## UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

## UNAN – León

Facultad De Ciencias y Tecnología

Departamento de Computación



Asterisk, una opción viable para la comunicación entre usuarios VoIP de una red de área local y la red telefónica conmutada.

Tesis para optar al título de

## **INGENIERO EN TELEMÁTICA**

Presentado por:

- Br. Alba Suyen Olivas Ochoa.
- Br. Cinthya Carolina Membreño Martínez.
- Br. Kenia Judith Loaisiga Navarrete.

## Tutor:

Msc. Denis Espinoza Hernández.

León, Agosto 2014

## AGRADECIMIENTOS

Primero queremos agradecerle a Dios por permitirnos un día más de vida y por haber puesto en nuestro camino a todas aquellas personas que nos ayudaron de alguna manera al desarrollo de la tesis.

A nuestros padres por darnos siempre la mejor educación que hayamos podido tener y su apoyo incondicional siempre que los necesitábamos.

Al Msc. Denis Espinoza por ser nuestro tutor y apoyarnos en todas las correcciones de la Tesis y en el planteamiento de la misma.

# DEDICATORIA

Por todo el apoyo que me dieron durante todo el tiempo y en el desarrollo de mi tesis quisiera dedicar este trabajo a mi esposo y a mi hija preciosa Keitlyn que los amo con todo mi corazón.

A mi mamá y a mi mamita que las quiero mucho ellas me motivaron para salir adelante y no darme por vencida en los obstáculos que la vida nos pone.

A mis amigas y a todas las personas que siempre estaban a mi lado.

Kenia Judith Loaisiga Navarrete.

## DEDICATORIA

Primeramente le agradezco a Dios y a la Virgen María por darme las fuerzas para seguir adelante, por concederme la familia que tengo y por haberme permitido terminar mis estudios universitarios.

A mi madre Alicia Martínez Trujillo por su amor, su constante dedicación hacia mí y hacia a mi hermanita, por su paciencia, su apoyo, su esfuerzo y por creer en mí y alentarme siempre. A mi padre Freddy Membreño Molina por su apoyo, por ser un hombre trabajador y perseverante. A mi hermanita Melissa Membreño Martínez por su compañía, su cariño y por ser parte de mi vida. A mi mita Bertha Membreño que la quiero mucho y que siempre estará en mi corazón.

A mi esposo Cristhiam Pacheco Centeno le agradezco por su incondicional apoyo, por su amor, compañía y por estar a mi lado en los buenos y malos momentos.

Al Msc. Denis Espinoza tutor de la tesis por guiarnos y ayudarnos en el desarrollo del trabajo, por habernos brindado su tiempo y experiencia. También les agradezco a mis compañeras Alba Olivas Ochoa y Kenia Loaisiga Navarrete por su compromiso, su esfuerzo y dedicación para lograr culminar la tesis.

Cinthya Carolina Membreño Martínez.

# DEDICATORIA

Agradezco a Dios por darme fuerzas para finalizar mis estudios universitarios.

A mi mamá Consuelo Ochoa Puerto por todo el apoyo que me brindo y estar siempre a mi lado.

A mis hermanos Rodrigo Ochoa y Alexander Olivas Ochoa por darme ánimo en todo momento.

A mis abuelitos (Q.D.P) por ser mi inspiración y ejemplo de superación.

Y a mis amigos por todo el apoyo que me brindaron.

Alba Suyen Olivas Ochoa.

## Contenido

CAPÍTU	LO 1	INTRODUCCIÓN	1
1.1	Anteo	cedentes	2
1.2	Plant	eamiento del Problema	4
1.3	Justif	icación	5
1.4	Objet	tivos	7
CAPÍTU	LO 2	Marco Teórico	8
2.1	Introd	ducción a la telefonía	9
2.2	Voice	e over Internet Protocol (VoIP)	10
2.3	Proto	colos de Transporte	15
2.4	VolP	y telefonía convencional: Algunas diferencias	16
2.5	Hard	ware para VoIP	17
2.6	Softw	vare de telefonía VoIP	19
2.7	¿Qué	es SIPp?	24
CAPÍTU	LO 3	Diseño metodológico	25
3.1	Mate	riales utilizados:	26
3.2	Etapa	as del proyecto	27
CAPÍTU	LO 4	DESARROLLO	30
4.1	Integ	ración de Asterisk a la red telefónica conmutada	31
4.2	Evalu	ación de la calidad de la llamada	48
4.3	Reali	zación de pruebas de rendimiento de Asterisk	50
CAPÍTU	LO 5	Conclusiones	64
5.1	Conc	lusiones	65
5.2	Reco	mendaciones	65
ANEXO	S		66
5.3	Anex	o 1: Instalación y Configuración del servidor Asterisk	67

5.4	Anexo 2: Configuración de los archivos sip.conf, extensions.conf y voicemail.conf	70
5.5	Anexo 3: Instalación del Softphone SJphone (en el lado del cliente)	72
5.6	Anexo 4: Instalación y Configuración de la tarjeta Authentic X100P SE FXO PCI	75
5.7	Anexo 5: Instalación, Configuración y pruebas realizadas con la herramienta SIPp	83
BIBLIO	GRAFÍA	96

## Índice de Figuras:

Figura 1: Telefonía Tradicional y VoIP	. 10
Figura 2: Ejemplo de un flujo SIP	. 14
Figura 3: Dispositivo FXO	. 18
Figura 4: FXO/FXS sin centralita	. 19
Figura 5: Tarjeta Authentic X100P SE FXO PCI	. 23
Figura 6: Esquema de Red únicamente para clientes VoIP	. 31
Figura 7:Usuario2 llamando al Usuario3	. 33
Figura 8: SJphone del Usuario3 donde puede aceptar o rechazar la llamada	. 33
Figura 9: Llamada establecida entre clientes VoIP	. 34
Figura 10: Proceso de la llamada en el lado del servidor	. 34
Figura 11: Esquema de Red del Servidor Asterisk conectado a la PSTN	. 35
Figura 12: Conexiones de las entradas de la tarjeta X100p	. 36
Figura 13: Tarjeta PCI reconocida por el servidor Asterisk	. 36
Figura 14: Estado de la tarjeta conectada a la PSTN	. 37
Figura 15: Iniciamos el servicio DAHDI	. 37
Figura 16: Plan de marcación del contexto privilegiados	. 38
Figura 17: dialplan from-pstn	. 39
Figura 18: Monitorización de las llamadas en el servidor	. 39
Figura 19: Llamada entrante desde la PSTN	. 40
Figura 20: Llamada entrante desde la PSTN redirigida a una extensión interna	. 40
Figura 21: Escenario de red donde se crea el IVR	. 41
Figura 22: Visualizando dialplan from-pstn	. 44
Figura 23: Plan de marcación del contexto menu-dia	. 44
Figura 24: Dialplan del contexto privilegiados	. 45
Figura 25: Plan de marcación del contexto miprimerejemplo	. 45
Figura 26: Usuario selecciona la opción 1 del IVR	. 46
Figura 27: Usuario selecciona la opción 2 del IVR	. 46
Figura 28: Usuario selecciona una opción inválida	. 47
Figura 29: Llamada redireccionada a la operadora	. 47
Figura 30: Escenario SIPp envío de mensajes	. 50
Figura 31: Cliente SIPp archivo de audio pcap	. 51
Figura 32: modules.conf codecs agregados	. 53

Figura 33: SIPp configuración en el archivo sip.conf	54
Figura 34: SIPp configuración en extensions.conf	54
Figura 35: Ejecución del cliente SIPp con los valores que desbordan el servidor Asterisk	57
Figura 36: Finalización del cliente SIPp con los valores que desbordan el servidor Asterisk	57
Figura 37: Servidor SIPp resultados en donde se desborda Asterisk	58
Figura 38: Desborde en el Servidor Asterisk	59
Figura 39: Escenario SIPp con descarga	60
Figura 40: Cliente SIPp prueba con descarga	61
Figura 41: Descarga con JDownloader	62
Figura 42: Descarga con FlashGet	62
Figura 43: Configuración del SJphone paso 1	72
Figura 44: Configuración del SJphone paso 2	73
Figura 45: Configuración del SJphone paso 3	73
Figura 46: Información del Usuario1	74
Figura 47: Usuario1 creado satisfactoriamente	74
Figura 48: Ejecución del comando dmesg	77
Figura 49: Ejecución del comando dahdi_hardware	78
Figura 50: Ejecución del comando dahdi_tool	78
Figura 51: Ejecución del comando dahdi_cfg -vv	79
Figura 52: Estado de la tarjeta sin conexión a la PSTN	79
Figura 53: Estado de la tarjeta con conexión a la PSTN	79
Figura 54: Iniciar dahdi	80
Figura 55: Ejecución del comando dahdi_genconf -v	80
Figura 56: Visualización del archivo system.conf de dahdi	80
Figura 57: Visualización del archivo dahdi-channels.conf de dahdi	81
Figura 58: Visualización del archivo chan_dahdi.conf	82
Figura 59: Archivo XML del cliente SIPp parte1	84
Figura 60: Archivo XML del cliente SIPp parte 2	85
Figura 61: Archivo XML del servidor SIPp parte 1	86
Figura 62: Archivo XML del servidor SIPp parte 2	87
Figura 63: Cliente SIPp primera prueba sin descarga con SIPp	88
Figura 64: Servidor SIPp primera prueba sin descarga con SIPp	89
Figura 65: Cliente SIPp segunda prueba sin descarga con SIPp	90
Figura 66: Servidor SIPp segunda prueba sin descarga con SIPp	91

Figura 67: Cliente SIPp tercera prueba sin descarga con SIPp	. 92
Figura 68: Servidor SIPp tercera prueba sin descarga con SIPp	. 93
Figura 69: Cliente SIPp cuarta prueba sin descarga con SIPp	. 94
Figura 70: Servidor SIPp cuarta prueba sin descarga con SIPp	. 95

## Índice de tablas:

Tabla 1: Ventajas y Desventajas de la telefonía VoIP	. 17
Tabla 2: Servicios que ofrece Asterisk.	. 20
Tabla 3 : Materiales Hardware	26
Tabla 4: Materiales Software	27
Tabla 5: Resultados de la calidad de las llamadas a la PSTN	. 48
Tabla 6: Prueba SIPp valores donde el servidor Asterisk se desborda	. 56
Tabla 7: Directorios de Asterisk	69
Tabla 8: Directorios de Dahdi	. 77



## Resumen

Actualmente las circunstancias laborales han hecho que las empresas perciban la necesidad de estar bien comunicadas para lograr la eficiencia de sus objetivos, es por esto que necesitan de una estructura de comunicación a bajo precio, eficiente y acorde a sus necesidades.

Muchas empresas del campo de las telecomunicaciones ofrecen productos que satisfacen estas necesidades, pero sus soluciones comerciales son cerradas, propietarias, y frecuentemente costosas Private Branch Exchange (PBX), por tal motivo la mayoría de las empresas hoy en día utilizan una tecnología abierta que proporciona un estándar de comunicaciones Voz sobre IP (VoIP), que son flexibles y no están sujetas a las limitaciones de ningún fabricante. Los estándares abiertos permiten la posibilidad de emular la funcionalidad de los sistemas de telefonía tradicionales con un programa funcionando en un ordenador, entre ellos tenemos Asterisk, un sistema de código abierto que utiliza una plataforma servidor estándar y tarjetas PCI para las interfaces de telefonía. Asterisk ha incorporado la mayoría de estándares de telefonía del mercado tanto las tradicionales (TDM) con el soporte de puertos de interfaz analógicas (FXS y FXO) y RDSI, como las de telefonía IP. (Basto, 2007)

En toda empresa es importante tanto la comunicación interna como la externa, por la necesidad de comunicarse a largas distancias y para el desarrollo de esta, por ende decidimos realizar este trabajo para enriquecer la comunicación incluyendo muchos servicios, tecnologías y dispositivos utilizando Asterisk como nuestro servidor de telefonía, con la finalidad de hacer llamadas desde la red interna VoIP hacia la telefonia tradicional.



# CAPÍTULO 1 INTRODUCCIÓN

1

## 1.1 Antecedentes

La tecnología ha avanzado rápidamente a lo largo de los años innovando la comunicación, hace 30 años Internet no existía, y las comunicaciones se realizaban por medio del teléfono a través de la red telefónica pública conmutada (PSTN), pero con el pasar de los años y el avance tecnológico han ido apareciendo nuevas tecnologías y aparatos bastante útiles que nos han permitido pensar en nuevas formas de comunicación: PCS, teléfonos celulares y finalmente la popularización de la gran red Internet. (La tecnologia a traves del tiempo)

Hoy en día podemos ver una gran revolución en comunicaciones: todas las personas usan los computadores e Internet en el trabajo y en el tiempo libre para comunicarse con otras personas, intercambiar datos y a veces para hablar con más personas usando distintas aplicaciones, el cual particularmente comenzó a difundir en el mundo la idea que en el futuro se podría utilizar una comunicación en tiempo real por medio del PC: Voice Over Internet Protocol (VoIP). La tecnología voz sobre IP o VoIP tuvo sus inicios en 1973, su desarrollo y evolución como tecnología fue aceptada en el 2004 y como resultado se obtuvo un VoIP desarrollado; por tal razón es una alternativa comercial viable para llamadas de voz. (Bermeo, 2009)

La Telefonía sobre Internet o VoIP es el presente y futuro de las comunicaciones por sus económicas tarifas, buena calidad en la comunicación ya que proporciona el poder para usar la telefonía convencional junto con las nuevas tecnologías. Cada vez Voz IP gana más terreno sobre la telefonía pública conmutada (telefonía tradicional) y son más populares entre empresas y consumidores. (internet)

En la industria de telecomunicaciones uno de los programas más implementados es Asterisk, que comenzó en 1999 y fue desarrollado por Mark Spencer, este es un software Open Source que se adecua a las necesidades telefónicas para resolver de manera eficaz las necesidades de negocio de las medianas y pequeñas empresas a unos costos mínimos. (noa)

Actualmente en Nicaragua la comunicación de Voz sobre IP con Asterisk es cada vez más empleada, existen distintas empresas que ofrecen estos servicios, por ejemplo PBX Nicaragua S.A una empresa que garantiza que sus equipos para centrales telefónicas VoIP son versátiles y que se acoplan a las necesidades del cliente, además es lo más conveniente para reducir sus costos hasta un 80% en facturas telefónicas. (PBX NICARAGUA)

En otros países se han realizado distintos estudios acerca de la implementación y configuración de Asterisk como servidor VoIP, para que pueda comunicarse a la telefonía tradicional:

#### ✤ Asterisk: Configuración de Zapata.

Este documento fue escrito para mostrar cómo se configura Asterisk para que actúe como interfaz con la telefonía convencional a través del módulo Zapata Telephony o Zaptel. Cubre la forma en la que se presentan las tecnologías en España, la configuración del módulo, la comprobación de las líneas y el uso que hace Asterisk de ellas. (Rubio Jiménez, (PDF)Asterisk)

#### \* Asterisk, como tener una centralita de teléfonos en tu casa.

Este consiste en definir los componentes para crear una centralita de teléfonos con Asterisk. (SeMaToVe, Asterisk, como tener una centralita de telefonos en tu casa, 2008)

#### Implementando un servidor de Asterisk.

Este blog aborda la implementación de Asterisk como servidor de VoIP, además el funcionamiento de Asterisk con la incorporación de un dispositivo analógico en la distribución Ubuntu Server 12.04. (Restrepo, Implementando un servidor de Asterisk (método Julio Restrepo), 2012)

## Voz sobre IP implementada con Asterisk: Teoría y Práctica.

Esta monografía se basa en el estudio de la Voz sobre IP, una nueva tecnología en el mundo de las telecomunicaciones utilizando Asterisk como servidor VoIP. (Larios Lira, Rivera, & Zapata)



## 1.2 Planteamiento del Problema

Debido a los costos que implica contratar una Private Branch Exchange (PBX) apareció una tecnología abierta que proporciona un estándar de comunicaciones Voz sobre IP (VoIP) como una alternativa de telefonía con costos reducidos, gracias a la incorporación de la telefonía IP muchas empresas pueden tener sus propias Private Branch Exchange (PBX) de una manera más económica que las comunicaciones tradicionales, además de poder ofrecer una gran variedad de funcionalidades que no se pueden utilizar con los sistemas convencionales.

Uno de los sistemas de comunicaciones más completo, avanzado y económico es Asterisk un software abierto, versátil, no propietario, que posee una arquitectura modular y tiene distintas funcionalidades que controla y gestiona las comunicaciones de cualquier tipo, ya sean analógicas, digitales o Voz sobre IP( VoIP), pero por sí solo, no es una herramienta plug-and-play que venga lista para hacer llamadas, sino que es necesario atravesar por numerosos pasos (descarga, compilación, instalación y configuración) para que pueda realizar labores útiles.

Cabe destacar que en toda empresa existe la necesidad de poder hablar desde el interior hasta el exterior. Una empresa unida y con alto nivel de compromiso necesita de una buena comunicación tanto interna como externa aunque esta no es una de las funciones principales de Asterisk pero se puede lograr con las diferentes configuraciones.

## Pregunta general:

¿Es Asterisk una opción viable para las pequeñas empresas que necesitan comunicación entre los dispositivos VoIP de una red de área local y la red telefónica convencional?

## Preguntas específicas:

- ¿Cuáles son los procedimientos para permitir que la PBX Asterisk se comunique con la red telefónica tradicional?
- ¿Qué calidad de servicio ofrece esta implementación en las llamadas que se realizan desde los teléfonos VoIP hacia la red telefónica tradicional?
- ¿Cuál es la capacidad del servidor Asterisk en base a la cantidad de conexiones concurrentes que este es capaz de soportar?

4

## 1.3 Justificación

Hoy en día existen distintos sistemas de comunicación, como la comunicación por cable, la inalámbrica, la telefonía tradicional, la telefonía móvil, Internet, etc. La mayoría de las empresas que ofrecen los servicios de una Private Branch Exchange (PBX) son propietarias, cerradas y frecuentemente costosas, es por esto que las pequeñas y medianas empresas que hagan uso de estos servicios necesitan de una estructura de comunicación a bajo precio, eficiente y acorde a sus necesidades, por tal motivo se ha incorporado una nueva tecnología que proporciona un estándar de comunicaciones Voz sobre IP (VoIP) y también se han desarrollo diferentes aplicaciones de software libre tal como Asterisk que brindan estos servicios, esto representa un beneficio aún mayor para las empresas que lo utilicen, ya que no solamente bajan los gastos de facturación del servicio sino que también los costos de licencias. Por tal razón decidimos implementar una PBX con Asterisk como servidor de telefonía VoIP de bajo costo y conectado a la red telefónica convencional.

Debido a que Asterisk procesa las llamadas telefónicas de manera digital y está orientado al uso de redes de dispositivos digitales, para poder realizar llamadas desde un ordenador hacia un teléfono fijo es necesaria la compra de un dispositivo electrónico FXO para conectar directamente al servidor Asterisk a la red análoga PSTN y convertir la voz digital a voz análoga para ser transportada por la PSTN y viceversa.

#### 1.3.1 Originalidad

Asterisk ha incorporado la mayoría de estándares de telefonía del mercado, tanto las tradicionales (TDM) con el soporte de puertos de interfaz analógicas (FXS y FXO) y RDSI (básicos y primarios), como las de telefonía IP a bajo coste, pero actualmente en nuestro país la comunicación de Voz IP con Asterisk es muy poco empleada tanto en pequeñas y medianas empresas, aunque existen numerosas tecnologías que permiten hablar por la red, Asterisk incluye muchas características que anteriormente solo estaban disponibles en costosos sistemas propietarios PBX.

#### 1.3.2 Alcance

Una vez realizadas todas las configuraciones y con la tarjeta analógica instalada el servidor Asterisk brindará al usuario los siguientes servicios:

- Efectuar llamadas desde un ordenador hacia la telefonía convencional.
- Permitirá toda la funcionalidad de una centralita convencional.



## 1.3.3 Producto

El producto entregable será, un servidor que actue como una centralita telefónica por medio de Asterisk quien tendrá incorporado una tarjeta electrónica que permita convertir las señales analógicas en digital y viceversa, para poder establecer la comunicación entre el ordenador y la línea tradicional. Este servidor con las siguientes características:

- Completo: Porque dispone de todas las funciones de una PBX.
- Flexibilidad: porque permite que las comunidades sean dueñas de su propio desarrollo y que puedan adaptarlas a sus propias necesidades.
- Confidencialidad: Porque al momento de recibir o realizar una llamada no serán escuchadas por terceras personas aunque se encuentren en la misma línea telefónica.

## 1.3.4 Impacto

Con la implementación de Asterisk las pequeñas y medianas empresas tendrán un sistema de comunicación de bajo coste, además proporcionara un servicio de telefonia en donde se efectuaran llamadas desde un ordenador hacia un teléfono convencional y se podrá tener un mejor control dentro de la empresa como por ejemplo quien está hablando con quien, líneas externas ocupadas, cuantas personas están esperando en cola para ser atendidas, entre otros.

## 1.4 Objetivos

## 1.4.1 Objetivo general

Evaluar si Asterisk es una opción viable para las pequeñas empresas que necesitan comunicación entre los dispositivos VoIP de una red de área local y la red telefónica conmutada.

## 1.4.2 Objetivos específicos

- Detallar los procedimientos para permitir que la PBX Asterisk se comunique con la red telefónica conmutada y viciversa.
- Determinar la calidad que ofrece esta implementación en las llamadas que se realizan desde los teléfonos VoIP hacia la red telefónica conmutada.
- Estimar las capacidades del servidor Asterisk en base a la cantidad de conexiones concurrentes que este es capaz de soportar.



# CAPÍTULO 2 MARCO TEÓRICO



## 2.1 Introducción a la telefonía

#### 2.1.1 La red telefónica conmutada

La red telefónica conmutada, o en la literatura inglesa PSTN, fue creada para transmitir la voz humana. Tanto por la naturaleza de la información a transmitir, como por la tecnología disponible en la época en que fue creada, esta es de tipo analógico. Hasta hace poco se denominaba RTC o Red Telefónica Conmutada, pero la aparición del sistema RDSI (digital pero basado también en la conmutación de circuitos), ha hecho que se prefiera utilizar la terminología RTB para la primitiva red telefónica (analógica), reservando las siglas RTC para las redes conmutadas de cualquier tipo (analógicas y digitales); así pues, la RTC incluye la primitiva RTB y la moderna RDSI (Red Digital de Servicios Integrados). RTB es en definitiva la línea que tenemos en el hogar o la empresa, cuya utilización ha estado enfocada fundamentalmente hacia las comunicaciones mediante voz, aunque cada vez más ha ido tomando auge el uso para transmisión de datos como fax, Internet, etc. (Introducción a la Telefonía)

La telefonía tradicional o fija es un sistema que permite hacer llamadas, un dispositivo de telecomunicación diseñado para transmitir señales vocales por medio de señales eléctricas, que funciona gracias a la RTC, red desarrollada para el transporte de señales sonoras pero que igualmente puede transferir datos. La red fija de telefonía asigna a cada abonado un número y le autoriza a realizar y recibir llamadas telefónicas, asistencia telefónica mediante operadora, conexión con las redes telefónicas móviles, transferencia de datos y funciones adicionales como contestador, llamada en espera o desvío de llamadas.

## 2.1.2 Private Branch Exchange (PBX)

Un PBX o PABX (Private Branch Exchange y Private Automatic Branch Exchange), es cualquier central telefónica conectada directamente a la red pública de telefonía por medio de líneas troncales para gestionar además de las llamadas internas, las entrantes y salientes con autonomía sobre cualquier otra central telefónica. Este dispositivo generalmente pertenece a la empresa que lo tiene instalado y no a la compañía telefónica, de aquí el adjetivo Privado a su denominación. (C.V, SISCOMEX)

El uso más común de una PBX es compartir de una a varias líneas telefónicas con un grupo de usuarios, tiene la propiedad de ser capaz de redirigir las llamadas entrantes a uno o varios teléfonos. De manera similar, permite a un teléfono escoger una de las líneas telefónicas para realizar una llamada al exterior, de la misma forma que un enrutador en Internet la PBX es responsable de dirigir los paquetes de un origen a su destino, es decir es el encargado de transmitir llamadas telefónicas. Tambien ofrece la posibilidad de crear servicios de valor añadido como transferencias de llamadas, llamadas a tres, pasarela de voz a



correo o servicios basados en una respuesta de voz interactiva (IVR), etc. (Escudero Pascual & Berthilson, Voz sobre IP, 2007)

## 2.2 Voice over Internet Protocol (VoIP)

Una definición general de Voz sobre IP (también conocida como telefonía IP) es la posibilidad de transportar conversaciones telefónicas en paquetes IP en lugar de los circuitos de transmisión telefónicos. Cuando hablamos de "VoIP", nos referimos a "la telefonía de internet" en el sentido más amplio de la expresión. El término VoIP no se refiere a ninguno de los mecanismos concretos que existen para llevar las señales de voz de un sitio a otro en la red. Las alternativas tecnológicas de VoIP se pueden dividir de una manera sencilla en dos grandes grupos: tecnologías cerradas-propietarias y sistemas abiertos. (Escudero Pascual & Berthilson, Voz sobre IP, 2007)



Figura 1: Telefonía Tradicional y VoIP



VoIP no es un servicio es una tecnología que permite la unión de dos mundos históricamente separados, el de la transmisión de voz y el de la transmisión de datos. Utilizando VoIP no existe solo una sola forma de realizar una llamada, veremos las distintas opciones que nos presenta esta tecnología:

## ATA (Analog Telephone Adaptor)

Este adaptador permite conectar teléfonos comunes (de los que utilizamos en la telefonía convencional) a su computadora o a su red para utilizarlos con VoIP. El adaptador ATA es básicamente un transformador de analógico a digital. Este toma la señal de la línea de teléfono tradicional y la convierte en datos digitales listos para ser transmitidos a través de internet.

## Teléfonos IP (hardphones)

Estos teléfonos a primera vista se ven como los teléfonos convencionales, con un tubo, una base y cables. Sin embargo los teléfonos ip en lugar de tener una ficha RJ-11 para conectar a las líneas de teléfono convencional estos vienen con una ficha RJ-45 para conectar directamente al router de la red y tienen todo el hardware y software necesario para manejar correctamente las llamadas VoIP.

## Llamadas Computadora a Computadora

Esta es la manera más fácil de utilizar VoIP, todo lo que se necesita es un micrófono, parlantes y una tarjeta de sonido, además de una conexión a internet preferentemente de banda ancha. Exceptuando los costos del servicio de internet usualmente no existe cargo alguno por este tipo de comunicaciones VoIP entre computadora y computadora, no importa las distancias. (Tipos de Comunicación en la Telefonia IP - Telefonia Voz IP). (Tipos de Comunicación en la Telefonia IP - Telefonia Voz IP)

## 2.2.1 Protocolos en la telefonía VolP

Existen varios protocolos comúnmente usados para VoIP, estos protocolos definen por ejemplo la manera en como los codecs se conectan entre si y hacia otras redes usando VoIP. Estos también incluyen especificaciones para codecs de audio.



#### 2.2.1.1 Protocolo H.323

El protocolo más usado es el H.323, un standard creado por la International Telecomunication Union (ITU), H.323 es un protocolo muy complejo que fue originalmente pensado para videoconferencias. Este provee especificaciones para conferencias interactivas en tiempo real, para compartir datos y audio como aplicaciones VoIP. Actualmente H.323 incorpora muchos protocolos individuales que fueron desarrollados para aplicaciones específicas. H.323 es una larga colección de protocolos y especificaciones. Eso es lo que lo permite ser usado en tantas aplicaciones. El problema con H.323 es que no fue específicamente dirigido a VoIP.

#### 2.2.1.2 Protocolo SIP

Una alternativa al H.323 surgió con el desarrollo del Session Initiation Protocol (SIP). SIP es un protocolo mucho más lineal, desarrollado específicamente para aplicaciones de VoIP. Más eficiente que H.323, SIP toma ventaja de los protocolos existentes para manejar ciertas partes del proceso.

SIP es similar al protocolo HTTP en muchos sentidos, incluso tiene algunos mensajes de error en común, como el "404 no encontrado" y el "403 servidor ocupado". Los componentes presentes en SIP son:

- Agentes de Usuario (User Agent, UA): Existen dos tipos de agentes de usuario, los cuales están presentes siempre, y permiten la comunicación cliente-servidor:
- a. Agente de usuario cliente (UAC): El UAC genera peticiones SIP y recibe respuestas.
- b. Agente de usuario servidor (UAS): El UAS responde a las peticiones SIP.
  - Servidores SIP: Existen tres clases lógicas de servidores. Un servidor puede tener una o más de las siguientes clases:
- a. Servidor de Redirección (Redirect Server): Reencamina las peticiones que recibe hacia el próximo servidor.
- b. Servidor Proxy (Proxy Server): Corren un programa intermediario que actúa tanto de servidor como de cliente para poder establecer llamadas entre los usuarios.
- c. Servidor de Registro (Registre Server): Hace la correspondencia entre direcciones SIP y direcciones IP. Este servidor solo acepta mensajes REGISTER, lo que hace fácil la localización de los usuarios, pues el usuario donde se encuentre siempre tiene que registrarse en el servidor.



Se define dos tipos de mensajes SIP: Peticiones y Respuestas:

- Peticiones SIP. Se definen seis métodos básicos:
- a. **INVITE:** Permite invitar un usuario a participar en una sesión o para modificar parámetros de una sesión ya existente.
- b. ACK: Confirma el establecimiento de la sesión.
- c. **OPTION:** Solicita información de algún servidor en particular.
- d. BYE: Indica término de una sesión.
- e. CANCEL: Cancela una petición pendiente.
- f. **REGISTER:** Registra al Agente de Usuario.
  - Respuestas SIP: Existen también mensajes SIP como respuesta a las peticiones. Existen seis tipos de respuestas, que se diferencian por el primer dígito de su código. Estas son:
- a. 1xx: Mensajes provisionales.
- b. 2xx: Respuestas de éxito.
- c. 3xx: Respuestas de redirección.
- d. **4xx:** Respuestas de fallas de método.
- e. 5xx: Respuestas de fallas de servidor.
- f. **6xx:** Respuestas de fallas globales.

## ✤ Ejemplo de un flujo SIP:





- a. Las dos primeras transacciones corresponden al registro de los usuarios. Los usuarios deben registrarse para poder ser encontrados por otros usuarios. En este caso, los terminales envían una petición REGISTER, donde los campos from y to corresponden al usuario registrado. El servidor Proxy, que actúa como Register, consulta si el usuario puede ser autenticado y envía un mensaje de OK en caso positivo.
- b. La siguiente transacción corresponde a un establecimiento de sesión. Esta sesión consiste en una petición INVITE del usuario al proxy. Inmediatamente, el proxy envía un TRYING 100 para parar las retransmisiones y reenvía la petición al usuario B. El usuario B envía un Ringing 180 cuando el teléfono empieza a sonar y también es reenviado por el proxy hacia el usuario A. Por último, el OK 200 corresponde a aceptar la llamada (el usuario B descuelga).
- c. En este momento la llamada está establecida, pasa a funcionar el protocolo de transporte RTP con los parámetros (puertos, direcciones, codecs, etc.) establecidos en la negociación mediante el protocolo SDP.



d. La última transacción corresponde a una finalización de sesión. Esta finalización se lleva a cabo con una única petición BYE enviada al Proxy, y posteriormente reenviada al usuario B. Este usuario contesta con un OK 200 para confirmar que se ha recibido el mensaje final correctamente.

Uno de los desafíos que enfrenta el VoIP es que los protocolos que se utilizan a lo largo del mundo no son siempre compatibles. Llamadas VoIP entre diferentes redes pueden meterse en problemas si chocan distintos protocolos. Como VoIP es una nueva tecnología, este problema de compatibilidad va a seguir siendo un problema hasta que se genere un standard para el protocolo VoIP. (Protocolos en la Telefonia IP, Protocolos VoIP - Telefonia Voz IP)

## 2.3 Protocolos de Transporte

RTP (Real-time Transport Protocol): Este protocolo define un formato de paquete para llevar audio y video a través de Internet. Este protocolo no usa un puerto UDP determinado, la única regla que sigue es que las comunicaciones UDP se hacen vía un puerto impar y el siguiente puerto par sirve para el protocolo de Control RTP (RTCP).

La inicialización de la llamada normalmente se hace por el protocolo SIP o H.323. El hecho de que RTP use un rango dinámico de puertos hace difícil su paso por dispositivos NAT y firewalls, por lo que se necesita usar un servidor STUN (Simple Traversal of UDP over NAT, RFC3489). STUN es un protocolo de red que permite a los clientes que estén detrás de un NAT saber su dirección IP pública, el tipo de NAT en el que se encuentran y el puerto público asociado a un puerto particular local por el NAT correspondiente. Esta información se usa para iniciar comunicaciones UDP entre dos hosts que están detrás de dispositivos de NAT.

Las aplicaciones que usan RTP son menos sensibles a la pérdida de paquetes, pero son típicamente muy sensibles a retardos, por lo que se usa UDP para esas aplicaciones. Por otro lado, RTP no proporciona calidad de servicio, pero este problema se resuelve usando otros mecanismos, como el marcado de paquetes o independientemente en cada nodo de la red.

RTCP (Real-time Transport Control Protocol): El protocolo de control RTP se basa en la transmisión de paquetes de control fuera de banda a todos los nodos participantes en la sesión. Tiene 3 funciones principales:

a. Provee realimentación en la calidad de la data.



- b. Utiliza nombres canónicos (CNAME) para identificar a cada usuario durante una sesión.
- c. Como cada participante envía sus tramas de control a los demás, cada usuario sabe el número total de participantes. Este número se usa para calcular la tasa a la cual se van a enviar los paquetes. Más usuarios en una sesión significan que una fuente individual podrá enviar paquetes a una menor tasa de bits.

## 2.4 VoIP y telefonía convencional: Algunas diferencias

La telefonía convencional posee características como transferencia automática de llamadas, llamada en espera, contestador automático, teleconferencia, números extras, entre otras. Cuando haces una conexión vía Internet, además de lo tradicional, también es posible enviar fotos y documentos a través de tu computadora, inclusive manteniendo una conversación en ese mismo instante. La tecnología VoIP además de todas esas ventajas sobre los servicios de telefonía común, está en constante perfeccionamiento, mejorando permanentemente la calidad de la conexión por Internet. En algunos casos ya es posible una calidad de sonido superior a la calidad de una conexión telefónica convencional. (Informática-Hoy, VoIP,ventajas y beneficios-Informatica Hoy)

En cuanto al aspecto comercial, la telefonía VoIP es en promedio 90% más económica que la telefonía tradicional y la calidad de la comunicación es casi igual.

Respecto a la parte técnica, en una llamada telefónica normal, la central telefónica establece una conexión permanente entre ambos interlocutores, conexión que se utiliza para llevar las señales de voz. En una llamada telefónica por IP, los paquetes de datos que contienen la señal de voz digitalizada y comprimida se envían a través de Internet a la dirección IP del destinatario. Cada paquete puede utilizar un camino para llegar, están compartiendo un medio, una red de datos. Cuando llegan a su destino son ordenados y convertidos de nuevo en señal de voz.

La diferencia principal entre las redes de voz (VoIP) y la red de conmutación de circuitos (PSTN), es que la conmutación de circuitos establece un circuito físico para la llamada que se está realizando entregando un circuito cerrado, mientras que en una de VoIP el medio es compartido para todos los paquetes de las diferentes llamadas que se estén efectuando, por ende los recursos en este tipo de red son reutilizables. (Ramos Bermeo, (PDF) Capitulo 1 Antecedentes de VoIP y Asterisk, 2009)



Telefonía VolP			
Ventajas:	Desventajas:		
• Se utiliza y mantiene una sola red, tanto para voz como de datos.	<ul> <li>Requerimiento de ancho de banda mínimo en enlaces compartidos.</li> </ul>		
Crecimiento sin necesidad de ampliar la centralita.	• Nat: El protocolo SIP es un protocolo "not Nat friendly.		
Integracion con aplicaciones.	· Control y poquetización: Los poquetos llogon		
<ul> <li>Teléfonos software y hardware.</li> </ul>	desordenados. Se tienen que controlar.		
Videoconferencia.	Latencia: Es el retardo que sufre un paquete en una comunicación entre dos puntos		
Menos costes de implementación.			
• Portabilidad.	Protocolo UDP para transporte. No retransmisión.		
(Casanova, Asterisk VoIP, Introducción a los sistemas VoIP)			

#### Tabla 1: Ventajas y Desventajas de la telefonía VoIP

## 2.5 Hardware para VolP

Con el estado de madurez que ha alcanzado la tecnología VoIP, hoy en día a la hora de seleccionar hardware, nos encontraremos con que hay una gran variedad de fabricantes y modelos de equipos de donde seleccionar. Cada uno con sus ventajas y desventajas, a la hora de configurar y obtener soporte técnico.

En los inicios de la VoIP, para un usuario utilizar una aplicación de voz sobre protocolo de internet VoIP, significa conectar un micrófono y unos auriculares a la computadora. En la actualidad, gracias a diversos dispositivos y adaptadores para teléfonos, es posible utilizar los servicios VoIP sin depender de la computadora y mediante un teléfono normal.

Debido al gran número de hardware para VoIP que existe, se han dividido en tres grupos principales:

- Adaptadores telefónicos: Módulos externos como pasarelas (Gateway), que sirve de puente entre las señales analógicas y digitales.
- Teléfonos IP: Conectados a una computadora o conectados directamente a la red VoIP.



Hardware interno para computadora: Diferentes tipos de tarjetas para la conexión de equipos analógicos y digitales, en una computadora. El hardware que se necesite, dependerá de la aplicación y la complejidad de la red VoIP. ([PDF]Capitulo I : Marco teórico conceptual).

## 2.5.1 Dispositivos FXO (Foreign Exchange Office)

La interfaz Foreign Exchange Office o FXO es el puerto por el cual se recibe a la línea Telefónica. Los puertos FXO cumple la funcionalidad de enviar una indicación de colgado o descolgado conocida como cierre de bucle. Un ejemplo de interfaz FXO es la conexión telefónica que tienen los teléfonos analógicos, fax, etc. Es por ello que a los teléfonos analógicos se les denomina "dispositivos FXO".



Figura 3: Dispositivo FXO

## 2.5.2 Dispositivos FXS (Foreign Exchange Station)

Esta tarjeta de Interfaz permite conectar teléfonos análogos o tradicionales a un computador, en este caso el Servidor IP. De esta manera, se pueden realizar y recibir llamadas desde teléfonos análogos tanto hacia el interior de la red LAN (ya sea a Softphone, Teléfonos IP o Teléfonos Análogos conectados a la Tarjeta FXS) o el exterior de esta red, como puede ser la PSTN u otra Red IP

A modo de resumen se quiere destacar que dos puertos se pueden conectar entre sí con la condición de ser de distinto tipo, es decir, FXO y FXS son siempre pareja (similar a un enchufe (macho/hembra). (Introducción a la Telefonía)





En la figura 4. Se muestra el escenario de un hogar tradicional. Como podemos apreciar siempre se conectan entre sí interfaces de distintos tipo, es decir, FXS con FXO o viceversa. El teléfono posee una interfaz FXO como se muestra en la imagen, el cual es conectado a la roseta de la compañía telefónica FXS.

## 2.5.3 Diferencia entre FXO y FXS

Un Foreign Exchange Office (FXO) es cualquier dispositivo que, desde el punto de vista de la central telefónica, actúa como un teléfono tradicional. Un FXO debe ser capaz de aceptar señales de llamadas o ring, ponerse en estado de colgado o descolgado, y enviar y recibir señales de voz. Asume que un FXO es como un "teléfono" o cualquier otro dispositivo que "suena", mientras que un Foreign Exchange Station (FXS) es lo que está situado al otro lado de una línea telefónica tradicional (la estación). Un FXS envía el tono de marcado, la señal de llamadas que hace sonar los teléfonos y los alimentas. En líneas analógicas, un FXS alimenta al FXO. (Escudero Pascual & Berthilson, Voz sobre IP, 2007)

## 2.6 Software de telefonía VoIP

Actualmente existen una gran cantidad de empresas dedicadas específicamente a implementar los servicios de telefonía IP dentro de otras empresas, sin embargo al buscar una solución robusta y de bajo coste nos encontramos con Asterisk el cual es hoy en día uno de los programas más utilizados por ser un software PBX libre y flexible. Éste fue el software que utilizamos en nuestra implementación y por tal razón conoceremos más de funcionamiento de Asterisk.

## 2.6.1 ¿Qué es Asterisk?

Es una implementación libre de una centralita telefónica. El programa permite tanto que los teléfonos conectados a la centralita puedan hacer llamadas entre ellos como servir de pasarela a la red telefónica



tradicional. El código del programa fue originalmente creado por Mark Spencer (Digium) basado en las ideas y el trabajo previo de Jim Dixon (Proyecto de telefonía Zapata). Aunque Asterisk puede funcionar en muchos sistemas operativos, GNU/Linux es la plataforma más estable y en la que existe un mayor soporte. (Escudero Pascual & Berthilson, Voz sobre IP, 2007)

## 2.6.1.1 Servicios que ofrece Asterisk

Transferencias de llamadas, internas y externas.	
Desvío de llamadas si está ocupado o no contesta.	
Registro y listados de llamadas entrantes y salientes, con gráficas de consumo.	
Llamada en espera (Hold).	
Grupos de llamadas (Ring Groups).	
Identificador de llamante (CallerID).	
Operadora digital (menús interactivos y guiados).	
Grabación de llamadas entrantes y salientes.	
Gestión de listas negras (números telefónicos con acceso prohibido).	
Monitorización de llamadas en curso.	
Salas de conferencia (2 o más terminales simultáneamente).	
Recepción de fax desde el propio sistema y posterior envío por e-mail.	
Soporta videoconferencia con protocolos SIP e IAX2.	
Buzones de Voz (general, individuales, por grupos) protegidos por contraseña.	
Sistema DISA12. (método por el cual una persona externa a la oficina puede realizar llamadas a través de la centralita.	

#### Tabla 2: Servicios que ofrece Asterisk.

(Casanova, Asterisk VoIP, Introducción a los sistemas VoIP)



#### 2.6.1.2 Beneficios de Asterisk como PBX

- a. Ampliación del número de anexos sin adquirir otra PBX.
- b. Funcionalidades avanzadas para teléfonos analógicos existentes.
- c. Administración local y remota a través de la Web.
- d. Asignación de categoría de servicios, locales, celulares.
- e. Cantidad ilimitada de anexos y usuarios.
- f. Convergencia de redes, la red de telefonía con la red.

## 2.6.1.3 Beneficios de Asterisk como Servidor VolP

- a. Disminución de costos en llamadas telefónicas entre sucursales.
- b. Troncalización con otras oficinas (sucursales) o instituciones a "costo cero" en comunicación a través de la línea de comunicación por Internet.
- c. Conexiones remotas a través de autenticación de usuarios.
- d. Encriptación en el momento de conexión y durante toda la comunicación.

## 2.6.2 Ficheros de configuración de Asterisk

A continuación se citan los principales ficheros de configuración en Asterisk:

El fichero **sip.conf** contiene parámetros relacionados con la configuración SIP de Asterisk. En este fichero se definen variables generales, clientes y servidores SIP; se estructura en secciones donde cada sección se define por un nombre entre corchetes seguido de las opciones de cada sección.

El fichero **extensions.conf** contiene el dialplan del Asterisk, el plan maestro de control o de flujo de ejecución para todas las operaciones. Controla cómo se manejan y se encaminan las llamadas entrantes y salientes. Aquí es donde se configura el comportamiento de todas las conexiones con su PBX.

El contenido de "extensions.conf" se organiza en secciones, que pueden ser tanto para configuraciones estáticas y definiciones, o para los componentes ejecutables del dialplan a los cuales nos referimos con el nombre de contextos. Las secciones de configuraciones son **general**, **globals** y los nombres de los contextos son definidos enteramente por el administrador de sistema.



El termino **contexto** se usa para definir que reglas o grupos de reglas del plan de marcado (extension.conf) se deben aplicar a esa llamada concreta. El contexto representa el punto de entrada de la llamada en el plan de marcado.

El fichero voicemail.conf es donde se configura todo lo relacionado con el contestador. Si recibimos una llamada y no contestamos o la línea está ocupada, entrará en función el contestador.

## 2.6.3 Integración de Asterisk a la red telefónica conmutada

## 2.6.3.1 Asterisk- Estándares Abiertos y Código Libre

Los estándares abiertos permiten que cualquiera pueda implementar un sistema con garantías de interoperabilidad. Gracias a esa interoperabilidad de nuestro diseño no sólo podemos crear nuestra red telefónica sino que, además, podemos conectarla a la red telefónica global. (Escudero Pascual & Berthilson, Voz sobre IP, 2007)

## 2.6.3.2 Conexión de Asterisk con la PSTN

Asterisk es una aplicación software libre de una central telefónica (PBX). Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o bien a una RDSI tanto básicos como primarios.

Para conectar teléfonos normales analógicos hacen falta unas tarjetas telefónicas FXS o FXO Digium u otros fabricantes, ya que para conectar el servidor a una línea externa no vale con un simple módem, por tal motivo para que nuestra PBX VoIP con Asterisk se conecte a la línea convencional le hemos incorporado la tarjeta Authentic X100P SE FXO PCI al servidor: En la figura 5 podemos ver la tarjeta. (Solutec, Asterisk-Central Telefónica PBX)





Figura 5: Tarjeta Authentic X100P SE FXO PCI

El X100P proporciona una única interfaz FXO con todas las funciones para conectar el servidor de código abierto Asterisk PBX a la PSTN (Public Switched Telephone Network).

X100P Authentic permite que Asterisk realice o reciba llamadas de una línea telefónica analógica tradicional. El X100P es un componente económico e ideal para la construcción de Respuesta de Voz Interactiva (IVR) y aplicaciones de correo de voz.

Es compatible con todas las características estándar de llamadas mejorada para cualquier dispositivo FXO incluyendo identificador de llamadas, llamada en conferencia y llamada en espera / identificador de llamadas. También cuenta con la última revisión de los chipsets de arreglo de acceso a datos (DAA) originales con numerosas correcciones de errores.

Con la tarjeta X100P integrada a Asterisk PBX, se puede fácilmente, implementar económicamente los servicios de llamadas sofisticados y flexibles. Estos servicios van desde el menú de voces interactivas (IVR), puertas de enlace VoIP de múltiples protocolos, correo de voz (voicemail), etc. (Authentic X100p Fxo Digium Asterisk Voip Pbx Centralilla)



SIPp es una aplicación Open Source, de análisis de rendimiento del protocolo SIP. Es la principal herramienta empleada, capaz de realizar múltiples llamadas de forma simultánea empleando dicho protocolo. SIPp también puede enviar tráfico RTP (audio/vídeo) a través del reenvío de tráfico RTP capturado en un archivo PCAP. SIPp muestra en pantalla los resultados estadísticos de las pruebas que se estén ejecutando (ratio de llamadas por segundo, retraso de llegada de paquetes y estadísticas sobre cada uno de los mensajes del protocolo SIP).

## 2.7.1 Funcionamiento de SIPp

El funcionamiento de SIPp es de tipo cliente/servidor, su uso se basa en definir una serie de escenarios para el cliente, y otros para el servidor, donde en cada uno de estos se establece el flujo SIP que se desea generar (flujo de una petición INVITE).

#### 2.7.1.1 Cliente SIPp

El cliente SIPp creará las llamadas que componen la carga de trabajo del servidor Asterisk, además contendrá un archivo .xml en donde se indica el flujo de audio pcap que estaremos enviando. Las llamadas que el cliente efectuará no serán transmitidas de un solo al servidor Asterisk, primeramente pasarán por el servidor SIPp.

#### 2.7.1.2 Servidor SIPp

El servidor SIPp es el encargado de recibir las llamadas creadas por el cliente SIPp, igualmente contendrá un archivo .xml en donde se establece que se debe esperar tanto el tráfico SIPp como el tráfico RTP en la dirección IP asignada al servidor SIPp.

La diferencia entre actuar como cliente o servidor SIPp depende del escenario de configuración del equipo que se haga uso.


## CAPÍTULO 3 DISEÑO METODOLÓGICO

En esta sección se enumeran los diferentes materiales empleados en este trabajo y se describen las etapas a través de las cuales se abordó nuestra investigación.

## 3.1 Materiales utilizados:

### 3.1.1 Hardware

Los materiales hardware empleados en nuestro trabajo fueron:

Material	Descripción	Costo aproximado
	Doble procesador Intel	\$200
PC de Escritorio.	@ 2.20 GHZ. Memoria RAM DDR2 de 2GB. Esta PC	
	actuará como PBX utilizando Asterisk.	
	Wireless ADSL2+MODEM	\$ 50
	Model: ET5300	
Un Modem ADSL Eltel.	Input: 12 VDC / 1A – Output:0.40A	
	Part:V1	
	Son tarjetas compatibles con el sistema Asterisk, las	\$ 60
Tarjeta Authentic X100P	cuales permiten implementar de manera muy	
SE FXO PCI.	económica un sistema de PBX tradicional.	
	Empleados para interconectar los diferentes	\$5
4 Cables UTP Cat. 5	dispositivos de red.	
	Utilizadas como usuarios VoIP, cada máquina tiene	\$400 - \$600
3 Laptops.	distintas características y diferentes marcas. La	
	memoria RAM consta entre 2 y 4 Gigas y el	
	procesador consta entre 1.80 GHZ y 2.13GHZ.	

#### Tabla 3 : Materiales Hardware



## 3.1.2 Software

Software	Descripción		
Sistema operativo CentOs 6.4	Arquitectura x86_64, Fácil mantenimiento,		
	idoneidad para el uso a largo plazo en entornos de		
	producción, desarrollo activo de módulos y		
	aplicaciones, diseñado para servidores, seguridad y		
	estabilidad.		
Asterisk 11.7.0	Es un programa de software libre (bajo licencia		
	GPL) que proporciona funcionalidades de una		
	central telefónica (PBX)		
	Tecnología de interfaz de dispositivo de código		
Dahdi 2.9.0	abierto desarrollada por Digium utilizado para el		
	control de sus dispositivos y otras tarjetas de		
	interfaz de telefonía.		
SJphone	Es un softphone usado como cliente VoIP.		
SIPp 3.4.0	Es una aplicación Open Source, de análisis de		
	rendimiento del protocolo SIP.		

## 3.2 Etapas del proyecto

Para abordar y cumplir todos nuestros objetivos, se dividió el trabajo en las siguientes etapas:

### 3.2.1 Etapa I: de exploración

En esta etapa se recopilo información acerca del tema, haciendo búsquedas en libros, sitios web, etc.

### 3.2.2 Etapa II: Integración de Asterisk a la red telefónica conmutada

En esta etapa se integró Asterisk a la red telefónica conmutada empleando para ello la tarjeta FXO. Para esto se realizaron una serie de sub-etapas que se describen a continuación:



#### 3.2.2.1 Comunicación entre usuarios VolP

En esta prueba se usaron dos clientes Windows VoIP para que se comuniquen entre sí a través del servidor Asterisk.

## 3.2.2.2 Llamadas desde usuarios VoIP a la red telefónica conmutada empleando la tarjeta FXO

Utilizando el servidor Asterisk con la tarjeta FXO integrada y conectado a la línea telefónica conmutada se estableció comunicación entre un cliente VoIP con un número valido dentro de la red telefónica convencional, del mismo modo cuando exista una llamada entrante en la PSTN esta sea redirigida a una extensión de uno de los clientes VoIP registrado.

#### 3.2.2.3 Creación de un IVR para las llamadas entrantes desde la red telefónica conmutada

Se creó un menú de respuesta inmediata (IVR) para que el usuario elija una de las opciones a la cual será redirigido y si no elige nada este será comunicado con la operadora que en nuestro caso, será una extensión de uno de los clientes la cual se encargara de transferir la llamada. Si acaso presiona una opción inválida de nuevo se le reproduzca el menú.

#### 3.2.3 Etapa III: Evaluación de la calidad de la llamada

En esta etapa se evaluó la calidad de la llamada tomando en cuenta distintos factores como: el retardo, la interferencia, el volumen y el eco.

#### 3.2.4 Etapa IV: Realización de pruebas de rendimiento de Asterisk

Para probar el rendimiento del servidor Asterisk se empleó un generador de tráfico SIP que permitiera someter al servidor Asterisk a una alta carga de trabajo. El simulador de tráfico SIP empleado fue el SIPp, las pruebas realizadas se desglosan a continuación:

#### 3.2.4.1 Máximo número de conexiones concurrente en un escenario ideal

Se determinó el número de llamadas simultáneas VoIP que puede soportar el servidor Asterisk.



#### 3.2.4.2 Validación de los resultados en un escenario con carga en la red y en los equipos

Se evaluó si el rendimiento del servidor Asterisk es estable cuando existe carga en la red y se verificó si soporta el mismo número de conexiones concurrentes.

#### 3.2.5 Etapa V: Redacción del informe final

Redacción y organización del documento final, el cual contiene la explicación de cada una de las etapas de este proyecto y solución de cada uno de los objetivos planteados.



## CAPÍTULO 4 DESARROLLO



## 4.1 Integración de Asterisk a la red telefónica conmutada

Con Asterisk integrado a la red telefónica conmutada empleando para ello la tarjeta FXO. Se realizaron una serie de etapas que se describen a continuación:

#### 4.1.1 Comunicación entre usuarios VolP

En este primer escenario se logró la comunicación entre clientes VoIP a través del servidor Asterisk y se realizó lo siguiente:

- Creación de las cuentas sip en el fichero sip.conf.
- Realizar el plan de marcado en extensions.conf.
- Stablecer el buzón de voz de cada cliente en voicemail.conf.
- Configuración de los clientes VoIP utilizando un Softphone estos pueden ejecutarse en Linux o Windows.
- Establecer llamadas entre los Softphone a través de la PBX.



Figura 6: Esquema de Red únicamente para clientes VoIP



Para el desarrollo de este apartado utilizamos una PC que actuó como servidor el cual tenía un sistema operativo con una distribución Linux, en este caso usamos CentOS versión 6.4 debido a que tiene mejor compatibilidad con las tarjetas hardware que se le pueden integrar al servidor, por el lado del cliente se instaló un softphone el SJphone que está disponible en las plataformas Windows, Linux o Mac.

En el servidor Asterisk se modificaron los siguientes ficheros de configuración sip.conf, extensions.conf y voicemail.conf (Ver Anexo 2: Configuración de los archivos sip.conf, extensions.conf y voicemail.conf)

Una vez configurado todos los archivos antes mencionados nos situamos en la consola de Asterisk

#Asterisk --vvvc

Y luego ejecutamos el siguiente comando

\*CLI> sip reload *II* recarga nuestro fichero sip.conf

Por el lado del cliente se instaló y configuró el softphone SJphone (Ver Anexo 3: Instalación del Softphone SJphone (en el lado del cliente)). Una vez instalado el SJphone seguiremos con el proceso de configuración para los usuarios de la red, no olvidando que dichos usuarios fueron previamente declarados en el servidor Asterisk. Se configurarán tantos usuarios como se tengan previamente declarados.

Es importante destacar que esta configuración del SJphone se realiza de acuerdo a las cuentas que tengamos definidas en el fichero sip.conf de nuestro servidor Asterisk.

Volvemos a nuestro servidor y dentro de la consola de Asterisk ejecutamos.

#sip show peers //Verifica que los clientes estén registrados correctamente en Asterisk



Si deseamos comunicarnos con nuestro cliente desde el servidor Asterisk lo realizamos con el siguiente comando:

#console dial 2001@miprimerejemplo //Realiza una llamada al usuario 2001

Ahora es posible comunicarse con otros clientes que estén utilizando un softphone y que en el servidor estén debidamente registrados. En la siguiente imagen se muestra al usuario2 con extensión 2001 realizando una llamada al usuario3 con extensión 2002.



Figura 7:Usuario2 llamando al Usuario3

En el SJphone del usuario 3 se visualizará la siguiente imagen con dos opciones aceptar y rechazar la llamada.



Figura 8: SJphone del Usuario3 donde puede aceptar o rechazar la llamada



Al aceptar la llamada el Usuario3 en la pantalla del SJphone del Usuario 2 y 3 se mostrará que la llamada se ha establecido correctamente.



Figura 9: Llamada establecida entre clientes VoIP

Mientras tanto por el lado del servidor podemos monitorizar el proceso de la llamada.



Figura 10: Proceso de la llamada en el lado del servidor



# 4.1.2 Llamadas desde usuarios VoIP a la red telefónica conmutada empleando la tarjeta FXO

En este escenario se configuró el servidor Asterisk incorporándole una tarjeta FXO-FXS, de esta forma lograremos comunicarnos con la línea telefónica convencional o PSTN.



Figura 11: Esquema de Red del Servidor Asterisk conectado a la PSTN

Al realizar la conexión del servidor Asterisk a la PSTN se hizo lo siguiente:

- a) Instalación de la tarjeta FXO/FXS.
- b) Modificar el dialplan para que:
  - un cliente se comunique con un número valido dentro de la red telefónica.
  - Al tener una llamada entrante a nuestra PSTN este sea redirigido a una extensión de un cliente.
- c) Comprobar su debido funcionamiento.



En la siguiente imagen podemos observar la conexión de las entradas de la tarjeta.



Figura 12: Conexiones de las entradas de la tarjeta X100p

Ya instalado y configurado ahora viene la instalación y configuración de la tarjeta Motorola Wildcard PCI X100P (Ver Anexo 4: Instalación y Configuración de la tarjeta Authentic X100P SE FXO PCI). Con la tarjeta instalada podemos observar en la siguiente ilustración que el sistema detecta un puerto analógico (FXO) de telefonía e informática que es compatible con el sistema Asterisk, las cuales te permiten implementar de manera muy económica un sistema de PBX tradicional para su hogar u oficina. Con el siguiente comando se comprobó que efectivamente la tarjeta ha sido reconocida por el sistema: # lspci –v

01:01.0	Communication controller: Motorola Wildcard X100P Subsystem: Efar Microsystems Device 0001
	Flags: bus master, medium devsel, latency 64, IRQ 18
	I/O ports at e800 [size=256]
	Memory at febff000 (32-bit, non-prefetchable) [size=4K]
	Capabilities: [40] Power Management version 2
	Kernel driver in use: wcfxo
	Kernel modules: wcfxo
[root@l	ocalhost ~]# lspci   grep X100P
91:01.0	Communication controller: Motorola Wildcard X100P





Como podemos notar nos sale que el estado de nuestra tarjeta está inactivo RED pero cuando tenemos conectado el servidor con la PSTN el estado de la tarjeta dice activo battery.

[roo	ot@loca	alho	ost ~]# /	/etc/init.d	d/dahdi	i statı	ıs		
###	Span	1:	WCFX0/0	"Wildcard	X100P	Board	1"	(MASTER)	
1	FX0		FXSKS	(In	use)	(EC: MO	<u>5</u> 2 -	- ACTIVE)	(battery)

#### Figura 14: Estado de la tarjeta conectada a la PSTN

Iniciamos manualmente dahdi

# /etc/init.d/dahdi start

[root@localhost ~]# / Loading DAHDI hardwar	etc/init.d/dahd: e modules:	i start			
wcfxo:			[	0K	]
Running dahdi_cfg:	-		[	0K	]

Figura 15: Iniciamos el servicio DAHDI

Una vez que la tarjeta está funcionando correctamente y con las cuentas de los usuarios creadas en el fichero sip.conf nos dirigimos al fichero extensions.conf y creamos un plan de marcado salientes hacia la PSTN.

[privilegiados]

// Para extensiones locales

exten => \_20XX,1,Dial(SIP/\${EXTEN},8,Ttm)

exten => \_20XX,n,VoiceMail(\${EXTEN}@buzon)

exten => \_20XX,n,Hangup()

// Para llamadas salientes a la red PSTN

exten => \_NXXXXXXX,1,Dial(DAHDI/1/\${EXTEN},10) // N indica que puede ser cualquier número y X indica la cantidad de

dígitos a marcar.

exten => \_NXXXXXX,n,Hangup() //Corta la comunicación.



Guardamos. Entramos en la consola de Asterisk

#Asterisk --vvvc

Recargamos nuestro plan de marcado

\*CLI> dialplan reload

Si deseamos visualizar nuestro dialplan.

\*CLI> dialplan show privilegiados

*CLI> dialplan sho	w privilegiados				
[ Context 'privilegiados' created by 'pbx config' ]					
'_20XX' =>	<ol> <li>Dial(SIP/\${EXTEN},8,Ttm)</li> </ol>	[pbx_config]			
—	<ol><li>VoiceMail(\${EXTEN}@buzon)</li></ol>	[pbx_config]			
	3. Hangup()	[pbx_config]			
'_NXXXXXXX' =>	<ol> <li>Dial(DAHDI/1/\${EXTEN},10)</li> </ol>	[pbx_config]			
	2. Hangup()	[pbx_config]			
-= 2 extensions (5	nriorities) in 1 context =-				

#### Figura 16: Plan de marcación del contexto privilegiados

Posteriormente haremos un plan de marcado de llamadas entrantes. Definimos un contexto que se llame from-pstn. Aquí indicamos que hacer cuando recibamos algo de la línea análoga o digital.

[from-pstn]

exten => s,1,Answer() //Contesta la llamada y abre el canal

exten => s,n,Dial(SIP/2001,10) /La llamada es direccionada a una extensión interna

exten => s,n,Hangup() //Cuelga la llamada



Guardamos y entramos en la consola de Asterisk

#Asterisk -vvvc

Recargamos nuestro plan de marcado y lo visualizamos

\*CLI> dialplan reload

\*CLI> dialplan show from-pstn //Visualizamos el dialplan



Figura 17: dialplan from-pstn

Se probó la comunicación entre el servidor y la PSTN. En la siguiente figura se puede observar que desde el usuario 2001 se está llamando a un número telefónico y que la llamada fue contestada y posteriormente colgada. También se refleja que el mismo usuario llamó a un número móvil y sucedió lo mismo.



Figura 18: Monitorización de las llamadas en el servidor



Posteriormente se hace la prueba desde un número de celular llamando al número de la PSTN 2311-1681, repica el teléfono y la llamada es direccionada a una extensión interna de la red VoIP, la llamada es contestada por el usuario que le pertenece esa extensión y se puede comunicar con la persona que está llamando. Como se puede apreciar en la consola de Asterisk inicialmente la llamada es contestada, luego se direcciona a la extensión 2001 de la red VoIP, además se observa que en esa extensión el SJphone está repicando, la llamada fue atendida y al rato fue colgada.



Figura 19: Llamada entrante desde la PSTN

Por el lado del cliente: Aquí la llamada ya ha sido dirigida a la extensión interna y se puede visualizar en el softphone el número de quien está llamando, la comunicación se realizó satisfactoriamente

۲	Incoming call from "086406597" From IP address 192.168.1.7 : 5060	Accept Ignore	Coperational (00:04) (GSM)
			Nute Transfer CONE (20)>>> SAVE 086406597 27/03/2014 02:06:18 p.m., not at
			086406597 27/03/2014 02:03:42 p.m., not ar Usuario2 24/03/2014 11:06:31 a.m., not ar

Figura 20: Llamada entrante desde la PSTN redirigida a una extensión interna

## 4.1.3 Creación de un IVR para las llamadas entrantes desde la red telefónica conmutada

Configurar un IVR a la entrada de la FXO que realice lo siguiente:

- Si presiona 1 o 2 se redirija a la extensión indicada.
- Si presiona "\*" o una opción invalida se le reproduzca de nuevo el menú de voz.
- Si no presiona nada la llamada sea atendida por la operadora.



Figura 21: Escenario de red donde se crea el IVR

IVR son las siglas de Interactive Voice Response, que se traduce del inglés como Respuesta de Voz Interactiva. Es un sistema automatizado de respuesta interactiva, orientado a entregar y/o capturar información a través del teléfono, además permite que la persona que llama transfiera la llamada a la extensión deseada haciendo uso de menús interactivos, activados mediante la marcación del teclado del teléfono. Se deberá configurar una IVR a la entrada de la FXO el cual realice lo siguiente:



Reproduzca el archivo de audio del menú IVR que tendrá las siguientes opciones:

- > Tiempo de espera para que marque.
- si el usuario marca 1 o 2 la llamada debe de direccionarse a la extensión indicada donde opción 1 es el departamento de ventas que será la extensión 2000 y opción 2 el departamento de soporte que será la extensión 2002.
- > si el usuario marca \* la llamada es direccionada al menú del IVR.
- se debe considerar que si el usuario no marca nada en el tiempo de espera la llamada sea direccionada a la operadora que será la extensión 2001.
- > Si el usuario presiona una opción incorrecta entonces le indicara un mensaje de opción inválida.

En el fichero **extensions.conf** tenemos:

## [menu-dia] exten => s,1,Answer() //Contestamos la llamada y abrimos el canal exten => s,n,Background(menu\_1) //Escuchamos las opciones del menú exten => s,n,WaitExten(5) //Tiempo de espera por la extensión //OPCIONES exten => 1,1,Goto(miprimerejemplo,2000,1) //Si elige 1 la llamada es direccionada al 2000 exten => 2,1,Goto(miprimerejemplo,2002,1) //Si elige 2 la llamada es direccionada al 2002 exten => \*,1,Goto(s,1) //si elige \* la llamada es direccionada al menú del IVR exten => t,1,Playback(operadora) //si en el tiempo de espera no presiona nada entonces la llamada es dirigida a la operadora en este caso a la extensión 2001 exten => t,n,Goto(miprimerejemplo,2001,1) //Llamada direccionada a la operadora exten => t,n,Hangup() exten => i,1,Playback(pbx-invalid) //Mensaje que le dice que usted presiono algo invalido exten => i,n,Goto(s,1) //Como marco algo incorrecto entonces le vuelve a salir el menú del IVR [privilegiados] exten => \_20XX,1,Dial(SIP/\${EXTEN},8,Ttm) exten => \_20XX,n,VoiceMail(\${EXTEN}@buzon) exten => \_20XX,n,Hangup() exten => NXXXXXX,1,Dial(DAHDI/1/\${EXTEN},10) exten => \_NXXXXXX,n,Hangup() exten => 2005,1,Goto(menu-dia,s,1) //Creamos esta extensión que es la que va a contener el IVR





include => miprimerejemplo

[from-pstn] //Cuando nos llamen a nuestra línea

exten => s,1,Answer() //Contestamos y abrimos el canal

exten => s,n,Goto(menu-dia,s,1) //Lo redirigimos al contexto

menu-dia que es donde está el IVR

include => menu-dia

include => privilegiados

[miprimerejemplo]

exten => 2002,1,Dial(SIP/2002,30,Ttm)

exten => 2002,2,Voicemail(2002@buzon,u)

exten => 2002,3,Hangup()

exten => 2001,1,Dial(SIP/2001,30,Ttm)

exten => 2001,2,Voicemail(2001@buzon,u)

exten => 2001,3,Hangup()

- exten => 2000,1,Dial(SIP/2000,30,Ttm)
- exten => 2000,2,Hangup()
- exten => 2000,3,Hangup()
- exten => \*98,1,Answer
- exten => \*98,2,Wait(1)
- exten => \*98,3,VoiceMailMain(\${CALLERID(num)}@buzon)
- exten => \*98,4,Hangup
- ;exten para comunicación
- exten => 3000,1,VoiceMailMain(2000@buzon)
- exten => 3001,2,VoiceMailMain((2001@buzon)

Guardamos. Entramos en la consola de Asterisk



Recargamos nuestro plan de marcado

\*CLI> dialplan reload

Si deseamos visualizar nuestro dialplan

\*CLI> dialplan show from-pstn

*CLI> dialplan [ Context 'from	show from-pstn u-pstn' created by 'pby config' l	
's' =>	1. Answer() 2. Goto(menu-dia s 1)	[pbx_config] [pbx_config]
Include => Include =>	'menu-dia' 'privilegiados'	[pbx_config] [pbx_config] [pbx_config]
-= 1 extension	(2 priorities) in 1 context. =-	

Figura 22: Visualizando dialplan from-pstn

Mediante consola CLI visualizamos la configuración del menú

\*CLI> dialplan show menu-dia

*CLI> dial	plan show menu-dia	
[ Context	'menu-dia' created by 'pbx config' ]	
'*' =>	1. Goto(s,1)	[pbx config]
'1' =>	<ol> <li>Goto(miprimerejemplo,2000,1)</li> </ol>	[pbx_config]
'2' =>	<ol> <li>Goto(miprimerejemplo,2002,1)</li> </ol>	[pbx config]
'i' =>	<ol> <li>Playback(pbx-invalid)</li> </ol>	[pbx_config]
	<ol><li>Goto(s,1)</li></ol>	[pbx config]
's' =>	1. Answer()	[pbx_config]
	<ol><li>Background(menu_1)</li></ol>	[pbx_config]
	<ol> <li>WaitExten(5)</li> </ol>	[pbx_config]
't' =>	<ol> <li>Playback(operadora)</li> </ol>	[pbx config]
	<ol><li>Goto(miprimerejemplo,2001,1)</li></ol>	[pbx_config]
	3. Hangup()	[pbx_config]
-= 6 exten	sions (11 priorities) in 1 context. =-	

Figura 23: Plan de marcación del contexto menu-dia



A continuación visualizamos el plan de marcación del contexto

\*CLI> dialplan show privilegiados

*CLI> dialplan sho [ Context 'privile	w privilegiados egiados' created by 'pbx config'	1
'2005' =>	1. Goto(menu-dia,s,1)	[pbx config]
' 20XX' =>	<ol> <li>Dial(SIP/\${EXTEN},8,Ttm)</li> </ol>	[pbx config]
—	<ol><li>VoiceMail(\${EXTEN}@buzon)</li></ol>	[pbx config]
	3. Hangup()	[pbx_config]
'_NXXXXXXX' =>	<ol> <li>Dial(DAHDI/1/\${EXTEN},10)</li> </ol>	[pbx_config]
	2. Hangup()	[pbx_config]
Include =>	'miprimerejemplo'	[pbx_config]
-= 3 extensions (6	5 priorities) in 1 context. =-	

Figura 24: Dialplan del contexto privilegiados

\*CLI> dialplan show miprimerejemplo

weite dielelen eb		
*CLI> dialplan sn	low miprimerejempio	
[ Context 'miprim	erejemplo' created by 'pbx config' ]	
'*98' =>	1. Answer()	[pbx_config]
	2. Wait(1)	[pbx_config]
	<ol><li>VoiceMailMain(\${CALLERID(num)}@buzon)</li></ol>	[pbx_config]
	4. Hangup()	[pbx_config]
'2000' =>	1. Dial(SIP/2000,30,Ttm)	[pbx_config]
	2. Hangup()	[pbx_config]
	3. Hangup()	[pbx_config]
'2001' =>	<ol> <li>Dial(SIP/2001,30,Ttm)</li> </ol>	[pbx_config]
	<ol><li>Voicemail(2001@buzon,u)</li></ol>	[pbx_config]
	3. Hangup()	[pbx_config]
'2002' =>	1. Dial(SIP/2002,30,Ttm)	[pbx_config]
	<ol><li>Voicemail(2002@buzon,u)</li></ol>	[pbx_config]
	3. Hangup()	[pbx_config]
'3000' =>	<ol> <li>VoiceMailMain(2000@buzon)</li> </ol>	[pbx_config]
'3001' =>	<ol><li>VoiceMailMain((2001@buzon)</li></ol>	[pbx_config]
-= 6 extensions (	15 priorities) in 1 context. =-	

Figura 25: Plan de marcación del contexto miprimerejemplo

Comprobación del IVR llamando desde un móvil:

a) Cuando el usuario presiona la opción 1 que es el departamento de ventas con extensión 2000

*CLI> Starting simple switch on 'DAHDI/1-1'
Executing [s@from-pstn:1] Answer("DAHDI/1-1", "") in new stack
Executing [s@from-pstn:2] Goto("DAHDI/1-1", "menu-dia,s,1") in new stack
Goto (menu-dia,s,1)
Executing [s@menu-dia:1] Answer("DAHDI/1-1", "") in new stack
Executing [s@menu-dia:2] BackGround("DAHDI/1-1", "menu_1") in new stack
<dahdi 1-1=""> Playing 'menu 1.gsm' (language 'en')</dahdi>
Executing [1@menu-dia:1] Goto("DAHDI/1-1", "miprimerejemplo,2000,1") in n
ew stack
Goto (miprimerejemplo,2000,1)
Executing [2000@miprimerejemplo:1] Dial("DAHDI/1-1", "SIP/2000,30,Ttm") i
n new stack
[Mar 31 17:36:26] WARNING[3627][C-00000003]: app_dial.c:2437 dial_exec_full: Una
ble to create channel of type 'SIP' (cause 20 - Subscriber absent)
== Everyone is busy/congested at this time (1:0/0/1)
Executing [2000@miprimerejemplo:2] Hangup("DAHDI/1-1", "") in new stack
== Spawn extension (miprimerejemplo, 2000, 2) exited non-zero on 'DAHDI/1-1'
Hanging up on 'DAHDI/1-1'
Hungup 'DAHDI/1-1'

Figura 26: Usuario selecciona la opción 1 del IVR

b) Cuando el usuario presiona la opción 2 que es el departamento de soporte con extensión 2002

*CLI>	Starting simple switch on 'DAHDI/1-1'
	Executing [s@from-pstn:1] Answer("DAHDI/1-1", "") in new stack
	<pre>Executing [s@from-pstn:2] Goto("DAHDI/1-1", "menu-dia,s,1") in new stack</pre>
	Goto (menu-dia,s,1)
	Executing [s@menu-dia:1] Answer("DAHDI/1-1", "") in new stack
	Executing [s@menu-dia:2] BackGround("DAHDI/1-1", "menu_1") in new stack
	<dahdi 1-1=""> Playing 'menu 1.gsm' (language 'en')</dahdi>
	Executing [2@menu-dia:1] Goto("DAHDI/1-1", "miprimerejemplo,2002,1") in r
ew sta	ck
	Goto (miprimerejemplo,2002,1)
	Executing [2002@miprimerejemplo:1] Dial("DAHDI/1-1", "SIP/2002,30,Ttm") i
n new	stack
== U	sing SIP RTP CoS mark 5
	Called SIP/2002
	Started music on hold, class 'default', on DAHDI/1-1
	SIP/2002-00000006 is ringing
	SIP/2002-00000006 answered DAHDI/1-1
	Stopped music on hold on DAHDI/1-1
== S	pawn extension (miprimerejemplo, 2002, 1) exited non-zero on 'DAHDI/1-1'
	Hanging up on 'DAHDI/1-1'
	Hungup 'DAHDT/1-1'

Figura 27: Usuario selecciona la opción 2 del IVR

c) Cuando el usuario presiona una opción inválida se vuelve a reproducir el menú para que elija una opción válida.



Figura 28: Usuario selecciona una opción inválida

d) Si el usuario no elige ninguna opción del IVR la llamada es re-direccionada a la operadora.



Figura 29: Llamada redireccionada a la operadora



## 4.2 Evaluación de la calidad de la llamada

#### > Primera prueba: Calidad de las llamada a la PSTN

Primeramente se comprobó que se efectuaran satisfactoriamente las llamadas tanto desde un cliente VoIP hacia un número valido dentro de la red telefónica conmutada y viceversa, como la comunicación entre clientes VoIP. Posteriormente se establecieron 10 llamadas a distintos números de la red telefónica conmutada para evaluar la calidad de las llamadas del servidor Asterisk conectado a la PSTN, tomando en cuenta los siguientes factores: interferencia, eco, volumen y retardo.

En esta prueba les preguntamos a todas las personas con las que nos comunicamos como percibían la voz, si escuchaban alguna interferencia, eco, si el volumen estaba bien y si notaban algún retraso en la conversación, después que nos respondían le notificábamos que la llamada la realizábamos mediante una computadora.

Obteniéndose los siguientes resultados:

	In	terferei	ncia	Eco		Volumen		Retardo			
Número Telefónico	Mucha	Poca	Ninguna	Mucho	Poco	Ninguno	Alto	Bajo	Bien	Si	No
2315-3652											
2311-6273											
2312-4052											
2315-4454											
2312-4145											
2312-4078											
2315-4700											
2311-6956											
2315-1989											
2311-2122											

#### Tabla 5: Resultados de la calidad de las llamadas a la PSTN



Al finalizar la prueba comprobamos que no influye el tiempo que dure la llamada en relación a la estabilidad y fluidez de la comunicación. Las personas que respondieron a nuestras preguntas comentaron que no notaban la diferencia y que la comunicación era buena, que no hubo ningún tipo de interferencia ni retardo, no se escuchaba entrecortada la voz y el volumen era apropiado, solamente el 20% respondió que escucharon un poco de eco.

#### Segunda prueba: ¿Qué sucede cuando la llamada ya está establecida y se levanta el teléfono que está conectado a la interfaz FXS de nuestro servidor?

En la telefonía tradicional es fácil escuchar una conversación telefónica ajena utilizando un teléfono que esté en la misma línea telefónica, mientras que en la telefonía VoIP no. Al realizar la prueba utilizando el servidor Asterisk conectado a la PSTN y conectábamos un el teléfono a la interfaz FXS del servidor y lo levantábamos teniendo una llamada establecida podíamos no solo escuchar toda la conversación sino también unirnos a ella.

#### > Tercera prueba: incrementar el número de llamadas.

Lo que se realizó en esta prueba es verificar que se puede establecer, mantener y terminar una llamada el tiempo que se desee. Como por ejemplo: establecer una llamada desde un cliente VoIP hacia un número de la red telefónica conmutada y realizar una llamada entre clientes VoIP.

Primeramente se estableció una llamada desde uno de los clientes hacia un número valido dentro de la red de telefonía convencional, posteriormente se realizó una llamada entre clientes VoIP. La comunicación de la primera llamada antes y durante de realizar la comunicación entre clientes VoIP fue exitosa.



## 4.3 Realización de pruebas de rendimiento de Asterisk

Para probar el rendimiento del servidor Asterisk se empleó un generador de tráfico SIP que permitiera someter al servidor Asterisk a una alta carga de trabajo. El simulador de tráfico SIP empleado fue el SIPp, las pruebas realizadas se desglosan a continuación:

#### 4.3.1 Máximo número de conexiones concurrente en un escenario ideal

El objetivo de esta prueba de rendimiento fue determinar el número aproximado de llamadas simultáneas que puede llegar a soportar el servidor Asterisk VoIP, utilizando la herramienta SIPp para generar el tráfico en el protocolo SIP. La siguiente imagen nos muestra los mensajes que se envían entre el cliente SIPp, el servidor Asterisk y el servidor SIPp al realizar la prueba.





#### 4.3.1.1 Cliente SIPp

Utilizando la opción pcapplay fue posible hacer la inserción de archivos de audio en formato PCAP en un flujo RTP de la conexión codificando la grabación del audio en formato G.729.

Es importante señalar que antes de realizar las pruebas se debe modificar el límite del tamaño de pila del sistema operativo a 4096 para evitar la aparición de errores durante la ejecución. Para ello ejecutamos el siguiente comando:

ulimit –s 4096		

Una vez eliminada esta limitación software, se procede a crear el escenario en el cliente SIPp que se va a utilizar para la prueba de rendimiento del servidor Asterisk. Creamos el escenario en un archivo con el nombre cliente.xml (Ver Anexo 5: Instalación, Configuración y pruebas realizadas con la herramienta SIPp) y lo guardamos en el directorio donde ha sido instalado SIPp.

En el archivo xml ejecutado por el cliente SIPp, va indicado el audio que el cliente SIPp va a enviar al servidor Asterisk con destino al servidor SIPp. En nuestro caso el archivo de audio se llama prueb y tiene una duración de un minuto.



Figura 31: Cliente SIPp archivo de audio pcap



#### 4.3.1.2 Servidor SIPp

El equipo que empleamos como servidor SIPp, utiliza la opción rtp\_echo para hacer eco con el audio que le llegue desde el servidor Asterisk y poder así establecer una comunicación de sonido bidireccional. Al igual que sucedía en el cliente SIPp, el servidor SIPp utiliza un script en el cual se especifican los mensajes SIP que se envían desde el servidor SIPp al servidor Asterisk para establecer la comunicación de forma correcta, es decir, define como se van a contestar las peticiones que realizan los clientes.

Creamos el escenario en un archivo con el nombre server.xml (Ver Anexo 5: Instalación, Configuración y pruebas realizadas con la herramienta SIPp) y lo guardamos en el directorio donde haya sido instalado SIPp.

#### 4.3.1.3 Configuración de Asterisk para la realización de las pruebas de tráfico.

Fue necesario modificar el límite máximo de número de archivos abiertos en nuestro sistema operativo a 20480 (tamaño de pila). Para ello ejecutaremos la siguiente instrucción en la consola de comandos del equipo:

ulimit -n 20480

Se realizaron dos modificaciones en el código fuente que resultan necesarias para la realización de las pruebas:

- En el archivo chan\_sip.c modificamos el valor MAX\_RETRANS a 1 para generar un intervalo de 2 segundos para los destinos sin conexión (Dispositivo No Disponible, Ruta No Disponible, VoIP no Disponible).
- En el archivo autoservice.c el valor de MAX\_AUTOMONS lo incrementamos a 2048 para prevenir el mensaje de alerta "Exceeded maximum number of automatic monitoring events".

Una vez compilado, debemos configurar Asterisk modificando tres archivos de configuración que se encuentran en el directorio de Asterisk, que son: modules.conf, sip.conf y extensions.conf.



Dado que en la prueba de rendimiento los códec de audio a emplear fueron el códec g711 (a-law) y el códec g729, es necesario habilitar en Asterisk dichos códec: Para ello en el archivo modules.conf, se incluye el siguiente código:

```
isip.conf i extensions.conf i modules.conf i m
```

#### Figura 32: modules.conf codecs agregados

Para configurar el cliente SIPp en el servidor Asterisk, agregamos lo siguiente en el archivo sip.conf:

context = pruebasipp

allowguest = yes; Aceptar las llamadas.

canreinvite = no; Si el trafico RTP pasa siempre por Asterisk.

trustrpid = yes; Confiar en una cabecera Remote-Party-ID.

sendrpid = yes; Enviar la cabecera Remote-Party-ID.

disallow = all; Deshabilitar el uso de todos los codecs.

allow=alaw; Permite el uso del códec alaw.

allow=gsm; Permite el uso del códec gsm.

allow=g729; Permite el uso del códec g729.



A continuación se ilustra las configuraciones realizadas en el archivo sip.conf:

```
🗋 sip.conf 🗱
 "callcounter" setting (for extension states in queue and subscriptions)
; You are encouraged to use the dialplan groupcount functionality
; to enforce call limits instead of using this channel-specific method.
 You can still set limits per device in sip.conf or in a database by using
 "setvar" to set variables that can be used in the dialplan for various
limits.
[general]
context=pruebasipp
allowquest=yes
canreinvite=no
trustrpid=yes
sendrpid=yes
disallow=all
allow=alaw
allow=gsm
allow=g729
```

#### Figura 33: SIPp configuración en el archivo sip.conf

De esta manera el servidor Asterisk aceptará llamadas sin requerir la autenticación de éstas. Por último, configuramos el contexto para el cliente SIPp en el archivo extensions.conf agregando las siguientes líneas en el código del archivo:

exten => Ext\_SIPp,1,Dial(SIP/\${EXTEN}@IP\_ServidorSIPp)

Las configuraciones realizadas en el archivo extensions.conf, se ilustra a continuación:

📄 extensions.conf  🕷	📄 sip.conf  🗶
[pruebasipp] exten => 1000,1,Dia	(SIP/\${EXTEN}@192.168.1.10) ;esta ip es del servidor SIPp

Figura 34: SIPp configuración en extensions.conf



#### 4.3.1.4 Desarrollo de la prueba

En primer lugar, ejecutamos nuestro servidor Asterisk. Una vez ya iniciado procedemos a ejecutar el servidor SIPp a través del siguiente comando en el directorio de instalación de SIPp. Como se observa a continuación:

#### ./ sipp -sf server.xml -nd -i IP\_servidorSIPp -mi IP\_servidorSIPp -rtp\_echo

El parámetro -sf, cargar el escenario definido en el archivo server.xml.

El parámetro –nd modifica el comportamiento de SIPp de la siguiente manera:

- > Ante una retransmisión por timeout, aborta la llamada.
- Ante la recepción de un mensaje BYE o un CANCEL inesperados, se envía el mensaje 200 OK y se aborta la llamada.
- Ante la recepción de cualquier otro mensaje inesperado se envía un BYE o un CANCEL para finalizar o abortar la llamada respectivamente.

El parámetro -i permite recibir tanto el tráfico SIP como el tráfico RTP, utilizando el parámetro -mi. Con el parámetro rtp\_echo, el servidor SIPp realiza un eco de todo el audio que recibe procedente del cliente SIPp, simulándose una llamada real.

Por último, ejecutamos el cliente SIPp a través del siguiente comando y de igual manera nos ubicamos en el directorio de instalación de SIPp. Como se observa a continuación:

./sipp -sf cliente.xml -m num\_llamadas\_enviar -r llamadas\_segundo -l num\_llamadas\_simultaneas -s extension\_cliente\_Asterisk -nd -i IP\_clienteSIPp remote\_host IP\_servidorAsterisk

Los parámetros que se indican en el comando se detallan a continuación:

El parámetro –sf, cargar el escenario definido en el archivo cliente.xml.

El parámetro –m, establece el número total de llamadas que dicho cliente enviará.

El parámetro –r, establecen el número de llamadas por segundo.

El parámetro –I, establece el número máximo de llamadas simultáneas que el cliente puede realizar.

El parámetro -s, indica la extensión destino a llamar.

El parámetro remote\_host, se específica el host remoto al que el cliente SIPp va a enviar las peticiones

SIP, es decir la dirección de red del servidor Asterisk cuya capacidad queremos medir.



Se realizaron muchas pruebas con la herramienta SIPp luego de las cuales se determinó que para que ocurra un desbordamiento de memoria en el servidor Asterisk no influyen las llamadas por segundo, ni el número máximo de llamadas, lo que afecta es la cantidad de llamadas concurrentes el cual en nuestro caso debe ser un valor de 80 con un margen de +/- 3 para que ocurra.

A continuación se muestra una de las pruebas con los valores en donde el servidor Asterisk se desborda por no soportar la cantidad de llamadas simultáneas: Cantidad de Llamadas 1000->Llamadas por segundo 50->Llamadas Simultáneas 85

	Servidor SIPp	Cliente SIPp			
-sf	Server.xml	-sf	Cliente.xml		
-nd –i	192.168.1.10	-m	1000		
-mi	192.168.1.10	-r	50		
		-I	85		
- rtp_echo	Realiza un eco de todo el audio que recibe procedente del cliente SIPp	-s	1000		
		-nd -i	IP del servidor SIPp: 192.168.1.20		
		remote_host	IP del servidor Asterisk: 192.168.1.7		

#### Tabla 6: Prueba SIPp valores donde el servidor Asterisk se desborda



root@carolina-laptop:/home/carol: -m 1000 -r 50 -l 85 -s 1000 -nd In pcap pcap/prueb, npkts 9015 max pkt length 180 base port 7078 Warning: open file limit > FD_SET ZE = 1024 Resolving remote host ': Call-rate(length) Port Tota 50.0(0 ms)/1.000s 5061 90 Call limit reached (-m 1000), ( 2 calls (limit 85) 0 Running, 4 Paused, 0 Woken up 0 dead call msg (discarded)	ina/Descarg -i 192.164 TSIZE; limi 192.168.1.7 cenario Scr al-time To 51.55 s 9.048 s per Per P 43	as/sipp-3 B.1.20 remo ting max. T Done reen tal-calls 1000 riod 1 ms reak was 85 82 out-of-0	.4.0# ./sip ote_host 19 # of open 	op -sf cliente.xml 92.168.1.7 files to FD_SETSI Change Screen ost 1.7:5060(UDP) resolution ter 1 s discarded)
3 open sockets	0	000 loct	oriad PTD	rate (kR/c)
2002972 Totat KIP pekts sent	0.	dod tast j		Tale (KD/S)
medut Unexpected-Msg INVITE>	Messages 1000	Retrans 240	Timeout 48	Unexpected-Msg
100 <	952	Θ	Θ	96
180 <	950	Θ	Θ	22
200 < E-RTD1	950	0	0	0
ACK> [ NOP ]	950	Θ		
Pause [ 1:00]	950			44
BYE>	950	36	4	
200 <	946	Θ	Θ	40

Figura 35: Ejecución del cliente SIPp con los valores que desbordan el servidor Asterisk

Start Time	Statistic	s Screen	[1-9]: Change Screen
	2014-07-03	12:59:04:259	1404413944.259608
Last Reset Time Screen	2014-07-03	13:15:05:788	1404414905.788247
Current Time	2014-07-03	13:15:05:837	1404414905.837730
Counter Name	Periodic val	ue	Cumulative value
Elapsed Time	00:00:00:049	)	00:16:01:578
Call Rate dod)	0.000 cps		1.040 cps
Incoming call created OutGoing call created Total Call created Current Call	0   0   2		0   1000   1000
Successful call	0		946
Failed call	0		52
Response Time 1	00:00:00:00	)	00:00:00:008
Call Length	00:00:00:00		00:00:57:281

Figura 36: Finalización del cliente SIPp con los valores que desbordan el servidor Asterisk

.168.1.10 -m1 192.168.1	Scenar <u>io Sc</u>	creen	[1-9]	: Change Screen
Port Total-time Tot 5060 970.78 s	al-calls Transpor 1090 UDP	rt		
1 new calls during 0.9	32 s period	3 ms sched	uler resolu	ution
140 calls	and a second	Peak was 2	51 calls,	after 606 s
0 Running, 142 Paused,	3 Woken up			
3 open sockets	lueu)			
0 Total echo RTP pckts	1st stream (	0.000 last	period RT	P rate (kB/s)
0 Total echo RTP pckts	2nd stream (	0.000 last	period RT	P rate (kB/s)
	Messages	Retrans	Timeout	Unexpected-Msg
·····> INVITE	950	θ	θ	695
< 180	950	0		
< 200	950	θ	θ	
> ACK	E-RTD1 950	0	0	0
> BYE	950	0	0	0
< 200	950	θ		
[ 4000ms] Pause	950			θ
	Statistics	Screen	[1-9]	: Change Screen
Start Time	2014-07-03 1	3:00:25.91	0531 14044	14025.910531
Last Reset Time	2014-07-03 1	3:16:35.76	9612 14044	14995.769612
Current Time	2014-07-03 1	3:16:36.76	1826 14044	14996.701826
Counter Name	Periodic value		Cumula	ative value
Elapsed Time	1 00:00:00:93200	0	00:16:	10:791000
Call Rate	1.073 cps		1.1	123 cps
Incoming call created	1		1 10	990
OutGoing call created	0		200 A 100	0
Total Call created	1 140		10	990
	1 140			
Successful call	1 0		1 9	50
Failed call	θ		1	0
Response Time 1	1 00:00:00:0000	θ	1 00:00:	00:001000
Call Length	1 00:00:00:00000	0	i 00:01:	03:905000

Figura 37: Servidor SIPp resultados en donde se desborda Asterisk



En el servidor Asterisk cuando este se desborda se observó lo siguiente:



Figura 38: Desborde en el Servidor Asterisk

# 4.3.2 Validación de los resultados en un escenario con carga en la red y en los equipos

En este apartado se realizó una prueba utilizando SIPp para evaluar el rendimiento del servidor Asterisk con uno de los valores del apartado anterior, mientras se realizaban descargas al servidor desde el cliente SIPp, servidor SIPp y un usuario común que utilizó FlashGet y JDownloader como acelerador de descarga. Escenario:



Figura 39: Escenario SIPp con descarga

A continuación se muestran las imágenes de las pruebas realizadas:

> Cliente SIPp: Total Llamadas 1000->Llamadas por segundo 50->Llamadas Simultaneas 80




root@carolina-laptop:/home/carolina/Descargas/sipp-3.4.0# ./sipp -sf cliente.xml -m 1000 -r 50 -l 80 -s 1000 -nd -i 192.168.1.20 remote host 192.168.1.7 In pcap pcap/prueb, npkts 9015 max pkt length 180 base port 7078 Warning: open file limit > FD SETSIZE; limiting max. # of open files to FD SETSI ZE = 1024Resolving remote host '192.168.1.7'... Done. ------ Scenario Screen ------ [1-9]: Change Screen --Call-rate(length) Port Total-time Total-calls Remote-host 50.0(0 ms)/1.000s 5061 781.70 s 1000 192.168.1.7:5060(UDP) Call limit reached (-m 1000), 0.000 s period 0 ms scheduler resolution 0 calls (limit 80) Peak was 80 calls, after 1 s 0 Running, 42 Paused, 0 Woken up 0 dead call msg (discarded) 0 out-of-call msg (discarded) 1 open sockets 
 Messages
 Retrans
 Timeout
 Unexpected-Msg

 1000
 347
 0
 0

 1000
 347
 0
 0

 1000
 0
 0
 0
 INVITE ----> 100 <-----180 <-----Ö 200 <----- E-RTD1 1000 Θ Θ 1000 ACK ----> A [ NOP ] Pause [ 1:00] BYE -----> 1000 1000 1000 Θ Θ Θ Ø 200 <-----0 ----- Test Terminated ----------- Statistics Screen ------ [1-9]: Change Screen --Start Time | 2014-07-03 13:28:52:132 1404415732.132603 2014-07-03 13:41:53:886 1404416513.886908 Last Reset Time 2014-07-03 13:41:53:887 1404416513.887097 Current Time Counter Name | Periodic value | Cumulative value ..... 00:00:00:000 00:13:01:754 Elapsed Time Call Rate 0.000 cps 1.279 cps ...... Incoming call created | 0 0 OutGoing call created | Θ 1000 Total Call created 1000 Current Call Θ Θ Successful call 1000 Failed call 0 0 00:00:00:000 Response Time 1 00:00:00:059 00:00:00:000 Call Length 00:01:00:070 ----- Test Terminated -----

Figura 40: Cliente SIPp prueba con descarga

### Usuario Común:

		Add new d	lownload	b			٤	3	
URL: Referrer:	http://192	168.1.7/html/vis	sual2010.isc	C.					
Category:	Downl	oaded			~				
Save to:	C:\Downloads v			Browser					
Rename:	visual2010.iso				Use Comment				
Username: Password:			Th	Open the F reads From	ile Afb main s	erDown ite 10	oad Ç		
9			7	Downloader					_ 🗆 ×
► 00 U	* *	+ ±	<b>5</b> 9	DO	9			2.00.0000	2.05 MD/1
🔮 Descargas	Capturador	de Enlaces 🎂 Aju	istes						
Nombre [+] = Shakira Ban [+] = Beri 1 [1]	iboo Zamba Capo	Servidor [youtube.com] [mediafire.com]		Estado			3	Progress 29.32 MB/29.32 MB 63.04 MB/63.04 MB	E.
[+] == visual2010.	iso [1]	[http inks]		[1/1] Tier	spo 06m	461 @ 3.0	5 MB/s	1.25 G8/2.46 G8	

Figura 41: Descarga con JDownloader

	Default - FlashGet								
nage	age Tools Help								
De	Image: Second								
	Name	Size	Completed	Percent	Elapsed	Left	Speed	Num	URL
▶,	visual2010.iso	2.46 GB	1.77 GB	72%	00:17:07	00:06:13	2.96M	5/5	http://192.168.1.7/html/visual2010.iso

### Figura 42: Descarga con FlashGet

Mientras se realizaron llamadas y se efectuaron varias descargas al mismo tiempo, no se generaron pérdidas de llamadas, ni el servidor Asterisk colapso, solamente se incrementó las retransmisiones en el lado del cliente SIPp.



Al realizar el siguiente cálculo comprobamos la capacidad de ancho de banda que se necesitó para transmitir las 80 llamadas simultáneas:

80.8 kbps (Ancho de banda requerido por el códec G729)

x 80 (Cantidad de llamadas simultáneas)

### 6464.00 kbps

Al pasarlos a megabits resultó la siguiente cantidad: 6.464 Mbps, esa es la capacidad que se necesitó para establecer las 80 llamadas por lo que confirmamos que las limitaciones que afectan el rendimiento del servidor no se deben al sistema operativo CentOS de 64 bits que tiene instalado, ni se debe a la capacidad del router Eltel, ni a los cables UTP Cat. 5, ni al códec utilizado G729 sino que está directamente relacionado con las características de la máquina.



## CAPÍTULO 5 CONCLUSIONES



### 5.1 Conclusiones

Al culminar nuestro trabajo de tesis concluimos que con la integración de la tarjeta Motorola Wildcard PCI X100P al servidor Asterisk se logró encaminar las llamadas VoIP a la red telefónica convencional y viceversa, teniendo una facilidad de uso debido a la gran cantidad de funciones que tenemos a disposición tales como transferencia de llamadas, desvió de llamadas, buzón de voz, IVR y un sin número más de servicios.

Con el servidor conectado a la PSTN se realizaron distintas pruebas para evaluar la calidad en las llamadas VoIP hacia la línea convencional en donde se apreció que la comunicación fue efectiva, clara y sin ningún tipo de interferencia.

Para toda empresa que apenas se está constituyendo, implementar una PBX representa una gran inversión en equipos, instalación y posterior administración; de igual modo, una empresa que ya tenga un sistema PBX instalado se enfrenta a unos altos costos de mantenimiento y problema al momento de necesitar más extensiones, pero con la aparición de VoIP y el desarrollo de una PBX como Asterisk, las empresas tienen hoy en día una alternativa viable en cuanto a las PBX privadas lo que representa una solución de voz económica, escalable, y práctica para disminuir costos.

### 5.2 **Recomendaciones**

Realizar pruebas de rendimiento del servidor Asterisk en un entorno real como por ejemplo en un laboratorio de computación donde cada estudiante deberá tener en su PC un softphone para poder realizar llamadas VoIP a otros estudiantes que estén dentro del mismo laboratorio. Ademas incorporar otras funcionalidades que ofrece asterisk como las Video llamadas y comparar si sucede el mismo resultado.



## ANEXOS



## 5.3 Anexo 1: Instalación y Configuración del servidor Asterisk

Para poder compilar Asterisk desde el código fuente es necesario tener un entorno de compilación en tu sistema. Asegúrate de tener los siguientes paquetes instalados:

- bison (un generador de analizadores sintácticos)
- > zlib y zlibdevel (bibliotecas de compresión desarrollo)
- > ncurses y ncursesdev (bibliotecas de utilidades de consola desarrollo)
- openssl y openssldev (libssldev)
- (SSL bibliotecas de desarrollo)
- libc6dev (cabeceras y bibliotecas de desarrollo GNU C)
- gcc y make (el compilador C de GNU y la utilidad make)
- > mpg123 (permite a la centralita reproducir archivos mp3 para ofrecer servicio de música)

# yum install bison zlib zlibdevel ncurses ncursesdev openssl openssldev ssl libc6dev gcc make mpg123

El siguiente paso es descargar Asterisk de la página oficial (http://www.asterisk.org ), una vez descargado lo descomprimimos siguiendo los siguientes pasos:

Paso 1. Ejecutar:

# tar xvzf asterisk-11.7.0.tar.gz //Descomprime el archivo

# cd astersk-11.7.0 //nos ubicamos dentro del directorio de Asterisk

Paso 2. Para compilar

# make

Paso 3. Para instalar

# make install

Paso 4. Para instalar los "scripts" de arranque:

# make config

Paso 5. Para instalar los ficheros de configuración de ejemplo:

# make samples

Paso 6. Para instalar la documentación de desarrollo:

# make progdocs

### 5.3.1 Instalar sonidos en español

Es importante recalcar que los archivos de audios de Asterisk están en inglés, para añadir los archivos de audios traducidos al español basta con descargar el archivo asterisk-sonidos-es-0.4.tar.gz y luego lo descomprimimos en el directorio de sonidos de Asterisk (var/lib/asterisk/sounds) se nos crea un carpeta llamada "es "la cual contiene los archivos en español.

#cd /var/lib/asterisk/sounds #tar zxvf /var/lib/asterisk/sounds/asterisk-sonidos-es-0.4.tar.gz

Para que asterisk utilice estos sonidos solo es necesario añadir en sus archivos de configuración la orden:

Language=es



п

### 5.3.2 Directorios de Asterisk.

### Tabla 7: Directorios de Asterisk

Directorios	Descripción
	En este directorio se encuentran los archivos de configuración de Asterisk como
/etc/Asterisk/	los archivos de configuración del dialplan y sip entre otros.
	En este directorio se encuentran todos los módulos cargables de Asterisk, por
	defecto Asterisk carga todos los módulos de este directorio, pero estos pueden
	ser deshabilitados editando el archivo modules.conf, para que no sean cargados
/usr/lib/Asterisk/modules	durante la partida de Asterisk.
	En este directorio encontramos el archivo astdb y otros subdirectorios, el archivo
/var/lib/Asterisk/	astdb es la base de datos del sistema Asterisk.
	Contiene subdirectorios como: outgoing/, qcall/, tmp/, and voicemail/, Asterisk
	supervisa los directorios outgoing/, qcall/ con los archivos de textos que contienen
	las llamadas, con el fin de generar una llamada tan solo moviendo o copiando el
/var/spool/Asterisk/	chivo estructurado a outgoing/. En el directorio tmp/ se guarda información
	temporal, para que dos llamadas no puedan escribir y leer al mismo tiempo. En el
	directorio voicemail/ se guardan todos los voicemail y los saludos de los usuarios.
/var/run/	En este directorio se guardan toda la información de los procesos (pid) del
	sistema, incluidos los de Asterisk.
	Contiene los logs del sistema estos pueden ser configurados según las
/var/log/Asterisk/	necesidades en el archivo logger.conf.
	Como la mayoría de las aplicaciones de GNU/Linux Asterisk, consta con una línea
Asterisk Command Line	de comandos, la cual permite cargar Asterisk de distintas maneras. Para esto
(CLI)	cargamos opciones al iniciar la aplicación.



• Configuración de sip.conf

### [2000]

username=2000 //Nombre de la extensión del primer usuario. type=friend ; //Con este tipo se podrá hacer y recibir llamadas. secret=familia1 //Contraseña de acceso. language=es //Específica el lenguaje a español. callerid="Usuario1"<2000> //Nombre de usuario y extensión. qualify=yes //Tiempo de latencia no mayor a 2000ms. nat=no //No hay nat en la conexión de la extensión con Asterisk. host=192.168.1.11 // IP del cliente que tiene el softphone. context=miprimerejemplo //El contexto que usará la extensión. mailbox=2000@buzon //configuración de la bandeja de mail.

Ahora configuraremos la extensión para el segundo usuario.

[2001]	
--------	--

username=2001 //Nombre de la extensión del primer usuario. type=friend //Con este tipo se podrá hacer y recibir llamadas. secret=familia1 //Contraseña de acceso. callerid="Usuario2"<2001> //Nombre de usuario y extensión. language=es //Específica el lenguaje a español. qualify=yes //Tiempo de latencia no mayor a 2000ms. nat=no //No hay nat en la conexión de la extensión con Asterisk. host=192.168.1.13 // IP del cliente que tiene el softphone. context=miprimerejemplo //El contexto que usará la extensión. mailbox=2001@buzon //configuración de la bandeja de mail.



• Configuración de extensions.conf

Se configuró el plan de marcación, en el que se maneje las aplicaciones básicas de una llamada.

//Archivo extension.conf [miprimerejemplo] exten => 2000,1,Dial(SIP/2000,30,Ttm) exten => 2000,2,Voicemail(2000@buzon,u) exten => 2000,3,Hangup() exten => 2001,1,Dial(SIP/2001,30,Ttm) exten => 2001,2,Voicemail(2001@buzon,u) exten => 2001,3,Hangup() //Esta extensión es para llamar a nuestro buzón de voz exten => \*98,1,Answer exten => \*98,2,Wait(1) exten => \*98,3,VoiceMailMain(\${CALLERID(num)}@buzon) exten => \*98,4,Hangup() //exten para comunicación exten => 3000,1,VoiceMailMain(2000@buzon) exten => 3001,2,VoiceMailMain((2001@buzon)

• Configuración de voicemail.conf

Se crearon los buzones de voz de ambos usuarios y también se le asignó una contraseña. Al buzón 2000 se le va dar la contraseña 5678 y el buzón 2001 la contraseña 8765

[buzon]

2000 => 5678,Usuario1,usuario1@seminario.com

2001 => 8765,Usuario2,usuario2@seminario.com



## 5.5 Anexo 3: Instalación del Softphone SJphone (en el lado del cliente)

Primero descargamos el programa SJphone, el cual encontramos en Internet y que además es gratuito.

Debemos seguir los siguientes pasos:

- Ingresar al siguiente sitio web: http://www.counterpath.com/sjphone-download.html
- Seleccionar la plataforma que utilice (Windows, Linux o Mac).

Una vez descargados e instalados en las tres máquinas que actuaran de clientes procedemos a configurarlo .Para ello abrimos el softphone.

- > Pasos para la configuración del SJphone:
  - 1. Hacemos clic en menú como lo indicamos en la figura de abajo, luego nos mostrará varias opciones y seleccionamos Profiles.

SJphone	Options	<b>e</b>
PC to PC (SIP) Ready to call	Skins   Interface   Neighborhood   ILS Direct User Information   Call Options Profiles Audi Name:	ory   Support o   Hot Keys
Click 'Dial' button to choose respondent	E-mail:	
	Comments:	* *
1         2 abc         3 def           4 ghi         5 jki         6 mno           7 pars         8 tuv         9 wxyz           *         0 oper         #	Image Use image: Image should be a 32x32 bmp, png or jpeg file less than 10 Kb.	rowse

Figura 43: Configuración del SJphone paso 1



2. Luego damos click en New para crear un usuario, rellenamos los campos y presionamos OK.

Profile		Status	create new ric		
PC to PC (H.323)		initialized	Profile name:	Usuario 1	OK
PC to PC (SIP)		initialized	File name:	Usuario 1.ini	Cancel
			Profile type:	Calls through SIP Proxy	Help
			Important no	te	
			Calls through SIP proxy info Caller informa	SIP Proxy: Profile for a call through a SIP proxy. ormation is permanently stored in the profile. stion can be easily changed by re-initializing.	
New	Edk	Delete			
0	Indiation	Barrow I			

Figura 44: Configuración del SJphone paso 2

 Luego seleccionamos la opción SIP Proxy introducimos la dirección IP del servidor, el puerto 5060 definido por las especificaciones del protocolo SIP, por último el dominio y pulsamos OK.

😻 Profile Options 🛛 🖾	Ref Profile Options
General   Initialization SIP Proxy Advanced   DTMF   STUN   Name: Usuario 1	General   Initialization SIP Proxy   Advanced   DTMF   STUN   Proxy domain: 192.168.1.7 : 5060
Type: SIPProxy File: Usuario1.ini Version: 1	User domain: servidor.com Register with proxy reference of the service of the ser
OK Cancel	Image: Window Cancel

### Figura 45: Configuración del SJphone paso 3



4. Especificamos el nombre del usuario y la clave como aparece en el fichero sip.conf para poder registrarse en ella y presionamos OK.

Service: Usuario1	X
Please enter this information to initialize the service profile	ок
Account: 2001	
Password:	Cancel
	Help
Save service information permanently	

Figura 46: Información del Usuario1

5. En la figura de abajo podemos ver que nuestra cuenta ha sido creada satisfactoriamente y que nuestro usuario actual está en uso para poder comunicarse con otros clientes.

Options		X		
Skins   Interface   User Information   C	Neighborhood ILS all Options Profiles	Directory Support Audio Hot Keys		
Profile		Status		
PC to PC (H.323) initialized PC to PC (SIP) initialized				
Usuario1	Usuario 1			
New	Edit	Delete		
Use	Initialize	Rename		
	01	Cancel		

Figura 47: Usuario1 creado satisfactoriamente

## 5.6 Anexo 4: Instalación y Configuración de la tarjeta Authentic X100P SE FXO PCI

Para la instalación de la tarjeta en un servidor Linux, distribución CentOS estos son los pasos a seguir:

- > Apagar el computador.
- > Desconectar el cable de alimentación.
- > Abrir el chasis del Computador.
- > Insertar la tarjeta en una ranura PCI disponible.
- > Volver a poner el chasis del computador.
- > Conectar el cable de alimentación al computador.
- > Conectar la línea telefónica al puerto FXO de la tarjeta a través de un cable telefónico.
- Encender el computador.

Normalmente CentOS reconoce automáticamente la nueva tarjeta instalada y le asigna un IRQ para que pueda comunicarse con el procesador del computador.

Considerando que asterisk trabaja sobre un sistema Linux, es necesario establecer una capa de abstracción entre el hardware provisto (las tarjetas Digium y compatibles), y nuestro sistema operativo. Para ello disponemos del paquete DAHDI, el cual provee principalmente de todo el sistema de interfaz entre estos dos elementos, y los Drivers específicos de las tarjetas para poder ser reconocidas por nuestro sistema en el arranque.

Para esta parte debemos descargar de la página oficial de Asterisk los paquetes:

- > DAHDI
- > LIBPRI



A continuación seguiremos los siguientes pasos:

Paso 1. Ejecutar:

#tar xvzf dahdi-linux-complete-2.8.0.1.tar.gz //Descomprime el archivo
#cd dahdi-linux-complete-2.8.0.1 //accedemos al directorio descomprimido de dahdi

Paso 2. Para compilar

# make

Paso 3. Para instalar

# make install

Paso 4. Para instalar los "scripts" de arranque:

# make config

Accedemos donde tenemos descargados los paquetes:

a. Ejecutar:

#tar xvzf libpri.tar.gz //Descomprime el archivo
#cd libpri-2.8.0 //Accedemos al directorio descomprimido de libpri

b. Compilamos

# make

c. Instalamos

# make install

### 5.6.1 Directorios de DAHDI

Dentro de este directorio /etc/dahdi existen cuatro ficheros principales:

### Tabla 8: Directorios de Dahdi

Directorios	Descripción
	Encargado de gestionar los módulos en el Linux Kernel, para que se inicien automáticamente en
Modules	el arranque.
	Provee la información exacta sobre cómo se conectarán las interfaces con nuestras tarjetas y
	otros parámetros específicos de las mismas, como lo relacionado a los canceladores de echo,
system.conf	comportamientos por zonas mundiales, etc
	Se pueden especificar la inicialización o descarga de ciertos módulos adicionales, y scripts Shell
init.conf	que queramos que sean ejecutados en la carga de la aplicación dahdi en el inicio.
	Existe una aplicación capaz de configurar un system.conf genérico, según las tarjetas que se
genconf_parameters	hayan reconocido a través del modules, de forma totalmente automática.

### 5.6.2 Configuración de DAHDI

> Para ver qué tipo de tarjeta tenemos en nuestro sistema se ejecutó el siguiente comando:

#	dmesg	
---	-------	--



ara



visualizar el dispositivo tecleamos

# dahdi\_hardware

[root@localhost ~]#	dahdi_hardwar	0		
pci:0000:01:01.0	_wcfxo+	1057:5608	Wildcard	X100P

Figura 49: Ejecución del comando dahdi\_hardware

A continuación se ejecutó el comando dahdi\_tool esta herramienta nos permite diagnosticar el estado de la tarjeta instalada en el sistema.

Asterisk Console on 'localhost.localdomain' 💥 root@localhost:~	×
DAHDI Tool (C)2002-2008 Digium, Inc.	<u>~</u>
DAHDI Telephony Interfaces	
Alarms Span	
OK Wildcard X100P Board 1	Ť
	1 I
Select Quit	
Span 1: 1 total channels, 1 configured F1=Detai	ls F10=Quit 🗸

Figura 50: Ejecución del comando dahdi\_tool

> Verificamos el funcionamiento de la tarjeta

# dahdi\_cfg -vv

[root@localhost asterisk]# dahdi_cfg -vv DAHDI Tools Version - 2.8.0	
DAHDI Version: 2.8.0.1 Echo Canceller(s): Configuration ====================================	
Channel map:	
Channel 01: FXS Kewlstart (Default) (Echo Canceler: none) (Slaves: 01)	
1 channels to configure.	

Figura 51: Ejecución del comando dahdi\_cfg -vv

Para ver el estado de nuestra tarjeta

# /etc/init.d/dahdi status

[roo	ot@loc	alho	ost ~]# /	/etc/init.o	d/dahd:	i sta	atus		
###	Span	1:	WCFX0/0	"Wildcard	X100P	Boar	d 1"	(MASTER)	RED
1	FX0		FXSKS	(In	use)	(EC:	MG2 -	INACTIVE	) RED

Figura 52: Estado de la tarjeta sin conexión a la PSTN

Como podemos notar nos sale que el estado de nuestra tarjeta está inactivo RED pero cuando tenemos conectado el servidor con la PSTN el estado de la tarjeta dice activo battery.

[roo	t@loc	alhost ~l# /	etc/init d/dahdi status	
[100	-	atilost ~]# /		
###	Span	1: WCFX0/0	"Wildcard X100P Board 1" (MASTER)	
1	FX0	FXSKS	(In use) (EC: MG2 - ACTIVE) (battery)	

Figura 53: Estado de la tarjeta con conexión a la PSTN

Iniciamos manualmente dahdi

# /etc/init.d/dahdi start



Figura 54: Iniciar dahdi

> A continuación generamos la configuración para poder trabajar con la tarjeta análoga o con la tarjeta digital. Tecleamos el comando dahdi\_genconf –v el cual genera una plantilla de configuración.

# dahdi\_genconf -v

[root@localh	ost ~]# dahdi_genconf -v
Default para	meters from /etc/dahdi/genconf_parameters
Generating /	etc/dahdi/span <sub>k</sub> types.conf
Generating /	etc/dahdi/assigned-spans.conf
Generating /	etc/dahdi/system.conf
Generating /	etc/asterisk/dahdi-channels.conf

Figura 55: Ejecución del comando dahdi\_genconf -v

> Ahora pasamos a modificar los archivos de configuración de dahdi. Visualizamos el archivo system.conf

# cat /etc/dahdi/system.conf

[root@localhost ~]# cat /etc/dahdi/system.conf
# Autogenerated by /usr/sbin/dahdi_genconf on Fri Mar   7 11:30:57 2014
# If you edit this file and execute /usr/sbin/dahdi genconf again,
# your manual changes will be LOST.
# Dahdi Configuration File
#
# This file is parsed by the Dahdi Configurator, dahdi cfg
#
# Span 1: WCFXO/0 "Wildcard X100P Board 1" (MASTER)
fxsks=1
echocanceller=mg2,1
# Global data
loadzone = us
defaultzone = us





Cómo podemos notar en la parte inferior del fichero dice que fue autogenerado por /usr/sbin/dahdi\_genconf seguido de la fecha en que fue generado lo único que vamos a modificar en este archivo luego que dahdi\_genconf lo generó es la zona donde estamos trabajando y de ahí se deja todo por defecto.

Contenido del fichero system.conf

- fxsks=1: Es el tipo de señalización utilizada por el canal. En este caso protocolo FXS Koolstart.
- echocanceller=mg2,1 : Especifica el tipo de cancelación del echo software (mg2) y el canal (en este caso existe solamente 1)
- loadzone =us: Determina la zona de comunicación donde estamos trabajando.
- defaultzone=us: Zona predefinida para la señalización (tono de llamada, tono de ocupado, etc.)

> Nos ubicamos en el archivo /etc/asterisk/dahdi-channels.conf copiamos el span 1 donde aparece el FXS y lo pegamos en la parte final del archivo /etc/asterisk/chan\_dahdi.conf porque este es el archivo que lee dahdi sino no va a funcionar la configuración, como notamos la FXO está en el canal 1 el contexto from\_pstn lo debemos tener definido en el archivo extensions.conf

[root@localhost ~]# cat /etc/asterisk/dahdi-channels.conf ; Autogenerated by /usr/sbin/dahdi\_genconf on Fri Mar 7 11:30:57 2014 ; If you edit this file and execute /usr/sbin/dahdi\_genconf again, ; your manual changes will be LOST. ; Dahdi Channels Configurations (chan\_dahdi.conf) ; ; This is not intended to be a complete chan\_dahdi.conf. Rather, it is intended ; to be #include-d by /etc/chan\_dahdi.conf that will include the global settings ; ; Span 1: WCFX0/0 "Wildcard X100P Board 1" (MASTER) ;;; line="1 WCFX0/0/0 FXSKS (EC: MG2 - INACTIVE)" signalling=fxs\_ks callerid=asreceived group=0 context=from-pstn channel => 1

Figura 57: Visualización del archivo dahdi-channels.conf de dahdi



En esta imagen visualizamos lo que contiene el fichero /etc/asterisk/chan\_dahdi.conf

[root@localhost ~]# cat /etc/asterisk/chan dahdi.conf DAHDI Telephony Configuration file [trunkgroups] [channels] usecallerid=yes callwaiting=yes usecallingpres=yes callwaitingcallerid=yes threewaycalling=yes transfer=yes canpark=yes cancallforward=yes callreturn=yes echocancel=yes echocancelwhenbridged=yes group=1 callgroup=1 pickupgroup=1 ;ESTO ES LO QUE COPIAMOS DEL ARCHIVO /etc/asterisk/dahdi-channels.conf signalling=fxs ks callerid=asreceived group=0 context=from-pstn channel => 1

Figura 58: Visualización del archivo chan\_dahdi.conf



# 5.7 Anexo 5: Instalación, Configuración y pruebas realizadas con la herramienta SIPp

La instalación de SIPp se realiza, tanto para el equipo que actuará como cliente SIPp como para el equipo que actuará como servidor SIPp. En primer lugar, hay que descargar e instalar las dependencias necesarias para poder compilar SIPp.

Desde un equipo en Linux, ejecutamos el siguiente comando:

apt-get install c++ libncurses5-dev libpcap0.8-dev libnet1-dev

Ahora descargamos el paquete SIPp y lo descomprimimos ejecutando los siguientes comandos:

Wget http://surfnet.dl.sourceforge.net/sourceforge/sipp/sipp-3.4.0.tar.gz
tar zxvf sipp-3.4.0.tar.gz

Para finalizar, compilamos SIPp en la opción pcapplay para permitir la reproducción de dicho tipo de archivos. Ejecutando las siguientes líneas en el terminal de Linux:

cd sipp.svn	
make pcapplay	

### 5.7.1 Archivos XML del Cliente SIPp y Servidor SIPp

Por el lado del Cliente SIPp el archivo XML se llama cliente.xml y por el lado del Servidor SIPp el archivo XML se llama server.xml

#### ➢ Cliente SIPp:

```
GNU nano 2.2.2
                                        Archivo: cliente.xml
!DOCTYPE scenario SYSTEM "sipp.dtd">
:!-- This program is free software; you can redistribute it and/or
:!-- modify it under the terms of the GNU General Public License as
:!-- published by the Free Software Foundation; either version 2 of the
:!-- License, or (at your option) any later version.
!-- This program is distributed in the hope that it will be useful,
!-- but WITHOUT ANY WARRANTY; without even the implied warranty of
!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
!-- GNU General Public License for more details.
 !-- You should have received a copy of the GNU General Public License
 !-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
                             Sipp 'uac' scenario with pcap (rtp) play
 <!-- In client mode (sipp placing calls), the Call-ID MUST be <!-- generated by sipp. To do so, use [call_id] keyword.
       Contact: sip:sipp@[local_ip]:[local_port]
       Max-Forwards: 70
       Subject: Performance Test
       Content-Type: application/sdp
       Content-Length: [len]
       v=0
       o=user1 53655765 2353687637 IN IP[local ip type] [local ip]
       S=-
       c=IN IP[local ip type] [local ip]
       t=0 0
       m=audio [auto media port] RTP/AVP 8 101
       a=rtpmap:8 PCMA/8000
       a=rtpmap:101 telephone-event/8000
       a=fmtp:101 0-11,16
    11>
 <!-- By adding rrs="true" (Record Route Sets), the route sets
<!-- are saved and used for following messages sent. Useful to test
<!-- against stateful SIP proxies/B2BUAs.
<recv response="200" rtd="true" crlf="true">
 <!-- Packet lost can be simulated in any send/recv message by
 <!-- by adding the 'lost = "10"'. Value can be [1-100] percent.
```

Figura 59: Archivo XML del cliente SIPp parte1



Figura 60: Archivo XML del cliente SIPp parte 2



### Servidor SIPp

```
GNU nano 2.2.2
                                                                                                    Fichero: server.xml
?xml version="1.8" encoding="ISO-8859-1" ?>
!DOCTYPE scenario SYSTEM "sipp.dtd">
<!-- This program is free software; you can redistribute it and/or
<!-- modify it under the terms of the GNU General Public License as
<!-- published by the Free Software Foundation; either version 2 of the
<!-- License, or (at your option) any later version.
       This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
<!-- Free Software Foundation, Inc.,
<!-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 <!-- The '[last_*]' keyword is replaced automatically by the <!-- specified header if it was present in the last message received
  <!-- message, all occurences are concatenated (CRLF seperated) <!-- to be used in place of the '[last_*]' keyword.
          Content-Length: 0
      11>
       Contact: <sip:[local_ip]:[local_port];transport=[transport]>
Content-Type: application/sdp
Content-Length: [len]
       o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
       c=IN IP[media_ip_type] [media_ip]
        t=0 0
       m=audio [media_port] RTP/AVP 18
a=rtpmap:18 G729/8000
```

Figura 61: Archivo XML del servidor SIPp parte 1



Figura 62: Archivo XML del servidor SIPp parte 2



### 5.7.2 Pruebas realizadas con la herramienta SIPp sin descargas

> Total Llamadas 1000-> Llamadas por segundo 20-> Llamadas Simultaneas 80

Cliente SIPp:

<pre>root@carolina-laptop:/home -m 1000 -r 20 -l 80 -s 10 In pcap pcap/prueb, npkts max pkt length 180 bccc sect 7070</pre>	e/carolina/Des 900 -nd -i 192 9015	scargas/sipp-3 2.168.1.20 rem	.4.0# ./s ote_host	ipp -sf cliente.xm 192.168.1.7	nl
Warning: open file limit : ZE = 1024	<pre>&gt; FD_SETSIZE;</pre>	limiting max.	# of open	n files to FD_SETS	51
Resolving remote	host '192.168	3.1.7' Done			
Call-rate(length) Por 20.0(0 ms)/1.000s 506	t Total-time 1 782.15	o Screen e Total-calls 5 1000	Remote-1 192.168	: Change Screen host .1.7:5060(UDP)	
Call limit reached (-m : 0 calls (limit 80)	1000), 0.000 s	s period 0 ms Peak was 80	schedule calls, a	r resolution fter 4 s	
0 dead call msg (discard 1 open sockets	woken up ded)	0 out-of-ca	ll msg (d	iscarded)	
	Messad	es Retrans	Timeout	Unexpected-Msg	
INVITE>	1000	θ	θ		
100 <	1000	Θ	Θ	0	
180 <	1000	0	Θ	Θ	
200 <	E-RTD1 1000	θ	Θ	θ	
ACK> [ NOP ]	1000	θ			
Pause [ 1:00]	1000			0	
BYE>	1000	θ	0		
200 <	1000	θ	0	Θ	
	Statistic	s Screen	[1-9]	: Change Screen	
Start Time	2014-07-03	11:13:31:983	140440	07611.983055	
Last Reset Time	2014-07-03	11:26:34:168	14044	08394.168299	
Current Time	2014-07-03	11:26:34:168	140440	08394.168403	
Counter Name	Periodic val	ue	Cumula	tive value	
Elapsed Time Call Rate	00:00:00:00 0.000 cps		00:13:0	02:185 78 cps	j);
Incoming call created	0		1	0	1
OutGoing call created Total Call created Current Call	0			90 90	
Successful coll			1 20		
Failed call	0		100	0	
Response Time 1 Call Length	00:00:00:00	)	00:00:0	00:008 00:010	

Figura 63: Cliente SIPp primera prueba sin descarga con SIPp

### Servidor SIPp:

root@ubuntu:/home/curacao/Escritorio/sipp-3.4.0# ./sipp -sf server.xml -nd -i 19 2.168.1.10 -mi 192.168.1.10 -rtp\_echo ------ Scenario Screen ------ [1-9]: Change Screen ---Total-time Total-calls Transport 850.98 s 1125 UDP Port 850.98 s 5060 0 new calls during 0.608 s period 125 calls 4 ms scheduler resolution Peak was 255 calls, after 666 s 0 Running, 127 Paused, 2 Woken up 0 dead call msg (discarded) 3 open sockets 0 Total echo RTP pckts 1st stream 0 Total echo RTP pckts 2nd stream 0.000 last period RTP rate (kB/s) 0.000 last period RTP rate (kB/s) Messages Retrans Timeout Unexpected-Msg ·····> INVITE 1000 649 Θ 0 <----- 180 <----- 200 -----> ACK 1000 Ø Θ 1000 Θ E-RTD1 1000 Θ Θ Θ ----> BYE 1000 Θ Θ Θ <----- 200 [ 4000ms] Pause 1000 Θ 1000 0 Start Time Last Reset Time 2014-07-03 11:28:50.193866 1404408530.193866 Current Time 2014-07-03 11:28:50.805606 1404408530.805606 Counter Name | Periodic value | Cumulative value Elapsed Time | 00:00:00:611000 | 00:14:10:999000 Call Rate 0.000 cps 1.322 Cps Incoming call created | Θ 1125 OutGoing call created Total Call created Θ 0 1125 Current Call 125 Successful call Θ 1000 Failed call Θ . . . . . . . . . . . . 90:00:00:00:0000 Response Time 1 | 00:00:00:001000 Call Length 90:00:00:00:00000 00:01:04:152000 ----- Test Terminated -----

Figura 64: Servidor SIPp primera prueba sin descarga con SIPp



> Total Llamadas 1000->Llamadas por segundo 50->Llamadas Simultaneas 80

Cliente SIPp:

root@carolina-laptop:/home/carolina/Descargas/sipp-3.4.0# ./sipp -sf cliente.xml -m 1000 -r 50 -l 80 -s 1000 -nd -i 192.168.1.20 remote host 192.168.1.7 In pcap pcap/prueb, npkts 9015 max pkt length 180 base port 7078 Warning: open file limit > FD SETSIZE; limiting max. # of op<u>en files to FD SETSI</u> ZE = 1024Resolving remote host '192.168.1.7'... Done. ------ Scenario Screen ------- [1-9]: Change Screen ---Call-rate(length) Port Total-time Total-calls Remote-host 50.0(0 ms)/1.000s 5061 780.95 s 1000 192.168.1.7:5060(UDP) Call limit reached (-m 1000), 0.000 s period 0 ms scheduler resolution Peak was 80 calls, after 1 s 0 calls (limit 80) 0 Running, 41 Paused, 0 Woken up 0 dead call msg (discarded) 0 out-of-call msg (discarded) 1 open sockets Messages Retrans Timeout Unexpected-Msg 1000 1 0 1000 1 0 0 INVITE ----> 100 <-----0 0 180 <---- 1000 0 200 <---- E-RTD1 1000 0 Θ 0 ACK -----> 1000 0 [ NOP ] 1000 1000 0 0 1000 0 0 Pause [ 1:00] 0 BYE -----> 200 <-----> 0 
 Start Time
 2014-07-03
 11:29:42:816
 1404408582.816264
 Last Reset Time | 2014-07-03 11:42:43:798 1464409363.798957 2014-07-03 11:42:43:799 1404409363.799152 Current Time Counter Name | Periodic value | Cumulative value 
 Elapsed Time
 | 00:00:00:000
 | 00:13:00:982

 Call Rate
 | 0.000 cps
 | 1.280 cps
 Incoming call created | 0 OutGoing call created | 0 Θ 1000 Total Call created | Current Call | 0 1000 Successful call | 0 Failed call | 0 | 1000 | 0 Response Time 1 | 00:00:00:000 Call Length | 00:00:00:000 | 00:60:00:009 | 00:61:00:011

Figura 65: Cliente SIPp segunda prueba sin descarga con SIPp

Servidor SIPp:

oot@ubuntu:/home/curacao/Escritorio/sipp-3.4.0# ./sipp -sf server.xml -nd -i 19 .168.1.10 -mi 192.168.1.10 -rtp echo ------ Scenario Screen ------ [1-9]: Change Screen ------Port Total-time Total-calls Transport 5060 885.79 s 1126 UDP 0 new calls during 0.288 s period 4 ms scheduler resolution 126 calls Peak was 256 calls, after 666 s 0 Running, 128 Paused, 0 Woken up 0 dead call msg (discarded) 3 open sockets 0 Total echo RTP pckts 1st stream 0 Total echo RTP pckts 2nd stream 0.000 last period RTP rate (kB/s) 0.000 last period RTP rate (kB/s) Messages Retrans Timeout Unexpected-Msg ----> INVITE 1000 θ Θ 630 1000 0 1000 0 <----- 180 1000 1000 E-RTD1 1000 <----- 200 0 0 1000 0 0 1000 0 1000 θ ----> ACK Θ 0 ----> BYE 0 <----- 200 [ 4000ms] Pause 0 ------ Statistics Screen ------ [1-9]: Change Screen ---2014-07-03 11:30:53.037851 1404408653.037851 Start Time Last Reset Time 2014-07-03 11:45:38.549860 1404409538.549860 2014-07-03 11:45:38.841802 1404409538.841802 Current Time -----Counter Name | Periodic value | Cumulative value . . . . . . . . . . . . . . ...... Elapsed Time | 00:00:00:291000 00:14:45:803000 Call Rate 0.000 cps 1.271 cps Incoming call created Θ 1126 OutGoing call created | Θ Θ Total Call created 1126 Current Call 126 Successful call 1000 Θ Failed call θ Response Time 1 00:00:00:000000 00:00:00:001000 Call Length 00:00:00:00:000000 00:01:04:152000 ----- Test Terminated -----

Figura 66: Servidor SIPp segunda prueba sin descarga con SIPp



> Total Llamadas 1000->Llamadas por segundo 80->Llamadas Simultaneas 80

Cliente SIPp:

root@carolina-laptop:/h -m 1000 -r 80 -l 80 -s	ome/carolina/Descar 1000 -nd -i 192.10	rgas/sipp-3 58.1.20 rem	0.4.0# ./s	ipp -sf cliente.xml 192.168.1.7
In pcap pcap/prueb, npk	ts 9015			
max pkt length 180				
base port 7078				
ZE = 1024	t > FD_SETSIZE; (1	niting max.	# от оре	n files to FD_SETSI
Resolving remo	te host '192.168.1.	7' Done		
Call_cate(length) D	Total time	reen	Remote	: Change Screen
80.0(0 ms)/1.000s 5	961 780.70 s	1000	192.168	.1.7:5060(UDP)
Call limit reached (-	m 1000), 0.000 s pe	eriod 0 ms	schedule	r resolution
0 Running, 41 Paused.	0 Woken up	Cak was or	, caces, a	iter 1 3
0 dead call msg (disc	arded) (	out-of-ca	all msg (d:	iscarded)
1 open sockets				
(1253) (1253)	Messages	Retrans	Timeout	Unexpected-Msg
INVITE	> 1000	1	0	
100 <	- 1000	0	0	8
180 <	- E-PTD1 1000	0	0	e e e e e e e e e e e e e e e e e e e
200	- E-KIDI 1000		•	•
ACK	> 1000	θ		
Pause [ 1:00	1 1000			1
BYE	- 1000	0	0	
200 <	- 1000	Θ	0	θ
	Statistics 9	creen	[1.9]	· Change Screen
Start Time	1 2014-07-03 11	1:48:51:13	14044	09731.139861
Last Reset Time	2014-07-03 12	2:01:51:868	3 14044	10511.868905
1 march 10 million 20	1 2021 07 02 07			
Current lime	2014-07-03 12	2:01:51:80	9 14044	10511.869287
Counter Name	Periodic value		Cumula	tive value
Elapsed Time	1 00:00:00:00		1 00:13:	00:729
Call Rate	0.000 cps		1.2	81 cos
	-+			
Incoming call created	0		T	θ
OutGoing call created	0		16	00
Total Call created			10	00
Current Call	Î Đ			
Successful call	<u>а</u>		1 10	оо ОО
Failed call	i é		1 10	θ
				<u></u>
Response Time 1	1 00:00:00:00		1 00:00:	00:010
Call Length	00:00:00:00		00:01:	00:013
	······ Test Termin	nated		
	the second second	the second se		

Figura 67: Cliente SIPp tercera prueba sin descarga con SIPp

Servidor SIPp:

root@ubuntu:/home/curaca 2.168.1.10 -mi 192.168.1	o/Escritorio/sig .10 -rtp_echo	op-3.4.0# ./:	sipp -sf se	erver.xml -nd -i 19	
Port Total-time Tot 5060 813.64 s	al-calls Transp 1121 UDP	Screen	[1-9]	: Change Screen	
0 new calls during 0.4 121 calls 0 Running, 163 Paused, 0 dead call msg (disca	16 s period 1 Woken up rded)	3 ms sched Peak was 20	uler resolu 60 calls, a	ition after 665 s	
3 open sockets 0 Total echo RTP pckts 0 Total echo RTP pckts	lst stream 2nd stream	0.000 last 0.000 last	period RTM period RTM	P rate (kB/s) P rate (kB/s)	
·····> INVITE	Message 1000	es Retrans 0	Timeout 0	Unexpected-Msg 590	
< 180 < 200	1000	0	θ		
> ACK	E-RTD1 1000	0	0	0	
> BYE < 200	1000 1000	0 0	θ	0	
t 4000msj Pause	Statistic	s Screen	[1-9	: Change Screen -	
Last Reset Time	2014-07-03	12:03:36.7	65617 1404	410616.765617	
Current Time	2014-07-03	12:03:37.1	81816 1404	410617. <mark>181816</mark>	
Counter Name	Periodic val	lue	Cumul	ative value	
Elapsed Time Call Rate	00:00:00:416	5000 5	00:13   1.	:33:654000 378 cps	
Incoming call created OutGoing call created Total Call created	0			121 0 121	
Successful call Failed call			1 1	900 0	
Response Time 1 Call Length	00:00:00:000	0000	00:00	:00:001000 :04:152000	

Figura 68: Servidor SIPp tercera prueba sin descarga con SIPp



> Total Llamadas 1000->Llamadas por segundo 100->Llamadas Simultaneas 80

Cliente SIPp:

<pre>root@carolina-laptop:/ho -m 1000 -r 100 -l 80 -s</pre>	me/carolina/Desca 1000 -nd -i 192.	rgas/sipp-3 168.1.20 re	.4.0# ./s. mote host	<pre>ipp -sf cliente.xml   192.168.1.7</pre>
In pcap pcap/prueb, npkt	s 9015			
max pkt length 180				
base port 7078				
Warning: open file limit	<pre>&gt; FD_SETSIZE; lin</pre>	miting max.	# of oper	n files to FD_SETSI
ZE = 1024				
Resolving remot	e host '192.168.1	.7' Done	1000	
Call rate/length) Do	Scenario S	creen	[1-9]	: Change Screen
100 0/0 mc)/1 000c 50	51 790 50 c	10tal-catts	192 169	1 7.5060(100)
100.0(0 ms)/1.0005 50	01 700.09 5	1000	192.100	.1.7.5000(00P)
Call limit reached (-m	1000), 0,000 s p	eriod 0 ms	schedule	r resolution
0 calls (limit 80)		Peak was 80	calls, a	fter 0 s
0 Running, 41 Paused,	9 Woken up			
0 dead call msg (disca	rded)	0 out-of-ca	ll msg (d	iscarded)
1 open sockets				
		-	action	Variation and the second
THEFT	Messages	Retrans	Timeout	Unexpected-Msg
INVITE>	1000	0	9	
180 <	1000	0	0	0
200 <	E-8TD1 1000	A	Å	Ä
200 4	L-1101 1000	•		
ACK>	1000	θ		
[ NOP ]				
Pause [ 1:00]	1000			θ
BYE>	1000	θ	θ	
200 <	1000	0	θ	θ
	Statistics	Screen	[1-9]	Change Screen
Start Time	1 2014-07-03 1	2.45.16.441	140441	8716 401736
Score Fine	1 2024 07 05 1			0/10/401/50
Last Recet Time	1 2014-07-03 1	.18.17.629	140441	1497 029203
Last Reset Time	1 2014-07-03 1	2.10.17.625	14044	1457.020205
Current Time	1 2011 07 02 1		14044	1407 000005
Current lime	2014-07-03 1	2:18:17:029	14044	1497.029305
				And the second s
	I food a dia and a			
Counter Name	Periodic value		Cumutat	ive value
Elapsed Time	00:00:00:00		00:13:0	06:618
Call Rate	0.000 cps		1.28	31 cps
• • • • • • • • • • • • • • • • • • • •	+		• • • • • • • • • • • •	
Incoming call created	9			6
OutGoing call created	0		100	Ð
Total Call created			190	96
Current Call	0			a) 199
	+		+	
Successful call	e e		190	)G
Failed call	Ð			6
Response Time 1	00:00:00:00		00:00:0	06:011
Call Length	00:00:00:00		00:01:0	06:014
	Test Ternir	nated		

Figura 69: Cliente SIPp cuarta prueba sin descarga con SIPp

### Servidor SIPp:

oot@ubuntu:/home/curacao/Escritorio/sipp-3.4.0# ./sipp -sf server.xml -nd -i 19 2.168.1.10 -mi 192.168.1.10 -rtp\_echo ------ Scenario Screen ------- [1-9]: Change Screen ---Port Total-time Total-calls Transport 5060 1052.79 s 1155 UDP 0 new calls during 0.624 s period 4 ms scheduler resolution 155 calls Peak was 260 calls, after 666 s 0 Running, 157 Paused, 2 Woken up 0 dead call msg (discarded) 3 open sockets 0 Total echo RTP pckts 1st stream 0 Total echo RTP pckts 2nd stream 0.000 last period RTP rate (kB/s) 0.000 last period RTP rate (kB/s) Messages Retrans Timeout Unexpected-Msg ----> INVITE 1000 0 0 781 <---- 180 1000 A <----- 200 1000 Θ Θ ----> ACK E-RTD1 1000 Θ 0 0 0 0 ----> BYE 1000 0 ----- 200 1000 0 [ 4000ms] Pause 1000 0 ------ Statistics Screen ------ [1-9]: Change Screen ---Start Time 2014-07-03 12:06:30.154057 1404410790.154057 2014-07-03 12:24:02.329854 1404411842.329854 Last Reset Time 2014-07-03 12:24:02.957599 1404411842.957599 Current Time | Periodic value | Cumulative value Counter Name Elapsed Time | 09:00:00:627000 Call Rate | 0.000 cps 00:17:32:803090 1.097 cps Incoming call created | 0 OutGoing call created | 0 Total Call created | 1155 Θ 1155 155 Current Call -----Successful call | 0 Failed call | 0 1060 Θ ................... 
 Response Time 1
 00:00:00:000000

 Call Length
 00:00:00:000000
 00:00:00:001000 00:01:04:152030 ······ Test Terminated ·····

Figura 70: Servidor SIPp cuarta prueba sin descarga con SIPp



# BIBLIOGRAFÍA


- (s.f.). Obtenido de Tipos de Comunicación en la Telefonia IP Telefonia Voz IP: http://www.telefoniavozip.com/voip/tipos-de-comunicacion-en-la-telefonia-ip.htm
- [PDF]Capitulo I : Marco teórico conceptual. (s.f.). Obtenido de [PDF]Capitulo I : Marco teórico conceptual: www.isis.ufg.edu.sv/www.isis/.../TE/.../621.385-Z25i-Capitulo%20I.pdf
- Instalación y configuración de la tarjeta X100P en Asterisk. (1 de Diciembre de 2009). Obtenido de Instalación y configuración de la tarjeta X100P en Asterisk: www.voztovoice.org/?q=node/241
- Authentic X100p Fxo Digium Asterisk Voip Pbx Centralilla. (s.f.). Obtenido de Authentic X100p Fxo Digium Asterisk Voip Pbx Centralilla: http://www.x100p.com/products/FXO.php
- Basto, J. M. (7 de 2007). [PDF] Tesis Completa Unidad Tizimín- Universidad Autónoma de Yucatán. Obtenido de [PDF] Tesis Completa - Unidad Tizimín- Universidad Autónoma de Yucatán: www.tizimin.uady.mx/tesis/tesis\_final\_jesus.pdf
- Bermeo, H. R. (2009). *Capitulo 1 ANTECEDENTES DE VoIP y ASTERISK*. Obtenido de Capitulo 1 ANTECEDENTES DE VoIP y ASTERISK: dspace.ups.edu.ec/bitstream/123456789/32/7/Capitulo1.pdf
- C.V, S. y. (s.f.). SISCOMEX. Obtenido de SISCOMEX: http://www.siscomex.com.mx/index.php?option=com\_content&view=article&id=51&Itemid=40
- Escudero Pascual, A., & Berthilson, L. (2007). Voz sobre IP. En *Redes Inalámbricas en los países en Desarrollo* (segunda ed., pág. 232). Limehouse Book Sprint Team.
- Escudero Pascual, A., & Berthilson, L. (2007). Voz sobre IP. En *Redes Inalámbricas en los países en Desarrollo* (pág. 229). Limehouse Book Sprint Team.
- Escudero Pascual, A., & Berthilson, L. (2007). Voz sobre IP. En *Redes inalámbricas en los países en Desarrollo* (pág. 230). Limehouse Book Sprint Team.
- Escudero Pascual, A., & Berthilson, L. (2007). Voz sobre IP. En *Redes Inalámbricas en los países en Desrrollo* (pág. 243). Limehouse Book Sprint Team.



- Escudero Pascual, A., & Berthilson, L. (2007). Voz sobre IP. En Redes Inalámbricas en los países en Desarrollo (segunda ed., pág. 234). Limehouse Book Sprint Team.
- Informática-Hoy. (s.f.). VOIP, ventajas y beneficios Informatica Hoy. Obtenido de http://www.informaticahoy.com.ar/voz-ip-voip/VOIP-ventajas-y-beneficios.php
- internet, V. M. (s.f.). VOIP-MUNDO proveedor de Voz IP o Telefonia IP. Obtenido de VOIP-MUNDO proveedor de Voz IP o Telefonia IP: http://www.voip-mundo.com/
- Introducción a la Telefonía. (s.f.). Obtenido de Introducción a la Telefonía: www.naser.cl/sitio/Down\_Papers/Introduccion%20a%20la%20telefonia.pdf
- La tecnologia a traves del tiempo. (s.f.). Obtenido de La tecnologia a traves del tiempo: http://www.taringa.net/posts/apuntes-y-monografias/7652098/La-tecnologia-a-traves-del-tiempo.html
- Larios Lira, F., Rivera, A., & Zapata, Y. (s.f.). Voz sobre IP implementada con Asterisk: Teoría y Práctica.
- noa. (s.f.). Asterisk ® The open source PBX. Obtenido de Asterisk ® The open source PBX: http://www.noasistemas.es/tutorial-sobre-telefonia-ip/121
- PBX NICARAGUA. (s.f.). Obtenido de http://www.pbxnicaragua.com/
- Protocolos en la Telefonia IP, Protocolos VoIP Telefonia Voz IP. (s.f.). Obtenido de http://www.telefoniavozip.com/voip/protocolos-en-la-telefonia-ip.htm
- Ramos Bermeo, H. (2009). (PDF) Capitulo 1 Antecedentes de VoIP y Asterisk. Obtenido de (PDF) Capitulo 1 Antecedentes de VoIP y Asterisk: dspace.ups.edu.ec/bitstream/123456789/32/7/Capitulo1.pdf
- Restrepo, J. (3 de Ener de 2012). Implementando un servidor de Asterisk (método Julio Restrepo). Obtenido de http://juliorestrepo.wordpress.com/2012/01/03/implementando-un-servidor-de-asterisk-metodojulio-restrepo/

Rubio Jiménez, M. A. (s.f.). (PDF) Asterisk. Obtenido de bosqueviejo.org/book/pdf/asterisk\_zaptel



- SeMaToVe. (7 de Agosto de 2008). Asterisk, como tener una centralita de telefonos en tu casa. Obtenido de http://sematove.wordpress.com/2008/08/07/asterix-como-tener-una-centralita-de-telefonos-en-tucasa/
- Sistemas y Construcción de México S.A de C.V. (s.f.). S/SCOMEX. Obtenido de http://www.siscomex.com.mx/index.php?option=com\_content&view=article&id=51&Itemid=40
- Solutec, G. (s.f.). Asterisk-Central Telefónica PBX. Obtenido de Asterisk-Central Telefónica PBX: www.solutecperu.com/spsac/asterisk-central-telefónica-pbx
- *Tipos de Comunicación en la Telefonia IP Telefonia Voz IP*. (s.f.). Obtenido de Tipos de Comunicación en la Telefonia IP Telefonia Voz IP: http://www.telefoniavozip.com/voip/tipos-de-comunicacion-en-la-telefonia-ip.htm