ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"IMPLEMENTACIÓN DEL PROTOCOLO SS7 SOBRE CONEXIONES ENTRE DOS SERVIDORES ASTERISK"

TESINA DE SEMINARIO

Previo la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Freddy Andrés Huayamave Vera

Pedro Eduardo Rivadeneira Loor

GUAYAQUIL – ECUADOR

2013

AGRADECIMIENTO

Agradezco a Dios por haberme dado las fuerzas para seguir adelante en una etapa muy difícil en mi vida. Agradezco a mis padres, a mi familia, a mis amigos y a mi pareja por haberme dado ánimos cuando más los necesitaba. De antemano les quedo muy agradecido a todas las personas que me ayudaron de una u otra manera a lo largo de mi carrera universitaria.

Freddy Andrés Huayamave Vera

Ha sido un largo trecho el que ha habido que recorrer para culminar esta etapa de mi vida. Camino largo y tortuoso que hubiera sido difícil de recorrerlo sin la ayuda y apoyo de las personas que me apoyaron.

Es por eso que ahora quiero presentar mis sentimientos de profunda gratitud y agradecimiento primeramente a Dios quien me dio la fuerza y energía para no desmayar, luego a mis padres por su ejemplo, esfuerzo y sacrificios que hicieron por mí, a mi familia y a mis profesores, especialmente a aquellos que estuvieron más cerca a mi, quienes compartieron sus conocimientos y experiencia conmigo. A mis amigos que siempre estuvieron junto a mí en los momentos difíciles y por último a la ESPOL, Institución que me acogió durante todos estos años.

Pedro Eduardo Rivadeneira Loor

DEDICATORIA

Este trabajo se lo dedico a mi madre; a pesar de todas las adversidades ella siempre estuvo a mi lado y fue mi inspiración para poder culminar esta gran meta en mi vida.

Freddy Andrés Huayamave Vera

En gratitud y reconocimiento a su abnegación y amor hacia mi persona, dedico este trabajo a mis padres.

Pedro Eduardo Rivadeneira Loor

TRIBUNAL DE SUSTENTACIÓN

_____

Ing. Washington Adolfo Medina Moreira.

Profesor del Seminario de Graduación

_____

Ing. Luis Fernando Vásquez Vera.

Profesor Delegado del Decano

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesina, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral".

(Reglamento de Graduación de la ESPOL)

_____ _____

Freddy Huayamave Vera **Pedro Rivadeneira Loor**

RESUMEN

Este proyecto tiene como propósito general la implementación de 3 protocolos (SIP, IAX, SS7) sobre Asterisk, indicando sus ventajas y desventajas entre sí; esto con la finalidad de otorgar una guía acerca de cuál protocolo utilizar en la creación de una central telefónica con base en Asterisk.

La implementación de este proyecto busca demostrar las ventajas que se tienen al implementar el protocolo SS7, comparándolo con los protocolos SIP e IAX; así también como administrar y garantizar de forma eficiente tanto los recursos económicos como tecnológicos que nos ofrece un E1.

Por medio de este proyecto daremos una guía para el desarrollo e implementación del servicio de telefonía a través de Voz sobre IP; con esto se reducirán los costos en la adquisición, mantenimiento y monitoreo de equipos de comunicación evitando gastos innecesarios en centrales telefónicas que representen costos excesivos, considerando las opciones existentes en el mercado.

INDICE GENERAL

AGRADECIMIENTO II

DEDICATORIA IV

TRIBUNAL DE SUSTENTACIÓN V

DECLARACIÓN EXPRESA VI

RESUMEN VII

INDICE GENERAL VIII

INDICE DE FIGURAS XIII

ABREVIATURAS XVIII

INTRODUCIÓN XXV

CAPÍTULO 1 1

1.1 Antecedentes 1

1.1.1 Comparación de tecnologías 3

1.1.2 Comienzos de la voz sobre IP 4

1.2 Justificación 6

1.3Descripción del problema 7

1.3.1Objetivo General 8

1.3.2 Objetivos Específicos 8

1.4 Metodología 9

CAPÍTULO 2 11

2.1 Definición de Asterisk 11

2.1.1Funcionalidades de Asterisk 12

2.2 Que es VoIP 14

2.2.1Telefonía Convencional vs TelefoníaIP 15

2.2.2Telefonía IP a bajos costos 19

2.2.3Tiempos Muertos en las Comunicaciones 20

2.2.4Conmutación de Paquetes en la Telefonía IP 21

2.2.4.1 Datagramas 22

2.2.4.2Circuitos Virtuales 22

2.2.5VentajasVoIP 24

2.2.6Codecs VoIP 25

2.3 Sistema de señalización 27

2.3.1 Protocolos de señalización 27

2.4 Definición del Sistema de Señalización N° 7 28

2.4.1 Ventajas de la señalización SS7 29

2.4.2 Arquitectura de SS7 30

2.4.3 Capas del modelo SS7 34

2.4.3.1 La capa física (MTP-1) 34

2.4.3.2 Parte de transferencia de mensajes del nivel 2 (MTP-2) 35

2.4.3.3 Parte de transferencia de mensajes del nivel 3 (MTP-3) 35

2.4.3.4 Señalización de control de conexión de pieza (SCCP) 37

2.4.3.5 La parte de usuario RDSI (ISUP) 37

2.4.3.6Capacidades de la parte de aplicación de transacciones (TCAP) ... 38

2.5 Apoyo SS7 en Asterisk 39

2.5.1 LIBISUP 39

2.5.2 SS7box 39

2.5.3 Chan_ss7 39

2.5.4 Libss7 40

2.6 Session Initiation Protocol (SIP) 40

2.6.1Funcionalidades 41

2.6.2 Mensajes 43

2.7 Protocolo IAX2 45

CAPÍTULO 3 52

3.1 Asterisk 52

3.1.1Requerimiento para iniciar el proyecto en Asterisk 53

3.1.1.1 Hardware 54

3.1.1.2 Software 55

3.1.1.3 Virtualización de host y teléfonos IP 57

3.2 Generalidades Dahdi 58

3.3 Configuración de Archivos del proyecto Asterisk 59

3.3.1 SIP.CONF 60

3.3.2 Configuración IAX.CONF 64

3.3.3 Configuración EXTENSIONS.CONF 66

3.3.4 ConfiguraciónSS7.CONF 68

3.3.5 Configuración SYSTEM.CONF 69

3.3.6 Configuración dahdi_channels.conf 69

CAPÍTULO4 71

4.1 Configuración del Equipo de Interconexión 71

4.2 Creación del Servicio 75

CAPÍTULO 5 81

5.1 Etapa de pruebas entre servidores (instalación) 81

5.2 Inicializando con Asterisk 91

5.3 Pruebas con Equipos 100

5.4 Tabla Comparativa entre Protocolos 145

5.5 Guía de Elección en Base a Costes y Eficiencia 148

CONCLUSIONES 154

RECOMENDACIONES 158

BIBLIOGRAFÍA 160

ANEXOS 164

INDICE DE FIGURAS

Fig.1.1. Diseño de una central telefónica convencional [1] 2

Fig.2.1. Representación de una llamada VoIP 19

Fig. 2.2. Tipos de codificación VoIP [10] 26

Fig. 2.3. SS7 Signaling Points [21] 31

Fig. 2.5. MTP nivel 1 [18] 35

Fig. 2.6. MTP [18] 36

Fig. 2.7 registros en llamada SIP [13] 43

Fig. 2.7 registros en llamada IAX [15] 47

Fig. 3.1 Cable cruzado para E1/T [22] 55

Fig. 3.2. Diagrama esquemático de conexión 57

Fig. 3.3. Grupo de extensiones (SIP) en cada servidor. 58

Fig. 3.4. Grupo de extensiones (IAX2) en cada servidor. 58

Fig. 4.1. Red Implementada 74

Fig. 4.2 Ejemplo de extensiones 76

Fig. 4.3 Diagrama de Flujo del servicio. 76

Fig. 4.4 Diagrama de Flujo del plan de marcado en el servicio. 79

Fig. 5.1 Instalación de UBUNTU 84

Fig. 5.2. Selección de lenguaje y Teclado. 85

Fig. 5.3. Aceptar Selección de Lenguaje 85

Fig. 5.4. Nombre del Servidor 86

Fig.5.5 Encriptación Directorio 87

Fig. 5.6. Configuración de Reloj 87

Fig. 5.7. Selección de Disco 88

Fig. 5.8 Partición del disco 88

Fig. 5.9. Administración de Actualizaciones del sistema. 89

Fig. 5.10. Lista de Programas. 90

Fig. 5.11. GRUB boot loader 90

Fig. 5.12. Instalación Completa 91

Fig. 5.13. Grafica de Menuselect 93

Fig 5.14. Configuración de IP DHCP 96

Fig. 5.15 Configuración de IP STATIC 96

Fig.5.16 Diagrama de flujo - Registro de usuario SIP 102

Fig.5.16.1 Analizado de paquetes - Registro de usuario SIP (B) 103

Fig. 5.17. Llamada SIP visto en wireshark 104

Fig 5.18. Llamada SIP Servidor A 104

Fig. 5.19 Llama entre dos usuarios SIP en Asterisk 105

Fig. 5.20 Diagrama de Flujo Llamada SIP entre servidores 106

Fig. 5.21 Culminación de llamada SIP visto en wireshark 107

Fig. 5.22 Registro de usurio IAX visto en wireshark 108

Fig. 5.23 Establecimiento de llamada IAX visto en wireshark 108

Fig. 5.24 Llamada en curso IAX visto en wireshark 109

Fig. 5.25 Culminación de llamada IAX visto en wireshark 109

Fig. 5.26 Llama IAX en Asterisk 110

Fig. 5.27 Extensiones SIP e IAX2 registradas 111

Fig. 5.28 Llamada mostrando Playing 114

Fig.5.29 Proceso completo visto desde Asterisk 115

Fig. 5.30 Plan de marcado para SS7 118

Fig. 5.31 Proceso de llamada en SS7 119

Fig. 5.32 Cuentas SIP registradas en zoiper (servidor A) 120

Fig. 5.33 Cuentas SIP registradas en zoiper (servidor B) 121

Fig. 5.34 Registro de extensiones servidor A 122

Fig. 5.35 Registro de extensiones servidor B 122

Fig. 5.36 Servidor A enlazado con SIP e IAX 123

Fig. 5.37 Servidor B enlazado con SIP e IAX 123

Fig. 5.38 Proceso de llamada SIP (servidor A) 124

Fig. 5.39 Proceso de llamada SIP, RINGING (servidor A) 125

Fig. 5.40 Proceso de llamada (servidor B) 125

Fig. 5.41 Proceso de llamada, RINGING (servidor B) 126

Fig. 5.42 Cinco llamadas simultáneas (servidor A) 126

Fig. 5.43 Cinco llamadas simultáneas (servidor B) 127

Fig. 5.44 Carga en el servidor A 128

Fig. 5.45 Carga en el servidor B 128

Fig. 5.46 Cuentas registradas en el servidor A 129

Fig. 5.47 Cuentas registradas en el servidor B 130

Fig. 5.48. Cuentas IAX del servidor A registradas en softphone 130

Fig. 5.49 Cuentas IAX del servidor B registradas en softphone 131

Fig.5.50 Llamadas IAX del servidor A al servidor B 131

Fig.5.51 Llamadas IAX del servidor A al servidor B, RINGING 132

Fig.5.52. Llamadas del servidor B al servidor A 133

Fig.5.53. Llamadas del servidor B al servidor A,RINRING en servidor A 133

Fig.5.54. Cinco llamadas del servidor B al servidor A (servidor A) 134

Fig.5.55. Cinco llamadas del servidor B al servidor A (servidor B) 134

Fig. 5.56 Llamada con respuesta del IVR 135

Fig. 5.57 Redireccionamiento de llamada 136

Fig. 5.58 Proceso de transferencia de llamada 137

Fig. 5.59 Proceso de transferencia de llamada 138

Fig. 5.60 Transferencia de llamada completa 138

Fig. 5.61 Plan de marcado para llamada ss7 con SIP 140

Fig. 5.62 Proceso de llamada ss7 con SIP, RINRING 141

Fig. 5.63 Proceso de llamada ss7 con SIP, llamada establecida 141

Fig. 5.64 Plan de marcado para llamada ss7 con IAX 143

Fig. 5.65 Proceso de llamada ss7 con IAX 143

Fig. 5.66 Proceso de llamada ss7 con IAX, llamada establecida 144

Fig. 5.67 Mensaje de culminación de llamada en SS7 145

ABREVIATURAS

Asterisk	Es un programa de software libre (bajo licencia GPL) que proporciona funcionalidades de una central telefónica (PBX).
ATA	Un Adaptador de Teléfono Analógico o ATA es un dispositivo utilizado para conectar uno o más teléfonos analógicos estándar a un sistema de telefonía digital o a un sistema de teléfono no estándar.
CAS	Señalización por Canal Asociado (CAS) La información de señalización se transmite por los mismos canales que la información de usuario, es decir que la señalización viaja por el canal asignado para el efecto y es uno de los de voz.
CCS	Señalización por canal común (CCS) La información de señalización se transmite por un canal diferente al empleado por la información de usuario,constituyendo una red independiente denominada red de señalización.
Codecs	Es la abreviatura de codificador-decodificador. Describe una especificación desarrollada en software, hardware o una combinación de ambos, capaz de transformar un archivo con un flujo de datos (stream) o una señal.

DAHDI	Las siglas DAHDI hacen referencia a Digium/Asterisk Hardware Device Interface, es decir, una interfaz para toda la lista de productos Digium y compatibles que conecta con el sistema Asterisk, considerando que hablamos de productos que conectan concretamente con la PSTN, la telefonía clásica, de toda la vida
DHCP	De sigla en inglés de Dynamic Host Configuration Protocol, en español protocolo de configuración dinámica de hostes un protocolo de red que permite a los clientes de una red IP obtener sus parámetros de configuración automáticamente.
Delay	Retraso en el tiempo que tarda en ejecutarse una instrucción.
E1	Es un formato europeo de transmision digital ideado por el ITU-TS y su nombre fue dado por la administración de la Conferencia Europea de Correos y Telecomunicaciones (CEPT).
GTT	Global Title Translation, Traducción de título global.
GSM	Global System for Mobile communications, Sistema global para comunicaciones móviles

IP Es una etiqueta numérica que identifica, de manera lógica y jerárquica, a un interfaz de un dispositivo dentro de una red que utilice el protocolo IP (Internet Protocol),

IAX (Inter-Asterisk eXchange protocol) es uno de los protocolos utilizado por Asterisk, Es utilizado para manejar conexiones VoIP entre servidores Asterisk, y entre servidores y clientes que también utilizan protocolo IAX.

IVR Interactive Voice Response. Es una tecnología de telefonía en el que un usuario con un teléfono por tonos puede interactuar con una base de datos para adquirir o ingresar información. Este sistema no requiere de intervención humana al teléfono ya que la interacción del usuario está limitada a las acciones que el sistema le permita realizar.

ISUP Integrated Services Digital Network User Part, Parte de usuario de la red digital de servicios integrados (RDSI)

LAN Red De Área Local, es una red informática que interconecta los ordenadores en un área limitada, como una casa, la escuela, laboratorio de informática, o un edificio de oficinas utilizando medios de red.

MTP	El Protocolo de Transferencia de Medios, que lo introdujo, como un protocolo para dispositivos de almacenamiento inteligente basado y compatible con Picture Transfer Protocol (PTP)
NAT	NAT (Network Address Translation - Traducción de Dirección de Red) es un mecanismo utilizado por routers IP para intercambiar paquetes entre dos redes que asignan mutuamente direcciones incompatibles.
PC	Una computadora personal u ordenador persona
PBX	Un PBX (siglas en inglés de Private Branch Exchange),o más bien Central Secundaria Privada Automática; es en realidad 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
PSTN	Public Switched Telephony Network, o bien RPTC (Red Pública Telefónica Conmutada o también llamada RTC o RTB) representa todos los dispositivos y medios de transmisión y conmutación necesarios para comunicar dos terminales mediante un circuito que se establece para esa comunicación, y que cuando termina desaparece.

RTC	Red Telefónica Conmutada es una red de comunicación diseñada primordialmente para transmisión de voz, aunque pueda también transportar datos
RTP	Las siglas de Real-time Transport Protocol (Protocolo de Transporte de Tiempo real). Es un protocolo de nivel de sesión utilizado para la transmisión de información en tiempo real, como por ejemplo audio y vídeo en una video-conferencia
SS7	El sistema de señalización por canal común n.º 7 es un conjunto de protocolos de señalización telefónica empleado en la mayor parte de redes telefónicas mundiales
SIP	Session Initiation Protocol (SIP o Protocolo de Inicio de Sesiones) es un protocolo permiten trabajar con sesiones multimedia en tiempo real y permite que los diferentes agentes de usuarios puedan encontrarse y ponerse de acuerdo con el tipo de sesión que quieran compartir
Switch	Es el dispositivo analógico que permite interconectar redes operando en la capa 2 o de nivel de enlace de datos del modelo OSI.
SSP	Service Switching Point (Punto de Servicio de Conmutación) Programa capaz de mandar señales activadas para puntos

	control de servicio y para consultar estas bases de datos para información; para procesar llamadas de teléfono
STP	Shield Twisted Pair (Par de Cobre Trenzado Protegido) Una línea de transmisión de dos cables metálicos torcidos de cobre que es protegida por una funda de material.
SCP	Service Control Point (Punto de Servicio de Control) Un programa que habilita a las computadoras transportadoras a ofrecer mejores servicios atendiendo números 800, facturar llamadas por cobrar y llamadas en conferencia, como también tarjetas de crédito, implicando al cliente con interacción de datos
SDL	La señalización de enlace de datos se recomienda tener un canal full-duplex ("bidireccional"), que opera a una velocidad de transferencia de bits estándar de 64 kbps
SL	Denomina canal de señalización (SL,Signaling Link) concierne al procedimiento de control de línea necesario para fiabilizar la transmisión de mensajes de señalización.
SCCP	Skinny Call Control Protocol o SCCP es un protocolo propietario de control de terminal define como un conjunto de mensajes entre un cliente ligero y el administrador de la llamada

TCAP	Transaction Capabilities Applications Protocol, Protocolo de aplicaciones para capacitación de transacciones
TCP	Transmission Control Protocol (en español Protocolo de Control de Transmisión) o TCP, es uno de los protocolos fundamentales en Internet
T1	Se utiliza para designar circuitos digitales que funcionan a velocidades de 1,544 Mbit/s
UDP	User Datagram Protocol (UDP) es un protocolo del nivel de transporte basado en el intercambio de datagramas (Encapsulado de capa 4 Modelo OSI).
VoIP	Voz sobre Protocolo de Internet, también llamado Voz sobre IP, Voz IP, VozIP, (VoIP por sus siglas en inglés, Voice over IP),

INTRODUCIÓN

Es necesario tener conocimiento de las tecnologías que existen en el mercado de telefonía. Habiendo elegido la implementación de VoIP por medio de Asterisk.Esto quiere decir que la señal de voz se envía en forma digital por paquetes en lugar de las formas tradicionales (analógica) por medio de una compañía telefonía convencional.

Con la implementación de este proyecto se puede demostrar la utilidad que tiene el protocolo SS7 sobre conexiones E1, las mismas que nos van a proveer 30 canales de voz simultáneamente, aprovechando los beneficios que nos brindan los mismos.

Esta implementa consta de dos servidores Asterisk, en cada uno va a estar instalada una tarjeta D110P/D410P, ambos servidores estar interconectados por un switch o un router, al utilizar uno de estos dispositivos podremos monitorear el tráfico que va de un servidor a otro y así obtener los resultados esperados.

El principal objetivo de este proyecto es implementar telefonía IP entre dos servidores Asterisk utilizando los protocolos SS7, SIP, IAX; realizando una comparación entre los protocolos.

CAPÍTULO 1

ANTECEDENTES Y JUSTIFICACIÓN

1.1 Antecedentes

Las redes telefónicas se componen, fundamentalmente, del propio aparato telefónico conectado mediante redes de cableado urbano a las centrales telefónicas locales (*Central Office o Local Exchange*). Por otro lado, todas las centrales locales están interconectadas entre sí mediante centrales intermedias de tránsito(*Tandem Office o Toll Office*).

Las centrales telefónicas constan de un equipo de conmutación, que es el que selecciona el circuito por el que se establece la llamada, y de equipos de transmisión, que es el que transmite la señal al resto de centrales con las que esta interconectada.

Durante el siglo XX, la mecánica de los primeros equipos de conmutación fue sustituida por mecanismos eléctricos, electrónicos y finalmente por circuitos digitales. De la misma forma, los sistemas de transmisión analógicos (ver figura 1.1), con cable coaxial o radio, fueron sustituidos por sistemas digitales sobre fibra óptica o radio digital[1].

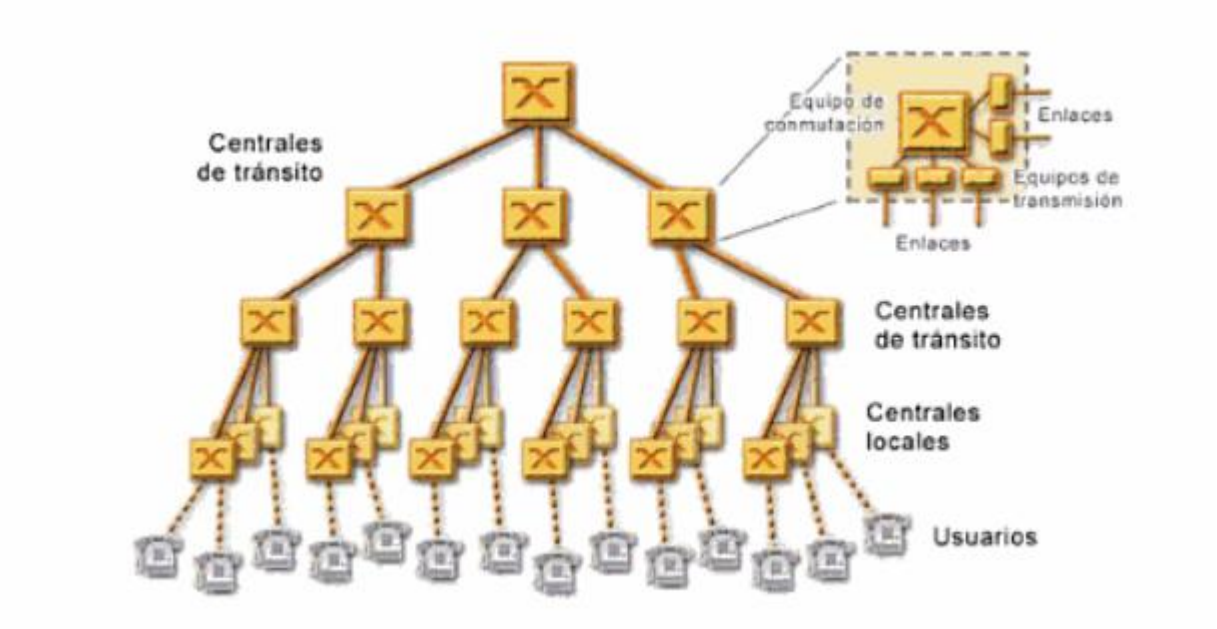


Fig.1.1. Diseño de una central telefónica convencional [1]

A pesar de ello, la evolución de la red de telefonía ha tenido algo en común: se trata de una red de conmutación de circuitos. Esto quiere decir que para realizar una comunicación se van interconectando circuitos de los distintos tramos de redes, hasta que entre uno y otro participante quede establecida una conexión directa en exclusividad. Se puede decir que entre ambos extremos existe un circuito ininterrumpido que los interconecta durante todo

el tiempo que dura la comunicación, independientemente de que se hable o no.

El gran inconveniente de las señales analógicas es que se ven afectadas, con relativa facilidad, por los ruidos introducidos por el entorno y por los propios sistemas empleados en su transmisión. Los ruidos e interferencias son de naturaleza analógica, igual que la señal, con lo que en el extremo distante no es fácil de distinguir qué parte de la señal recibida se corresponde con la original, y qué otra parte se corresponde con el ruido. Por el contrario, las señales digitales tienen la gran ventaja de ser muy resistentes al ruido, ya que en el extremo distante se pueden identificar muy bien los ceros y los unos originales, y separarlos de la señal analógica del ruido[1].

1.1.1 Comparación de tecnologías

Una línea telefónica comprende el par de hilos de cobre que llega a casa desde la central, y por extensión, también recibe este nombre el propio servicio telefónico. Desde el punto de vista técnico, una línea telefónica puede transmitir cualquier frecuencia comprendida entre los 300 y los 3400 Hz, lo que quiere decir que el ancho de banda es de 3100 Hz.

La voz humana emite frecuencias que pueden llegar a los 10 Khz, pero como la línea telefónica corta todas las frecuencias superiores a los 3400 Hz,

estoes lo que causa que la voz por teléfono tenga ese sonido tan particular.No obstante, la línea telefónica se utiliza para ofrecer otros servicios, como son el fax o la transmisión de datos. En estos casos, un ancho de banda de 3100 Hz limita la velocidad de transmisión de datos a 56kbps. Esto quiere decir que, aunque la línea telefónica ofrezca una calidad de sonido excelente, tenga grandes limitaciones para la transmisión de datos.

Las redes de datos, como IP, tienen unos requerimientos distintos: por un lado admiten un mayor retardo, incluso, no importa si unos paquetes tardan más en llegar que menos; por otro lado, si el orden de la información se altera durante la transmisión, siempre se puede recompensar en destino sin problemas. Los pequeños retardos de las redes de datos (del orden de los milisegundos) no tienen la másmínima importancia a la hora de recibir un correo electrónico o de ver una página Web. Lo que si tiene gran importancia para las redes de datos es que llegue al destino absolutamente toda la información, sin que nada se pierda. Para la voz, sin embargo, lo importante es que llegue la información en tiempo real, no importando que se pierdan pequeños fragmentos mientras que el resultado siga siendo comprensible.

1.1.2 Comienzos de la voz sobre IP

Los primeros años de Internet pasaron en un entorno militar y universitario casi exclusivamente. Con los 90, la red fue evolucionando hacia un entorno

cada vez más comercial, a la vez que veía como sus usuarios crecían de forma exponencial. No obstante, en esa época los usuarios solían conectarse a Internet utilizando una línea telefónica y un modem. En ese entorno no parecía tener sentido digitalizar la voz para transmitirla por Internet, ya que para acceder a Internet se utilizaba la red telefónica. Por otro lado, la mayoría de los PC no disponían de tarjeta de audio, eran ordenadores sordos y mudos.

En 1998, fabricantes de equipos, como Cisco o Lucent, decidieron desarrollar dispositivos (*routers)* especialmente pensados para manejar la voz. También se construyeron los primeros *gateways* que permitían hacer llamadas de PC a teléfono y viceversa. Ese año aparecieron algunas compañías en Estados Unidos que permitían a sus usuarios realizar, con sus propios aparatos telefónicos, llamadas telefónicas gratuitas de larga distancia a cambio de escuchar unos breves anuncios publicitarios al comienzo y finalización de la llamada.

Microsoft sacó en agosto de 1996 su conocido software *Netmeeting*, el cual permitía establecer comunicaciones de voz del tipo Pc a Pc. Muchos usuarios obtuvieron sus primeras experiencias de VoIP con este programa. En el año 2001 este software evolucionaria hacia Messenger.

En enero de 2001, Vonage empezó a ofrecer servicios de VoIP para empresas. Desde entonces los servicios de VoIP han ido creciendo en el

entorno empresarial. Muchas empresas utilizan VoIP en sus centros de atención al cliente, teniendo como ejemplo a Skype, el software VoIP más extendido del mundo en la actualidad[1].

1.2 Justificación

Es necesario tener conocimiento de las tecnologías que existen en el mercado de telefonía. Habiendo elegido la implementación de VoIP por medio de Asterisk, se implementa hardware básico para la instalación y configuración de los servidores Asterisk; a la vez se utiliza el teléfono IP y el softphone, pudiendo integrar funciones de multiplexación e interconexiones entre equipos utilizando para esto el protocolo SS7.Todo esto será muy útil y de fácil manejo para la gestión y administración de la central telefónica.

Asterisk es un software que permite la construcción a costes extremadamente reducidos de soluciones de Voz sobre IP. Estas soluciones pueden sustituir a las actuales centrales telefónicas prácticamente todos los ámbitos imaginables, desde las pequeñas oficinas hasta los grandes entornos corporativos.

1.3 Descripción del problema

El proyecto a realizar consiste en la implementación del protocolo SS7 sobre conexiones entre dos servidores Asterisk

Asterisk es un software que permite la construcción de soluciones de Voz sobre IP (Central Telefónica VoIP), utilizando Asterisk en su entorno es posible olvidar las limitaciones tradicionales de las centrales telefónicas; esto es, no más problemas de alcanzar el máximo de extensiones posibles, evitando de esta forma el tener problemas de saturación que afectan actualmente al producto tradicional.

Al implementar este proyecto buscamos demostrar las ventajas acerca del protocolo SS7; como también garantizar de forma eficiente los recursos tecnológicos y económicos, a la vez se realizará una comparación con los protocolos SIP e IAX, para satisfacer el conjunto de necesidades del usuario final en relación al costo-beneficio.

Por medio de este proyecto daremos una guía para el desarrollo e implementación del servicio de telefonía a través de Voz sobre IP; con esto se reducirán los costos en la adquisición, mantenimiento y monitoreo de equipos de comunicación evitando gastos innecesarios en centrales telefónicas que representen costos excesivos, considerando las opciones existentes en el mercado.

Al implementar telefonía IP en esta red se pueden dar los servicios de 1-800, correo de voz, llamadas en espera entre otras.

1.3.1Objetivo General

Implementación de telefonía IP entre dos servidores Asterisk utilizando los protocolos SS7, SIP, IAX, realizando una comparación de costo y eficiencia entre ellos.

1.3.2 Objetivos Específicos

- Facilitar una guía básica para la configuración e instalación de servidores Asterisk.
- Demostrar y comprobar la señalización SS7 entre ambos servidores Asterisk.
- Indicar las ventajas y desventajas en el uso de SS7 frente a otros protocolos referentes a telefonía (SIP, IAX).
- Realizar pruebas sobre el funcionamiento de los servicios configurados entre los dos servidores Asterisk.

- Recopilar información técnica, a través de los resultados esperados, para completar la guía de elección que ayude a elegir el protocolo de telefonía a utilizar, dependiendo de los requerimientos del cliente.

1.4 Metodología

La implementación de este proyecto incluye un manual donde el lector encontrará una forma básica y sencilla para configurar Asterisk; la descripción de comandos, *codecs* y protocolos a utilizar se detallarán en un lenguaje sencillo facilitando la comprensión y funcionamiento del mismo para el usuario común.

A través de la configuración de los archivos *.conf* se realizará la conexión entre los dos servidores utilizando los protocolos SS7, SIP, IAX. Se utilizarán los tres protocolos mencionados con el objeto de indicar cuál es el mejor protocolo a utilizar, dependiendo de los requerimientos del usuario final. En este caso, la persona o empresa que requiera implementar una central telefónica utilizando Asterisk.

En la etapa de pruebas recolectaremos información acerca de parámetros como calidad de voz, ruido, *delay*, entre otros; esto con la finalidad de proceder a realizar la comparación de protocolos por medio de parámetros establecidos, cumpliendo con el objetivo general del proyecto.

Finalmente, se proporcionará una guía económica con precios y tecnologías actuales que implementen la telefonía VoIP esperando que el usuario pueda escoger de acuerdo a sus necesidades y poder adquisitivo, la tecnología y protocolo a utilizar al momento de crear una central telefónica para su empresa u oficina. Todo el software y código a utilizar en este proyecto será *open-source* y *freeware*; en caso de utilizar un hardware o software, con la licencia respectiva, se obtendrá los permisos necesarios para su operación.

CAPÍTULO 2

FUNDAMENTO TEÓRICOS

2.1 Definición de Asterisk

Asterisk es un software libre tipo PBX, es decir que funciona como una central secundaria privada y automática. Se puede emplear como una central telefónica conectada directamente a la red pública de teléfonos por medio de líneas troncales para gestionar, además de las llamadas internas, las entrantes y/o salientes, con autonomía sobre cualquier otra central telefónica. Principalmente para ejecutarse en Linux, pero puede trabajar con Windows (emulado) y BSD. Asterisk funciona a partir del protocolo IP y puede interactuar con gran parte de equipos de telefonía basado en los estándares de telefonía, al momento de implementar el hardware usado es relativamente económico.

Asterisk combina más de 100 años de experiencia en telefonía, y un potente conjunto de aplicaciones de telecomunicaciones altamente integrados.
El poder de asterisk está, en su naturaleza adaptable, complementado por el cumplimiento de estándares sin precedentes. Las aplicaciones como correo de voz, conferencia organizada, cola de llamadas y agentes, la música en espera y redireccionamiento de llamada son todas las características estándares integradas en el software.

Asterisk puede parecer un poco intimidante y complejo para un usuario nuevo, por lo que la documentación es muy importante para su crecimiento. Cabe señalar que el empleo de sistemas PBX previene de los teléfonos de toda una oficina, de manera independiente a la red de telefonía local pública (RTC), ya que el PBX trabaja como un switch de red y con ello se ahorra el empleo de una línea propia, con salidas de llamadas y cargos mensuales hacia la central telefónica que regresan nuevamente para establecer comunicación interna.

2.1.1Funcionalidades de Asterisk

- Autentificación de llamadas con respuesta automatizada.
- Almacenamiento y recuperación en base de datos
- Acceso directo al sistema interno.

- Configuración de música a elección para el proceso de espera, con un sistema de reproducción aleatoria y control de volumen.
- Configurable para trabajar con conferencia de voz.
- Configuración de llamada en espera.
- Desvío de llamadas al interno en el caso que la extensión está ocupada o no responde.
- Desvío de llamada variable.
- Empleo de agentes locales y remotos.
- Identificación de llamadas con opciones de bloqueo. Este sistema también se aplica a las llamadas en espera.
- Integración con Base de Datos.
- Monitorización de llamadas, con opciones de aparcamiento de llamadas.
- Marcación predictiva.
- Opciones de transferencia de llamadas.
- Opciones de marcado por nombre.
- Opciones de transferencia no supervisada de llamadas (automatizada).
- Opciones de registros detallados de llamadas.
- Opciones de privacidad.
- Protocolo de establecimiento abierto (OSP).
- Receptor de alarmas agregar mensaje.

- Recuperación de llamadas (DID y ANI).
- Sistema de menú en pantalla ADSI (Interfaz Analógico para presentación de Servicios).
- Sistema de grabación de llamadas.
- Sistema de escucha de llamadas.
- Tonos de llamada distintivos.

2.2 Que es VoIP

El termino VoIP que viene del inglés*Voice Over Internet Protocol* (voz sobre protocolo de internet), es una tecnología que permite la transmisión de la voz en forma de paquetes de datos, que emplea el protocolo IP *(Internet Protocol)* y envía la señal de voz en forma digital (paquetes), en lugar de enviarla (en forma digital o analógica) a través de circuitos utilizables sólo para telefonía, como la Red Telefónica Pública Conmutada de las compañías telefónicas convencionales.

Básicamente VoIP toma señales analógicas del flujo de audio para transformarlas en datos digitales y poder ser transmitidos a través de internet hacia un host determinado. Este tráfico puede circular por cualquier red IP, ya sea ésta por Internet o por redes de área local LAN [2].

Es importante conocer la diferencia entre Voz sobre IP y Telefonía IP. VoIP es el conjunto de normas, dispositivos y protocolos; en definitiva, la tecnología que permite la transmisión de la voz sobre el protocolo IP. La Telefonía IP o Telefonía Voz IP es el conjunto de nuevas funcionalidades de la telefonía que utilizan tecnología para realizar llamadas IP, por ejemplo a través de un teléfono IP.

En sus inicios las conversaciones mediante VoIP solían ser de baja calidad debido a que el acceso a internet y el ancho de banda por usuario era limitado, esto se vio superado por la tecnología actual y la proliferación de conexiones de banda ancha de mayor capacidad; un ejemplo de esto es el uso VoIP para transmitir llamadas de larga distancia, reduciendo costos implementados por las operadoras de telefonía convencional[3].

2.2.1Telefonía Convencional vs Telefonía IP

Los sistemas de telefonía convencional tradicionalmente han usado el sistema de conmutación por circuitos para implementación de sus servicios, esta metodología ha sido utilizada por más de 100 años. En este sistema cuando una llamada es realizada la conexión se mantiene durante todo el tiempo que perdure la comunicación, a esto se le denomina conmutación por

circuito porque la conexión está realizada entre dos puntos hacia ambas direcciones.

En una llamada telefónica normal, la central telefónica establece una conexión permanente entre ambos usuarios, conexión que se utiliza para llevar las señales de voz. En una llamada telefónica 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, por lo cual cada paquete puede utilizar un camino distinto para llegar a su destino[4].

Para entender el funcionamiento de una comunicación en telefonía IP, primero vamos a definir como se realiza una comunicación mediante conmutación de circuitos.

1. El usuario que realiza una llamada levanta el teléfono y escucha el tono de marcado. Esto indica que hay una conexión con el operador local de telefonía.
2. Marca el número de teléfono al cual desea llamar.
3. La llamada se transmite a través del conmutador (*switch*) de su operador apuntando hacia el teléfono marcado.
4. Una conexión es creada entre el teléfono del cual se está marcando y el de la persona que recibe la llamada; en este proceso el operador de telefonía utiliza varios conmutadores para lograr la comunicación entre las 2 líneas.

5. El teléfono de la persona a la que se llama suena y ésta contesta la llamada abriendo la conexión en el circuito.
6. Se realiza la comunicación por un tiempo determinado hasta que termina la llamada y se cuelga el teléfono.
7. Cuando se cuelga el teléfono el circuito se cierra automáticamente, de esta manera libera la línea y todas las líneas que intervinieron en la comunicación.

Para una comunicación en VoIP, vamos a suponer que las dos personas que se quieren comunicar entre ellas, tienen servicio por medio de un proveedor VoIP y los dos tienen sus teléfonos analógicos conectados a través de un adaptador digital-analógico llamado ATA.

Una comunicación mediante telefonía VoIP entre 2 usuarios funcionaria de la siguiente manera (ver figura 2.1.):

1. Se descuelga el teléfono, y al realizar esta acción se envía una señal al conversor analógico-digital llamado ATA.
2. Se recibe la señal en el ATA y envía un tono de llamado, esto nos indica que ya se tiene conexión a internet.
3. Los números son convertidos a digital por el ATA y guardados temporalmente, cuando se marca el número de teléfono de la persona que se desea llamar,.

4. La información del número telefónico es enviada a un proveedor VoIP. Las computadoras del proveedor VoIP analiza este número para asegurarse que está en un formato valido.
5. Se determina a quien corresponde este número y el proveedor lo transforma en una dirección IP.
6. Los dos dispositivos que intervienen en la llamada son conectados por el proveedor. Se envía una señal al ATA de la persona que recibe la llamada para que éste haga sonar el teléfono de la otra persona.
7. Una comunicación entre ambas computadoras es establecida una vez que la otra persona levanta el teléfono. Es decir que cada sistema está esperando recibir paquetes del otro sistema. En el medio, Los paquetes de voz, la comunicación son manejados por la infraestructura de internet de la misma forma que haría con un email o con una página web. Para obtener una comunicación entre los sistemas deben funcionar en el mismo protocolo. Los sistemas implementan dos canales, uno en cada dirección.
8. Durante los periodos de tiempo de la conversación, el sistema de la persona que llama y el sistema de la persona que recibe la llamada transmiten y reciben paquetes entre sí.
9. Cuando se cuelga el teléfono, se termina la llamada. En este momento el circuito es cerrado.

10. Finalizada la llamada el ATA envía una señal al proveedor de Telefonía IP para informar que la llamada ha sido finalizada..

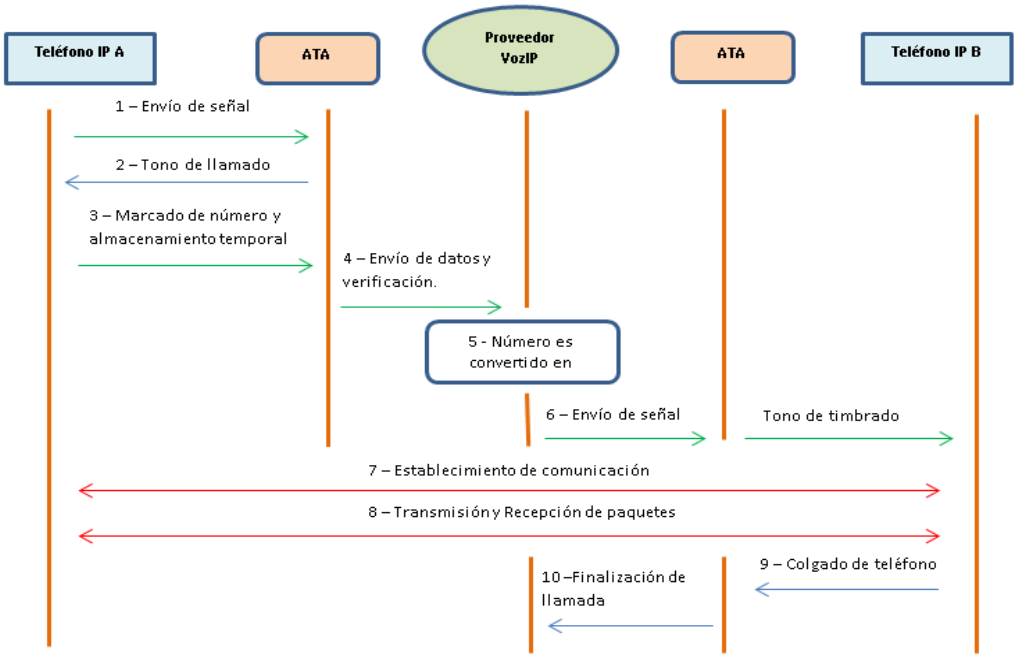


Fig.2.1.Representación de una llamada VoIP

2.2.2 Telefonía IP a bajos costos

A inicios de la telefonía convencional, cada llamada debía tener un cable que iba de un extremo al otro de la comunicación durante todo el tiempo que durara la misma. Si por ejemplo una persona ubicada en América del Norte tenía que realizar una llamada a otra persona en América del Sur, los conmutadores de su operadora telefónica conectaban cables a lo largo de todo el recorrido para formar un camino entre los 2 extremos de la

comunicación; si la llamada duraba 10 minutos se usaban esos cables conmutados que iban a lo largo de todo el recorrido entre los dos extremos durante la conversación, y se convertían en costos muy altos para este tipo de llamadas.

En la actualidad las comunicaciones telefónicas son mucho más eficientes, por eso su costo no es tan alto. Las voces son digitalizadas y pueden viajar junto a otras a través de un cable de fibra óptica(o algún otro medio físico) por la mayoría del trayecto. Estas llamadas son transmitidas con una calidad de 64 kbps en cada dirección, por un total de transmisión de 128kb (64kb de ida y 64kb de vuelta). Como existen 8Kb en un KiloByte (KB), esto se transforma en una transmisión de 16KB por cada segundo que el circuito este abierto. Entonces, en una comunicación de 10 minutos, el total transmitido seria de 9,600KB, lo que es prácticamente equivalente a 10 megas[5].

2.2.3Tiempos Muertos en las Comunicaciones

En una conversación telefónica generalmente los dos agentes no hablan al mismo tiempo, por lo tanto solo la mitad de la conexión se encuentra en uso en un determinado instante de tiempo; debido a esto se podría reducir a la mitad el tamaño de la conversación sin afectar la calidad de la comunicación.

Además, una gran cantidad de tiempo en las conversaciones es tiempo muerto, tiempo en el que ninguno de los dos habla. Si pudiéramos remover esos intervalos de tiempo muerto, el tamaño de la conversación sería todavía más pequeño. Entonces, en lugar de enviar una cadena continua de bits (ambos de silencio o ruidos), enviamos paquetes en los momentos que se produce ruido. Esta es la funcionalidad básica de la conmutación por paquete[5].

2.2.4 Conmutación de Paquetes en la Telefonía IP

Un Paquete en telefonía IP está formado por los datos y la información en donde consta la ruta a seguir hasta el del destino del paquete.

Los paquetes forman una cola y se transmiten lo más rápido posible; además la red puede seguir aceptando datos aunque la transmisión se torne lenta. Con esto existe la posibilidad de manejar prioridades enviando cierto tipo de paquetes antes que otros, de acuerdo a su importancia.

Existen dos técnicas para la conmutación de paquetes: los Datagramas y los Circuitos Virtuales [6].

2.2.4.1 Datagramas

Cuando se usa la técnica de datagrama cada paquete es tratado de forma independiente, conteniendo cada uno la dirección de destino. La red puede encaminar cada fragmento hacia el equipo receptor por rutas distintas sin garantizar que los paquetes lleguen en el orden adecuado ni que todos lleguen al destino[7].

2.2.4.2 Circuitos Virtuales

Los usuarios pueden intercambiar datos por medio de un circuito virtual que es un sistema de comunicación de conmutación transparente para el usuario. Un ejemplo de éste es el TCP o protocolo de control de transmisión, que es una forma de comunicación a través de paquetes. Por medio de éste la información o datos son empaquetados en bloques también conocidos como paquetes y su tamaño lo determina la red. Los paquetes por lo general incluyen cabeceras con información de control. Los paquetes se transmiten a la red para su direccionamiento al destino final. Al encontrarse un paquete con un nodo intermedio, éste guarda momentáneamente la información y envía los paquetes a otro nodo de acuerdo a las cabeceras de control. Cabe notar, que en este caso los nodos no precisan tomar decisiones de encaminamiento, porque la dirección a seguir es dada en el propio paquete.

Por otro lado, la conexión se mantiene abierta y constante por la conmutación de circuitos, y una pequeña conexión es abierta por el intercambio de paquetes que usan la telefonía IP. Esta conexión es suficiente para enviar una pequeña cantidad de información, entre sistemas, y es conocida como paquete. Esto funciona de la siguiente manera:

1. Para el envío de la información la computadora divide en pequeños paquetes la información y los etiqueta con una dirección en cada uno, indicando a los dispositivos de red a donde enviar los mismos.
2. Dentro de cada paquete hay una porción de la información donde está enviando, la voz.
3. La computadora emisora envía un paquete al router (A) más cercano y se olvida del mismo, éste envía el paquete a otro que se encuentra más cerca del destino, ese router (B) se lo envía a otro que se encuentra aún todavía más cerca del destino, y ese a otro más cerca, y así sucesivamente.
4. Cuando la computadora receptora finalmente recibe los paquetes (que pueden haber tomado caminos completamente diferentes para haber llegar ahí), El receptor usa las instrucciones que están dentro los paquetes para rearmar los datos en su estado original.
5. Los paquetes son enviados por la ruta menos congestionada esto hace que el intercambio de paquetes sea eficiente. El intercambio también

libera a las computadoras, de forma que éstas pueden también aceptar información proveniente de otras computadoras.

2.2.5 Ventajas VoIP

Costo: Una de las principales ventajas que mantiene la telefonía VoIP en comparación a la telefonía convencional debido a que utiliza la misma red para la transmisión de voz y datos.

Factibilidad de Conexión: Con VoIP se puede realizar una llamada desde cualquier punto con conectividad a internet. Debido a que la información trasmitida por los teléfonos IP es por medio del internet [8].

Servicios Integrados: Características por las cuales las operadoras de telefonía convencional cobran tarifas adicionales. Un servicio de VoIP las incluye:

- Identificación de llamadas.
- Servicio de llamadas en espera
- Servicio de transferencia de llamadas
- Repetir llamada
- Devolver llamada
- Llamada de 3 líneas (*three-way calling*).

- Estructura de Red Simplificada: Una llamada VoIP usa compresión de señal de voz y usa los circuito virtuales, transmitiendo los paquetes de diferentes tipos de datos y de diferentes llamadas por una línea al mismo tiempo, simplificando la estructura; a diferencia de una llamada telefónica convencional que requiere una gran red de centrales telefónicas conectadas entre sí mediante fibra óptica, satélites de telecomunicación.

2.2.6 Codecs VoIP

Un Codec, (del ingléscoder-decoder), convierte una señal de audio analógico en un formato de audio digital. Esta es la esencia del VoIP, la conversión de señales entre analógico-digital.

Los codecs realizan un muestreo de la señal de audio. Por ejemplo, el codec G.711 toma 64,000 muestras por segundo, realiza la conversión digital, comprime la señal y la transmite. En el otro extremo las 64,000 muestras son reconstruidas, se pierden fragmentos de audio muy pequeños que es imposible para el oído humano notar está perdida, y ésta suena como una sucesión continua de audio. Existen diferentes frecuencias de muestras de la señal en VOIP; esto depende del codec que se esté usando[10].

- 64,000 veces por segundo
- 32,000 veces por segundo

- 8,000 veces por segundo

Lo que les permite a los *codecs* tomar las muestras, ordenar, comprimir y empaquetar los datos son algoritmos muy avanzados. El algoritmo CS-ACELP (conjugate-structure algebraic-code-excited linear prediction) es uno de los algoritmos más comunes en VoIP. CS-ACELP ayuda a organizar el ancho de banda disponible.

Codec Information				Bandwidth Calculations					
Codec & Bit Rate (Kbps)	Codec Sample Size (Bytes)	Codec Sample Interval (ms)	Mean Opinion Score (MOS)	Voice Payload Size (Bytes)	Voice Payload Size (ms)	Packets Per Second (PPS)	Bandwidth MP or FRF.12 (Kbps)	Bandwidth w/cRTP MP or FRF.12 (Kbps)	Bandwidth Ethernet (Kbps)
G.711 (64 Kbps)	80 Bytes	10 ms	4.1	160 Bytes	20 ms	50	82.8 Kbps	67.6 Kbps	87.2 Kbps
G.729 (8 Kbps)	10 Bytes	10 ms	3.92	20 Bytes	20 ms	50	26.8 Kbps	11.6 Kbps	31.2 Kbps
G.723.1 (6.3 Kbps)	24 Bytes	30 ms	3.9	24 Bytes	30 ms	33.3	18.9 Kbps	8.8 Kbps	21.9 Kbps
G.723.1 (5.3 Kbps)	20 Bytes	30 ms	3.8	20 Bytes	30 ms	33.3	17.9 Kbps	7.7 Kbps	20.8 Kbps
G.726 (32 Kbps)	20 Bytes	5 ms	3.85	80 Bytes	20 ms	50	50.8 Kbps	35.6 Kbps	55.2 Kbps
G.726 (24 Kbps)	15 Bytes	5 ms		60 Bytes	20 ms	50	42.8 Kbps	27.6 Kbps	47.2 Kbps
G.728 (16 Kbps)	10 Bytes	5 ms	3.61	60 Bytes	30 ms	33.3	28.5 Kbps	18.4 Kbps	31.5 Kbps
G722_64k(64 Kbps)	80 Bytes	10 ms	4.13	160 Bytes	20 ms	50	82.8 Kbps	67.6Kbps	87.2 Kbps
ilbc_mode_20(15.2Kbps)	38 Bytes	20 ms	NA	38 Bytes	20 ms	50	34.0Kbps	18.8 Kbps	38.4Kbps
ilbc_mode_30(13.33Kbps)	50 Bytes	30 ms	NA	50 Bytes	30 ms	33.3	25.867 Kbps	15.73Kbps	28.8 Kbps

Fig. 2.2. Tipos de codificación VoIP [10]

2.3 Sistema de señalización

La señalización en la red de telecomunicaciones se basa en la señalización-establecimiento de llamada, conexión, desmontaje y la facturación. Las dos formas de señalización utilizada por la red son:

- Señalización de canal asociado (CAS)
- Señalización por canal común (CCS)

La señalización es un proceso por el cual nos aseguramos que la señal de voz se origina, transmite, conmuta, entrega correctamente, y permite generar datos para la gestión de la operación y la facturación de la llamada.

2.3.1 Protocolos de señalización

Los protocolos de señalización en VOIP cumplen funciones similares a sus homólogos de la telefónica tradicional, es decir tareas de establecimiento de sesión, control del proceso de la llamada, entre otros. Se encuentra en la capa 5 del modelo OSI, es decir en la Capa de Sesión.

Existen algunos protocolos de señalización que han sido desarrollados por diferentes fabricantes u organizaciones como la ITU o el IETF; algunos son:

- SIP
- IAX
- H.323

- MGCP
- SS7

2.4 Definición del Sistema de Señalización N° 7

El Sistema de Señalización de Canal Común número 7 (*Common Channel Signaling System* No 7, SS7 o C7) es un modelo universal utilizado en el mercado de las telecomunicaciones, determinado por el Parte de Estandarización de Telecomunicaciones de la *Internacional Telecommunications Unión* (ITU-T), en la cual establecen los distintos tipos de instrucciones y etiquetas por los que los compendios de malla en una tela oficial de telefonías (PSTN o *Public Switched Telephone Network*), mercantilizan los argumentos en la que una malla de señalización analógica, con el objeto de repercutir en la forma en el que se realizan llamadas telefónicas, que comprende los esquemas celulares y también los comunes [16].

El SS7 reside en la aclimatación de un canal distinto al canal de voz, utilizado solamente a la señalización. Esta aclimatación a consistir en dos objetivos, la primera que una malla transporte un dialogo, la segunda consiste en que una malla retenga la señalización; de esta forma se cumple que tanto el uno como el otro se realicen de forma autónoma [18].

Es por este sistema que el SS7 es un método de señalización que se define por la transferencia de paquetes de datos a grandes velocidades y por recibir la señal con distintos compendios de la red [16].

El tipo de señalización de malla habitual como es SS7, tiene como fin que ciertos nodos de la malla examinen los caracteres y de ahí trasladar cierta tarea establecida. De acuerdo a esta forma de señalización, el SS7 brinda significativos servicios para el beneficiario, los cuales son: identificador de llamadas, los números gratuitos 1-800, 1-700 y tipologías de portal de los números telefónicos. De lo dicho anteriormente, SS7 accede a que la señal se tome en cuenta siempre, así sea el caso de que no consten llamadas formadas. [18].

2.4.1 Ventajas de la señalización SS7

- Mejorar la flexibilidad y rapidez en la interconexión de una llamada.
- Mejor aprovechamiento de los enlaces entre las centrales.
- Mejora el control y la gestión (tasación) en cada llamada lo que ofrece al usuario una mejor calidad de servicio.
- Señalización es bidireccional en las centrales.

- Económica utiliza menos hardware que los sistemas anteriores.

- Admite que la información de señalización sea modificada en tiempo real, entre redes de conmutación.

2.4.2 Arquitectura de SS7

Uno a uno los puntos de señalización (*signaling point*), en la malla SS7 (elemento de la red), se identifica con un cifrado numérico ideal que tiene por nombre *point code (*código del punto). Los *point codes* son transportados en los mensajes entre los puntos de señalización con el objetivo de establecer el origen y el receptor de los mensajes. Los puntos de señalización manejan una lista de ruteo, para identificar el trayecto óptimo para que el mensaje pueda llegar a su puesto [21].

Hay tres clases de puntos de señalización en una red SS7(ver figura 2.3.):

- SSP (*Service Switching Point*. Punto de Intercambio de Servicio)

- STP (*Signal Transfer Point.* Punto de Transferencia de Señal)

- SCP (*Service Control Point*. Punto de Control de Servicio)

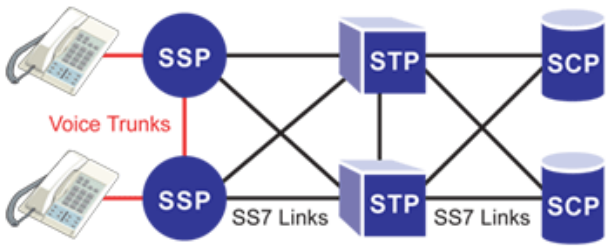


Fig. 2.3. SS7 Signaling Points [21]

Los SSP's son switches en los que se inician, finalizan o se devuelven las llamadas. Un SSP remite mensajes de señalización a otros SSPs con el fin de formar, gestionar e independizar cada uno de los circuitos de voz que se necesitan para cumplir una llamada.

En otra instancia, también se puede conseguir exportar postulaciones a alguna base de datos concentrada (un SCP), con el objetivo de decretar el reenvío de una llamada gratuita, un modelo puede de esto son los números 800 y 900. Un SCP contesta con el número telefónico real, inscrito con el número 800/900 marcado. Igualmente, es usual que se utilice un número variado si es que el primero se encuentra ocupado o si la llamada no se contesta en un tiempo determinado.

El tráfico de la red entre los puntos de señalización puede enrutarse por medio de un *switch* de paquetes llamado STP, cuya obligación es la de atrapar cada paquete y trasladarlo hacia uno de los muchos enlaces de señalización (*signaling link*), apoyándose en la información de ruteo introducida en el mensaje SS7.

STP ofrece un uso reformado de la red SS7 al excluir los enlaces inmediatos entre los diferentes puntos de señalización. También un STP puede llevar a cabo *universal title translation*, que es el modo en el que un componente de red final se establece por medio de los dígitos en el mensaje. Además un STP puede realizar las veces de un método para controlar la salida y llegada de los mensajes SS7 que se intercambian con otras redes [18].

La red SS7 es crítica para el procesamiento de las llamadas, por lo que los SCPs como los STPs se extienden de dos en dos, alejados físicamente para certificar que la red persista en la actividad, no obstante puede ser que alguno de estos elementos fracase. Los enlaces entre los puntos de señalización también se colocan de dos en dos, por lo que esta actividad se transporte entre todos los enlaces. Si uno de los enlace falla, éste es remitido por otro enlace. SS7 ofrece contenidos de retransmisión y corrección de faltas.

Sobre los diferentes enlaces se puede decir que estos se organizan por el prototipo, de acuerdo a su utilidad en la red de señalización SS7 [18].

Enlace A: Un enlace A enlaza un SP a un STP. Este canal permite que un SP pueda acceder a la red de señalización. Sólo se transmiten los mensajes originados por o destinados al *Signaling End Point*.

Enlace B**:** Un enlace B (*Bridge)* puente, son utilizados para conectar pares de STPs del mismo nivel jerárquico, enlaza un STP a otro STP.

Enlace C: Un enlace C (Cruzado) enlaza dos STPs del mismo par. El tráfico de señalización normal no transita a través de un canal C, salvo en periodo de congestión. Los únicos mensajes que circulan por los canales C en situación normal son los mensajes de gestión de red. Como máximo, 8 canales pueden enlazar dos STPs del mismo par.

Enlace D: Los enlace D (Diagonal) enlazan un par de STPs de un primer nivel jerárquico con un par de STPs de un segundo nivel jerárquico. Los STPs secundarios dentro de la misma red se conectan vía enlaces tipo D.

Enlace E: Los enlace E (extendido) conecta un SP con otro STP distante que no forma parte del par de STPs locales en el SP, es decir un SSP con un STP. Los enlaces E proveen una ruta alternativa para conectar con un STP.

Enlace F: Un enlace F *(Fully associated)* completamente asociado, conecta a dos *Singnaling End Points* (SSPs y SCPs). Los enlaces F no se usan regularmente en las redes con STPs, ya que se diseñaron para conectar directamente dos puntos de señalización.

2.4.3 Capas del modelo SS7

La red SS7 consiste en un grupo de elementos interconectados que intercambian mensajes para obtener funciones de señalización.

SS7 está dividido en capas y es similar con el modelo OSI. La figura 2.4.muestra cuáles son las capas que integran el modelo SS7.

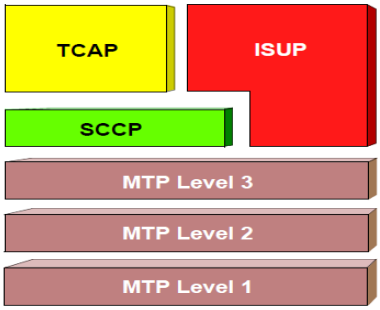


Fig. 2.4. Capas del modelo SS7 [18].

2.4.3.1 La capa física (MTP-1)

La MTP-1 determina las características eléctricas y físicas de los enlaces de la red SS7, y es el enlace de señalización de datos (SDL, *Signaling Data Link*) que consiste en un par de canales de transmisión digital que operan a 64 kbits/s y que transportan las unidades de datos SS7 entre dos puntos de señalización[18].

2.4.3.2 Parte de transferencia de mensajes del nivel 2 (MTP-2)

El MTP-2 ofrece la funcionalidad de la capa de enlace. Garantiza el procedimiento de control de línea necesario para que dos puntos extremos de un enlace de señalización puedan transmitir mensajes de señalización; a esto denomina canal de señalización (SL, *Signaling Link*). Además, realiza la comprobación de errores, control de flujo, y el control de secuencia [18].

En la figura 2.5.se muestra la transferencia de mensajes del nivel 1 y el nivel 2.

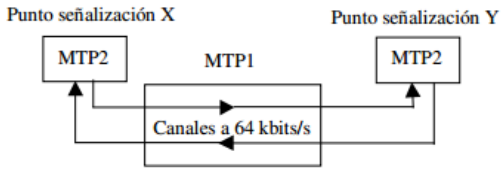


Fig. 2.5. MTP nivel 1 [18]

2.4.3.3 Parte de transferencia de mensajes del nivel 3 (MTP-3)

El MTP-3 Incorpora funciones tales como direcciones de nodo, rutas alternativas, y el control de la congestión; es la interfaz entre el MTP y los usuarios MTP (Protocolos de nivel 4) en un punto de señalización. Al mismo tiempo, MTP3 integra procedimientos para re-enrutar los mensajes cuando se produce un error en la red de señalización.

A nivel de un punto de señalización, están presentes una entidad MTP1 y una entidad MTP2 por canal de señalización, y una única entidad MTP3. Los

canales de señalización transportan tramas de señalización de mensaje (*MSU, Message Signal Unit*), tramas de señalización de estado del enlace de señalización (*LSSU, Link Status Signal Unit*) y tramas de señalización de relleno (*FISU, Fill-In Signal Unit)*.

Las LSSUs y las FISUs son generadas a nivel de una entidad MTP2 en uno de los extremos del canal de señalización, y terminan en una entidad MTP2 en el otro extremo del mismo canal.

Los mensajes de señalización reenviados por el nivel superior (es decir, MTP3) son transmitidos sobre el canal de señalización bajo la forma de tramas de señalización de tamaño variable. Estas tramas de señalización contienen, además de las informaciones de señalización, las informaciones de petición de transporte que aseguran el buen funcionamiento del canal de señalización, la figura 2.6. muestra la transferencia de mensajes por todos los niveles MTP[18].

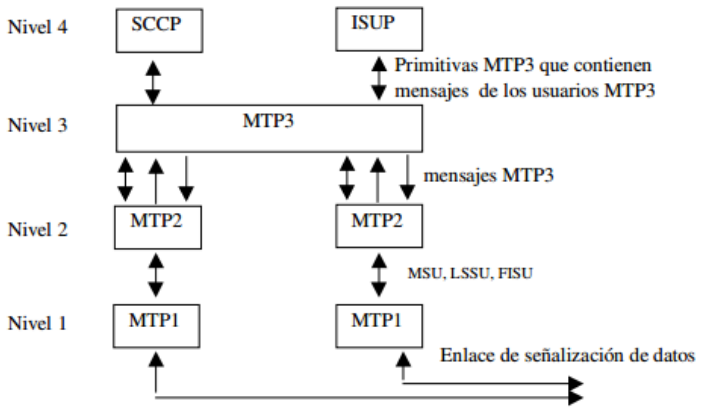


Fig. 2.6. MTP [18]

2.4.3.4 Señalización de control de conexión de pieza (SCCP)

La SCCP proporciona funciones adicionales a la MTP, para apoyar redes sin conexiones, servicios de redes orientados a conexión y a traducciones de título global (GTT). SCCP proporciona los números de subsistemas para permitir que los mensajes sean dirigidos a aplicaciones específicas o subsistemas en los puntos específicos de señalización. SCCP se usa como la capa de transporte para los servicios basados TCAP.

GTT: Añade la posibilidad de realizar, incrementar enrutamientos y liberar el punto de señalización de origen; además de conocer cada posible destino. Un título mundial es una dirección (un número 800, el número de la tarjeta telefónica, o número de identificación de abonado móvil), que es traducido por SCCP en un código de punto de destino (número de subsistema). Un número de subsistema únicamente identifica una solicitud en la señalización de destino punto[20].

2.4.3.5 La parte de usuario RDSI (ISUP)

Define el protocolo usado para configurar, administrar y liberar circuitos troncales que transportan voz y datos entre los SSP. Sin embargo, las llamadas que se originan y terminan en el mismo conmutador no utilizan señalización ISUP.

Detalla los mensajes y el protocolo empleado en la interconexión y ejecución de la llamada, ésta puede ser de voz y datos mediante la red pública conmutada (PSN). A pesar de su nombre, ISUP se utiliza para llamadas RDSI y no RDSI [20].

2.4.3.6 Capacidades de la parte de aplicación de transacciones (TCAP)

Las TCAP permite el intercambio, a través de SS7, de información entre aplicaciones, no relacionada con los circuitos, usando el servicio no orientado a conexiones SCCP y se Utiliza para los servicios de base de datos, tales como tarjetas telefónicas, 800, y la AIN, así como de conmutador a conmutador de servicios, incluyendo la marcación de repetición y devolución de llamadas.

Solicitudes y respuestas enviadas entre SSPs y SCPs se transportan en mensajes TCAP. Por ejemplo, un SSP envía una solicitud TCAP para determinar la ruta asociada a un número 800/888 marcado, y para verificar el PIN (*personal identification number*) del usuario de una tarjeta de llamada (*calling card*). En redes móviles (IS-41 y GSM), TCAP transporta mensajes MAP (*Mobile Application Part*) enviados entre las centrales celulares y las Bases de Datos que soportan la autenticación de usuario, la identificación del equipo y el *roaming*[20].

2.5 Apoyo SS7 en Asterisk

2.5.1 LIBISUP

Libisup reemplaza a la libpri en la arquitectura de Asterisk y admite usar características estándares y de aplicaciones. Se puede instalar como una interfaz a la RTB con señalización SS7. Es la primera solución SS7 desarrollada para usarlo con Asterisk [19].

2.5.2 SS7box

Sangoma rival de Digium presenta una solución llamada ss7box la cual introdujo junto con ss7boost.

Se usa para la interconexión de redes PSTN e IP en todas las redes y emplea tarjetas de hardware de Sangoma.

Las diversas funciones de SS7box y SS7boost pueden definirse como *router* y g*ateway* de señalización para aplicaciones ITU y ANSI, permite funciones de red inteligente a través de una red IP[19].

2.5.3 Chan_ss7

Es el software controlador del canal en SS7 (*chan_SS7*), es código abierto creado por Sifira A/S. La licencia es GPL (Licencia General Pública).

Inserta una implementación de la capa de MTP2, y un gran subconjunto de funciones de ISUP. Por su reparto de carga y conmutación puede ser útil desde el punto de vista de gestión de red [19].

2.5.4 Libss7

Los desarrolladores de Asterisk crearon la aplicación Libss7 como apoyo para Asterisk, usando el Zap Chan como controlador de canal y un hardware zaptel como una interfaz con la red SS7.

Se seleccionó a *chan_ss7* entre todas las aplicaciones disponibles, ya que es la solución más utilizada, para así trabajar de una manera eficiente con cada archivo necesario para la implementación del proyecto [19].

2.6 Session Initiation Protocol (SIP)

Existen muchas aplicaciones en Internet que necesitan la creación de sesiones, un intercambio de datos entre diferentes entidades. La implementación de estos intercambios se vuelve difícil por el comportamiento de las entidades que participan en ella: los usuarios cambian de localización, pueden tener varias maneras de identificarse e intercambiar datos muchas veces de manera simultánea.

El protocolo SIP trabaja en cooperación con protocolos que permiten trabajar con sesiones multimedia en tiempo real, y permite que los diferentes agentes de usuarios (aplicaciones que usan para comunicarse) puedan encontrarse y ponerse de acuerdo con el tipo de sesión que quieran compartir. Para localizar a los participantes de una sesión, SIP permite la creación de una infraestructura de hosts, a los que las aplicaciones de usuario pueden enviar peticiones de registro, invitación a las sesiones y otras solicitudes[11].

2.6.1 Funcionalidades

SIP es un protocolo de nivel de aplicación que puede establecer, modificar y finalizar sesiones multimedia; en este caso llamadas telefónicas vía internet. SIP también permite invitar participantes a sesiones existentes como lo son las conferencias *multicast*.

SIP considera cinco aspectos para el establecimiento y finalización de comunicaciones.

- Localización del usuario: hace referencia al sistema final que se utilizara para la comunicación.
- Disponibilidad del usuario: indica si el usuario quiere participar en las comunicaciones.
- Capacidad de los usuarios: Indica el medio y los parámetros del medio que se utilizaran.

- Inicio de la sesión: llamada, establecimiento de la sesión y parámetros en los dos extremos de la comunicación.
- Gestión de la sesión: Incluye la transferencia y finalización de sesiones, modificando los parámetros de la sesión y llamando a los servicios.

SIP no ofrece servicios, sino primitivas que se pueden utilizar para ofrecer diferentes tipos de servicio; tampoco ofrece servicios de control de las conferencias, ni define como se debe gestionar una conferencia. Sin embargo se puede utilizar para iniciar una conferencia que usa otro protocolo para realizar el control de la conferencia.

El protocolo SIP define principalmente seis tipos de solicitudes[12]:

- *Invite*: establece una sesión.
- *Ack*: confirma una solicitud *INVITE*.
- *Bye*: finaliza una sesión.
- *Cancel*: cancela el establecimiento de una sesión.
- *Register*: comunica la localización de usuario (nombre de equipo, IP).
- *Options*: comunica la información acerca de las capacidades de envío y recepción de teléfonos SIP.

Y seis clases de respuestas:

1xx: respuestas informativas, tal como 180, la cual significa teléfono sonando.

2xx: respuestas de éxito.

3xx: respuestas de redirección.

4xx: errores de solicitud.

5xx: errores de servidor.

6xx: errores globales.

2.6.2 Mensajes

En la figura 2.7 se analiza detalladamente una llamada SIP en la que existen varias transacciones, una transacción SIP se realiza con un intercambio de mensajes entre un cliente y un servidor [13].

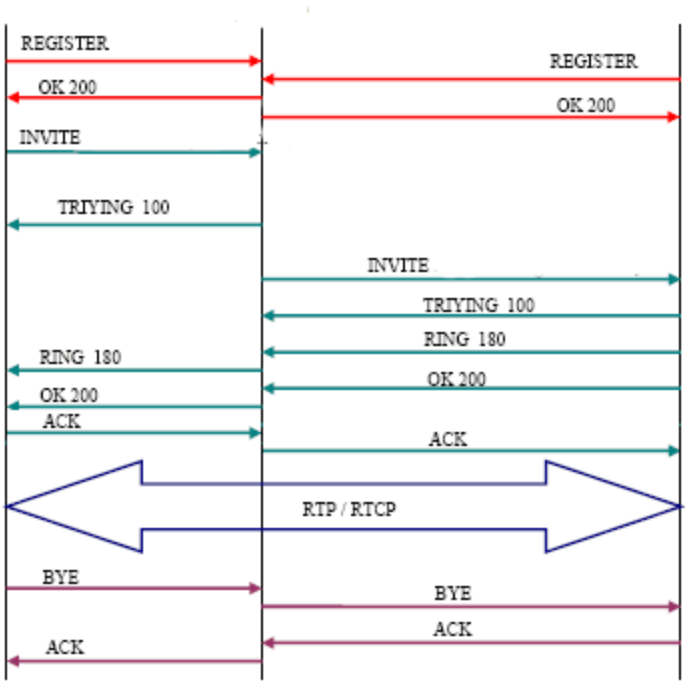


Fig. 2.7 registros en llamada SIP [13]

- Para el registro de usuarios se realzan las dos primeras transacciones, los cuales deben estar registrados para poder ser encontrados por otros usuarios. Los terminales envían una petición *REGISTER*, donde los campos *from* y *to* son el usuario registrado. El servidor Proxy, que actúa como *Register*, éste realiza la autenticación y en caso positivo envía un mensaje de *OK* [13]
- La transacción siguiente es para un establecimiento de sesión. Esta sesión consiste en una petición *INVITE* del usuario al proxy. Inmediatamente, el proxy envía un *TRYING100* 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) [13].

- 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[13].

- 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[13].

SIP hace posible esta comunicación gracias a dos protocolos que son RTP/RTCP y SDP.

El protocolo RTP se usa para transportar los datos de voz en tiempo real (igual que para el protocolo H.323), mientras que el protocolo SDP se usa para la negociación de las capacidades de los participantes, tipo de codificación, etc.

Los usuarios SIP son identificados usando una estructura tipo e-mail llamada SIP URl como por ejemplo: sip:1234@sipnetwork.com.

2.7 Protocolo IAX2

IAX, versión 2, es un protocolo diseñado específicamente para VoIP por la misma persona que desarrolló asterisk, que intenta evitar los problemas que existen con el protocolo SIP.

Este producto intenta, por ejemplo, evitar los problemas que existen cuando uno de los teléfonos IP está detrás de un *router NAT* o intenta reducir el *overhead* producido por el protocolo RTP.

La característica diferenciadora de este producto frente a SIP es que la señalización de la llamada y la información de audio, viajan siempre por el mismo puerto UDP. Al contrario de lo que sucede con SIP, donde los dos teléfonos negocian una conexión RTP/RTCP distinta de la que se usa para la señalización (lo cual puede dar problemas al atravesar un *firewall* y un *router* NAT). IAX utiliza siempre el mismo puerto tanto para la señalización como para el flujo de datos de audio, por lo que únicamente es necesario abrir una puerta en el *firewall* o el *router NAT*.

Adicionalmente, si se trata de una conexión entre dos centrales IP, el protocolo IAX permite meter varias conversaciones sobre una misma conexión, teniendo como único límite el ancho de banda disponible.

Finalmente, es necesario destacar que aunque IAX utilice siempre el mismo puerto, eso no significa que todo el flujo de datos, por ejemplo de audio, tenga que pasar siempre por la central IP. El propio protocolo tiene previsto que si un determinado nodo detecta que él mismo no es necesario para mantener la comunicación, puede indicar a los otros dos nodos que se debe renegociar la conexión, sin perderla, para que el flujo de datos lo intercambien los dos teléfonos, sin tener que pasar por el servidor[14].

Para entender el funcionamiento del protocolo IAX es necesario mostrar la figura 2.7 que presenta las tres fases en las que se divide una llamada en IAX, y los mensajes que se generan entre los dos terminales.

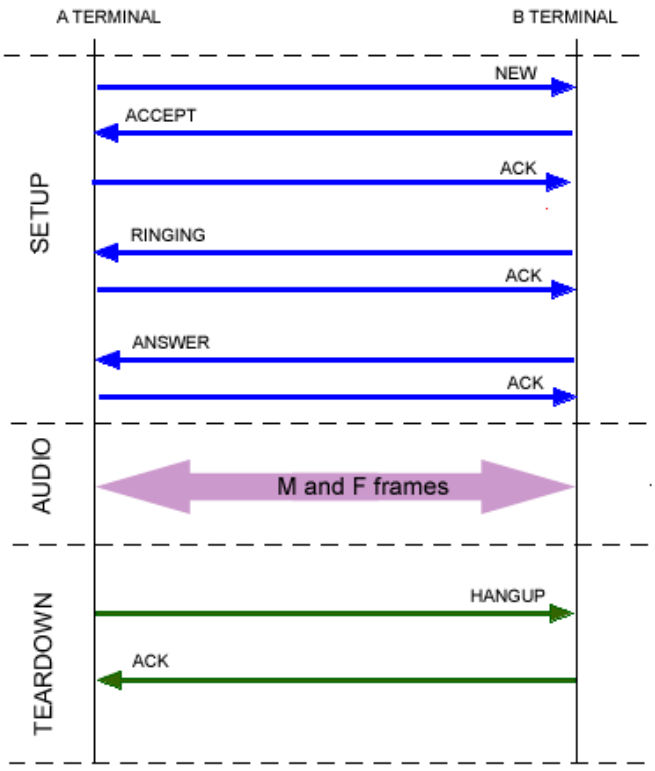


Fig. 2.7 registros en llamada IAX [15]

Una llamada IAX o IAX2 tiene las siguientes transacciones:

- *Call Setup*

Una terminal (A) comienza una conexión y envía un nuevo (*new*) mensaje. El terminal al que se llama (B) responde con un mensaje de aceptación (*accept*) y la terminal que llama responde con un mensaje de reconocimiento (*ACK*).

Luego, la terminal B da la señal de timbrado y envía un *ACK* para confirmar la recepción del mensaje. Finalmente la terminal B acepta la llamada con un mensaje de respuesta que es reconocido por la terminal A; con esto la configuración de llamada se encuentra establecida [15].

- *Flujo multimedia*

Las tramas M y F son enviadas en cualquier dirección con datos de audio, cada flujo está compuesto principalmente por mini tramas IAX que contienen un *header* de 4 bytes que indica la eficiencia del ancho de banda; este flujo es complementado por tramas llenas periódicamente.

- *Tramas*

Incluyen información de sincronización, y es importante notar que los mensajes de audio usan el protocolo UDP que se usa en los mensajes de señalización evitando problemas de *NAT.*

- *Desmontaje de llamada*

El desmontaje de conexión solo envía un mensaje de *Hangup* y *ACK* del mensaje.

CAPÍTULO 3

IMPLEMENTACIÓN DEL PROYECTO

3.1 Asterisk

Asterisk es una aplicación que permite el control y gestión de comunicaciones, ya sean analógicas, digitales o VoIP, mediante los diversos protocolos que se necesiten para su implementación. Su implementación bajo *OpenSource*presenta muchas ventajas para los desarrolladores dándoles la posibilidad de crear sistemas de comunicaciones de excelente calidad, seguridad y versatilidad.

Con el pasar del tiempo esta aplicación se ha convertido en el software necesario para el reemplazo de las anteriores aplicaciones analógicas o digitales, permitiendo la integración con la tecnología actual que es VoIP; con esto se convierte en uno de los sistemas más avanzados y económicos a la hora de brindar servicios de comunicación. Una de sus principales ventajas

es la facilidad de programación que presenta, no se necesita tener conocimientos avanzados en ningún lenguaje de programación; basta con leer los comandos básicos, y cómo interpreta asterisk cada uno de sus archivos, para poder ejecutar un sistema.

Otra de sus ventajas implica en la posibilidad de integrarlo como sistema híbrido, ya que permite gestionar comunicaciones telefónicas tradicionales (analógicas, digitales, móviles) y comunicaciones IP mediante el uso de los protocolos estándar de VoIP.[23]

Existen diferentes versiones de asterisk en la actualidad, la versión a utilizar en este proyecto será Asterisk 1.8 que presenta una buena funcionalidad y compatibilidad con los *drivers* necesarios para la identificación de las tarjetas E1/T1 a utilizar.

3.1.1Requerimiento para iniciar el proyecto en Asterisk

Para la implementación de este proyecto, a parte del hardware y software, se necesita:

- Conocimientos básicos de lenguajes de programación, para poder interpretar como lee asterisk los archivos de configuración (.conf).
- Conocimientos básicos de Networking.

- Al principio puede resultar frustrante comprender la programación en asterisk.

3.1.1.1 Hardware

En la parte de hardware tenemos los siguientes recursos.

- 2 Computadores con procesador Pentium IV en adelante.

  Se recomienda utilizar estos computadores para que no exista alguna sobrecarga en el servidor, que impida una buena comunicación entre los usuarios.
- Memoria RAM superior a 512 Mb
- Disco duros SATA 20 GB
- 2 tarjetas con soporte E1/T1

  Para la implementación de SS7, solo se necesita un *spam* (conector o puerto) para la interconexión entre ambos servidores mediante SS7.
- Teléfonos IPs

  Telefonos que soporten los protocolos VoIP, la mayoría de estos soportan el protocolo SIP.
- *Switch* o *Router* de preferencia inalámbrico, requeridos para la conexión entre ambos servidores.
- 1 Monitor y teclado

Si no se puede establecer una conexión remota, los dispositivos antes mencionados son necesarios para acceder a los servidores directamente.

- Cable RJ-45

  Indispensable para la conexión física entre ambos servidores.

- Cable cruzado para E1/T1.

  Se necesita para la conexión entre los dos servidores mediante SS7, en la siguiente imagen (ver figura 3.1) se presenta su configuración.

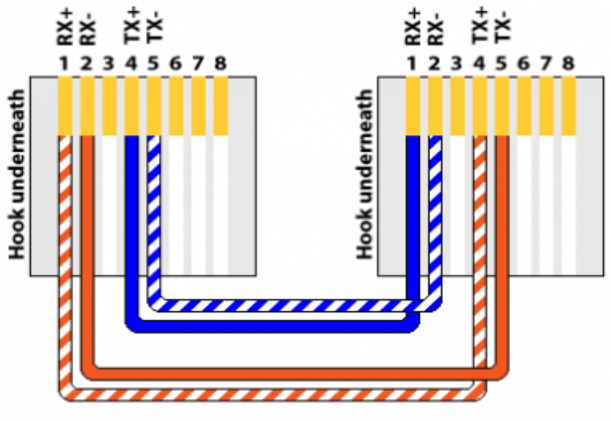


Fig. 3.1 Cable cruzado para E1/T[22]

3.1.1.2 Software

- Linux S.O.

  Se recomienda usar CentOS, o Ubuntu Server; por lo general asterisk es instalado en CentOS pero en nuestra caso usaremos Ubuntu para demostrar que también se puede trabajar en dicho sistema operativo sin complicaciones.

- Softphone.

  Utilizaremos las versiones gratuitas de Zoiper y X-Lite, dos clases de softphone muy populares en la red.

  Para descargar Zoiper: http://www.zoiper.com/download_list.php.Para descargar X-Lite:http://www.counterpath.com/xlite-comparison.html

- Wireshark

  Analizador y capturador de paquetes, lo utilizaremos en algunas pruebas que realizaremos en capítulos más adelante. Puede ser descargado desde:

  http://www.wireshark.org/download.html.

- Putty

  Cliente SSH para establecer conexión remota hacia los servidores.

  Descarga:

  http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

- VmWare Player

  Nos ayudara con la virtualización de sistemas operativos en los cuales instalaremos los softphone a utilizar para el registro de cuentas en los servidores.

Descarga: http://www.vmware.com/products/player/

3.1.1.3 Virtualización de host y teléfonos IP

En la etapa de pruebas necesitaremos de 10 dispositivos que soporten VoIP, por lo que usaremos 10 softphones que se encuentran en 10 sistemas operativos emulados en máquinas virtuales a través del software *VmWare* (*Virtual Machine*); al hacer esto podemos registrar múltiples cuentas en cada uno de nuestros servidores sin necesidad de tener físicamente 10 teléfonos IP.

En la figura 3.2 se muestra la representación del diagrama de conexión entre nuestros servidores y las extensiones a utilizar.

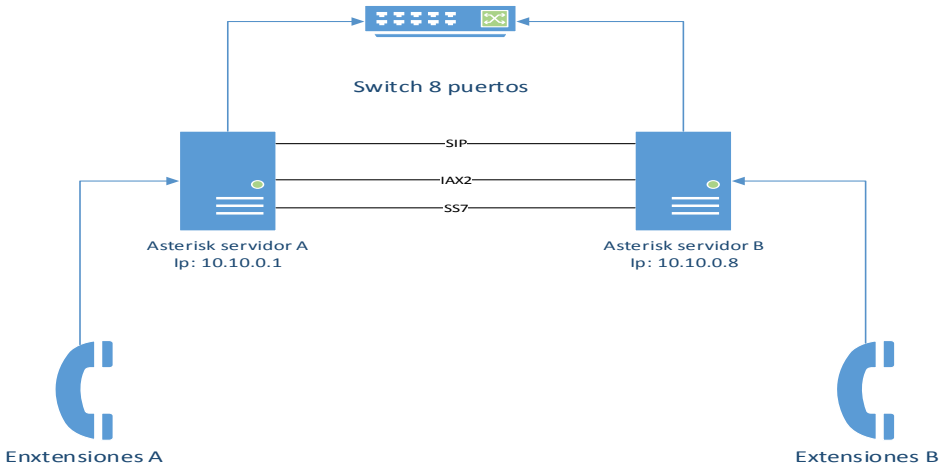


Fig. 3.2. Diagrama esquemático de conexión

A continuación se presenta el detalle de la configuración de extensiones para SIP (Figura 3.3) e IAX (Figura 3.4) en cada servidor; más adelante se indicará para qué sirven estas extensiones y dónde son configuradas.

Grupo de extensiones Servidor A (SIP)		Grupo de extensiones Servidor B (SIP)	
Ext.	**Cuenta**	**Ext.**	**Cuenta**
1001	TelefonoA1	3001	TelefonoB1
1002	TelefonoA2	3002	TelefonoB2
1003	TelefonoA3	3003	TelefonoB3
1004	TelefonoA4	3004	TelefonoB4
1005	TelefonoA5	3005	TelefonoB5
1006	TelefonoA6	3006	TelefonoB6

Fig. 3.3. Grupo de extensiones (SIP) en cada servidor.

Grupo de extensiones Servidor A (IAX2)		Grupo de extensiones Servidor B (IAX2)	
Ext.	**Cuenta**	**Ext.**	**Cuenta**
8001	TiaxA1	9001	TiaxB1
8002	TiaxA2	9002	TiaxB2
8003	TiaxA3	9003	TiaxB3
8004	TiaxA4	9004	TiaxB4
8005	TiaxA5	9005	TiaxB5
8006	TiaxA6	9006	TiaxB6
8007	TiaxA7	9007	TiaxB7
8008	TiaxA8	9008	TiaxB8
8009	TiaxA9	9009	TiaxB9
8010	TiaxA10	9010	TiaxB10

Fig. 3.4. Grupo de extensiones (IAX2) en cada servidor.

3.2 Generalidades Dahdi

Las siglas DAHDI hacen referencia a *Digium/Asterisk Hardware Device Interface*, es decir, una interfaz para toda la lista de productos Digium (y

compatibles) que conecta con el sistema Asterisk, considerando que hablamos de productos que conectan concretamente con la PSTN (*Public Switched Telephone Network*, o Red de Telefonía Conmutada) y la telefonía clásica.

El driver DAHDI se instala en ambas interfaces analógicos y digitales. Se puede acceder a las tarjetas de comunicaciones por un interfaz de *kernel*. En *etc/asterisk/chan_dahdi.conf* se recopila la configuración de los interfaces de *hardware*. *dahdi-channels.conf (etc/astersik/dahdi-channels.conf)* que es donde está la configuración asterisk para la utilización de dichas interfaces de hardware.

3.3 Configuración de Archivos del proyecto Asterisk

En la implementación de este proyecto editaremos ciertos archivos de configuración, que permitirán la conexión entre ambos servidores utilizando SIP, IAX y SS7 respectivamente.

Este capítulo describe los parámetros básicos necesarios para la configuración de cada archivo; la configuración total, es decir, el archivo completo de cada fichero utilizado está descrito en la parte de Anexos.

La ubicación de estos archivos se encuentra en la ruta */etc/asterisk/*

3.3.1 SIP.CONF

En este archivo se configuran las cuentas que utilizan el protocolo SIP para la comunicación entre usuarios. A continuación se muestran los parámetros básicos de configuración para este fichero, teniendo en cuenta los comentarios que están precedidos por '; (punto y coma) '.

En la consola del servidor digitamos:

[root@asterisk -]# vim /etc/asterisk/sip.conf

Pulsamos la tecla 'I' para proceder a editar el archivo.

Parámetros generales

[general]

Siempre va la línea [general] al inicio del archivo *sip.conf.*

context;

Contexto o grupo que se usa en el plan de marcado del archivo*extensions.conf*

allowguest

Deshabilita llamadas sin autenticación, puede ser *yes* o *no*.

Srvlookup

Hace posible encontrar los registros DNS SRV para llamadas salientes SIP de acuerdo a los nombres de dominio. Puede ser *yes* o *no*.

Udpbindaddr

Indica la dirección IP de la interfaz de red por la cual asterisk va a escuchar las solicitudes UDP. Por lo general toma la dirección IP 0.0.0.0 indicando a asterisk que escuche las peticiones UDP por cualquier interfaz de red conectada al servidor.

Transport

Protocolo de transporte a utilizar. Puede ser UDP o TCP.

Qualify

Indicaal servidor el estado del dispositivo, conectado o desconectado. Sus valores pueden ser ‘xxx’ que indica el tiempo en milisegundos que toma para indicar que si está conectado o no, *yes* para mostrar el estado de conexión, *no* para no mostrar el estado de conexión.

Creación de cuentas

Los parámetros necesarios para la creación de cuentas son:

[nombre_de_cuenta]

Indica el nombre de la cuenta que se debe utilizar en el teléfono IP o softphone a registrar.

Type

Representa el tipo de extensión. Sus valores pueden ser: F*riend* para hacer y recibir llamadas; *user* solo recibir llamadas y *peer* solo para realizar llamadas.

Secret

Clave de autentificación.

Context

Contexto o grupo de extensiones en donde será usada dicha cuenta. Ver archivo *extensions.conf.*

Language

Define las señales para un país, debe estar presente en el archivo *indications.conf.*

Disallow

Deshabilita la entrega o recepción de todos los audios de *codecs* para esta cuenta, esto se lo hace para que no elija cualquier códec aleatoriamente.

Allow

Especifica el *codec* de audio que puede entregar o recibir esta cuenta.

Nat

Asume que el dispositivo está en una red privada, y debe utilizar *nat* para poder salir a una red externa. Puedes ser *yes* o *no*.

Registro de asterisk como cliente SIP

Las siguientes líneas se utilizan para registrar asterisk como cliente SIP en el servidor B con el objetivo de realizar llamadas entre ambos servidores utilizando el protocolo SIP. En el archivo *sip.conf* del servidor B encontraremos una línea parecida a esta.

Register=>cuentaSB:passwordSB@IPSB:puerto/cuentaSA.

CuentaSB= Es la cuenta que usa el servidor B para conectarse al servidor A.

passwordSB= Es la contraseña que usa la cuentaSB.

IPSB= Dirección IP que pertenece al servidor B

Puerto= puerto de escucha del protocolo a utilizar.

cuentaSA= Es la cuenta que usa el servidor A para conectarse en el servidor B por medio de SIP.

Se procede a crear la cuenta con la cual se puede registrar el servidor A en el servidor B.

La configuración completa de este archivo se encuentra en la parte de anexos.

3.3.2 Configuración IAX.CONF

Al igual que en el archivo *sip.conf*, en este fichero se configuran los parámetros para la creación de cuentas que puedan registrarse por medio del protocolo IAX, así mismo para registrar los servidores asterisk por medio de este protocolo.

Cabe recordar que IAX o su versión actual IAX2 es un protocolo creado para asterisk y aún no se encuentra estandarizado, por lo que aún no se encuentran muchos dispositivos compatibles con IAX en el mercado a un costo accesible.

Parámetros de configuración

Este archivo usa parámetros similares a los descritos en el archivo *sip.conf*, por lo que sólo se pondrá énfasis en los parámetros que no se describieron en la sección anterior.

[general]

bindport

El puerto UDP usado por este protocolo, por lo general es 4569.

bindaddr

IP que asterisk usará para "escuchar" los pedidos de conexiones. Se usa 0.0.0.0 para que asterisk escuche por todas las interfaces del computador

Delayrect

Aumenta la seguridad para "*brute force password attacks*" y difiere el envío de los rechazos de autenticación. Puede ser *yes* o *no*.

Bandwidth

Indica el tipo de *codec* a utilizar. Puede ser *low*, acepta todos los *codecs* medios excepto G.726 y ADPCM; *medium*, acepta todos los *codecs* excepto *slinear*, *ulaw* y *alaw*; *high*, acepta todos los *codecs* de audio.

Creación de cuentas

Para la creación de cuentas se pueden utilizar los mismos parámetros que el archivo *sip.conf* exceptuando por el parámetro *nat* que no es necesario en este protocolo.

Registro de asterisk como cliente IAX

En las siguientes líneas se indica como registrar asterisk por medio de IAX en otro servidor.

El proceso de registro es similar al del archivo *sip.conf* pero hay que indicar los siguientes parámetros al momento de crear la cuenta para registrar asterisk como cliente IAX.

Trunk

Se establece para utilizar el *trunking* ofrecido por IAX2, puede ser *yes* o *no*.

Deny

Denegar llamadas entrantes desde cualquier IP, va acompañado del parámetro *permit* para habilitar la única IP, por la que asterisk escucha llamadas entrantes IAX.

Permit

Dirección IP permitida para las llamadas entrantes.

Para ver la configuración completa del archivo *iax.conf* verificar en la sección de anexos.

3.3.3 Configuración EXTENSIONS.CONF

El plan de marcado de la central telefónica para cada contexto y para cada cuenta se halla en este archivo. Aquí se definen las reglas de marcación o en su defecto las opciones que deben pasar (mensajes, IVR, tono de marcado, correo de voz, llamadas, etc) si se digita tal número.

Parámetros de configuración

[general]

Static

Indica si se graba o no el plan de marcado desde la consola asterisk ejecutando dialplan sabe. Puede ser *yes* o no.

Autofallthrough

Si se observa *yes*es porque alguna llamada por algún índice si la salida del plan y se terminará.

Clearglobalvars

Si está activado se liberan las variables globales cuando se recargan las extensiones o se reinicia asterisk.

Priorityjump

Si tiene valor *yes*, la aplicación soporta *jumping* o salto a diferentes prioridades.

Creación de contextos

En esta parte se crea el plan de marcado, se definen las reglas a seguir para cada cuenta y los mensajes que puedan o no presentarse como por ejemplo un IVR.

[contexto]

Se define el nombre del contexto a utilizar, aquí deber indicarse el nombre del contexto que se ha definido en los archivos *sip.conf* y *iax.conf* para cada una de las cuentas.

Sentencia para crear extensión

exten => numero_a_marcar,prioridad,aplicación.

La línea anterior representa una línea de comando básica para la creación de una extensión donde se indica el número a marcar, la prioridad que recibe la extensión y la aplicación que debe ejecutar asterisk.

En la sección de anexos se puede encontrar el archivo completo, el cual incluye comentarios en la mayor parte de sus líneas para facilitar la interpretación del lector.

3.3.4 ConfiguraciónSS7.CONF

En este archivo se configuran los parámetros de conexión SS7 para cada servidor, a continuación se muestra el detalle de los mismos.

Este archivo no viene incluido al instalar asterisk, por lo que no lo encontraremos sin antes haber instalado *chan_ss7*. Al descomprimir

chan_ss7 dentro del directorio viene el archivo ss7.conf el cual debemos copiar a la ruta */etc/asterisk*.

Los parámetros que nos interesa modificar se encuentran resaltados de color azul. (ver en anexos, pág.178 servidor A , pag.198 servidor B)

3.3.5 Configuración SYSTEM.CONF

Este archivo sirve para la configuración de señalización en las tarjetas E1/T1 que vamos a utilizar para la conexión de los dos servidores por medio de ss7. (Ver en anexos , pág 182)

Para el servidor B, realizar el mismo procedimiento recordando utilizar el nombre de la tarjeta correspondiente y cambiar el *clock source* en el archivo *system.conf*.

Después de haber realizado la configuración ejecutamos un *reboot* para que los cambios surjan efecto.

3.3.6 Configuración dahdi_channels.conf

Este archivo debe ser modificado ya que si no lo hacemos puede generar problemas en la señalización en el *span* o conector a utilizar para la conexión por medio de ss7.

Debemos comentar todas las líneas de código pertenecientes al *span* que usemos para la señalización; en este ejemplo usamos el *span/conector1* de la tarjeta por lo cual debe quedar comentado de la siguiente manera.

; Span 1: TE4/0/1 "T4XXP (PCI) Card 0 Span 1" (MASTER) HDB3/CCS/CRC4 ClockSource.

;group=0,11

;context=from-pstn

;switchtype = euroisdn

;signalling = pri_cpe

;channel => 1-15,17-31

;context = default

;group = 63

Observamos que cada línea inicia con un ‘punto y coma’, esto significa que están comentadas y no serán tomadas en cuenta por asterisk cuando lea este archivo.

La configuración completa de este archivo se puede encontrar en la parte de anexos.

CAPÍTULO4

4.1 Configuración del Equipo de Interconexión

El equipo de interconexión puede ser un *switch* o un *router*, dependiendo de los requerimientos para la implementación, a veces resulta útil tener un equipo inalámbrico, ya que en la actualidad existen diversos softwares y aplicaciones que pueden ejecutarse en dispositivos inteligentes o *tablets*.En caso de utilizar un *switch* no podremos conectarnos vía inalámbrica a nuestros servidores y por lo tanto no tendremos acceso a llamadas IP.

Debido a que los protocolos SIP e IAX viajan por medio de IP, utilizaremos un *router* inalámbrico por las facilidades ya descritas, sin embargo para la conexión por medio de SS7 se deben usar tarjetas con soporte E1/T1 ya que

no es posible utilizar una interfaz de red común, debido a su incompatibilidad con SS7.

Además, se debe usar un cable cruzado T1/E1 para interconectar ambas tarjetas; la configuración de este cable se proporciona en la parte de anexos.

En la figura 4.1 mostramos la red con las IP asignadas a cada uno de los equipos.

Dirección de red:10.10.0.0

Puede ser cualquier dirección IP.Se recomienda una IP privada ya que en este proyecto no es necesario una IPpública para su implementación.

Máscara de subred: 255.255.255.0

La cantidad de host puede ser cualquiera, en este caso usamos mascara /24 con una capacidad de 254 *host*.

Dirección IP de *router*: 10.10.0.254

Por lo general se asigna la primera o la última dirección IP de host para el equipo de interconexión.

Configuración de *router*: Modo *Bridge*

Se configura en este modo para que no entregue direcciones IP automáticamente (DHCP).

Dirección IP Servidor A: 10.10.0.1

Dirección IP Servidor B: 10.10.0.8

Se puede elegir cualquier dirección IP de *host* disponible para asignarle a cada uno de los servidores y a las cuentas SIP o IAX respectivas.

Grupo de extensiones Servidor A: 10.10.0.10 – 10.10.0.19

Grupo de extensiones Servidor B: 10.10.0.20 – 10.10.0.29

Recordemos que para SS7 no se utilizan las interfaces de red, por lo tanto es independiente de esta configuración.

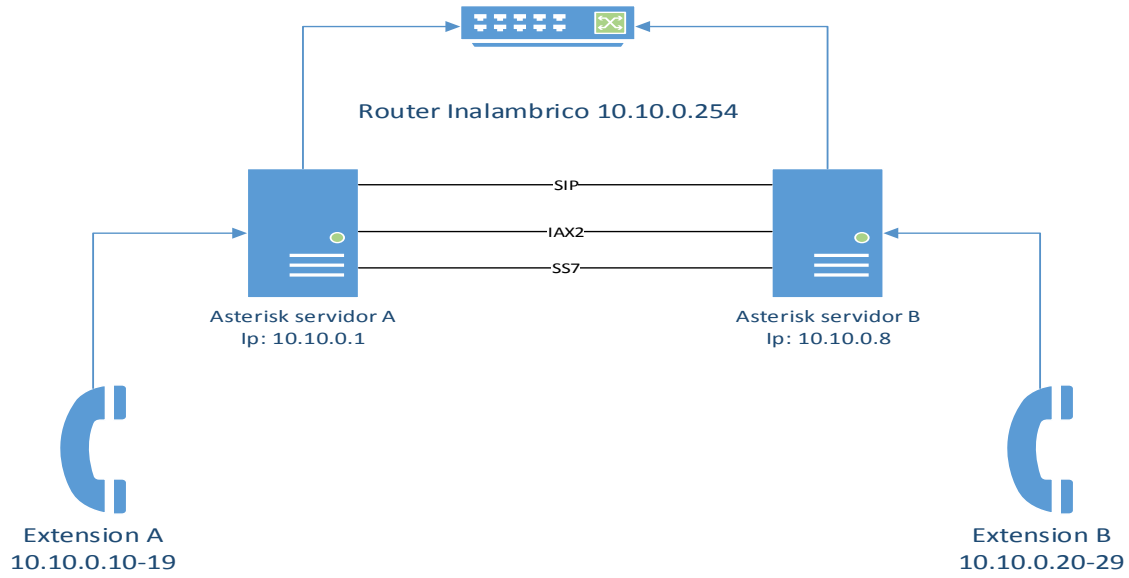


Fig. 4.1. Red Implementada

Debemos recordar que si vamos a utilizar *NAT* (*Network Address Translate*) en el archivo *sip.conf*, tenemos que agregar el parámetro *nat=yes* en cada una de las cuentas para que no existan inconvenientes al momento de que los paquetes viajen por una red externa. Sino vamos a usar este parámetro, se recomienda no ponerlo o en su defecto dejarlo comentado.

4.2 Creación del Servicio

Asterisk puede implementar múltiples servicios tales como centrales telefónicas, correo de voz, llamadas en espera, entre otras.En este proyecto abarcaremos un tipo de servicio básico y común usado en algunas instituciones.

Este servicio está basado en la implementación de un IVR, donde el usuario llama y escucha una grabación que le presente una serie de opciones para comunicarse con el departamento o persona que se indica en la grabación.

En el siguiente ejemplo se parte de la existencia de dos instituciones separadas que representarán a los servidores A y B, respectivamente (ver figura 4.2).

Como toda empresa hay ocasiones en que es necesario transferir una llamada, esto lo podemos conseguir por medio del *Button Transfer* que se encuentran en los softphone o dispositivos que soporten VoIP. Al hacer las pruebas correspondientes se realiza con éxito la transferencia de llamadas entre extensiones del mismo servidor es decir:

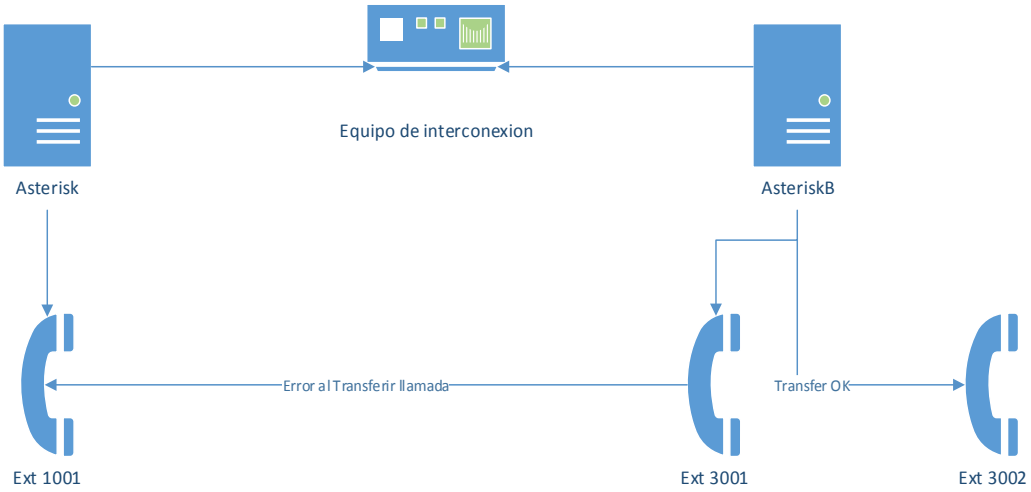


Fig. 4.2 Ejemplo de extensiones

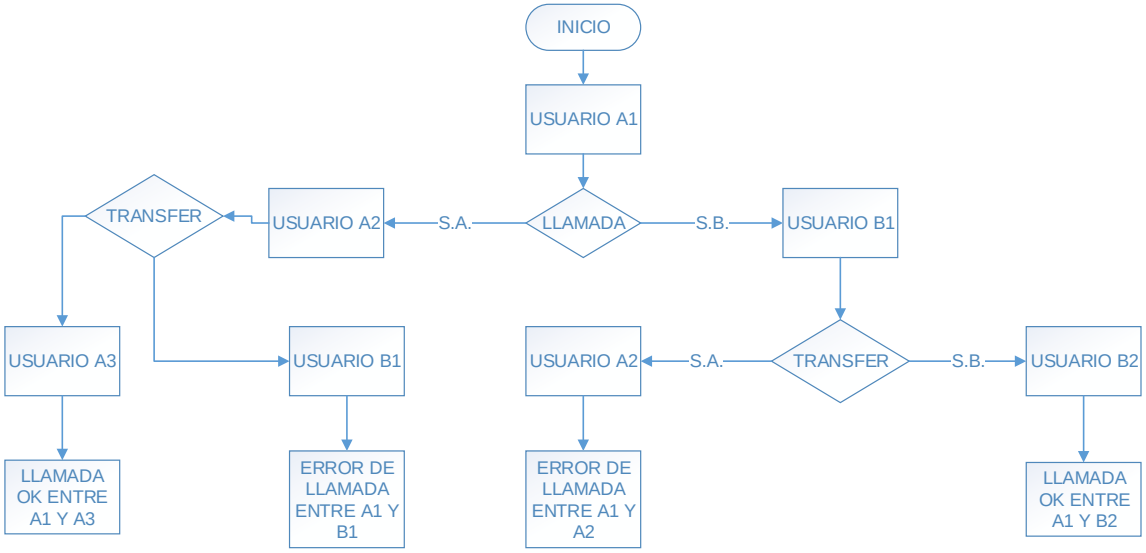


Fig. 4.3Diagrama de Flujo del servicio.

En la figura 4.3 se muestra el diagrama de flujo del servicio. Como vemos,se pueden realizar transferencias entre cuentas que pertenezcan al mismo servidor, sin embargo no se pueden realizar transferencias entre cuentas de

distinto servidor; con un poco de lógica vemos que esto no generaría inconvenientes ya que para comunicarse entre cuentas del mismo servidor basta con transferir la llamada desde las extensiones propias de este servidor y no utilizar las extensiones del servidor externo.

A su vez, se emplea una extensión de ayuda en caso de no tener conocimiento sobre la distribución de extensiones en la institución opuesta.

Para esto empleamos las aplicaciones ofrecidas por asterisk que son: *Goto(), Playback(), Background().*

La aplicación *Goto()* nos indica que al marcar una extensión, ésta nos redirigirá a otro grupo de contexto que tendrá otras opciones, a continuación un ejemplo:

exten => 999,1,Goto(Menu,nuevaext,1)

[Menu]

exten => nuevaext,1,Answer()

En este ejemplo al discar 999, nos lleva a un nuevo contexto (Menú en este caso) y a continuación se introduce el nombre de la extensión que puede ser alfanumérica para que ejecute la aplicación *Answer()*

Playback().- Esta aplicación se usa para crear un *Dialplan* en donde se escuche una grabación que proporcione un mensaje auditivo al usuario que esté realizando la llamada.

Background().- Esta aplicación se usa para crear menú de voces, esta es la que escuchemos generalmente cuando nos indican “presione X número para comunicarse con X departamento”, al momento de presionar el número indicado en la grabación, de inmediato procede a comunicarnos con la cuenta correspondiente.Siguiendo el ejemplo anterior tenemos:

[Menu]

exten => nuevaext,1,Answer()

same => n,Background(soporte) ; soporte es el mensaje de audio utilizado.

Same => n,WaitExten(5) ; espera 5 segundos para que el usuario digite una opción.

Exten => 1,1,Goto(contexto,extensión1,1)

Exten => 2,1,Goto(contexto,extension2,1)

Exten => 3,1,Goto(contexto,extension3,1)

Tomando desde el ejemplo anterior, al marcar 999 la aplicación *Goto* nos llevará donde se encuentra el mensaje de audio con las diferentes opciones, este a su vez indica que digitó presionar para comunicarse con la cuenta respectiva y una vez marcado puede transferir nuestra llamada ya sea a la extensión1 si se marcó 1, extensión2 si se marcó 2 o extensión3 si se marcó 3.

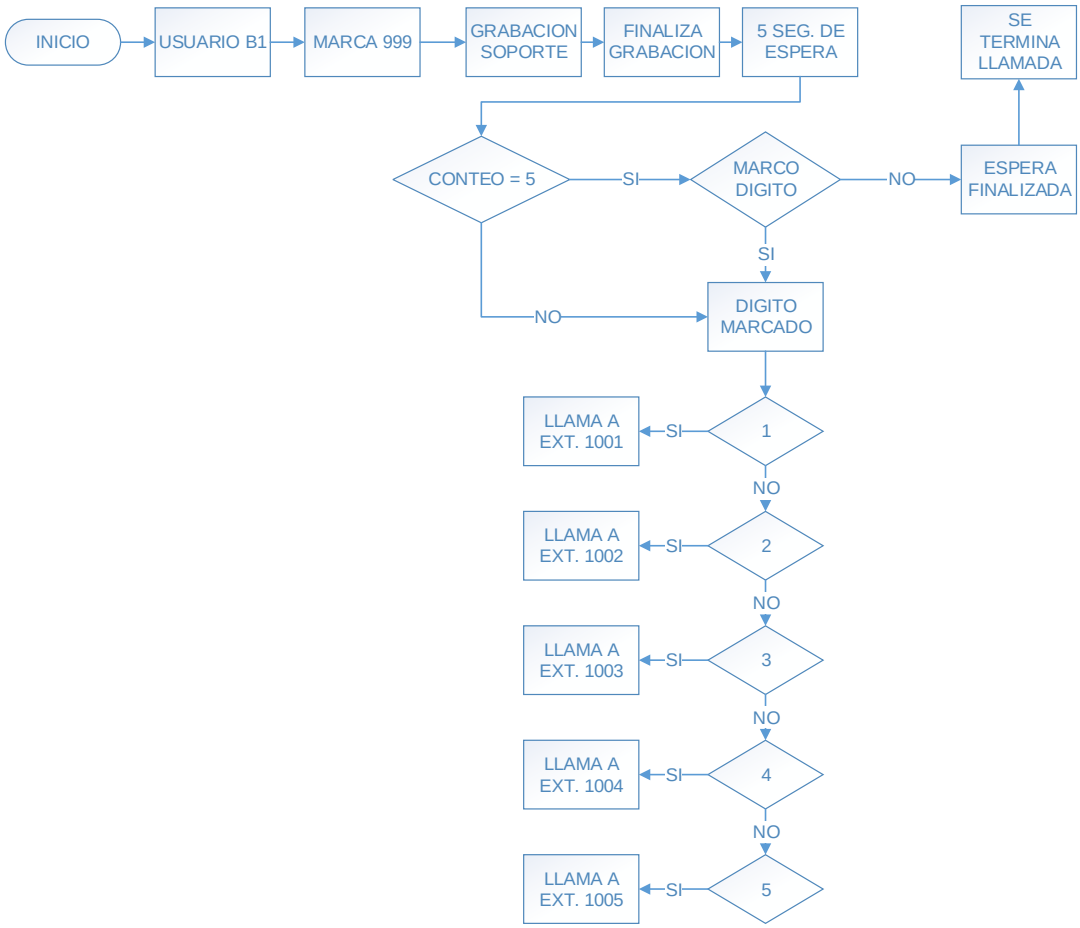


Fig. 4.4Diagrama de Flujo del plan de marcado en el servicio.

La figura 4.4 muestra el diagrama de flujo que representa los pasos a seguir cuando el usuario B1, conectado al servidor B, marca el número 999;

escucha la grabación de soporte que le indique los números a digitar para comunicarse con las extensiones respectivas.

En el capítulo de pruebas se describe como están configurados los archivos *extensions.conf* de cada servidor para proporcionar este servicio.

CAPÍTULO 5

FUNCIONAMIENTO Y PRUEBAS DEL PROYECTO

En este capítulo se detalla la parte técnica utilizada para la implementación del proyecto, las instalaciones, configuraciones, pruebas y funcionamiento se muestran a continuación.

5.1 Etapa de pruebas entre servidores (instalación)

Para poder programar en asterisk primero debemos conocer algunos comandos de Linux, que serán de mucha ayuda para la configuración.

Comandos básicos y programas de Linux utilizados en la implementación de este proyecto.

apt-get : Descarga e instalación de paquetes.

update: Toma una nueva lista de paquetes.

upgrade: Realiza un *upgrade.*

install: Instala nuevos paquetes.

remove: Remueve paquetes.

&&: Se puede utilizar para ejecutar varias líneas de comando al mismo tiempo.

subversion: Control de sistema de código abierto, se puede registrar la fuente de los archivos y documentos, se asemeja a un servidor de archivos.

ncurses: Librería que actualiza los caracteres en la pantalla con una razonable optimización.

libssl-dev: Representa a las librerías *openssl*.

libxml2-dev: Instala el lenguaje XML; este lenguaje describe cierta clase de información en algunos tipos de documentos como HTML.

vim-nox: Editor visual mejorado.

mkdir: Comando para crear directorios.

./configure: Ayuda en la compilación de un programa creando un archivo *makefile*.

make: Ejecuta las instrucciones que se encuentran en el archivo *makefile*.

make install: Encuentra el objetivo de instalación en archivo *makefile* para instalar el programa.

make config: Junto con los comandos anteriores, se utiliza para la completa instalación de un programa.

chown: Cambia los derechos de usuario en directorios.

cp: Copia archivos, (cp fuente destino).

cat: Crea archivos y permite añadir información al momento de crearlo.

mv: Renombra archivos (mv nombre_original nuevo_nombre).

rm: Remueve archivos, (rm *nombre_archivo*).

wget: Descarga de archivos, similar a *subversion*.

Instalación de UBUNTU SERVER 12.04

Vamos a realizar una instalación básica de este sistema operativo, debido a que nuestra prioridad es la configuración de asterisk; hemos decidido trabajar sobre Ubuntu para comprobar el desempeño de asterisk en este sistema operativo ya que en la mayoría de los casos se instala sobre CentOS.

La imagen que contiene los datos de este sistema operativo la podemos obtener de http://www.ubuntu.com/download/server

Una vez descargado el archivo procedemos a quemar la imagen en un CD/DVD o a su vez crear un USB *booteable,* en caso de no poseer una unidad lectora de discos.

A continuación se muestra gráficamente los pasos para una instalación básica.

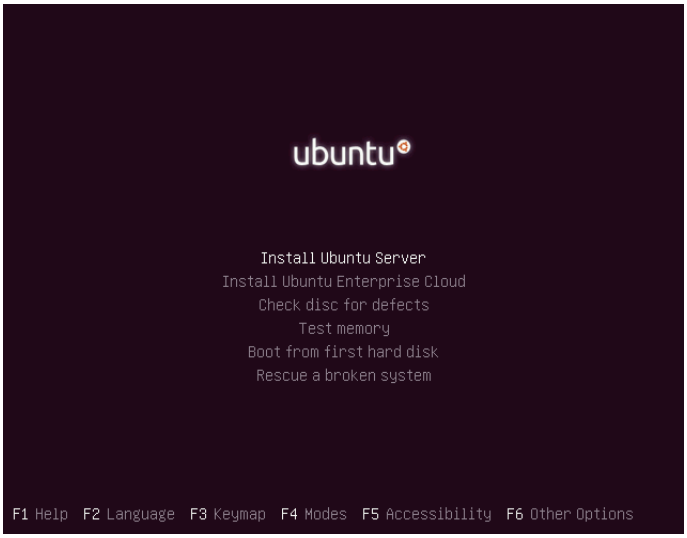


Fig. 5.1 Instalación de UBUNTU

Pantalla inicial de Ubuntu server, seleccionamos *Install Ubuntu Server* (ver figura 5.1)

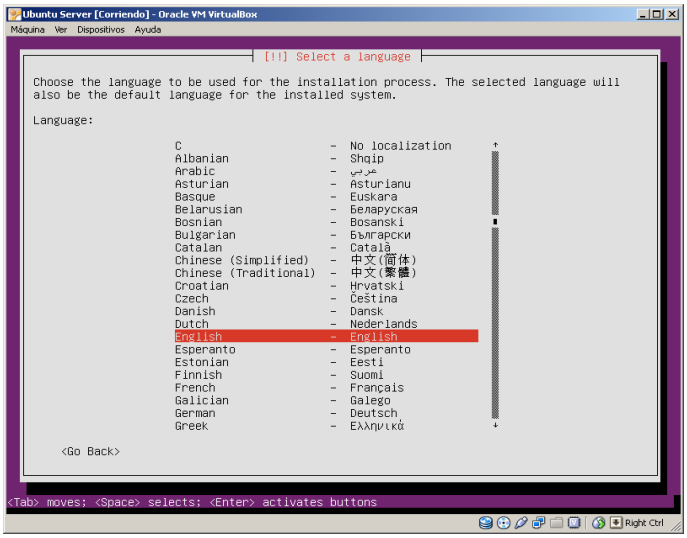


Fig. 5.2. Selección de lenguaje y Teclado.

Se despliega un menú, el cual nos permite escoger el lenguaje que deseamos utilizar, así como el tipo de teclado que tengamos; como se muestra en la figura 5.2.Luego de esto seleccionamos *continue* (ver figura 5.3) para aceptar el lenguaje antes determinado.

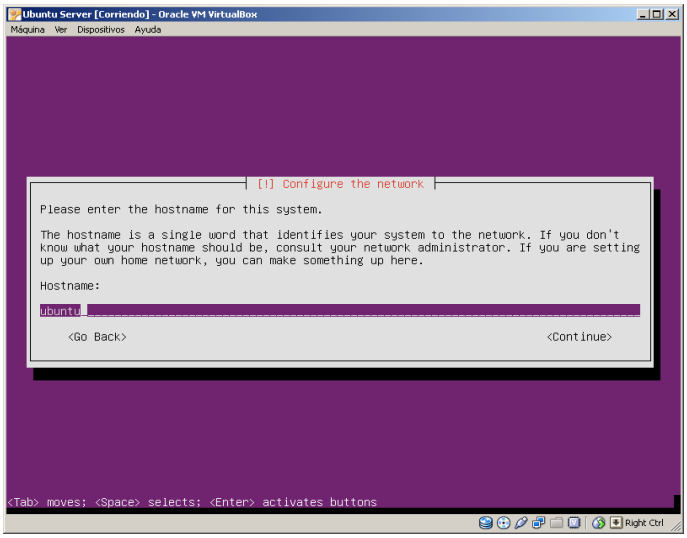


Fig. 5.3. Aceptar Selección de Lenguaje

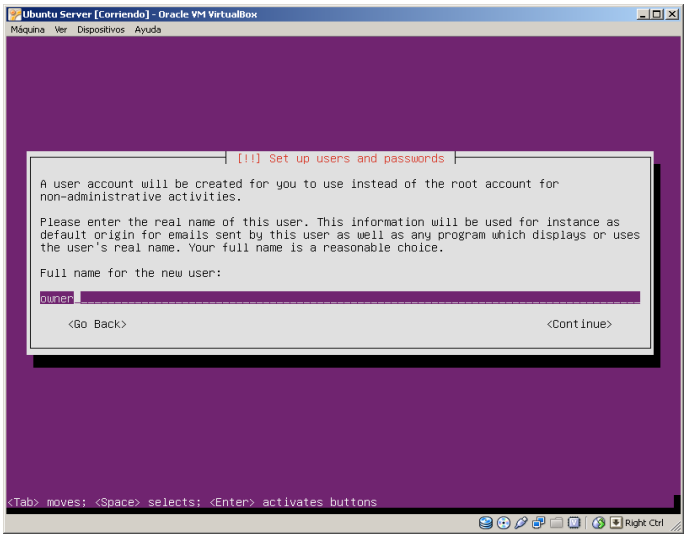


Fig. 5.4. Nombre del Servidor

Luego solicita ingresar un nombre a nuestro servidor y el nombre de usuario para iniciar sesión (figura 5.4).Este usuario es diferente al usuario *root* y no posee derechos de administrador; también nos da una opción de ingresar una contraseña.

Inmediatamente se nos pregunta si queremos encriptar el directorio *home*.En nuestro caso no encriptaremos el directorio (ver figura 5.5) y configuramos el reloj con la zona horario en donde estemos ubicados (ver figura 5.6).

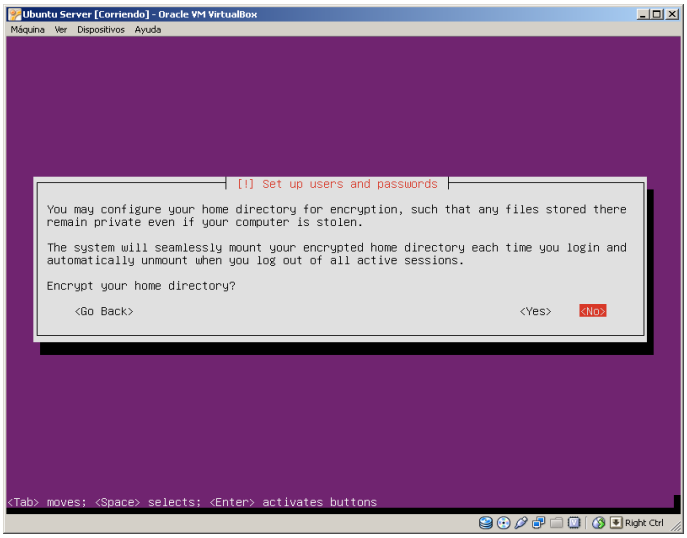


Fig.5.5 Encriptación Directorio

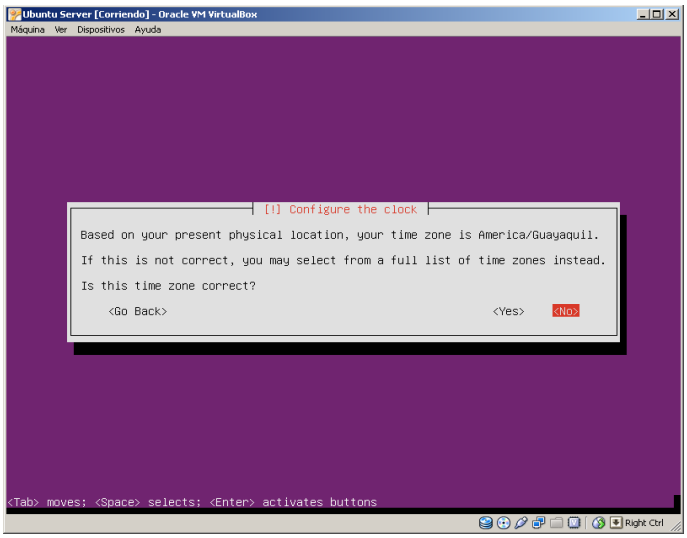


Fig. 5.6. Configuración de Reloj

Para continuar con la instalación, escogemos la segunda opción como se puede observar en la figura 5.7, la cual es para la selección del disco de almacenamiento. Como en este proyecto usaremos el disco entero, seleccionamos *YES*, paraluego seguir con la partición del disco en caso de

ser necesaria (ver figura 5.8), seleccionamos *CONTINUE* para aceptar lo antes ingresado.

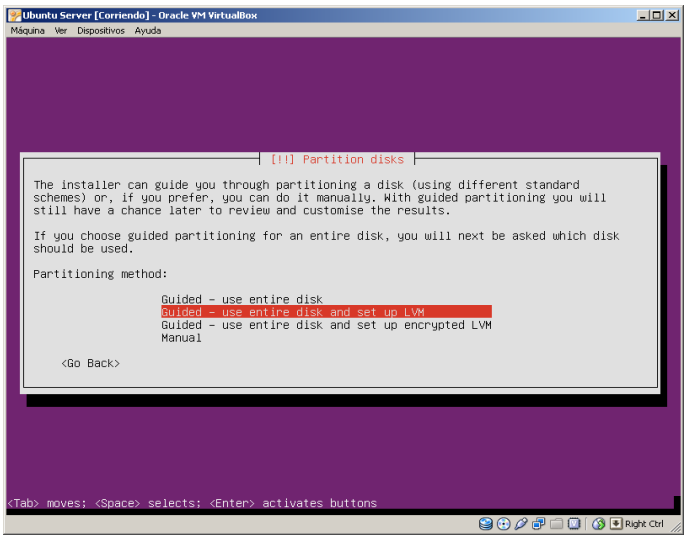


Fig. 5.7. Selección de Disco

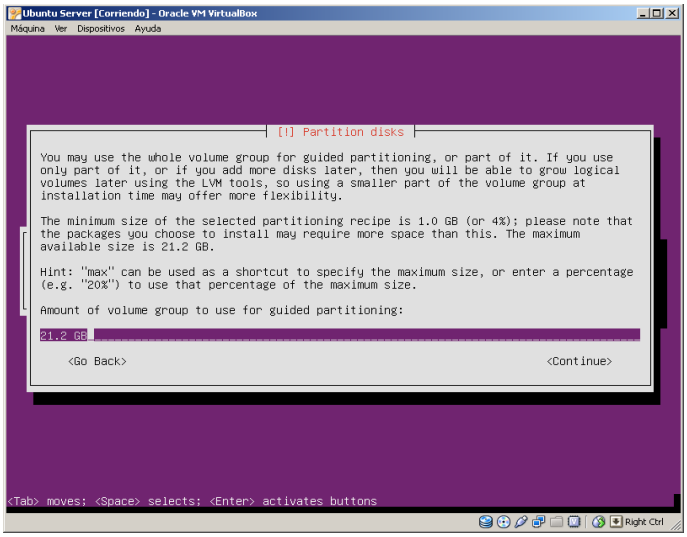


Fig. 5.8 Partición del disco

A continuación, esperamos que el proceso de partición termine por sí solo. Durante este procedimiento nos pide que digitemos la información de nuestro

proxy, pero simplemente le damos *CONTINUE* porque no estamos detrás de un *firewall* que nos impida el acceso a internet.Durante este proceso se va preparando el sistema para luego escoger los programas que deseamos instalar, seleccionamos *OpenSSH server* (ver figura 5.10).Durante el proceso de instalación el sistema nos pregunta de qué manera vamos administrar las actualizaciones (ver figura 5.9).

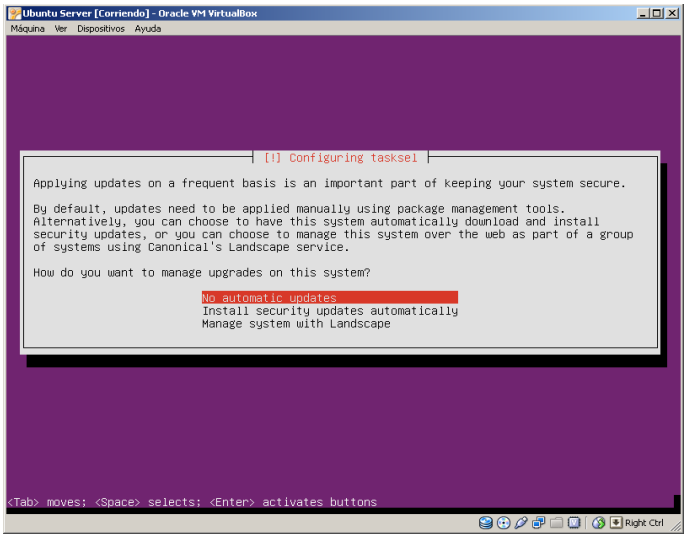


Fig. 5.9. Administración de Actualizaciones del sistema.

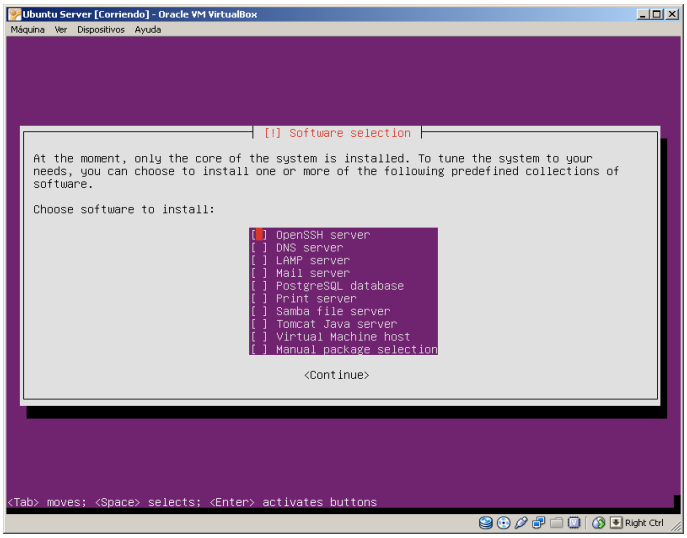


Fig. 5.10. Lista de Programas.

Por último cuando termina el proceso, instalamos el *GRUB boot loader* al MBR, esto en caso de tener algún otro sistema operativo en nuestro servidor (ver figura 5.11).

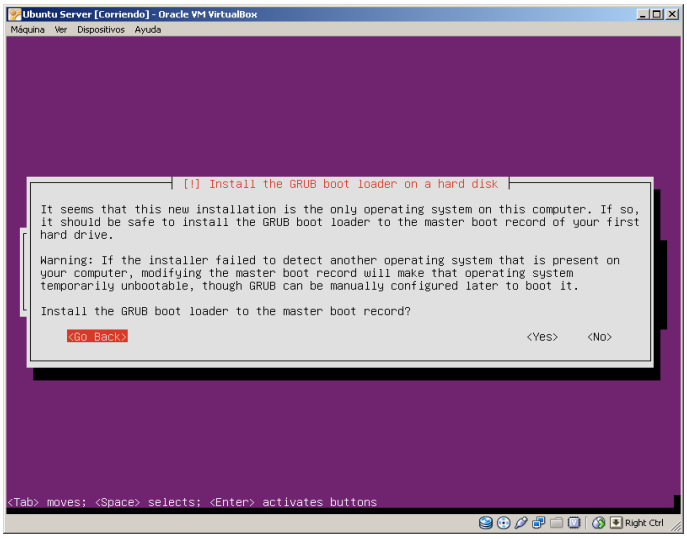


Fig. 5.11. GRUB boot loader

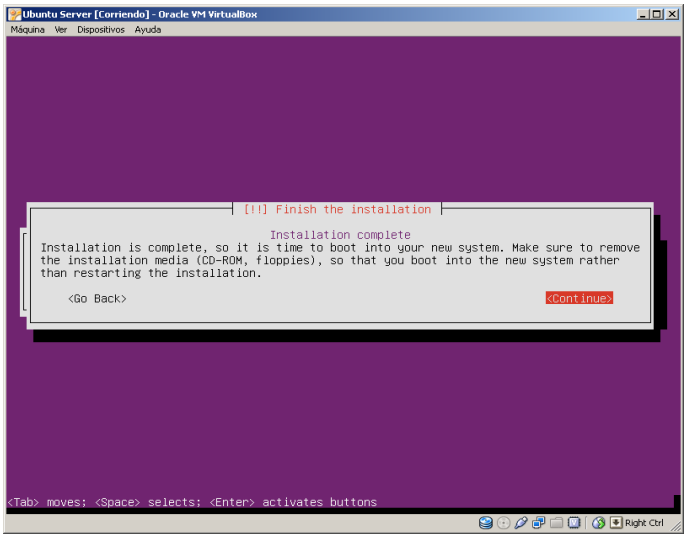


Fig. 5.12. Instalación Completa

5.2 Inicializando con Asterisk

Una vez instalado Ubuntu Server 12.04 procedemos con la instalación de Asterisk 11.

Antes de iniciar el proceso de instalación hay que asegurarse que el sistemaoperativo esté actualizado, para ello ejecutamos en la terminal del servidor:

[owner@asterisk -]# sudoapt-get update && apt-get upgrade –y && reboot

Luego instalamos las dependencias básicas que necesitaremos para Asterisk.

[owner@asterisk -]# sudoapt-get install build-essential wget libssl-dev libncurses5-dev libnewt-dev libxml2-dev linux-headers-$(uname –r) libsqlite3-dev

Descargar las versiones actualizadas del Asterisk, Dahdi, Lipbri.

cd /usr/src/

*[owner@asterisk -]# sudowget*http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-current.tar.gz

*[owner@asterisk -]# sudowget*http://downloads.asterisk.org/pub/telephony/lipbri/lipbri-1.4-current.tar.gz

*[owner@asterisk -]# sudowget*http://downloads.asterisk.org/pub.telephony/asterisk/asterisk-11-current.tar.gz

Extraer los archivos descargados.

*[owner@asterisk -]# sudo tar xzvf dahdi-linux-complete**

*[owner@asterisk -]# sudo tar xzvf lipbri**

*[owner@asterisk -]# sudo tar xzvf asterisk**

Instalación de DAHDI.

*[owner@asterisk -]# cd /usr/src/dahdi-linux-complete**

[owner@asterisk -]# sudomake&& make install && make config

Instalación de Libpri.

*[owner@asterisk -]# cd /usr/src/libpri**

[owner@asterisk -]# sudomake&& make install

Instalamos asterisk, cuando aparezca *menuselect,*seleccionar las opciones deseadas, luego dar *Save & Exite* continuando con la instalación ver figura 5.13.

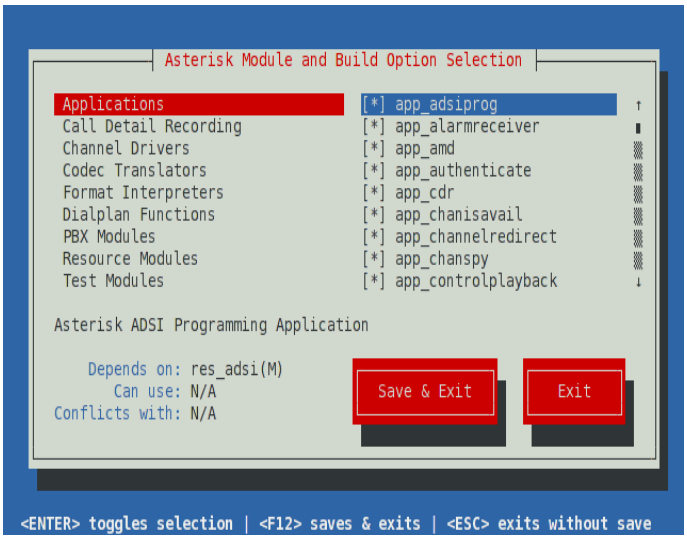


Fig. 5.13. Grafica de Menuselect

Ejecutando *make samples* se crean archivos de configuración básicos para que el usuario los tome de guía al momento de crear su propia configuración.

*[owner@asterisk -]# cd /usr/src/asterisk**

[owner@asterisk -]# sudo./configure &&sudo make menuselect && sudo make && sudo make install && sudo make config && sudo make samples

Inicializar Dahdi.

[owner@asterisk -]# sudo /etc/init.d/dahdi start

Inicializar Asterisk y conectar a CLI.

[owner@asterisk -]# sudoetc/init.d/asterisk start

[owner@asterisk -]# sudo asterisk –rvvv

Verificar la instalación chequeando la versión de Dahdi y Libpri instalados.

**CLI > dahdi show versión*

**CLI > pri show version*

Editor *Visual Vim.*

Dentro de Linux tenemos varios editores de texto en los que podemos nombrar *nano*, *gedit*, *vim*, entre otros; debido a su sencillez y fácil manejo usaremos el editor *vim* para lo cual ofrecemos una pequeña guía para su manejo.

Crear archivo: *> vim /directorio/nombre_archivo*.

Una vez q tengamos abierto el archivo en nuestra consola presionamos la tecla “I” (*Insert*) para editar o añadir información.

Cuando ya hayamos terminado de editar tecleamos la tecla *ESC* y luego escribiremos “!” para guardar los cambios; para salir del editor pulsamos *exit*.

Configuración de IP en Ubuntu Server*.*

La interconexión a través de protocolo Ethernet de ambos servidores se hará mediante una red sencilla, sin embargo en algunos casos habrá que conectarse a Internet para descargar actualizaciones o algún programa que se requiera en su momento; para esto editaremos el archivo *interfaces* que se encuentra ubicado en la ruta */etc/network/.*

Recordar utilizar el comando *sudo* para ejecutar como administrador; si se está ingresando como usuario *root* no es necesario este comando.

IP dinámica.

[owner@asterisk-]# sudo vim /etc/network/interfaces.Para configurarlo de manera dinámica simplemente dejamos la configuración con *dhcp* (ver Fig.5.14)

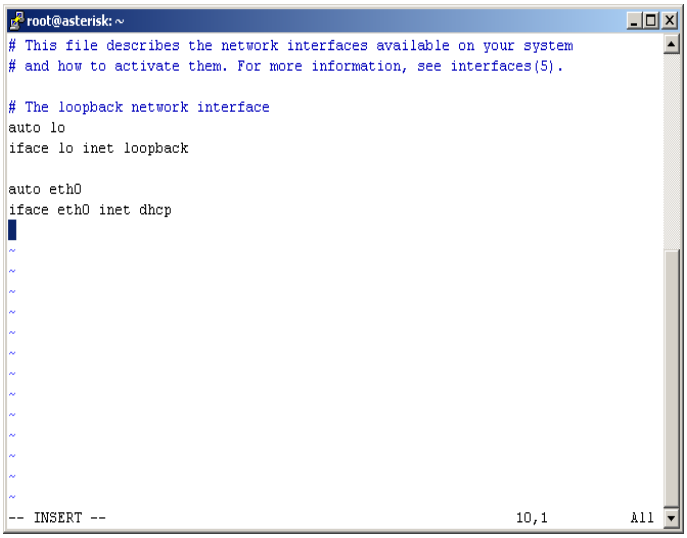


Fig 5.14. Configuración de IP DHCP

Estos parámetros también pueden ser configurados de manera estática (*static*).En dicha configuración tenemos IP, mascara de subred, pueta de enlace (ver figura 5.15).

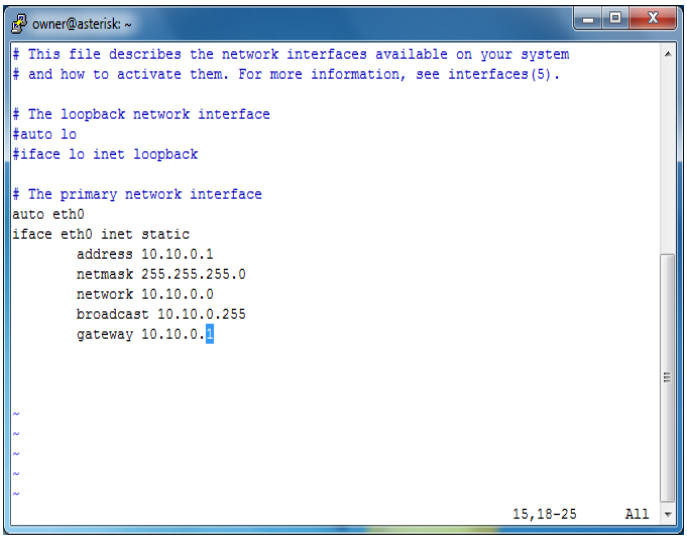


Fig. 5.15 Configuración de IP STATIC

Una vez que hemos configurado procedemos a reiniciar el servicio de red.

[owner@asterisk -]# sudo /etc/init.d/networking restart

Para verificar los cambios ejecutamos y lo realizamos con la siguiente línea de comando.

[owner@asterisk -]# sudo ifconfig

En caso de no haber cambiado la dirección IP en la tarjeta de red, procedemos a reiniciar el servidor.

[owner@asterisk -]# sudo reboot

Open SSH Serve.

Con esta aplicación podemos acceder remotamente a nuestros servidores; si queremos conectarnos desde cualquier parte en donde tengamos acceso a Internet debemos configurar una IP pública en cada servidor, si no se dispone de esta IP solo podremos acceder dentro de la red local en donde se encuentren los computadores.

Instalación

[owner@asterisk -]# sudo apt-get install openssh-server

En la parte de configuración podemos modificar el puerto a utilizar, asi como configurar el*passwords* para la conexión remota.

[owner@asterisk -]# sudo vim /etc/ssh/sshd_config

Correr asterisk en Segundo plano.

Si necesitamos seguir ejecutando comandos en la terminal del servidor, ejecutamos el siguiente comando para correr asterisk sin estar en la consola *CLI*.

[owner@asterisk -]# asterisk -cvvv

Ubicación de módulos en Asterisk

Un módulo es un componente que provee una funcionalidad específica, como por ejemplo un controlador de canal *(chan_iax2.so),* o un recurso que permite la conexión a una tecnología externa. La ubicación de los módulos en asterisk se encuentra en:

/usr/lib/asterisk/modules

Ubicación dearchivos de configuración en asterisk

Son archivos que definen parámetros para varios canales, recursos, módulos y funciones que puedan usarse, este es el directorio en que se focalizará la mayor parte del trabajo.

/etc/asterisk/

Ubicación de ejecutables y configuración (*setup*)

En esta dirección se encuentran algunos archivos de configuración, tales como asterisk, dahdi.

/etc/init.d/

Al manejar asterisk pueden existir ciertos inconvenientes con los derechos de usuario en algunos directorios, es por esto que procedemos a redefinir los derechos de administrador paralos siguientes directorios:

sudo chown - R usuario; usuario /usr/lib/asterisk/

sudo chown - R usuario; usuario /var/lib/asterisk/

sudo chown - R usuario; usuario /var/spool/asterisk/

sudo chown - R usuario; usuario /var/log/asterisk/

sudo chown - R usuario; usuario /var/run/asterisk/

sudo chown usuario; usuario /usr/sbin/asterisk/

En donde *usuario* es el nombre de la cuenta que va a manejar asterisk.

La consola de asterisk ofrece una lista de detalles al momento de realizar o recibir llamadas y otras funciones, sin embargo puede presentar algunos mensajes *warnings* que pueden resultar molestos para el usuario.

Este tipo de mensajes pueden ser habilitados o deshabilitados en el archivo *logger.conf*ubicado en la ruta */etc/Asterisk*.

Habilitar usuario *root*

Utilizar el comando sudo y a la vez escribir la contraseña de administrador, esto con el tiempo puede causar molestia; por lo que es preferible trabajar con la cuenta root del sistema para evitar este inconveniente.

sudo passwd –u root

Si no existe *password* no se coloca el parámetro *–u*; a continuación escribimos el *password* de la cuenta *root* y se ingresa nuevamente.

5.3 Pruebas con Equipos

Entre las herramientas de monitoreo tenemos una diversidad de opciones disponibles para Linux, en las que podemos mencionar:

Wireshark: Analizador de protocolos de red, eficiente para la captura y análisis de paquetes; para utilizarlo es recomendable instalar una interfaz gráfica al servidor y esto no conviene si se posee una PC de pocos recursos.

Jnettop: Visualizador de red, captura el tráfico que pasa a través del *host* en el cual se corre la aplicación y muestra la información según el ancho de banda que ocupe.

Enlace de descarga: http://jnettop.kubs.info/wiki/?id=Download; o en la consola de Ubuntu server ejecutar:

apt-get install jnettop

Ifstat: Herramienta básica que muestra la actividad en la interface de red seleccionada.

apt-get install ifstat

Muestras con Wireshark

Para tomar las muestras con *Wireshark* debemos tener instalado en el servidor la herramienta *Tcpdump* que permite capturar paquetes recibidos y transmitidos, entiempo real, en la red en que se encuentra nuestro servidor.

Ingresamos a la consola de nuestro servidor como usuario *root* y creamos un nuevo directorio en donde guardaremos las capturas.

[root@asterisk -]# mkdircapturas

Entramos al directorio creado

[root@asterisk-]# cdcapturas

Para proceder con la captura utilizamos el comando:

[root@asterisk -]# tcpdump–i any –s 1500 -w capturaXX.pcap

Donde “XX” es el número de captura, las que se grabaran con extensión *.pcap* que puede ser leída por *Wireshark*.

Una vez ejecutado el comando comenzará a capturar paquetes de todas las interfaces en el servidor, si queremos que solo tome capturas de una interface agregamos el nombre de la interfaz (eth0,eth1,eth2…) al lado del comando –i.

Para terminar con la captura tecleamos *CTRL+C*.

Ahora ya podemos copiar las capturas creadas para analizar con *wireshark*.

Mensajes SIP al momento de registrar un usuario SIP a través del softphone Zoiper en Windows.

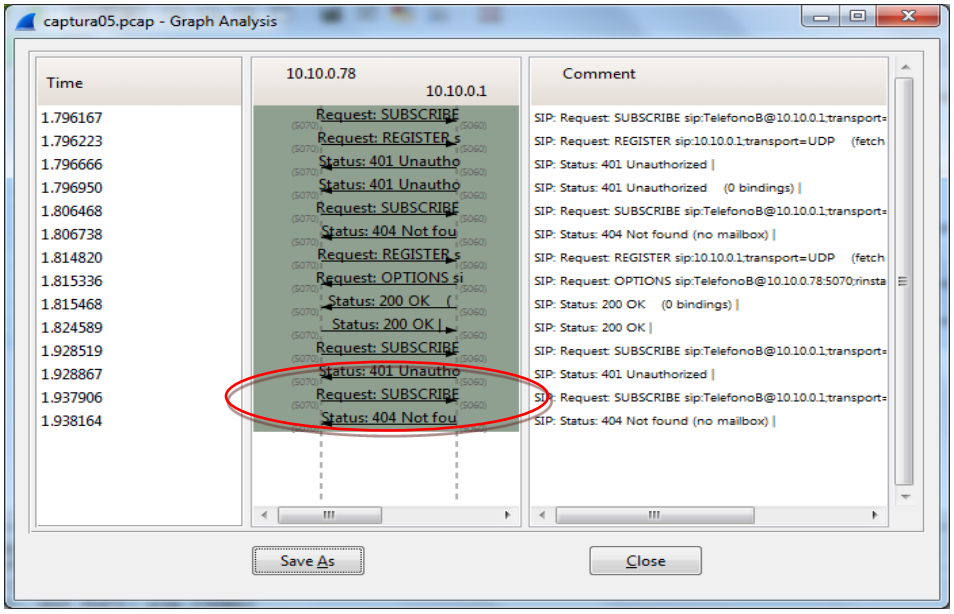
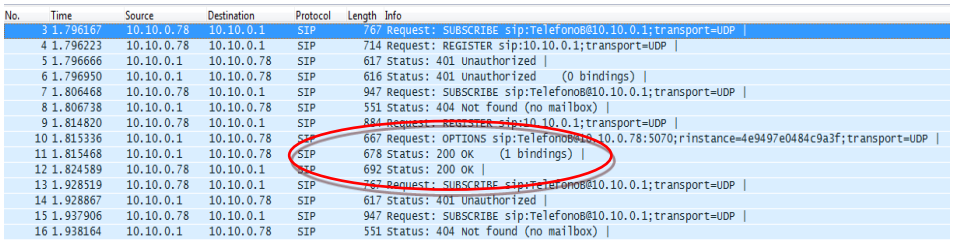


Fig.5.16Diagrama de flujo - Registro de usuario SIP



No.	Time	Source	Destination	Protocol	Length	Info
3	1.796167	10.10.0.78	10.10.0.1	SIP	767	Request: SUBSCRIBE sip:TelefonoB@10.10.0.1;transport=UDP |
4	1.796223	10.10.0.78	10.10.0.1	SIP	714	Request: REGISTER sip:10.10.0.1;transport=UDP |
5	1.796666	10.10.0.1	10.10.0.78	SIP	617	Status: 401 Unauthorized |
6	1.796950	10.10.0.1	10.10.0.78	SIP	616	Status: 401 Unauthorized (0 bindings) |
7	1.806468	10.10.0.78	10.10.0.1	SIP	947	Request: SUBSCRIBE sip:TelefonoB@10.10.0.1;transport=UDP |
8	1.806738	10.10.0.1	10.10.0.78	SIP	551	Status: 404 Not found (no mailbox) |
9	1.814820	10.10.0.78	10.10.0.1	SIP	884	Request: REGISTER sip:10.10.0.1;transport=UDP |
10	1.815336	10.10.0.1	10.10.0.78	SIP	667	Request: OPTIONS sip:TelefonoB@10.10.0.78:5070;rinstance=4e9497e0484c9a3f;transport=UDP |
11	1.815468	10.10.0.1	10.10.0.78	SIP	678	Status: 200 OK (1 bindings) |
12	1.824589	10.10.0.78	10.10.0.1	SIP	692	Status: 200 OK |
13	1.928519	10.10.0.78	10.10.0.1	SIP	767	Request: SUBSCRIBE sip:TelefonoB@10.10.0.1;transport=UDP |
14	1.928867	10.10.0.1	10.10.0.78	SIP	617	Status: 401 Unauthorized |
15	1.937906	10.10.0.78	10.10.0.1	SIP	947	Request: SUBSCRIBE sip:TelefonoB@10.10.0.1;transport=UDP |
16	1.938164	10.10.0.1	10.10.0.78	SIP	551	Status: 404 Not found (no mailbox) |

Fig.5.16.1 Analizado de paquetes - Registro de usuario SIP (B)

Lasfiguras(5.16 -5.16.1) nos indica los pasos para el registro de una cuenta SIP en asterisk, vemos dos estados de *OK 200* lo cual significa una correcta asociación con el servidor.

En las figura 5.17 se describe una llamada SIP, se utiliza un filtro RTP, ya que por este protocolo es por el que viaja el audio perteneciente a la llamada.

Con esta captura de paquetes es posible volver a reproducir la llamada, esto sería muy útil para cuestiones de seguridad.

No.	Time	Source	Destination	Protocol	Length	Info
38	16.297717	10.10.0.8	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0xB1546656, Seq=11244, Time=550161452, Mark
39	16.297940	10.10.0.1	10.10.0.78	RTP	87	PT=GSM 06.10, SSRC=0x772D92D6, Seq=15330, Time=550161448, Mark
41	16.331972	10.10.0.78	10.10.0.1	RTP	87	PT=GSM 06.10, SSRC=0x58844B70, Seq=17354, Time=1974535790, Mark
42	16.332128	10.10.0.1	10.10.0.58	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x20552685, Seq=41659, Time=1974535784, Mark
43	16.351913	10.10.0.78	10.10.0.1	RTP	87	PT=GSM 06.10, SSRC=0x58844B70, Seq=17355, Time=1974535950
44	16.352005	10.10.0.1	10.10.0.58	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x20552685, Seq=41660, Time=1974535944
45	16.371924	10.10.0.78	10.10.0.1	RTP	87	PT=GSM 06.10, SSRC=0x58844B70, Seq=17356, Time=1974536110
46	16.372010	10.10.0.1	10.10.0.58	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x20552685, Seq=41661, Time=1974536104
47	16.391958	10.10.0.78	10.10.0.1	RTP	87	PT=GSM 06.10, SSRC=0x58844B70, Seq=17357, Time=1974536270
48	16.392042	10.10.0.1	10.10.0.58	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x20552685, Seq=41662, Time=1974536264
49	16.411945	10.10.0.78	10.10.0.1	RTP	87	PT=GSM 06.10, SSRC=0x58844B70, Seq=17358, Time=1974536430
50	16.412029	10.10.0.1	10.10.0.58	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x20552685, Seq=41663, Time=1974536424
51	16.431924	10.10.0.78	10.10.0.1	RTP	87	PT=GSM 06.10, SSRC=0x58844B70, Seq=17359, Time=1974536590
52	16.432011	10.10.0.1	10.10.0.58	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x20552685, Seq=41664, Time=1974536584
53	16.451897	10.10.0.78	10.10.0.1	RTP	87	PT=GSM 06.10, SSRC=0x58844B70, Seq=17360, Time=1974536750
54	16.451984	10.10.0.1	10.10.0.58	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x20552685, Seq=41665, Time=1974536744
57	16.467819	10.10.0.58	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x10020DE4, Seq=18806, Time=2570043030, Mark
58	16.468012	10.10.0.1	10.10.0.78	RTP	87	PT=GSM 06.10, SSRC=0x5BD8DB0E, Seq=15331, Time=2570043024, Mark
59	16.471922	10.10.0.78	10.10.0.1	RTP	87	PT=GSM 06.10, SSRC=0x58844B70, Seq=17361, Time=1974536910
60	16.472013	10.10.0.1	10.10.0.58	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x20552685, Seq=41666, Time=1974536904
61	16.487494	10.10.0.58	10.10.0.1	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x10020DE4, Seq=18807, Time=2570043190
62	16.487632	10.10.0.1	10.10.0.78	RTP	87	PT=GSM 06.10, SSRC=0x5BD8DB0E, Seq=15332, Time=2570043184

Fig. 5.17. Llamada SIP visto en wireshark

En asterisk este proceso se muestra de la siguiente manera. (ver figura 5.18).

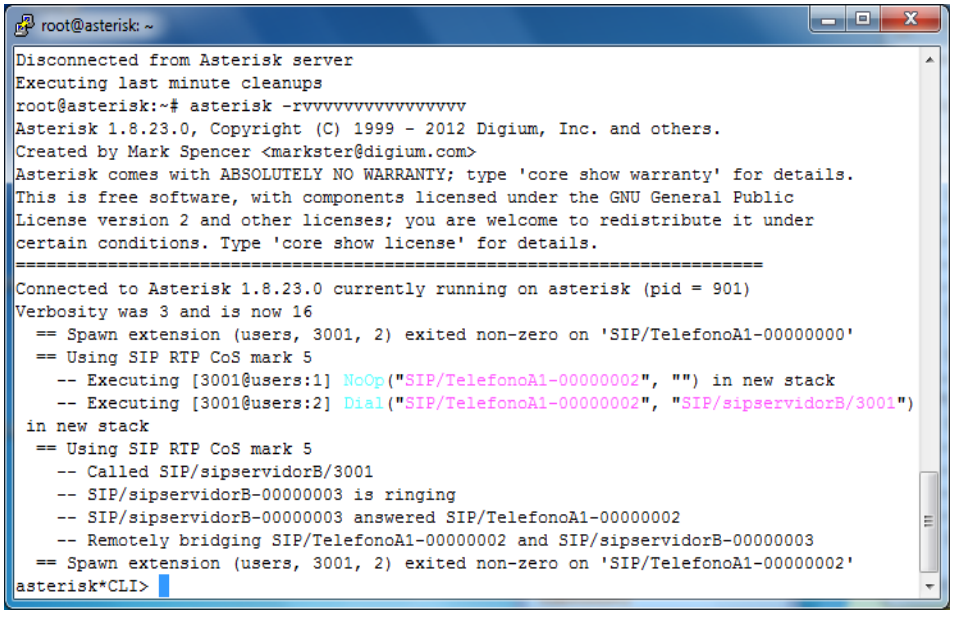


Fig 5.18. Llamada SIP Servidor A

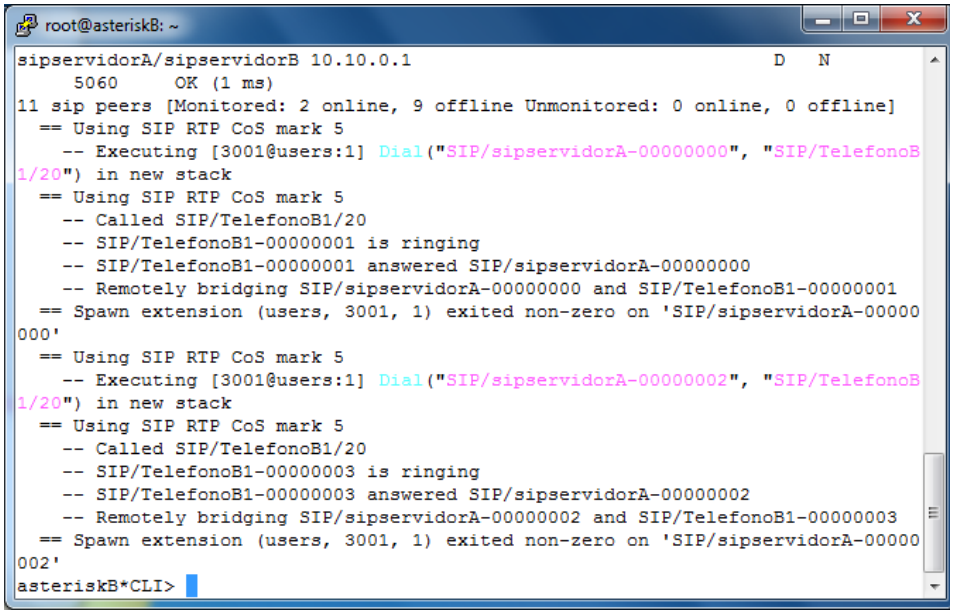


Fig. 5.19 Llama entre dos usuarios SIP en Asterisk

En las figuras 5.18 y 5.19 vemos el proceso de llamada desde la consola asterisk.En este ejemplo se realiza una llamada desde una cuenta conectada desde el servidor A (Teléfono A1) hacia una cuenta conectada al servidor B (Teléfono B1)

Las líneas en color rosa indican el proceso, para lo cual realizaremos un diagrama de flujo para una mejor interpretación.

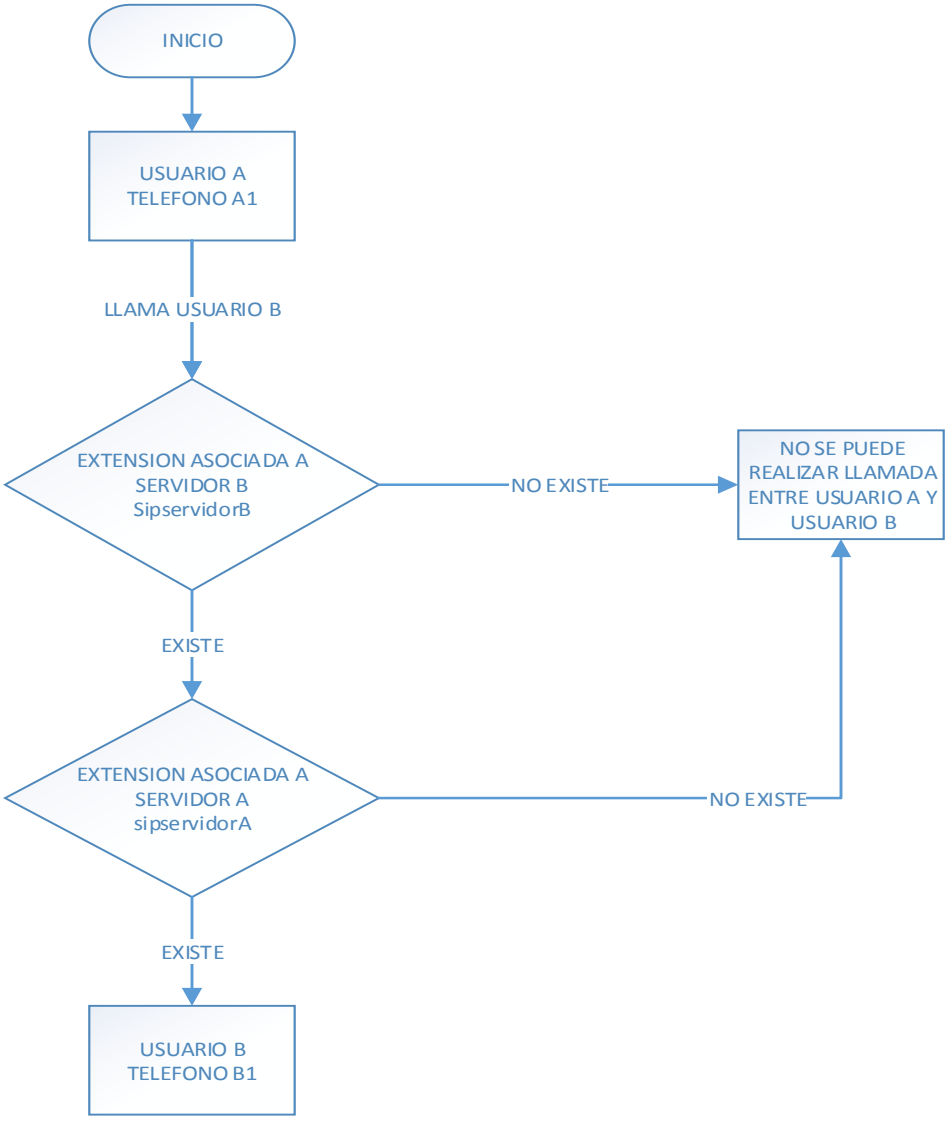


Fig. 5.20 Diagrama de Flujo Llamada SIP entre servidores

El diagrama de flujo anterior, representado en la figura 5.20, nos indica que para realizar una llamada entre un usuario conectado al servidor A y un usuario conectado al servidor B deben existir dos extensiones de asociación entre los servidores.Estas extensiones hacen de asterisk un cliente SIP, lo cual permite que se puedan registrar en el otro servidor.

Si no existieran estas dos extensiones no se pudieran realizar llamadas entre ambos servidores a través del protocolo SIP. El mismo procedimiento se aplica para la conexión por medio de IAX.

La figura 5.21 nos muestra la terminación de una llamada.Observamos el mensaje *BYE* utilizado por este protocolo para indicar que ha finalizado la comunicación.

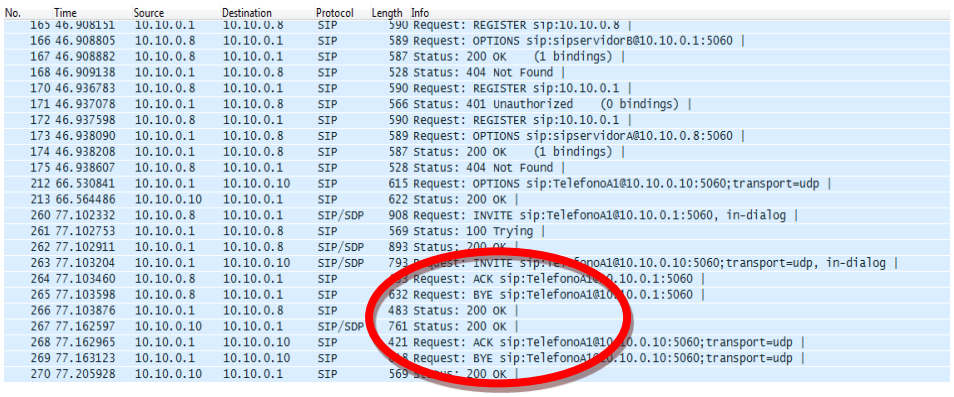
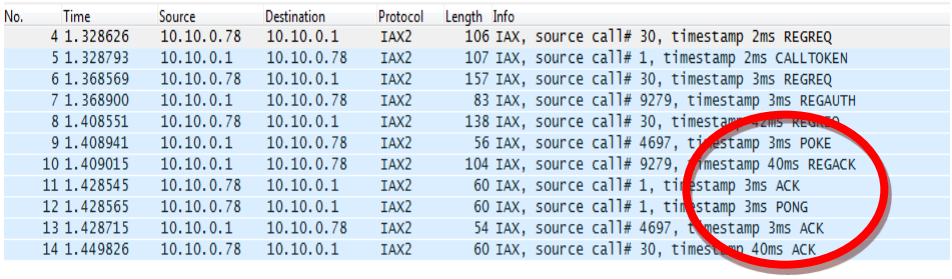


Fig 5.21 Culminación de llamada SIP visto en wireshark

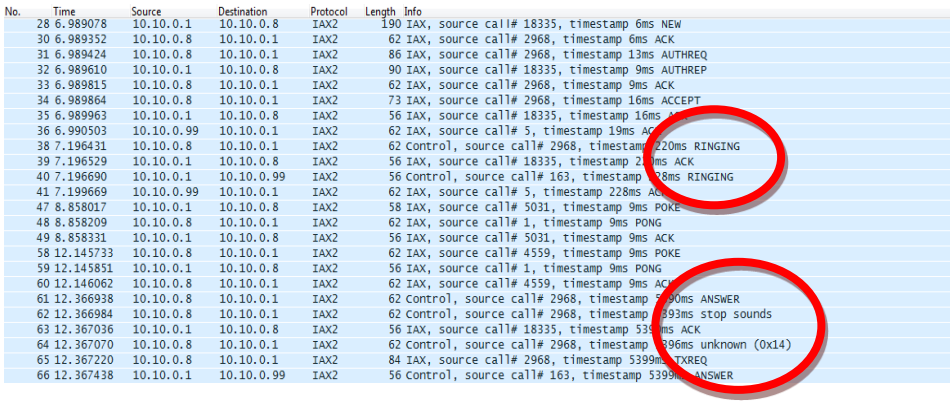
La figura 5.22 nos muestra los paquetes que se transmiten al momento de suscribir o registrar un usuario IAX en el servidor asterisk, observamos cómo son enviados los mensajes *ACK,* los que indican que se ha establecido la conexión.



No.	Time	Source	Destination	Protocol	Length	Info
4	1.328626	10.10.0.78	10.10.0.1	IAX2	106	IAX, source call# 30, timestamp 2ms REGREQ
5	1.328793	10.10.0.1	10.10.0.78	IAX2	107	IAX, source call# 1, timestamp 2ms CALLTOKEN
6	1.368569	10.10.0.78	10.10.0.1	IAX2	157	IAX, source call# 30, timestamp 3ms REGREQ
7	1.368900	10.10.0.1	10.10.0.78	IAX2	83	IAX, source call# 9279, timestamp 3ms REGAUTH
8	1.408551	10.10.0.78	10.10.0.1	IAX2	138	IAX, source call# 30, timestamp 42ms REGREQ
9	1.408941	10.10.0.1	10.10.0.78	IAX2	56	IAX, source call# 4697, timestamp 3ms POKE
10	1.409015	10.10.0.1	10.10.0.78	IAX2	104	IAX, source call# 9279, timestamp 40ms REGACK
11	1.428545	10.10.0.78	10.10.0.1	IAX2	60	IAX, source call# 1, timestamp 3ms ACK
12	1.428565	10.10.0.78	10.10.0.1	IAX2	60	IAX, source call# 1, timestamp 3ms PONG
13	1.428715	10.10.0.1	10.10.0.78	IAX2	54	IAX, source call# 4697, timestamp 3ms ACK
14	1.449826	10.10.0.78	10.10.0.1	IAX2	60	IAX, source call# 30, timestamp 40ms ACK

Fig. 5.22 Registro de usurio IAX visto en wireshark

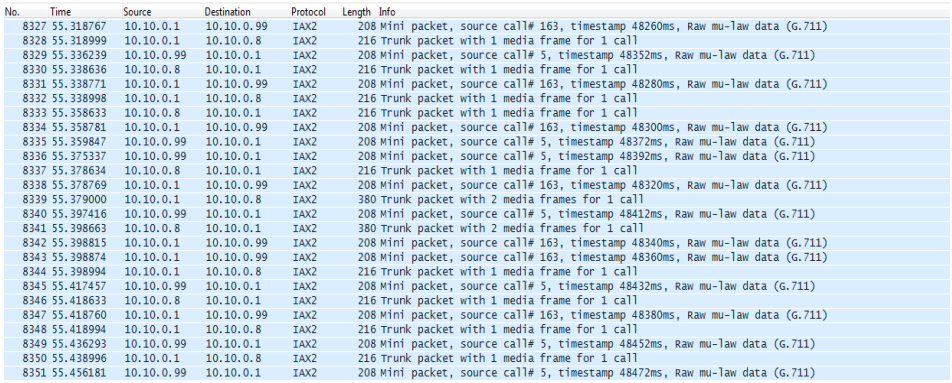
La siguiente figura 5.23 nos muestra los mensajes que utiliza IAX para el establecimiento de una llamada; vemos el mensaje *RINGING*lo cual significa que hay un tono de timbrado y a la vez observamos los mensajes *ANSWER*cuando ya se establece la comunicación entre ambos usuarios IAX.



No.	Time	Source	Destination	Protocol	Length	Info
28	6.989078	10.10.0.1	10.10.0.8	IAX2	190	IAX, source call# 18335, timestamp 6ms NEW
30	6.989352	10.10.0.8	10.10.0.1	IAX2	62	IAX, source call# 2968, timestamp 6ms ACK
31	6.989424	10.10.0.8	10.10.0.1	IAX2	86	IAX, source call# 2968, timestamp 13ms AUTHREQ
32	6.989610	10.10.0.1	10.10.0.8	IAX2	90	IAX, source call# 18335, timestamp 9ms AUTHREP
33	6.989815	10.10.0.8	10.10.0.1	IAX2	62	IAX, source call# 2968, timestamp 9ms ACK
34	6.989864	10.10.0.8	10.10.0.1	IAX2	73	IAX, source call# 2968, timestamp 16ms ACCEPT
35	6.989963	10.10.0.1	10.10.0.8	IAX2	56	IAX, source call# 18335, timestamp 16ms ACK
36	6.990503	10.10.0.99	10.10.0.1	IAX2	62	IAX, source call# 5, timestamp 19ms ACK
38	7.196431	10.10.0.8	10.10.0.1	IAX2	62	Control, source call# 2968, timestamp 220ms RINGING
39	7.196529	10.10.0.1	10.10.0.8	IAX2	56	IAX, source call# 18335, timestamp 220ms ACK
40	7.196690	10.10.0.1	10.10.0.99	IAX2	56	Control, source call# 163, timestamp 228ms RINGING
41	7.199669	10.10.0.99	10.10.0.1	IAX2	62	IAX, source call# 5, timestamp 228ms ACK
47	8.858017	10.10.0.1	10.10.0.8	IAX2	58	IAX, source call# 5031, timestamp 9ms POKE
48	8.858209	10.10.0.8	10.10.0.1	IAX2	62	IAX, source call# 1, timestamp 9ms PONG
49	8.858331	10.10.0.1	10.10.0.8	IAX2	56	IAX, source call# 5031, timestamp 9ms ACK
58	12.145733	10.10.0.8	10.10.0.1	IAX2	62	IAX, source call# 4559, timestamp 9ms POKE
59	12.145851	10.10.0.1	10.10.0.8	IAX2	56	IAX, source call# 1, timestamp 9ms PONG
60	12.146062	10.10.0.8	10.10.0.1	IAX2	62	IAX, source call# 4559, timestamp 9ms ACK
61	12.366938	10.10.0.8	10.10.0.1	IAX2	62	Control, source call# 2968, timestamp 5390ms ANSWER
62	12.366984	10.10.0.8	10.10.0.1	IAX2	62	Control, source call# 2968, timestamp 5393ms stop sounds
63	12.367036	10.10.0.1	10.10.0.8	IAX2	56	IAX, source call# 18335, timestamp 5396ms ACK
64	12.367070	10.10.0.8	10.10.0.1	IAX2	62	Control, source call# 2968, timestamp 5396ms unknown (0x14)
65	12.367220	10.10.0.8	10.10.0.1	IAX2	84	IAX, source call# 2968, timestamp 5399ms TXREQ
66	12.367438	10.10.0.1	10.10.0.99	IAX2	56	Control, source call# 163, timestamp 5399ms ANSWER

Fig. 5.23Establecimiento de llamada IAX visto en wireshark

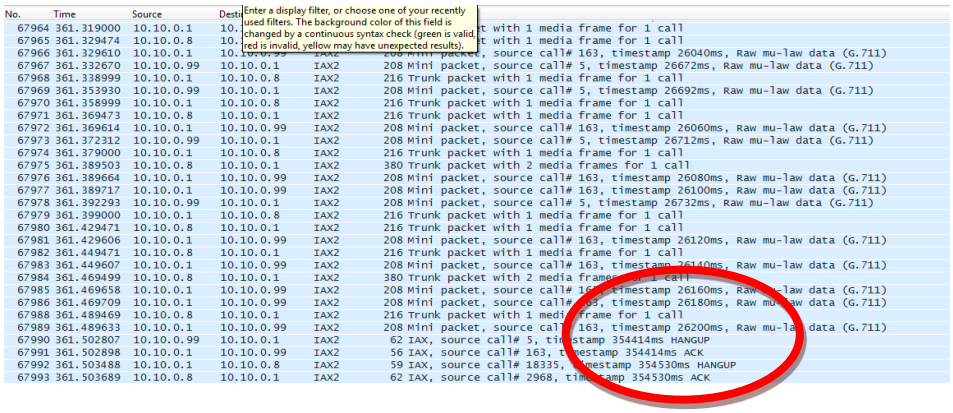
En la figura 5.24 nos muestra el transcurso de la llamada, aquí se puede observar las tramas o *frames* que utiliza el protocolo IAX para transportar el audio, a su vez se ve el *codec* de audio utilizado en esta llamada (*G.711*)



No.	Time	Source	Destination	Protocol	Length	Info
8327	55.318767	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 48260ms, Raw mu-law data (G.711)
8328	55.318999	10.10.0.1	10.10.0.8	IAX2	216	Trunk packet with 1 media frame for 1 call
8329	55.336239	10.10.0.99	10.10.0.1	IAX2	208	Mini packet, source call# 5, timestamp 48352ms, Raw mu-law data (G.711)
8330	55.338636	10.10.0.8	10.10.0.1	IAX2	216	Trunk packet with 1 media frame for 1 call
8331	55.338771	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 48280ms, Raw mu-law data (G.711)
8332	55.338998	10.10.0.1	10.10.0.8	IAX2	216	Trunk packet with 1 media frame for 1 call
8333	55.358633	10.10.0.8	10.10.0.1	IAX2	216	Trunk packet with 1 media frame for 1 call
8334	55.358781	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 48300ms, Raw mu-law data (G.711)
8335	55.359847	10.10.0.99	10.10.0.1	IAX2	208	Mini packet, source call# 5, timestamp 48372ms, Raw mu-law data (G.711)
8336	55.375337	10.10.0.99	10.10.0.1	IAX2	208	Mini packet, source call# 5, timestamp 48392ms, Raw mu-law data (G.711)
8337	55.378634	10.10.0.8	10.10.0.1	IAX2	216	Trunk packet with 1 media frame for 1 call
8338	55.378769	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 48320ms, Raw mu-law data (G.711)
8339	55.379000	10.10.0.1	10.10.0.8	IAX2	380	Trunk packet with 2 media frames for 1 call
8340	55.397416	10.10.0.99	10.10.0.1	IAX2	208	Mini packet, source call# 5, timestamp 48412ms, Raw mu-law data (G.711)
8341	55.398663	10.10.0.8	10.10.0.1	IAX2	380	Trunk packet with 2 media frames for 1 call
8342	55.398815	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 48340ms, Raw mu-law data (G.711)
8343	55.398874	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 48360ms, Raw mu-law data (G.711)
8344	55.398994	10.10.0.1	10.10.0.8	IAX2	216	Trunk packet with 1 media frame for 1 call
8345	55.417457	10.10.0.99	10.10.0.1	IAX2	208	Mini packet, source call# 5, timestamp 48432ms, Raw mu-law data (G.711)
8346	55.418633	10.10.0.8	10.10.0.1	IAX2	216	Trunk packet with 1 media frame for 1 call
8347	55.418760	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 48380ms, Raw mu-law data (G.711)
8348	55.418994	10.10.0.1	10.10.0.8	IAX2	216	Trunk packet with 1 media frame for 1 call
8349	55.436293	10.10.0.99	10.10.0.1	IAX2	208	Mini packet, source call# 5, timestamp 48452ms, Raw mu-law data (G.711)
8350	55.438996	10.10.0.1	10.10.0.8	IAX2	216	Trunk packet with 1 media frame for 1 call
8351	55.456181	10.10.0.99	10.10.0.1	IAX2	208	Mini packet, source call# 5, timestamp 48472ms, Raw mu-law data (G.711)

Fig. 5.24 Llamada en curso IAX visto en wireshark

La figura 5.25 muestra los mensajes *HANGUP* utilizados por IAX para determinar la culminación o cierre de una llamada.



No.	Time	Source	Destination	Protocol	Length	Info
67964	361.319000	10.10.0.1	10.			t with 1 media frame for 1 call
67965	361.329474	10.10.0.8	10.			t with 1 media frame for 1 call
67966	361.329610	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 26040ms, Raw mu-law data (G.711)
67967	361.332670	10.10.0.99	10.10.0.1	IAX2	208	Mini packet, source call# 5, timestamp 26672ms, Raw mu-law data (G.711)
67968	361.338999	10.10.0.1	10.10.0.8	IAX2	216	Trunk packet with 1 media frame for 1 call
67969	361.353930	10.10.0.99	10.10.0.1	IAX2	208	Mini packet, source call# 5, timestamp 26692ms, Raw mu-law data (G.711)
67970	361.358999	10.10.0.1	10.10.0.8	IAX2	216	Trunk packet with 1 media frame for 1 call
67971	361.369473	10.10.0.8	10.10.0.1	IAX2	216	Trunk packet with 1 media frame for 1 call
67972	361.369614	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 26060ms, Raw mu-law data (G.711)
67973	361.372312	10.10.0.99	10.10.0.1	IAX2	208	Mini packet, source call# 5, timestamp 26712ms, Raw mu-law data (G.711)
67974	361.379000	10.10.0.1	10.10.0.8	IAX2	216	Trunk packet with 1 media frame for 1 call
67975	361.389503	10.10.0.8	10.10.0.1	IAX2	380	Trunk packet with 2 media frames for 1 call
67976	361.389664	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 26080ms, Raw mu-law data (G.711)
67977	361.389717	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 26100ms, Raw mu-law data (G.711)
67978	361.392293	10.10.0.99	10.10.0.1	IAX2	208	Mini packet, source call# 5, timestamp 26732ms, Raw mu-law data (G.711)
67979	361.399000	10.10.0.1	10.10.0.8	IAX2	216	Trunk packet with 1 media frame for 1 call
67980	361.429471	10.10.0.8	10.10.0.1	IAX2	216	Trunk packet with 1 media frame for 1 call
67981	361.429606	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 26120ms, Raw mu-law data (G.711)
67982	361.449471	10.10.0.8	10.10.0.1	IAX2	216	Trunk packet with 1 media frame for 1 call
67983	361.449607	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 26140ms, Raw mu-law data (G.711)
67984	361.469499	10.10.0.8	10.10.0.1	IAX2	380	Trunk packet with 2 media frames for 1 call
67985	361.469658	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 26160ms, Raw mu-law data (G.711)
67986	361.469709	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 26180ms, Raw mu-law data (G.711)
67988	361.489469	10.10.0.8	10.10.0.1	IAX2	216	Trunk packet with 1 media frame for 1 call
67989	361.489633	10.10.0.1	10.10.0.99	IAX2	208	Mini packet, source call# 163, timestamp 26200ms, Raw mu-law data (G.711)
67990	361.502807	10.10.0.99	10.10.0.1	IAX2	62	IAX, source call# 5, timestamp 354414ms HANGUP
67991	361.502898	10.10.0.1	10.10.0.99	IAX2	56	IAX, source call# 163, timestamp 354414ms ACK
67992	361.503488	10.10.0.1	10.10.0.8	IAX2	59	IAX, source call# 18335, timestamp 354530ms HANGUP
67993	361.503689	10.10.0.8	10.10.0.1	IAX2	62	IAX, source call# 2968, timestamp 354530ms ACK

Fig. 5.25Culminación de llamada IAX visto en wireshark

La figura 5.26 muestra el proceso de una llamada en IAX con los mensajes que se generan vistosdesde asterisk.

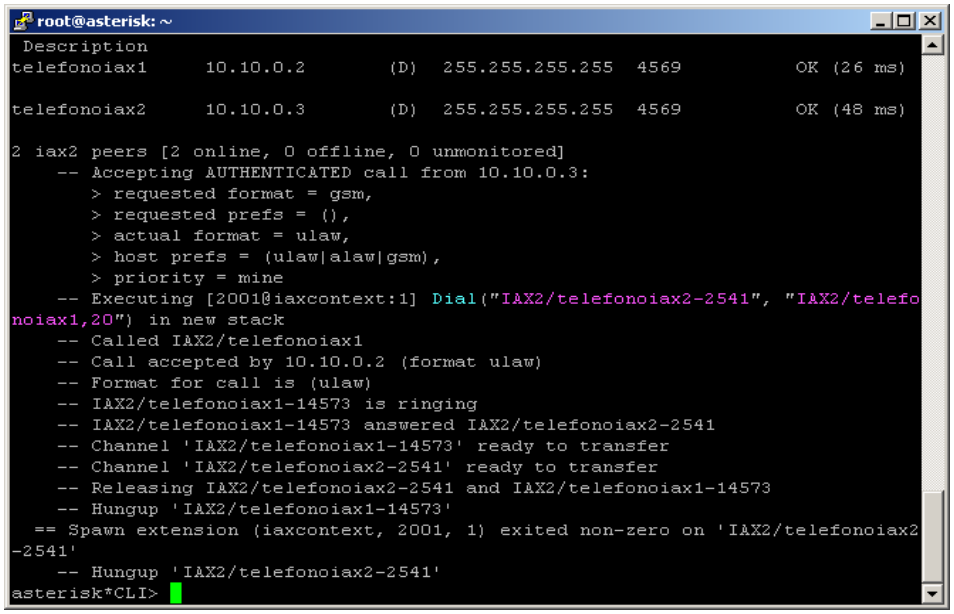


Fig. 5.26 Llama IAX en Asterisk

Una vez que se tengan registradas las extensiones SIP o IAX pueden ser verificadas al mismo tiempo dentro de asterisk, así como lo muestra la figura 5.27.

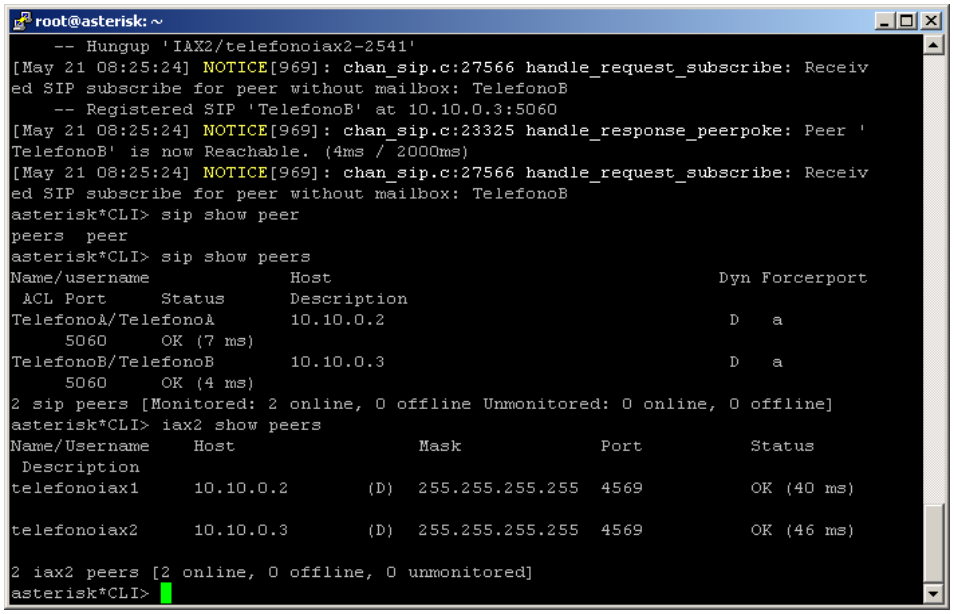


Fig. 5.27 Extensiones SIP e IAX2 registradas

Creación de servicio – Implementación

En esta parte describiremos las configuraciones realizadas para la implementación del servicio descrito en el capítulo anterior.

Servidor A

Añadimos al archivo *extensions.conf* las siguientes líneas:

;Menu de ayuda para Servidor B

exten => 002,1,NoOp()

exten => 002,n,Dial(SIP/sipservidorB/Bhelp); al marcar 002 nos llevara al servidor B y usará la extensión *Bhelp* para las opciones de marcado.

;Menu de ayuda para Servidor A

exten => Ahelp,1,Goto(Ayuda,helpext,1)

[Ayuda]

exten => helpext,1,Answer()

same => n,Background(soporte)

same => n,WaitExten(5)

exten => 1,1,Goto(users,1001,1)

exten => 2,1,Goto(users,1002,1)

exten => 3,1,Goto(users,1003,1)

exten => 4,1,Goto(users,1004,1)

exten => 5,1,Goto(users,1005,1)

Servidor B

exten => 001,1,NoOp()

exten => 001,n,Dial(SIP/sipservidorA/Ahelp);al marcar 001 nos lleva al servidor A donde tomará la extensión *Ahelp* para las opciones de marcado.

;Menu de ayuda

exten => Bhelp,1,Answer()

same => n,Background(soporte)

same => n,Wait(2)

exten => 1,1,Goto(users,3001,1)

exten => 2,1,Goto(users,3002,1)

exten => 3,1,Goto(users,3003,1)

exten => 4,1,Goto(users,3004,1)

exten => 5,1,Goto(users,3005,1)

Para grabar el mensaje de audio a utilizar, editamos las siguientes líneas en cualquiera de los dos servidores en el archivo*extension.conf*.

;Grabacion IVR

exten => 12345,1,Answer()

exten => 12345,n,Wait(1)

exten => 12345,n,Record(soporte.gsm)

exten => 12345,n,Wait(1)

Al marcar 12345 después de 1 segundo sonará un *beep* el cual indica que comienza a grabar el archivo *soporte.gsm* por medio de la aplicación *Record*; para finalizar la aplicación o grabación del mensaje tecleamos # en el dispositivo donde hayamos marcado 12345.

El archivo lo podemos encontrar en la ruta */var/lib/asterisk/sounds/*

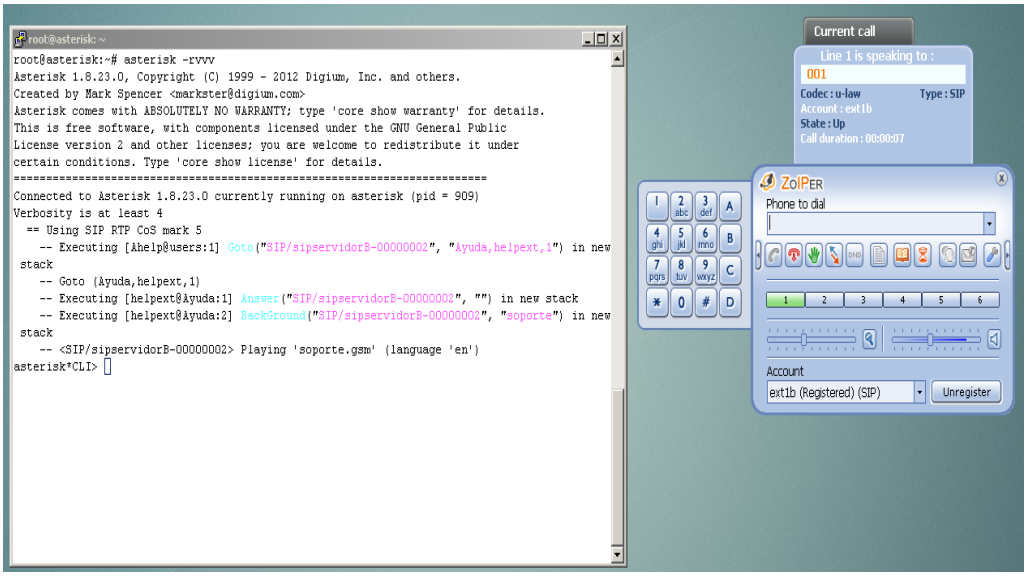


Fig. 5.28 Llamada mostrando Playing

La figura 5.28 muestra una llamada desde una extensión conectada al servidor B; se ha marcado 001 y tal como se describió en el archivo *extensions.conf* la consola de asterisk muestra un *Playing soporte.gsm* siendo éste el mensaje utilizado para el IVR.

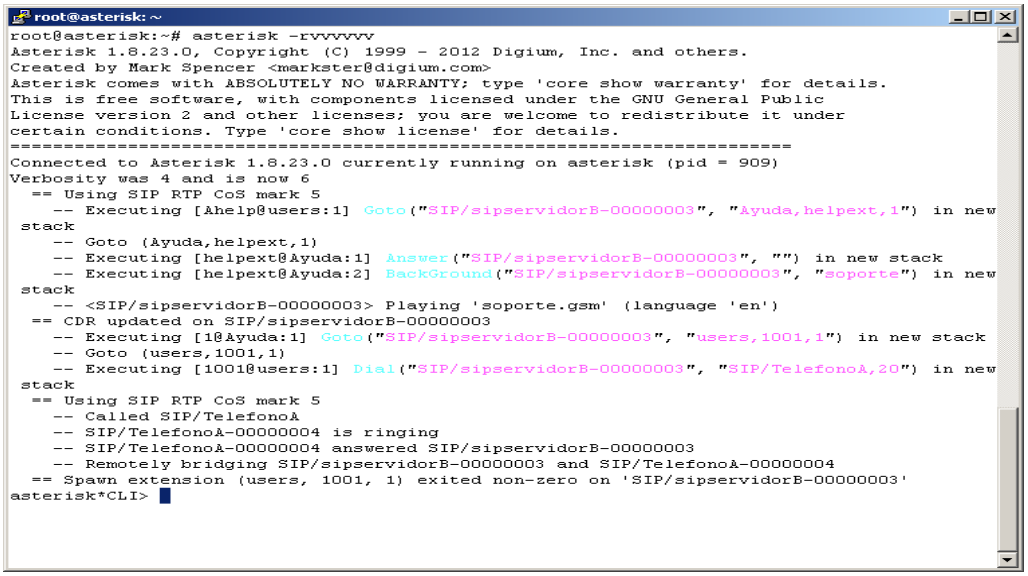


Fig.5.29 Proceso completo visto desde Asterisk

En la figura 5.29 se muestra el proceso completo de este servicio; observamos que después de la sentencia *Playing* se ejecuta 1@Ayuda:1. Esto indica que hemos presionado el número 1 en nuestro dispositivo, por lo que deberá sonar el TeléfonoA que corresponde a la extensión 1001; en cada uno de los archivos *extensions.conf* de cada servidor se encuentran detallados los pasos de este proceso.

Pruebas SS7

Una vez realizadas las configuraciones en los archivos necesarios para el enlace SS7, procedemos a verificar el establecimiento de una llamada a través de esta señalización.

Para comprobar el estado del enlace vamos a la consola de Asterisk y digitamos:

Servidor A

asterisk*CLI> ss7 status

Linkset idle busy initiating resetting total incoming total outgoing

Siuc30 0 0 0 0 0

Servidor B

asteriskB*CLI> ss7 status

Linkset idle busy initiating resetting total incoming total outgoing

Siuc 30 0 0 0 0 0

Si en el parámetro *idle* aparece el numero 30 significa que estamos listos para utilizar el enlace por medio de SS7; caso contrario aparecerá lo siguiente.

Servidor A

asterisk*CLI> ss7 status

Linkset idle busy initiating resetting total incoming total outgoing

Siuc 0 0 0 30 0 0

Servidor B

asteriskB*CLI> ss7 status

Linkset idle busy initiating resetting total incoming total outgoing

Siuc 0 0 0 30 0 0

En este caso tendremos que revisar cual es la causa del problema, pudiendo ser estas:

- Tarjetas insertadas de forma incorrecta en ranura PCI.
- Cable utilizado para el enlace en mal estado; recordar el tipo de configuración del cable debe ser Cruzado T1/E1.
- Falla en las ranuras PCI de los servidores.
- Error en la configuración de los archivos necesarios para ss7.

Como podemos observar la figura 5.30 muestra el plan de marcado a utilizar para la prueba ss7, en el servidor A encontramos que al digitar 777 se procede a realizar una llamada a la extensión 5000 del servidor B utilizando SS7; en el servidor B tenemos un *playback* de un archivo de grabación que viene incluido en asterisk por default.

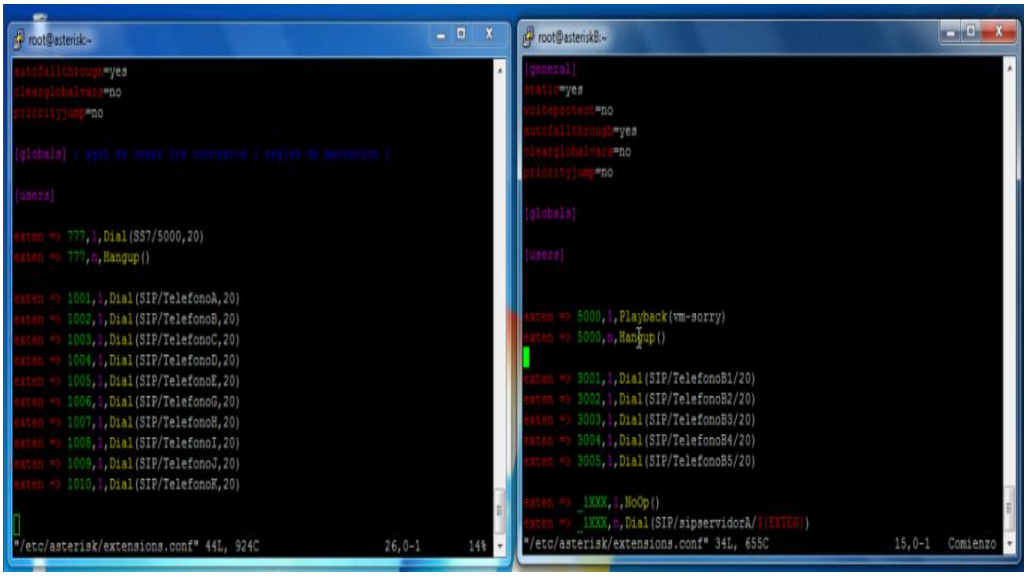


Fig. 5.30 Plan de marcado para SS7

Un proceso completo de la llamada, a través de SS7 (ver figura 5.30), muestra que en el Servidor A (terminal ssh inferior), cuando se digita 777, de inmediato procede a llamar a la extensión 5000 por SS7; en el servidor B (terminal ssh superior) se ejecuta el *Playback* de la grabación configurada en *extensión.conf* y luego se produce el *Hangup* respectivo.

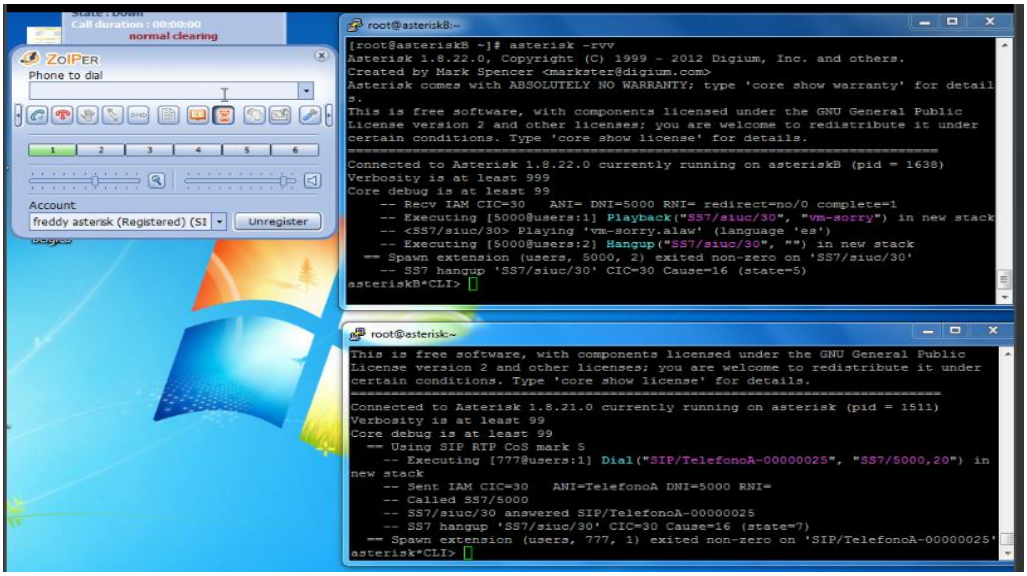


Fig. 5.31 Proceso de llamada en SS7

Mediante estas dos imágenes (ver figura 5.31), se comprueba una correcta señalización ss7 usando los dos servidores Asterisk.Cabe recalcar que solo se usa Ethernet para la transmisión de paquetes, entre el Servidor A y la cuenta que está llamando al Servidor B. El envío y recepción de paquetes pertenecientes a la llamada se realizan por medio de ambas tarjetas E1/T1 configuradas en cada servidor; es por este motivo que no se puede obtener una captura de paquetes por medio de *Wireshark*.

A continuación se procederán a realizar 4 escenarios para comprobar la versatilidad y funcionalidad de asterisk en este proyecto.

- Escenario 1

En este escenario se plantea realizar 5 llamadas simultáneas por medio de SIP entre ambos servidores.

En la siguiente figura 5.32 se muestra el registro de 5 cuentas SIP cuya información se encuentra en el Servidor A; observamos que todas tienen el estado *Registered*, lo cual indica que ya están asociadas al servidor A.

Fig. 5.32Cuentas SIP registradas enzoiper(servidor A)

Así mismo tenemos la figura 5.33 de las cuentas registradas en el servidor B; como vemos también muestran el estado *Registered* que significa una correcta asociación con el servidor.

El estado *Registered* indica que se han configurado correctamente los parámetros: *IP de Servidor*, *User ID* y *Password*. En caso de no mostrar

dicho estado, deben verificarse estos parámetros y así mismo comprobar si existe una conexión con servidor.

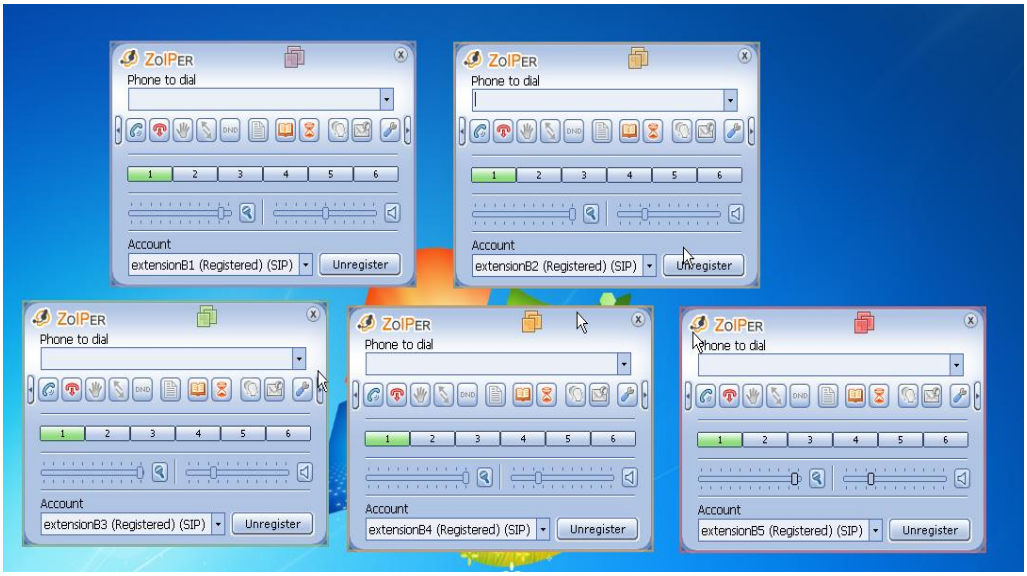


Fig. 5.33Cuentas SIP registradas enzoiper(servidor B)

En las siguientesimágenes se muestran las cuentas registradas, vistas desde la consola de asterisk, observamos un *OK* junto con un tiempo en milisegundos (ver fig. 5.34 - 5.35), esta información la obtenemos gracias al parámetro *qualify*, configurado en cada una de las cuentas del archivo *sip.conf* de cada servidor.

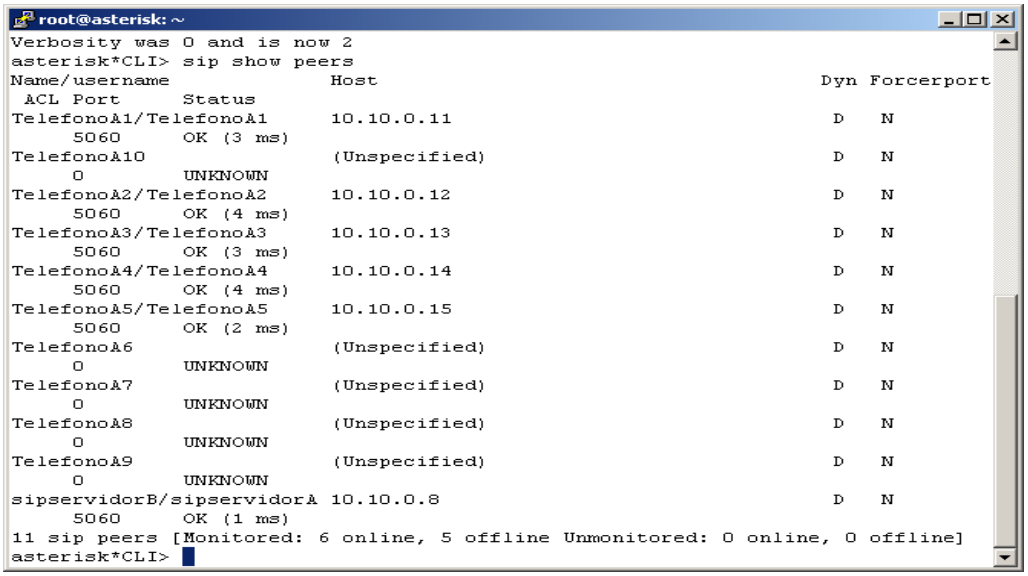


Fig. 5.34 Registro de extensiones servidor A

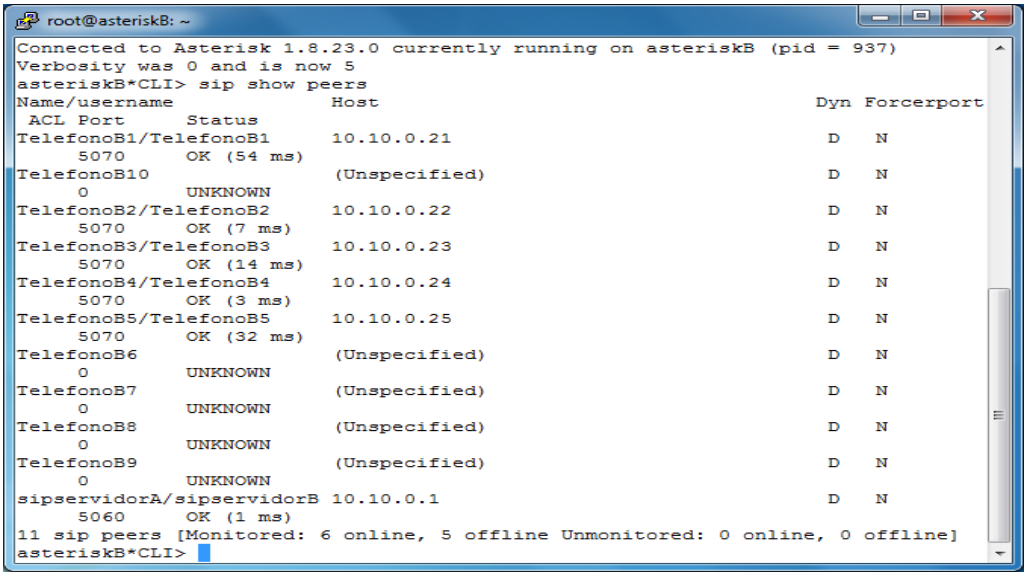


Fig. 5.35 Registro de extensiones servidor B

Vemos en las figuras (5.36 – 5.37) como se encuentran enlazados ambos servidores tanto por SIP y como IAX, esta configuración no afecta al rendimiento entre las llamadas de los usuarios.

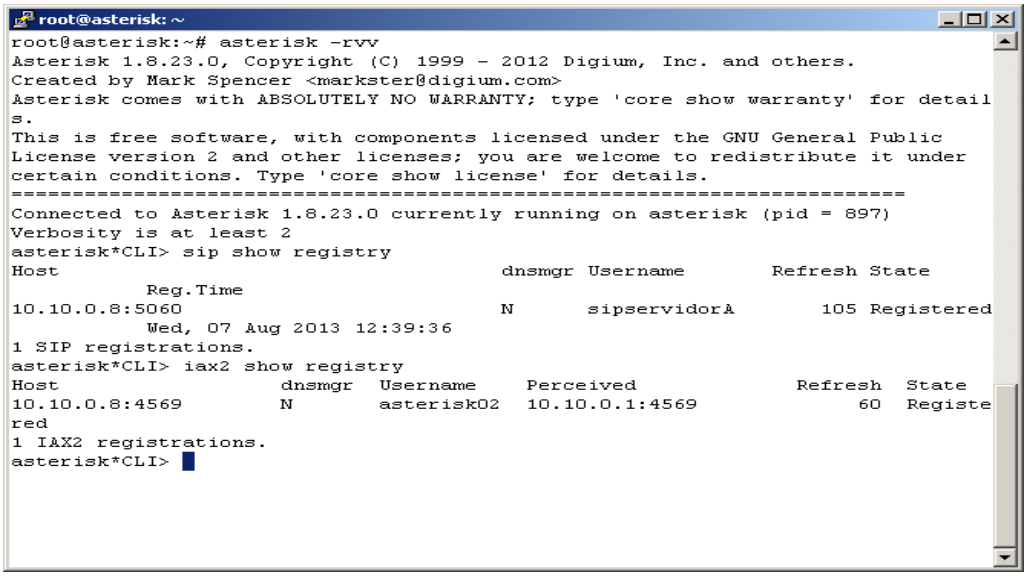


Fig. 5.36 Servidor A enlazado con SIP e IAX

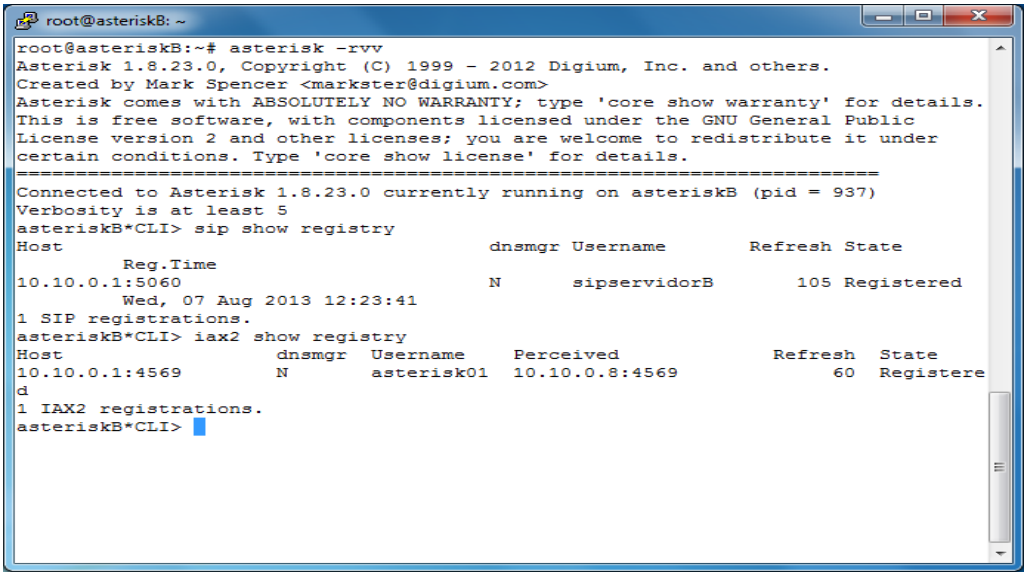


Fig. 5.37 Servidor B enlazado con SIP e IAX

En las figuras (5.38 – 5.39) se muestran el proceso de llamada.Vemos en cada softphone los siguientes mensajes:

Wait for answer: En espera de una contestación por parte del usuario al que se llama.

Codec: Tipo de códec de audio a utilizar.

Fig. 5.38 Proceso de llamada SIP (servidor A)

Codec GSM: Es el *codec* que va a entregar la cuenta al momento de responder la llamada.

Type SIP

Status RINGING: Existe un usuario que necesita comunicarse con dicha cuenta por lo cual se muestra un tono de timbrado.

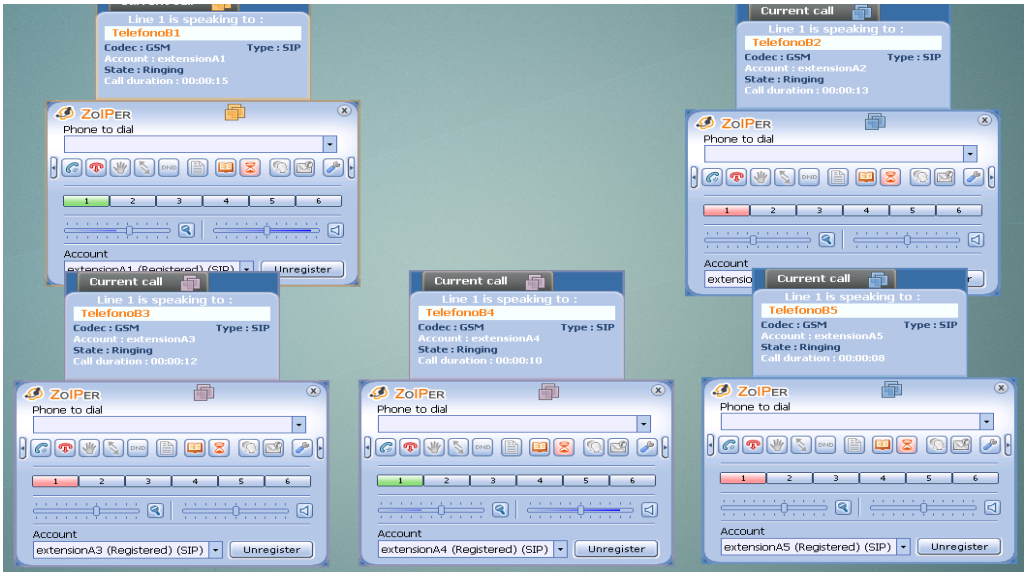


Fig. 5.39 Proceso de llamada SIP, RINGING(servidor A)

Las siguientes imágenes (ver figuras 5.40 – 5.41) muestran el mismo proceso, pero esta vez se realizan las llamadas desde el servidor B hacia el servidor A.

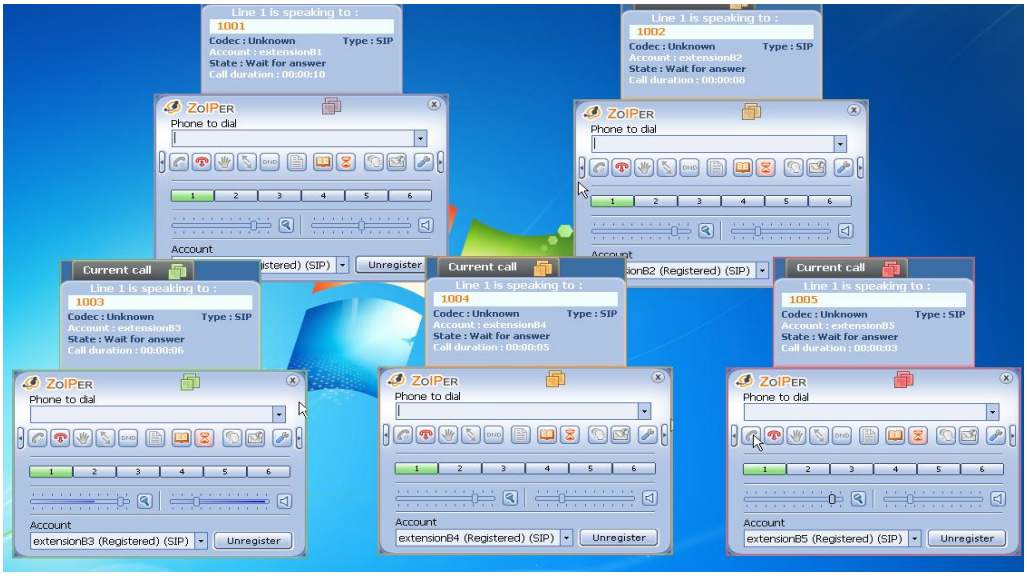


Fig. 5.40 Proceso de llamada (servidor B)

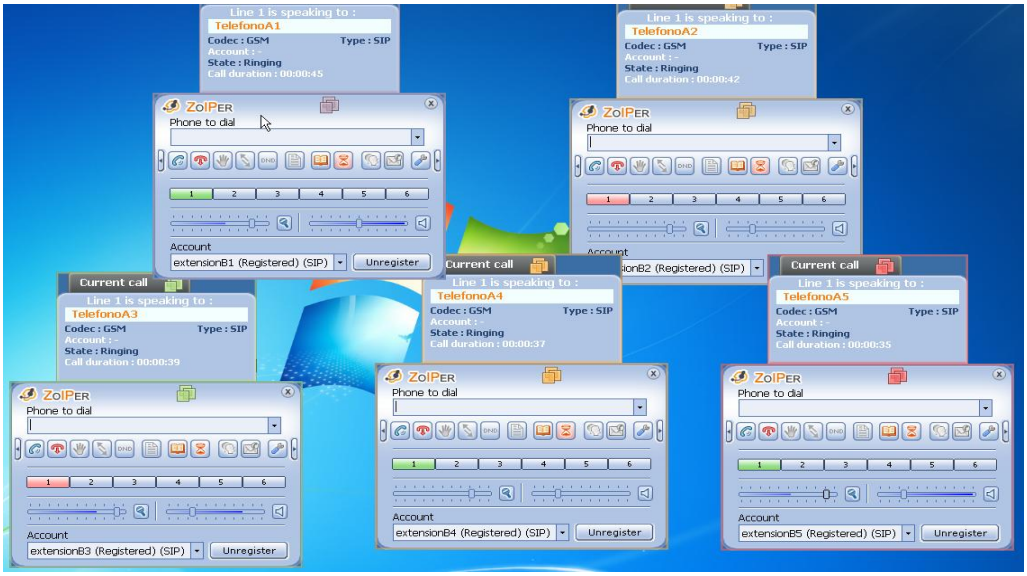


Fig. 5.41 Proceso de llamada, RINGING (servidor B)

Codec GSM: *Codec* de audio con el cual está siendo ejecutada la llamada.

Status UP: Significa que hay una llamada en progreso.

Call duration: Tiempo de duración de la llamada.

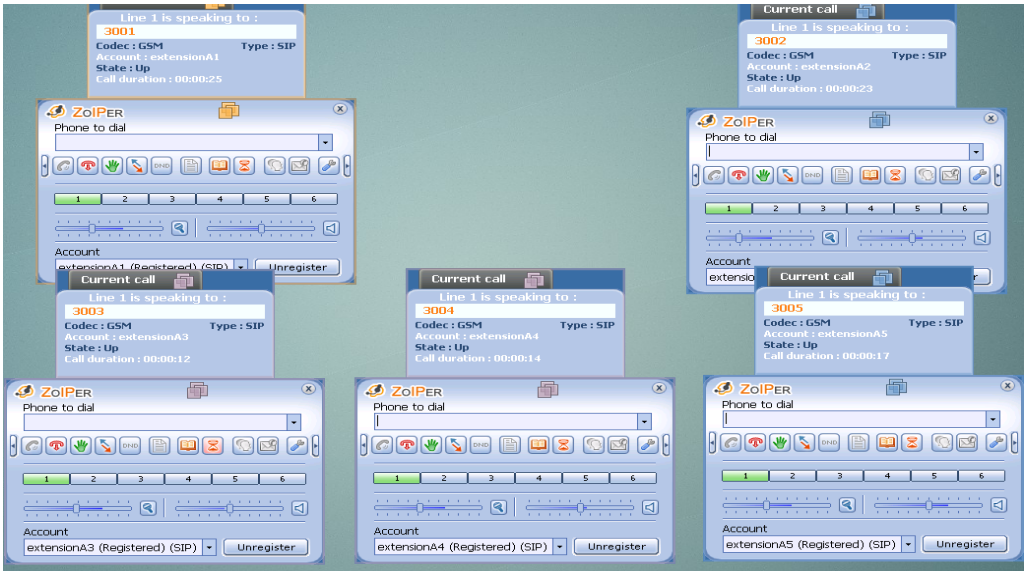


Fig. 5.42 Cinco llamadas simultáneas (servidor A)

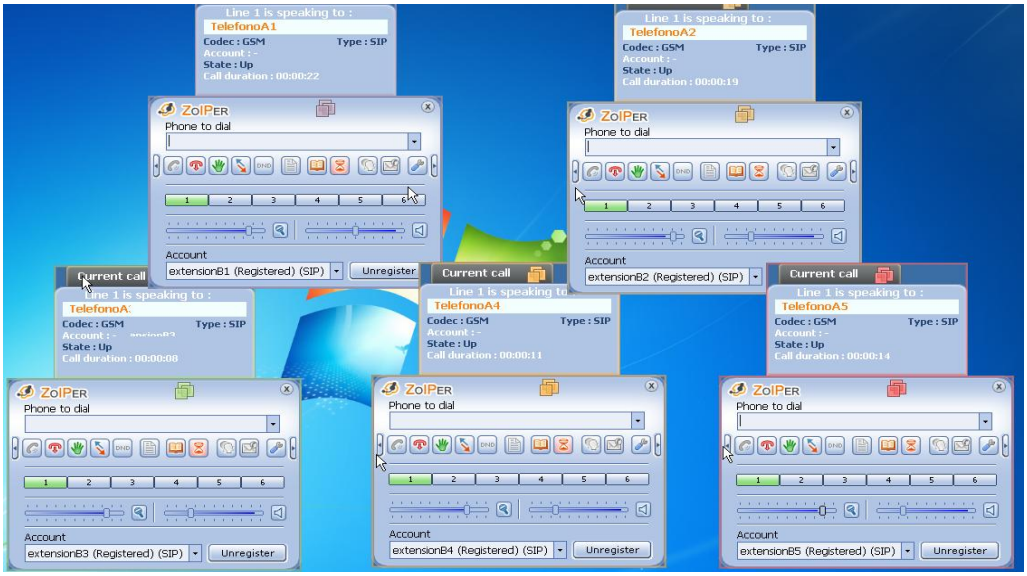


Fig. 5.43 Cinco llamadas simultáneas (servidor B)

Como hemos visto en las imágenes anteriores(ver Fig. 5.42 – 5.43), las 5 llamadas se ejecutan satisfactoriamente por medio del protocolo SIP sin ningún inconveniente.

Además, este tipo de llamadas genera poca carga al servidor, cumpliendo con los bajos requerimientos que necesita Asterisk al momento de realizar llamadas SIP (ver fig. 5.44 – 5.45).

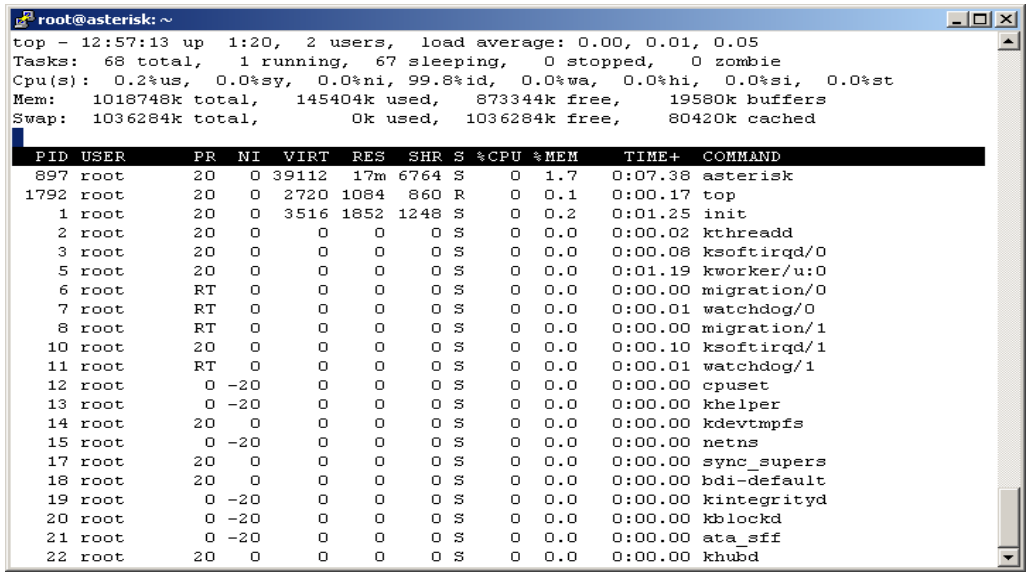


Fig. 5.44 Carga en el servidor A

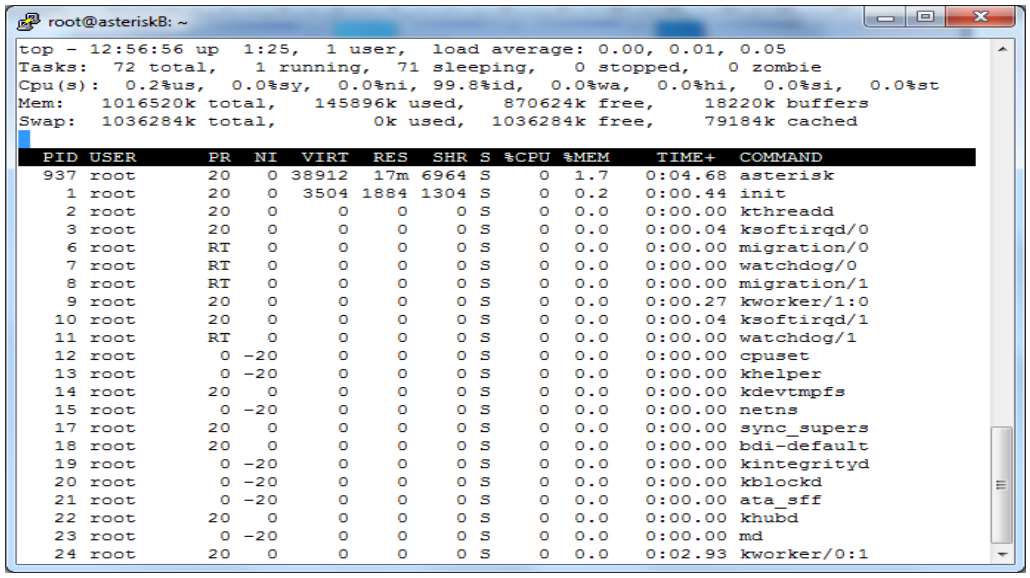


Fig. 5.45 Carga en el servidor B

- Escenario 2

A continuación se realiza el mismo procedimiento que en el escenario 1, pero esta vez con el protocolo IAX2; es decir,se plantea realizar 5 llamadas simultáneas por medio de IAX entre ambos servidores.

Las siguientes imágenes(ver fig. 5.46 – 5.47) muestran las cuentas asociadas, tanto al servidor A como al servidor B desde la consola asterisk.

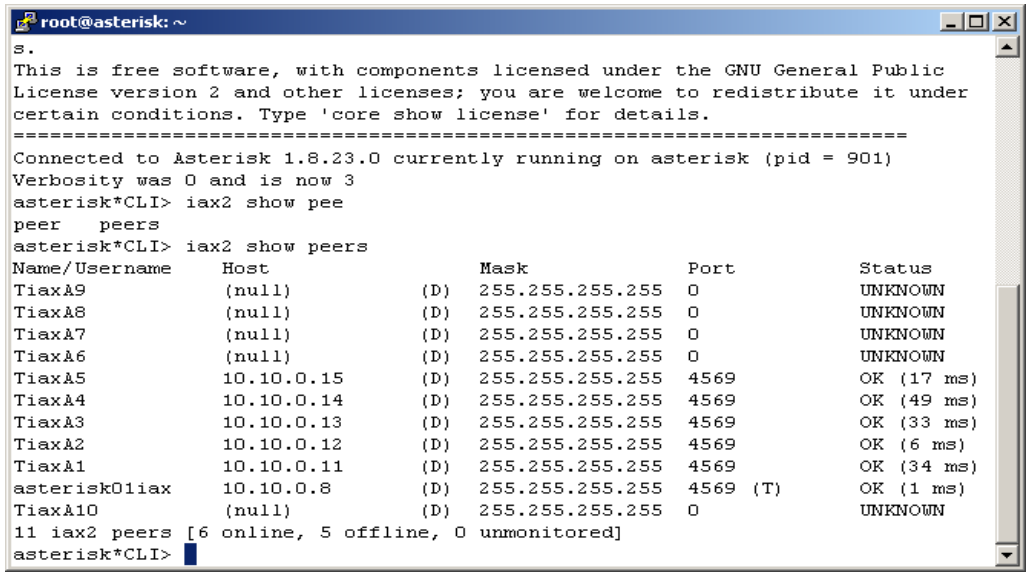


Fig. 5.46 Cuentas registradas en el servidor A

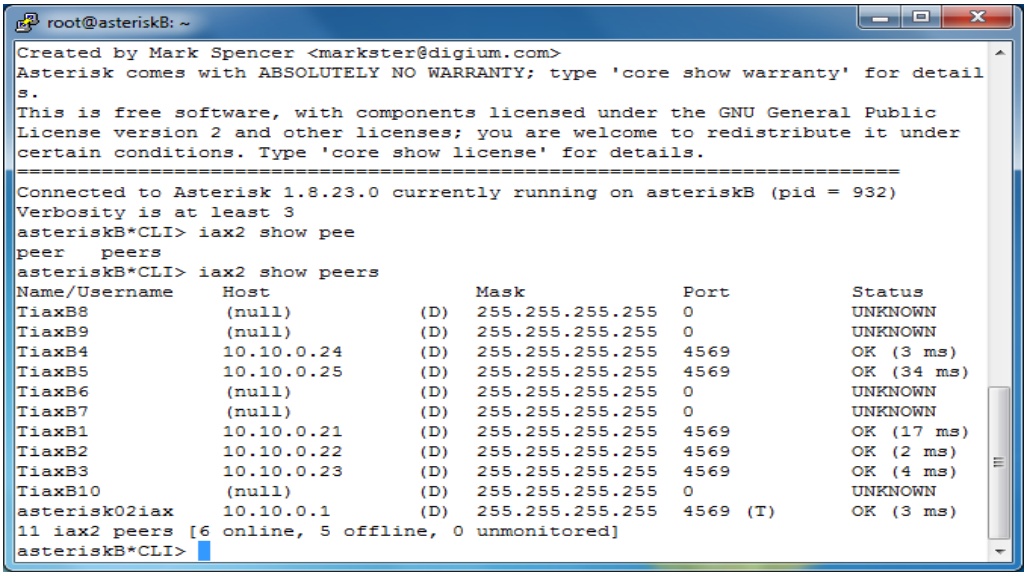


Fig. 5.47 Cuentas registradas en el servidor B

Observamos cómo se encuentran registradas cada una de las cuentas IAX en los softphones (ver fig. 5.48 -5.49). Recordemos que los parámetros de configuración los obtenemos del archivo *iax.conf*.

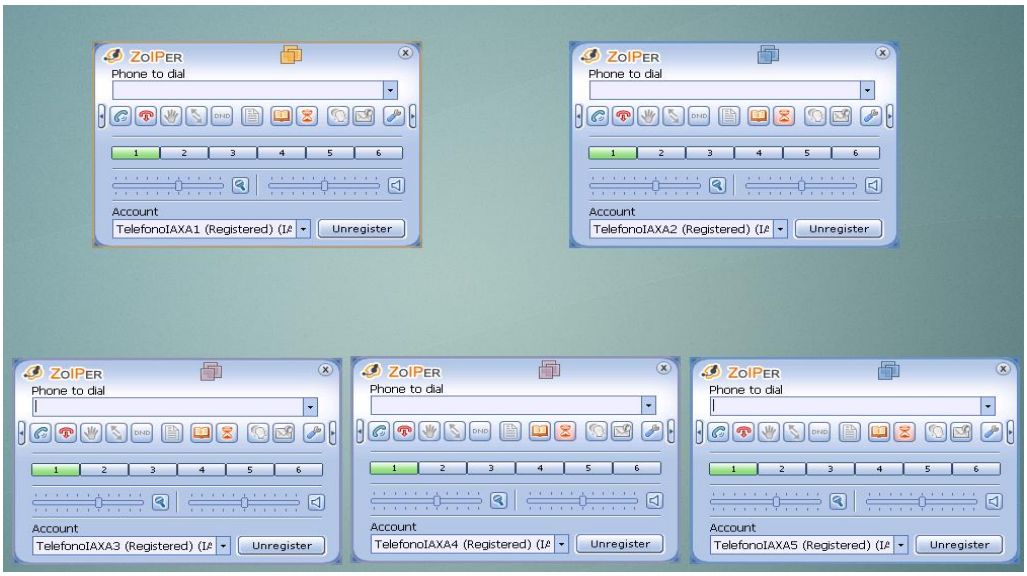


Fig. 5.48. Cuentas IAX del servidor A registradas en softphone

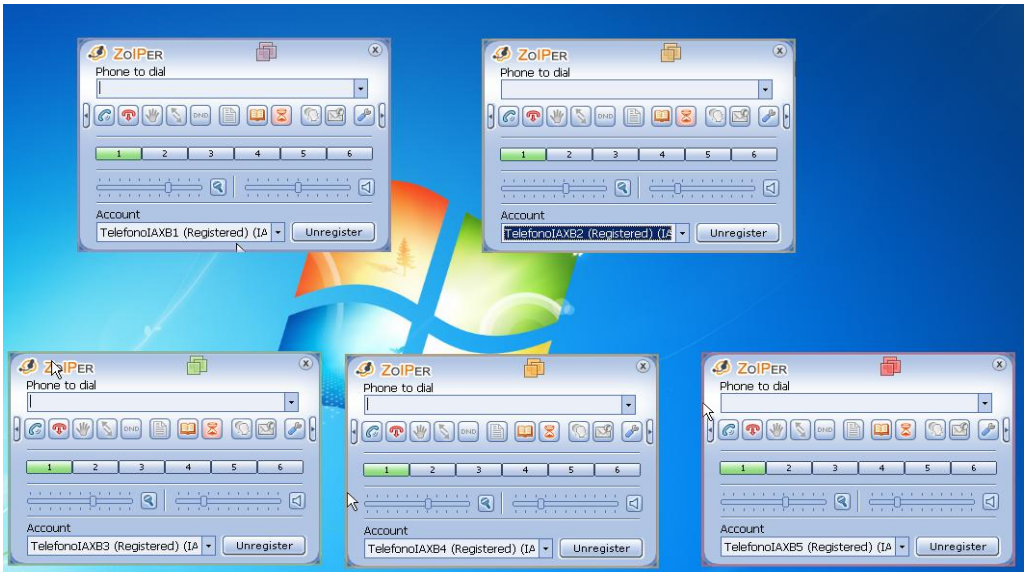


Fig. 5.49 Cuentas IAX del servidor B registradas en softphone

La imagen 5.50muestra como cada una de las cuentas está realizando una llamada hacia las cuentas del servidor B, que son 9001,9002,9003,9004 y 9005 respectivamente.

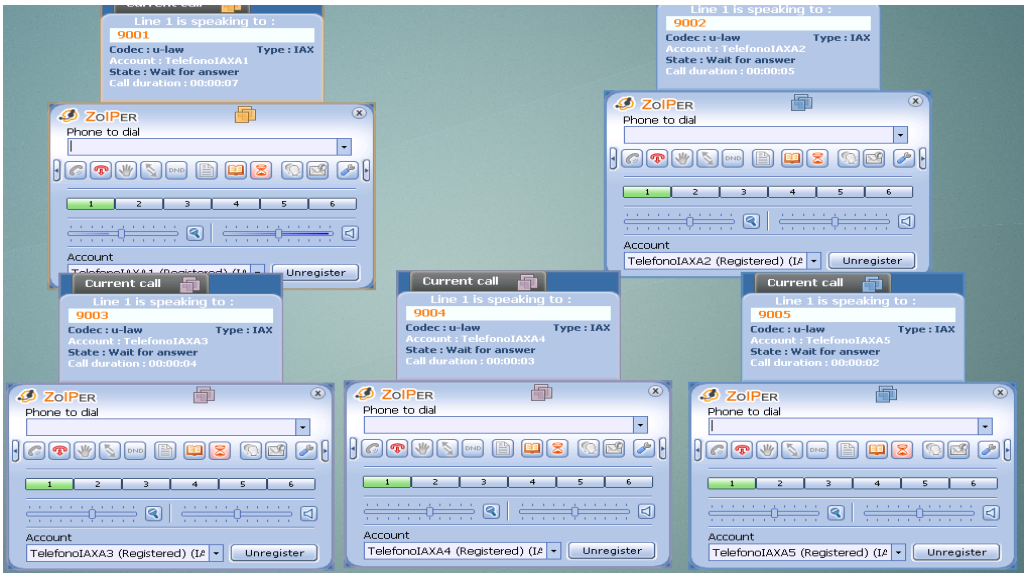


Fig.5.50Llamadas IAX del servidor A al servidor B

La figura 5.51. muestra como empiezan a sonar, *RINGING*, las cuentas en el servidor B.

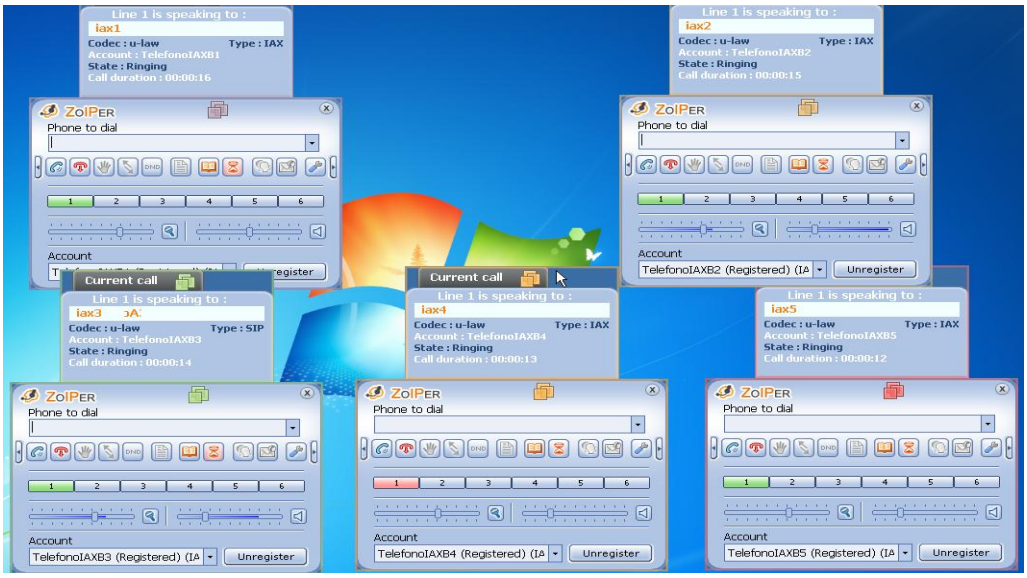


Fig.5.51 Llamadas IAX del servidor A al servidor B, RINGING

Se repite el proceso anterior, pero esta vez realizando llamadas desde el servidor B.En la figuras 5.52, se observa como se generan las llamadas desde el servidor B en espera de una respuesta del servidor A.

Mientras se está en espera de la respuesta; las extensiones del servidor A están con el *RINGING,*como podemos observar en la figura 5.53.

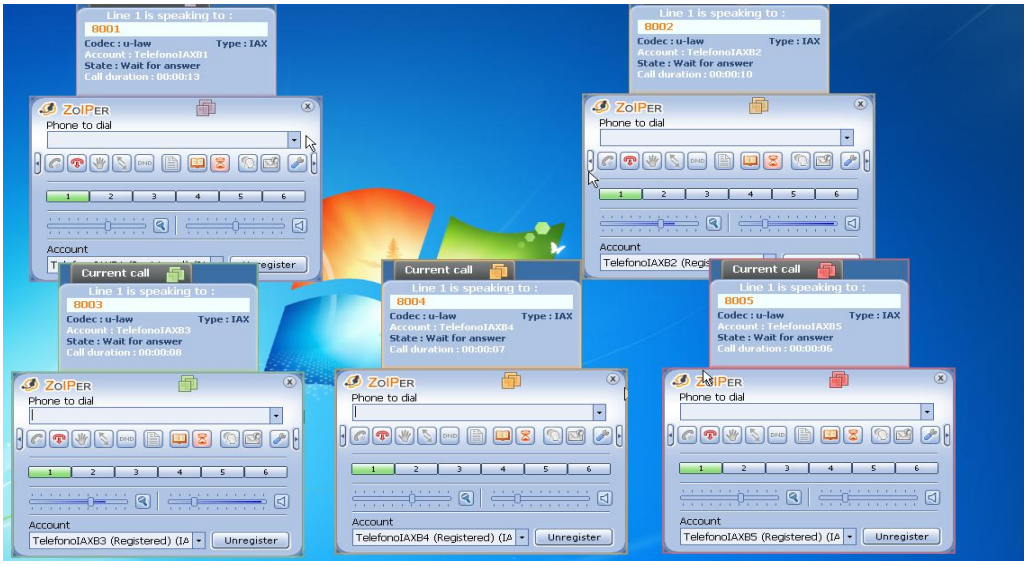


Fig.5.52. Llamadas del servidor B al servidor A

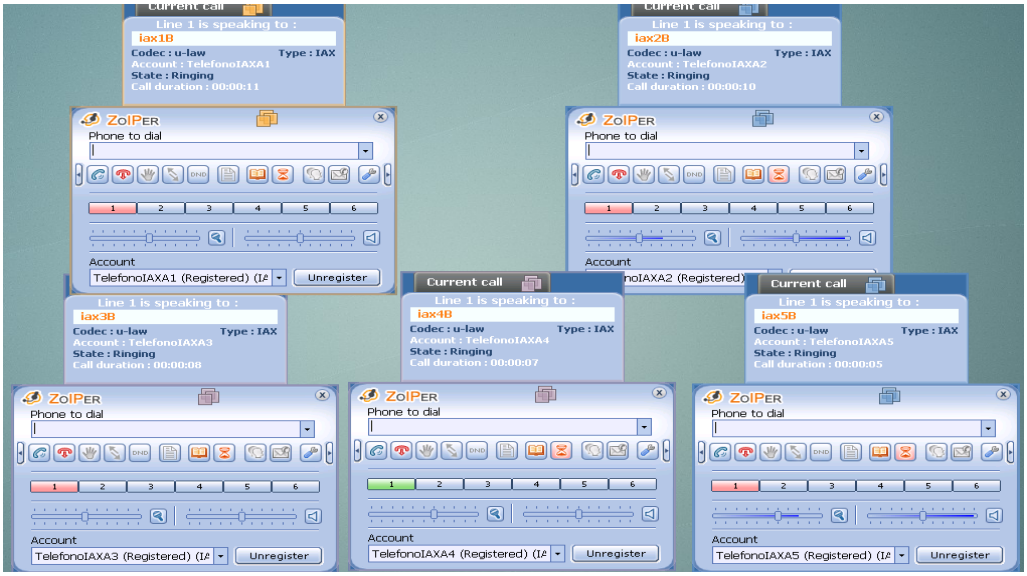


Fig.5.53. Llamadas del servidor B al servidor A,RINRING en servidor A

A continuación se muestra la ejecución de las 5 llamadas que se han realizado satisfactoriamente a través de IAX, entre ambos servidores, servidor A ver fig.5.54, servidor B ver fig. 5.55.

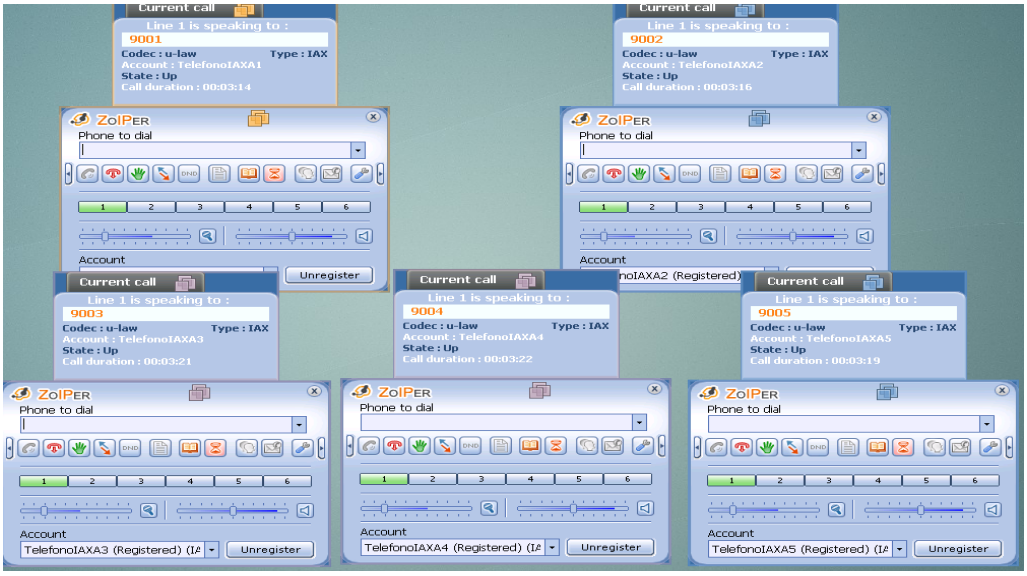


Fig.5.54.Cinco llamadas del servidor B al servidor A (servidor A)

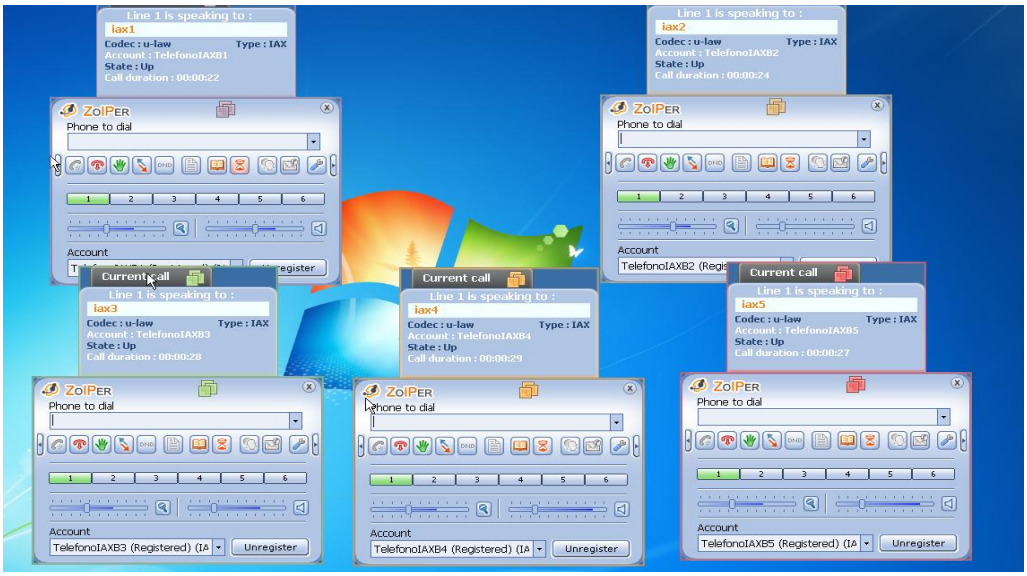


Fig.5.55.Cinco llamadas del servidor B al servidor A (servidor B)

- Escenario 3

Este escenariobusca comprobar la utilidad del servicio planteado en este proyecto, el cual consta de un IVR de ayuda para la ubicación de las extensiones hacia los usuarios o departamentos que correspondan.

En el softphone de la derecha vemos lo siguiente (ver fig. 5.56):

002: Número marcado para escuchar el IVR del Servidor B desde una cuenta registrada en el servidor A.

Codec GSM: Codec utilizado para tocar el mensaje de grabación.

Status UP: Indica que la grabación está siendo ejecutada; en este momento el usuario escucha el IVR.

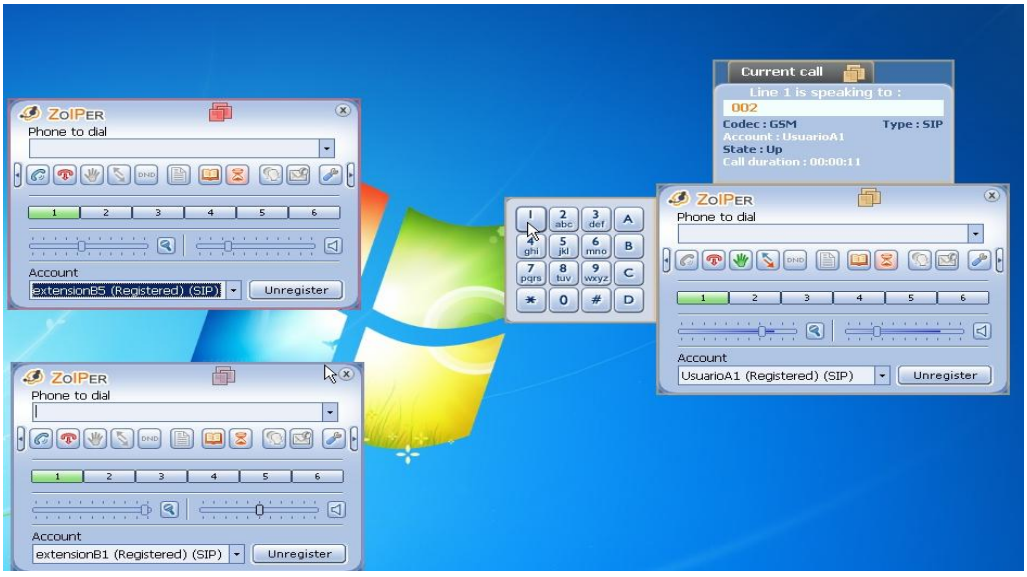


Fig. 5.56Llamada con respuesta del IVR

En la figura 5.57, se observa en el softphone de la derecha, un 1 marcado. Esto indica que el usuario ha marcado el número 1 y por lo tanto debe sonar la extensión asociada a la extensión B1, como vemos en el softphone de la esquina inferior izquierda.

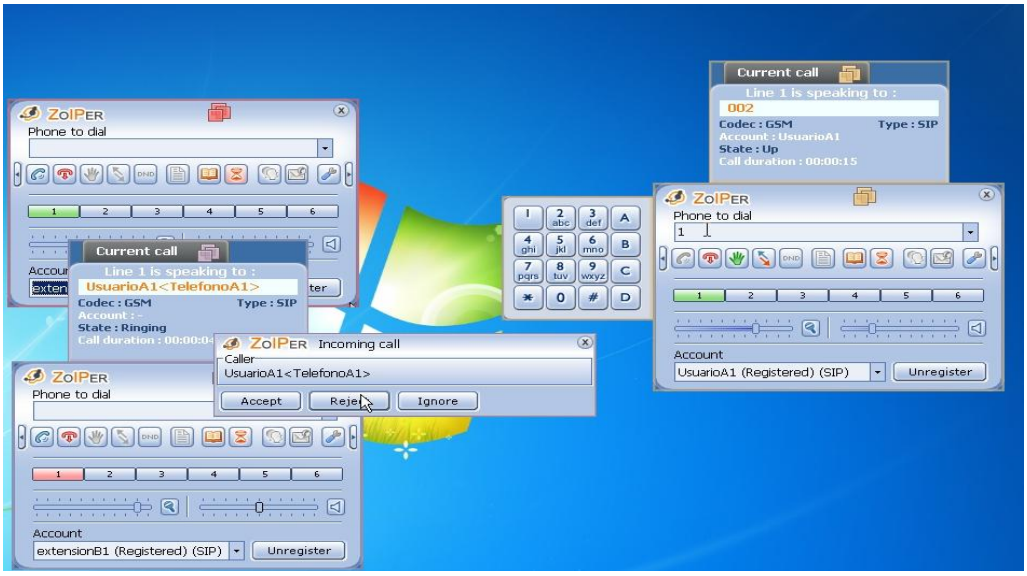


Fig. 5.57Redireccionamiento de llamada

En esta parte el usuario A1 (softphone de la derecha) pide transferencia hacia la extensión B5 (softphone de arriba), mientras está hablando con la extensión B1 (softphone de la izquierda). Este procede a realizar el transfer digitando el número 3005 que pertenece a la extensión B5 (ver fig. 5.58).

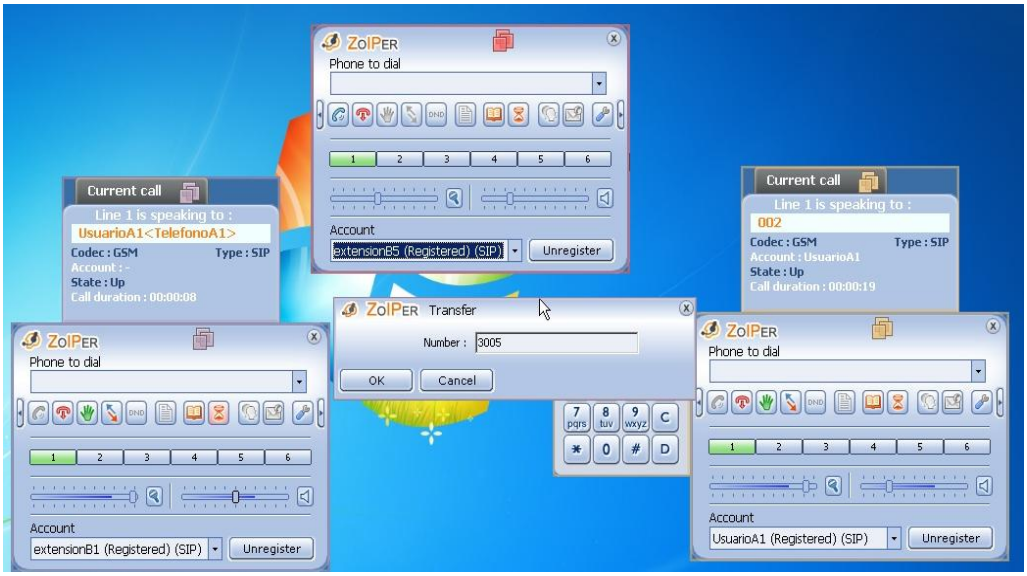


Fig. 5.58Proceso de transferencia de llamada

En la figura 5.59 vemos un *normal clearing* en la extensión B1 lo cual indica que ya no existe llamada con esta cuenta, además observamos un *RINGING*en la extensión B5 producido por el usuario A1 que solicitó transferencia.

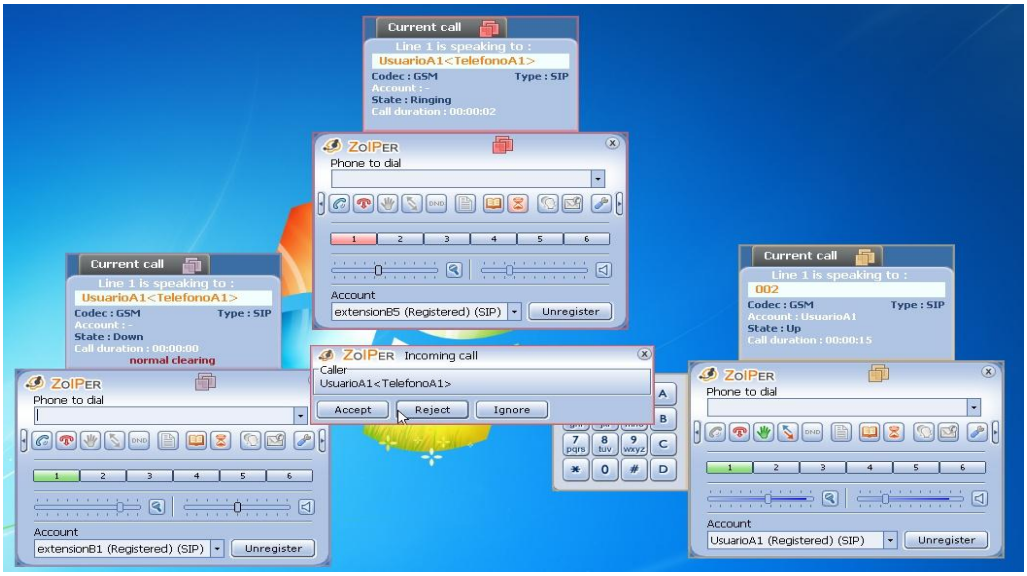


Fig. 5.59Proceso de transferencia de llamada

En la figura 5.60 ya se encuentra establecida la llamada; con esto se cumple la funcionalidad del servicio en este escenario.

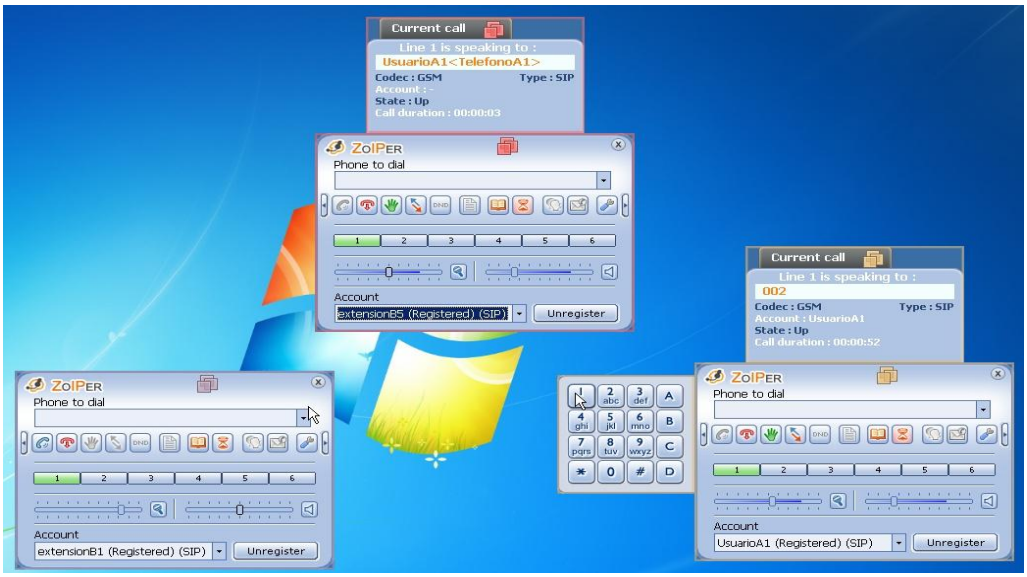


Fig. 5.60 Transferencia de llamada completa

- Escenario 4

Realizar una llamada SIP y otra llamada IAX por medio de SS7 para comprobar la estabilidad del enlace.

En la figura 5.61 vemos el plan de marcado para establecer una llamada ss7 por medio de SIP.En el servidor A tenemos:

Exten => 777,1,Dial(SS7/5000,20)

Exten => 777,n,Hangup()

Y en el servidor B tenemos:

Exten => 5000,1,Dial(ss7sipB/20)

Exten => 5000,n,Hangup()

Las líneas anteriores nos indican que al marcar el número 777 desde una cuenta SIP registrada en el servidor A, deberá contestar la cuenta ss7sipB cuyos parámetros se encuentran en el archivo *sip.conf* del servidor B ; todo este proceso se realiza por SS7.

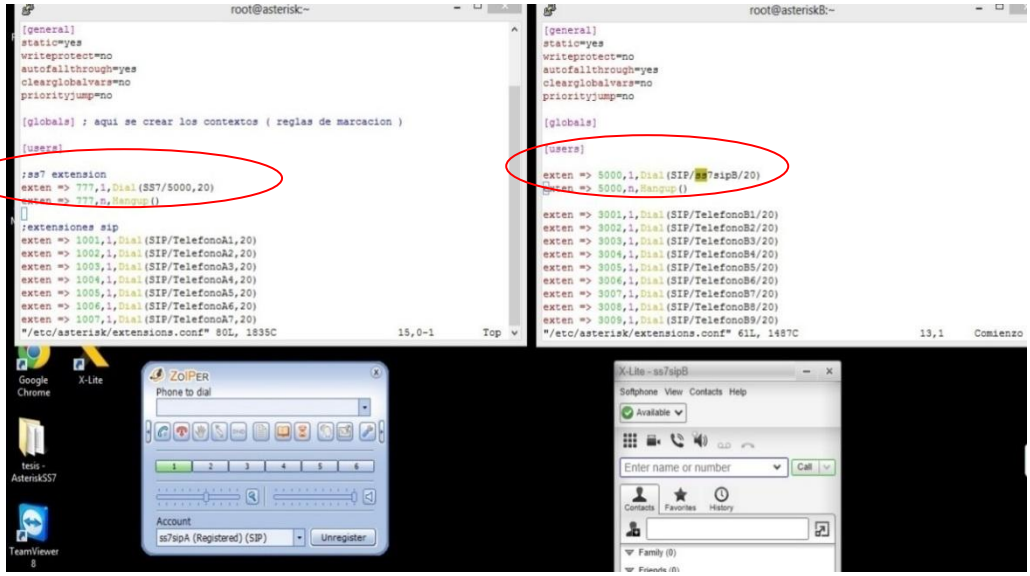


Fig. 5.61 Plan de marcado para llamada ss7 con SIP

Vemos en la consola de asterisk cómo se realiza el proceso de la llamada.En el servidor A observamos los mensajes, CIC,ANI, DNI, RNI que son propios de la aplicación *chan_ss7* para la ejecución de ss7.

Además, observamos el mensaje *RINGING*, lo cual indica que se ha establecido la llamada y se espera una contestación (ver fig. 5.62).

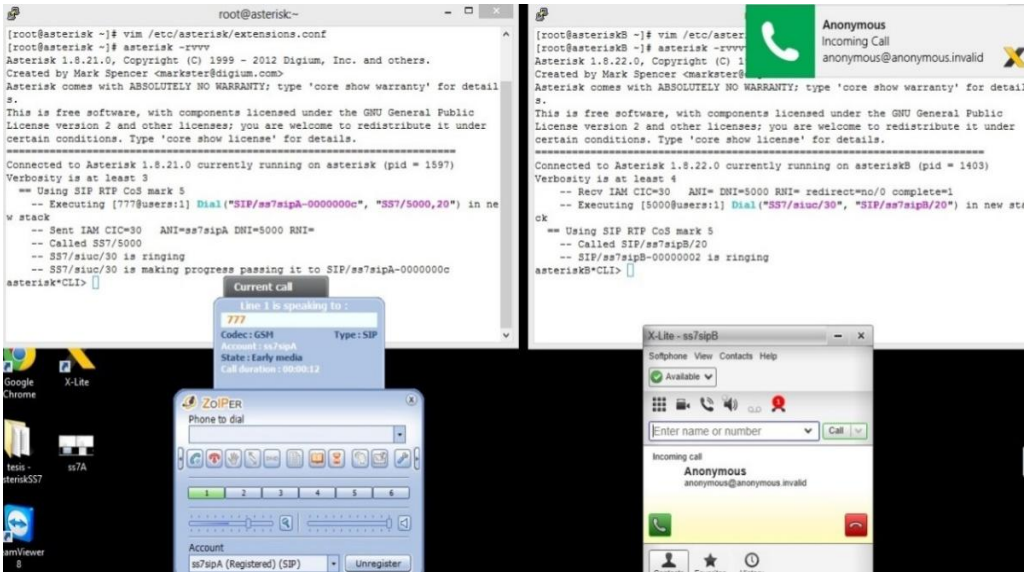


Fig. 5.62 Proceso de llamada ss7 con SIP, RINRING

La figura 5.63 muestra el mensaje *ANSWERED*en la consola de asterisk, además observamos el estatus *UP* en el softphone de la izquierda, esto indica que hay una comunicación o llamada establecida en ese momento.

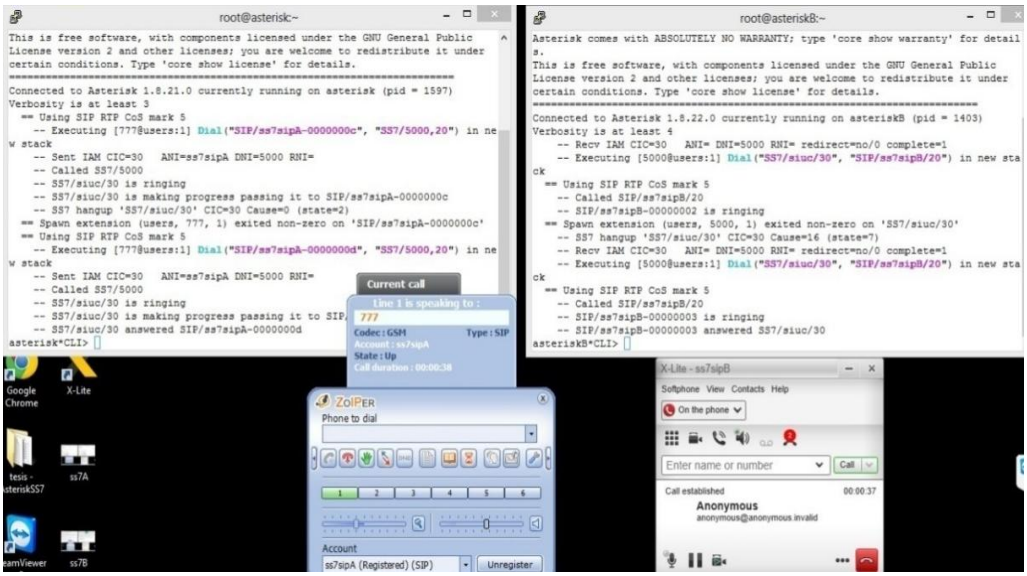


Fig. 5.63 Proceso de llamada ss7 con SIP, llamada establecida

A continuación realizaremos el mismo procedimiento pero para una llamada entre cuentas IAX.

Observamos el plan de marcado a continuación

Servidor A

Exten => 778,1,Dial(SS7/5100,20)

Exten => 778,n,Hangup()

Servidor B

Exten => 5100,1,Dial(IAX2,ss7sipB/20)

Exten => 5100,n,Hangup()

Esta configuración indica que al marcar el numero 778 deberá contestar la cuenta *ss7sipB* cuyos parámetros se encuentran en el archivo *iax.conf* del servidorB (ver fig. 5.64).

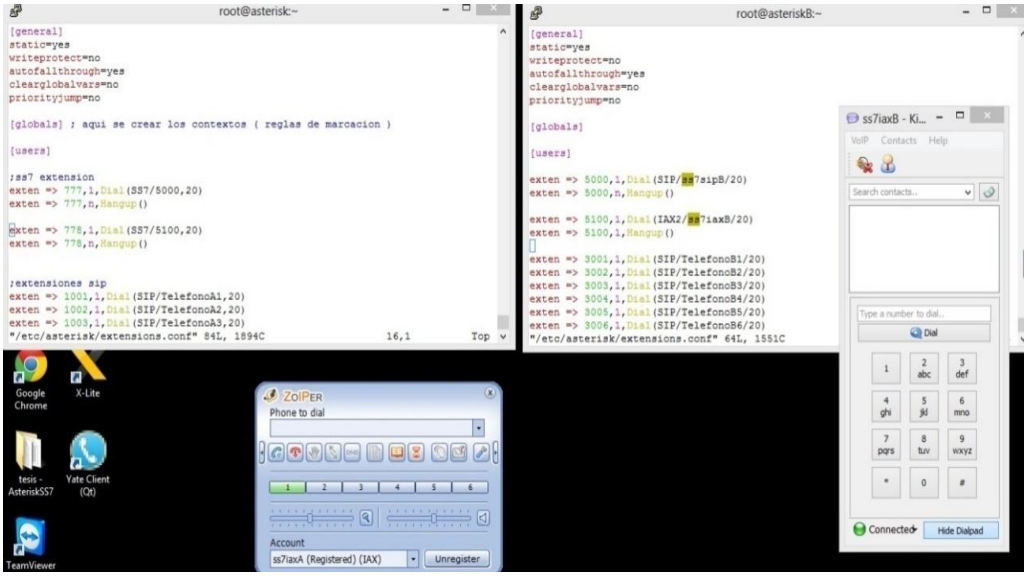


Fig. 5.64 Plan de marcado para llamada ss7 con IAX

Al igual que en la llamada SIP por medio de SS7, en la figura 5.65 vemos los mensajes que muestra la consola de asterisk para el establecimiento de la comunicación.

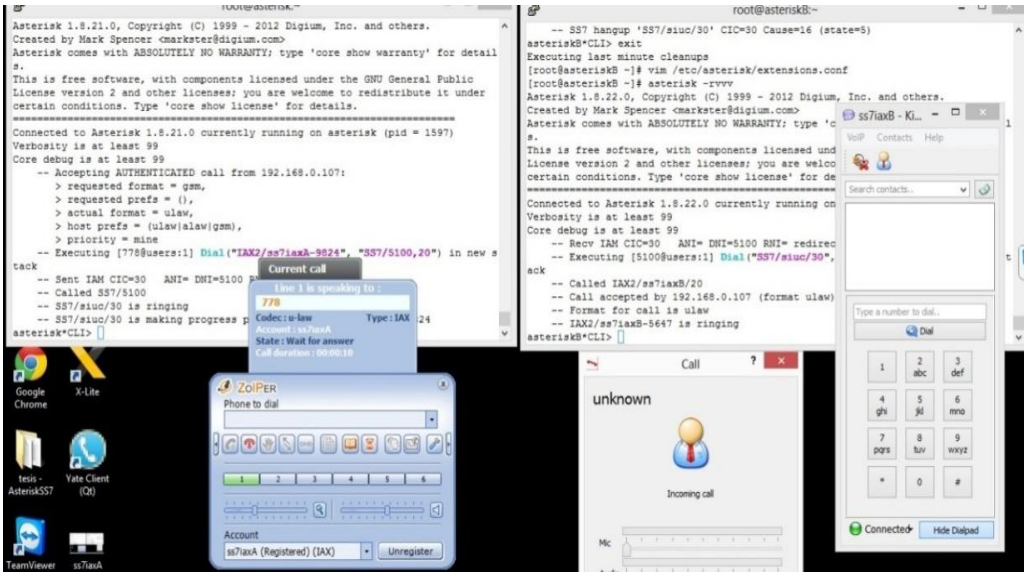


Fig. 5.65 Proceso de llamada ss7 con IAX

A continuación se muestra la llamada establecida, verificada por el *estatus UP;* observamos en el softphone de la izquierda que el tipo de llamada es IAX (ver fig.5.66).

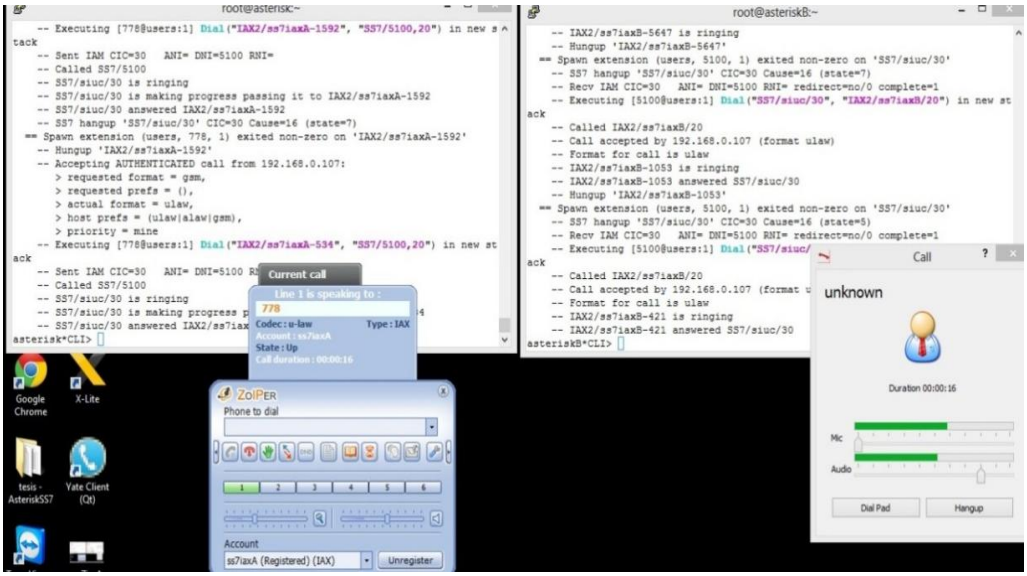


Fig. 5.66 Proceso de llamada ss7 con IAX, llamada establecida

La consola de asterisk nos muestra el mensaje *Hangup*, esto significa que ha concluido satisfactoriamente la llamada como se puede observar en la figura 5.67.

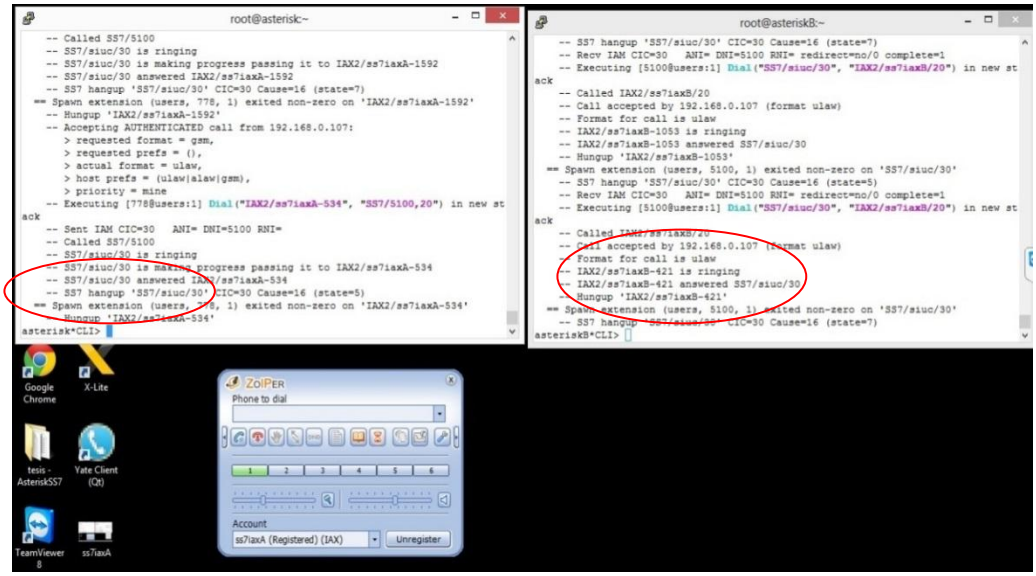


Fig. 5.67Mensaje de culminación de llamada en SS7

Con esto comprobamos la funcionalidad de las llamadas SIP e IAX por medio de SS7; como vemos, se pueden realizar sin ningún inconveniente y con una buena calidad.

5.4 Tabla Comparativa entre Protocolos

	SIP	IAX
NAT	La señalización y los datos se transportan independientemente en SIP, lo que hace que se produzcan problemas de *NAT* en el flujo de audio, cuando éste superar	La señalización y los datos viajan juntos en IAX y esto permite evitar los problemas de *NAT* que usualmente se crean en SIP.

	los *routers* y firewalls. SIP puede necesitar un servidor STUN para los problemas de NAT.	
ANCHO DE BANDA	Son mensajes de texto cuando se usa SIP.	IAX procura disminuir lo más posible la información de las cabeceras de los mensajes, y al mismo tiempo disminuyendo el ancho de banda.
ESTANDARIZACIÓN	Hace muchos años que SIP es un protocolo estandarizado por IETF y lo implementan todos los fabricantes de equipos y software.	Aún no está estandarizado el protocolo IAX debido a que fue creado para resolver los problemas de NAT y por ello que no es utilizado en muchos dispositivos que se encuentran en el mercado.
PUERTOS	SIP usa un puerto (5060) para señalización y dos RTP por cada conexión de audio	Para el envió de la información de señalización y datos de llamada IAX usa un puerto (4569). Por

	(tres puertos minimo).	este motivo usa un mecanismo de multiplexión o t*runking.*
FLUJO DE AUDIO AL USAR UN SERVIDOR	La señalización de control en SIP pasa siempre por el servidor; sin embargo, la información de audio (flujo RTP) puede transportarse de un extremo a otro sin pasar por servidor SIP.	Todo el tráfico de audio debe pasar obligatoriamente por el servidor IAX, al pasar la señalización y los datos conjuntamente, y esto ocasiona un incremento en el ancho de banda que los servidores IAX deben soportar, más aun cuando hay muchas llamadas simultáneas.
FUNCIONALIDADES	Cualquier información podría ser transmitía por SIP ya que es un protocolo de propósito general, pudiendo cualquier información, audio o vido, ser transmitida sin dificultad.	IAX es un protocolo creado para telefonía IP y transmisión de video. Puede presentar funcionalidades interesantes como la de enviar o recibir planes de marcado

		(*dialplans*) que son muy útiles al usarlo junto con los servidores asterisk.

5.5 Guía de Elección en Base a Costes y Eficiencia

Para realizar una toma de decisión correcta, al momento de querer implementar una central telefónica, haremos una comparación entre los protocolos SIP, IAX2 y SS7 en base a varias características de estos protocolos, para que de acuerdo a esto el usuario tenga un conocimiento más claro al momento de implementar una central, según sus necesidades.

Las principales diferencias ente IAX2 y SIP son las siguientes: El protocolo IAX2 maneja un sólo puerto de comunicaciones, el UDP 4569, esto quiere decir que tanto la señalización y el flujo de audio RTP de todas las llamadas viajan multiplexadas por el mismo canal, haciéndolo un protocolo casi transparente para los firewalls. Por otra parte, el protocolo SIP utiliza 3 puertos de comunicaciones :uno para señalización 5060 y dos para el flujo de audio RTP, haciendo que este protocolo necesite de procesos adicionales para resolver los problemas de *NAT*, en algunos casos con la utilización de un servidor *STUN*.

IAX2 reduce el ancho de banda utilizado en una llamada, al codificar los mensajes de señalización de forma binaria, así mismo IAX2 también intenta reducir la cantidad de información requerida para la señalización.

SIP es un protocolo que ya es un estándar IETF, por eso es muy común encontrar hardware y dispositivos que manejan este protocolo. En cambio, IAX2 aún está a la espera de su estandarización y en la actualidad se cuenta con muy pocos dispositivos que lo utilicen.

Aunque aparentemente el protocolo IAX2 hace que la señalización y la voz viajen por el mismo canal y sea una ventaja en los procesos de NAT, elimina la posibilidad de utilizar servidores proxy para las llamadas, y esto lleva a que el flujo de audio RTP tenga que pasar obligatoriamente por el servidor, incrementando notablemente los costos por transferencia de datos del servidor.

La utilización de un servidor Proxy SIP, en un escenario VoIP, nos da la posibilidad de manejar miles de llamadas sin tener la carga del flujo de datos RTP de la llamada, ya que el audio viaja de un punto a otro sin tener que pasar por el servidor, de ahí la ventaja de mantener ambos tipos de datos (señalización y voz) separados.

En conclusión, ninguno de los dos protocolos es mejor o peor que el otro, simplemente tienen usos en aplicaciones y situaciones diferentes; por ejemplo, si se requiere de una estructura donde los problemas de *NAT* son críticos se puede optar por utilizar IAX2, pero si por el contrario se requiere de un escenario donde haya que manejar muchas llamadas, y se quiera disminuir los costos por transferencia de datos, utilizando SIP y un servidor Proxy, sería una mejor alternativa.

Existen infinidad de aplicaciones en las que hay que definir qué protocolo es el más adecuado, pero con la práctica se lo podrá resolver cada vez más rápido.

Por otra parte, tenemos las ventajas y características importantes del SS7, las cuales son muy superiores a los otros dos protocolos antes comparados. SS7 tiene alta capacidad, es decir con un solo enlace de señalización soporta cientos de troncales. Además, muestra una alta velocidad, puede establecer una llamada a través de varias centrales en menos de un segundo. Los enlaces de datos son de 56 Kbps nacional y de 64 Kbps internacionales. Presenta un ahorro, ya que puede ser usado por un amplio rango de servicios de telecomunicaciones y requiere menos hardware que los sistemas anteriores. El SS7 puede aplicarse a todas las redes de telecomunicaciones nacionales e internacionales, así como en redes de servicios especializados (RSE) y en las redes de servicios digitales.

SS7 utiliza líneas separadas a las de la voz de los datos para dar información a la central sobre las llamadas en tránsito. Esto ha proporcionado muchas ventajas, una de ellas, es la optimización de la utilización del tiempo en las líneas existentes actualmente para voz; esto es debido a que la conmutación de las líneas se produce al momento de completarse la llamada. Además, como SS7 se ejecuta sobre líneas digitales y sistemas computarizados, entonces la optimización del proceso es mayor; teniendo la posibilidad de ofrecer otros servicios, tales como por ejemplo: acceso a bases de datos, llamado de vuelta automático, identificación del número de llamadas y muchos otros.

SS7 utiliza un canal de señalización común para todos los circuitos (en vez de tener señalización asociada a cada canal de tráfico), en caso, que se tengan uno o más enlaces de señalización, que pueden tener la posibilidad de utilizar diferentes rutas que permitan tener mayor flexibilidad y seguridad. Un enlace puede manejar cientos de circuitos. En éste las señales de control se transmiten en canales dedicados exclusivamente a señalización, agrupando la señal de varios canales de voz y permitiendo que los conmutadores permitan intercambiar mensajes, en vez de secuencia de tonos y pudiendo operar bajo dos modalidades: canal asociado y canal no asociado.

A medida que nos movemos hacia la convergencia entre la red telefónica de conmutación de circuitos y el mundo IP de conmutación de paquetes, SS7 ha sido importante para los desarrolladores que buscan integrar los dos mundos y aprovechar lo mejor de ambos.

Por estas características y ventajas en la redes podemos concluir que el protocolo SS7 tiene superioridad ante los otros dos protocolos, con el único requerimiento que se tenga conocimiento suficiente para su configuración y manejo al momento de la implementación.

En esta guía se encuentran tres protocolos muy buenos para la implementación de una central telefónica; todo de acuerdo a las necesidades que tengamos en varios escenarios.

CONCLUSIONES

1. Los softwares libres proveen de herramientas útiles y adaptables al entorno, gracias a su soporte, continuo crecimiento y mejoramiento sin costo alguno. En la implementación del proyecto logramos el objetivo de crear una guía básica para la instalación y configuración de Asterisk.

2. Con la implementación del proyecto y a través de múltiples pruebas, se permitió la comunicación entre los dos servidores Asterisk empleando el código abierto distribuido del software Asterisk, con esto se pudo demostrar que el proyecto tiene eficiencia y eficacia al momento de interconectar correctamente los dos servidores junto con los protocolos SIP, IAX2 y SS7.

3. Gracias a las importantes características de Asterisk, como su bajo costo de implementación y servicios de valor agregado preferibles en comparación a una central telefónica convencional podemos aseverar que Asterisk es un buen recurso a implementar en una empresa.

4. Los protocolos SIP, IAX2 y SS7 demostraron al momento de interconectar los dos servidores Asterisk, una la correcta comunicación entre ellos; por lo que se creó una guía de elección dependiendo en base al costo y eficiencia del objetivo al inicio de este proyecto.

5. La guía propuesta sobre la instalación de Asterisk en este proyecto puede utilizarse para cualquier instalación de este software sea cual fuera el servicio que se requiera implementar. A la fecha de esta tesis, la versión actual de *Asterisk es la 11 LTS*, sin embargo esta misma guía puede ser usada para versiones que aparezcan en una fecha posterior a este proyecto.

6. Habiendo realizado los respectivos cambios en los archivos de configuración para SS7, observamos en las consolas de Asterisk que existen 30 canales para la transmisión de audio o datos, esto indica una correcta configuración y conexión entre ambos servidores por medio de *SS7*. Además al realizar la llamada de prueba se muestran

los mensajes ISUP (CIC, IAM, DNI) que forman parte de este tipo de señalización.

7. Al momento de probar el funcionamiento nos percatamos de que pueden trabajar los 3 protocolos al mismo tiempo. El funcionamiento de un protocolo es independiente para los otros, lo que significa que mientras realizamos una llamada SIP al mismo tiempo podemos hacer una llamada *IAX* o *SS7*. El número de llamadas simultáneas dependerá del ancho de banda, tipo de códec y recursos de hardware donde se encuentra instalado Asterisk.

8. Si se requiere crear una central telefónica interna en una empresa, basta con los protocolos *SIP* e *IAX* para su implementación. Estos trabajan por medio de IP y los costos son muy bajos en comparación a SS7. Si requerimos de una conexión a la red pública, tanto de telefonía fija como de telefonía móvil, lo recomendable es trabajar con el protocolo SS7 aunque éste demande costos elevados para llevarse a cabo. Es importante resaltar los beneficios que brinda SS7 tales como robustez en la corrección de errores, *roaming*, portabilidad de número local, mayor seguridad en las llamadas, entre otros.

RECOMENDACIONES

1. Durante la implementación del proyecto es importante tener muy en cuenta el estado del led de la tarjeta Digium E1/T1,el color rojo indicará que no se están entendiendo los equipos y color verde significá la comunicación exitosa.

2. Se debe comprobar que estén correctamente conformados cada uno de los archivos de configuraciones utilizados para la realización del proyecto y así evitar problemas al momento de realizar las pruebas del mismo.

3. En base a lo realizado con SS7 podríamos aplicar, el identificador de llamadas, los números gratuitos 1-800 y características de portabilidad del número telefónico como servicios en una empresa.

4. Al momento de la elección de los *codecs* es muy importante tener en cuenta cuáles son los apropiados ya que no todos los equipos no soportan ciertos *codecs*.

Con la finalización de esta implementación, se aconseja seguir trabajando en la implementación de proyectos en los cuales se obtengan aún más los beneficios que nos proporcionan Asterisk y SS7, puesto que los mismos son herramientas que nos van a brindar apoyo en la creación y administración de una central telefónica, la misma que podría brindar múltiples tipos de servicios.

BIBLIOGRAFÍA

[1] Carballar, José, VoIP: La telefonía en Interne, Paraninfo, 2007. [2] Telefonía Voz Ip, Todo sobre Voz Ip, http://www.telefoniavozip.com/voip/que-es-la-telefonia-ip.htm, fecha de consulta junio 2013.

[3] Voz Telecom, Diferencia entre VoIp y la Telefonía Ip, http://www.voztele.com/voip-telefonia-ip/voip/voip.htm, fecha de consulta junio 2013.

[4] Telefonía Voz Ip, Telefonía IP vs Telefonía Convencional, http://www.telefoniavozip.com/voip/telefonia-ip-vs-telefonia-convencional.htm, fecha de consulta junio 2013.

[5] Telefonía Voz Ip, ¿Porque la Telefonía IP es más barata?, http://www.telefoniavozip.com/voip/la-telefonia-ip-es-mas-barata.htm, fecha de consulta junio 2013.

[6] Wikipedia, circuito virtuales, http://es.wikipedia.org/wiki/Circuito_virtual, fecha de consulta junio 2013.

[7] Wikipedia, Datagrama, https://es.wikipedia.org/wiki/Datagrama, fecha de consulta junio 2013.

[8] Telefonía Voz Ip, Ventajas de la Telefonía Ip. http://www.telefoniavozip.com/voip/ventajas-de-la-telefonia-ip.htm, fecha de consulta junio 2013. [9] Telefonía Voz Ip, Codecs Ip, http://www.telefoniavozip.com/voip/codecs-voip.htm, fecha de consulta junio 2013.

[10] Slisar, Steve, Voice Codecs for VoIP Phone Systems, http://virion.com.au/2010/08/broadband-speeds-and-voip-phone-system/, fecha de publicación Agosto 2010.

[11] Barceló, José y Iñigo, Jordi y Llorente, Silva, Protocolos y aplicaciones Internet. Barcelona, Editorial UOC, 2008.

[12] Villalón, José, VoIp: Funcionamiento básico del protocolo SIP, http://www.securityartwork.es/2008/03/03/voip-protocolo-sip/, fecha de publicación marzo 2008.

13] Voip Foro, Ejemplo de comunicación SIP, http://www.voipforo.com/SIP/SIPejemplo.php, fecha de consulta junio 2013.

[14] Rivera, Octavio y Ropero, Jorge y Otros, Servicios en Red (pp. 209-211). Madrid: Paraninfo, 2010 [15] Voip Think, Ejemplo de comunicación IAX, http://www.en.voipforo.com/IAX/IAX-example-messages.php, fecha de consulta junio 2013.

[16] Campos, Alfredo, ¿Qué es el protocolo SS7?, Tomado de http://alfredocampos.blogspot.com/2006/12/qu-es-el-protocolo-ss7.html, fecha de publicación diciembre 2006.

[17] NewNet Products ADC, Network Architecture. https://staff.ti.bfh.ch/mdm1/Kursunterlagen/SS7/Architecture%20SS7.pdf, fecha de consulta julio 2013

[18] Efort, Red de señalización número 7, http://www.efort.com/media_pdf/SS7_ES_EFORT.pdf, fecha de consulta julio 2013.

[19] Virtualpabx, Asterisk SS7 How to, http://virtualpabx.wordpress.com/2011/07/21/asterisk-ss7-howto/, fecha de publicación Julio 2011.

[20] Sunrise Telecom Incorporated, Introduction to Signaling System No. 7, http://www.syrus.ru/files/docs/control/tech/Introduction%20to%20SS7.pdf, fecha de consulta julio 2013.

[21] PT Simple Smarter Signaling, SS7 Tutorial, http://www.pt.com/resources/tutorials/ss7-tutorial, fecha de consulta julio 2013.

[22] VoIp-Info, configuración de cable cruzado E1/T1, http://www.voip-info.org/storage/users/59/27059/images/2957/medium.png, fecha de consulta julio 2013.

[23]Comunidad de usuarios de Asterisk-ES. Introducción de Asterisk, http://comunidad.asterisk-es.org/index.php?title=Introduccion_a_Asterisk, fecha de consulta julio 2013.

ANEXOS

Servidor A

Sip.conf

[general]

context=default

allowguest=no ; Deshabilita llamadas sin autenticación

srvlookup=yes

udpbindaddr=0.0.0.0 ; Subred de trabajo

transport=udp ; Protocolo de transporte

qualify=yes ; Para que aparezca el status del dispositivo softphone

register=sipservidorA:password@10.10.0.8/sipservidorB

[sipservidorB]

type=friend

secret=password

context=users

qualify=yes

host=dynamic

language=en

disallow=all

allow=gsm

allow=ulaw

allow=alaw

[TelefonoA1]

type=friend

secret=1234 ; Password de telefono

;username= es el nombre de la extension sip en este caso está colocando TelefonoA pero se puede cambiar

host=dynamic ; Dispositivo puede tener cualquier Ip para conectarse.

context=users

nat=yes

disallow=all

allow=gsm

allow=ulaw

allow=alaw

[TelefonoA2]

type=friend

secret=4321

host=dynamic

context=users

nat=yes

disallow=all

allow=gsm

allow=ulaw

allow=alaw

```
[TelefonoA3]
type=friend
secret=5678
host=dynamic
context=users
nat=yes
disallow=all
allow=gsm
allow=ulaw
allow=alaw

[TelefonoA4]
type=friend
secret=7893
host=dynamic
context=users
nat=yes
disallow=all
allow=gsm
allow=ulaw
allow=alaw

[TelefonoA5]
type=friend
```

secret=8950

host=dynamic

context=users

nat=yes

disallow=all

allow=gsm

allow=ulaw

allow=alaw

[TelefonoA6]

type=friend

secret=8742

host=dynamic

context=users

nat=yes

disallow=all

allow=gsm

allow=ulaw

allow=alaw

[TelefonoA7]

type=friend

secret=9876

host=dynamic

```
context=users
nat=yes
disallow=all
allow=gsm
allow=ulaw
allow=ulaw

[TelefonoA8]
type=friend
secret=7896
host=dynamic
context=users
nat=yes
disallow=all
allow=gsm
allow=ulaw
allow=ulaw

[TelefonoA9]
type=friend
secret=9988
host=dynamic
context=users
nat=yes
```

disallow=all

allow=gsm

allow=ulaw

allow=ulaw

[TelefonoA10]

type=friend

secret=8899

host=dynamic

context=users

nat=yes

disallow=all

allow=gsm

allow=ulaw

Extensions.conf

```
[general]
static=yes
writeprotect=no
autofallthrough=yes
clearglobalvars=no
priorityjump=no
[globals] ; aquí se crear los contextos ( reglas de marcación )
[users]
exten => 777,1,Dial(SS7/5000,20)
exten => 777,n,Hangup()
; Extensiones sip
exten => 1001,1,Dial(SIP/TelefonoA1,20)
exten => 1002,1,Dial(SIP/TelefonoA2,20)
exten => 1003,1,Dial(SIP/TelefonoA3,20)
exten => 1004,1,Dial(SIP/TelefonoA4,20)
exten => 1005,1,Dial(SIP/TelefonoA5,20)
exten => 1006,1,Dial(SIP/TelefonoA6,20)
exten => 1007,1,Dial(SIP/TelefonoA7,20)
exten => 1008,1,Dial(SIP/TelefonoA8,20)
exten => 1009,1,Dial(SIP/TelefonoA9,20)
exten => 1010,1,Dial(SIP/TelefonoA10,20)
; Extensiones SIP del servidor B
exten => _3XXX,1,NoOp()
```

```
exten => _3XXX,n,Dial(SIP/sipservidorB/${EXTEN})
exten => _3XXX,n,Hangup()
;Menu de ayuda para Servidor B
exten => 002,1,NoOp()
exten => 002,n,Dial(SIP/sipservidorB/Bhelp)
;Menu de ayuda para Servidor A
exten => Ahelp,1,Goto(Ayuda,helpext,1)
[Ayuda]
exten => helpext,1,Answer()
same => n,Background(soporte)
same => n,WaitExten(5)
exten => 1,1,Goto(users,1001,1)
exten => 2,1,Goto(users,1002,1)
exten => 3,1,Goto(users,1003,1)
exten => 4,1,Goto(users,1004,1)
exten => 5,1,Goto(users,1005,1)
; Grabación IVR
exten => 12345,1,Answer()
exten => 12345,n,Wait(1)
exten => 12345,n,Record(soporte.gsm)
exten => 12345,n,Wait(1)


[iaxgroup]
```

```
exten => 8001,1,Dial(IAX2/TiaxA1,20)
exten => 8002,1,Dial(IAX2/TiaxA2,20)
exten => 8003,1,Dial(IAX2/TiaxA3,20)
exten => 8004,1,Dial(IAX2/TiaxA4,20)
exten => 8005,1,Dial(IAX2/TiaxA5,20)
exten => 8006,1,Dial(IAX2/TiaxA6,20)
exten => 8007,1,Dial(IAX2/TiaxA7,20)
exten => 8008,1,Dial(IAX2/TiaxA8,20)
exten => 8009,1,Dial(IAX2/TiaxA9,20)
exten => 8010,1,Dial(IAX2/TiaxA10,20)
; Extensiones IAX del servidor B
exten => _9XXX,1,NoOp()
exten => _9XXX,n,Dial(IAX2/asterisk01iax/${EXTEN})
exten => _9XXX,n,Hangup()
```

IAX.conf

```
[general]

;bindport=4569

bindaddr=0.0.0.0

delayrect=yes

srvlookup=yes

bandwidth=low

disallow=all

allow=ulaw

allow=alaw

allow=gsm

register => asterisk02iax:trunk@10.10.0.8 ; aqui se pone la ip del otro
servidor

[asterisk01iax]

type=friend

user=trunk

secret=trunk

host=dynamic

trunk=yes

qualify=yes

deny=0.0.0.0/0.0.0.0

permit=10.10.0.8/255.255.255.0

context=iaxgroup     ;contexto que esté utilizando en extensiones sip
```

```
[TiaxA1]
type=friend
host=dynamic
secret=1234iax
context=iaxgroup
qualify=yes
callerid=iax1

[TiaxA2]
type=friend
host=dynamic
secret=4321iax
context=iaxgroup
qualify=yes
callerid=iax2

[TiaxA3]
type=friend
host=dynamic
secret=3333iax
context=iaxgroup
qualify=yes
callerid=iax3
```

```
[TiaxA4]
type=friend
host=dynamic
secret=4444iax
context=iaxgroup
qualify=yes
callerid=iax4

[TiaxA5]
type=friend
host=dynamic
secret=5555iax
context=iaxgroup
qualify=yes
callerid=iax5

[TiaxA6]
type=friend
host=dynamic
secret=6666iax
context=iaxgroup
qualify=yes
callerid=iax6
```

```
[TiaxA7]
type=friend
host=dynamic
secret=7777iax
context=iaxgroup
qualify=yes
callerid=iax7

[TiaxA8]
type=friend
host=dynamic
secret=8888iax
context=iaxgroup
qualify=yes
callerid=iax8

[TiaxA9]
type=friend
host=dynamic
secret=9999iax
context=iaxgroup
qualify=yes
callerid=iax9
```

```
[TiaxA10]
type=friend
host=dynamic
secret=1010iax
context=iaxgroup
qualify=yes
callerid=iax10
```

SS7.conf

[linkset-siuc]

; The linkset is enabled

enabled => yes

; The end-of-pulsing (ST) is not used to determine when incoming address is complete

enable_st => no

; Reply incoming call with CON rather than ACM and ANM

use_connect => yes

; The CIC hunting policy (even_mru, odd_lru, seq_lth, seq_htl) is even CIC numbers, most recently used

hunting_policy => even_mru

; Incoming calls are placed in the ss7 context in the asterisk dialplan

context => users

; The language for this context is da

language => es

; The value and action for t35. Value is in msec, action is either st or timeout

; If you use overlapped dialling dial plan, you might choose: t35 => 4000,st

t35 => 15000,timeout

; The subservice field: national (8), international (0), auto or decimal/hex value

; The auto means that the subservice is obtained from first received SLTM

subservice => auto

; The host running the mtp3 service

; mtp3server => localhost

[link-l1]

; This link belongs to linkset siuc

linkset => siuc

; The speech/audio circuit channels on this link

channels => 1-15,17-31

; The signalling channel

schannel => 16

; To use the remote mtp3 service, use 'schannel => remote,16'

; The first CIC

firstcic => 1

; The link is enabled

enabled => yes

; Echo cancellation

; echocancel can be one of: no, 31speech (enable only when transmission medium is 3.1Khz speech), allways

; echocancel => no

; echocan_train specifies training period, between 10 to 100 msec

; echocan_train => 350

; echocan_taps specifies number of taps, 32, 64, 128 or 256

; echocan_taps => 128

[host-asterisk]

; chan_ss7 auto-configures by matching the machines host name with the host-<name>

; section in the configuration file, in this case 'gentoo1'. The same

; configuration file can thus be used on several hosts.

; The host is enabled

enabled => yes

; The point code for this SS7 signalling point is 0x8e0

opc => 0x1

; The destination point (peer) code is 0x3fff for linkset siuc

dpc => siuc:0x2

; Syntax: links => link-name:digium-connector-no

; The links on the host is 'l1', connected to span/connector #1

links => l1:1

; The SCCP global title: translation-type, nature-of-address, numbering-plan, address

;globaltitle => 0x00, 0x04, 0x01, 4546931411

;ssn => 7

;route => 919820405471:ra_geb, 919820367598:ra_geb, 919820706441:ra_geb, :ra_geb

[jitter]

;------------------------------ JITTER BUFFER CONFIGURATION ----------------------------

; jbenable = yes ; Enables the use of a jitterbuffer on the receiving side of a

  ; SIP channel. Defaults to "no". An enabled jitterbuffer will

  ; be used only if the sending side can create and the receiving

  ; side can not accept jitter. The SIP channel can accept jitter,

   ; thus a jitterbuffer on the receive SIP side will be used only

; if it is forced and enabled.

; jbforce = no ; Forces the use of a jitterbuffer on the receive side of a SIP

; channel. Defaults to "no".

; jbmaxsize = 200 ; Max length of the jitterbuffer in milliseconds.

; jbresyncthreshold = 1000 ; Jump in the frame timestamps over which the jitterbuffer is

; resynchronized. Useful to improve the quality of the voice, with

; big jumps in/broken timestamps, usually sent from exotic devices

; and programs. Defaults to 1000.

; jbimpl = fixed ; Jitterbuffer implementation, used on the receiving side of a SIP

; channel. Two implementations are currently available - "fixed"

; (with size always equals to jbmaxsize) and "adaptive" (with

; variable size, actually the new jb of IAX2). Defaults to fixed.

; jblog = no; Enables jitterbuffer frame logging. Defaults to "no".

system.conf

Autogenerated by /usr/sbin/dahdi_genconf on Fri Aug 02 10:40:45 2013

; 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: TE4/0/1 "T4XXP (PCI) Card 0 Span 1" HDB3/CCS RED

;group=0,11

;context=from-pstn

;switchtype = euroisdn

;signalling = pri_cpe

;channel => 1-15,17-31

;context = default

;group = 63

; Span 2: TE4/0/2 "T4XXP (PCI) Card 0 Span 2" HDB3/CCS/CRC4 RED

group=0,12

context=from-pstn

switchtype = euroisdn

signalling = pri_cpe

channel => 32-46,48-62

context = default

group = 63

; Span 3: TE4/0/3 "T4XXP (PCI) Card 0 Span 3" HDB3/CCS/CRC4 RED

group=0,13

context=from-pstn

switchtype = euroisdn

signalling = pri_cpe

channel => 63-77,79-93

context = default

group = 63

; Span 4: TE4/0/4 "T4XXP (PCI) Card 0 Span 4" (MASTER) HDB3/CCS/CRC4 ClockSource

group=0,14

context=from-pstn

switchtype = euroisdn

signalling = pri_cpe

channel => 94-108,110-124

context = default

group = 63

Servidor B

Sip.conf

```
[general]
context=default
allowguest=no
srvlookup=yes
udpbindaddr=0.0.0.0
transport=udp
qualify=yes
register=sipservidorB:password@10.10.0.1/sipservidorA
[sipservidorA]
type=friend
secret=password
context=users
qualify=yes
host=dynamic
language=en
disallow=all
allow=gsm
allow=ulaw
allow=alaw

[TelefonoB1]
type=friend
```

secret=1234B

context=users

;nat=yes

host=dynamic

disallow=all

allow=gsm

allow=ulaw

allow=alaw

[TelefonoB2]

type=friend

secret=4321B

host=dynamic

context=users

;nat=yes

disallow=all

allow=gsm

allow=ulaw

allow=alaw

[TelefonoB3]

type=friend

secret=5678B

host=dynamic

```
context=users
;nat=yes
disallow=all
allow=gsm
allow=ulaw
allow=alaw

[TelefonoB4]
type=friend
secret=7893B
host=dynamic
context=users
;nat=yes
disallow=all
allow=gsm
allow=ulaw
allow=alaw

[TelefonoB5]
type=friend
secret=8950B
host=dynamic
context=users
;nat=yes
```

```
disallow=all
allow=gsm
allow=ulaw
allow=alaw

[TelefonoB6]
type=friend
secret=5529B
host=dynamic
context=users
;nat=yes
disallow=all
allow=gsm
allow=ulaw
allow=alaw

[TelefonoB7]
type=friend
secret=8922B
host=dynamic
;nat=yes
context=users
disallow=all
allow=gsm
```

```
allow=ulaw
allow=alaw

[TelefonoB8]
type=friend
secret=3322B
host=dynamic
context=users
;nat=yes
disallow=all
allow=gsm
allow=ulaw
allow=alaw

[TelefonoB9]
type=friend
secret=2243B
host=dynamic
context=users
;nat=yes
disallow=all
allow=gsm
allow=ulaw
allow=alaw
```

```
[TelefonoB10]
type=friend
secret=3820B
host=dynamic
context=users
;nat=yes
disallow=all
allow=gsm
allow=ulaw
allow=alaw
```

Extensions.conf

```
[general]
static=yes
writeprotect=no
autofallthrough=yes
clearglobalvars=no
priorityjump=no
[globals]
[users]
exten => 5000,1,Playback(vm-sorry)
exten => 5000,n,Hangup()
[users]
exten => 3001,1,Dial(SIP/TelefonoB1/20)
exten => 3002,1,Dial(SIP/TelefonoB2/20)
exten => 3003,1,Dial(SIP/TelefonoB3/20)
exten => 3004,1,Dial(SIP/TelefonoB4/20)
exten => 3005,1,Dial(SIP/TelefonoB5/20)
exten => 3006,1,Dial(SIP/TelefonoB6/20)
exten => 3007,1,Dial(SIP/TelefonoB7/20)
exten => 3008,1,Dial(SIP/TelefonoB8/20)
exten => 3009,1,Dial(SIP/TelefonoB9/20)
exten => 3010,1,Dial(SIP/TelefonoB10/20)
; Extensiones en el servidor A
exten => _1XXX,1,NoOp()
```

```
exten => _1XXX,n,Dial(SIP/sipservidorA/${EXTEN})
exten => _1XXX,n,Hangup()
; Extensión de ayuda
exten => 001,1,NoOp()
exten => 001,n,Dial(SIP/sipservidorA/Ahelp)
;Menu de ayuda
exten => Bhelp,1,Answer()
same => n,Background(soporte)
same => n,Wait(2)
exten => 1,1,Goto(users,3001,1)
exten => 2,1,Goto(users,3002,1)
exten => 3,1,Goto(users,3003,1)
exten => 4,1,Goto(users,3004,1)
exten => 5,1,Goto(users,3005,1)
[iaxgroupB]
exten => 9001,1,Dial(IAX2/TiaxB1/20)
exten => 9002,1,Dial(IAX2/TiaxB2,20)
exten => 9003,1,Dial(IAX2/TiaxB3,20)
exten => 9004,1,Dial(IAX2/TiaxB4,20)
exten => 9005,1,Dial(IAX2/TiaxB5,20)
exten => 9006,1,Dial(IAX2/TiaxB6,20)
exten => 9007,1,Dial(IAX2/TiaxB7,20)
exten => 9008,1,Dial(IAX2/TiaxB8,20)
exten => 9009,1,Dial(IAX2/TiaxB9,20)
```

```
exten => 9010,1,Dial(IAX2/TiaxB10,20)

exten => _8XXX,1,NoOp()

exten => _8XXX,n,Dial(IAX2/asterisk02iax/${EXTEN})

exten => _8XXX,n,Hangup()
```

IAX.conf

```
[general]
bindport=4569
bindaddr=0.0.0.0
delayreject=yes
srvlookup=yes
bandwith=low
disallow=all
allow=ulaw
allow=alaw
allow=gsm
register => asterisk01iax:trunk@10.10.0.1
[asterisk02iax]
type=friend
user=trunk
secret=trunk
host=dynamic
trunk=yes
qualify=yes
deny=0.0.0.0/0.0.0.0
permit=10.10.0.1/255.255.255.0
context=iaxgroupB
```

```
[TiaxB1]
type=friend
host=dynamic
secret=1234iaxB
context=iaxgroupB
qualify=yes
callerid=iax1B

[TiaxB2]
type=friend
host=dynamic
secret=4321iaxB
context=iaxgroupB
qualify=yes
callerid=iax2B

[TiaxB3]
type=friend
host=dynamic
secret=4353iaxB
context=iaxgroupB
qualify=yes
callerid=iax3B
```

```
[TiaxB4]
type=friend
host=dynamic
secret=3522iaxB
context=iaxgroupB
qualify=yes
callerid=iax4B

[TiaxB5]
type=friend
host=dynamic
secret=2380iaxB
context=iaxgroupB
qualify=yes
callerid=iax5B

[TiaxB6]
type=friend
host=dynamic
secret=9019iaxB
context=iaxgroupB
qualify=yes
callerid=iax6B
```

[TiaxB7]

type=friend

host=dynamic

secret=8901iaxB

context=iaxgroupB

qualify=yes

callerid=iax7B

[TiaxB8]

type=friend

host=dynamic

secret=2910iaxB

context=iaxgroupB

qualify=yes

callerid=iax8B

[TiaxB9]

type=friend

host=dynamic

secret=1892iaxB

context=iaxgroupB

qualify=yes

callerid=iax9B

```
[TiaxB10]
type=friend
host=dynamic
secret=2325iaxB
context=iaxgroupB
qualify=yes
callerid=iax10B
```

SS7.conf

[linkset-siuc]

; The linkset is enabled

enabled => yes

; The end-of-pulsing (ST) is not used to determine when incoming address is complete

enable_st => no

; Reply incoming call with CON rather than ACM and ANM

use_connect => yes

; The CIC hunting policy (even_mru, odd_lru, seq_lth, seq_htl) is even CIC numbers, most recently used

hunting_policy => even_mru

; Incoming calls are placed in the ss7 context in the asterisk dialplan

context => users

; The language for this context is da

language => es

; The value and action for t35. Value is in msec, action is either st or timeout

; If you use overlapped dialling dial plan, you might choose: t35 => 4000,st

t35 => 15000,timeout

; The subservice field: national (8), international (0), auto or decimal/hex value

; The auto means that the subservice is obtained from first received SLTM

subservice => auto

; The host running the mtp3 service

;mtp3server => localhost

[link-l1]

; This link belongs to linkset siuc

linkset => siuc

; The speech/audio circuit channels on this link

channels => 1-15,17-31

; The signalling channel

schannel => 16

; To use the remote mtp3 service, use 'schannel => remote,16'

; The first CIC

firstcic => 1

; The link is enabled

enabled => yes

; Echo cancellation

; echocancel can be one of: no, 31speech (enable only when transmission medium is 3.1Khz speech), allways

;echocancel => no

; echocan_train specifies training period, between 10 to 100 msec

;echocan_train => 350

; echocan_taps specifies number of taps, 32, 64, 128 or 256

;echocan_taps => 128

[host-asteriskB]

; chan_ss7 auto-configures by matching the machines host name with the host-<name>

; section in the configuration file, in this case 'gentoo1'. The same

; configuration file can thus be used on several hosts.

; The host is enabled

enabled => yes

; The point code for this SS7 signalling point is 0x8e0

opc => 0x2

; The destination point (peer) code is 0x3fff for linkset siuc

dpc => siuc:0x1

; Syntax: links => link-name:digium-connector-no

; The links on the host is 'l', connected to span/connector #1

links => l1:1

; The SCCP global title: translation-type, nature-of-address, numbering-plan, address

; globaltitle => 0x00, 0x04, 0x01, 4546931411

; ssn => 7

; route => 919820405471:ra_geb, 919820367598:ra_geb, 919820706441:ra_geb, :ra_geb

[jitter]

;------------------------------ JITTER BUFFER CONFIGURATION ----------------------------

; jbenable = yes ; Enables the use of a jitterbuffer on the receiving side of a

; SIP channel. Defaults to "no". An enabled jitterbuffer will

; be used only if the sending side can create and the receiving

; side can not accept jitter. The SIP channel can accept jitter,

;thus a jitterbuffer on the receive SIP side will be used only

; if it is forced and enabled.

; jbforce = no; Forces the use of a jitterbuffer on the receive side of a SIP

; channel. Defaults to "no".

; jbmaxsize = 1000 ; Max length of the jitterbuffer in milliseconds.

; jbresyncthreshold = 1000 ; Jump in the frame timestamps over which the jitterbuffer is

; resynchronized. Useful to improve the quality of the voice, with

; big jumps in/broken timestamps, usually sent from exotic devices

; and programs. Defaults to 1000.

; jbimpl = fixed; Jitterbuffer implementation, used on the receiving side of a SIP

; channel. Two implementations are currently available - "fixed"

; (with size always equals to jbmaxsize) and "adaptive" (with

; variable size, actually the new jb of IAX2). Defaults to fixed.

; jblog = no ; Enables jitterbuffer frame logging. Defaults to "no".

Dahdi_channels.conf

; Autogenerated by /usr/sbin/dahdi_genconf on Fri Aug 02 10:50:24 2013

; 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: WCT1/0 "Digium Wildcard TE110P T1/E1 Card 0" (MASTER) CCSHDB3/

;group=0,11

;context=from-pstn

;switchtype = euroisdn

;signalling = pri_cpe

;channel => 1-15,17-31

;context = default

;group = 63

; Span 2: WCFXO/0 "Wildcard X100P Board 1" RED

;;; line="32 WCFXO/0/0 FXSKS (In use) (EC: MG2 - INACTIVE)"

signalling=fxs_ks

callerid=asreceived

group=0

context=from-pstn

channel => 32

callerid=

group=

context=default

Tarjeta openvox D410P – especificaciones

http://www.openvox.cn/en/products/t1e1j1-cards/d410p.html#ds

Tarjeta openvox D110P – especificaciones.

http://www.openvox.cn/en/products/t1e1j1-cards/d110p.html

D410P

Descripción general

D410P es una tarjeta de alto rendimiento, rentable cuatro lapso de voz digital.Se proporciona una interfaz compacta y potente para asterisco que puede soportar E1/T1/J1 e interfaz PRI.Con la mejora de velocidad de E / S, la tarjeta reduce el uso de la CPU y el aumento de la densidad de tarjeta por servidor.D410P es totalmente compatible con la aplicación Asterisk.El controlador de código abierto compatible con una interfaz API para el desarrollo de aplicaciones personalizadas.

D410P admite los protocolos de telefonía y de datos estándar de la industria, incluyendo primaria RDSI (tanto americanos N. y Norma Euro) familias de protocolos de voz, PPP, Cisco, HDLC, y los modos de datos Frame Relay.Ambas interfaces del lado de línea y el tronco del lado son compatibles.

Las solicitudes de destino

- Protocol (VoIP), servicios de voz sobre Internet
- Complejo árboles IVR
- "Meet-Me" Conferencing Puente
- Llamadas Plataformas Tarjeta
- Gateways VoIP (soporte SIP, H.323 y IAX)
- PBX / IVR Servicios
- Router Voz / Data (reemplazar routers caros)
- PRI / Switch Compatibilidad - red o CPE

Especificaciones técnicas

- Cuatro T1/E1/J1 puertos con interfaz PCI para aplicaciones de voz y datos de alto rendimiento
- Mediometraje tarjeta PRI
- Conector RJ48
- 32 bit 33MHz PCI y totalmente compatible con PCI 2.3
- 32 intercambios de datos bus master DMA de bits a través de la interfaz PCI a 132 Mbytes / seg
- Hasta 30, 60, 120 llamadas de voz simultáneas respectivamente
- Detección automática de compatibilidad con 5 V y 3,3 V PCI buses
- Energía: 2.6W mínimo, máximo 3,9 W a 3,3 V o 5 V.
- Temperatura de funcionamiento: 0 ° C a 50 ° C

- Temperatura de almacenamiento: -65 ° C a 125 ° C
- Dimensión: los 13cm * 10.3cm * 1.8cm
- Peso: 97g

PRI Interruptor de compatibilidad

- EuroISDN (PRI o PRA) - Q.931/Q.921
- Red o CPE
- AT & T 4ESS
- National ISDN 2
- CAS Modos de voz
- DMS 100
- 5E Lucent

Sistemas operativos

- Linux (todas las versiones, revisiones y distribuciones a partir de 1.0)

Requisitos mínimos de hardware

- 800-MHz Pentium III
- 128 MB de RAM
- Disponible ranura PCI

D110P

Descripción general

D110P es un producto único telefonía lapso E1/T1/J1 digital.

D110P es una tarjeta de alto rendimiento, rentable solo palmo de voz digital.Se proporciona una interfaz compacta y potente para asterisco que puede soportar E1/T1/J1 e interfaz PRI.

D110P soporta los protocolos estándar de la industria, incluyendo MFR2, PRI, Cisco PPP, Frame Relay, etc El bajo perfil permite que quepa dentro de una caja de montaje en rack de 2U.
D110P es 100% compatible con el hardware de Digium TE110P.Puede funcionar con controlador TE110P sin parche.

Las solicitudes de destino

- Protocol (VoIP), servicios de voz sobre Internet
- Complejo árboles IVR
- "Meet-Me" Conferencing Puente
- Llamadas Plataformas Tarjeta
- Gateways VoIP (soporte SIP, H.323 y IAX)
- PBX / IVR Servicios
- Router de voz / datos (routers caros replasce)
- PRI / Switch Compatibilidad - red o CPE

Especificaciones técnicas

- SingleT1/E1/J1 puerto con interfaz PCI para la voz de alto rendimiento y aplicaciones de datos
- Mediometraje tarjeta PRI
- Conector RJ48
- 32 bit 33MHz PCI y totalmente compatible con PCI 2.3
- 32 intercambios de datos bus master DMA de bits a través de la interfaz PCI a 132 Mbytes / seg
- Hasta 30, 60, 120 llamadas de voz simultáneas respectivamente
- Detección automática de compatibilidad con 5 V y 3,3 V PCI buses
- Energía: Máximo 1.2W en 3,3 V o 5 V.
- Temperatura de funcionamiento: 0 ° C a 50 ° C
- Temperatura de almacenamiento: -65 ° C a 125 ° C
- Dimensión: el 12.3cm * 6.5cm * 1.8cm
- Peso: 54g

Interruptores de compatibilidad

- EuroISDN (PRI o PRA) - Q.931/Q.921
- Red o CPE
- AT & T 4ESS
- National ISDN 2
- CAS Modos de voz
- DMS 100
- 5E Lucent

Sistemas operativos

- Linux (todas las versiones, revisiones y distribuciones a partir de 1.0)

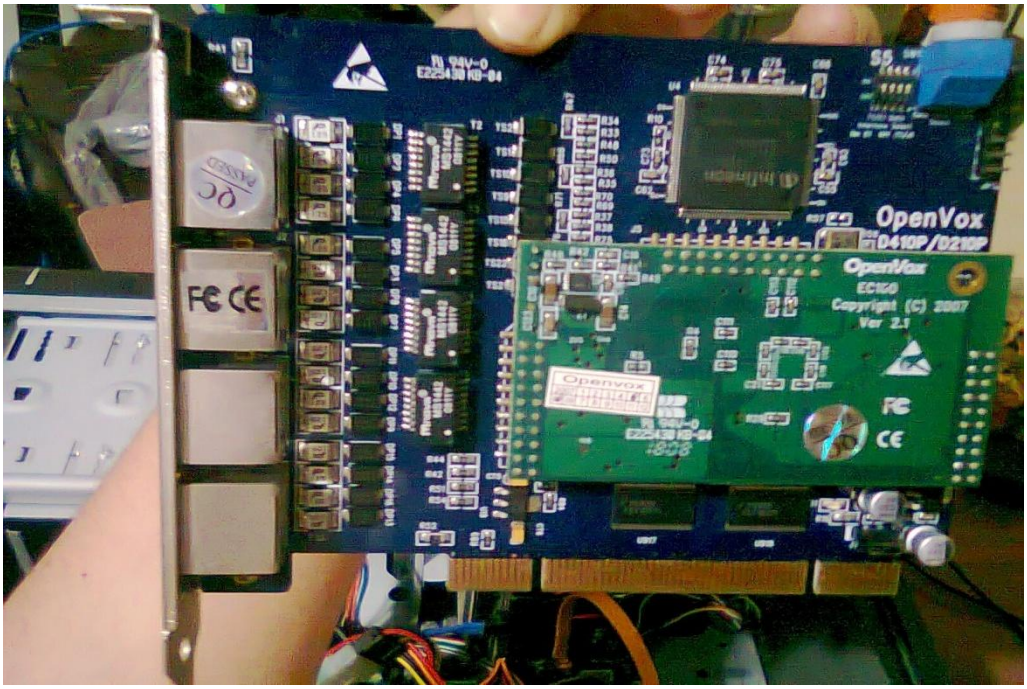
Requisitos mínimos de hardware

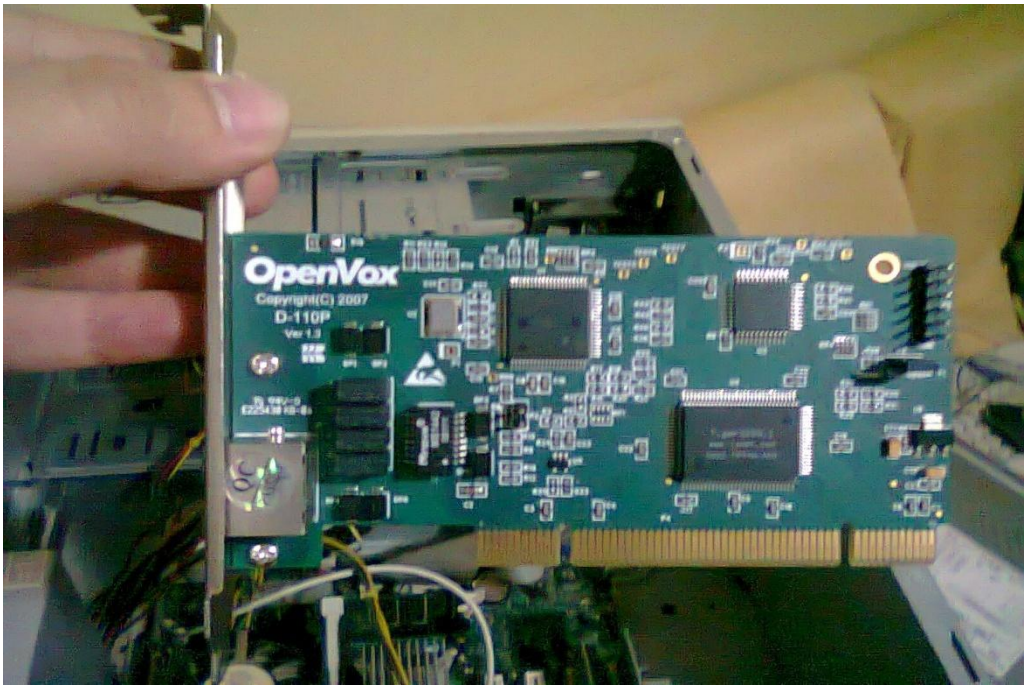
- 800-MHz Pentium III
- 128 MB de RAM
- Disponible ranura PCI

Tarjetas OPENVOX

Fotos de tarjetas en la implementación

Items / Pictures									
Products	D110P	D110E	DE115P	DE115E	DE210P	DE210E	DE410P	DE410E	
Bus Type	PCI 2.2+	PCI-E 1.0+	PCI 2.2+	PCI-E 1.0+	PCI 2.2+	PCI-E 1.0+	PCI 2.0+	PCI-E 1.0+	
Ports	1 RJ48	1 RJ48	1 RJ48	1 RJ48	2 RJ48	2 RJ48	4 RJ48	4 RJ48	
Dimensions (mm)	122.2×63.5×16	119.7×68.8×16	127.8×103.5×16	127.8×108.1×16	127.8×103.5×16	127.8×108.1×16	127.8×103.5×16	127.8×108.1×16	
Weights (g)	54	54	104	104	106	129	135	129	
Optional EC Module	N/A	N/A	EC100-32	EC100-32	EC100-64	EC100-64	EC100-128	EC100-128	
Bus Master DMA	√	√	√	√	√	√	√	√	
E1/T1	PRI	√	√	√	√	√	√	√	√
	SS1	√	√	√	√	√	√	√	√
	SS7	√	√	√	√	√	√	√	√


OpenVox
D410P/D210P
OpenVox
EC1K0
Copyright (C) 2007
Ver 2.1
FC
CE


OpenVox
Copyright(C) 2007
D-110P
Ver 1.3