Prototipo de red GSM basada en SDR

SDR-based GSM network prototype

Carla Parra
Nuevas Comunicaciones Iberia, S.A., Barcelona, España
carla.parra@nucom.es
ORCID: 0000-0002-7974-471X

Martha Cecilia Paredes
Escuela Politécnica Nacional, Quito, Ecuador
cecilia.paredes@epn.edu.ec
ORCID: 0000-0001-5789-4568

Germán Arévalo
Universidad Politécnica Salesiana, Quito, Ecuador
garevalo@ups.edu.ec
ORCID: 0000-0001-7034-5774

Christian Tipantuña
Escuela Politécnica Nacional, Quito, Ecuador
christian.tipantuna@epn.edu.ec
ORCID: 0000-0002-8655-325X

doi: https://doi.org/10.36825/RITI.10.21.004

Recibido: Junio 06, 2022
Aceptado: Agosto 08, 2022

Resumen: Este artículo presenta la implementación de un prototipo de red GSM (*Global System for Mobile Communications*) basado en SDR (*Software Defined Radio*) utilizando el software OpenBTS basado en Linux y el hardware USRP (*Universal Software Radio Peripheral*). Se ha elegido GSM por ser uno de los estándares de mayor éxito e impacto en el ecosistema de la telefonía móvil y base para las siguientes generaciones. Al integrar GSM con SDR es posible implementar un prototipo de red GSM experimental, el cual puede llegar a convertirse en un banco de pruebas para futuras innovaciones y desarrollos en la dinámica SDR cuyo objetivo que es brindar infraestructuras móviles alternativas para poblaciones rurales en Ecuador y en el mundo que forman parte de la gran brecha digital.

Palabras clave: *GSM, SDR, OpenBTS, USRP.*

Abstract: This paper presents the implementation of a GSM (Global System for Mobile Communications) network prototype based on SDR (Software Defined Radio) using the Linux-based OpenBTS software and the USRP (Universal Software Radio Peripheral) hardware. GSM has been chosen because it is one of the most successful standards with the most significant impact on the mobile telephony ecosystem. It has been the key to the development of subsequent generations. By integrating GSM with SDR, it is possible to implement a prototype

of an experimental GSM network, which can become a testbed for future innovations and developments in SDR dynamics that seek to provide alternative mobile infrastructures to rural populations in Ecuador and around the word that they still are part of the great digital divide.

Keywords: *GSM, SDR, OpenBTS, USRP.*

1. Introducción

En este artículo se detalla la implementación de un prototipo de red GSM (*Global System for Mobile communications*) basada en SDR (*Software Defined Radio*), mediante el cual dos teléfonos móviles pueden realizar una llamada telefónica y enviar un mensaje de texto corto. Es precisamente, después del surgimiento de la tecnología SDR que se han desarrollado varios proyectos *open source* para telefonía orientados a la telefonía móvil, y uno en especial ha asumido un rol fundamental en el desarrollo de proyectos investigativos en el área de las redes móviles y este es OpenBTS [1].

¿Por qué GSM?, porque además de ser la tecnología móvil más exitosa que ha cubierto más del 90% de la población mundial, la robustez de su comunicación [2] y la experiencia previa con el estándar motivan la puesta en marcha de proyectos open source como OpenBTS.

La combinación de la interfaz aire GSM con la red de retorno VoIP (*Voice over* IP) podrían ser la base de un nuevo modelo de red celular de tipo comunitario de bajo costo. Dado que las nuevas redes híbridas no son fácilmente compatibles con las redes tradicionales y que económicamente no representarían rentabilidad alguna para los operadores móviles, es casi improbable ver este tipo de innovación implementado por los proveedores [3]. Sin embargo, la experiencia previa con GSM, el desarrollo de la radio definida por software y de sistemas de energía sostenibles, así como el soporte de VoIP y la gestión de políticas públicas en el marco de la universalidad del uso de las tecnologías de la información han puesto sobre la mesa proyectos como OpenBTS [3].

2. GSM (Global System for Mobile Communications)

2.1. GSM y su vinculación con la SDR

Aun siendo GSM la red móvil más grande y con mayor despliegue alrededor del mundo y que ha venido proporcionando un servicio de telecomunicaciones vital a miles de millones de personas, se conoce que gran parte de la población mundial aún viven fuera de su cobertura, mayormente en las zonas rurales [4]. Y pese a los impactos positivos y el desarrollo que ha traído consigo la telefonía celular, aún existen limitaciones para proporcionar un verdadero acceso universal al servicio de telefonía celular [4].

Las innovaciones tecnológicas recientes, en particular las basadas en la tecnología SDR de bajo costo y en software *open source*, han desafiado el status quo, y ahora, una comunidad rural ya puede tener su propia infraestructura de red celular de bajo costo, auto gestionable y auto sustentable [4].

Investigaciones recientes proponen un nuevo modelo de conectividad celular de tipo comunitario potencialmente factible para brindar cobertura a poblaciones rurales; pero aspectos como las políticas de acceso al espectro y su legalidad dificultan implementarlo, pese a que las áreas rurales suelen tener cantidades sustanciales de espectro con licencia que no se utiliza activamente [4].

Se ha definido al espectro no utilizado GSM como espacios en blanco GSM (GSM *whitespaces*), conocido también como espectro del usuario primario (PU, *Primary User*), el cual no se usa en una zona en particular, y por lo tanto, podría ser reutilizado por un operador secundario (SU, *Secondary User*) sin interferir con el titular de la licencia primaria. El objetivo subyacente es regular el espectro para lograr un intercambio dinámico del mismo [4].

Es así que se ha propuesto un esquema de intercambio de espectros híbridos, fundamentado en el concepto de NGSM (*Nomadic* GSM) que busca una coexistencia segura entre usuarios primarios y secundarios (los usuarios primarios usan el espectro con licencia mientras que los secundarios aprovechan dicho espectro cuando está libre) para minimizar conflictos [4]. Además, este esquema brinda a los reguladores la posibilidad de controlar el espectro que es utilizado por operadores secundarios [4].

Varios proyectos de infraestructuras de redes móviles alternativos ya son una realidad, redes de telefonía comunitarias en localidades de México (Oaxaca) [6], Zambia (Zheleva) [7] e Indonesia (Papua) [8] ya se

encuentran operando. La red celular de Oaxaca ya cuenta con licencia del espectro de tipo experimental a corto plazo, el resto de las redes funcionan sin licencias. Escenarios como estos motivan la implementación de proyectos como OpenBTS en plataformas hardware SDR. Para ello, es necesario proponer mecanismos para licenciar y regular efectivamente el acceso al espectro [4] en busca de fomentar la innovación tecnológica y fortalecer el derecho a un auténtico acceso universal del servicio de telefonía celular [6].

2.2. Arquitectura de GSM

Una red GSM se compone de tres subsistemas como se muestra en la Figura 1. El primero es el Subsistema de Red y Conmutación (*NSS, Network and Switching Subsystem*) cuyo elemento principal es el MSC (*Mobile Switching Center)* y contiene las bases de datos HLR (*Home Location Register*), VLR (*Visitor Location Register*) y AUC (*Authentication Unit Center*). El segundo es el Subsistema de Estación Base (BSS*, Base Station Subsystem*) formado por un Controlador de Sistema Básico (BSC, *Basic System Controller*), la Estación Transceptora Base (BTS, *Base Transceiver Station*) y la Estación Móvil (MS, *Movil Station*) [9]. Y el tercero es el subsistema de Operación y Mantenimiento (OMSS, *Operation and Maintenance Subsytem*) [9].

El subsistema NSS (llamado también núcleo de la red GSM) realiza funciones claves relacionadas con las llamadas extremo-extremo, manejo de la movilidad, gestión de los suscriptores, la conmutación y la comunicación con otras redes como la ISDN (*Integrated Services Digital Network)* y la PSTN (*Public Switched Telephone Network*). Además, el NSS efectúa acciones relacionadas con la autenticación y la validación de los equipos [10]. Las bases de datos de GSM son: HLR conocida como la base de datos master permanente y lleva un registro de los suscriptores y almacena el número de teléfono de cada usuario, suscripciones de servicio, permisos y datos de autenticación. La base de datos VLR almacena datos temporales y atiende a los suscriptores que visitan la red diariamente; además, es responsable de la localización y del almacenamiento de todos los datos de los usuarios que se encuentran en un área determinada [10]. La base de datos AuC es la encargada de la autenticación del suscriptor y de almacenar los datos y claves confidenciales utilizadas para la autenticación y el cifrado de la información. Y, por último, la base de datos EIR es la encargada del registro los datos del equipo para su verificación [10].

El MS es el equipo utilizado por el suscriptor para acceder a los servicios proporcionados por la red GSM; tradicionalmente, se consideraba parte del BSS. La función principal del MS es transmitir y recibir voz y datos a través de la interfaz aire [9]. Adicional, se consideran dos partes esenciales de una MS: el Módulo de Identidad del Suscriptor (SIM, *Susbcriber Identity Module*) y el Equipo Móvil (*Mobile Equipment*) [11].

Por último, el tercer subsistema es el OMSS, a través del OMC (*Operational and Maintenance Center*) el ingeniero de soporte monitorea, diagnostica y soluciona los problemas que se presenten en el sistema de gestión de la red de telecomunicaciones TMN (*Telecommunications Management Network*) [9].

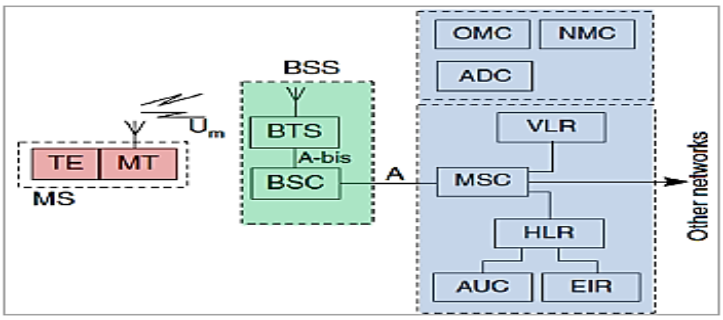


Figura 1. Arquitectura de la red GSM [10].

2.3. Principios y descripción del sistema de radio de GSM

El sistema GSM usa TDMA (*Time Division Multiple Access*) dentro de FDMA (*Frecuency Division Multiple Access*) para un acceso múltiple [12]. Una banda de frecuencia de 25 MHz está dividida en frecuencia portadoras utilizando un esquema FDMA. A cada estación base se asigna una o más portadoras dentro de una celda. Las frecuencias portadoras se encuentran separadas 200 KHz una de otra [12]. Normalmente, una banda de 25 MHz debe ser dividida en 125 frecuencias portadoras, pero en GSM, la primera frecuencia portadora se utiliza como una banda de protección entre GSM y otros servicios que podrían estar trabajando a frecuencias más bajas [12].

La banda de frecuencia se divide en 124 portadoras de subida/bajada. Cada una está dividida en 8 ranuras de tiempo (time slots) para permitir que al menos 7 usuarios puedan acceder a la red usando la misma portadora. Para el enlace de subida se ha asignado la banda de frecuencia desde los 890 a los 915 MHz y para el enlace de bajada desde los 935 a los 915 MHz.

2.4. SDR (Software Defined Radio)

Según el *Wireless Innovation Forum* se define a la SDR como: "radio en las que algunas o todas las funciones de la capa física se definen por software" [13]. En otras palabras, SDR es una tecnología donde módulos de software son ejecutados en tiempo real en plataformas genéricas de microprocesadores, procesadores digitales de señales o en circuitos lógicos programables. En la Figura 2 se muestra un esquema genérico de la arquitectura SDR que consta de tres bloques funcionales: sección RF (*Radio Frecuency*), sección de IF (*Intermediate Frecuency*) y la sección banda base.

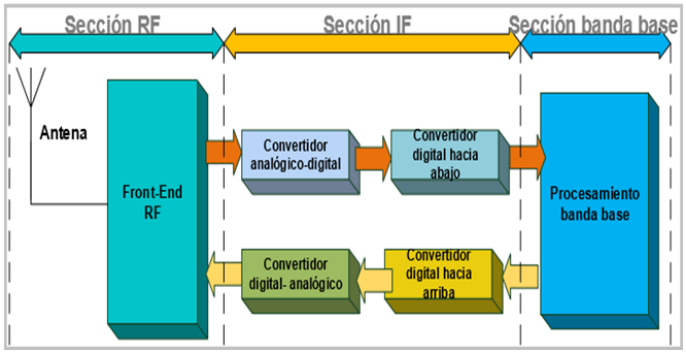


Figura 2. Arquitectura genérica de la SDR [14].

A continuación, detallaremos cada componente de las SDR:

1. La sección RF (RF *front-end*): es la encargada de transmitir o recibir la señal de radio frecuencia (RF) a través de la antena, acoplándola y realizando una conversión descendente a una señal IF (en la recepción). En transmisión se realiza una conversión ascendente para convertir la señal IF a RF, seguida de una etapa de amplificación [14].
2. La sección IF: esta sección tiene como función principal pasar la señal IF a banda base y digitalizarla en la recepción o convertir la señal banda base a IF y realizar la conversión digital-analógica en transmisión [14].
3. Esta sección contiene los bloques: ADC (*Analog-to-Digital Converter*) y DAC (*Digital-to-Analog Converter*) encargados de la conversión de analógico a digital en recepción y de digital a analógico en transmisión [14]. La sección IF también contiene el bloque DDC (*Digital Down Conversion*), el cual realiza la conversión digital descendente, es decir, baja la señal IF a banda base (en recepción) y el bloque DUC (*Digital Up Conversion*) que realiza una conversión digital ascendente, es decir, se encarga de que la señal IF suba a banda base (en transmisión) [14].
4. Sección de banda base: en esta sección se realiza operaciones como: configuración de la conexión, ecualización, salto de frecuencia (*frequency hopping*), cronometraje de recuperación (*timing recovery*), etc. [14].

2.5. USRP (Universal Software Radio Peripheral)

Con el vertiginoso avance de las SDR, varias plataformas hardware (propietarias y hardware *open source*) SDR se han desarrollado como, por ejemplo: Ettus USRP, Range Networks, RAD-1, Rice WARP, UmTRX, etc., las cuales se han integrado con proyectos software *open source* como OpenBTS para ofrecer una solución integral a los requerimientos de las SDR y han acelerado su innovación [15].

USRP es una plataforma orientada para SDR desarrollado por Ettus Research, un proveedor mundial de hardware dirigido a la implementación de SDR [16]. Permite la implementación de radios digitales, proporcionando la infraestructura de procesamiento digital y RF. Cuenta con dos niveles de tarjetas, en el primer

nivel se encuentra una tarjeta principal o placa madre (*motherboard*) que contiene: un FPGA (Altera Cyclone EP1C12), 4 convertidores analógicos a digitales (ADC) de alta velocidad, cada uno de 12 [bits/muestra] y 64 [millones de muestras/segundo] (MS/s); 4 convertidores digitales a analógicos (DAC) de alta velocidad, cada uno de 14 [bits/muestra] y 128 [millones de muestras/segundo]; un controlador programable (Cypress FX2 USB 2.0 o una interfaz Gigabit Ethernet) para la conexión con un host; y la fuente DC (*Direct Current*) de alimentación. En el segundo nivel se tiene las tarjetas secundarias o hijas (*daughterboards*), dos para transmisión y dos para recepción [17].

2.6. OpenBTS

OpenBTS es un proyecto basado en C++ de código abierto bajo la licencia AGPL (*Affero General Public License*) dedicado a revolucionar las redes móviles mediante la sustitución de protocolos tradicionales de telecomunicaciones y hardware propietario complejo por protocolos de internet y arquitecturas de software y hardware flexibles [1].

El proyecto OpenBTS simula una red GSM con base en el funcionamiento de los siguientes servicios centrales: Asterisk o conmutador VoIP, responsable de manejar las solicitudes del protocolo SIP (*Session Initiation Protocol*) SIP Invite para establecer las llamadas; SIP *Authorization* (*SIPAuthServe)*, aplicación que procesa las solicitudes de registro SIP *Register* de usuarios los SIP que OpenBTS genera cuando un teléfono intenta unirse a la red; SIP *Message Queue* (SMQueue), aplicación que procesa solicitudes SIP que OpenBTS genera cuando un teléfono intenta enviar un SMS (*Short Message Service*); y finalmente OpenBTS, que contiene la mayoría de la pila (*stack*) de protocolos de GSM y es el responsable de implementar la interfaz aire GSM en software [18].

La comunicación entre OpenBTS y los teléfonos móviles se efectúa a través de los protocolos SIP y RTP (*Real-time Transport Protocol*) sobre UDP (*User Datagram Protocol*) en el lado de la red IP (*Internet Protocol*), es así como OpenBTS y el hardware USRP forman la red central que se observa en la Figura 3 [18].

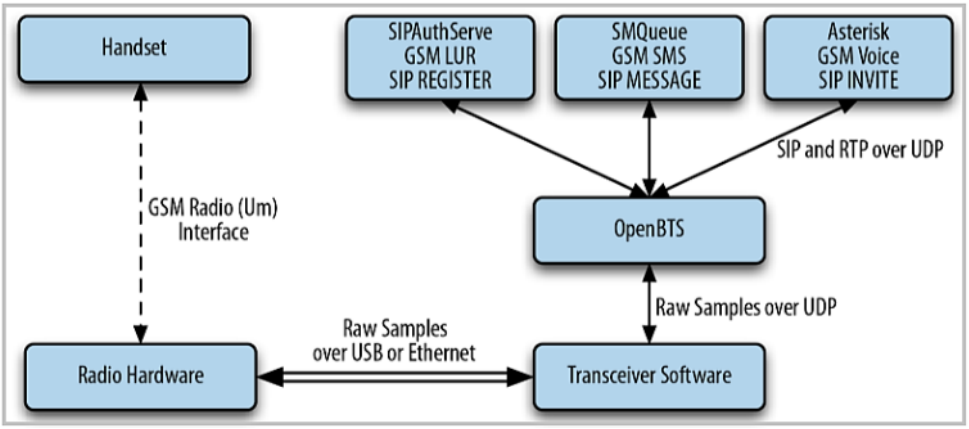


Figura 3. Arquitectura de OpenBTS [18].

3. Implementación

Antes de la implementación, en la Figura 4 se observa una comparación entre la red GSM convencional y una red GSM basada en OpenBTS. Si bien, se trata de una comparación subjetiva; no obstante, aporta una idea concisa de lo que representa una red GSM basada en SDR. En la parte superior de la Figura 4 se tiene los tres subsistemas de una red GSM tradicional y en la parte inferior los elementos hardware (URSP) y software (OpenBTS) de una red GSM experimental.

Figura 4. Comparación subjetiva entre la red GSM convencional y OpenBTS.

En la Figura 5 se detalla el proceso del establecimiento de una red GSM basada en OpenBTS para realizar una llamada telefónica y enviar un mensaje de texto. Primero se tiene un proceso de sincronización entre la BTS y la MS mediante los canales de GSM SCH (*Synchronization Channel*) y FCH (*Frecuency Channel*). Al encontrar un canal de servicio de la BTS, se procede con la búsqueda del ARFCN (*Absolute Radio Frecuency Channel Number*) disponible y otros parámetros como el MCC (*Mobile Country Code*) y MNC (*Mobile Network Code*). El siguiente paso es establecer los parámetros mínimos de ruido e interferencia del canal; solo si el RSSI (*Received Signal Strength Indicator*) es mayor que el nivel de ruido establecido y el FER (*Frame Error Rate*) < 15 % los terminales móviles podrán acceder a la red GSM, realizar una llamada telefónica y enviar un mensaje corto [5]; en el caso que el RSSI < ruido y el FER > 15 %, el proceso de búsqueda del mejor ARFCN se volverá a repetir, con la finalidad de ofrecer una buena calidad de la comunicación para el SU y de no interferir con el PU. A este último proceso con base en los valores del RSSI se lo conoce como acceso dinámico al espectro basado o radio cognitiva.

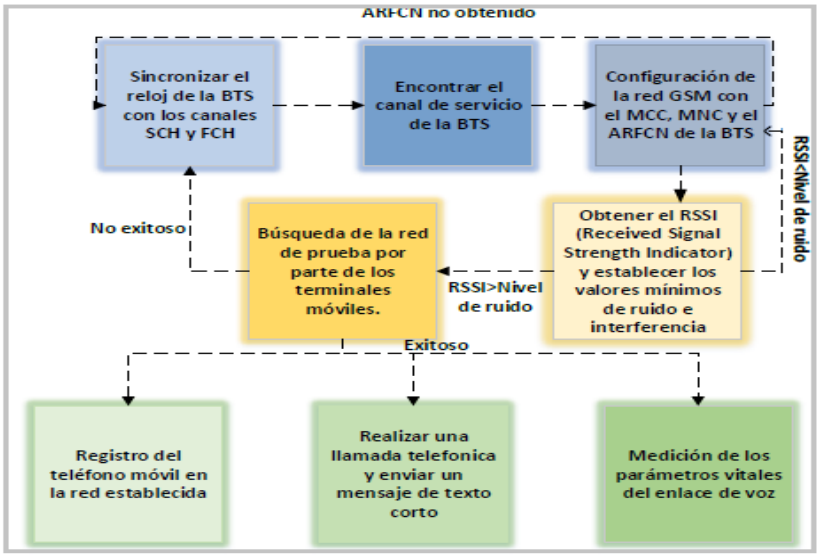


Figura 5. Pasos del establecimiento de una llamada en una red GSM definida por software [5].

A continuación, se describe el proceso de instalación de OpenBTS basado en las referencias [18] y [19].

Paso 1: Verificación del firmware del USRP NI-2920

Para que la plataforma hardware USRP pueda interactuar con el proyecto OpenBTS, es necesario que en el equipo SDR (USRP NI-2920) se encuentre instalado el firmware correspondiente. El firmware en los dispositivos USRP proporciona compatibilidad con el controlador de software de la plataforma hardware, y en algunos casos es posible que este componente deba ser instalado, actualizado o reemplazado. Cabe mencionar que la versión del firmware del dispositivo USRP NI-2920 es la 12.4 y no ha sido necesario actualizarla.

Paso 2: Instalación del sistema operativo Ubuntu
Para el presente proyecto se ha seleccionado la distribución Ubuntu 16.04.5 LTS (*Long-Term-Support*) Xenial Xerus Desktop de 64 bits. Para la instalación se ha asignado alrededor de 40 GB de almacenamiento para la partición raíz y se ha creado un usuario de nombre "openbts".

Paso 3: Instalación de prerrequisitos para OpenBTS
Para que la instalación de OpenBTS sea exitosa, es necesario que se instalen previamente ciertas librerías y dependencias, este proceso es realizado ejecutando el siguiente comando:

```
sudo apt-get install software-properties-common python-software-properties
```

Paso 4: Instalación de los repositorios Git
El proyecto OpenBTS utiliza varias características de Git como el seguimiento a través de submódulos (o ramas) para asegurarse de que su cliente (openbts) sea compatible con las versiones más recientes de OpenBTS. En este contexto, ejecutando los comandos mostrados a continuación se añade un repositorio para obtener las últimas versiones de Git para gestionar las actualizaciones del usuario openbts.

```
sudo add-apt-repository ppa:git-core/ppa
sudo apt-get update
sudo apt-get install git
```

Paso 5: Descarga del código fuente de OpenBTS
Instalado el sistema Git se procede a la descarga de los scripts de desarrollo. Para lo cual se debe clonar el repositorio dev desde Git Hub ejecutando los siguientes comandos:

```
cd ~
git clone https://github.com/RangeNetworks/dev.git
cd ~dev

./clone.sh
```

Una vez que los componentes del proyecto OpenBTS se encuentran en el entorno de desarrollo, es posible seleccionar una rama o versión específica para compilar. Se selecciona la rama *master* de OpenBTS.

```
./switchto.sh master
```

A continuación, se procede a instalar las dependencias (entre estas Asterisk 11.7 y los algoritmos de autenticación y cifrado de GSM) de OpenBTS

```
sudo dpkg -i *.deb
```

Para verificar la instalación de todos los paquetes necesarios se puede digitar el comando que se muestra a continuación:

```
sudo apt-get install -f
```

Paso 6: Instalación de OsmoTRX
OsmoTRX es un transceptor de radio definido por software que implementa la capa física de una BTS y su código se basa en el transceptor Transceiver52M del proyecto OpenBTS. Se ha utilizado este transceptor como alternativa para los transceptores propios de OpenBTS, los cuales han generado problemas al compilarlos. Para la instalación de OsmoTRX es necesario instalar la librería libosmocore. Previo a la instalación de libosmocore se instalarán las siguientes dependencias:

```
sudo apt-get install gnutls-bin
sudo apt-get install gnutls-dev
sudo apt-get install build-essential libtool libtalloc-dev shtool autoconf automake
git-core pkg-config make gcc
sudo apt-get install libpcsclite-dev
```

Ahora se procede a la instalación de libosmocore, clonando su repositorio con los siguientes comandos:

```
git clone git://git.osmocom.org/libosmocore.git
cd libosmocore/
autoreconf –i
./configure
make
sudo make install
sudo ldconfig –i
cd ..
```

Luego se procede a instalar OsmoTRX:

```
cd ~
git clone https://github.com/osmocom/osmo-trx.git
cd osmo-trx
./autogen.sh
./configure
make
sudo make install
```

Paso 7: Conexión de OpenBTS con el dispositivo USRP NI-2920

Una vez instalado OpenBTS, se procede a realizar las pruebas de conectividad, para ello se ha asignado a la interface Gigabit Ethernet la IP 192.168.10.1/24, la cual se encuentra dentro de la red 192.168.10.2/24 del USRP NI-2920. A continuación, se procede a inspeccionar que el dispositivo efectivamente esté conectado y su controlador se encuentre instalado. Para esta finalidad, Ettus Research proporciona los comandos uhd_find_devices y uhd_usrp_probe para detectar e inspeccionar automáticamente los equipos SDR conectados, respectivamente [18].

Paso 8: Alineación de las antenas

La alineación de las antenas tiene un rol importante al momento de transmitir y recibir la señal. Se ha utilizado un par de antenas VERT de 900 MHz de goma, tipo pato [18]. Se debe ubicar las antenas perpendicularmente con la finalidad que de no saturar de energía el enlace ascendente.

Paso 9: Inicialización de los servicios

Una vez realizadas las pruebas de conectividad y la alineación de las antenas, OpenBTS y sus servicios asociados deben ser inicializados en el siguiente orden:

Terminal 1: Se inicializa el servicio de mensajería a través del servicio smqueue con el siguiente comando:

```
sudo /usr/local/sbin/smqueue
```

Terminal 2: Se inicializa el servicio de registro y autenticación de suscriptores a través del servicio sipauthserve ejecutando el siguiente comando:

```
sudo /usr/local/sbin/sipauthserve
```

Terminal 3: Se inicializa el servicio de conmutación de llamadas a través del servicio Asterisk mediante el siguiente comando:

```
sudo /usr/sbin/asterisk -vvvv
```

Terminal 4: Antes de inicializar OpenBTS se debe iniciar el transceptor OsmoTRX con los siguientes comandos:

```
sudo apt-get install osmo-trx -f
sudo osmo-trx -f
```

Terminal 5: Finalmente se inicializa el servicio OpenBTS mediante los siguientes comandos:

```
cd /OpenBTS
sudo ./OpenBTS
```

Paso 10: Llamadas de prueba
Se ha considerado realizar llamadas de prueba para comprobar el funcionamiento del prototipo. Para ello se ha realizado las siguientes configuraciones en OpenBTS:

```
OpenBTS> rxgain 20
OpenBTS> config Control.LUR.SendTMSIs "1"
OpenBTS> config Control.LUR.OpenRegistration ".*"
```

El primer comando configura la ganancia del receptor; el segundo comando permite activar los intercambios LUR (*Location Update Request*) para el intercambio del IMSI por el TMSI de un terminal móvil; y el tercer comando da paso a un registro abierto para cualquier suscriptor.

Paso 11: Búsqueda de la red
Se ha considerado realizar una llamada de eco (marcando a la extensión 2600), la cual develará algún problema relacionado con la calidad del enlace descendente y posteriormente una de tono (marcando a la extensión 2602), la cual verificará que Asterisk esté funcionando correctamente. Para este propósito, resulta necesario realizar ciertas configuraciones en los terminales móviles, las cuales son detalladas a continuación:

1. Seleccionar “Ajustes”.
2. Seleccionar “Redes móviles”.
3. Seleccionar “Operadores de red”.
4. Seleccionar “Buscar Redes”.

Esta última acción permite iniciar una búsqueda de la red de prueba, este procedimiento puede tardar unos minutos hasta que la red móvil implementada se encuentre en la lista de los operadores móviles disponibles (si el ARFCN disponible cumple con las características establecidas). Si la red no ha sido detectada se puede forzar la búsqueda seleccionando nuevamente las opciones mostradas anteriormente o simplemente reiniciando el terminal móvil.

Paso 12: Personalización básica de la red
La personalización de la red es una de las características que hace de este prototipo un sistema funcional, reconfigurable y flexible. En esta sección se muestra como personalizar la red GSM en función de los requerimientos de los suscriptores. Para cambiar el nombre de la red por defecto a “EPNGSM”, se utiliza el siguiente comando:

```
OpenBTS> config GSM.Identity.ShortName EPNGSM
```

Para personalizar el mensaje de bienvenida se debe ejecutar el siguiente comando:

```
OpenBTS>config Control.LUR.NormalRegistration.Message Bienvenido a la red EPNGSM
```

Para personalizar el mensaje de registro fallido se ejecuta el comando que se indica a continuación:

```
OpenBTS>config Control.LUR.FailedRegistration.Message Su dispositivo no se encuentra registrado en la red.
```

Existen varias opciones disponibles para la configuración de la red GSM basada en SDR, desde ajustes de potencia, ganancia y ruido, configuraciones de mensajes de registro, hasta configuraciones más complejas como ajustes en timers, configuraciones de alarmas, entre otros.

Paso 13: Plan de marcado

El plan de marcado (dialplan) no es más que un conjunto de acciones ordenadas que ejecuta Asterisk cuando se marca un número en particular. Para empezar con el plan de marcado se debe asignar a cada IMSI (es decir, a cada terminal móvil) un MSISDN o un número telefónico. Al IMSI 740000117650736 se ha asignado el número telefónico 0521002000 y al IMSI 740020171054488 se ha asignado el número telefónico 0521003000. La configuración del plan de marcado se detalla a continuación con base en la referencia [20]. En un nuevo terminal, se procede a asociar a cada IMSI su respectivo número telefónico, para lo cual es necesario la ejecución de los siguientes comandos:

```
sudo asterisk -rx "database put IMSI IMSI740000117650736 0521002000"
sudo asterisk -rx "database put PHONENUMBER 0521002000 IMSI740000117650736"
sudo asterisk -rx "database put IMSI IMSI740020171054488 0521003000"
sudo asterisk -rx "database put PHONENUMBER 0521003000 IMSI740020171054488"
```

Para asegurarse que los números telefónicos y sus respectivos IMSI se hayan guardado exitosamente se ejecuta el siguiente comando:

```
asterisk –rx “database show”
```

El resultado del comando anterior se muestra a continuación:

```
/IMSI/IMSI740000117650736: 0521002000
/IMSI/IMSI740020171054488 : 0521003000
/PHONENUMBER/0521002000 : IMSI740000117650736
/PHONENUMBER/0521003000 : IMSI740020171054488
```

Para continuar con el plan de marcado se realiza una copia de seguridad del archivo /etc/asterisk/extensions.conf con el siguiente comando:

```
mv /etc/asterisk/extensions.conf /etc/asterisk/extensions.conf.original
```

Ahora, se procede a crear un nuevo archivo extensions.conf ejecutando el siguiente comando:

```
vi /etc/asterisk/extensions.conf
```

Luego, se debe agregar la siguiente línea al archivo /etc/asterisk/extensions.conf:

```
#include extensions-custom.conf
```

A continuación, se crea el archivo extensions-custom.conf, a través del siguiente comando:

```
vi /etc/asterisk/extensions-custom.conf
```

Posteriormente, se debe agregar el plan de marcado al archivo /etc/asterisk/extensions-custom.conf, el cual permitirá la comunicación entre los dispositivos móviles:

```
[from-openBTS]
exten => _0521X.,1,Verbose(Dialplan started)
same = n,Set(CALLER_IMSI=${CALLERID(num)})
same = n,Verbose(Get CID from CALLER_IMSI: ${CALLER_IMSI})
same = n,Set(CID=${DB(IMSI/${CALLER_IMSI})})
same = n,Set(CALLERID(num)=${CID})
same = n,Verbose(Get IMSI from EXTEN: ${EXTEN})
same = n,Set(IMSI=${DB(PHONENUMBER/${EXTEN})})
same = n,Dial(SIP/00101100010/${IMSI})
same = n,Hangup
```

Finalmente, se habilita el plan de marcado mediante el comando que se muestra a continuación:

```
sudo asterisk -rx "dialplan reload"
```

Paso 14: Registro de usuarios

Para añadir los usuarios a la base de datos sipauthserve.db, se ejecutan los siguientes comandos:

```
cd dev/NodeManager
./nmcli.py sipauthserve subscribers create “Nombre” IMSI«IMSI» «MSISDN»
```

En tanto, los siguientes comandos permiten el registro de los suscriptores con los que se han realizado las pruebas:

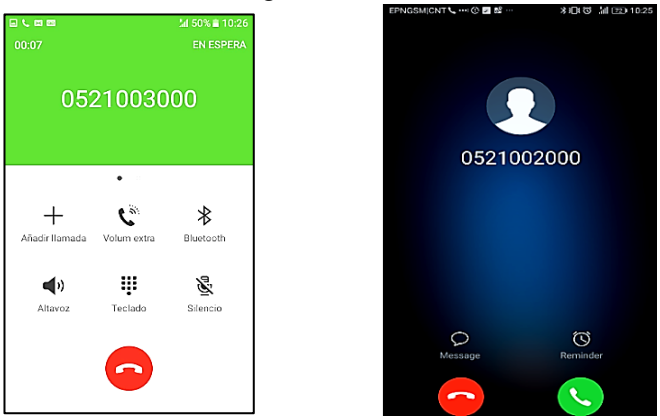
```
./nmcli.py sipauthserve subscribers create Carly IMSI740000117650736 0521002000
./nmcli.py sipauthserve subscribers create Manuela IMSI740020171054488 0521003000
```

Para verificar si un suscriptor ha sido añadido con éxito, se puede ejecutar el siguiente comando:

```
./nmcli.py sipauthserve subscribers read
```

Paso 15: Llamada telefónica entre dos terminales móviles

Para iniciar la llamada, como primer paso se debe buscar la red GSM implementada. Cabe mencionar que, si los dispositivos móviles estuviesen bloqueados por el operador de red autorizado, estos no lograrán conectarse a la red GSM. Enseguida se procede a realizar las llamadas telefónicas entre los dos abonados registrados, marcando sus MSISDN asignados. Como se observa en la Figura 6, las llamadas telefónicas se han efectuado con éxito.



a) Terminal que origina la llamada b) Terminal que recibe la llamada

Figura 6. Llamada telefónica entre dos abonados.

En la Figura 7 se observa la señalización entre OpenBTS y un terminal móvil durante una llamada telefónica. Primero el terminal solicita un canal a OpenBTS, a continuación, el canal es asignado y la llamada es establecida, en este punto OpenBTS envía una solicitud SIP INVITE a Asterisk y este responde con tres status: 100 *Trying*, 182 *Ringing* y 200 OK. A continuación, OpenBTS envía al terminal móvil un "*Alerting*", y a continuación, un "*Connect*"; el terminal móvil responde a OpenBTS con un "Connect ACK". Después de este proceso de señalización, finalmente se efectúa la llamada telefónica entre los terminales móviles utilizando el protocolo de transporte RTP (*Real Time Transport Protocol*) [18].

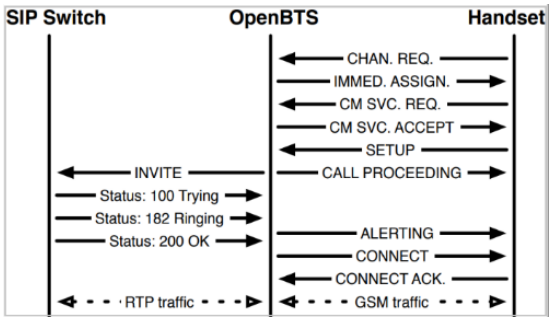
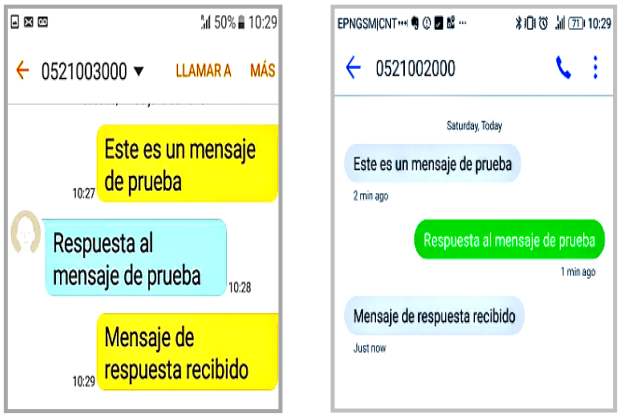


Figura 7. Señalización entre OpenBTS y el terminal móvil [18].

Paso 16: Envió de un mensaje de texto corto entre dos suscriptores

En esta sección se verifica el funcionamiento del servicio de mensajería del prototipo de red GSM, para lo cual se realizará el intercambio de un mensaje de texto corto entre los terminales móviles. En la Figura 8 se observa que los mensajes de textos han sido enviados y recibidos con éxito.



a) Terminal que envía un SMS b) Terminal que responde al SMS

Figura 8. Envío de un SMS entre dos terminales móviles.

4. Pruebas de validación de prototipo

En esta sección se presenta un conjunto de pruebas para la validación del prototipo.

4.1. Alcance de las llamadas telefónicas en un ambiente outdoor

Para realizar la valoración del alcance de las llamadas telefónicas entre los dispositivos móviles en un ambiente outdoor, se ha tomado como escenario un espacio amplio y libre de interferencias. Se han realizado 5 mediciones de la potencia de la señal recibida por los terminales cada 10 [m], esta medición de potencia se ha realizado con la herramienta Cell Info de Play Store Network. Se ha determinado un alcance máximo de 100 [m] en donde el prototipo muestra un desempeño óptimo. Cabe mencionar que la comunicación no se interrumpió aún alrededor de los 200 [m]; no obstante, la calidad de la misma disminuyó notablemente. En la Figura 9 se observa el resultado de las mediciones realizadas.

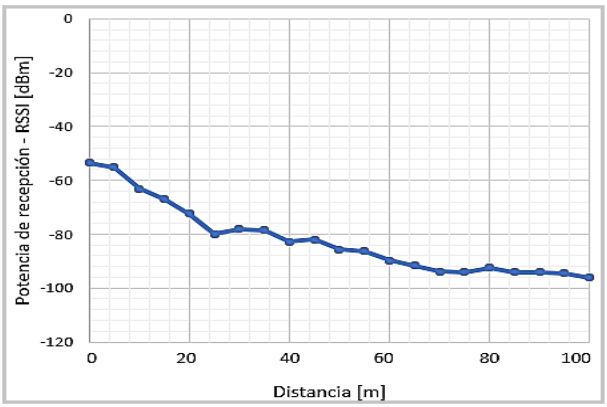


Figura 9. Alcance de las llamadas en un ambiente *outdoor*.

4.2. Alcance de las llamadas telefónicas en un ambiente indoor

Para verificar el desempeño de la red GSM en un ambiente *indoor* se ha considerado como escenario las instalaciones de los Laboratorios de Comunicaciones Inalámbricas. Se han efectuado 5 mediciones cada 2 [m]. Como se observa en la Figura 10, con base en las pruebas se ha determinado que el máximo alcance que tiene el prototipo en este escenario es de aproximadamente 38 [m] en condiciones óptimas.

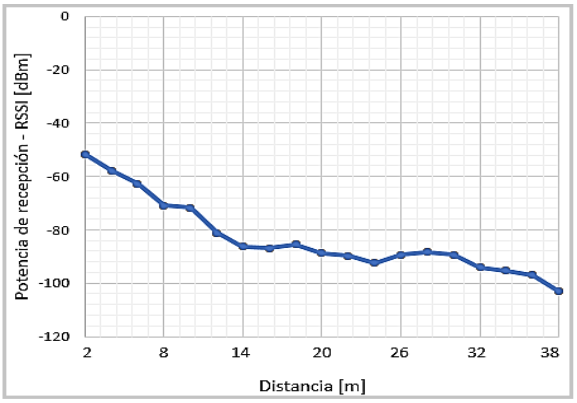


Figura 10. Alcance de las llamadas en un ambiente *indoor*.

4.3. Análisis de la estabilidad de una llamada telefónica en un ambiente indoor

Un indicador del desempeño del prototipo es la estabilidad con la que se efectúa una llamada telefónica, es decir, el tiempo que puede durar una llamada sin que esta finalice por causa de fallas en el sistema. Como se observa en la Fig. 11, la estabilidad del sistema fue verificada al establecer una llamada telefónica entre los dispositivos móviles de 1 hora de duración. Tiempo en el cual la calidad del servicio de voz no presento ruido, interferencia, distorsión o degradación de ningún tipo.

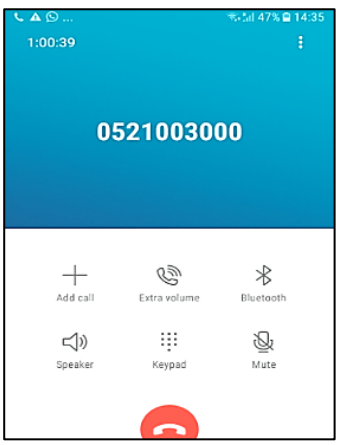
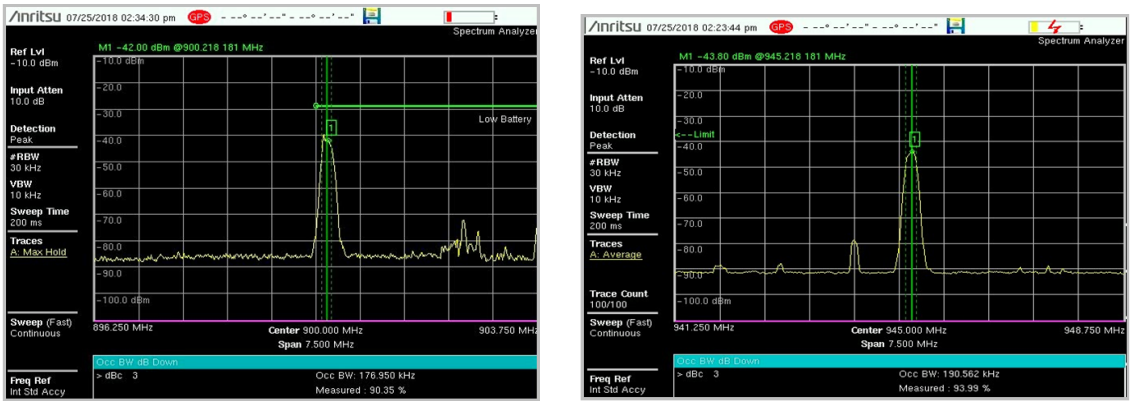


Figura 11. Duración de una llamada en un escenario *indoor*.

4.4. Análisis espectral de enlace ascendente y descendente

En primera instancia, se ha procedido a analizar el espectro del enlace ascendente (*uplink*), cuyo rango va desde los 890 MHz hasta los 915 MHz. En la Figura 12a se observa el espectro de un canal en el enlace ascendente cuyo ancho de banda de portadora es de 176.95 KHz. Este valor responde a las limitaciones inherentes del hardware del prototipo; sin embargo, cabe indicar que, durante la realización de las pruebas de establecimiento de llamadas, el ancho de banda medido del enlace ascendente presentó una oscilación en torno al valor teórico de un canal GSM, 200 KHz. En cuanto al análisis del espectro del enlace descendente (*downlink*) cuyo rango va desde los 935 MHz a los 960 MHz, en la Figura 12b se observa que el ancho de banda de portadora medido es de 190.562 KHz, cercano al valor de los 200 KHz definido en el estándar GSM. Al igual que en el caso anterior, la diferencia con el valor teórico estandarizado se atribuye a las limitaciones y características propias del equipo.



a) Espectro enlace ascendente b) Espectro enlace ascendente

Figura 12. Espectro de los enlaces ascendente y descendente de un canal GSM.

4.5. Tiempo promedio del envió de un mensaje de texto

Con base a las mediciones realizadas se determinó que el tiempo promedio de envío-recepción de un SMS entre dos dispositivos móviles en un ambiente *indoor* es de 6.7 [s], como se observa en la Figura 13. Para estas pruebas se ha considerado el alcance máximo en un ambiente *indoor* y se han realizado 5 mediciones cada 2 [m].

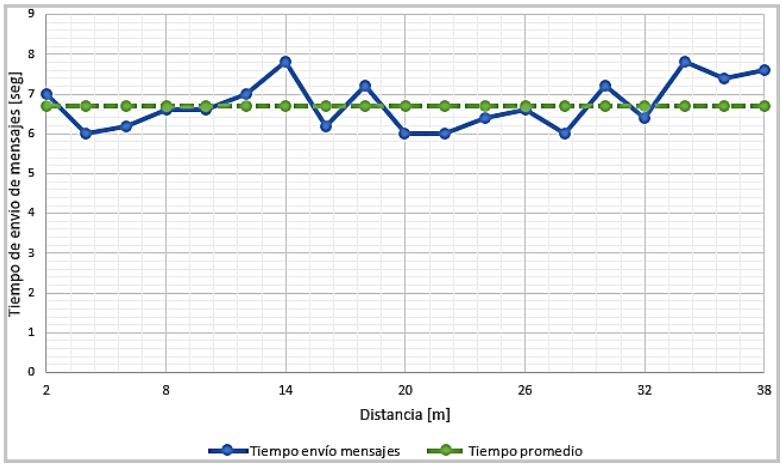


Figura 13. Tiempo promedio de envió-recepción de un SMS.

4.6. Evaluación del consumo computacional de los recursos del sistema

El consumo de recursos cuando el sistema está activo y aún no se ha realizado ninguna llamada es del 43.3% en CPU y de 0.7% en memoria. OpenBTS, Asterisk y Osmo-TRX consumen alrededor del 44% de recursos del sistema. Se puede concluir que el incremento en recursos computacionales de CPU desde que el sistema se encuentra activo hasta cuando se efectúa dos llamadas simultáneas es del 23.5%, en tanto que el consumo de memoria se mantiene constante.

5. Conclusiones

Se implementó un prototipo de red GSM, utilizando la plataforma de SDR USRP NI-2920 y el proyecto de código libre OpenBTS. El sistema permite la realización de llamadas de voz y el envío de mensajes cortos entre los terminales móviles. Además, la versatilidad del sistema brinda la posibilidad de tener una red personalizable en un entorno real de telefonía móvil.

El sistema prototipo presenta un servicio de telefonía celular de buena calidad tanto en el servicio de voz como en el tiempo de envío de los mensajes de texto. Así lo demuestran las pruebas realizadas de alcance, de estabilidad de las llamadas telefónicas y de evaluación del tiempo promedio de envío de mensajes. La solución implementada permite el despliegue de una red GSM totalmente funcional, flexible, reprogramable y reconfigurable, la cual trabaja acorde a las directrices y protocolos especificados en el funcionamiento del estándar GSM. El prototipo constituye un ecosistema abierto por dos motivos principalmente, el primero, permite el libre acceso al sistema de cualquier dispositivo que tenga una tarjeta SIM (siempre y cuando no esté bloqueado por el operador móvil autorizado); y segundo, el prototipo puede operar sin licencias, pese a utilizar una banda licenciada, debido a que es un sistema de bajo consumo de energía (alrededor de 1 [W]) y principalmente, porque su funcionamiento se basa en el criterio del acceso dinámico al espectro. Por sus pequeñas dimensiones, peso reducido y consumo de energía, el sistema de red es completamente portable y puede ser desplegada y transportada sin necesidad de utilizar equipamiento especializado para su transporte e instalación. Es importante mencionar que, para testear la robustez del sistema, se realizaron dos llamadas telefónicas simultáneas (es decir cuatro dispositivos móviles simultáneos conectados a la red GSM - SDR), las cuales se efectuaron sin presentar interferencias, retardos o cualquier otra característica que pudiese afectar la calidad de la llamada.

En el presente trabajo se ha implementado el servicio de voz y mensajería GSM, una de las futuras aplicaciones de esta investigación es el desarrollo de un prototipo USRP, UMTS, LTE o 5G en el que se pueda acceder a los datos móviles. El desarrollo de este prototipo busca promover a mediano o largo plazo el desarrollo de infraestructuras de telecomunicaciones alternativas, *open-source* y/o híbridas de bajo costo y auto gestionables para y por las propias comunidades rurales que aún son parte de la brecha digital en Ecuador y en otras regiones del mundo. Para lograrlo, en primera instancia, se deberá socializar iniciativas como estas cuyo objetivo sea promover un marco legal comunitario que fortalezca la gestión universal del acceso al espectro radioeléctrico para que las redes de telecomunicaciones comunitarias puedan operar bajo dos principios fundamentales: la legalidad y la universidad del acceso a las tecnologías de la información, eje promovido por la Naciones Unidas y la UNESCO; por lo cual, en este contexto es mandatorio crear estrategias económicas, políticas y sociales que respalden la participación activa de sus principales actores, las comunidades rurales del Ecuador y del mundo entero.

6. Referencias

[1] OpenBTS. (2017). *Open Source Cellular Infrastructure*. http://openbts.org

[2] Pace, P., Loscri, V. (2012). *OpenBTS: A step forward in the cognitive direction*. 21st International Conference on Computer Communications and Networks ICCCN, Munich, Germany. https://doi.org/10.1109/ICCCN.2012.6289232

[3] Burges, D. A., Samra, H. S. (2008). *The OpenBTS Project*. https://docs.huihoo.com/openbts/OpenBTS-Project.pdf

[4] Hasan, S., Heimerl, K., Harrison, K., Ali, K., Roberts, S., Sahai, A., Brewer, E. (2014). *GSM Whitespaces: An Opportunity for Rural Cellular Service*. IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN), McLean, VA, USA. https://doi.org/10.1109/DySPAN.2014.6817804

[5] Yuva Kumar, S., Saitwal, M. S., Ali Khan, M. Z., Desai, U. B. (2014). *Cognitive GSM OpenBTS*. IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems, Philadelphia, PA, USA. https://doi.org/10.1109/MASS.2014.68

[6] Rhizomatica. (2018). *Community Telecommunications Infrastructure*. https://www.rhizomatica.org/

[7] Zheleva, M., Paul, A., Johnson, D. L., Belding, E. (2013). *Kwiizya: Local Cellular Network Services in Remote Areas*. 11th annual international conference on Mobile systems, applications, and services, Taipei, Taiwan. https://doi.org/10.1145/2462456.2464458

[8] Heimerl, K., Hasan, S., Ali, K., Brewer, E., Parikh, T. (2013). *Local, sustainable, small-scale cellular networks*. Sixth International Conference on Information and Communication Technologies and Development, Cape Town South Africa. https://doi.org/10.1145/2516604.2516616

[9] Gu, G., Peng, G. (2010). *The survey of GSM wireless comunication system*. International Conference on Computer and Information Application, Tianjin, China. https://doi.org/10.1109/ICCIA.2010.6141552

[10] Eberspächer, J., Vögel, H. J., Bettstetter, C., Hartmann, C. (2008). *GSM - Architecture, Protocols and Services* (3rd. Ed.). John Wiley & Sons Ltd.

[11] Narang, N., Kasera, S. (2007). *2G mobile networks: GSM and HSCSD*. McGraw-Hill.

[12] Hamzah, S. A., Ibrahim, S. A., Zainal, M. S., Ismail, M. (2005). *Analysis and receiving of downlink GSM signal using spectrum analyzer.* Asia-Pacific Conference Applications Electromagnetics, Johor, Malaysia. https://doi.org/10.1109/APACE.2005.1607840

[13] Wireless Innovation Forum. (2017). *What is Software Defined Radio?* http://www.wirelessinnovation.org/Introduction_to_SDR

[14] Wipro Technologies. (2002). *Software-Defined Radio*. http://www.ab4oj.com/dl/sdr_wipro.pdf

[15] Wireless Innovation Forum. (2017). *Software Defined Radio – Defined.* http://www.wirelessinnovation.org/Introduction_to_SDR

[16] Garcia Reis, A. L., Barros, A. F., Gusso Lenzi, K., Pedroso Meloni, L. G., Barbin, S. E. (2012). Introduction to the software-defined radio approach. *IEEE Latin America Transactions, 10* (1), 1156–1161. https://doi.org/10.1109/TLA.2012.6142453

[17] Ettus, M. (2024). *USRP User's and Developer's Guide.* https://www.olifantasia.eu/gnuradio/usrp/files/usrp_guide.pdf

[18] Ledema, M. (2015). *Getting Started with OpenBTS*. O´Reilly.

[19] Usama, M. (2017). *Setting up OpenBTS on National Instruments USRP 2922.* https://github.com/usamamuneeb/tech-guides/wiki/Setting-up-OpenBTS-on-National-Instruments-USRP-2922

[20] Raharja, A. (2016). *Instalación y configuración OpenBTS 5.0.* https://antonraharja.com/2016/03/16/instalasi-dan-konfigurasi-openbts-5-0/