

**ANÁLISIS DE REQUERIMIENTOS PARA EL DISEÑO DE UN LABORATORIO
REMOTO DE RADIO DEFINIDO POR SOFTWARE (SDR)**

DIEGO ANDRÉS SARMIENTO CAMPOS

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES
BUCARAMANGA
2017**

**ANÁLISIS DE REQUERIMIENTOS PARA EL DISEÑO DE UN LABORATORIO
REMOTO DE RADIO DEFINIDO POR SOFTWARE (SDR)**

DIEGO ANDRÉS SARMIENTO CAMPOS

**Trabajo Presentado en Cumplimiento de los Requisitos para obtener el
Grado de Ingeniero Electrónico**

**Director:
ÓSCAR MAURICIO REYES TORRES,
Doctor en Ingeniería Electrónica**

**Codirector:
DAVID EUGENIO FORERO MARTÍNEZ,
Magister en Ciencias de la Información y las comunicaciones**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES
BUCARAMANGA
2017**

DEDICATORIA

*A Dios por darme la vida, a mi abuelita por cuidarme desde el cielo y a mis padres
por ser siempre fuerza, guía y motivo.*

Diego Andrés Sarmiento Campos.

AGRADECIMIENTOS

A Dios, porque sin él nada de esto hubiera sido posible.

*A mi padre, Joselin Sarmiento Rojas por su apoyo y amor Incondicional y nunca
dejar de creer en mí.*

*A mi madre, Sara Campos García por su infinito amor y acompañarme y apoyarme
en todo este camino.*

*A mi hermano, Jose Luis Sarmiento Campos por ser mi eterno compañero de vida
y siempre estar presente en ella.*

*A mi hermana, Kylie Dajhane Sarmiento Campos por ser el motor que me dio
motivos y fuerzas para continuar.*

*A mi abuelita, Peregrina García de Campos porque desde el cielo siempre me
cuidó y en vida siempre me consintió y alcahueteó.*

*A mi novia, Yeni Paola Trujillo Lambert por su amor, constancia y compañía en
todos los momentos que compartimos juntos.*

*A mis amigos, Mario José Castañeda Valderrama y Juan Felipe Remolina García
por ser cómplices y partícipes de cuanta ocurrencia existiera por tantos años.*

*A Darwin Díaz Arenas, Julio Armando Bustamante Miranda, Helbert Fabian León
Dávila, Juan Carlos Gutiérrez Julio, Juan Diego Cortés Echávez y Bayron David*

Torres Cortecero, por todos los momentos que vivimos juntos en la convivencia del hogar y el apoyo incesante y mutuo en las situaciones difíciles.

A la familia Trujillo Lambert, por acogerme en el seno de su hogar como un miembro más y hacerme sentir de esa manera todo el tiempo.

A mis compañeros y amigos de colegio e infancia, por compartir conmigo tanto tiempo y tantas cosas en los mejores años de mi vida.

A mi familia, por darme su apoyo incondicional y guiarme en momentos de oscuridad.

Al profesor, Francisco José Martínez Pérez por su amistad y confianza en todo el tiempo que trabajamos juntos.

Al profesor, Óscar Mauricio Reyes Torres por su constancia y entrega a la labor pedagógica y por las enseñanzas brindadas.

Diego Andrés Sarmiento Campos.

CONTENIDO

	Pág.
INTRODUCCIÓN	16
1. ¿QUÉ ES UN LABORATORIO REMOTO BASADO EN SDR?.....	18
1.1. LABORATORIO REMOTO	18
1.2. RADIO DEFINIDO POR SOFTWARE (SDR).....	20
2. ANTECEDENTES	23
2.1. REFERENTES INTERNACIONALES	23
2.2. REFERENTES NACIONALES.....	27
2.3. REFERENTES INSTITUCIONALES	29
3. MODELO DE LABORATORIO REMOTO BASADO EN SDR.....	30
3.1. GENI LAB	30
3.2. ORBIT LAB	33
4. BANCO DE PRUEBAS REALIZADO EN LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.....	44
4.1 ACCESO REMOTO A DISPOSITIVOS USRP BAJO LA MISMA LAN	44
4.2 ACCESO REMOTO A DISPOSITIVOS USRP MEDIANTE INTERNET	54
5. RESULTADOS DE LAS PRUEBAS REALIZADAS EN LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.....	57
5.1 USRP X310.....	57
5.2 USRP1	58
6. LABORATORIO REMOTO DE COMUNICACIONES BASADO EN SDR EN LA UNIVERSIDAD INDUSTRIAL DE SANTANDER	60

6.1 INVENTARIO ACTUAL DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.....	60
6.2 CRECIMIENTO SUGERIDO PARA LA IMPLEMENTACIÓN DE UN LABORATORIO REMOTO BASADO EN SDR EN LA UIS.....	62
6.3 LABORATORIO REMOTO DE COMUNICACIONES DE SDR SUGERIDO PARA LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.....	65
7. CONCLUSIONES.....	67
8. TRABAJO FUTURO.....	69
BIBLIOGRAFIA.....	71
ANEXOS.....	77

LISTA DE FIGURAS

	Pág.
Figura 1. Descripción del funcionamiento de un laboratorio remoto	19
Figura 2. Radio Definido por Software ideal	21
Figura 3. Arquitectura de red de GENI Lab	33
Figura 4. Distribución de Hardware de Orbit Lab	34
Figura 5. Hardware de un <i>mini-rack (Massive-MIMO)</i>	35
Figura 6. Nodo de radio de Orbit Lab	36
Figura 7. Interacción, usuario-Orbit Lab	43
Figura 8. Esquema de conexión remota al USRP X310 en la misma subred	44
Figura 9. Especificaciones del USRP X310	46
Figura 10. Acceso al USRP X310 de manera remota en GNU-Radio	46
Figura. 11 Esquema de conexión remota al USRP X310 en la misma LAN	47
Figura 12. Esquema de conexión remota al USRP N210 en la misma LAN	49
Figura 13. Esquema de conexión remota al USRP1 en la misma LAN (usuario - software, USRP1)	50
Figura 14. Esquema de conexión remota al USRP1 en la misma LAN (usuario, USRP1- software)	50
Figura 15. Respuesta al comando “./usbsrv -l” en la terminal de Ubuntu del servidor	51
Figura 16. Respuesta al comando “./usbclnt -l” en la terminal de Ubuntu del cliente	52
Figura 17. Esquema de conexión remota al USRP1 en la misma LAN (usuario, software – USRP1)	53
Figura 18. Esquema de conexión remota al USRP X310 mediante internet.....	55
Figura 19. Esquema de conexión remota al USRP1 mediante internet (usuario - software, USRP1)	55

Figura. 20 Esquema de conexión remota al USRP1 mediante internet (usuario, USRP1 - software)	56
Figura. 21 Esquema de conexión remota al USRP1 mediante internet (usuario, software - USRP1)	56
Figura 22. Diagrama de bloques para el USRP X310.....	58
Figura 23. Diagrama de bloques para el USRP1	59
Figura. 24 <i>Switch</i> de antenas implementado por Orbit Lab [40]	64
Figura. 25 Propuesta de laboratorio remoto de comunicaciones de SDR en la UIS.	66

LISTA DE TABLAS

	Pág.
Tabla 1. Clasificación de los laboratorios según acceso y naturaleza del recurso.	19
Tabla 2. Algunos laboratorios basados en SDR a nivel mundial	24
Tabla 3. Características de los nodos de radio de primera generación	36
Tabla 4. Características de los nodos de radio de tercera generación tipo LV-67C	37
Tabla 5. Características de los nodos de radio de tercera generación tipo LV-67G	37
Tabla 6. Características de los nodos de radio de tercera generación tipo LV-67J	38
Tabla 7. Características de los nodos de radio de tercera generación tipo LV-67B	39
Tabla 8. Características de los nodos de radio de tercera generación tipo LV-67F	39
Tabla 9. Sandboxes, tipo de nodo y especialización	40
Tabla 10. Ubicación en la rejilla principal de los dispositivos de SDR	41

LISTA DE ANEXOS

	Pág.
ANEXO A – Instalación de UHD y GNU-Radio en cualquiera de sus versiones.....	78

Resumen

TÍTULO: ANÁLISIS DE REQUERIMIENTOS PARA EL DISEÑO DE UN LABORATORIO REMOTO DE RADIO DEFINIDO POR SOFTWARE (SDR) *

AUTORES: DIEGO ANDRÉS SARMIENTO CAMPOS **..

Palabras Clave: Laboratorio, remoto, testbed, Software Defined Radio (SDR).

Descripción:

En la primera parte de este trabajo se presenta una recopilación de información acerca de laboratorios remotos basados en sistemas de Radio Definido por Software (SDR – *Software Defined Radio*) que existen actualmente a nivel mundial. Esta recopilación se enfoca en las especificaciones y en la distribución del hardware en las instalaciones de cada uno de ellos. Además, se reseña el software utilizado, los servicios brindados a los usuarios y el manejo de la demanda de los recursos ofrecidos. Con base en lo anterior, se elige un modelo de referencia que permita definir los requerimientos de esta clase de laboratorio en un entorno como el de la Universidad Industrial de Santander, teniendo como criterios de selección: la cantidad, calidad y disponibilidad de información y la viabilidad de establecer una interacción usuario-laboratorio para evidenciar el funcionamiento de un laboratorio de esta naturaleza.

Finalmente, se elabora un escenario básico de pruebas (*testbed*) haciendo uso de los recursos físicos de hardware y software disponibles en la Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones (E3T), con el fin de validar el análisis de requerimientos realizado. Las pruebas ejecutadas permiten elaborar un concepto de viabilidad y establecer los alcances a los que puede apuntar la E3T en el diseño de un laboratorio remoto basado en SDR aprovechando los recursos disponibles actualmente y determinando prioritariamente cuáles recursos adicionales son necesarios para una adecuada implementación.

* Trabajo de grado

** Facultad de Ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Óscar Mauricio Reyes Torres, Ph.D.

Abstract

TITLE: ANÁLISIS DE REQUERIMIENTOS PARA EL DISEÑO DE UN LABORATORIO REMOTO DE RADIO DEFINIDO POR SOFTWARE*

AUTHORS: DIEGO ANDRÉS SARMIENTO CAMPOS**

Key words: Laboratory, remote, testbed, Software Defined Radio (SDR).

Description:

The first part of this work presents a compilation of information about remote laboratories based on Software Defined Radio (SDR) that currently exist worldwide. This compilation includes the specifications and distribution of the hardware in the facilities of each laboratory. In addition, it outlines the software used, the services provided to the users and the management of the demand for the offered resources. Based on the above, a reference model is chosen that allows defining the requirements of this kind of laboratory in an environment such as the Industrial University of Santander, having as selection criteria: quantity, quality and availability of information and the feasibility of establishing a user-laboratory interaction to demonstrate the operation of a laboratory of this nature.

Finally, a basic testing scenario (testbed) is made using the hardware and software resources available at the School of Electrical, Electronic and Telecommunications Engineering (E3T), in order to validate the requirement analysis performed. The tests carried out allow the elaboration of a concept of feasibility and establish the scope to which the E3T can target in the design of a remote laboratory based on SDR taking advantage of the currently available resources and determining as a priority what additional resources are necessary for an adequate implementation.

* Degree work.

** Physico-Mechanical Engineering Faculty. School of Electrical Engineering. Supervisor: Óscar Mauricio Reyes Torres, Ph.D

INTRODUCCIÓN

La tecnología de Radio Definido por Software¹ (SDR, por sus siglas en inglés) ha venido evolucionando en los últimos años sus aplicaciones en el área de las radiocomunicaciones, las cuales constan de tecnologías inalámbricas heterogéneas, conmutación y enrutamiento de protocolos de red, entre otras²³. Estos sistemas realizan la implementación en software de varios componentes que típicamente estarían implementados en hardware (mezcladores, filtros, moduladores/demoduladores, etc.). Un laboratorio remoto basado en SDR es un entorno que brinda la posibilidad de acceder a ambientes donde se puede trabajar, conocer y aprender remotamente acerca de las telecomunicaciones. A manera de ejemplo, para el caso colombiano, representaría una contribución a las herramientas usadas para el conocimiento y manejo del espectro radioeléctrico, tanto para los investigadores y entidades del estado (tal como la Agencia Nacional del Espectro⁴), como para la comunidad en general. Es a través de las tecnologías de la información y la comunicación (TIC)⁵, que es posible el acceso, manejo y comunicación de datos necesarios en el desarrollo de estos entornos.

Una validación en pequeña escala de un laboratorio remoto basado en SDR, permite evidenciar el funcionamiento, potencialidades y posibles debilidades de un laboratorio de

¹ W. H Tuttlebee. *Software defined radio: enabling technologies*. John Wiley & Sons, (Ed.). (2003).

² M. Ettus, & M. Braun, The Universal Software Radio Peripheral (USRP) family of low-cost SDRd. *Opportunistic Spectrum Sharing and White Space Access: The Practical Reality*, pp 3-23, (2015).

³ P. Deng, & M. Kavehrad, Real-time software-defined single-carrier QAM mimo visible light communication system. In *Integrated Communications Navigation and Surveillance (ICNS)*, (pp. 5A3-1), 2016. IEEE.

⁴ ANE [sitio web], Agencia Nacional del Espectro [Consulta 30 de enero 2017]. Disponible en: <http://www.ane.gov.co/>.

⁵ J. Huidobro, Tecnologías de información y comunicación. *Universidad Politécnica de Madrid*, pp 2, (2007).

este tipo. Teniendo en cuenta lo anterior, se planteó el objetivo general del presente trabajo, el cual es: “*Evidenciar el funcionamiento experimental básico de un laboratorio remoto de SDR mediante la realización de un testbed, basado en una revisión de experiencias internas y externas*”. Consecuentemente, también se plantearon los objetivos específicos, los cuales son:

- Recopilar información de laboratorios remotos basados en SDR disponibles a nivel mundial.
- Describir el funcionamiento de un laboratorio remoto basado en SDR y los servicios que brinda a los usuarios.
- Definir necesidades mínimas de software y hardware para el funcionamiento de un laboratorio remoto basado en SDR.
- Comparar el contexto de un laboratorio remoto existente basado en SDR con el de la Universidad Industrial de Santander.
- Realizar un testbed que aproveche los recursos físicos disponibles en la E3T.

Los resultados de este trabajo se encuentran organizados en este documento de la siguiente manera:

En el primer capítulo se describe lo que es un laboratorio remoto basado en SDR, enfatizando en los conceptos que componen este entorno. Posteriormente, con la finalidad de contextualizar el trabajo a realizar en la Universidad Industrial de Santander (UIS), en el segundo capítulo se encuentra la revisión de antecedentes a nivel nacional e internacional. En el tercer capítulo se presenta una descripción detallada de uno de los laboratorios referidos en el capítulo dos, con la finalidad de contar con un modelo de referencia para el desarrollo del presente proyecto y trabajos posteriores en el área. En los capítulos cuatro y cinco se documentan detalladamente los escenarios estudiados en el banco de pruebas realizado en la UIS, así como también los resultados obtenidos en cada una de las pruebas. Finalmente, además de las conclusiones del trabajo, se presenta un análisis documentado de cuáles serían los pasos a seguir para la adecuación y puesta en marcha de un laboratorio remoto de comunicaciones en la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones (E3T) de la UIS.

1. ¿QUÉ ES UN LABORATORIO REMOTO BASADO EN SDR?

Conocer lo que significa un laboratorio remoto basado en SDR requiere claridad en los conceptos que componen esta clase de entorno. A continuación, se detallan aspectos clave para tener claridad respecto a conceptos sobre laboratorio remoto y radio definido por software (SDR).

1.1. LABORATORIO REMOTO

Un laboratorio remoto⁶ está conformado por equipos físicos a los que se puede acceder a través de una red de datos (p. ej. Internet) para realizar diferentes clases de procesos que dichos equipos permitan *in situ* (Figura 1).

La operación básica de un laboratorio remoto consiste en: una solicitud de acceso por parte del usuario desde un dispositivo que cuente con acceso a una red de datos, la cual es atendida por un servidor de enlace según la disponibilidad, brindando así acceso a la base de datos y al servidor del laboratorio que gestiona la disponibilidad de manejo y control de hardware y software del laboratorio físico. El protocolo de comunicación más común para el manejo de usuarios en los laboratorios remotos es el protocolo de configuración dinámica de host (DHCP)⁷, que permite al servidor asignar a los usuarios direcciones IP dinámicas cuando estén disponibles, sabiendo en todo momento quién ha usado cada IP, cuánto tiempo lo ha hecho y a quién se la ha asignado después.

⁶ A. P. Medina, G. H. Saba, J. H. Silva, & E. L. de Guevara Durán, Los laboratorios virtuales y laboratorios remotos en la enseñanza de la ingeniería. *Revista internacional de educación en Ingeniería* (2011).

⁷ R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, & M. Carney, *Dynamic host configuration protocol for IPv6 (DHCPv6)* (No. RFC 3315), (2003).

Un laboratorio remoto habitualmente se confunde con un laboratorio virtual o un laboratorio distribuido. Al realizar una clasificación de los diferentes tipos de laboratorios existentes con base en la accesibilidad y uso de cada uno, difieren principalmente por la naturaleza del sistema de operación y la forma de acceder a los recursos que ofrece. Un laboratorio virtual es un sistema computacional que puede simular un laboratorio físico, en donde los experimentos se llevan a cabo siguiendo un procedimiento similar al que se sigue en un laboratorio convencional⁸ y un laboratorio distribuido es un laboratorio físico cuyo hardware se encuentra ubicado en diferentes puntos, y puede o no ser remoto. En la Tabla 1 se puede apreciar dicha clasificación.

Figura 1. Descripción del funcionamiento de un laboratorio remoto

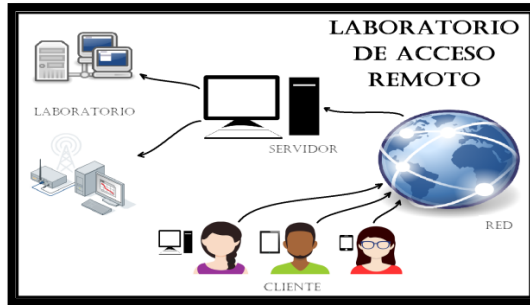


Tabla 1. Clasificación de los laboratorios según acceso y naturaleza del recurso

		Naturaleza del recurso	
		<i>Real</i>	<i>Simulado</i>
Acceso al recurso	<i>Local</i>	Laboratorio físico.	Laboratorio virtual monousuario

⁸ W. Yovany, & A. L. Fagua, Laboratorios Remotos y Virtuales: Una Herramienta para el Desarrollo de Prácticas en Ingeniería. Ciencia, Innovación y Tecnología, 1, pp 71-80, (2014).

	<i>Remoto</i>	Telelaboratorio o laboratorio remoto.	Laboratorio virtual multiusuario
--	---------------	---------------------------------------	----------------------------------

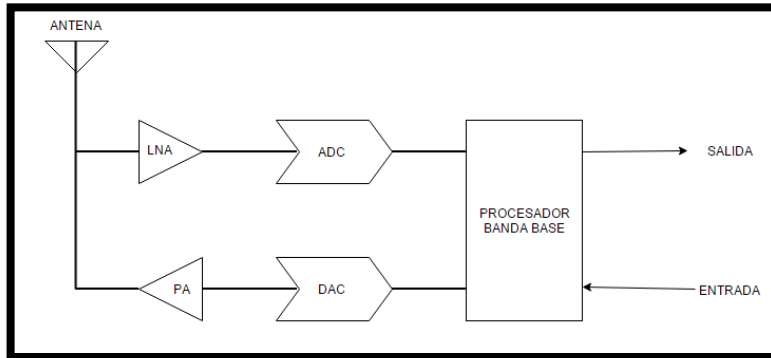
1.2. RADIO DEFINIDO POR SOFTWARE (SDR)

El concepto de radio definido por software, desarrollado por Joseph Mitola a mediados de la década del 90⁹, se centra en sistemas de radio que no dependen de manera exclusiva de dispositivos físicos, incorporando diversas etapas cuyas acciones son ejecutadas por un sistema de cómputo.

El impacto generado por SDR en el mundo de la radio es el remplazo de algunos circuitos que hacían parte de la composición de un equipo de radio por software capaz de llevar a cabo las mismas funciones. En la Figura 2 se observa el diagrama ideal de un sistema de radio definido por software, en el que el procesador maneja todas las funciones de radio, filtrado, *up / down conversion*, modulación / demodulación y banda base digital. En consecuencia, la antena se conectaría directamente al amplificador de bajo ruido (LNA) y el convertor análogo-digital (ADC) en la recepción de datos y al amplificador de potencia (PA) y el convertor digital-análogo (DAC) en la transmisión.

⁹ J. Mitola, The software radio architecture. IEEE Communications magazine, 33(5), pp 26-38, (1995).

Figura 2. Radio Definido por Software ideal



Un modelo de radio libre y sin restricciones de uso, resulta una alternativa de trabajo atractiva para los investigadores del área de las comunicaciones. GNU Radio¹⁰, por ejemplo, es una herramienta de desarrollo libre y abierta que provee bloques de procesamiento de señal para implementar sistemas de SDR y se encuentra basado en componentes de hardware. La Universidad Industrial de Santander (UIS) cuenta con algunos de dichos dispositivos, como son los USRP (*Universal Software Radio Peripheral*); los cuales son transceptores que junto con un computador se convierten en un sistema de generación de prototipos inalámbricos. Estos dispositivos incluyen generación y sincronización de reloj, FPGA, ADC, DAC, interfaz del proceso de comunicación y regulación de potencia¹¹.

Existe una clasificación por series de los USRP, la cual es llevada a cabo con la finalidad de suplir la funcionalidad requerida para cada aplicación sin exceder los recursos necesarios. Los dispositivos disponibles en la Universidad Industrial de Santander para estas series de la marca Ettus Research son: USRP1 (*Bus series*), N210 (*Networked series*), X310 (*X series*) y E310 (*Embedded series*). A continuación se realiza una descripción de las características de estos equipos:

¹⁰ GNU Radio [sitio web], GNU Radio [Consulta 12 de marzo 2017]. Disponible en: <https://gnuradio.org/>.

¹¹ D. Shen, Tutorial 4: The USRP Board. Introduction. SDR Documentation. Notre Dame, IN: University of Notre Dame (2005).

USRP1: Incluye una FPGA Altera Cyclone, doble ADC 64 MS/s, doble DAC 128 MS/s y conectividad USB 2.0 para proporcionar datos a los procesadores host. El USRP1 puede soportar dos *daughterboards* RF.

USRP N210: La Universidad Industrial de Santander tiene el USRP NI2920 de National Instruments, el cual tiene las mismas características del Ettus N210. Tiene una FPGA Xilinx Spartan 3A-DSP 3400, doble ADC 100 MS/s, doble DAC 400 MS/s y conectividad Gigabit Ethernet para la transmisión de datos. Permite la sincronización de varios dispositivos de la misma referencia mediante un puerto de expansión.

USRP X310: Cuenta con dos *daughterboards* de banda ancha que cubren desde 0 Hz hasta 6 GHz, 120 MHz de ancho de banda bandabase, múltiples opciones de interfaz de alta velocidad (PCIe, Giga Ethernet y 10 Giga Ethernet) y una FPGA KINTEX/.410T. La FPGA incorporada en este dispositivo provista con el UHD permite la interacción de éste con otros dispositivos USRP.

USRP E310: Cuenta con un transceptor 2x2 MIMO AD9361 que proporciona hasta 56 MHz de ancho de banda instantáneo y un rango de frecuencias de 70 MHz a 6 GHz, una FPGA Xilinx Zynq 7020 SoC que combinada con un procesador de doble núcleo ARM Cortex A9 permite una operación del dispositivo independiente. Adicionalmente, cuenta con periféricos como un receptor de GPS, dos puertos USB y entrada-salida. La plataforma USRP *Embedded Series* utiliza el framework OpenEmbedded para crear distribuciones personalizadas de Linux adaptadas a las necesidades específicas de la aplicación. El sistema operativo predeterminado está preinstalado con la API de software UHD y una variedad de herramientas de desarrollo de terceros como GNU Radio.

2. ANTECEDENTES

El concepto de laboratorio remoto de comunicaciones basado en SDR se ha venido desarrollando en diferentes escenarios, brindando a los usuarios una herramienta para la investigación en temas como: Wi-Fi, WiMAX, LTE, Bluetooth, entre otros. Dichos escenarios se pueden encontrar tanto a nivel nacional como internacional y algunos de ellos son descritos en el presente capítulo.

2.1. REFERENTES INTERNACIONALES

Actualmente existen varios laboratorios remotos basados en SDR a nivel mundial, que difieren entre sí principalmente por la cantidad de nodos y la variedad de servicios que tienen disponibles para brindar a los usuarios. Tienen como factor común que todos son de acceso libre y brindan las herramientas necesarias para conocer su entorno y facilitar su manejo. En la Tabla 2 se referencian cuatro de los laboratorios remotos basados en SDR que existen y operan actualmente, y dónde están ubicados.

Tabla 2. Algunos laboratorios basados en SDR a nivel mundial

Nombre	Ubicación geográfica
Orbit (<i>Open Access Research Testbed for Next Generation Wireless Networks</i>) - GENI Lab (<i>global environment for network innovation</i>)	Estados Unidos
FIBRE (<i>Future Internet Brazilian Environment for Experimentation</i>)	Brasil
NITOS (<i>Network Implementation Testbed Using Open Source Platforms</i>).	Francia
R2lab <i>anechoic chamber - a heterogeneous wireless testbed</i>	Francia

La línea de trabajo que manejan todos estos laboratorios es similar; sin embargo, en la mayoría de los casos la estructura de hardware y la configuración de software interna y de usuarios es diferente. A continuación, se presenta una recopilación de información en cuanto a especificaciones de hardware, uso de software, servicios brindados y manejo de usuarios de cada uno de los laboratorios nombrados en la Tabla 2.

Orbit Lab¹² es un laboratorio utilizado como soporte para los servicios inalámbricos que ofrece GENI Lab, los cuales son: proporcionar recursos de computación y de red, permitir la conectividad no IP entre recursos, facilitar la instalación de software y sistemas operativos personalizados, brindar la reproducibilidad de los experimentos y prestar instrumentos y herramientas de medición de manera remota. Orbit Lab es un banco de pruebas de acceso abierto destinado para la investigación de redes inalámbricas, tiene ubicadas geográficamente sus instalaciones físicas en EEUU en la Universidad de Rutgers en New

¹² Orbit [sitio web], Open-Access Research Testbed for Next-Generation Wireless Networks [Consultado 19 de octubre 2016]. Disponible en: <http://www.orbit-lab.org/>.

Jersey. De la misma manera el centro de operaciones de GENI Lab¹³ está ubicado en EEUU dentro Universidad de Indiana y es conocido como GMOC¹⁴ (*Geni meta-operations center*).

En cuanto a software, Orbit Lab puede ser visto como un conjunto de servicios en donde se introduce una definición experimental (*script*) y el usuario recibe los resultados de dicha definición luego de realizado el procesamiento interno. Es posible reiniciar cada uno de los nodos de una rejilla de 20x20 con la que cuenta Orbit Lab, cargar un sistema operativo y cambiar en cada nodo el software preestablecido. El *script* también especifica el filtrado y la recogida de los datos experimentales y genera un esquema de base para apoyar el análisis posterior de los mismos, lo que permite prestar un servicio a los usuarios que facilita trabajar en áreas tales como: coordinación del espectro *multi-radio*, redes de radio cognitiva, redes WiFi densas, Celular/WiFi *multi-homing*, entrega de contenidos móviles, protocolos de reconocimiento de ubicación, seguridad inalámbrica entre capas, la futura arquitectura de Internet y la computación en la nube móvil, entre otras.

Es importante mencionar que el manejo de usuarios es llevado a cabo mediante soporte para generadores de tráfico de uso común como *Iperf* y son regulados mediante el *Dynamic Host Configuration Protocol*.

FIBRE¹⁵ está compuesto por la integración de 13 nodos de experimentación, diez ubicados en Brasil, ampliamente distribuidos en siete estados brasileños, y tres en Europa: uno en la Fundación I2Cat (España), uno en la Universidad de Bristol (Reino Unido), y uno en la Universidad de Tesalónica (Grecia). Las especificaciones de la máquina dependen del

¹³ GENI [sitio web], Global Environment for Network Innovation [Consultado 19 de octubre de 2016]. Disponible en: <https://www.geni.net/>.

¹⁴ GMOC [sitio web], Geni Meta-Operations Center [Consultado 26 de octubre de 2016]. Disponible en: <http://gmoc.gnoc.iu.edu/>.

¹⁵ FIBRE [sitio web], Future Internet Brazilian Environment for experimentation [Consultado 28 de octubre de 2016]. Disponible en: <http://fibre.org.br/>.

software que se vaya a implementar al momento de usar el laboratorio (Xen, LDAP, OCF, OMF, Flowvisor, perfSONAR, VPN o Zabbix Proxy).

FIBRE funciona como un laboratorio virtual a gran escala para estudiantes e investigadores para probar nuevas aplicaciones y modelos de arquitectura de red, tanto en la conectividad física, como en los niveles de control de equipos y datos.

NITOS¹⁶ basa su trabajo en 100 nodos de gran alcance (algunos de ellos móviles), que cuentan con múltiples interfaces inalámbricas y una infraestructura de nube. La conectividad de red se establece mediante el uso de una serie de conmutadores modulares.

Las redes definidas por software (SDN) disponibles en NITOS, consisten en múltiples tecnologías OpenFlow, interruptores activados y conectados a los nodos. El banco de pruebas de Software Defined Radio (SDR) consta de USRP's unidos a los nodos inalámbricos, lo cual permite la experimentación con tecnologías inalámbricas heterogéneas y la conmutación y enrutamiento de protocolos de red. Además, los USRPs ofrecen a los investigadores la posibilidad de programar una serie de características de la capa física.

El control y la gestión de datos se realizan mediante el control y el marco de gestión (OMF) de software de código abierto. El cual permite lograr la reproducibilidad de la experimentación.

¹⁶ NITOS [sitio web], Network Implementation Testbed using Open Source platforms [Consultado 28 de octubre 2016]. Disponible en: <http://nitlab.inf.uth.gr/NITlab/nitos>.

El laboratorio remoto R2lab¹⁷ cuenta con 37 nodos que están disponibles para proporcionar una estructura de banco de pruebas. Los nodos están distribuidos en un diseño de cuadrícula y son personalizables, permitiendo una gran variedad de escenarios en la experimentación. Los nodos son totalmente abiertos y los usuarios pueden instalar cualquier paquete de software que necesiten. También se puede cargar el sistema operativo de elección en cualquier nodo. Lo que les permite tener acceso ssh a todos los nodos con privilegios de administración, y configurar los recursos disponibles o los mismos nodos, USRPs y móviles; para crear un entorno experimental amplio. El acceso al laboratorio es regulado mediante el *Dynamic Host Configuration Protocol*.

2.2. REFERENTES NACIONALES

A nivel nacional también se han desarrollado algunos avances relevantes en el área:

La Universidad de Antioquia (UdeA) desarrolló un laboratorio remoto de telecomunicaciones con la finalidad de que los estudiantes de la modalidad virtual, tuvieran disponibilidad para realizar sus prácticas de laboratorio sin necesidad de desplazarse a las instalaciones de la universidad como habitualmente se hacía. El laboratorio es completamente funcional, pero solamente está habilitado su uso para estudiantes vinculados a la Universidad de Antioquia. Cuentan con videotutoriales que facilitan la adaptación de los usuarios con el entorno de trabajo. El equipo consta de un analizador y generador de señal, un osciloscopio, un generador de funciones, entre otras herramientas y significó una inversión de 700 millones de pesos.

En la actualidad se han beneficiado no únicamente los estudiantes de la modalidad virtual de Telecomunicaciones, sino también de otras ingenierías como Telecomunicaciones presencial, Sistemas, Eléctrica y Electrónica, entre otras¹⁸.

¹⁷ r2lab [sitio web], R2lab anechoic chamber a heterogeneous wireless testbed [Consultado 1 de noviembre de 2016]. Disponible en: <http://r2lab.inria.fr/hardware.md>.

¹⁸ J. D. Umaña Gallego, Nuevo laboratorio remoto en la Universidad de Antioquia. http://www.elcolombiano.com/historico/nuevo_laboratorio_remoto_en_la_u_de_a-BAEC_270735, Noviembre 2013.

Es requisito ser estudiante de la Universidad de Antioquia para tener acceso al laboratorio y el proceso para manejar los recursos consiste en:

- Realizar una reserva del servidor.
- Descargar el asistente de la plataforma del laboratorio.
- Configuración VPN del equipo que se esté usando con la red de la universidad.
- Acceder a la plataforma del laboratorio con la contraseña enviada al correo electrónico cuando se realizó la reserva.
- Usar los recursos acordes a las necesidades.

De igual forma, teniendo como referente el laboratorio Orbit de la Universidad de Rutgers en New Jersey, el Grupo de Investigación en seguridad y sistemas de Comunicaciones del Programa en Ingeniería de Telecomunicaciones de la Universidad Militar Nueva Granada (UMNG) desarrolló un proyecto de investigación financiado por la Agencia Nacional del Espectro y la UMNG. El Wirid-Lab¹⁹, es un laboratorio enfocado para la docencia y desarrollo de proyectos de Investigación e Innovación en comunicaciones inalámbricas.

Wirid-Lab permite a los usuarios realizar experimentos mediante un acceso remoto a una infraestructura compuesta por equipos de Radio Definido por Software (SDR), generadores vectoriales de radiofrecuencia, generadores de microondas y analizadores de espectro.

El proceso de manejo del Wirid-Lab es similar al laboratorio desarrollado por la Universidad de Antioquia. Las variaciones entre estos laboratorios se basan en el método de conexión de los usuarios y la variedad de recursos (Wirid-Lab tiene SDR). En el Wirid-Lab los usuarios realizan la conexión mediante *ssh* y en el laboratorio de la UdeA se lleva a cabo mediante VPN.

¹⁹ Wirid Lab [sitio web], Wireless research innovation and development laboratory [Consultado 8 de diciembre de 2016]. Disponible en: <http://wirid-lab.umng.edu.co/>, Universidad Militar Nueva Granada.

2.3. REFERENTES INSTITUCIONALES

En la Universidad Industrial de Santander se desarrolló un trabajo de control remoto con SDR en el año 2015, realizado gracias a la alianza entre esta institución y la Agencia Nacional del Espectro (ANE). El proyecto de investigación SDRCol²⁰ llevado a cabo por el grupo de investigación Radiogis, tuvo como finalidad el uso de SDR como apoyo a la gestión del espectro radioeléctrico en Colombia; allí se accedió exitosamente de manera remota al equipo USRP E310.

Cabe resaltar que, aunque la institución cuenta con equipos especializados en el área que permiten la operación de sistemas de acceso remoto, el proyecto de SDRCol ha sido el único que ha evidenciado esta modalidad de trabajo. Los equipos son utilizados actualmente como herramienta de aprendizaje y desarrollo. Desde el 2011 se evidencian trabajos relacionados al manejo de SDR en diferentes áreas, realizados por los grupos de investigación CPS (Conectividad y Procesamiento de Señales)²¹²² y RadioGis²³ de la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones y el grupo HDSP (High Dimensional Signal Processing)²⁴ de la Escuela de Ingeniería de Sistemas.

²⁰ SDRCol2015 [sitio web], Proyecto SDRCol2015 [Consultado 1 de febrero de 2017]. Disponible en: <https://sites.google.com/a/radiogis.uis.edu.co/proysdrcol2015/home>.

²¹ J. M. Suarez Monsalve and V. Triana Galeano, Diseño de manual de prácticas para un laboratorio de comunicaciones digitales basado en la técnica de radio definido por software. Universidad Industrial de Santander, (2012).

²² A. F. Gómez García, Sistema de comunicación digital m-aria sobre una plataforma de radio definido por software basado en gnu radio, Universidad Industrial de Santander (2015).

²³ L. C. Díaz Bautista, M. J. Rangel Silva, and H. Ortega, Implementación de un downconverter sobre una FPGA bajo el concepto de software defined radio usando xilinx system generator. Universidad Industrial de Santander (2011).

²⁴ S. A. Pino Gallardo, Codificación conjunta de fuente-canal: Mapeos de Shannon-Kotelnikov sobre radio definido por software, Universidad Industrial de Santander, (2013).

3. MODELO DE LABORATORIO REMOTO BASADO EN SDR

Los laboratorios descritos en el capítulo 2 cuentan en su mayoría con un software destinado al manejo de señales de control y datos (usuarios-laboratorio), y un hardware que cuenta con componentes de radio y demás equipos destinados al trabajo en el área de comunicaciones.

Se presenta en este capítulo un modelo de laboratorio remoto basado en SDR siguiendo como referencia la plataforma y el sistema de operación de GENI Lab y el diseño de hardware implementado por Orbit Lab, teniendo en cuenta que estos dos laboratorios trabajan de manera complementaria, donde Orbit Lab es utilizado como soporte para los servicios inalámbricos que ofrece GENI Lab. Adicionalmente, el hecho de que se cuente con un antecedente a nivel nacional como es el caso de Wirid-Lab, que ha adoptado este modelo de laboratorio para su implementación, facilitaría el desarrollo de un laboratorio distribuido a nivel nacional.

3.1. GENI LAB

Es una plataforma que opera como un laboratorio virtual que interconecta algunos de los recursos inalámbricos disponibles en universidades a lo largo de Estados Unidos y presta servicios de manera remota a nivel mundial, para la investigación y educación en redes y sistemas distribuidos. Es muy apropiado para explorar redes a escala, promoviendo así innovaciones en la ciencia, la seguridad, los servicios y las aplicaciones de la red. GENI permite a los experimentadores:

- Obtener recursos computacionales de ubicaciones en diferentes lugares en los Estados Unidos.
- Conectar recursos de computación utilizando redes de capa 2 (enlace) en topologías que mejor se adapten a los experimentos requeridos.
- Instalar software personalizado o incluso sistemas operativos personalizados.

- Controlar cómo los conmutadores de red manejan los flujos de tráfico.
- Ejecutar protocolos propios de capa 3 (red) y superiores instalando software de protocolo en los recursos de cálculo del usuario y proporcionando controladores de flujo para los conmutadores.

3.1.1. Servicios brindados a los usuarios. GENI Lab es una herramienta útil para investigadores que requieran:

- Una infraestructura de experimentación a gran escala, pues este laboratorio tiene la capacidad de proporcionar recursos de computación y de red.
- Conectividad no IP entre recursos, debido a que permite configurar conexiones de enlace de datos entre recursos de computación, ejecutar una red propia y protocolos superiores que conectan estos recursos.
- Programación profunda, ya que permite programar no sólo los hosts finales de la red experimental sino también los *switches* en el núcleo de la red. Permitiendo la experimentación de nuevos protocolos de capa de red o de nuevos algoritmos de enrutamiento IP.
- Reproducibilidad, debido a que permite acceder a los recursos de la CPU del laboratorio y los recursos de red. Dando así control sobre el entorno de los experimentos y por lo tanto la capacidad para repetirlos en condiciones idénticas o muy similares.
- Instrumentación y herramientas de medición, pues este laboratorio cuenta con sistemas de instrumentación y medición para mediciones activas y pasivas, almacenamiento de datos de medición y herramientas para visualizar y analizar datos medidos.

3.1.2. Arquitectura. La arquitectura de la red GENI Lab está diseñada para permitir múltiples experimentos simultáneos, cada uno de los cuales puede utilizar diferentes protocolos y algoritmos de reenvío de paquetes. En la Figura 3 se muestra la arquitectura

sobre la que trabaja la red de GENI Lab, la cual consta de un plano de control que se ejecuta a través de internet y es utilizado para reservar, acceder, programar y administrar los recursos de computación y comunicación. De igual forma, consta de un plano de datos que es configurado dependiendo de la especificación de la topología de red deseada por el usuario.

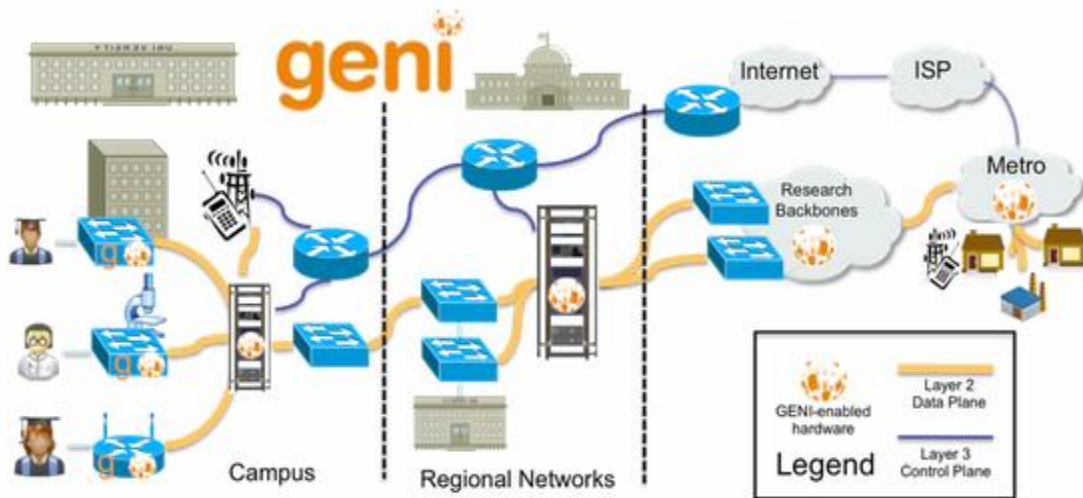
La arquitectura de la red GENI Lab está diseñada para permitir múltiples experimentos simultáneos, cada uno de los cuales puede utilizar diferentes protocolos y algoritmos de reenvío de paquetes. En la Figura 3 se muestra la arquitectura sobre la que trabaja la red de GENI Lab, la cual consta de un plano de control que se ejecuta a través de internet y es utilizado para reservar, acceder, programar y administrar los recursos de computación y comunicación. De igual forma, consta de un plano de datos que es configurado dependiendo de la especificación de la topología de red deseada por el usuario.

Los enlaces de red de GENI están divididos por VLANs Ethernet²⁵, es decir, varios experimentos comparten el mismo enlace físico y reciben distintas VLAN en ese enlace, lo que garantiza el aislamiento del tráfico entre los experimentos solicitados.

Cabe resaltar, que GENI es un banco de pruebas descentralizado, es decir, diferentes organizaciones alojan sus recursos en GENI y los ponen a disposición de los usuarios. Esto incluye recursos de red que son propiedad de proveedores regionales y nacionales de redes de investigación y educación y de campus que albergan *racks* de GENI o estaciones base inalámbricas. La configuración del plano de datos para un experimento solicitado por un usuario puede requerir coordinación entre varios proveedores de recursos de red. Adicionalmente, los usuarios tienen la posibilidad de acceder a los recursos de manera presencial (campus) y remota.

²⁵ P. J. Frantz, & G. O. Thompson, U.S. Patent No. 5,959,990. Washington, DC: U.S. Patent and Trademark Office, (1999).

Figura 3. Arquitectura de red de GENI Lab ²⁶



En particular, la infraestructura física para prestar los servicios asociados a sistemas basados en SDR, se encuentra en el Orbit Lab, cuya arquitectura se describe en la siguiente sección.

3.2. ORBIT LAB

Orbit Lab es un laboratorio utilizado como soporte para los servicios inalámbricos que ofrece GENI Lab, ya que es un banco de pruebas de acceso abierto destinado para la investigación de redes inalámbricas y cuenta con una infraestructura que se describe a continuación.

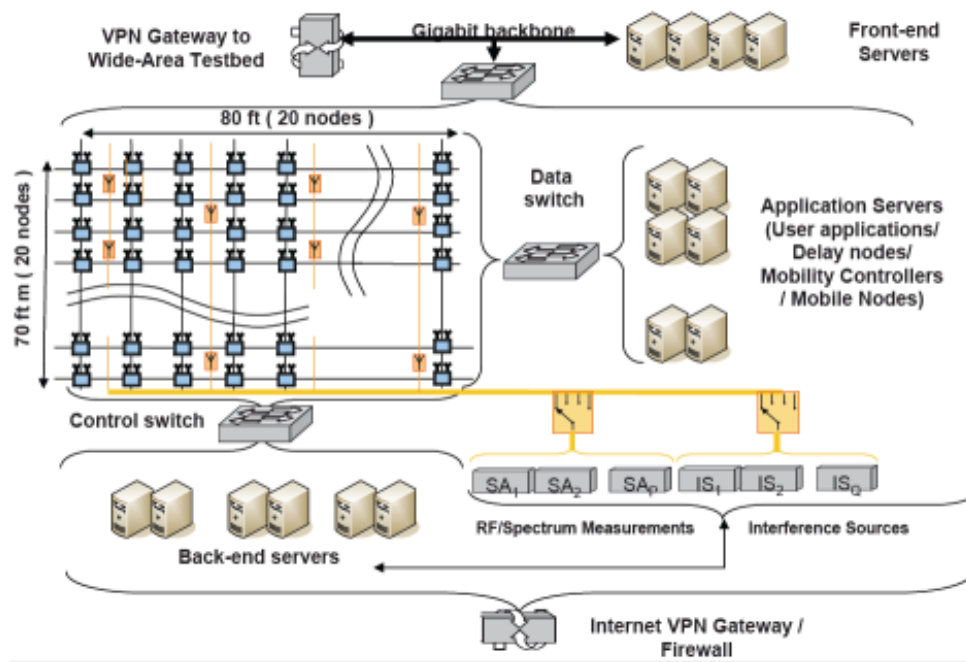
3.2.1. Hardware. En la Figura 4 se ilustra la distribución de hardware que usa Orbit Lab, la cual consta de

- Equipos destinados al control del sistema y manejo de datos (*VPN gateway to wide-area testbed, front-end servers, back-end servers, data switch and control switch*).
- Una rejilla de 20x20 nodos.

²⁶ GENI Lab [sitio web], Arquitectura de red de GENI Lab [Consultado 25 de noviembre de 2016]. Disponible en: <http://www.geni.net/documentation/geni-architecture/>.

- 9 pequeñas rejillas de prueba adicionales (también conocidas como *sandboxes* [SA, IS]).

Figura 4. Distribución de Hardware de Orbit Lab²⁷



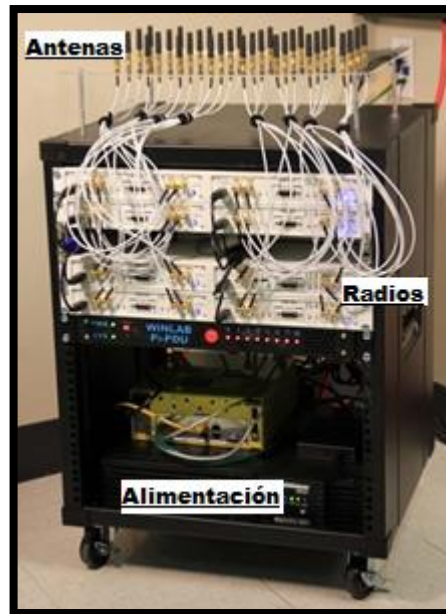
3.2.1.1, Rejilla (Grid). La rejilla de Orbit Lab se constituye de una red de nodos de 20x20, los cuales constan de una separación de 3 ft entre ellos y tiene un área total de 70x80 ft². Hay cuatro mini-racks de SDR's con 32 antenas (cada rack contiene 8 Ettus USRP X310 equipados con dos placas secundarias de radio UBX 160), cada uno está situado en una esquina de la rejilla. En la Figura 5 se puede apreciar el hardware correspondiente a un mini-rack, estos equipos son destinados específicamente para experimentos Massive-MIMO²⁸. Adicionalmente, cada fila de la rejilla consta de dos sistemas de conmutación

²⁷ Orbit Lab [sitio web], Distribución de hardware de Orbit Lab [Consultado 19 de octubre de 2016]. Disponible en: <http://www.orbit-lab.org/attachment/wiki/Hardware/hardware-50.png>.

²⁸ E. G. Larsson, O. Edfors, F. Tufvesson, & T. L. Marzetta, Massive MIMO for next generation wireless systems. IEEE Communications Magazine, 52(2), pp 186-195, (2014).

(switches) que varían dependiendo del requerimiento de recursos realizado por uno o varios usuarios, uno para realizar acciones de control y otro para el manejo de datos.

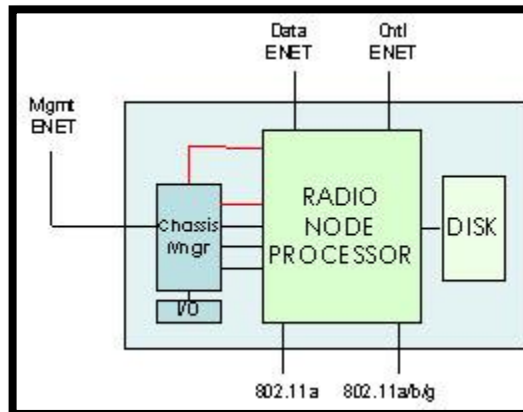
Figura 5. Hardware de un mini-rack (Massive-MIMO)²⁹



Cada nodo de la red de Orbit Lab consta de un nodo de radio con dos interfaces de radio, dos interfaces Ethernet para el control de experimento y de datos, y un chasis Manager (CM), como se puede observar en la Figura 6. Este último es un subsistema simple, fiable e independiente de la plataforma para la gestión y seguimiento de forma autónoma el estado de cada nodo de la red del banco de pruebas.

²⁹ Orbit Lab [sitio web], Hardware de un mini-rack (Massive-MIMO) [Consultado 2 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/attachment/wiki/Hardware/bDomains/aGrid/Massive-MIMO.jpg>.

Figura 6. Nodo de radio de Orbit Lab³⁰



Existen actualmente 3 versiones distintas de nodos de radio:

Los nodos de generación 1 cuyas características se presentan en la Tabla 3, que comprenden el mayor número de nodos que son usados en la red principal:

Tabla 3. Características de los nodos de radio de primera generación³¹

CPU	1 GHz VIA C3
Memoria	512 MB
Almacenamiento	Unidad IDE, la capacidad varía
Conectividad	2 x 100BaseT puertos Ethernet
Puertos	2 x USB, 2 x RS232
Potencia	Fuente de alimentación estándar 20-pin y 12V CPU

³⁰ Orbit Lab [sitio web], Descripción de Nodo de Orbit Lab [Consultado 2 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/attachment/wiki/Hardware/jCM/OrbitNode.jpg>.

³¹ Orbit Lab [sitio web], Características de nodo de generación 1 [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/aNodeVer1#ORBITNodeVersion1>.

Los nodos de generación 3 son usados en las *sandboxes*, en aplicaciones móviles y parte de la rejilla principal. Estos nodos se dividen en dos grupos: los principales y los móviles. A su vez, existen 3 tipos de nodos principales y 2 tipos de nodos móviles. Las características de cada tipo de nodo son descritas a continuación:

Nodos principales: Nodo tipo LV-67C: cuenta con 3 ranuras de expansión: un conector PCIe (x16), un Mini-PCIe y un conector Mini-PCI tipo III, las especificaciones técnicas se muestran en la Tabla 4. Este tipo de nodos es usado actualmente por la rejilla principal de Orbit y por algunas *sandboxes*.

Tabla 4. Características de los nodos de radio de tercera generación tipo LV-67C³²

CPU	LGA 1156 i7 Quad Core
Chipset	Intel BD82Q57 PCH, Intel Q57 PCH con Winbond W83627DHG
Memoria	8 GB DDR3 1066 / 1333MHz
Almacenamiento	SATA 300 con 160 GB 7200 RPM HD
Conectividad	2 x Intel 82574L 10/100/1000 controladores de Gigabit Ethernet
Puertos	8 x USB 2.0, 3 x RS232, 1 x RS232/422/485, 1 x IrDA
Potencia	Conector 1x24-Pin ATX y conector de alimentación (necesario) 1x4-pin 12V CPU.

Nodo tipo LV-67G: Cuenta con 3 ranuras de expansión: 1 x ranura PCI-Express X16, 1 x Mini-PCIe y 1 x conector Mini-PCI. Su uso se basa únicamente en algunas *sandboxes* y sus características se describen en la Tabla 5.

Tabla 5. Características de los nodos de radio de tercera generación tipo LV-67G³³

CPU	Intel i7 Quad Core en el conector LGA1155
------------	---

³² Orbit Lab [sitio web], Características de nodo de generación 3 tipo LV-67C [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/cNodeVer3/aMain/aLV67C#LV-67C>.

³³ Orbit Lab [sitio web], Características de nodo de generación 3 tipo LV-67G [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/cNodeVer3/aMain/bLV67G#LV-67G>.

Chipset	Intel Q67 Express Chipset
Memoria	8 GB DDR3 1066/1333 MHz
Almacenamiento	SATA 300 con 160 GB 7200 RPM HD
Conectividad	2 x Intel 82574L controladores de Gigabit Ethernet
Puertos	8 x USB 2.0, 3 x RS232, 1 x RS232/422/485, 1 x IrDA
Potencia	Fuente de alimentación estándar 24-pin ATX y P4 4-pin 12V.

Nodo tipo LV-67J: está compuesto por 3 ranuras de expansión: 1 x ranura PCI-Express X16 Gen3, 1 x conector Mini PCI, 1 tarjeta PCIEMini y las especificaciones técnicas mostradas en la Tabla 6. Este tipo de nodo es la versión más actual de los nodos principales de la generación 3. Sin embargo, se encuentra en fase de prueba y su uso no se ha desplegado oficialmente.

Tabla 6. Características de los nodos de radio de tercera generación tipo LV-67J³⁴

CPU	Intel Corp. Intel® Core™ i7-3770 CPU @ 3.40GHz
Chipset	Intel® Q77 Express chipset
Memoria	2 DDR3 1333MHz 16GiB
Almacenamiento	SATA 300 con 160 GB 7200 RPM HD
Conectividad	1 x Intel® 82574L Giga LAN and 1 x Intel® 82579LM Giga LAN
Puertos	4 x USB 2.0, 4 x USB 3.0, 3 x RS232, 1 x RS232/422/485
Potencia	Fuente de alimentación estándar 24-pin ATX y P4 4-pin 12V.

Nodos Móviles: Nodo tipo LV-67B: este tipo de nodo cuyas características se encuentran en la Tabla 7, es actualmente la base de todos los nodos móviles de Orbit Lab, está compuesto por 3 ranuras de expansión: un conector PCIe (x16), un Mini-PCIe y un conector Mini-PCI tipo III.

³⁴ Orbit Lab [sitio web], Características de nodo de generación 3 tipo LV-67J [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/cNodeVer3/aMain/cLV67J#LV-67J>.

Tabla 7. Características de los nodos de radio de tercera generación tipo LV-67B³⁵

CPU	Intel® Core™ 2 Duo con Socket P
Chipset	Intel GM45 y ICH9M
Memoria	8 GB DDR3 1066 / 1333MHz
Almacenamiento	SATA 300 con 160 GB 7200 RPM HD
Conectividad	2 x Intel 82573L controladores de Gigabit Ethernet
Puertos	6 x USB 2.0, 1 x RS232, 1 x RS232/422/485, 1 x IrDA, 1 x IEEE 1394
Potencia	Fuente de alimentación estándar 24-pin ATX o 9V ~ 24V DC de entrada

Nodo tipo LV-67F: consta de 3 ranuras de expansión: 1 x PCI-Express X16 ranura, 1 x Mini-PCIe y 1 x Mini-PCI socket. Las especificaciones técnicas de este tipo de nodo se encuentran en la Tabla 8.

Tabla 8. Características de los nodos de radio de tercera generación tipo LV-67F³⁶

CPU	Intel i7 Quad Core, rPGA988
Chipset	Mobile Intel QM57 Express chipset
Memoria	8 GB DDR3 800/1066 MHz
Almacenamiento	SATA 300 con 160 GB 7200 RPM HD
Conectividad	2 x Intel 82574L controladores de Gigabit Ethernet
Puertos	10 x USB 2.0, 1 x RS232, 1 x RS232/422/485, 1 x IrDA
Potencia	Fuente de alimentación estándar 24-pin ATX o 9V ~ 24V DC de entrada

Los nodos de cuarta generación están conformados por motherboards tipo Asus Q87T y LGA1150 están alimentadas por una única fuente de alimentación de 19V que los hace

³⁵ Orbit Lab [sitio web], características de nodo de generación 3 tipo LV-67B [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/cNodeVer3/bMobile/aLV67B#LV-67B>.

³⁶ Orbit Lab [sitio web], Características de nodo de generación 3 tipo LV-67F [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/cNodeVer3/bMobile/bLV67F#LV-67F>.

adecuados para su uso en automóviles y para otras aplicaciones móviles. Este tipo de nodos vienen en dos configuraciones: Nodos de mayor rendimiento con Intel I7-4770 3.4GHZ (8MB) con 16 GB de DDR3 1600 MHz RAM y Nodos de menor rendimiento con Intel I5-4440 3.1GHZ (6MB) con 8 GB de DDR3 1333 MHz RAM.

La generación 2 de nodos fue obsoleta.

3.2.1.2. Sandboxes. Son rejillas de prueba adicionales que aunque pueden cumplir con funciones generales demandas por los usuarios, su utilidad radica en que cada una de ellas tiene equipos especializados para llevar a cabo funciones específicas. En la Tabla 9 se describe el tipo de nodo y la especialización de cada una de estas rejillas adicionales:

Tabla 9. Sandboxes, tipo de nodo y especialización³⁷

Sandbox	Tipo de nodo	Especialización
SB1	Q87T	Atheros 9K and Intel Wireless N + WiMAX
SB2	LV-76G	USRP2, Atheros 9K
SB3	LV-67C	USRP2, Atheros 5K
SB4	LV-67G	Topologías basadas en atenuadores, Atheros 9K and WiMAX
SB5	LV-67C	USRP, Atheros 9K
SB6	LV-67C	CR-Kit (Zynq SDR), WARP, Intel devices & WiMAX
SB7	LV-67G	USRP2 Atheros 9K, ZigBee & WiMAX
SB8	LV-67C	Atheros 9K & WiMAX
SB9	AMD	NetFGPA, OpenFlow

3.2.1.3. SDR en Orbit Lab. La mayoría de los dispositivos de SDR en la red de Orbit Lab son varias generaciones de USRP, cada uno de estos equipos independientemente de si

³⁷ Orbit Lab [sitio web], Sandboxes, características [Consultado 9 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/bDomains/cSandboxes#Sandboxes>.

su conexión es mediante USB o Ethernet, está conectado a un nodo de generación 3 y se encuentra físicamente ubicado debajo del nodo al que está asociado

La mayoría de los dispositivos de SDR en la red de Orbit Lab son varias generaciones de USRP, cada uno de estos equipos independientemente de si su conexión es mediante USB o Ethernet, está conectado a un nodo de generación 3 y se encuentra físicamente ubicado debajo del nodo al que está asociado.

Los USRP disponibles en Orbit Lab son: USRP1, N210, B210 y X310, estos dispositivos se encuentran distribuidos alrededor de toda la red de Orbit Lab y su ubicación específica en la rejilla principal se encuentra en la Tabla 10. Sin embargo, la ubicación de estos equipos no solo se encuentra en la rejilla principal; las *sandboxes* (SB) también albergan varios equipos de SDR. La SB1 cuenta con 2 USRP B210, la SB5 cuenta con 2 USRP1 y las SB2, SB3 y SB7 cuentan cada una con 2 USRP2 N210.

Tabla 10. Ubicación en la rejilla principal de los dispositivos de SDR³⁸

Coordenada del nodo	Tipo de dispositivo	RF Daughter Boards (side A)	RF Daughter Boards (side B)
[18,20]	USRP X310	SBX120	CBX120
[3,20]	USRP X310	SBX120	CBX120
[18,1]	USRP X310	SBX120	CBX120
[3,1]	USRP X310	SBX120	CBX120
[2,3]	B210	-	-
[6,20]	B210	-	-
[9,1]	B210	-	-
[19,18]	B210	-	-
[5,16]	USRP2 N210	SBX	-
[8,8]	USRP2 N210	SBX	-
[8,13]	USRP2 N210	SBX	-

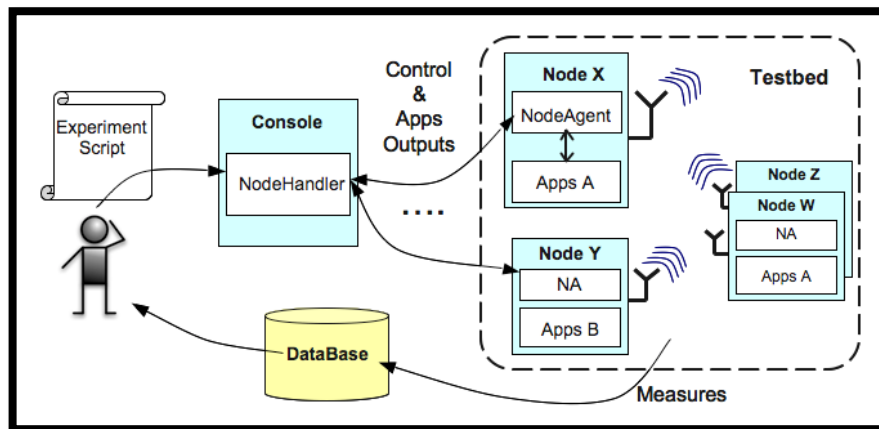
³⁸ Orbit Lab [sitio web], Ubicación de los dispositivos USRP de Orbit Lab [Consultado 18 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/fDevices/cSDR/gUSRP>.

[13,8]	USRP2 N210	SBX	-
[13,13]	USRP2 N210	SBX	-
[16,5]	USRP2 N210	SBX	-
[16,16]	USRP2 N210	SBX	-
[19,2]	USRP2 N210	SBX	-
[1,1]	USRP2 N210	SBX	-
[1,2]	USRP2 N210	SBX	-
[1,19]	USRP2 N210	SBX	-
[1,20]	USRP2 N210	SBX	-
[2,1]	USRP2 N210	SBX	-
[2,2]	USRP2 N210	SBX	-
[2,19]	USRP2 N210	SBX	-
[2,20]	USRP2 N210	SBX	-
[19,1]	USRP2 N210	SBX	-
[19,19]	USRP2 N210	SBX	-
[19,20]	USRP2 N210	SBX	-
[20,1]	USRP2 N210	SBX	-
[20,2]	USRP2 N210	SBX	-
[20,19]	USRP2 N210	SBX	-
[20,20]	USRP2 N210	SBX	-
[3,3]	USRP1	XCVR2450	WBX
[3,18]	USRP1	XCVR2450	WBX
[6,6]	USRP1	XCVR2450	WBX
[6,15]	USRP1	XCVR2450	WBX
[10,10]	USRP1	XCVR2450	WBX
[15,6]	USRP1	XCVR2450	WBX
[15,15]	USRP1	XCVR2450	WBX
[18,3]	USRP1	XCVR2450	WBX
[18,18]	USRP1	XCVR2450	WBX

3.2.2. Software. El banco de pruebas Orbit Lab es operado como un servicio compartido que permite llevar a cabo experimentos de redes inalámbricas de manera presencial y remota. Aunque solo se puede ejecutar un experimento a la vez por usuario, el uso del banco de pruebas es automático y permite que la funcionalidad del sistema sea eficiente,

guardando los resultados en una base de datos para su posterior análisis. Dicho de otra manera, Orbit Lab es un conjunto de servicios en donde se introduce una definición experimental (script) y el usuario recibe los resultados de dicha definición luego de realizado el procesamiento interno, En la Figura 7 se ilustra dicho procedimiento. Estas definiciones experimentales dadas por los usuarios, pueden llevar a cabo tareas tales como: reiniciar cada uno de los nodos en la rejilla principal, cargar un sistema operativo, cargar cualquier software de sistema modificado y software de aplicación en cada nodo y establecer los parámetros relevantes para el experimento descrito. De manera similar, para realizar control sobre el tráfico de datos o simplemente para monitorearlos, la secuencia de comandos también especifica el filtrado y el proceso de recopilación de los datos experimentales y genera un esquema de base de datos para soportar el análisis posterior de esos datos.

Figura 7. Interacción, usuario-Orbit Lab³⁹



³⁹ Orbit Lab [sitio web], interacción usuario-Orbit Lab [Consultado 21 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/attachment/wiki/Software/OMF-User-View.png>

4. BANCO DE PRUEBAS REALIZADO EN LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

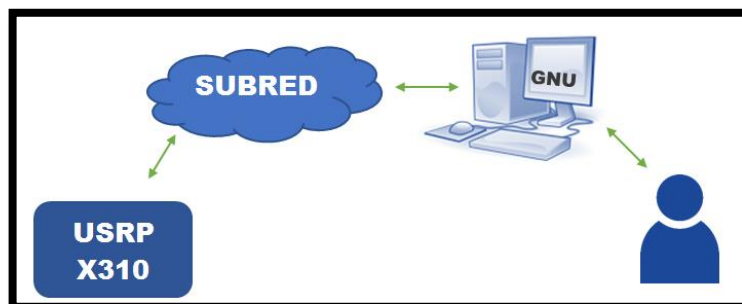
En este capítulo se presenta la descripción de las pruebas realizadas de manejo remoto de los dispositivos de SDR disponibles en la Universidad Industrial de Santander. Dichas pruebas se llevaron a cabo con la finalidad de verificar el acceso, la funcionalidad y el alcance de estos dispositivos de manera remota. Los resultados de las pruebas se presentan en el capítulo 5.

4.1 ACCESO REMOTO A DISPOSITIVOS USRP BAJO LA MISMA LAN

Se ejecutaron pruebas de acceso remoto a nivel LAN para cada uno de los dispositivos disponibles (USRP1, USRP N210, USRP X310). Las pruebas fueron diseñadas y llevadas a cabo acorde a las características de cada equipo.

4.1.1. USRP X310. Esta prueba consistió en el acceso remoto a un USRP X310 desde un computador conectado a la misma subred, como se muestra en la Figura 8. Se llevó a cabo con la finalidad de verificar la viabilidad de control del dispositivo sin el requerimiento de una conexión física con el computador.

Figura 8. Esquema de conexión remota al USRP X310 en la misma subred



Los pasos necesarios para realizar esta conexión son:

- 1- Se conecta el USRP X310 a la LAN mediante el puerto Gigabit Ethernet.
- 2- Se debe tener una versión de UHD igual o superior a la 3.7.3, la cual permite manipular el USRP X310 (ver Anexo 1).
- 3- En el computador mediante el cual se accede remotamente al radio (también conectado a la LAN mediante Ethernet) se abre una terminal y se ejecutan los siguientes comandos con la finalidad de cambiar la dirección IP por una que se encuentre disponible.

```
cd usr/local/lib/uhd/utils

./usrp_burn_mb_eeprom -args=""addr=192.168.xxx.xxx" --
values="ip-addr0=192.168.xxx.xxx"
```

Donde la IP que se pone en los argumentos (`-args`), es la que tiene actualmente el dispositivo y la que se pone en valores (`--values`) es la que queremos que tenga.

- 4- Se reinicia el USRP X310 para guardar las modificaciones.
- 5- Se realiza un *ping* a la nueva dirección para probar que el equipo se encuentre encendido y conectado a la LAN.
- 6- Se ejecuta el siguiente comando en la terminal:

```
uhd_usrp_probe -args="addr=192.168.xxx.xxx"
```

Con la finalidad de que confirmemos que está reconociendo el dispositivo de esa IP como un USRP X310. Como respuesta debemos obtener las especificaciones del dispositivo (Figura 9).

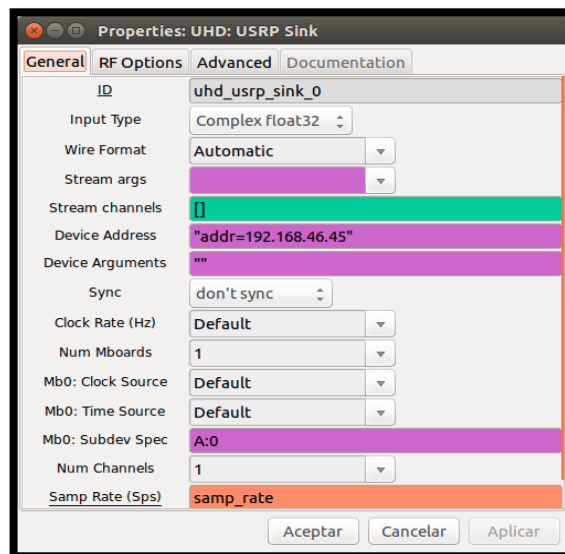
Figura 9. Especificaciones del USRP X310

```
Mboard: X310
revision: 6
product: 30410
mac-addr0: 00:80:2f:0a:fa:1d
mac-addr1: 00:80:2f:0a:fa:1e
gateway: 192.168.45.1
ip-addr0: 192.168.45.45
subnet0: 255.255.255.0
ip-addr1: 192.168.20.2
subnet1: 255.255.255.0
ip-addr2: 192.168.30.2
subnet2: 255.255.255.0
ip-addr3: 192.168.45.45
subnet3: 255.255.255.0
serial: F58EC2
FW Version: 4.0
FPGA Version: 19.0
```

Donde la dirección “ip-addr0” corresponde a la dirección IP del puerto Gigabit Ethernet y las direcciones “ip-addr2” e “ip-addr3” corresponden al puerto 10 Gigabit Ethernet.

- 7- El USRP X310 ya se encuentra listo para ser usado de manera remota bajo la misma LAN. Entonces, ingresamos a GNU-Radio y en los bloques de *UHD Source* ó *UHD Sink* (depende la aplicación), en la casilla de *Device Address* ingresamos la dirección IP asignada al equipo de la siguiente manera: “addr=192.168.xxx.xxx” (Figura 10).

Figura 10. Acceso al USRP X310 de manera remota en GNU-Radio



Aunque el dispositivo se encuentre conectado a una LAN, al estar conectado a una *subred*, no permite manipularlo desde una diferente si no se cumplen las siguientes especificaciones:

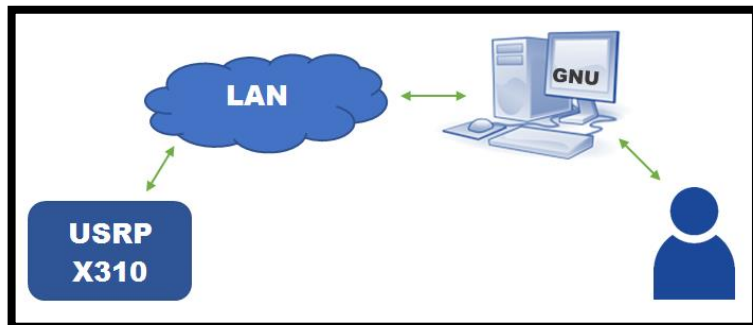
- 1- La versión de Ubuntu del computador desde donde se hace el acceso debe ser una versión LTS, debido a que algunos paquetes requeridos por el software no se encuentran disponibles para otras versiones.
- 2- La puerta de enlace (*Gateway*) del dispositivo debe corresponder a la *subred* a la que se encuentra conectado, a manera de ejemplo, si el dispositivo se encuentra conectado a la *subred* número 94 con una IP 192.168.94.xxx, la puerta de enlace debe ser 192.168.94.1. Este parámetro se modifica con los siguientes comandos ejecutados en la terminal de Ubuntu:

```
cd usr/lib/uhd/utils

./usrp_burn_mb_eeprom -args=""addr=192.168.xxx.xxx" --
values="gateway=192.168.xxx.xxx"
```

En la Figura 11 se observa el esquema de conexión remota al USRP X310 logrado al cumplir los requerimientos y la configuración descrita anteriormente.

Figura. 11 Esquema de conexión remota al USRP X310 en la misma LAN



Adicionalmente, si se realizan todos los pasos anteriores de manera puntual y aun así el acceso no es permitido porque se requiere una versión de *firmware* o de imagen de la FPGA diferente, se debe realizar una actualización de la imagen de la FPGA incorporada en el dispositivo, la cual se lleva a cabo conectando directamente el USRP X310 a cualquier computador con Ubuntu y ejecutando el siguiente comando en la terminal:

```
uhd_image_loader --args="addr=192.168.xxx.xxx,fpga=HG"
```

De manera similar, existe la posibilidad de que al modificar los parámetros de la EEPROM se hagan cambios indebidos y sea necesario reprogramarla. En ese caso el error es el siguiente:

```
Error: RuntimeError: No revision compat detected. MB EEPROM must  
be reprogrammed!
```

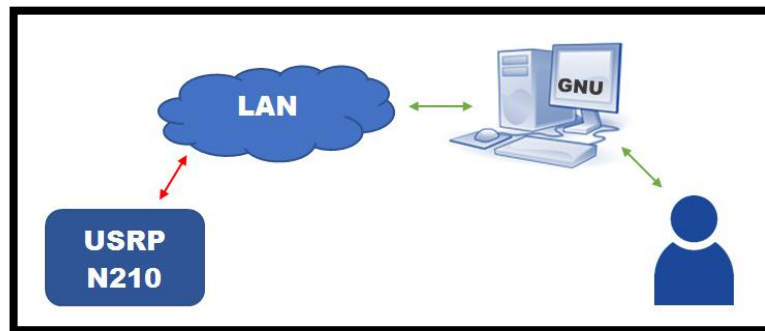
Para solucionarlo es necesario conocer el número de revisión del dispositivo, el cual se encuentra en un *sticker* en la *Motherboard* cerca del conector AUX I/O (es necesario destapar el dispositivo). A manera de ejemplo, el número visualizado en la *motherboard* del USRP X310 usado en el presente proyecto es el "154573F-02L", donde el número de revisión lo dice la letra antes del "-" y es la número 6 (C=3, D=4, E=5, F=6). Cuando se conozca el número de la revisión, la solución del problema se lleva a cabo ejecutando los siguientes comandos en la terminal de Ubuntu:

```
cd usr/local/lib/uhd/utils  
  
./usrp_burn_mb_eeprom --  
args="recover_mb_eeprom,addr=192.168.46.45" --values="revision=6"
```

4.1.2. USRP N210. Esta prueba se intentó llevar a cabo de igual forma que con el USRP X310, conectando el dispositivo directamente a la LAN y accediendo a él con un computador conectado a la misma (Figura 12). Sin embargo, el dispositivo presenta el inconveniente de que no es viable para realizar trabajo de forma independiente o al conectarse directamente a una red LAN; las características de hardware hacen que siempre sea necesaria la

presencia de un computador o un dispositivo que lo simule para usar el equipo. Por lo anterior, manejar remotamente el USRP N210 es viable únicamente si se cuenta con computador (o un dispositivo que lo simule) que tenga dos puertos de conexión Ethernet, uno para establecer la conexión directa a la LAN y el otro para comunicarse con el dispositivo y operarlo.

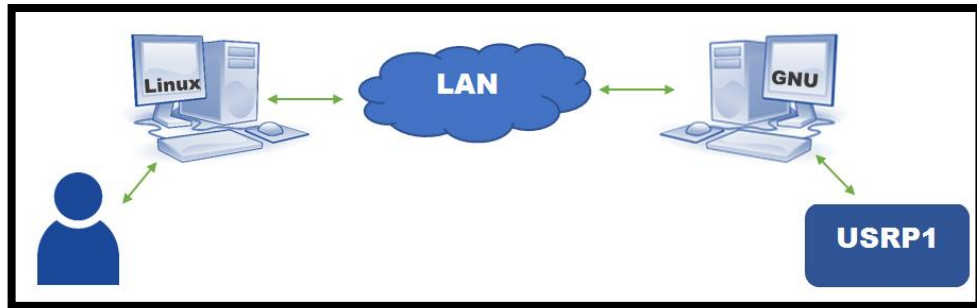
Figura 12. Esquema de conexión remota al USRP N210 en la misma LAN



4.1.3. USRP1. Este dispositivo pertenece a la serie Bus de Ettus Research, por lo que no cuenta con una tarjeta de red sino un puerto de conexión USB. Razón por la cual, al igual que con el USRP N210 es necesaria la presencia de un computador para su operación.

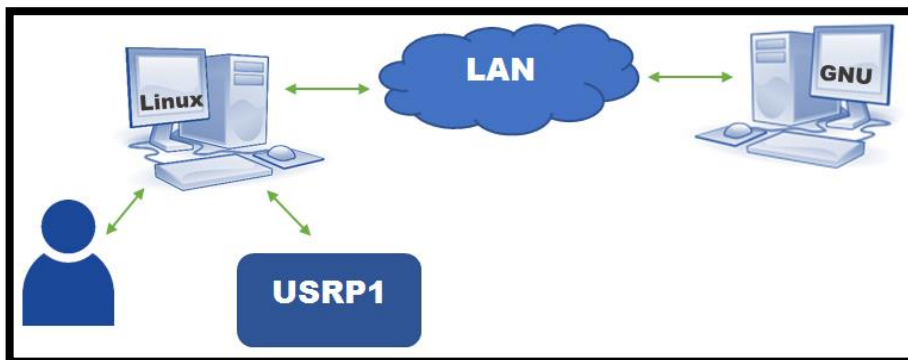
Prueba 1: La prueba mostrada en la Figura 13 se realizó con el desarrollo de una máquina virtual, en la que se instaló GNU-Radio y se conectó el dispositivo; para lograr la manipulación remota del dispositivo desde un computador que solamente contaba con Ubuntu 16.04 (funciona en cualquiera de sus versiones) y conexión a la LAN, se estableció una conexión a la máquina virtual por SSH consiguiendo así acceso al software y al dispositivo.

Figura 13. Esquema de conexión remota al USRP1 en la misma LAN (usuario - software, USRP1)



Prueba 2: En la Figura 14 se observa el esquema de conexión que se llevó a cabo en el desarrollo de la segunda prueba de acceso remoto con el USRP1, en este escenario el usuario y el dispositivo se encuentran en el mismo computador conectado a la LAN sin el software necesario para manipular el radio, dicho software debía ser suministrado por otro computador conectado de igual forma a la LAN.

Figura 14. Esquema de conexión remota al USRP1 en la misma LAN (usuario, USRP1- software)



El software fue puesto en la LAN con una máquina virtual, de manera similar a lo ejecutado en la prueba 1. Para lograr el control del USRP1 accediendo remotamente al software, en el computador donde se encontraba el usuario se accedió por SSH a la máquina virtual y se le compartió el puerto USB al que se encontraba conectado el USRP1. A continuación se presentan las instrucciones para compartir virtualmente el puerto de un computador a otro:

- 1- Se instala el servidor en el computador donde se encuentra el puerto que se desea compartir ejecutando el siguiente comando en la terminal de Ubuntu:

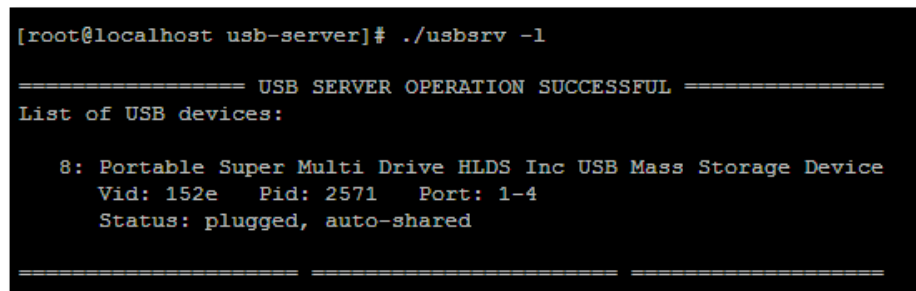
```
./installer.sh install-server
```

- 2- En el mismo computador se busca el número del puerto USB que queremos compartir, ejecutando en la terminal de Ubuntu el comando:

```
./usbsrv -l
```

Donde obtendremos la lista de todos los dispositivos USB conectados al computador y cada uno con un número asociado (Ver ejemplo en la Figura 15).

Figura 15. Respuesta al comando “./usbsrv -l” en la terminal de Ubuntu del servidor



```
[root@localhost usb-server]# ./usbsrv -l
===== USB SERVER OPERATION SUCCESSFUL =====
List of USB devices:

  8: Portable Super Multi Drive HLDS Inc USB Mass Storage Device
    Vid: 152e  Pid: 2571  Port: 1-4
    Status: plugged, auto-shared
=====
```

- 3- Se identifica cual es el puerto que queremos compartir y lo hacemos con el comando:

```
./usbsrv -share #
```

Donde #, es número del puerto asociado al dispositivo que queremos compartir (8 en el caso del ejemplo mostrado en la Figura 15).

En este punto ya se está compartiendo el puerto al que se encuentra conectado el USRP1.

- 4- Desde el computador donde se desea acceder al puerto compartido, se crea el cliente mediante la ejecución del siguiente comando en la terminal de Ubuntu:

```
./installer.sh install-client
```

- 5- En el mismo computador se crea la conexión mediante IP con el servidor con el siguiente comando:

```
./usbclnt -addserver 192.168.xxx.xxx
```

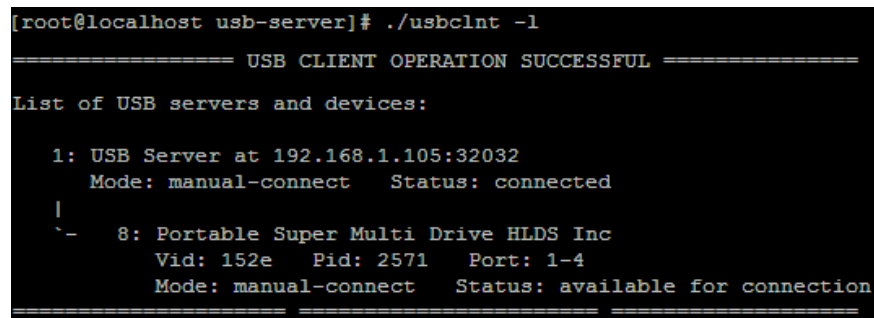
La dirección IP que se especifica en el comando es la del servidor.

- 6- El cliente ya debe reconocer los dispositivos que el servidor esté compartiendo, para verificarlo se ejecuta el comando:

```
./usbclnt -l
```

Donde debe aparecer algo similar a lo mostrado a manera de ejemplo en la Figura 16.

Figura 16. Respuesta al comando “./usbclnt -l” en la terminal de Ubuntu del cliente



```
[root@localhost usb-server]# ./usbclnt -l
===== USB CLIENT OPERATION SUCCESSFUL =====
List of USB servers and devices:
  1: USB Server at 192.168.1.105:32032
     Mode: manual-connect   Status: connected
  |
  `-- 8: Portable Super Multi Drive HLDS Inc
        Vid: 152e   Pid: 2571   Port: 1-4
        Mode: manual-connect   Status: available for connection
=====
```

- 7- Para establecer la conexión con el USRP1 desde el cliente solo resta ejecutar el comando:

```
./usbclnt -connect a-b
```

Donde “a” es el número de conexión al servidor y “b” es el número del puerto asociado al servidor (en el caso del ejemplo mostrado en la Figura 16, a=1 y b=8).

Para manipular el dispositivo desde el software GNU-Radio o para ver sus especificaciones técnicas desde la terminal de Ubuntu, es necesario tener permisos de súper usuario en el cliente.

Para tener permisos de súper usuario desde la terminal se ejecuta:

```
sudo su
```

Allí se solicitará la contraseña del computador (cliente) y se adjudican los permisos requeridos. En este punto ya es posible ejecutar comandos de *uhd* que reconocerán el USRP1 conectado en el servidor (*uhd_find_devices*, *uhd_usrp_probe*, etc).

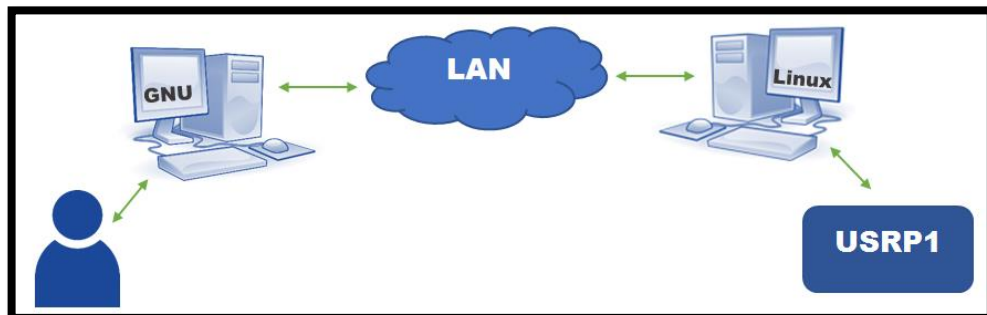
Para usar el software de GNU-Radio en el cliente como súper usuario, se ejecuta en la terminal el siguiente comando:

```
sudo gnuradio-companion
```

El software se ejecuta de manera común, la única diferencia es que reconoce los dispositivos compartidos desde el servidor y son manipulables.

Prueba 3: En la Figura 17 se observa el esquema de conexión que se llevó a cabo en el desarrollo de la tercera prueba de acceso remoto con el USRP1, en este escenario el usuario y el software (GNU-Radio) se encuentran en el mismo computador y el USRP1 se encuentra conectado a otro computador que cuenta únicamente con Ubuntu 16.04, ambos computadores cuentan con conexión a la LAN.

Figura 17. Esquema de conexión remota al USRP1 en la misma LAN (usuario, software – USRP1)



Los pasos para llevar a cabo esta prueba son iguales a los de la prueba número 2, difieren entre sí en que los roles de servidor y cliente se invierten. El servidor siempre será el computador al que se encuentra conectado el dispositivo que se quiere compartir (USR1 en este caso).

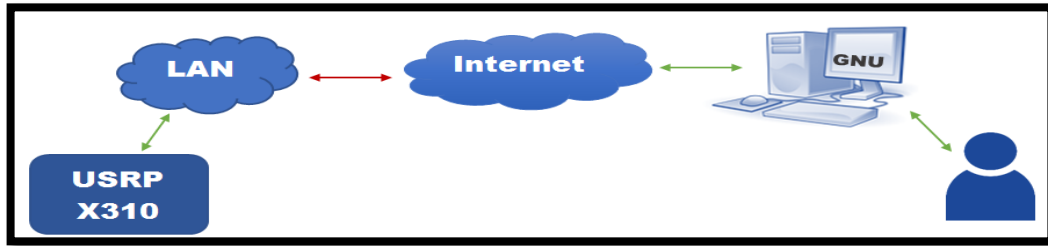
4.2 ACCESO REMOTO A DISPOSITIVOS USRP MEDIANTE INTERNET

Las pruebas de acceso remoto mediante internet, se llevaron a cabo con los dispositivos que superaron la prueba de acceso remoto en la LAN (USR1 X310 y USRP1), con la finalidad de ampliar el rango de acceso a ellos. Al igual que con las pruebas realizadas anteriormente, cada dispositivo tuvo un diseño específico de pruebas según sus características.

4.2.1. USRP X310. El control del USRP X310 mediante internet (Figura 18) no fue posible, se intentó tener la posibilidad de manipularlo de dos maneras diferentes:

- Se asignó una dirección IP pública al dispositivo y realizando el mismo proceso de conexión que se llevó a cabo exitosamente dentro de la LAN. Como resultado se obtuvo un error de *firmware* que no permitía ver las especificaciones técnicas del USRP X310 ni mucho menos manipularlo.
- Se ingresó a la LAN mediante una conexión VPN simulando el escenario que permitió el control del USRP X310 dentro de la LAN de manera exitosa. Como resultado se obtuvo una respuesta a un *ping* de la dirección IP correspondiente al radio, la cual mostraba que había un dispositivo operando en ella. Sin embargo, no se reconoció como un dispositivo USRP, negando la posibilidad de ver las especificaciones técnicas del equipo y de manipularlo.

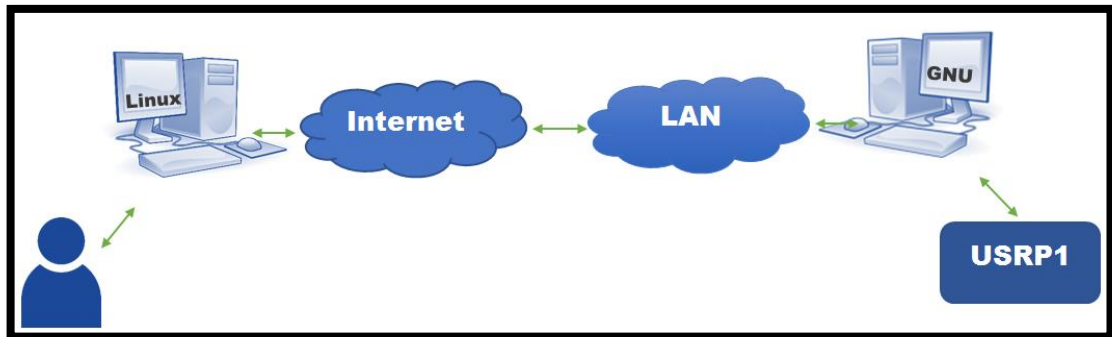
Figura 18. Esquema de conexión remota al USRP X310 mediante internet



4.2.2. USRP1. Al igual que en las pruebas previas realizadas con este equipo dentro la LAN, se establecieron los mismos tres (3) escenarios de manipulación remota desde internet. Sin embargo, solo se llevaron a cabo dos (2) de ellos debido a que la viabilidad del tercer escenario (Figura 21) era deducible de manera análoga al segundo (Figura 20).

En la prueba 1 (Figura 19) se ingresó a la LAN mediante una conexión VPN, posteriormente se realizó una conexión por SSH a la máquina virtual en donde se encontraba conectado el equipo e instalado el software, para así permitir la manipulación remota del USRP1.

Figura 19. Esquema de conexión remota al USRP1 mediante internet (usuario - software, USRP1)



En la prueba 2, al igual que en la prueba 1 se ingresó a la LAN mediante una conexión VPN. De allí en adelante, todo el proceso de compartir el puerto al que se encuentra conectado el USRP1 en el computador del usuario, es igual al descrito en el inciso anterior (4.1.2 - prueba 2 - Figura 14).

Figura. 20 Esquema de conexión remota al USRP1 mediante internet (usuario, USRP1 - software)

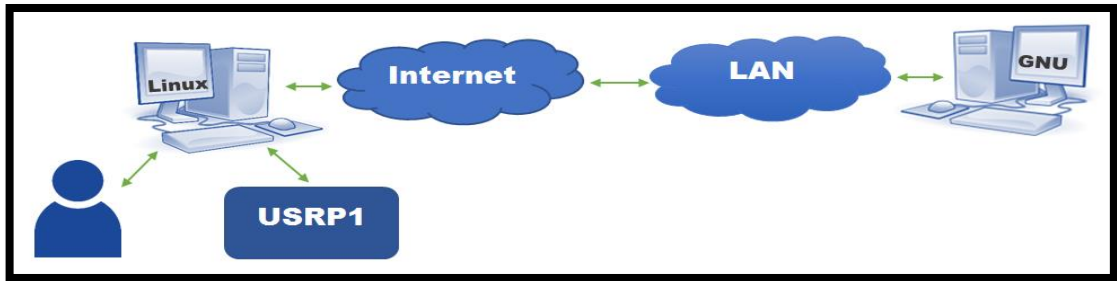
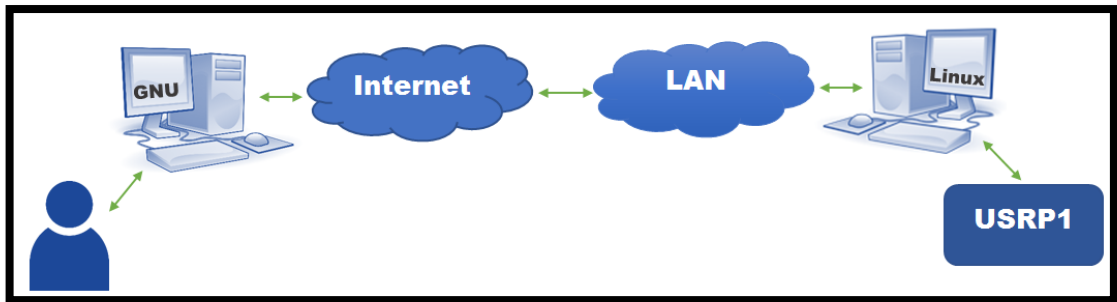


Figura. 21 Esquema de conexión remota al USRP1 mediante internet (usuario, software - USRP1)



5. RESULTADOS DE LAS PRUEBAS REALIZADAS EN LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

El banco de pruebas realizado en la Universidad Industrial de Santander, arrojó resultados favorables en su mayoría. Se presentaron limitantes en algunas configuraciones debido a que las características de ciertos equipos no permiten el tipo de conexión deseado o no era estable por factores externos (calidad de conexión a internet, recursos de los equipos y herramientas usadas).

A continuación, se presentan los resultados obtenidos al llevar a cabo cada una de las pruebas.

5.1 USRP X310

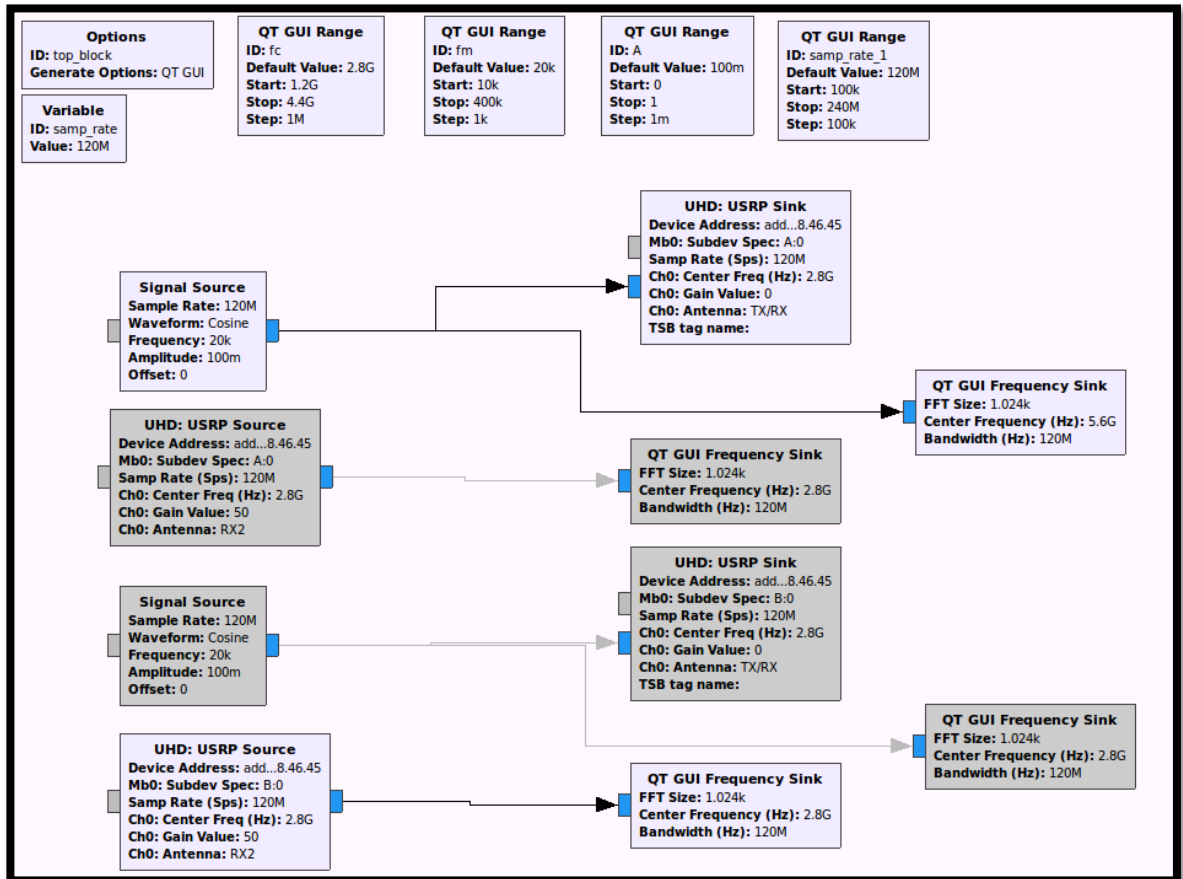
Para las pruebas realizadas con el USRP X310 se ejecutó el diagrama de bloques mostrado en la Figura 22, en donde se transmite por una *daughterboard* (SBX-120) y se recibe por la otra (CBX-120). Como ambas *daughterboard* tienen una frecuencia de operación diferente, el parámetro f_c ubicado en un control tipo *slider* está delimitado por los valores que son comunes entre dichas frecuencias. De manera similar, el *slider* del *samp_rate_1* varía con un valor tope determinado por el ancho de banda de las *daughterboards* (120MHz, el mismo para ambas), esta variable buscaba exigir en rendimiento al radio y observar la velocidad de respuesta en condiciones remotas.

Los resultados de la prueba dentro de la LAN (Figura 11) fueron satisfactorios, al cambiar los parámetros de la señal enviada o al trasladarla a través del espectro radioeléctrico, el retardo no era perceptible en la recepción (exceptuando el caso en el que la conexión a la red no era estable).

Las pruebas realizadas mediante la conexión a internet (Figura 18) con este dispositivo no pudieron ejecutarse de manera satisfactoria. Lo anterior debido a que al momento de asignarle una IP pública al equipo e intentar acceder a él como se hacía mediante la LAN,

no se visualizaban las especificaciones técnicas del mismo; en algunas ocasiones (haciendo exactamente lo mismo) se reconocía que era un USRP X310, pero tampoco permitía manipularlo por problemas aparentemente de *firmware*.

Figura 22. Diagrama de bloques para el USRP X310



5.2 USRP1

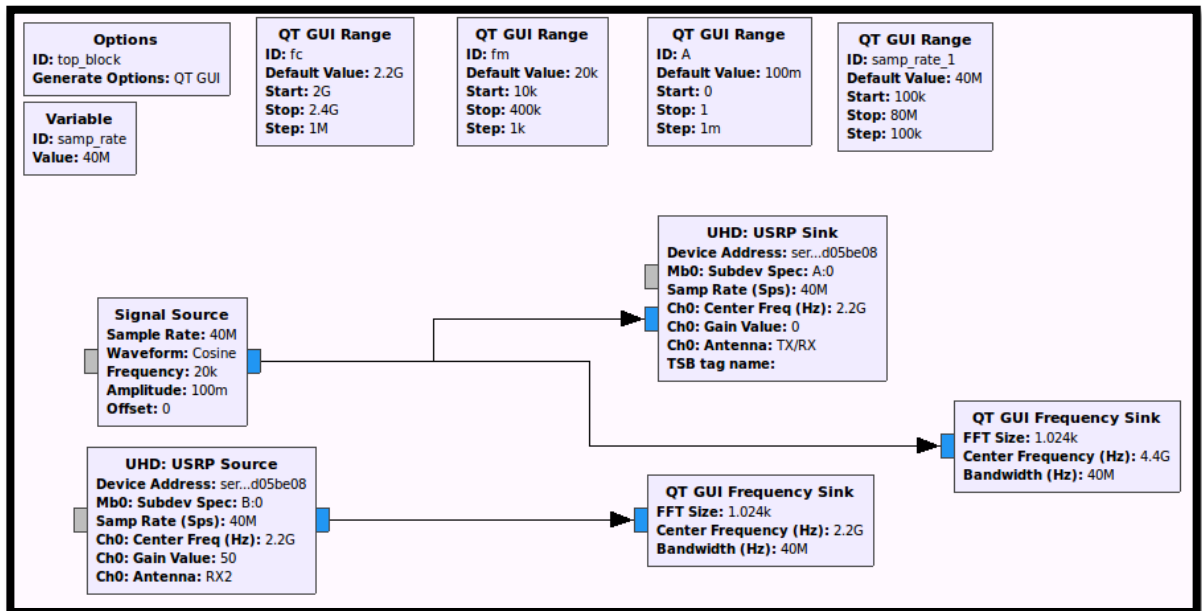
Las pruebas realizadas dentro de la LAN con este dispositivo (Figura 13, 14 y 17) arrojaron resultados satisfactorios siempre que la conexión a internet se mantuvo estable. Esta variable es primordial para el rendimiento y la eficiencia de la máquina virtual, que es quien alberga el software y realiza el procesamiento.

De manera similar a las pruebas realizadas con el USRP X310, se diseñó un diagrama de bloques acorde a las especificaciones técnicas del USRP1 y que exigiera los recursos disponibles en él, dicho diagrama se observa en la Figura 23. Las *daughterboard* de este equipo son de la misma referencia (RFX2200), por lo tienen un rango de frecuencia de operación igual y la variable fc abarca la totalidad de dicho rango. Por otro lado, el máximo valor que toma la variable $samp_rate_1$ es el doble del ancho de banda de las *daughterboards*, el cual tiene un valor de 40MHz.

En los escenarios de conexión puestos a prueba para el USRP1 (mediante la LAN y a través de internet), se encontró que el control del equipo de manera remota es posible en cualquiera de ellos; siempre y cuando se cuente con una buena conexión a internet y una máquina virtual con los recursos suficientes para soportar el procesamiento de GNU-Radio.

La máquina virtual que se usó para el banco de pruebas consta de 4 procesadores y 4 GB de RAM, el rendimiento era limitado cuando la conexión a internet era inestable o el procesamiento demandaba más recursos de los disponibles. El control del radio era de respuesta inmediata para diagramas de bloques de baja complejidad (*Signal source*, *UHD: USRP sink* y *sliders*), pero para sistemas más complejos el software se detenía de manera abrupta por escases de recursos.

Figura 23. Diagrama de bloques para el USRP1



6. LABORATORIO REMOTO DE COMUNICACIONES BASADO EN SDR EN LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

La Universidad Industrial de Santander en la actualidad cuenta con equipos destinados al trabajo en el área de las telecomunicaciones, los cuales pueden hacer parte de un laboratorio remoto basado en SDR. En el presente capítulo se listarán dichos equipos y se explicará la funcionalidad de aquellos que no se han nombrado anteriormente, así como también se mencionará cuál es crecimiento más adecuado en términos de equipos para la institución en esta línea de trabajo.

6.1 INVENTARIO ACTUAL DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

Actualmente, los equipos y componentes con los que cuenta la institución son:

- **8 USRP Ettus-N210**
- **2 USRP1**
- **1 USRP Ettus-X310**
- **1 USRP Ettus-E310**
- **Analizador vectorial de señales**
-

Permite medir de señales RF, realizar funciones digitales útiles de demodulación, medir señales rápidas de banda ancha o espectro extendido, analizar y generar señales más complejas que las que permite un instrumento escalar y mostrar espectrogramas en 3D.

4 Antenas LP0410: Cuentan con un conector SMA, frecuencia de trabajo de 400 MHz a 1 GHz, ganancia de 5-6 dBi y son compatibles con cualquier *daughterboard* que funcione dentro del mismo rango de frecuencias.

4 Antenas LP0965: Cuentan con un conector SMA, frecuencia de trabajo de 850 MHz a 6.5 GHz, ganancia de 5-6 dBi y es compatible con cualquier *daughterboard* que funcione dentro del mismo rango de frecuencias.

4 Antenas VERT400: Son antenas verticales omnidireccionales tri-banda de 144 MHz, 400 MHz y 1200 MHz, con una potencia máxima de 10 W, una longitud de 17 cm y un conector SMA.

4 Antenas VERT900: Son antenas verticales omnidireccionales de banda cuádruple, 824 a 960 MHz, 1710 a 1990 MHz, con una ganancia de 3dBi.

4 Antenas VERT2450: Son antenas verticales de banda dual de 2,4 a 2,48 GHz y de 4,9 a 5,9 GHz, con una ganancia de 3dBi.

1 Loop Back cable kit: Es un *kit* para USRP, incluye 2 cables SMA-M a SMA-M (60cm / 2ft) y 2 atenuadores SMA-F a SMA-M (30dB, 50Ohm, DC-6GHz).

2 cables MCX-Bulkhead: Incluye un conector de *bulkhead* MCX-M a SMA-F para conectar con el WBX y el DBSRX2.

2 BasicTX Daughterboard 1-250 MHz Tx: Utiliza dos transformadores de banda ancha para que coincidan con las salidas del DAC dual de un USRP a 50 Ohm, conexiones SMA, posee una capacidad de transmisión desde el USRP de 1 a 250MHz.

2 BasicRX Daughterboard 1-250 MHz Rx: Tiene acceso directo a las entradas ADC, no incluye un oscilador local y puede aceptar señales en modo real de 1 a 250MHz.

2 CBX 1200-6000 MHz Rx/Tx (40 MHz): Es un duplex completo, es un transceptor que cubre la banda de frecuencia de 1.2Ghz a 6Ghz, tiene un ancho de banda de 40Mhz. Los osciladores locales para las cadenas de recepción y transmisión funcionan de forma independiente, lo que permite un funcionamiento multi-frecuencia.

2 SBX 400-4400 MHz Rx/Tx (40 MHz): Tiene un transceptor que proporciona hasta 100mW de potencia de salida y una cifra de ruido típica de 5dB. Los osciladores locales para las cadenas de recepción y transmisión funcionan de forma independiente, permitiendo la operación de banda dual. Adicionalmente, permite el trabajo con sistemas MIMO y proporciona 40MHz de ancho de banda.

1 WBX 50-2200 MHz Rx/Tx (120 MHz): El WBX-120 es un dúplex completo de banda ancha, cubre la banda de frecuencia de 50MHz a 2,2GHz, tiene un ancho de banda de 120Mhz. Los osciladores locales para las cadenas de recepción y transmisión funcionan de forma independiente, permitiendo la operación de frecuencia dúplex.

1 CBX 1200-6000 MHz Rx/Tx (120 MHz): Es un dúplex completo de banda ancha transceptor que cubre la banda de frecuencia de 1.2Ghz a 6Ghz, cuenta con un ancho de banda de 120Mhz. Los osciladores locales para las cadenas de recepción y transmisión funcionan de forma independiente, permitiendo un funcionamiento multi-frecuencia.

1 SBX 400-4400 MHz Rx/Tx (120 MHz): Es un dúplex de banda ancha transceptor que cubre una banda de frecuencias de 400Mhz a 4,4Ghz y tiene un ancho de banda de 120Mhz.

6.2 CRECIMIENTO SUGERIDO PARA LA IMPLEMENTACIÓN DE UN LABORATORIO REMOTO BASADO EN SDR EN LA UIS

Teniendo como referente principal para esta sección el Wirid Lab de la UMNG, por ser un laboratorio remoto de SDR ya implementado en Colombia y estar basado en Orbit Lab, se sugiere la adquisición de los siguientes equipos como complemento a los ya existentes en el inventario de la UIS:

Servidor: Es requerido con la finalidad realizar la gestión de datos, recursos y usuarios en el laboratorio. Al ser asignado a funciones de virtualización, es necesaria una amplia memoria RAM (entre las 24 y 32 GB), 8 núcleos y 1 TB de almacenamiento como mínimo.

Computadores con dos tarjetas de red: Los cuales son necesarios para poder llevar a cabo el control remoto de los USRP NI2920, los cuales son el recurso más numeroso de SDR que tiene la Universidad Industrial de Santander. Dichos computadores deben tener y soportar el sistema operativo Ubuntu (16.04 en adelante, preferiblemente).

Generador vectorial de señales: Un generador de señales de banda base, RF y microondas brinda calidad de señal, flexibilidad y es de fácil de manejo. Estos equipos también ofrecen amplios rangos de frecuencias hasta 43,5 GHz (hasta 170 GHz con multiplicadores de frecuencia), anchos de banda de modulación de hasta 2 GHz, y son compatibles con todos los estándares digitales de comunicaciones móviles e inalámbricas.

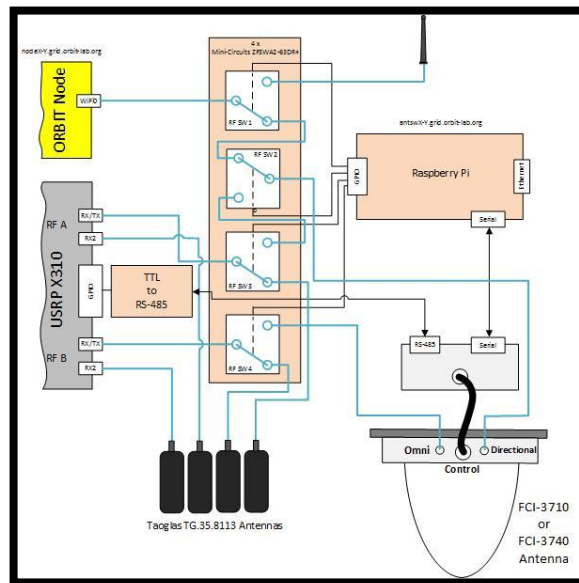
Sistema de respaldo y control de energía: Esto es importante porque al prestar servicios remotos disponibles las 24 horas del día, se pueden presentar fallas eléctricas que cortarían la oferta de recursos y como consecuencia se genera inconformidad en los usuarios y se ponen en riesgo algunos equipos. Además, en algunas oportunidades puede ser necesario el reinicio de los dispositivos con fines de operación y actualización. Buscando suplir las necesidades nombradas anteriormente, se recomienda la adquisición de al menos dos Smart-UPS 750-3000 VA 120/208/230 V de CA de montaje en armario, pues cada una de ellas cuenta con hasta 8 tomacorrientes controlables mediante un puerto USB o una tarjeta de administración de red.

El sistema Smart-UPS, ofrece protección para equipos electrónicos contra apagones totales y parciales del suministro de alimentación de la red eléctrica, caídas de tensión y sobretensiones. Además, suministra alimentación de reserva de batería hasta que el suministro de alimentación de la red eléctrica se restablece a niveles seguros o hasta que las baterías se descargan totalmente.

Switch de antenas: El uso de las antenas varía dependiendo el rango de frecuencia en el que se va a trabajar en cada práctica, por lo que sería ideal tener un sistema que realice el intercambio entre éstas sin tener que entrar en contacto directo. Lo anterior, con la finalidad de prolongar la vida útil de los equipos que se pueden deteriorar por la manipulación inadecuada de sus puertos.

Orbit Lab usa para este requerimiento el sistema mostrado en la Figura 24, en donde la antena es gestionada directamente por el USRP a través de un enlace GPIO-RS485 rápido o a través de servicios OMF que facilitan la selección del estado de potencia y la selección del patrón. Este hardware de conmutación de antenas consiste en: un banco de interruptores de RF, controladores RS-485 y Raspberry PI como control embebido controlado. Este hardware permite la selección remota de la antena y la dirección.

Figura. 24 Switch de antenas implementado por Orbit Lab⁴⁰



Filtros: De distinta naturaleza y que representarían una utilidad para distintas aplicaciones en las que se encuentre involucrado el manejo de bandas de frecuencias, tanto de transmisión como de recepción.

Amplificadores: Para cuando se desee una señal con mayor potencia a la que posee, con la finalidad de mejorar la claridad del mensaje recibido o transmitir señales que sean de fácil captación para el receptor.

⁴⁰ Orbit Lab [sitio web], *switch* de antenas [Consultado 24 de abril de 2017]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/IDirAnt#Antennas>

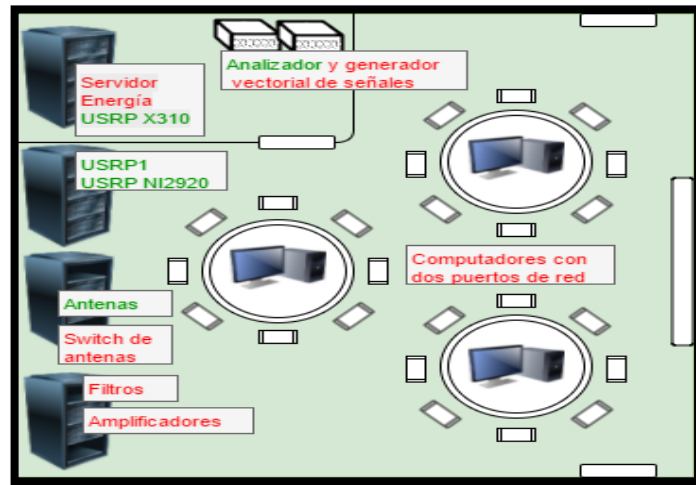
6.3 LABORATORIO REMOTO DE COMUNICACIONES DE SDR SUGERIDO PARA LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

En la Figura 25 se muestra una propuesta de laboratorio remoto de comunicaciones de SDR acorde al inventario con el que cuenta actualmente la Universidad Industrial de Santander y a las sugerencias realizadas en la sección 6.2. Los equipos que ya se encuentran disponibles para la implementación tienen su nombre en verde, mientras que los que aún no hacen parte del inventario de la UIS tienen su nombre en rojo.

En la parte interior del laboratorio hay una oficina que se encuentra bajo llave, en donde se encuentran en estantes los dispositivos de mayor valor y los encargados de administrar el suministro de potencia de la mayoría de los equipos del laboratorio (los que hacen parte del laboratorio remoto). Dentro de los equipos allí ubicados se encuentran los USRP X310, el analizador y el generador vectorial de señales, el servidor y las Smart-UPS.

En la parte externa a la oficina del laboratorio se encuentran los estantes con los USRP NI2920 y los USRP1, así como también se encuentran las antenas, el sistema de *switch* de las mismas, los filtros y amplificadores. Los computadores con doble tarjeta de red están distribuidos en las mesas ubicadas en la sala y la conexión entre éstos y los radios se realiza mediante un cableado en techo.

Figura. 25 Propuesta de laboratorio remoto de comunicaciones de SDR en la UIS.



7. CONCLUSIONES

La implementación de un laboratorio remoto de comunicaciones basado en SDR en la Universidad Industrial de Santander es un avance significativo en el trabajo institucional tanto para estudiantes como para investigadores del área. Brinda beneficios de accesibilidad, prolonga la vida útil a los equipos de laboratorio y presta servicios de software para quienes lo requieran. Aunque no todos los dispositivos USRP disponibles en la Universidad Industrial de Santander permiten la operación remota de forma independiente, es posible manipular remotamente cada uno de estos equipos con la configuración adecuada y los complementos de software y hardware que cada uno de ellos requiera.

El control remoto del USRP NI2920 es importante para el desarrollo de un laboratorio remoto de comunicaciones en la Universidad Industrial de Santander, siendo éste el recurso más numeroso de su clase en el inventario de la institución. Por lo anterior y para potenciar el uso de los recursos, es necesaria la adquisición de computadores con dos tarjetas de red, pues éste es el requerimiento de estos dispositivos para permitir su manipulación remota.

Es indispensable adecuar un servidor para gestionar todos los servicios que brinda un laboratorio de comunicaciones remoto de SDR. Facilita la accesibilidad a los equipos de forma remota, coordina el tráfico de usuarios y permite una distribución de recursos y equipos acorde a las demandas.

Es posible trabajar de manera local con un USRP que se encuentre configurado para manipularse de manera remota, por lo que este tipo de laboratorio remoto no presenta ninguna interferencia para el trabajo en el laboratorio físico, siempre y cuando no se realice de manera simultánea.

La conexión remota a los dispositivos desde Internet más adecuada, es acceder por VPN a la LAN y posteriormente ingresar al servidor mediante una conexión SSH a donde se encuentran los equipos físicos y el software de GNU-Radio. Lo anterior, debido a que no es seguro asignarles IP's públicas a los equipos y dejarlos expuestos abiertamente en Internet.

Es importante potencializar el manejo de las *daughterboards* disponibles en la Universidad Industrial de Santander, con la finalidad de prolongar la vida útil de éstas y los demás componentes que intervienen en la práctica. Para lograr lo anterior es necesario limitar el contacto físico de los usuarios con los equipos sin que se presente ninguna restricción de uso, por lo que se requiere la implementación de una configuración que las mantenga fijas en los dispositivos USRP, accediendo a ellas mediante la LAN y en donde el usuario conoce las especificaciones de cada una de ellas y puede decidir cuál es más conveniente para su requerimiento (frecuencia de trabajo, ancho de banda, etc).

Los equipos por adquirir para tener un laboratorio remoto de comunicaciones basado en SDR en la Universidad Industrial de Santander son complemento de los ya existentes, lo que quiere decir que el inventario actual de la institución en el área de las telecomunicaciones puede llegar a trabajar como una sola herramienta de múltiples funciones y servicios. Dentro de los cuales, se puede llegar al punto de que el gasto generado por la Universidad Industrial de Santander sea visto como una inversión, por la prestación de servicios y/o equipos a otras universidades o entidades del estado que requieran esta tecnología de punta para sus propios fines.

8. TRABAJO FUTURO

A corto plazo, se debe llevar a cabo la implementación de lo plasmado en el presente documento en el laboratorio de comunicaciones de la Universidad Industrial de Santander, brindando así la posibilidad de realizar prácticas pedagógicas y con fines investigativos de manera remota. También es primordial suplir los elementos esenciales para este tipo de laboratorio que no se encuentran disponibles actualmente, con la finalidad de que el escenario de trabajo sea amplio, logrando acceder a todos los radios de manera remota (especialmente al USRP NI2920).

A mediano plazo, es importante potenciar el uso del laboratorio para varias aplicaciones pertenecientes a las tecnologías inalámbricas, la conmutación y enrutamiento de protocolos de red, entre otros.

A largo plazo y en un caso ideal, el laboratorio de comunicaciones de la Universidad Industrial de Santander llegará a ser parte de un laboratorio distribuido, en asociación con otras universidades o entidades en modalidades de investigación y prestación de servicios.

BIBLIOGRAFIA

ANE [sitio web], Agencia Nacional del Espectro [Consulta 30 de enero 2017]. Disponible en: <http://www.ane.gov.co/>.

DENG, P., & KAVEHRAD, M. Real-time software-defined single-carrier QAM mimo visible light communication system. In Integrated Communications Navigation and Surveillance (ICNS), (pp. 5A3-1). IEEE, 2016.

DÍAZ BAUTISTA I. C., RANGEL SILVA M. J., AND ORTEGA H., Implementación de un downconverter sobre una FPGA bajo el concepto de software defined radio usando xilinx system generator. Universidad Industrial de Santander, (2011).

DROMS, R., BOUND, J., VOLZ, B., LEMON, T., PERKINS, C., & CARNEY, M. Dynamic host configuration protocol for IPv6 (DHCPv6) (No. RFC 3315), (2003).

ETTUS, M., & BRAUN, M. The Universal Software Radio Peripheral (USRP) family of low-cost SDRd. Opportunistic Spectrum Sharing and White Space Access: The Practical Reality, pp 3-23, (2015).

FIBRE [sitio web], Future Internet Brazilian Environment for experimentation [Consultado 28 de octubre de 2016]. Disponible en: <http://fibre.org.br/>.

FRANTZ, P. J., & THOMPSON, G. O. U.S. Patent No. 5,959,990. Washington, DC: U.S. Patent and Trademark Office, (1999)..

GENI [sitio web], Global Environment for Network Innovation [Consultado 19 de octubre de 2016]. Disponible en: <https://www.geni.net/>.

GENI LAB [sitio web], Arquitectura de red de GENI Lab [Consultado 25 de noviembre de 2016]. Disponible en: <http://www.geni.net/documentation/geni-architecture/>.

GMOC [sitio web], Geni Meta-Operations Center [Consultado 26 de octubre de 2016]. Disponible en: <http://gmoc.grnoc.iu.edu/>.

GNU Radio [sitio web], GNU Radio [Consulta 12 de marzo 2017]. Disponible en: <https://gnuradio.org/>.

GÓMEZ GARCÍA A. F., Sistema de comunicación digital m-aria sobre una plataforma de radio definido por software basado en gnu radio, Universidad Industrial de Santander, (2015).

HUIDOBRO, J. Tecnologías de información y comunicación. Universidad Politécnica de Madrid, pp 2, (2007).

LARSSON, E. G., EDFORS, O., TUFVESSON, F., & MARZETTA, T. L. Massive MIMO for next generation wireless systems. IEEE Communications Magazine, 52(2), pp 186-195, (2014).

MEDINA, A. P., SABA, G. H., SILVA, J. H., & dE GUEVARA Durán, E. L. Los laboratorios virtuales y laboratorios remotos en la enseñanza de la ingeniería. Revista internacional de educación en Ingeniería, (2011).

MITOLA, J. The software radio architecture. IEEE Communications magazine, 33(5), pp 26-38, (1995).

NITOS [sitio web], Network Implementation Testbed using Open Source platforms [Consultado 28 de octubre 2016]. Disponible en: <http://nitlab.inf.uth.gr/NITlab/nitos>.

ORBIT [sitio web], Open-Access Research Testbed for Next-Generation Wireless Networks [Consultado 19 de octubre 2016]. Disponible en: <http://www.orbit-lab.org/>.

ORBIT LAB [sitio web], Hardware de un mini-rack (Massive-MIMO) [Consultado 2 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/attachment/wiki/Hardware/bDomains/aGrid/Massive-MIMO.jpg>.

ORBIT LAB [sitio web], Distribución de hardware de Orbit Lab [Consultado 19 de octubre de 2016]. Disponible en: <http://www.orbit-lab.org/attachment/wiki/Hardware/hardware-50.png>.

ORBIT LAB [sitio web], Características de nodo de generación 1 [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/aNodeVer1#ORBITNodeVersion1>.

ORBIT LAB [sitio web], Características de nodo de generación 3 tipo LV-67C [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/cNodeVer3/aMain/aLV67C#LV-67C>.

ORBIT LAB [sitio web], Características de nodo de generación 3 tipo LV-67G [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/cNodeVer3/aMain/bLV67G#LV-67G>.

ORBIT LAB [sitio web], Características de nodo de generación 3 tipo LV-67J [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/cNodeVer3/aMain/cLV67J#LV-67J>.

ORBIT LAB [sitio web], características de nodo de generación 3 tipo LV-67B [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/cNodeVer3/bMobile/aLV67B#LV-67B>.

ORBIT LAB [sitio web], Descripción de Nodo de Orbit Lab [Consultado 2 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/attachment/wiki/Hardware/jCM/OrbitNode.jpg>.

ORBIT LAB [sitio web], Sandboxes, características [Consultado 9 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/bDomains/cSandboxes#Sandboxes>.

ORBIT LAB [sitio web], Ubicación de los dispositivos USRP de Orbit Lab [Consultado 18 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/fDevices/cSDR/gUSRP>.

ORBIT LAB [sitio web], Características de nodo de generación 3 tipo LV-67F [Consultado 3 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/hNodes/cNodeVer3/bMobile/bLV67F#LV-67F>.

ORBIT LAB [sitio web], interacción usuario-Orbit Lab [Consultado 21 de noviembre de 2016]. Disponible en: <http://www.orbit-lab.org/attachment/wiki/Software/OMF-User-View.png>.

ORBIT LAB [sitio web], switch de antenas [Consultado 24 de abril de 2017]. Disponible en: <http://www.orbit-lab.org/wiki/Hardware/IDirAnt#Antennas>

PINO GALLARDO S. A., Codificación conjunta de fuente-canal : Mapeos de Shannon-Kotelnikov sobre radio definido por software, Universidad Industrial de Santander, (2013).

R2lab [sitio web], R2lab anechoic chamber a heterogeneous wireless testbed [Consultado 1 de noviembre de 2016]. Disponible en: <http://r2lab.inria.fr/hardware.md>.

SDRCol2015 [sitio web], Proyecto SDRCol2015 [Consultado 1 de febrero de 2017]. Disponible en: <https://sites.google.com/a/radiogis.uis.edu.co/proysdrcol2015/home>.

SHEN, D. Tutorial 4: The USRP Board. Introduction. SDR Documentation. Notre Dame, IN: University of Notre Dame, (2005).

SUAREZ MONSALVE J. M. and TRIANA GALEANO V., Diseño de manual de prácticas para un laboratorio de comunicaciones digitales basado en la técnica de radio definido por software. Universidad Industrial de Santander, (2012).

TUTTLEBEE, W. H. Software defined radio: enabling technologies. John Wiley & Sons, (Ed.). (2003).

UMAÑA GALLEGO J. D., Nuevo laboratorio remoto en la Universidad de Antioquia. http://www.elcolombiano.com/historico/nuevo_laboratorio_remoto_en_la_u_de_a-BAEC_270735, Noviembre 2013.

WIRID LAB [sitio web], Wireless research innovation and development laboratory [Consultado 8 de diciembre de 2016]. Disponible en: <http://wirid-lab.umng.edu.co/>, Universidad Militar Nueva Granada.

YOVANY, W., & FAGUA, A. L. Laboratorios Remotos y Virtuales: Una Herramienta para el Desarrollo de Prácticas en Ingeniería. Ciencia, Innovación y Tecnología, 1, pp 71-80, (2014).

ANEXOS

ANEXO A - Instalación de UHD y GNU-Radio en cualquiera de sus versiones

UHD usa el compilador de GCC (*GNU Compiler Collection*) y es totalmente compatible con Linux. Es recomendable usar una versión de Linux LTS para que los paquetes de las nuevas actualizaciones se encuentren disponibles y no haya interferencia en la operación del software.

Todos los comandos presentados a continuación deben ejecutarse en la terminal, independientemente de la distribución de Linux.

Paso 1, Instalar los paquetes necesarios:

- Para Ubuntu:

```
sudo apt-get update
```

- Para Fedora 21:

```
sudo yum update
```

- Para Fedora 22, 23, 24 y 25:

```
sudo dnf update
```

Paso 2, instalar las dependencias requeridas por UHD y GNU Radio:

- Para Ubuntu 17.04:

```
sudo apt-get -y install git swig cmake doxygen build-essential
libboost-all-dev libtool libusb-1.0-0 libusb-1.0-0-dev libudev-
dev libncurses5-dev libfftw3-bin libfftw3-dev libfftw3-doc
libcxxunit-1.13-0v5 libcxxunit-dev libcxxunit-doc ncurses-bin
cpufrequtils python-numpy python-numpy-doc python-numpy-dbg
python-scipy python-docutils qt4-bin-dbg qt4-default qt4-doc
libqt4-dev libqt4-dev-bin python-qt4 python-qt4-dbg python-qt4-
dev python-qt4-doc python-qt4-doc libqwt6abi1 libfftw3-bin
libfftw3-dev libfftw3-doc ncurses-bin libncurses5 libncurses5-
dev libncurses5-dbg libfontconfig1-dev libxrender-dev libpulse-
dev swig g++ automake autoconf libtool python-dev libfftw3-dev
libcxxunit-dev libboost-all-dev libusb-dev libusb-1.0-0-dev
fort77 libsdl1.2-dev python-wxgtk3.0 git-core libqt4-dev python-
numpy ccache python-opengl libgsl-dev python-cheetah python-mako
python-lxml doxygen qt4-default qt4-dev-tools libusb-1.0-0-dev
libqwt5-qt4-dev libqwtplot3d-qt4-dev pyqt4-dev-tools python-
qwt5-qt4 cmake git-core wget libxi-dev gtk2-engines-pixbuf r-
base-dev python-tk liborc-0.4-0 liborc-0.4-dev libasound2-dev
python-gtk2 libzmq3-dev libzmq5 python-requests python-sphinx
libcomedi-dev python-zmq
```

- Para Ubuntu 16.04:

```
sudo apt-get -y install git swig cmake doxygen build-essential
libboost-all-dev libtool libusb-1.0-0 libusb-1.0-0-dev libudev-
dev libncurses5-dev libfftw3-bin libfftw3-dev libfftw3-doc
libcxxunit-1.13-0v5 libcxxunit-dev libcxxunit-doc ncurses-bin
cpufrequtils python-numpy python-numpy-doc python-numpy-dbg
python-scipy python-docutils qt4-bin-dbg qt4-default qt4-doc
libqt4-dev libqt4-dev-bin python-qt4 python-qt4-dbg python-qt4-
dev python-qt4-doc python-qt4-doc libqwt6abi1 libfftw3-bin
```

```
libfftw3-dev libfftw3-doc ncurses-bin libncurses5 libncurses5-  
dev libncurses5-dbg libfontconfig1-dev libxrender-dev libpulse-  
dev swig g++ automake autoconf libtool python-dev libfftw3-dev  
libcppunit-dev libboost-all-dev libusb-dev libusb-1.0-0-dev  
fort77 libsdl1.2-dev python-wxgtk3.0 git-core libqt4-dev python-  
numpy ccache python-opengl libgsl-dev python-cheetah python-mako  
python-lxml doxygen qt4-default qt4-dev-tools libusb-1.0-0-dev  
libqwt5-qt4-dev libqwtplot3d-qt4-dev pyqt4-dev-tools python-  
qwt5-qt4 cmake git-core wget libxi-dev gtk2-engines-pixbuf r-  
base-dev python-tk liborc-0.4-0 liborc-0.4-dev libasound2-dev  
python-gtk2 libzmq-dev libzmq1 python-requests python-sphinx  
libcomedi-dev python-zmq
```

- Para Ubuntu 15.04 y 15.10:

```
sudo apt-get -y install git swig cmake doxygen build-essential  
libboost-all-dev libtool libusb-1.0-0 libusb-1.0-0-dev libudev-  
dev libncurses5-dev libfftw3-bin libfftw3-dev libfftw3-doc  
libcppunit-dev libcppunit-doc ncurses-bin cpufrequtils python-  
numpy python-numpy-doc python-numpy-dbg python-scipy python-  
docutils qt4-bin-dbg qt4-default qt4-doc libqt4-dev libqt4-dev-  
bin python-qt4 python-qt4-dbg python-qt4-dev python-qt4-doc  
python-qt4-doc libqwt6 libfftw3-bin libfftw3-dev libfftw3-doc  
ncurses-bin libncurses5 libncurses5-dev libncurses5-dbg  
libfontconfig1-dev libxrender-dev libpulse-dev swig g++ automake  
autoconf libtool python-dev libfftw3-dev libcppunit-1.13-0v5  
libboost-all-dev libusb-dev libusb-1.0-0-dev fort77 libsdl1.2-  
dev python-wxgtk2.8 git-core libqt4-dev python-numpy ccache  
python-opengl libgsl0-dev python-cheetah python-mako python-lxml  
doxygen qt4-default qt4-dev-tools libusb-1.0-0-dev libqwt5-qt4-  
dev libqwtplot3d-qt4-dev pyqt4-dev-tools python-qwt5-qt4 cmake  
git-core wget libxi-dev gtk2-engines-pixbuf r-base-dev python-tk  
liborc-0.4-0 liborc-0.4-dev libasound2-dev python-gtk2 libzmq-
```

```
dev libzmq1 python-requests python-sphinx libcomedi-dev python-  
zmq
```

- **Para Ubuntu 14.04 y 14.10:**

```
sudo apt-get -y install git swig cmake doxygen build-essential  
libboost-all-dev libtool libusb-1.0-0 libusb-1.0-0-dev libudev-  
dev libncurses5-dev libfftw3-bin libfftw3-dev libfftw3-doc  
libcppunit-1.13-0 libcppunit-dev libcppunit-doc ncurses-bin  
cpufrequtils python-numpy python-numpy-doc python-numpy-dbg  
python-scipy python-docutils qt4-bin-dbg qt4-default qt4-doc  
libqt4-dev libqt4-dev-bin python-qt4 python-qt4-dbg python-qt4-  
dev python-qt4-doc python-qt4-doc libfftw3-bin libfftw3-dev  
libfftw3-doc ncurses-bin libncurses5 libncurses5-dev  
libncurses5-dbg libfontconfig1-dev libxrender-dev libpulse-dev  
swig g++ automake autoconf libtool python-dev libfftw3-dev  
libcppunit-dev libboost-all-dev libusb-dev libusb-1.0-0-dev  
fort77 libsdl1.2-dev python-wxgtk2.8 git-core libqt4-dev python-  
numpy ccache python-opengl libgsl0-dev python-cheetah python-  
mako python-lxml doxygen qt4-default qt4-dev-tools libusb-1.0-0-  
dev libqwt5-qt4-dev libqwtplot3d-qt4-dev pyqt4-dev-tools python-  
qwt5-qt4 cmake git-core wget libxi-dev gtk2-engines-pixbuf r-  
base-dev python-tk liborc-0.4-0 liborc-0.4-dev libasound2-dev  
python-gtk2 libzmq1 libzmq-dev python-requests python-sphinx  
libcomedi-dev
```

- **Para Fedora 21:**

```
sudo yum -y groupinstall "Engineering and Scientific"  
"Development Tools" "Software Development Tools" "C Development  
Tools and Libraries"
```

```
sudo yum -y install fftw-devel cppunit-devel wxPython-devel
boost-devel alsa-lib-devel numpy gsl-devel python-devel pygsl
python-cheetah python-mako python-lxml PyOpenGL qt-devel qt qt4
qt4-devel PyQt4-devel qwt-devel qwtplot3d-qt4-devel libusbx-
devel cmake git wget python-docutils cppzmq-devel PyQwt PyQwt-
devel qwt-devel gtk2-engines xmlrpc-c-"*" tkinter orc orc-devel
python-sphinx SDL-devel swig zeromq2-devel python-zmq comedilib
comedilib-devel thrift-devel python-thrift scipy zeromq zeromq-
devel
```

- Para Fedora 22, 23, 24 y 25:

```
sudo dnf -y groupinstall "Engineering and Scientific"
"Development Tools" "C Development Tools and Libraries"

sudo dnf -y install fftw-devel cppunit-devel wxPython-devel
boost-devel alsa-lib-devel numpy gsl-devel python-devel pygsl
python-cheetah python-mako python-lxml PyOpenGL qt-devel PyQt4-
devel qwt-devel qwtplot3d-qt4-devel libusbx-devel cmake python-
docutils PyQwt PyQwt-devel gtk2-engines xmlrpc-c-"*" tkinter
orc-devel python-sphinx SDL-devel swig perl-ZMQ-LibZMQ2 perl-
ZMQ-LibZMQ2 zeromq zeromq-devel python-requests gcc-c++ doxygen
zeromq-ada-devel cppzmq-devel perl-ZeroMQ amavisd-new-zeromq
amavisd-new-snmp-zeromq php-zmq python-zmq czmq uwsgi-logger-
zeromq comedilib comedilib-devel pygtk2 ncurses-"*" thrift-devel
python-thrift scipy
```

Luego de instalar las dependencias correctamente se debe reiniciar el sistema.

Paso 3, construir e instalar UHD:

Para realizar este paso es importante asegurarse de que ningún dispositivo USRP se encuentre conectado en ese momento.

Se crea una carpeta para almacenar el repositorio:

```
cd $HOME
mkdir workarea-uhd
cd workarea-uhd
```

A continuación, se clona el repositorio y se ubica en el directorio clonado:

```
git clone https://github.com/EttusResearch/uhd
cd uhd
```

Luego se selecciona la versión de UHD deseada. Se puede observar cuales versiones están disponibles con el siguiente comando:

```
git tag -l
```

Para luego seleccionar la requerida (se recomienda la 3.9.3 pues fue la que se usó en el desarrollo del presente trabajo de grado) de la siguiente manera:

```
# Example: For UHD 3.9.3:
git checkout release_003_009_003
```

Posteriormente, se crea una carpeta de compilación dentro del repositorio:

```
cd host
```

```
mkdir build  
cd build
```

Se invoca el CMake para crear los ejecutables:

```
cmake ../
```

Se corre el Make para construir el UHD:

```
make
```

Para corroborar que la instalación se ejecutó de manera adecuada se hace una prueba con el siguiente comando:

```
make test
```

Una vez construido el UHD, es necesario realizar el proceso de instalación del mismo:

```
sudo make install
```

A continuación, se actualiza el caché de la biblioteca compartida del sistema:

```
sudo ldconfig
```

Por último, para probar que el UHD se encuentre instalado de manera correcta se ejecuta el comando:

```
uhd_find_devices
```

De donde se debe obtener:

```
linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.009.003
```

```
No UHD Devices Found
```

Paso 4, construir e instalar GNU-Radio:

Se crea una carpeta para almacenar el repositorio:

```
cd $HOME  
mkdir workarea-gnuradio  
cd workarea-gnuradio
```

A continuación, se clona el repositorio y se ubica en el directorio clonado:

```
git clone --recursive https://github.com/gnuradio/gnuradio  
  
cd gnuradio
```

Luego se extrae la versión de GNU-Radio disponible:

```
git checkout v3.7.10.1
```

Posteriormente, se crea una carpeta de compilación dentro del repositorio:

```
mkdir build  
cd build
```

Se invoca el CMake para crear los ejecutables:

```
cmake ../
```

Se corre el Make para construir el GNU-Radio:

```
make
```

Para corroborar que la instalación se ejecutó de manera adecuada se hace una prueba con el siguiente comando:

```
make test
```

Una vez construido el GNU-Radio, es necesario realizar el proceso de instalación del mismo:

```
sudo make install
```

A continuación, se actualiza el caché de la biblioteca compartida del sistema:

```
sudo ldconfig
```

En este punto GNU-Radio se encuentra instalado y listo para ser usado.

Por último para ejecutar el software se corre el comando:

```
gnuradio-companion
```

Con la finalidad de solucionar inquietudes y encontrar futuras actualizaciones del software para nuevas distribuciones de Linux o versiones de Ubuntu, a continuación se presenta el enlace donde se encuentra el tutorial de la página de Ettus Research, el cual fue guía para el presente anexo: [https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_\(UHD_and_GNU_Radio\)_on_Linux](https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux).