4.tar.gz
```

Una vez descargados todos los paquetes procedemos a descomprimirlos para poder ejecutar su instalación y configuración. Esto lo hacemos con los siguientes comandos:

```
sudo tar zxvf dahdi-linux-complete*
```

```
sudo tar zxvf libpri*
```

```
sudo tar zxvf asterisk*
```

Luego de haber extraído los archivos, procedemos a la instalación de los componentes posteriormente su configuración, lo hacemos con los siguientes comandos:

Para DAHDI:

```
cd dahdi-linux-complete-2.6.0+2.6.0
```

```
sudo make all
```

```
sudo make install
```

```
sudo make config
```

Para LibPri:

```
cd libpri-1.4.12
```

```
sudo make install
```

Para Asterisk-addons:

```
cd asterisk-addons-1.6.2.4
```

```
sudo ./configure
```

```
sudo make
```

```
sudo make install
```

Como ya terminamos de instalar las dependencias procedemos a instalar Asterisk lo cual haremos con los siguientes pasos y comandos:

Primero descargamos Asterisk

```
sudo wget http://downloads.asterisk.org/pub/telephony/asterisk/ asterisk-11-current.tar.gz
```

Luego descomprimos el archivo

```
sudo tar xvzf asterisk-10-current.tar.gz
```

Luego entramos al directorio donde esta descargado para comenzar la compilación

```
cd asterisk-11.18.0/
```

Y luego ejecutamos

```
sudo ./configure
```

Cuando el comando termine de ejecutarse aparecerá una imagen como la figura9. En el cual aparece el símbolo de Asterisk en caracteres especiales.



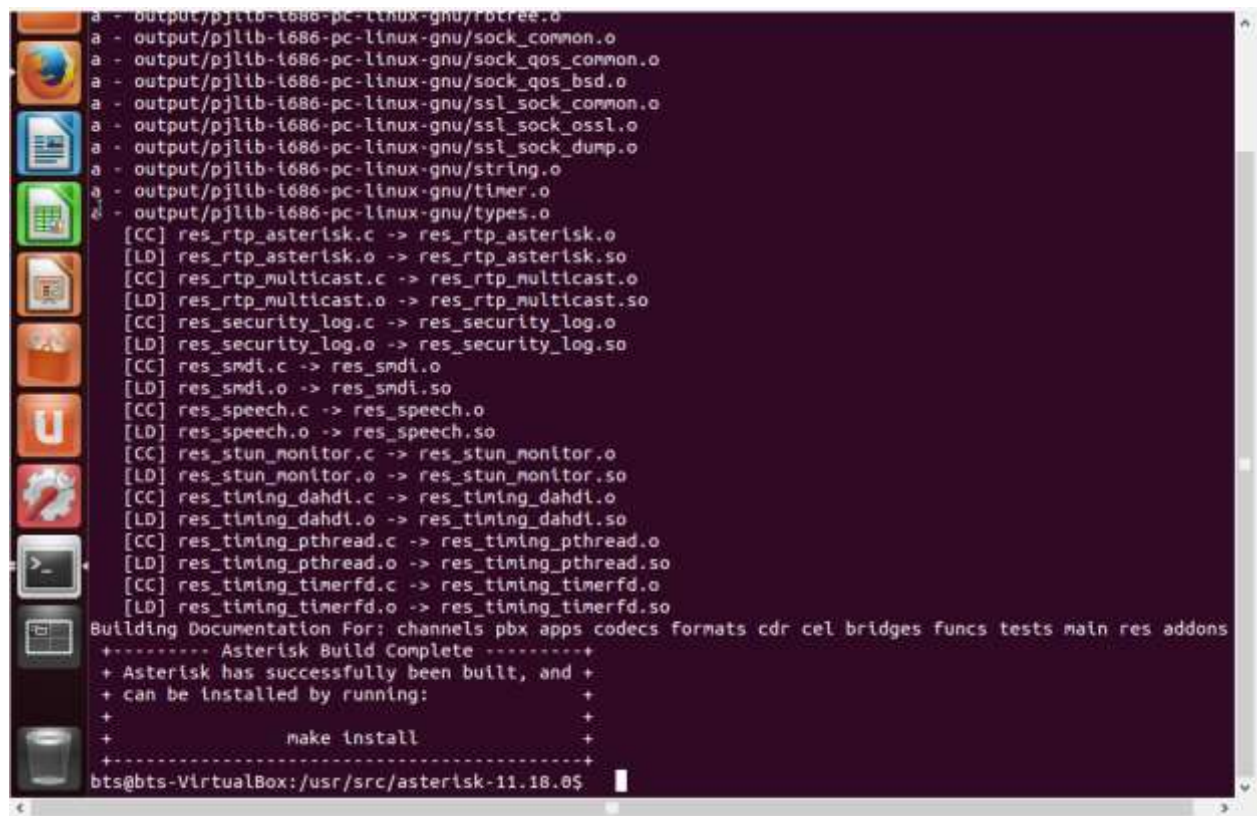
Figura 9. Instalación de Asterisk en Ubuntu 12.04.05 LTS.

Esto nos indica que ya estamos listos para proceder con la instalación de Asterisk, para esto volvemos a irnos al directorio de asterisk y comenzamos la instalación:

```
cd ~/asterisk
```

```
sudo make
```

Al finalizar esto, nos saldrá una imagen como la figura 10:



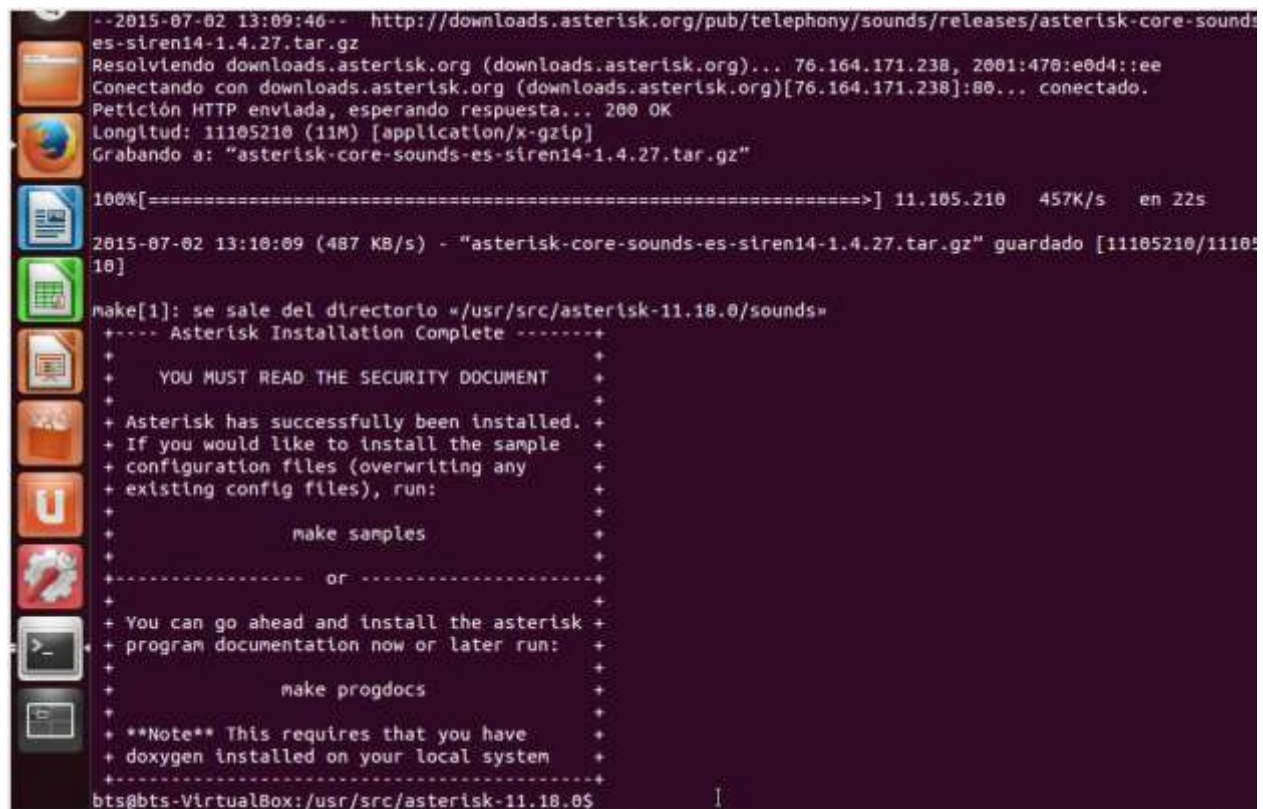
```
a - output/pjlib-t686-pc-linux-gnu/rotree.o
a - output/pjlib-t686-pc-linux-gnu/sock_common.o
a - output/pjlib-t686-pc-linux-gnu/sock_qos_common.o
a - output/pjlib-t686-pc-linux-gnu/sock_qos_bsd.o
a - output/pjlib-t686-pc-linux-gnu/ssl_sock_common.o
a - output/pjlib-t686-pc-linux-gnu/ssl_sock_ssl.o
a - output/pjlib-t686-pc-linux-gnu/ssl_sock_dump.o
a - output/pjlib-t686-pc-linux-gnu/string.o
a - output/pjlib-t686-pc-linux-gnu/timer.o
a - output/pjlib-t686-pc-linux-gnu/types.o
[CC] res_rtp_asterisk.c -> res_rtp_asterisk.o
[LD] res_rtp_asterisk.o -> res_rtp_asterisk.so
[CC] res_rtp_multicast.c -> res_rtp_multicast.o
[LD] res_rtp_multicast.o -> res_rtp_multicast.so
[CC] res_security_log.c -> res_security_log.o
[LD] res_security_log.o -> res_security_log.so
[CC] res_sndi.c -> res_sndi.o
[LD] res_sndi.o -> res_sndi.so
[CC] res_speech.c -> res_speech.o
[LD] res_speech.o -> res_speech.so
[CC] res_stun_monitor.c -> res_stun_monitor.o
[LD] res_stun_monitor.o -> res_stun_monitor.so
[CC] res_tuning_dahdi.c -> res_tuning_dahdi.o
[LD] res_tuning_dahdi.o -> res_tuning_dahdi.so
[CC] res_tuning_pthread.c -> res_tuning_pthread.o
[LD] res_tuning_pthread.o -> res_tuning_pthread.so
[CC] res_tuning_timerfd.c -> res_tuning_timerfd.o
[LD] res_tuning_timerfd.o -> res_tuning_timerfd.so
Building Documentation For: channels pbx apps codecs formats cdr cel bridges funcs tests main res addons
+----- Asterisk Build Complete -----+
+ Asterisk has successfully been built, and +
+ can be installed by running:             +
+                                         +
+               make install                +
+-----+
bts@bts-VirtualBox:/usr/src/asterisk-11.18.0$
```

Figura 10. Indicación de instalación de Asterisk lista para configurar.

Entonces es hora de ejecutar el último comando para terminar la instalación.

Sudo make install

Al terminar de ejecutarse este comando, sale una imagen como la de la figura 11, indicando que ya se ha instalado correctamente.



```
--2015-07-02 13:09:46-- http://downloads.asterisk.org/pub/telephony/sounds/releases/asterisk-core-sounds-es-siren14-1.4.27.tar.gz
Resolviendo downloads.asterisk.org (downloads.asterisk.org)... 76.164.171.238, 2001:470:e0d4::ee
Conectando con downloads.asterisk.org (downloads.asterisk.org)[76.164.171.238]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 11105210 (11M) [application/x-gzip]
Grabando a: "asterisk-core-sounds-es-siren14-1.4.27.tar.gz"
100%[=====>] 11.105.210 457K/s en 22s
2015-07-02 13:10:09 (487 KB/s) - "asterisk-core-sounds-es-siren14-1.4.27.tar.gz" guardado [11105210/11105210]

make[1]: se sale del directorio "/usr/src/asterisk-11.18.0/sounds"
+----- Asterisk Installation Complete -----+
+
+ YOU MUST READ THE SECURITY DOCUMENT +
+
+ Asterisk has successfully been installed. +
+ If you would like to install the sample +
+ configuration files (overwriting any +
+ existing config files), run: +
+
+ make samples +
+
+ ----- or ----- +
+
+ You can go ahead and install the asterisk +
+ program documentation now or later run: +
+
+ make progdocs +
+
+ **Note** This requires that you have +
+ doxygen installed on your local system +
+
+-----+
bts@bts-VirtualBox: /usr/src/asterisk-11.18.0$
```

Figura 11. Indicación de que Asterisk está correctamente instalado.

2.1.2 Configuración de OpenBTS

Para descargar e instalar el OpenBTS lo haremos desde la página de gnuradio.org utilizando el comando:

```
wget http://gnuradio.org/redmine/projects/gnuradio/wiki/OpenBTSUHD
```

Luego que se descargue procedemos a compilar e instalar:

```
cd openbts/public-trunk/
```



```
./configure
```

```
make
```

```
sudo make install
```

```
reboot now
```

Con esto queda instalado el OpenBTS, entonces lo que haremos a continuación es modificar las configuraciones que vienen precargadas en OpenBTS para que funcionen con la red que vamos a crear, para ello vamos a modificar el archivo llamado “OpenBTS.config” con el editor de texto, este se encuentra en el directorio “/apps” en la raíz donde instalamos OpenBTS.

“Aunque la red GSM trabaja en varias bandas de frecuencia, OpenBTS puede trabajar en 4 de las más usadas: GSM850, GSM900, GSM1800 y GSM1900. Cuando se describe la red GSM se habla sobre un sistema numérico de identificación. El IMSI es el número de identificación más importante dentro del archivo de configuración de OpenBTS y en Asterisk representa el nombre de usuario SIP.

Como se mencionó al hablar del sistema específico de numeración GSM, el IMSI es un número que identifica de forma única una MS dentro de la red. El IMSI se graba en la tarjeta SIM cuando el suscriptor se registra con el operador de telefonía móvil determinado en algún país. Tal como se muestra en la figura 12, el máximo número de dígitos del IMSI es 15 y está

compuesto del MCC, el MNC y el MSIN”. Julián David Vásquez Gutiérrez, Iván Fernando Santa Ramírez, José Valentín Restrepo Laverde. (2012). Prototipo De Una Estación Celular Portátil Para Atención De Emergencias. Medellin, Colombia.

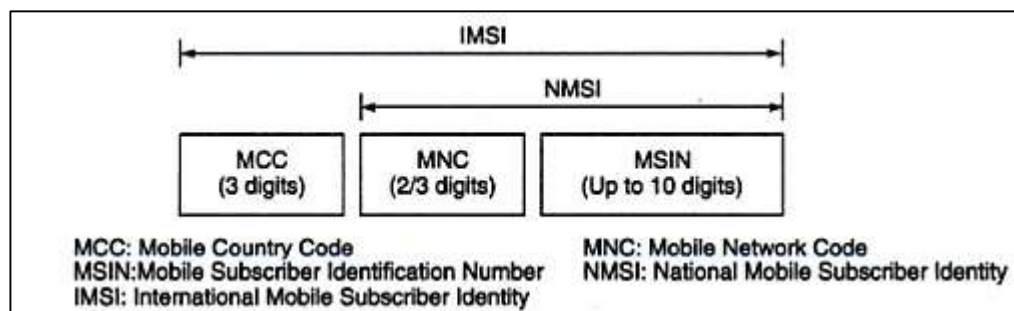


Figura 12. Diagrama del IMSI. Sumit Kasera, Nishit Naran. (2004). 3G Networks: Architecture, protocols and Procedures. New Delhi: Tata McGraw-Hill.

En la figura 13 se muestra una tabla con la lista de MCC y MNC utilizados en República Dominicana incluyendo también las frecuencias utilizadas por cada operadora.

República Dominicana				
370	01	Orange Dominicana	Operativa	GSM 850 / GSM 900 / GSM 1800 / GSM 1900 / WCDMA 900
370	02	Compañía Dominicana de Teléfonos, C por A. (Claro)	Operativa	CDMA2000 1900/ GSM 850/ GSM 1900/ WCDMA 850
370	03	Tricom S.A.	Operativa	AMPS 850/ CDMA 850
370	04	Viva Dominicana	Operativa	CDMA2000 1900/GSM 1900
370	05	Wind Telecom, S.A.	Operativa	WIMAX / GSM 1900

Tabla 1. Listado de MCC y MNC de Republica Dominicana, estos fueron asignados por la UIT (Unión Internacional de Telecomunicaciones) en su publicación: E.212: Plan de identificación internacional para redes públicas y suscripciones, aprobada en Mayo 2008.

Para modificar las configuraciones ejecutamos el editor para poder cambiar los parámetros, esto se hace con:

```
gedit ~/openbts-2.6.0Mamou/apps/OpenBTS.config
```

En este archivo se encuentran las configuraciones a las que trabajara la red, como la frecuencia a utilizar, los MCC, MNC y MSIN así como también el ARFCN (Absolute Radio-Frequency Channel Number o número de canal absoluta de radiofrecuencia).

Cuando coloquemos los datos que vamos a utilizar, guardamos y cerramos el archivo, luego ejecutamos:

Reboot now

Con esto queda configurado el OpenBTS para la interacción con Asterisk.

2.2 Hardware y Dongles

2.2.1 USRP (Universal Software Radio Peripheral o Periférico de Radio de Software Universal)

“USRP es un dispositivo de Hardware libre, que en conjunto con un computador permite implementar y diseñar sistemas de radiocomunicaciones potentes, flexibles a muy bajo costo y mínimo esfuerzo. Para probar su completo valor simplemente es necesario descargar e instalar GNU Radio.

La potente combinación de Hardware y Software libre se convierte en la plataforma ideal para que un computador convencional funcione como un software radio de alto ancho de banda. La gran comunidad de desarrolladores y usuarios han contribuido a la filosofía de diseño básico detrás del USRP que tiene como objetivo realizar todo el procesamiento de señales específicas como modulación, demodulación, interpolación. Todo lo anterior en un computador sin tener que comprar ningún software o pagar una licencia.

GNU Radio no es la única opción, USRP presenta un enorme nivel de flexibilidad que se ajusta a las opciones de los usuarios. Algunos de ellos han creado su propio ambiente SDR para correr sobre USRP, mientras otros han usado USRP integrado con software como LabVIEW® o MATLAB®/Simulink®.

A medida del crecimiento en el uso del USRP se ha ido creando un conjunto de productos que han sido agrupados dentro de lo que la ETTUS RESEARCH LLC ha denominado la familia de productos USRP. El USRP1 es el hardware original de la familia de productos USRP; está conformado por unos componentes necesarios para el procesamiento de señales y la implementación de aplicaciones de radio.

El montaje completo del USRP1 cuenta con 2 niveles de tarjetas: el primero es la tarjeta madre (motherboard) en donde se puede identificar la FPGA, la alimentación, la conexión vía USB y los 4 slots para conectar el segundo nivel conformado por las tarjetas hijas (daughterboards), que proporcionan flexibilidad, integrando completamente un RF front end que es implementado por medio de estas tarjetas hijas añadidas a el USRP1.

El USRP1 tiene cuatro ADC's de alta velocidad, cada uno a 12 bits por muestra, con una tasa de 64 millones de muestras por segundo (64 MSPS); en teoría se podría muestrear una señal de hasta 32 MHz. Cuenta con un PGA (Programmable Gain Amplifier, Amplificador de Potencia Programable) antes de los ADC's para amplificar la señal de entrada y utilizar el rango completo en caso de que la señal sea débil.

También tiene 4 DAC's de alta velocidad para transmisión, cada uno a 14 bits por muestra y una tasa de 128 millones de muestras por segundo (128 MSPS), contando de igual forma con un PGA después de los DAC's que proporcionan hasta 20 dB de ganancia. Estos 4 canales de entrada y 4 canales de salida son conectados a una FPGA (Field-Programmable Gate Array, Matriz de Compuertas Programables en Campo) Altera Cyclone EP1C12, la cual se conecta a un chip de interfaz USB2.0 (Universal Serial Bus versión 2, Bus Serial Universal versión 2), el Cypress FX2, y luego al computador. Hay que aclarar que la conexión del USRP1 al computador se realiza con una interfaz USB2.0, no trabaja con USB1.1. La FPGA es la parte más importante en el sistema del USRP1.

Básicamente lo que hace es realizar operaciones matemáticas de alto ancho de banda y reducir la tasa de datos para que puedan ser enviados a través de la interfaz USB2.0 al computador. En el USRP1, el procesamiento con alta frecuencia de muestreo se realiza en la FPGA, mientras el procesamiento con baja frecuencia de muestreo se realiza en el computador.

La configuración básica de la FPGA incluye dos DDC's completos, pero también es posible la implementación de 4 DDC's sin filtros de media banda. Esto permite tener 1, 2 o 4 canales de recepción separados. Las salidas de los ADC's van conectadas a las entradas de los DDC's. Los DDC's mezclan, filtran y diezman (desde 64 MHz) señales de entrada en la FPGA. Se utilizan en la recepción, esencialmente por dos razones: primero, para convertir la señal de banda de frecuencia intermedia a una señal en banda base y, segundo, para diezmar la señal, logrando

que la tasa de datos pueda ser adaptada a la interfaz USB2.0 y que sea razonable a la capacidad de procesamiento de los computadores. En la transmisión se realiza el proceso inverso: se necesita convertir una señal banda base a una señal de frecuencia intermedia, y enviarla a través los DAC's.

Esto se hace a través de los dos DUC's. En el lado de la transmisión también se usan filtros interpoladores CIC (Cascaded Integrator-Comb, Peine Integrador en Cascada) que interpolan las muestras antes de trasladar la señal digital a la frecuencia intermedia por los DUC's. Los DDC's y DUC's combinados con las altas tasas de muestreo simplifican en gran medida los requerimientos de filtrado analógico.

La FPGA está programada con el lenguaje de descripción de hardware Verilog y sintetizada con la edición web libre de Altera, Quartus II. El PCB que viene diseñado con la herramienta PADS. Los esquemáticos están hechos en gEDA (GPL Electronic Design Automation, GPL Automatización de Diseño Electrónico).

En la Figura 14 se puede ver el diagrama de bloques del USRP1 incluyendo las tarjetas hijas que forman el RF front end". Julián David Vásquez Gutiérrez, Iván Fernando Santa Ramírez, José Valentín Restrepo Laverde. (2012). Prototipo De Una Estación Celular Portátil Para Atención De Emergencias. Medellin, Colombia.

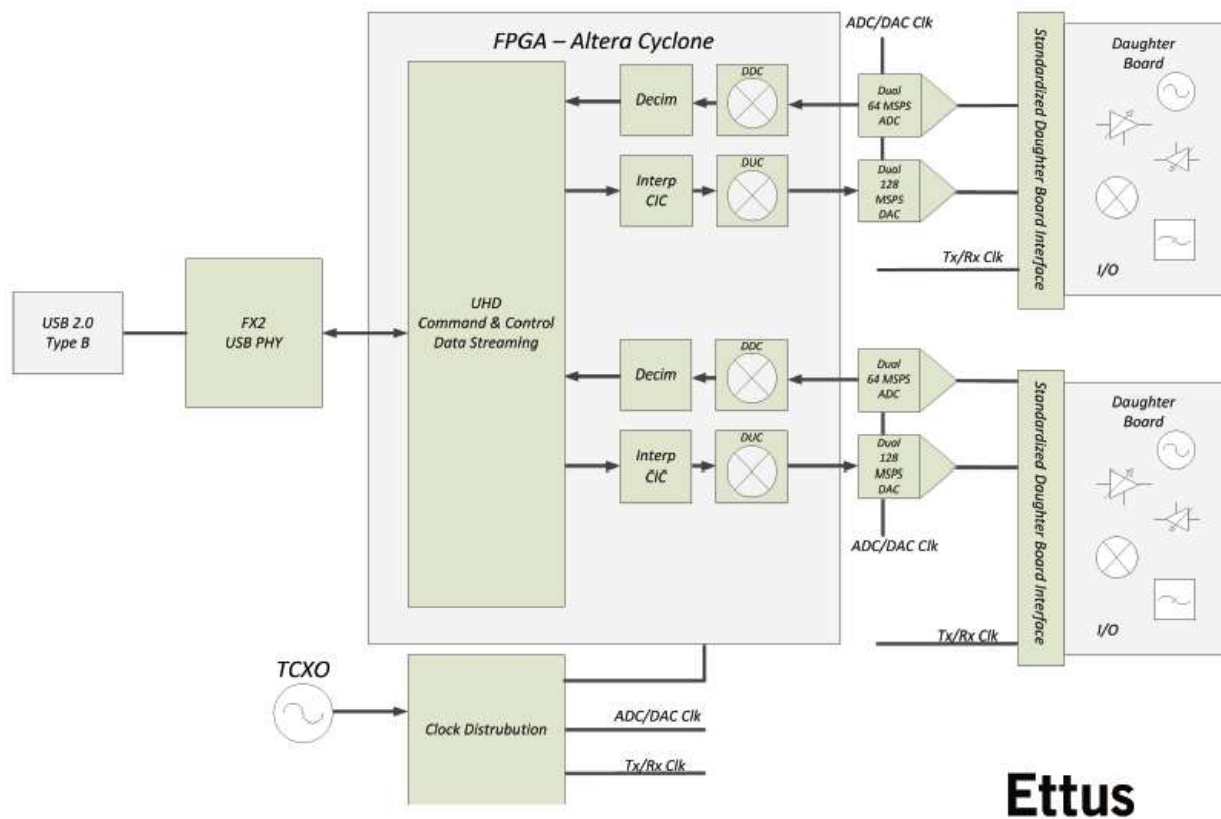


Figura 13. Diagrama de bloques del USRP1, incluyendo las tarjetas hijas o daughter board.

Ettus Research. (2012). USRP motherboard datasheet. 10/7/2015, de ETTUS RESEARCH LLC

Sitio web: http://www.ettus.com/content/files/07495_Ettus_USRP1_DS_Flyer_HR.pdf

2.2.2 SDR (Radio definido por software o Software Defined Radio)

“Un Software Radio es una radio cuyo canal modulaciones formas de onda se define en el software. Es decir, las formas de onda se generan como señales digitales muestreadas, convertida de digital a analógico a través de banda ancha DAC (convertidor de digital a analógico) y el posiblemente no convertidos de IF (Frecuencia Intermedia) de RF (Radio Frecuencia). El receptor, de manera similar, emplea un ADC de banda ancha (Convertidor

Analógico a Digital) que captura todos los canales del nodo radio de software. El receptor extrae, por conversos y desmodula la forma de onda de canal utilizando software en un procesador de propósito general.” Joseph Mitola. (1992). Software Radio: Survey, Critical Analysis and Future Directions. IEEE, 12, 13/23.

“La radio definida por software elimina problemas existentes en la tecnología anterior (hardware) como el alto coste de producción y la poca flexibilidad ante diferentes formas de onda. Algunos dispositivos que utilizan radio definida por Software son field programmable gate arrays (FPGA), procesado de señal digital (DSP) and Sistema programable on Chip (SoC).

Características:

Reconfigurabilidad: SDR permite la coexistencia de múltiples módulos software implementados con diferentes estándares en el mismo sistema que permite la configuración dinámica del sistema con tan solo seleccionar el módulo de software correspondiente para ejecutar.

Conectividad Ubicua: SDR permite la implementación de estándares de interferencia de aire como módulos software y múltiples instancia de estos módulos que implementan diferentes estándares pueden coexistir en el equipo de infraestructura y los terminales.

Interoperabilidad: SDR facilita la implementación de la arquitectura abierta de sistemas radio.

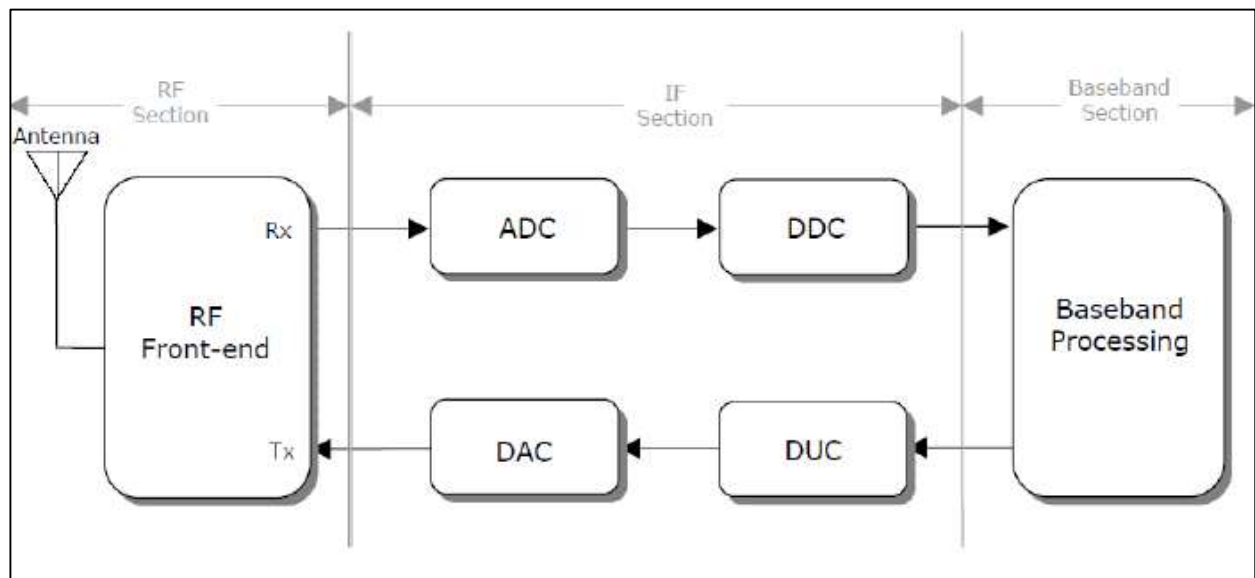


Figura 14. Arquitectura del SDR. María Dolores Cereceda Tobal. (Octubre 2011). Diseño y evaluación de un detector de energía para señales fm wireless microphone en radio cognitiva. Cartagena, Colombia: ETSIT, Escuela Técnica Superior de Ingeniería de Telecomunicación.

Tal como se muestra en la figura 15, La sección RF es la responsable de adaptar la señal. En recepción la señal de radio frecuencia es transformada a una señal de frecuencia intermedia. En transmisión, la señal analógica es transformada y amplificada.

En la sección IF, los bloques ADC transforma la señal analógica en digital y los bloques DAC transforman la señal digital en analógica. En recepción, los bloques DDC realizan una digital-Down-conversion (demodulación), en transmisión, los bloques DUC realizan una digital-up-conversion (modulación). Estos dos últimos bloques son programables. En lugar de utilizar IF, una aproximación de zero-IF puede ser elegida, la RF front-end realiza una conversión descendente directa a banda base. La sección de banda base realiza operaciones en banda base e implementa, mediante software, el protocolo de la capa de enlace.” María Dolores Cereceda Tobal. (Octubre 2011).

Diseño y evaluación de un detector de energía para señales fm wireless microphone en radio cognitiva. Cartagena, Colombia: ETSIT, Escuela Técnica Superior de Ingeniería de Telecomunicación.

2.3 Integración de las partes

2.3.1 Interacción de OpenBTS y USRP1

Para que OpenBTS interactúe con el hardware USRP1 tenemos que instalar GNU radio en nuestro servidor, esta aplicación es la encargada de hacer la gestión entre el software y el hardware, además es la aplicación que maneja los controladores.

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.

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 está 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.

Aunque no es una herramienta principalmente de simulación, GNU Radio complementa el desarrollo de algoritmos de procesamiento de señales a partir de datos previamente grabados o generados, evitando la necesidad de hardware de RF. GNU Radio está licenciado bajo la GNU General Public License (GPL) versión 3. Todo el código es propiedad de la Fundación del Software Libre.

Para descargar GNU radio

```
wget http://gnuradio.org/releases/gnuradio/gnuradio-3.4.2.tar.gz
```

Luego descomprimos el archivo e ingresamos a la carpeta donde están los archivos:

```
tar -zxvf gnuradio-3.4.2.tar.gz
```

```
cd gnuradio-3.4.2
```

Antes de la instalación debemos descargar algunas dependencias que son necesarias para instalar GNU Radio:

```
sudo apt-get -y install libfontconfig1-dev libxrender-dev libpulse-dev swig g++-4.3 automake  
autoconf libtool python-dev libfftw3-dev \ libcppunit-dev libboost-all-dev libusb-dev fort77  
sdcc sdcc-libraries \libsdl1.2-dev python-wxgtk2.8 git-core guile-1.8-dev \ libqt4-dev python-  
numpy ccache python-opengl libgsl0-dev \ python-cheetah python-lxml doxygen qt4-dev-  
tools \ libqwt5-qt4-dev libqwtplot3d-qt4-dev pyqt4-dev-tools python-qwt5-qt4
```

Ahora procedemos a hacer la instalación, esto en ocasiones tarda un tiempo largo, no es recomendable cerrar la terminal hasta que la instalación se haya completado:

```
./configure
```

```
make && make check
```

```
sudo make install
```

Si nos presenta un error indicando que el intérprete Python no encuentra el directorio de las librerías, se debe agregar la línea `/usr/local/lib` al archivo `ld.so.conf`. Esto normalmente es un problema que tiene Ubuntu, el archivo debe quedar justo como la figura 16.



Figura 15. Modificación del archivo `ld.so.conf` para poder hacer la configuración de GNU radio.

Luego de se presiona “Guardar” se cierra el editor y se ejecutan los comandos anteriores para que se complete la instalación del GNU radio. Una vez este proceso termine queda instalado correctamente la aplicación.

2.3.2 Conexión y reconocimiento del USRP1

Se conecta el USRP1 a la energía, y se conecta al computador vía USB, luego abrimos una terminal para escanear la red y seleccionar la ARFCN, esto lo hacemos desde la terminal con el siguiente comando:

```
kal -s GSM850 -F 52000000 -R B
```

Con esto se mostraran una lista de los canales disponibles y su potencia de transmisión, en este caso la mayor Potencia de transmisión es 137 por lo que seleccionamos esta, quedando el texto de la siguiente manera:

```
GSM.Band 850
```

```
$static GSM.Band
```

```
GSM.ARFCN 137
```

```
$static GSM.ARFCN
```

Y por último verificamos la dirección IP del servidor Asterisk. En este caso el servidor corre localmente con OpenBTS y se deja la IP por defecto 127.0.0.1.

Asterisk.IP 127.0.0.1

SIP.IP 127.0.0.1

CAPÍTULO III. PLANTEAMIENTO DE LA RED UTILIZANDO OPENBTS Y ASTERISK

3.1 Criterios de Planificación de la red

En esta primera parte del capítulo se van a establecer los parámetros que deben ser tomados en consideración al momento de implementar o planificar una red GSM, los cuales son: Calidad de servicio, tasa de llamadas exitosas y tasa de llamadas rechazadas.

3.1.1 Calidad de Servicio

La función de este parámetro es medir el funcionamiento u operación adecuado de la red que se va a implementar. Es una forma de tener cuantitativamente registrada el rendimiento del sistema. La calidad de servicio se subdivide en 4 medidas referenciados en la tabla 1, en la misma se muestran también los valores mínimos que puede tener cada parámetro, esto puede variar según el SLA (Service Level Agreement o Acuerdo de nivel de Servicio) que tenga el cliente con el proveedor.

Parámetro	Unidad	Utilidad	Valor Mínimo
Ancho de Banda	Kb/s	muestra la cantidad de datos que se puede transmitir	≥ 2000
Jitter	ms	La fluctuación que se puede producir en el Retardo	≤ 20 ms
Tasa de Perdidas	%	Razón de paquetes perdidos respecto de los enviados	$\leq 0,01$ %
Retardo o Latencia	ms	Es el promedio de tiempo que tardan en llegar los paquetes	≤ 20 ms

Tabla 2. Parámetros de medida de Calidad de Servicio

3.1.2 Tasa de llamadas exitosas

De acuerdo con la recomendación ITU-T B.18: en una red de telecomunicaciones, la intensidad de tráfico instantánea $A(t)$ en un conjunto de elementos de red es el número de elementos ocupados en un instante dado. Por lo general se considera 1 hora.

Pueden calcularse momentos estadísticos para un periodo de tiempo dado; por ejemplo, la intensidad de tráfico media está relacionada con la intensidad de tráfico instantánea $A(t)$ por la siguiente expresión:

$$\overline{A}(t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} A(t) dt$$

NOTA – El CCIF ("Le comité consultatif international des communications téléphoniques à grande distance" predecesor del CCITT y del ITU-T) dio en 1946 el nombre de «erlang» a la unidad de tráfico, en honor del matemático danés A.K. Erlang (1878-1929), que fue el fundador de la teoría del tráfico en telefonía.

La unidad de intensidad de tráfico se denomina Erlang, cuyo símbolo es E. Un erlang es la intensidad de tráfico en un conjunto de órganos, cuando sólo uno de ellos está ocupado de manera continua. Cuando el tráfico es de un (1) erlang significa que el elemento de red está totalmente ocupado durante el tiempo de medición, normalmente una hora.

Si una línea está ocupada durante una hora entonces cursa un tráfico de 3600 llamadas-segundos que a 36 llamadas de 100 seg de duración cada una, o a cualquier otra combinación que resulte en 3600 llamadas-segundo. Si 100 usuarios solicitan una llamada con una duración promedio de 3 minutos entonces el tráfico es:

$$A = \frac{100 \times 3 \times 60}{3600} = 5 \text{ Erlang}$$

Centi-Call Seconds – CCS

A parte del Erlang, también se usa el CCS (Centi-Call Seconds) como unidad de tráfico. 1 CCS equivale a una llamada con duración de cien segundos, de esta forma el tráfico en una línea ocupada totalmente durante una hora es de 36 CCS, por lo tanto:

$$1\text{erlang} = 36 \text{ CCS}$$

De esta forma si tenemos el tráfico en CCS, lo dividimos por 36 para obtener los Erlang respectivos. Si por el contrario, el tráfico está en Erlang lo multiplicamos por 36 para llevarlo a CCS.

Grado De Servicio

Es un atributo de calidad de servicio usado en la comunicaciones telefónicas en particular, y en general en los servicios basados en conmutación de circuitos, y se refiere a la probabilidad de bloqueo en el primer intento de una llamada, durante la hora pico, y se expresa como P_x donde x es menor que 1 y representa la probabilidad de bloqueo; por ejemplo $P.01$ significa que existe un 1% de probabilidad de bloqueo en la hora pico. Mientras más bajo es el grado de servicio es menor la probabilidad de bloqueo y por supuesto mejor el desempeño de la red. La mayoría de las redes de telefonía fija se diseñan para $P.01$ y las redes celulares para $P0.02$. El bloqueo ocurre cuando estando todos los recursos ocupados se trata de hacer una llamada la cual no puede ser atendida por la red. Cuando todas las líneas

conectadas entre los nodos de la red están ocupadas también ocurre el bloque, si se trata de hacer una llamada a un abonado de otra central.

3.1.3 El modelo Erlang B

Conociendo el tráfico y la cantidad de líneas disponible, este modelo calcula la probabilidad PB de que una llamada en su primer intento sea bloqueada y está basado en las siguientes premisas

1. La cantidad de usuarios es muy grande
2. Las llamadas llegan en forma aleatoria de acuerdo a una distribución de Poisson
3. Las llamadas se atienden según el orden de llegada
4. Las llamadas bloqueadas se pierden. Modelo con pérdidas, no hay lista de espera.
5. El tiempo de duración de las llamadas siguen una distribución exponencial

$$P_B = \frac{\frac{A^N}{N!}}{\sum_{i=0}^N \frac{A^i}{i!}}$$

Donde

A: es el tráfico en erlang durante la hora pico

N: cantidad de líneas del sistema

El modelo Erlang B ha sido ampliamente usado en el diseño de redes y sus resultados se encuentra tabulados pero son algo engorroso de usar; dichas tablas están diseñadas de manera que conociendo el tráfico en erlangs y el grado de servicio o probabilidad de bloqueo se pueda obtener la cantidad de líneas mínima para cursar el tráfico. Actualmente existen muchos sitios en Internet que permiten hacer dichos cálculos.

Ejemplo: Si el tráfico es de 5 erlang y el grado de servicio es 0.01, entonces la cantidad de líneas necesarias es 11, con lo que obtiene un grado de servicio 0.00829.

Intensity	GOS	Trunks	Actual GOS
5 E	0.01	11	0.00829

Figura 16. Modelo de tráfico *Atel Asesores C.A atel.asesores@cantv.net*
www.atelasesores.com.ve

3.2 Cálculo del presupuesto de potencia

Para establecer el alcance territorial que tendrá la red es necesarios conocer las especificaciones de los elementos que pertenecen al sistemas, tales como: pérdidas del espacio libre, potencia de transmisión ganancia de las antenas, entre otros. En la tabla 3 se muestras los elementos utilizados para el cálculo del presupuesto de potencia.

PARÁMETRO	DESCRIPCIÓN
S	Sensibilidad de la BTS o de la MS, que indica el grado de percepción de una señal en dBm.
P_T	Potencia de transmisión de la antena. En conjunto con la ganancia de la antena forman la PIRE (Potencia Isotrópica Radiada Equivalente) del transmisor en W (Vatios).
G	Ganancia de la antena transmisora-receptora.
G_D	Ganancia por diversidad. Ésta ganancia comúnmente se usa para balancear el enlace de bajada con el de subida y se ubica en el enlace de recepción de la BTS; usualmente se emplean 5dB para esta ganancia.
L_{LT}	Pérdidas de la línea de transmisión que incluyen las pérdidas totales del medio conductor empleado en conjunto con los conectores.
L_D	Pérdidas del duplexor.
L_{RPF}	Pérdidas del filtro pasa banda.
M_{DI}	Margen de degradación por interferencia.

Tabla 3. Descripción de los parámetros del presupuesto de potencia. E.E. Gaona García, M.A. Ávila Angulo, E.G. Muskus Rincón. (2014). Aproximación de la calidad de voz y cobertura en una red GSM de emergencia. Ciencia e Ingeniería Neogranadina, 24 (2), pp. 23 – 36

El primer elemento que se toma en cuenta es la estación móvil debido a que es el punto terminal del sistema y debemos conocer las características del receptor, luego seguimos con la BTS y por ultimo con la antena.

3.2.1 Cálculos para la Estación móvil

Según ETSI (European Telecommunications Standards Institute) la sensibilidad mínima que puede tener una estación móvil pequeña GSM 900 es -102 dBm por lo que se calcula por encima de esta cifra. La potencia máxima de transmisión la misma es de 39 dBm y la mínima es 29 dBm y el valor de la ganancia estándar es 0dBm por lo que los parámetros de la Estación Móvil son:

- $S_{MS} = -102 \text{ dBm}$
- $P_{TMS} = 39 \text{ dBm}$
- $G_{MS} = 0 \text{ dBm}$

3.2.2 Cálculos para la BTS

Los parámetros conocidos del USRP1 según el fabricante (ver anexo A) son su potencia de transmisión que es 32.62 mW (15 dBm) y un factor de ruido de 8 dB. Con estos

datos se calcula la sensibilidad de la USRP1 utilizando la figura de ruido y la relación mínima de $\frac{E_b}{N_0}$ de 8dB definida en la norma ETSI 05.05. La ecuación según Wayne Tomasi en su libro de Sistemas de comunicaciones Electrónicas para determinar la sensibilidad es la siguiente:

$$S_{BTS} = \frac{E_b}{N_0} F k T_0 W$$

Donde k, es la constante de Boltzman, $T_0=290K$ y el ancho de banda $W=271KHz$, (54dBHz). Teniendo la ecuación y los datos precedemos a hacer el cálculo de la sensibilidad de la BTS:

$$S_{BTS} = 8dB + 8dB + \left(290K \cdot 1.38 \times 10^{-23} \frac{J}{K} \right) + 54dBHz$$

$$S_{BTS} = 16 dB + 54dBHz - 174 dBm$$

$$\mathbf{S_{BTS} = 20dBm}$$

3.2.3 Antena

Como la antena que va a ser utilizada para la difusión GSM es una antena tipo dipolo su ganancia es 3dBi por consiguiente tenemos:

$$G_{BTS} = 3dB$$

3.3 Calculo de cobertura

Para realizar los cálculos de cobertura vamos a tomar en cuenta 3 modelos específicos que son normalmente usados para los enlaces de comunicaciones:

- MODELO OKMURA-HATA
- MODELO WALFISCH-IKEGAMI
- MODELO FRIIS

3.3.1 Modelo de Okmura-Hata

El modelo de Okmura-Hata es un modelo empírico que se desarrolló a partir de medidas realizadas en Tokio. De acuerdo con este modelo las pérdidas de propagación L definidas como

$$\frac{P_L}{P_T} = -L + G_T + G_R$$

Donde L es la pérdida de propagación, G_T es la ganancia de Transmisión y G_R es la ganancia de Recepción, y se pueden calcular para distancias $R > 1\text{KM}$ como:

$$\text{Zona urbana densa } L = A + B \log R - E \text{ (dB)}$$

$$\text{Zona urbana de baja densidad } L = A + B \log R - C \text{ (dB)}$$

$$\text{Zona rural } L = A + B \log R - D \text{ (dB)}$$

Donde R es la distancia medida en Kms y el cálculo de los parámetros A , B , C , D y E van a depender de la altura de la BTS (h_b), la altura de la antena terminal (h_m) (ambas expresadas en metros) y la frecuencia, expresada en Mhz. Se calculan como:

$$\begin{aligned}
A &= 69,55 + 26,16 \log f - 13,82 \log h_b \\
B &= 44,9 - 6,55 \log h_b \\
C &= 2(\log (f/28))^2 + 5,4 \\
D &= 4,78(\log f)^2 + 18,33 \log f + 40,94 \\
E &= 3,2(\log (11,75h_m))^2 - 4,97 \quad \text{ciudades grandes } f \geq 300 \text{ MHz} \\
E &= 8,29(\log (1,54h_m))^2 - 1,1 \quad \text{ciudades grandes } f < 300 \text{ MHz} \\
E &= (1,11 \log f - 0,7)h_m - (1,56 \log f - 0,8) \quad \text{ciudades medias y pequeñas}
\end{aligned}$$

Cabe destacar que el modelo tiene restricciones las cuales se muestran en la tabla 4 a continuación.

Parámetro	Mínimo	Máximo
Frecuencia	150 MHz	1500 MHz
Altura del Transmisor H _{Tx}	30 m	200 m
Altura del Receptor H _{Rx}	1 m	10 m

Tabla 4. Parámetros máximos y mínimos para el modelo de Okmura-Hata. Ángel Cardama Aznar, Lluís Jofre Roca, Juan Manuel Rius Casals, Jordi Romeu Robert, Sebastián Blanch Boris y Miguel Ferrando Bataller. (1998). Antenas. Barcelona: Edicions de la Universitat Politècnica de Catalunya.

3.3.2 Modelo Walfisch-Ikegami

Este modelo también es empírico, normalmente se utiliza para zonas urbanas pero también puede ser utilizado para las zonas rurales. Su ecuación es la siguiente:

$$a = 42.6 + 26 \log(d) + 20 \log(f)$$

Donde:

- a es la pérdida en la trayectoria medida en dB
- d es la distancia entre la BTS y la Estación móvil
- f es la Frecuencia medida en MHz

Los parámetros que limitan el uso de este modelo se ven en la tabla 5.

Parámetro	Mínimo	Máximo
Frecuencia	800 MHz	2 GHz
Altura del Transmisor H_{Tx}	4 m	50 m
Altura del Receptor H_{Rx}	1 m	3 m
Distancia entre Tx y Rx	20 m	5000 m

Tabla 5. Parámetros máximos y mínimos para el Modelo Walfisch-Ikegami. Peter Smyth. (2004). Mobile and Wireless Communications: Key Technologies and Future Applications. Reino Unido: British Telecommunications, PLC

3.3.3 Modelo Friis

Básicamente este modelo es muy similar al modelo de Walfisch-Ikegami, sin embargo hay parámetros que cambian. Este se deduce de las ecuaciones de Maxwell y permite calcular la potencia recibida a cierta distancia en condiciones ideales, es decir, sin obstáculos de ninguna naturaleza. Su ecuación es la siguiente:

$$a = 92.4 + 20 \log(d) + 20 \log(f)$$

Donde:

- a es la pérdida en la trayectoria medida en dB
- d es la distancia entre la BTS y la Estación móvil
- f es la Frecuencia medida en GHz

DIFICULTADES DE IMPLEMENTACIÓN

La implementación y pruebas de funcionamiento del proyecto no pudieron llevarse a cabo debido a dificultades de presupuesto y disponibilidad del equipo en el país. Se intentaron recrear las condiciones del sistema utilizando softwares de simulación como MatLab, Simulink y LabView sin embargo no se pudo hacer la simulación debido a que el software OpenBTS requiere tener conectado el USRP1 para su instalación y configuración.

Frente a esta situación se decidió realizar los cálculos necesarios para el planteamiento del desarrollo del sistema apegándonos a las normas establecidas y normalizadas por la UIT-R para que en una futura implementación se sepan los parámetros que deben tomarse en cuenta al momento de realizar las pruebas de funcionamiento.

CONCLUSIONES

Utilizando la virtualización de una BTS se contribuye a cerrar la brecha digital que tiene Republica Dominicana implementándose en las comunidades que poseen pocos habitantes y que los mismos no cuentan con comunicación móvil. Este proyecto es de rápida implementación por lo tanto en caso de algún desastre puede restablecerse rápido.

El computador se encarga de realizar todo el procesamiento de las llamadas, Asterisk se encarga del enrutamiento, control y conmutación de las llamadas mientras que el OpenBTS utiliza el USRP para generar la señal GSM y brindar la cobertura. Al utilizar la micro celda indoor se amplía la potencia de transmisión reduciendo la pérdida de señal.

Los modelos de propagación son métodos de cálculo de propagación que ayudan a entender cuáles son los alcances territoriales que tendrá la red y ayudan a realizar una proyección de cobertura.

RECOMENDACIONES

Este proyecto puede utilizarse para creación de celdas de emergencias en caso de desastres naturales, así la comunicación se pudiera restablecer de manera rápida. Fue diseñado para hacer llamadas, pero tiene las herramientas necesarias para poder también utilizar datos y mensajes SMS, que en caso de ser requeridos pueden configurarse y ser utilizados.

Se recomienda no utilizarse con fines fraudulentos, estos pueden ser penalizados por el Instituto Dominicano de las Telecomunicaciones (INDOTEL). Para la implementación se recomienda utilizar únicamente el USRP1 de Ettus Research, debido a que es el que está certificado y garantiza el completo funcionamiento del diseño.

Es necesario realizar el cálculo de los 3 modelos de propagación propuestos en el capítulo 3, así como también los demás cálculos mencionados para así se tiene una visión clara de la cobertura que brindará el sistema de comunicación y el presupuesto de potencia necesario para garantizar una buena calidad de servicio.

REFERENCIAS BIBLIOGRÁFICAS

- María Carmen España Boquera. (2003). Servicios avanzados de telecomunicación. Juan Bravo, España: Ediciones Díaz de Santos, S.A.
- Bernhard H. Walke, P. Seidenberg, M. P. Althof. (2003). UMTS: The Fundamentals. Schlembach Verlag, Alemania: Jhon Wiley et Sons, Ltd.
- Sumit Kasera, Nishit Naran. (2004). 3G Networks: Architecrure, protocols and Procedures. New Delhi: Tata McGraw-Hill.
- Rajneesh Agarwal, Bharat B Tiwari. (2005). Data Communication And Computer Networks. New Delhi: Vikas Publishing House.
- Christopher Bowick. (2008). RF Circuit Desing, Second Edition. Burlington, MA: Elsevier.
- Ajay R. Mishra. (2007). Advanced Cellular Network Planning and Optimization: 2G/2.5G/3G...Evolution to 4G. Chichester, Inglaterra: Jhon Wiley et Sons, Ltd
- Joseph Mitola. (1992). Software Radio: Survey, Critical Analysis and Future Directions. IEEE, 12, 13/23.
- María Dolores Cereceda Tobal. (Octubre 2011). Diseño y evaluación de un detector de energía para señales fm wireless microphone en radio cognitiva. Cartagena, Colombia: ETSIT, Escuela Técnica Superior de Ingeniería de Telecomunicación.
- Ettus Research. (2012). USRP motherboard datasheet. 10/7/2015, de ETTUS RESEARCH LLC Sitio web:
http://www.ettus.com/content/files/07495_Ettus_USRP1_DS_Flyer_HR.pdf
- Newton R. (Agosto 06, 2014). Asterisk Architecture. Junio 03, 2015, de Asterisk Sitio web: <https://wiki.asterisk.org/wiki/x/R4JJ>

- Julián D Vásquez, Iván F Santa, José V Restrepo. (2012). Prototipo de una Estación Celular Portátil para Atención de Emergencias. 29/5/2015, de Universidad Pontificia Bolivariana.
- Darío Flores. (2011). Manual de uso e instalación de OpenBTS (La liberalización de la telefonía celular). Los Condes, Chile: Consulting Latam Office.
- E.E. Gaona García, M.A. Ávila Angulo, E.G. Muskus Rincón. (2014). Aproximación de la calidad de voz y cobertura en una red GSM de emergencia. Ciencia e Ingeniería Neogranadina, 24 (2), pp. 23 – 36
- Wayne Tomasi. (2003). Sistemas de Comunicaciones Electrónicas. México: Pearson Education, Inc.
- Peter Smyth. (2004). Mobile and Wireless Communications: Key Technologies and Future Applications. Reino Unido: British Telecommunications, PLC
- Ángel Cardama Aznar, Lluís Jofre Roca, Juan Manuel Rius Casals, Jordi Romeu Robert, Sebastián Blanch Boris y Miguel Ferrando Bataller. (1998). Antenas. Barcelona: Edicions de la Universitat Politècnica de Catalunya.
- Michael Iedema. (2011). Welcome to OpenBTS. 2015, de Range Networks Sitio web: <https://wush.net/trac/rangepublic>
- Antonio Del Valle Díaz. (2010). Diseño, Integración y Optimización de Estaciones Base De Segunda Generación. Sevilla, España: Escuela Técnica Superior de Ingenieros.
- Otto Limann. (1989). Fundamentos de radio. España: MARCOMBO, S.A

Anexos