



AsteriskTM PBX

Guía de la configuración

Como construir y configurar un PBX con software libre
Asterisk versión 1.4

Flavio E. Gonçalves

Tercera Generación

1º. Edición/Janeiro/2007

rev. 8.2

Copyright 2005© por Flavio E. Gonçalves
Todos los derechos reservados

Traducción Oscar Osvaldo Fueyo Alvarez
Primera edición: Diciembre de 2006
Revisión: Héctor Kee Gomez, hkee@voziptelco.com
Cover art: Karla Braga

Gonçalves, Flavio Eduardo de A.
Asterisk PBX Guía de la configuración – Florianópolis:
Título Independente, 2007

362p.

ISBN: 978-85-906904-3-6

1. Tecnologia 2. Softswitches I. Gonçalves, Flavio
II. Título

Exclusivo para

Constato: oops@asteriskguide.com
Todos los derechos reservados. Es prohibida la reproducción total o parcial de esto libro.

Prefacio

El Asterisk PBX es, en mi opinión, una revolución en áreas de telefonía ip y PBX basado en software. Durante muchos años el mercado de la telefonía fue ligado a equipamientos propietarios fabricados por grandes compañías multinacionales. A pesar de tener equipamientos de bajo costo estas arquitecturas también presentan baja funcionalidad. La convergencia de datos y voz en poco tiempo va hacer que la telefonía sea apenas una aplicación mas de redes, tornando las actuales PBX obsoletas. Con la entrada de Asterisk, más y más empresas van a poder experimentar recursos como URA – unidad de respuesta audible, DAC – distribución automática de llamadas, movilidad, buzón de voz, y conferencia, antes restrictivo solo a grandes compañías debido al alto costo.

La telefonía IP cuando adquiera masa crítica hará con el PBX de cualquier empresa pueda hablar con el PBX de cualquier otra a través de Internet. A la hora de evaluar los beneficios de Asterisk preciso entrever este horizonte futuro que operadoras IP hacen intercomunicación automática con otros PBX. La economía en DDD y DDI es solo la punta del iceberg. Este libro fue creado con el objeto de facilitar la adopción de AsteriskPBX en países de lengua Hispana. Uno de los primeros problemas que encontré intentando aprender e implementar Asterisk fue la falta de documentación. . A pesar del “Libro del Asterisk handbook”, de asteriskdocs.org y de Wiki (www.voip-info.org) que fueron las principales fuentes de referencia para este material, las informaciones se encuentran desparramadas por aquí y por allá lo que torna difícil el aprendizaje.

A pesar de usar algunos ejemplos con equipos de mercado, este material no recomienda específicamente ningún equipo o proveedor de servicios. Úselos por su cuenta y riesgo. No tenemos la pretensión de encinar todo lo que existe sobre Asterisk en este libro, pues esto seria una misión casi imposible, nuevos recursos están siendo agregados todos los días y Asterisk tiene muchos. Nuestra principal pretensión en este material es de que el lector pueda tener acceso a los principales recursos y a partir de ellos pueda descubrir e implementar recursos mas avanzados.

Yo espero que Uds. Se diviertan tanto aprendiendo Asterisk cuanto yo me divertí escribiendo sobre el, tiempo y paciencia son requisitos indispensables para probar todos los recursos de este material.

Flávio E Goncalves
Diretor Geral
V.Office Networks
flavio@asteriskguide.com

Agradecimientos

Tengo aquí que agradecer a mi familia por la paciencia de verme trabajando las madrugadas y fines de semanas para que este material pudiese ser escrito. Agradezco a Clarice mi esposa y compañera por los incentivos y apoyo y a Cristina Gonçalves y Cristiano Soares por resolver todos los embates como la publicación, distribución, tapa, marketing que posibilitaron que este material llegase a los usuarios y a Oscar Osvaldo Fueyo Alvarez por la traducción al español.

Marcas registradas

Las marcas registradas de DIGIUM®, de ASTERISK®, de IAX® y de DUNDi® son característica de Digium Inc.

Este trabajo es independiente y no es patrocinado ni es endosado por Digium Inc.

Errores

Mientras que cada precaución se ha tomado que escribía este libro, el autor no asume ninguna responsabilidad de errores y de omisiones, o daños resultando del uso de la información contenida dentro de este libro. Si usted encuentra cualquier omisión o error, apreciamos que usted nos contacte al e-mail:

oops@asteriskguide.com

Sumario

INTRODUCCIÓN AL ASTERISK.....	11
1.1 OBJETIVOS DEL CAPÍTULO	11
1.2 QUE ES ASTERISK	11
1.3 ¿PORQUE ASTERISK?	15
1.4 ARQUITECTURA DE ASTERISK	17
1.5 VISIÓN GENERAL	20
1.6 VIEJO Y NUEVO MUNDO	21
1.7 UN SISTEMA PARA TESTES Y DESARROLLO	23
1.8 ESCENARIOS DE USO DE ASTERISK.....	24
1.9 BUSCANDO INFORMACIONES RELATIVAS A ASTERISK.....	29
1.10 SUMARIO	29
1.11 CUESTIONARIO.....	30
BAJANDO E INSTALANDO ASTERISK.....	32
2.1 OBJETIVOS DEL CAPÍTULO	32
2.2 HARDWARE MÍNIMO.....	32
2.3 ESCOGIENDO UNA DISTRIBUCION DE LINUX.....	34
2.4 INSTALANDO LINUX PARA ATENDER AL ASTERISK.....	35
2.5 PREPARANDO DEBIAN PARA ASTERISK.....	45
2.6 OBTENIENDO Y COMPILANDO ASTERISK.....	47
2.7 INICIANDO Y DETENIENDO ASTERISK	49
2.8 DIRECTORIOS DE INSTALACIÓN	50
2.9 LOS ARCHIVOS DE LOG E LA ROTACIÓN DE ELLOS.....	50
2.10 INICIANDO ASTERISK USANDO UN USUARIO DIFERENTE DE ROOT.....	52
2.11 CONSIDERACIONES SOBRE LA INSTALACIÓN DE ASTERISK.....	53
2.12 SUMARIO	54
2.13 QUESTIONARIO	55
PRIMEROS PASOS EN ASTERISK.....	58
3.1 OBJETIVOS DEL CAPÍTULO	58
3.2 GRAMÁTICA DE LOS ARCHIVOS DE CONFIGURACIÓN	58
3.3 GRAMÁTICA	59
3.4 CONFIGURANDO UNA INTERFACE CON LA RED PÚBLICA O UN PBX.....	60
3.5 CONFIGURACIÓN DE LOS TELÉFONOS IP SIP.....	62
3.6 INTRODUCCIÓN AL PLAN DE DISCADO	64
3.7 LABORATORIO	73
3.8 SUMARIO	75
3.9 CUESTIONARIO	75

CANALES ANALÓGICOS Y DIGITALES..... 79

4.1 OBJETIVOS	79
4.2 CONCEPTOS BÁSICOS	79
4.3 INTERFACES DE ACCESO A LA RED PÚBLICA	81
4.4 USANDO INTERFACES FXS, FXO Y E+M	81
4.5 LÍNEAS DIGITALES E1/T1, SEÑALIZACIÓN CAS Y CCS.	84
4.6. CONFIGURANDO UN CANAL DE TELEFONÍA EN ASTERISK	85
4.7 EJEMPLO 1 - INSTALACIÓN DE UNA PUERTA FXO Y OTRA FXS	86
4.8 EJEMPLO 2 – CARGA DE DOS CANALES E1-ISDN	89
4.9 CONFIGURANDO MFC/R2.....	97
4.10 OPCIONES DE CONFIGURACIÓN DEL ARCHIVO ZAPATA.CONF	112
4.11 NOMENCLATURA DE LOS CANALES ZAP	118
4.12 NOMENCLATURA DE LOS CANALES UNICALL	118

VOZ SOBRE IP..... 120

5.1 OBJETIVOS	120
5.2 INTRODUCCIÓN	120
5.3 BENEFICIOS DE LA VOZ SOBRE IP	120
5.4 ARQUITECTURA DE ASTERISK Y VOZ SOBRE IP	122
5.5 COMO ESCOGER UN PROTOCOLO	123
5.6 CONCEPTO DE PEERS, USERS Y FRIENDS	124
5.7 CODECS Y CONVERSIÓN DE CODECS	125
5.8 COMO ESCOGER EL CODEC.	126
5.9 OVERHEAD CAUSADO POR LOS ENCABEZADOS.....	127
5.10 INGENIERÍA DEL TRÁFICO	128
5.11 ESTRATEGIAS DE REDUCCIÓN DEL USO DE BANDA ANCHA	130
5.12 SUMARIO	131
5.13 CUESTIONARIO.....	132

EL PROTOCOLO IAX..... 135

6.1 OBJETIVOS DEL CAPÍTULO.....	135
6.2 INTRODUCCIÓN	135
6.3 TEORÍA DE OPERACIÓN	136
6.4 FORMATO DE LOS FRAMES.....	137
6.5 USO DE BANDA ANCHA.....	139
6.6 NOMENCLATURA DE LOS CANALES	141
6.7 ESCENARIOS DE USO.....	142
6.8 AUTENTIFICACIÓN EN EL IAX	151
6.9 CONFIGURACIÓN DEL ARCHIVO IAX.CONF	154
6.10 COMANDOS DE DEPURACIÓN DO IAX2.	159
6.11 SUMARIO	162
6.12 CUESTIONARIO.....	163

EL PROTOCOLO SIP..... 166

7.1 OBJETIVOS	166
7.2 VISIÓN GENERAL	166
7.3 ARCHIVO DE CONFIGURACIÓN SIP.CONF	172
7.4 CONFIGURANDO UN CLIENTE SIP.....	174
7.5 CONECTÁNDO SE A UN PROVEEDOR SIP.	179
7.6 AUTENTIFICACIÓN DE LAS LLAMADAS SIP ENTRANTES.	182
7.7 NOMENCLATURA DE LOS CANALES SIP.....	183
7.8 SIP NAT TRAVERSAL.....	184
7.9 NAT EN EL PASAJE DE LA SEÑALIZACIÓN SIP	186
7.10 NAT EN EL FLUJO DE MEDIA RTP	187
7.11 FORMAS DE PASAJES POR EL NAT	189
7.12 SOLUCIONES PRÁCTICAS PARA EL ASTERISK	193
7.13 CONSIDERACIONES FINALES SOBRE EL NAT.....	197
7.14 CUESTIONARIO.....	198

RECURSOS BÁSICOS DEL PLAN DE DISCADO..... 202

8.1 OBJETIVOS DEL CAPÍTULO	202
8.2 INTRODUCCIÓN	202
8.3 ESTRUCTURA DEL ARCHIVO EXTENSIONS.CONF	202
8.4 CONTEXTOS Y EXTENSIONES	205
8.5 EXTENSIONES.....	207
8.6 VARIABLES	212
8.7 EXPRESIONES.....	215
8.8 FUNCIONES	216
8.9 O PLAN DE DISCADO EN LA PRÁCTICA	218
8.10 EL ESTILO DE LA VERSIÓN 1.2	223
8.11 CREANDO UN PLAN DE DISCADO SIMPLE.....	224
8.12 SUMARIO	225
8.13 CUESTIONARIO.....	226

RECURSOS AVANZADOS DEL PLAN DE DISCADO..... 229

9.1 OBJETIVOS	229
9.2 INCLUSIÓN DE CONTEXTOS.....	229
9.3 COMO EL PLAN DE DISCADO ENCUENTRA LA EXTENSIÓN.....	231
9.4 RECIBIENDO UNA LLAMADA EN UN MENÚ DE VOZ.....	237
9.5 SWITCHES, ENCAMINANDO PARA OTRO ASTERISK	240
9.6 MACROS.....	241
9.7 LA BASE DE DATOS DE ASTERISK	242
9.8 PROGRAMANDO EL RECURSO DE LISTA NEGRA.....	243
9.9 CONTEXTOS BASADOS EN HORARIO	245
9.10 ABRIENDO UN NUEVO TONO DE DISCADO CON DISA.....	247
9.11 LIMITANDO EL NÚMERO DE LLAMADAS SIMULTANEAS	247

9.12 UN PLAN DE DISCADO EN LA PRÁCTICA	248
9.13 CUESTIONARIO.....	253

CONFIGURANDO RECURSOS AVANZADOS 256

10.1 OBJETIVOS	256
10.2 SOPORTE A LOS RECURSOS DE PBX	256
10.3 ESTACIONAMIENTO DE LLAMADAS.....	258
10.4 CAPTURA DE LLAMADAS - CALL-PICKUP	260
10.5 TRANSFERENCIA DE LLAMADAS - CALL TRANSFER.....	260
10.6 CONFERENCIA – CALL CONFERENCE	261
10.7 GRABANDO UNA COMUNICACIÓN	265
10.8 MÚSICA EN ESPERA – MUSIC ON HOLD.....	265
10.9 CUESTIONARIO.....	267

DAC DISTRIBUCIÓN AUTOMÁTICA DE LLAMADAS..... 269

11.1 OBJETIVOS	269
11.2 INTRODUCCIÓN	269
11.3 MIEMBROS	270
11.4 ESTRATEGIAS	270
11.5 MENÚ PARA EL USUARIO	271
11.6 NUEVOS RECURSOS	271
11.7 CONFIGURACIÓN.....	271
11.8 FUNCIONAMIENTO DE LAS FILAS	274
11.9 CUESTIONARIO.....	276

EL BUZÓN DE VOZ 279

12.1 OBJETIVOS	279
12.2 INTRODUCCIÓN	279
12.3 LISTA DE TAREAS PARA LA CONFIGURACIÓN	279
12.4 ARCHIVOS DE CONFIGURACIÓN DEL VOICEMAIL.	283
12.5 INTERFASE WEB PARA EL BUSON DE VOZ.	288
12.6 SUMARIO	289
12.7 CUESTIONARIO.....	290

BILLING EN ASTERISK..... 292

13.1 INTRODUCCIÓN	292
13.2 OBJETIVOS:	292
13.3 FORMATO DE LOS INFORMES DE CONTABILIDAD DE LLAMADAS DEL ASTERISK ..	292
13.4 CODIGOS DE CUENTA Y CONTABILIZACIÓN AUTOMÁTICA.	293
13.5 CAMBIANDO EL FORMATO DEL CDR	294
13.6 MÉTODOS DE ALMACENAMIENTO DEL CDR	294

13.7 APLICACIONES Y FUNCIONES	296
13.8 USANDO AUTENTIFICACIÓN DE LOS USUARIOS	297
13.9 USANDO LAS CONTRASEÑAS DEL BUZÓN DE VOZ.....	298
13.10 SUMARIO	299
13.11 CUESTIONARIO.....	299

INTEGRACIÓN DE ASTERISK USANDO AGI Y AMI..... 302

14.1 INTRODUCCIÓN	302
14.2 OBJETIVOS DE ESTE CAPÍTULO	302
14.3 PRINCIPALES FORMAS DE EXTENDER EL ASTERISK	302
14.4 EXTENDIENDO ASTERISK A TRAVÉS DE LA CONSOLA.	303
14.5 EXTENDIENDO ASTERISK USANDO EL APLICATIVO SYSTEM.....	303
14.6 LO QUE REPRESENTA AMI	304
14.7 CONFIGURANDO USUARIOS Y PERMISOS	305
14.8 ASTERISK MANAGER PROXY	309
14.9 ASTERISK GATEWAY INTERFACE	311
14.10 ALTERANDO EL CÓDIGO FUENTE	317
14.11 SUMARIO	317
14.12 CUESTIONARIO.....	319

ASTERISK REAL-TIME..... 322

15-1 INTRODUCCIÓN	322
15.2 OBJETIVOS DE ESTE CAPÍTULO	322
15.3 MODO DE FUNCIONAMIENTO DE ASTERISK REAL TIME	323
15.4 LAB 1 INSTALANDO EL ASTERISK REAL/TIME	324
15.5 COMO CONFIGURAR EL ASTERISK REAL TIME.....	324
15.6 CONFIGURACIÓN DE LA BASE DE DATOS.....	327
15.7 LAB 2 – INSTALANDO EL BANCO Y CREANDO LAS TABLAS	328
15.8 LAB 3 – CONFIGURANDO Y TESTEANDO EL ARA	331
15.9 SUMARIO	333
15.10 CUESTIONARIO.....	333

RESPUESTA DE LOS EJERCICIOS..... 335

RESPUESTAS DEL CAPÍTULO 1	335
RESPUESTAS DEL CAPÍTULO 2	337
RESPUESTAS DEL CAPÍTULO 3	339
RESPUESTAS DEL CAPÍTULO 5.....	342
RESPUESTAS DEL CAPITULO 6	344
RESPUESTAS DEL CAPITULO 7	346
RESPUESTAS DEL CAPITULO 8	348
RESPUESTAS DEL CAPITULO 9	350
RESPUESTAS DEL CAPÍTULO 10	352

RESPUESTAS DEL CAPÍTULO 11	353
RESPUESTAS DEL CAPÍTULO 12	355
RESPUESTAS DEL CAPÍTULO 13	357
RESPUESTAS DEL CAPÍTULO 14	359
RESPUESTAS DEL CAPÍTULO 15	361

Exclusivo para Vicente Miguel Perez Paredes

Introducción al Asterisk

En este capítulo vamos aprender que es Asterisk, cual es su arquitectura y como puede ser utilizado.

1.1 Objetivos del capítulo

Objetivos

Para el final de este capítulo el lector deberá estar apto para:

- Explicar que es Asterisk , como fue creado este, su vinculación con otros proyectos y el rol de Digium en Asterisk.
- Reconocer la arquitectura básica de Asterisk y empesar a familiarise con conceptos como channels y codecs.
- Descubrir escenarios de negocios donde Asterisk puede ser usado.

1.2 Que es Asterisk

Asterisk es un software PBX que usa el concepto de software libre (GPL). Digium, empresa que promueve el Asterisk, invierte en ambos aspectos, el desenvolvimiento de código fuente y en hardware de telefonía de bajo costo que funciona con Asterisk. El Asterisk corre en plataforma Linux y otras plataformas Unix con o sin hardware conectando a la red pública de telefonía, PSTN (Public Service Telephony Network).

El Asterisk permite conectividad en tiempo real entre las redes PSTN y redes Voip.

Con Asterisk, usted no tiene apenas un cambio excepcional de su PBX. El Asterisk es mucho más que un PBX central. Con Asterisk en su red, Usted puede crear cosas nuevas en telefonía como:

- Conectar empleados trabajando desde casa para un PBX de la oficina sobre conexiones de banda ancha.
- Conectar oficinas en varias provincias sobre IP. Esto puede ser hecho por Internet o por una red IP privada.
- Dar a los funcionarios, buzón de voz, integrándolo con una "web" y sus e-mail.
- Construir aplicaciones de respuesta automática por voz, que puede conectarlo a un sistema de pedidos, por ejemplo, o a otras aplicaciones internas.
- Dar acceso al PBX de la compañía para usuarios que viajan, conectando sobre la VPN de un aeropuerto o un hotel.
- Y mucho más...

Asterisk incluye muchos recursos que solo eran encontrados en sistemas de mensajería unificada "sistema encima de la línea" como:

- Música en espera para clientes en filas de espera, soportando streaming de media así como música en MP3.
- Filas de llamada donde agentes de forma conjunta atienden las llamadas y monitorean dicha fila.
- Integración para sintetización de la conversación (text-to-speech).
- Registro detallado de llamadas (call-detail-records) para integración con sistemas de tarificación.
- Integración con reconocimiento de voz (Tal como el software de código abierto para reconocimiento de voz).
- La habilidad de interfaces con líneas telefónicas normales, ISDN en acceso básico (2B+D) y primario (30B+D).

1.2.1 Cual es el papel de Digium?

Digium es fundada en Huntsville, Alabama. Digium es la creadora y desarrolladora primaria de Asterisk, el primer PBX de código abierto de la industria. Usado en conjunto con las placas de telefonía PCI, ellas ofrecen un manejo estratégico con excelente relación costo/beneficio para el transporte de voz y datos sobre arquitecturas TDM, conmutadas y redes Ethernet.

Digium es hoy el principal patrocinador de Asterisk y uno de los líderes de la industria de PBX en código abierto, siendo Mark Spencer el creador y principal soporte de Asterisk, él es hoy admirado por el gran trabajo que realizo y por la responsabilidad que supo acarrear.

1.2.2 El proyecto Zapata

El proyecto ZAPATA fue conducido por Jim Dixon. El es el responsable por el desarrollo del hardware de DIGIUM. Es interesante resaltar que el hardware también es abierto y puede ser producido por cualquier empresa. Hoy la placa con 4 E1/T1s es producida por Digium, Sangoma y también por Varion. La historia del proyecto zapata puede ser vista en:

<http://www.asteriskdocs.org/modules/tinycontent/index.php?id=10>)

Una pequeña traducción puede ser encontrada abajo,

Por Jim Dixon

Hace 20 o 25 años atrás, La AT&T comenzó a ofrecer una API permitiendo a los usuarios personalizar la funcionalidad de su sistema de buzón de voz y auto-atención de llamadas el cual recibió el nombre Audix. Audix corría en plataforma Unix y costaba como todo en telefonía antes y hasta el momento, millares de dólares por puerta con una funcionalidad bastante limitada.

En una tentativa de tornar las colas posibles y atractivas (Especialmente para quien no tenía un PBX AT&T) algunos fabricantes vieron que con una placa que podía ser colocada en un PC que corriera con DOS y respondía a una única línea telefónica (FXO solamente) habría posibilidades. Las placas no tenían una calidad tan buena en cuanto a las actuales y muchas terminaron como secretarías electrónicas siendo igualmente malas en calidad.

Nuevas placas de telefonía fueron lanzadas con precios altos y las compañías continuaron gastando una faja de miles de dólares por puerta. Al final de cuentas, con los márgenes altos de muchos fabricantes, las placas de telefonía poseían mucha capacidad de procesamiento en forma de DSPs, procesadores de señales digitales. Si ud. observa aun hoy un gateway de voz sobre ip, va a ver que buena parte del costo todavia está relacionada a los DSPs.

Todavía, el poder de procesamiento de los microcomputadores continuó creciendo. De forma que para probar el concepto inicial compré una placa Mitel89000C "ISDN Express Development Card" y escribí un driver para el FreeBSD. La placa ocupó poco procesamiento de un Pentium III 600Mhz, probando que si no fuese por la limitación de I/O (La placa trataba de forma ineficiente la I/O exigiendo muchos wait-states) ella podría atender de 50 a 75 canales. Como resultado de este acontecimiento, yo salí y compré lo necesario para crear un nuevo diseño de tarjeta ISA que usase el I/O de forma eficiente. Yo conseguí dos T1s (48 canales) de datos transferidos sobre el bus entre memoria y el microprocesador y el PC gerencia esto sin problemas. Entonces ya tenía yo las placas ofrecidas para la venta (Unas 50 fueron vendidas) y coloqué el diseño completo (incluyendo archivos de ploteo de la placa) en la web. .

Como el concepto era revolucionario y sabia que haría repercusiones en la industria, Yo decidí colocar un nombre inspirado en el revolucionario Mexicano y di el nombre a la organización de Emiliano Zapata y decidiendo llamar a la placa "tormenta". Así comenzó la telefonía ZAPATA. Escribí un driver completo y lo coloque en la red. La respuesta que yo obtuve fue casi siempre, "Muy bien... y ud. tiene para Linux?".

Personalmente yo nunca había visto Linux correr antes, mas fui rápidamente a el Fry's (Una tienda enorme de productos electrónicos, famosa en EUA) y compré una copia de Linux Red Hat 6.0. Di una ojeada a los drivers y usé el Vídeo Spigot como base para traducir el driver de BSD para Linux.

De cualquier forma mi experiencia con Linux no era buena y comencé a tener problemas en desarrollar el módulo del kernel en la forma de módulos cargables. De cualquier forma lo libere en la red sabiendo que algún guru en Linux se reiría de él y talvez me ayudaría a reformarlo en un "Linux" apropiado. En 48 horas recibí un e-mail de un sujeto de Alabama (Mark Spencer), que se ofreció para hacer exactamente esto. Me entere que, él decía que tenia algo que seria perfecto para a todo esto en conjunto (El Asterisk).

En ese momento Asterisk era un concepto funcional, porque no tenía una forma real de funcionar de forma práctica e útil. El casamiento del sistema de telefonía Zapata y el diseño de bibliotecas de hardware/driver y interfaces permitirían a él crecer para ser un PBX real que podría hablar con teléfonos y líneas reales.

A colación de esto, Mark era brillante en VOIP, redes, en la parte interna del sistema etc., y tenía un gran interés en teléfonos y telefonía, pero tenía experiencia limitada en sistemas de telefonía y como estos funcionaban, particularmente en el área de interfaces de hardware. Desde el inicio yo estaba y siempre estuve para ayudarlo en estas áreas, ambos estuvimos proporcionándonos información e implementando código de drivers y de switch (PBX). Nosotros y mas recientemente otros, hicimos un buen equipo trabajando en un objetivo común; el de alcanzar las implementaciones en tecnología de Telecom al público por un costo realista.

Desde la tarjeta ISA, yo diseñé la "Tormenta 2 PCI Quad T1/E1, la cual Mark vende como Digium T400P e E400P, y ahora Varion está vendiendo como V400P (Ambos T1 e E1). Todos los archivos del proyecto (incluyendo foto y archivos de ploteo) están disponibles en zaptelephony.org (<http://www.zapatatelephony.org>) para uso público. Pero diseños de mayor densidad están en camino,

Como cualquiera puede ver, con el trabajo dedicado de Mark (un montón de mi tiempo y de otras personas) los drives de Zaptel y el software de Asterisk, son tecnologías que vienen de un largo tiempo atrás y que están creciendo y mejorándose día tras día”.

1.3 ¿Porque Asterisk?

Yo recuerdo que mi primer contacto con Asterisk produjo en mi rechazo, como casi siempre nos pasa con la primera reacción de encontrarnos con algo nuevo que compete con aquello que no conocemos. Fue lo que sucedió, la primera vez que vi Asterisk. Este ofrecía una solución que yo estaba presentando en ese momento. De cualquier forma, yo siempre procuro levantar toda la información sobre las alternativas de los proyectos que hago y intento descubrir cuales son los puntos fuertes y los débiles de una solución como es Asterisk. Puedo decir que después de algunos días yo quede pasmado, sabía que Asterisk traería un cambio profundo en todo el mercado de las telecomunicaciones y voz sobre IP. Asterisk es el Apache de la telefonía. Déjenme entonces dar varias razones para elegir Asterisk y algunas limitaciones que aún existen desde la publicación de este libro.

1.3.1 Reducción extrema de costos

Si usted compara un PBX tradicional con Asterisk talvez la diferencia sea pequeña, principalmente por los costos de hardware y los teléfonos IP. Entretanto, Asterisk solo puede ser comparado a un PBX digital. Comparar una central analógica de cuatro líneas FXO y 16 ramales con Asterisk es injusto.

Cuando usted agrega recursos avanzados como Voz sobre IP, URA e DAC, la diferencia de costo es menor, en diversas oportunidades. Para dar un ejemplo, una única puerta de URA hoy con acceso a un mainframe, cotizada recientemente para un cliente nuestro, costó por lo menos 10 veces el precio que costaría con Asterisk.

1.3.2 Tener control de su sistema de telefonía

Este es uno de los beneficios mas citados, en vez de esperar que alguien configure su PBX propietario (algunos ni dan una indicación para el cliente final sobre esto), lo configura usted mismo. Total libertad e interfaces estándar. En fin de cuentas es LINUX y es libre.

1.3.3 Ambiente de desarrollo fácil y rápido

Asterisk puede ser programado en C con las APIs nativas, o en cualquier otro lenguaje usando AGI.

1.3.4 Rico y abundante en recursos

Como hemos resaltado desde el comienzo, pocos son los recursos encontrados en equipamientos PBX vendidos en el mercado que no puedan ser encontrados o creados en Asterisk. En él ya se puede encontrar todo lo que tiene un PBX tradicional...

1.3.5 Es posible proveer contenido dinámico por teléfono.

Como Asterisk es programado con C u otros lenguajes de dominio de la mayoría de los programadores, las posibilidades de proveer contenido dinámico por teléfono no tienen límites.

1.3.6 Plan de discado flexible y poderoso

Una vez más el Asterisk se supera. Si pensamos, la mayoría de las centrales, ni siquiera poseen la misma ruta de costo menor. Con Asterisk este proceso es simple y práctico.

1.3.7 Corre bajo Linux y es código abierto

Una de las cosas mas fantásticas de Linux es la comunidad de software libre. Cuando yo accedo al Wiki, o los foros de software en código abierto yo percibo que la adopción de nuevos usuarios es muy rápida, millares de preguntas cuestiones y relatos de problemas son enviados todos los días. Asterisk es probablemente uno de los softwares que más personas tienen disponibles para testear y probar. Esto torna el código estable y permite una rápida resolución de problemas.

1.3.8 Limitaciones de la arquitectura de Asterisk

Asterisk usa una CPU de servidor para procesar los canales de voz, en vez de tener un DSP (procesador de señales digitales) dedicado a cada canal. Todavía que esto permitió que el costo fuese reducido para las placas E1/T1, el sistema es muy dependiente de la performance de CPU. Mi recomendación es preservar al máximo la CPU de Asterisk, córranlo siempre en una máquina dedicada y prueben el dimensionamiento antes de implantarlo. En mi opinión, Asterisk debe ser siempre implementado en una VLAN específica para VoIP, cualquier tempestad de broadcasts causada por loops o virus puede comprometer su funcionamiento debido a el uso de CPU de las placas de red cuando este fenomeno se da.

1.4 Arquitectura de Asterisk

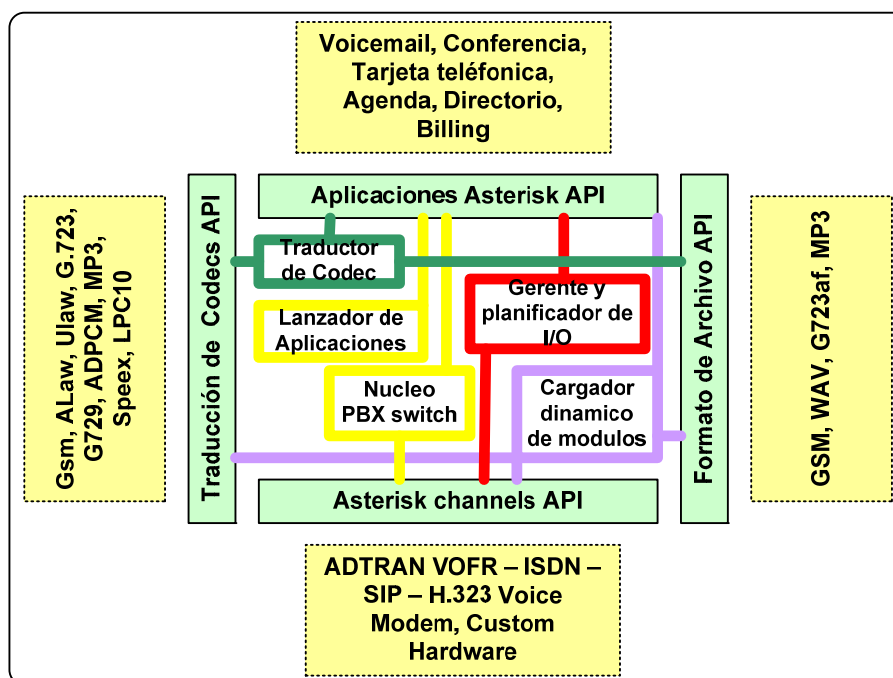


Figura 1.1 Arquitectura de Asterisk.

La figura de arriba nos muestra la arquitectura básica de Asterisk. Vamos explicar abajo los conceptos relacionados con este esquema como los canales, los codecs y las aplicaciones.

1.4.1 Canales

Un canal es el equivalente a una línea telefónica en la forma de un circuito de voz digital. Este generalmente consiste de una señal analógica en un sistema POTS¹ o alguna combinación de CODEC y protocolos de señalización (GSM con SIP, Ulaw con IAX). En un principio las conexiones de telefonía eran siempre analógicas y por eso, más susceptibles a ruidos y ecos. Más recientemente, buena parte de la telefonía paso para el sistema digital, donde la señal analógica es codificada en forma digital usando normalmente PCM (Pulse Code Modulation). Esto permite que un canal de voz sea codificado en 64 Kilobits/segundo sin ser compactado.

Canales para la red pública de telefonía

Digium zaptel. **chan_zap**

- TE410P – 4xE1/T1 (PCI-X 3.3V)
- TE412P – 4xE1/T1 (PCI-X 3.3V) con supresión de eco

¹ POTS – Plain Old Telephony System, sistema de telefonía convencional, basado normalmente en líneas analógicas.

- T405P – 4xE1/T1 (PCI-X 5V)
- TE407P – 4xE1/T1 (PCI-X 5V) con supresión de eco
- TE205P – 2xE1/T1 (PCI-X 5V)
- TE207P – 2xE1/T1 (PCI-X 5V) con supresión de eco
- TE210P – 2xE1/T1 (PCI-X 3.3V)
- TE212P – 2xE1/T1 (PCI-X 3.3V) con supresión de eco
- TE110P – 1xE1/T1 (PCI-X 3.3, 5V)
- B410P - - 4 x ISDN BRI
- TDM400P – 4 puertas analógicas FXS o FXO
- TDM800P – 8 puertas analógicas
- TDM2400 – 24 puertas analógicas FXS o FXO

Otros channel drivers para la red pública:

- **chan_phone**: Tarjetas quicknet phonejack y linejack.
- **chan_misdn, chan_visdn, chan_capi**: Tarjetas ISDN (Eicon, Beronet, Junghanns)
- **chan_unicall**: Tarjetas Digium con señalización MFC/R2
- **chan_ss7**: Tarjetas Digium con señalización 7 (ISUP).
- **chan_bluetooth**: Permite el uso de dispositivos Bluetooth

Canales para voz sobre IP

- **chan_sip**: Session Initiation Protocol.
- **chan_iax**: Inter-Asterisk Exchange Protocol 2.
- **chan_h323**: ITU H.323
- **chan_mgcp**: IETF MGCP.
- **chan_sccp**: Cisco SCCP

Canales internos para Asterisk

- **chan_agent**: Un canal de agente DAC. Dial String (Agent/agentnumber)
- **chan_console**: Console: Cliente de consola de Linux, driver para placas de sonido (OSS o ALSA). Dial string: console/dsp;
- **chan_local**: Pseudo canal. Hace un “loop” en el plan de discado. Dial string: Local/extension@context

1.4.2 Codecs y Conversores de CODEC

Obviamente es deseado colocar tantas llamadas cuanto sea posible en una red de datos. Esto puede ser hecho codificando en una forma que use menos banda ancha. Este es el papel de CODEC (COder/DECoder), algunos CODECs como el g.729 permite codificar a 8 Kilobits por segundo,

una compresión de 8 para 1. Otros ejemplos son ulaw, alaw, gsm, ilbc y g729.

Asterisk soporta los siguientes CODECs:

- G.711 ulaw (usado en EUA) – (64 Kbps).
- G.711 alaw (usado en Europa y Brasil) – (64 Kbps).
- G.723.1 – Modo Pass-through
- G.726 - 32kbps en Asterisk1.0.3, 16/24/32/40kbps
- G.729 – Precisa adquisición de licencia, a menos que este siendo usando en modo pass-thru.(8Kbps)
- GSM – (12-13 Kbps)
- iLBC – (15 Kbps)
- LPC10 - (2.5 Kbps)
- Speex - (2.15-44.2 Kbps)

1.4.3 Protocolos

Enviar datos de un teléfono a otro sería fácil si los datos encontrasen su propio camino para el otro teléfono destino. Desafortunadamente esto no sucede así, es preciso un protocolo de señalización para establecer las conexiones, determinar el punto de destino, y también cuestiones relacionadas a señalización de telefonía como el tono y tiempo de campanilla, identificador da llamada, desconexión etc. Hoy es común el uso de SIP (Session Initiated Protocol), muy usado hoy, y otros protocolos también muy en auge en el mercado como lo es el H.323, el MGCP y mas recientemente el IAX que es excepcional cuando se trata de trunking y NAT (Network Address Translation). Asterisk soporta:

- SIP
- H323
- IAXv1 y v2
- MGCP
- SCCP (Cisco Skinny).

1.4.4 Aplicaciones

Para conectar las llamadas de entrada con las llamadas de salida u otros usuarios de Asterisk son usadas diversas aplicaciones como es Dial, por ejemplo. La mayor parte de las funcionalidades de Asterisk son creadas en forma de aplicaciones como son estas el VoiceMail (buzón de voz), Meetme (conferencia), entre otras. Usted puede ver las aplicaciones disponibles en Asterisk usando el comando "core show applications" en la interface de línea de comando del Asterisk. Más allá de las aplicaciones en la versión central existen aplicaciones que pueden ser adicionadas a partir de archivos asterisk-addons y de terceros.

1.5 Visión General

Abajo vamos a mostrar algunos escenarios de uso de Asterisk y como ellos encajan en su modelo actual de telefonía.

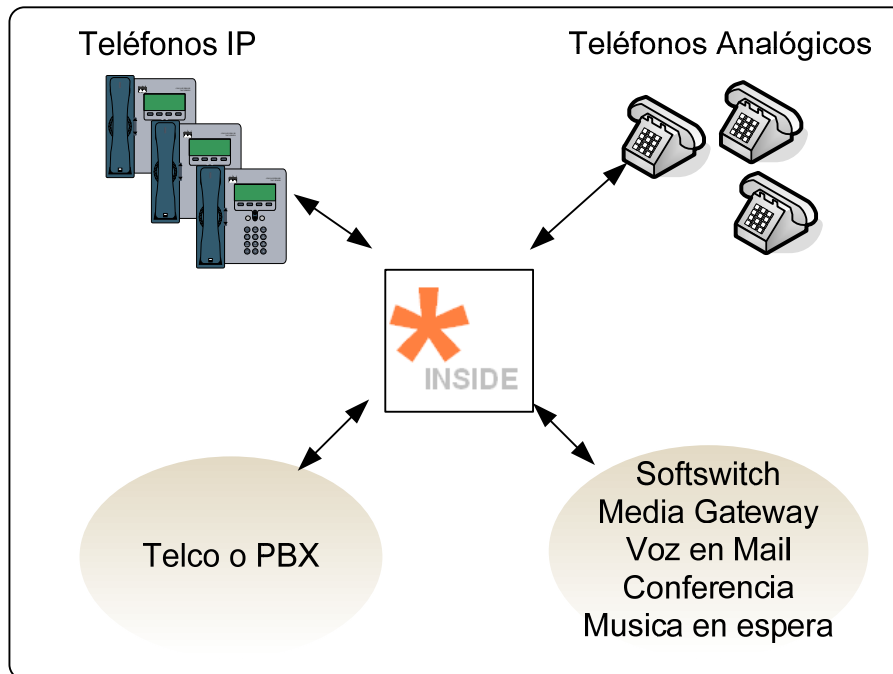


Figura 1.2 - Visión General de Asterisk

Dentro de una visión general, Asterisk es un PBX híbrido que integra tecnologías como TDM² y telefonía IP con funcionalidad de unidades de respuesta automática y distribución automática de llamadas. En este momento de la lectura del libro es probable que usted no esté entendiendo todos estos términos, pero a lo largo de los capítulos, estará cada vez más familiarizado con ellos. En la figura de arriba podemos ver que Asterisk se puede conectar a una operadora de telecomunicaciones o un PBX usando interfaces analógicas o digitales. Los teléfonos pueden ser IP, analógicos o ADSI que es un teléfono analógico con display digital.

² TDM – TDM - multiplexación por división de tiempo, toda a telefonía convencional está basada en este concepto, cuando hablamos en TDM estaremos refiriéndonos a circuitos T1 e E1. E1 es más común en Brasil e Europa, T1 es más usado en EUA.

1.6 Viejo y nuevo mundo

1.6.1 Telefonía usando el modelo de PBXs/Soft-switches

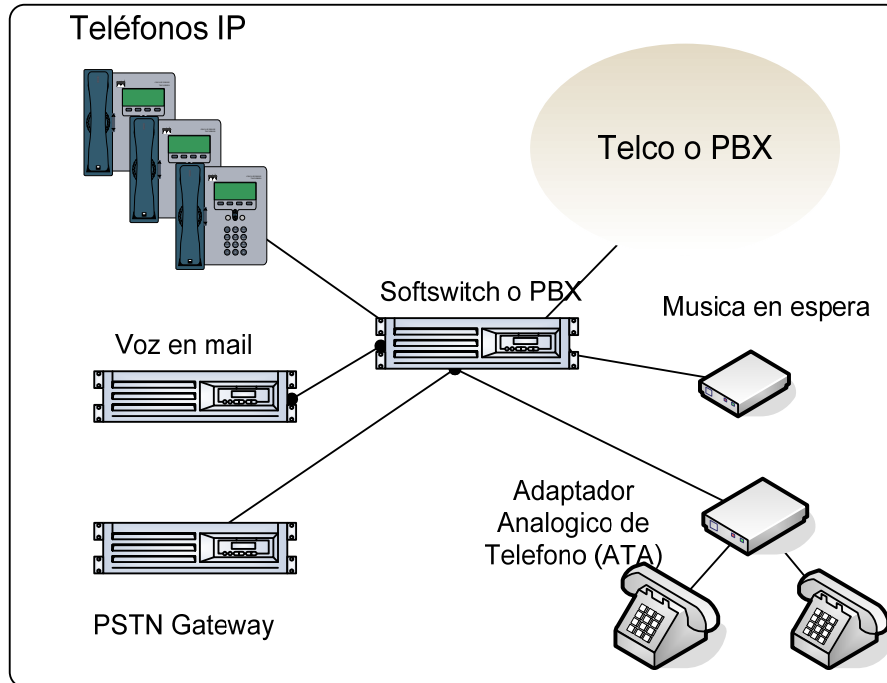


Figura 1.3 - PBX por software de tipo convencional (softswitch)

Es bastante común en estos días el uso de softswitches o PBXs. Entretanto la forma de comercialización de estos equipos sigue muchas veces la lógica mostrada en la figura 1.3. Los componentes son hechos de forma separada y muchas veces de diferentes fabricantes. Los costos de adquisición de cada uno de estos componentes es elevado y la integración muchas veces es difícil.

1.6.2 Telefonía a la manera de Asterisk

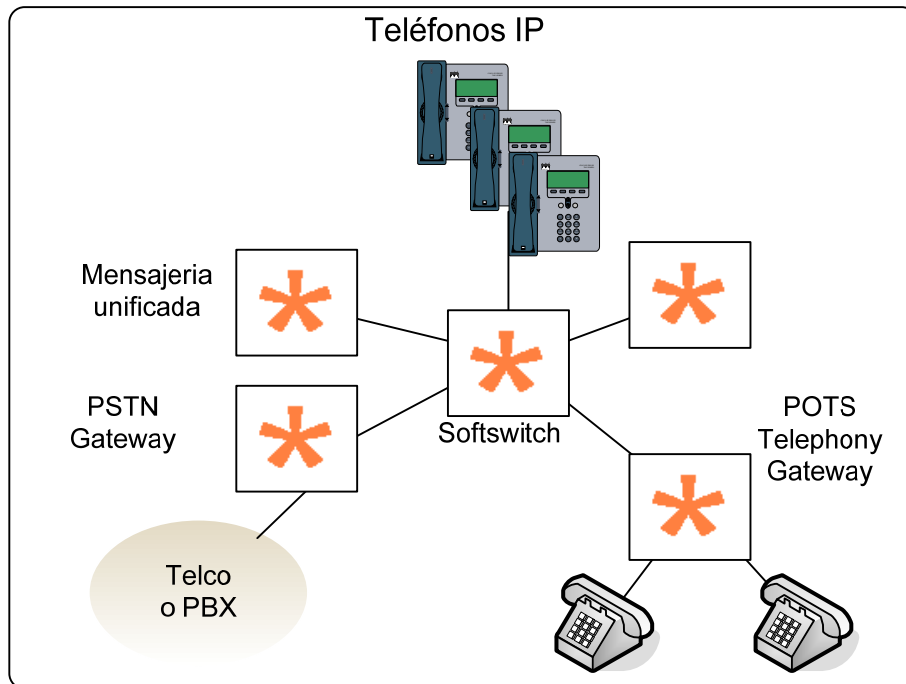


Figura 1.4 – Telefonía a la manera Asterisk

Asterisk realiza todas estas funciones de forma integrada, el licenciamiento es gratuito (GPL General Public License) y puede ser hecho en un único o en varios servidores de acuerdo con un dimensionamiento apropiado. Es increíble decir esto, pero puedo demostrar que es más fácil implementar Asterisk de que tomar, especificar y licenciar un sistema de telefonía convencional.

Exclusivo para *Investigación*

1.7 Un sistema para testes y desarrollo

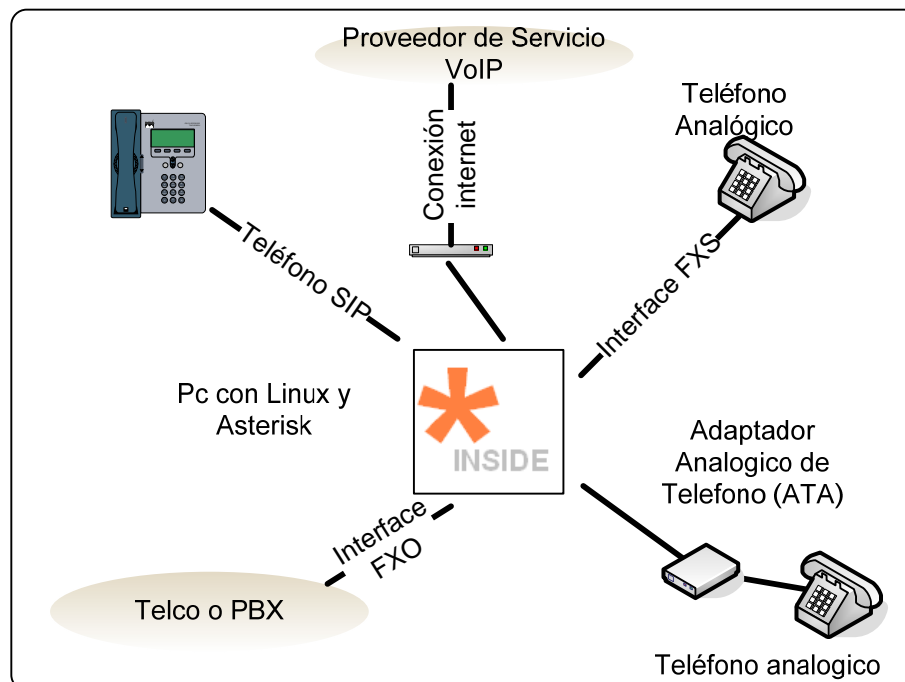


Figura 1.5 - El clásico PBX 1x1

En la figura 1.5, Usted ve un ejemplo de un PBX de un tronco y una línea. Este es uno de los sistemas más simples que usted puede construir con Asterisk. A pesar de tener poca utilidad práctica este permite que se conceptualicen algunos puntos importantes. En primer lugar el PBX 1x1 posee una placa FXO (Foreign Exchange Office) para ser ligada a las operadoras o a una interface de ramal. Usted puede adquirir una placa de esta de Digium con nombre TDM400P.

1.7.1 Primera opción: Un FXS y un FXO

Es la forma más simples y también más costosa.

1.7.2 Proveedor de servicio VoIP, softphone o ATA

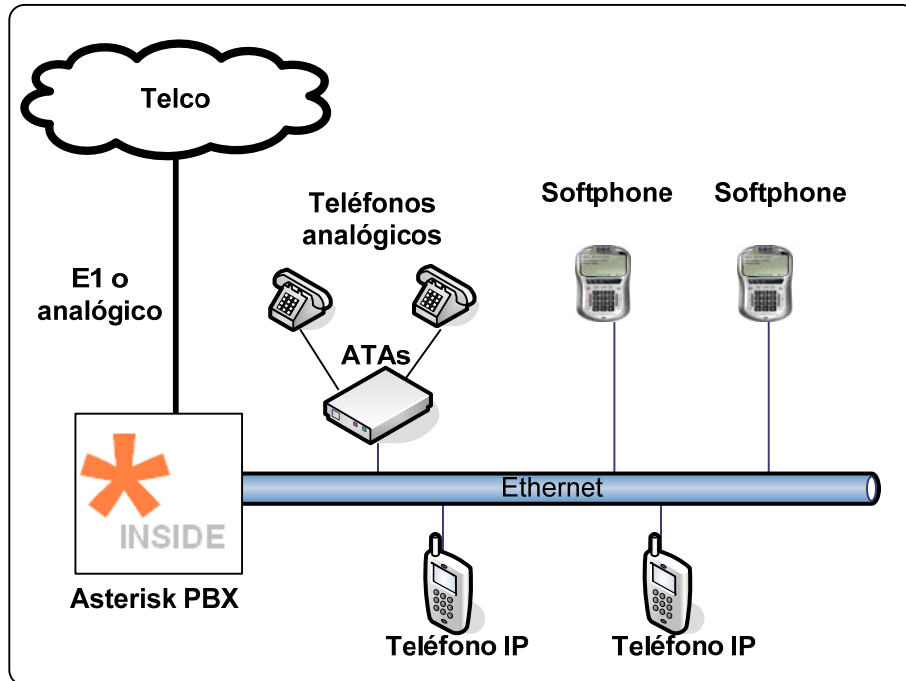
Es la forma más económica, pero no hay conexión hasta la red pública

1.7.3 Una interface FXO y un ATA o softphone

No es la forma más económica, pero si tiene una conexión hasta la red pública y si puede encontrar tarjetas con una interface FXO (X100P) con bajo costo.

1.8 Escenarios de uso de Asterisk

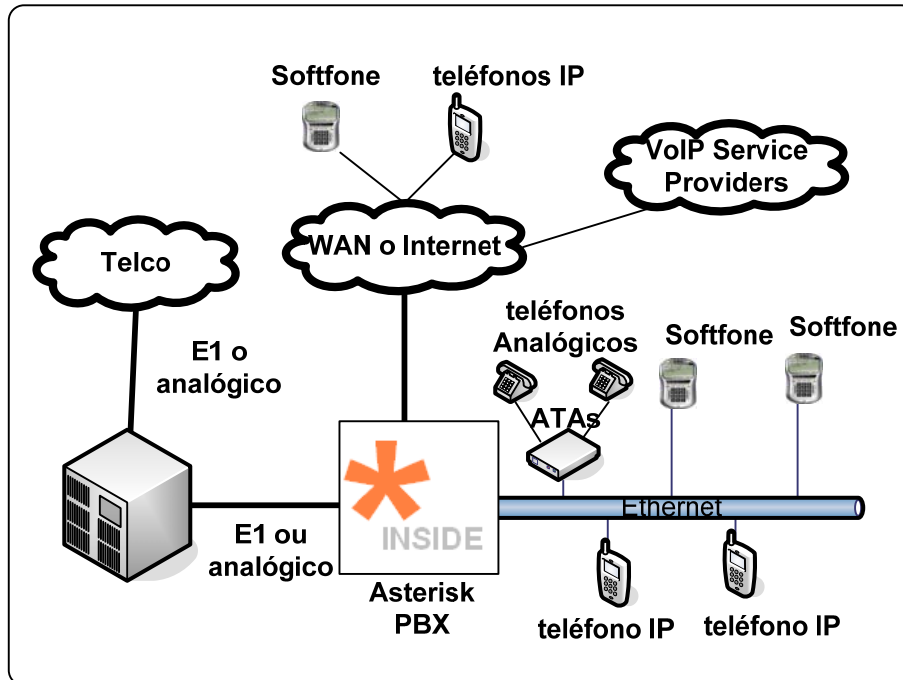
1.8.1 IP PBX



Lo escenario mas común es lo IP PBX para empresas pequeñas y medias. Si usted compara lo Asterisk as los PBX analógicos el es mas dispendioso. Todavía los recursos son mucho mas sofisticados. Si usted compara el Asterisk a otros PBX con los mismos recursos (buzón de voz, distribuidor automático de llamadas) Asterisk es menos costoso.

Exclusivo para el curso

1.8.2 Asterisk integrado con un PBX previamente instalado



Otro escenario mucho común es el Asterisk lo Asterisk integrado con un PBX que esta previamente instalado. Normalmente es menos costoso poner la telefonía IP con Asterisk que actualizar lo PBX convencional.

1.8.3 Intercomunicación de filiales con casa matriz

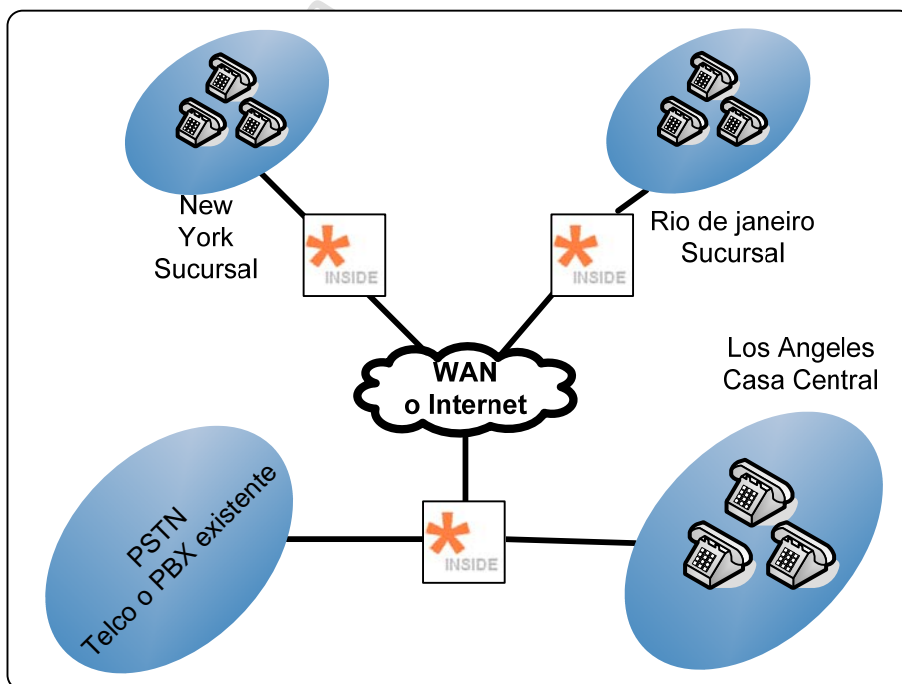
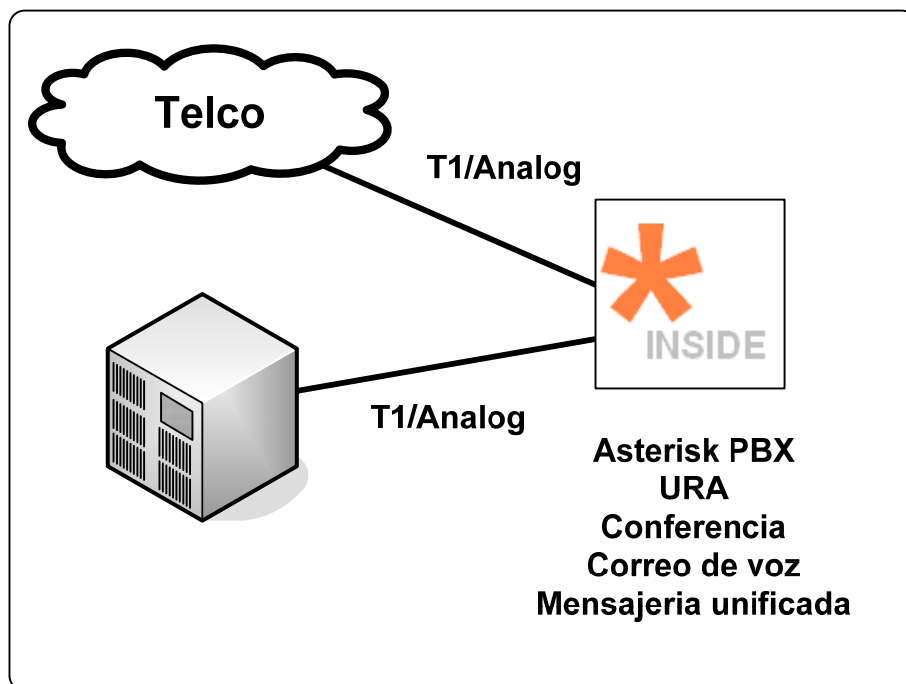


Figura 1.7 – Intercomunicación de casa matriz con filiales

Asterisk posee funcionalidad de un gateway de media. El puede convertir las señales analógicas (FXS, FXO) o digitales (ISDN) viniendo de la central telefónica, o de los teléfonos de cliente en voz sobre IP y transmitir por la red corporativa de datos. La convergencia propicia la reducción del número de circuitos y un mejor aprovechamiento de los recursos. Los proyectos más comunes son conocidos como "Toll-Bypass" (Contorneando la tarificación), pues se eliminan los costos de operadora de larga distancia de los teléfonos de las filiales de la empresa.

1.8.4 Servidor de aplicaciones



Vamos a conceptualizar de una forma un poco más detallada:

Buzón de voz – Permite que cuando el usuario no atiende el teléfono por estar ocupado o ausente, reciba un "prompt" solicitando que deje un mensaje. Es semejante a una secretaria electrónica o caja de mensajes de un celular. Asterisk presenta esta funcionalidad, sin costo adicional.

Sistema de mensajería unificada – Es un sistema donde todos los mensajes son encaminados para un único lugar, por ejemplo, la casilla de buzón de un usuario. En este caso los mensajes de e-mail, junto con los mensajes de buzón de voz y fax serían encaminados para la casilla postal del usuario. En Asterisk también se da la posibilidad de hacerlo.

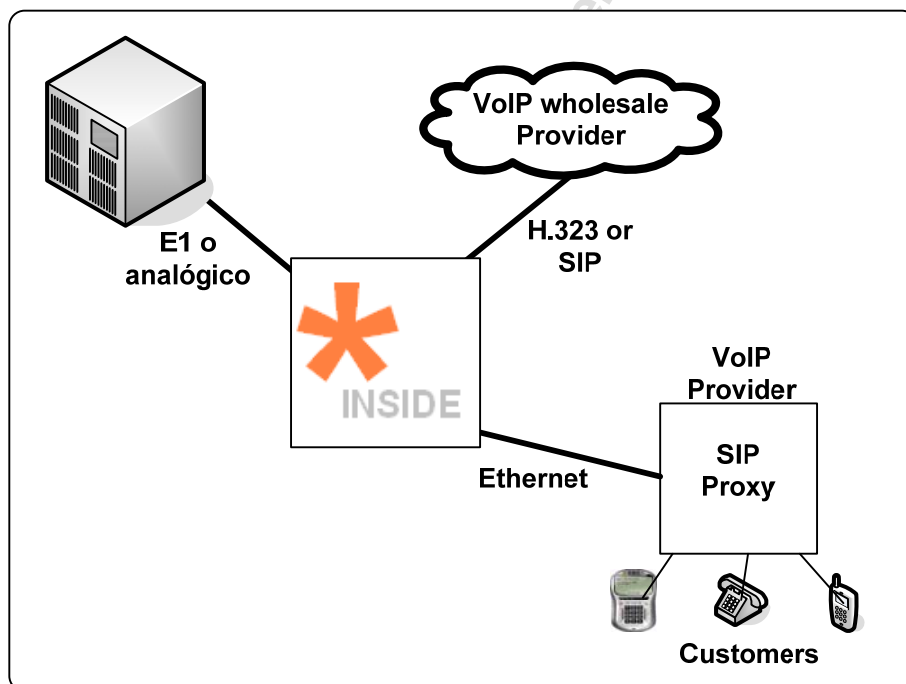
Unidad de respuesta automática – Para mí este es uno de los puntos más altos de Asterisk, el permite crear una URA muy fácil de personalizar. Esto permite que usted, por ejemplo, cree un sistema de consulta de stock y precios para los vendedores, un sistema de atención a disposición de los pedidos e innumerables aplicaciones. Es bueno recordar que al

contrario de los EUA y Europa, la cantidad de computadoras por latinoamericanos es relativamente pequeña y el teléfono aun es uno de los medios de acceso mas universales.

Sala de Conferencia – Permite que varios usuarios hablen en conjunto. Usted escoge un ramal para armar la sala de conferencia y todos los que discan para ella están inmediatamente conectados.

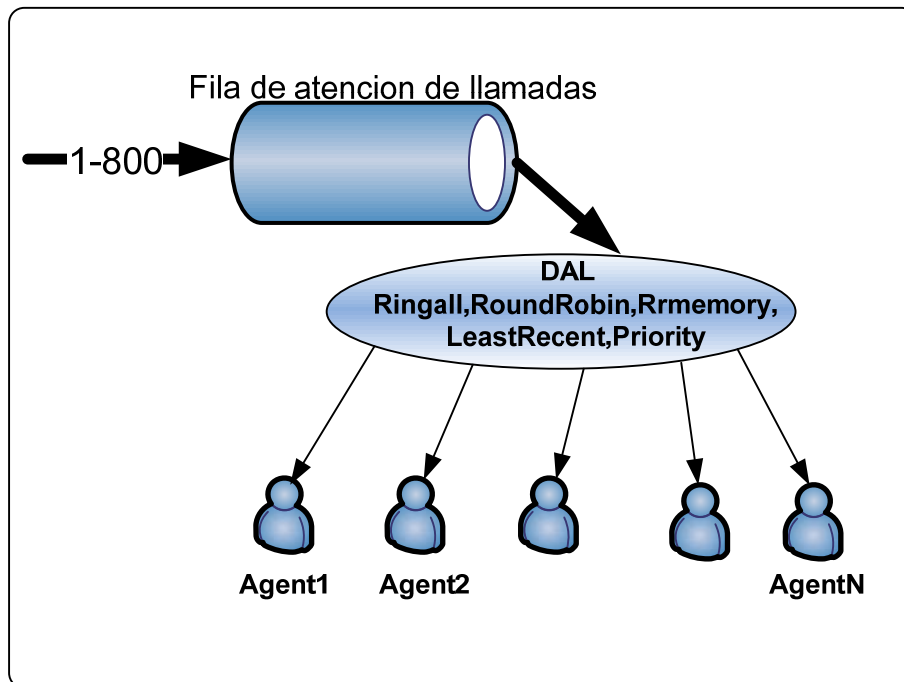
Estas son algunas de las funcionalidades actuales de Asterisk, nuevas aplicaciones están surgiendo a cada día. Usando recursos como AGI (Asterisk Gateway Interface) que es muy semejante al CGI (Common Gateway Interface), las posibilidades de programación son muy amplias, el lenguaje de programación puede ser escogido por el desarrollador. Creo que serán cada vez más comunes los portales de voz, para diseminar la información por un conjunto aun mayor de personas.

1.8.5 Media Gateway



Media Gateway – Un media gateway permite que sus conexiones en telefonía analógica puedan ser convertidas en Voz sobre IP, por ejemplo, y transmitidas por la red de datos ante otra oficina sin pasar por la tarificación de la red pública. Este es el punto número uno de la implementación de voz sobre IP, reducir la cuenta. Si usted tiene un Asterisk en cada filial, usted puede interconectarlos usando IAX trunked, una de las mejores tecnologías de conexión de PBX por IP. Esto es lo que me da miedo, que todavía de ser un software libre, este recurso de protocolo en particular es superior al que yo tengo conocido como equipos basados en tecnologías propietarias

1.8.6 Contact Center



Distribuidor automático de llamadas y fila de atención de llamadas

Este es uno de los conceptos menos obvios de la telefonía. La primera vez que yo vi. Esto, yo pensé: pero yo ya tengo esto en mi central telefónica, ella distribuye para varios ramales, el primero que se atiende deja de hacer sonar los otros, porque tanto "OOHH" para esta cosa de DAC. La respuesta es simple, en un DAC, las personas normalmente se autentifican en una fila de atención de llamadas para recibir dichas llamadas, el distribuidor verifica que usuario está con el teléfono libre antes de pasar la llamada. Si ningún operador estuviese libre este asegura la llamada en una fila con aquella "musiquita" y un mensaje como "Usted esta comunicado con..... Su llamada es muy importante....." (Que nosotros lo adoramos a Ud!!) etc. Cuando el primer telefonista es liberado, es DAC quien pasa la comunicación. DAC es fundamental en cualquier sistema de atención. Hay mucho mas sobre DAC de lo que aquí esta escrito, el sistema de llamada rotativa puede ser mucho mas sofisticado. Un DAC normalmente tiene un costo elevado en la mayoría de las plataformas convencionales.

Dialer automático – Esto es muy útil en telemarketing, se puede programar el sistema para discado automático y distribuir en una fila. Pero esta es una tecnología que es vendida separadamente en otros PBX. En Asterisk usted puede programar un discado y existen diversos ejemplos de dialers disponibles en Internet.

1.9 Buscando informaciones relativas a Asterisk.

En esta sección vamos mostrar las principales Fuentes de informaciones relativas a Asterisk

1. www.asterisk.org

Soporte	-> http://www.asterisk.org/support .
Base de datos de informaciones	-> http://kb.digium.com/
Forum	-> http://forums.digium.com
Errores	-> http://bugs.digium.com

2. www.asteriskexperts.com.br

3. www.asteriskdocs.org

4. www.asteriskguru.com

5. www.voip-info.org (la mayor wikipedia relativa a Asterisk)

6. <http://svn.digium.com/view/asterisk/branches/1.4/doc/>

1.10 Sumario

Asterisk es un software con licenciamiento GPL, que transforma un PC común en una poderosa central telefónica. Fue creado por Mark Spencer de Digium que comercializa el hardware de telefonía. El Hardware de telefonía también es abierto y fue desarrollado por Jim Dixon en el proyecto Zapata Telephony.

La arquitectura de Asterisk se compone básicamente de:

- **CANALES** que pueden ser analógicos, digitales y/o Voip.
- **PROTOCOLOS** de comunicación como SIP, H323, MGCP y IAX que son responsables por la señalización de telefonía.
- **CODECS** que hacen la codificación de voz de un formato para otro, permitiendo que sea transmitida con compresión de hasta ocho veces (G729a).
- **APLICACIONES** que son responsables por la funcionalidad del PBX.

Asterisk puede ser usado en innumerables aplicaciones, desde un PBX para una pequeña empresa hasta sistemas de respuesta automática de alta densidad.

1.11 Cuestionario

1. Marque las opciones correctas.

Asterisk tiene cuatro componentes básicos de arquitectura

- CANALES
- PROTOCOLOS
- AGENTES
- TELEFONOS
- CODECS
- APLICACIONES

2. Si fuera necesario crear un PBX con 4 enlaces y ocho teléfonos, ud. puede usar un PC con Linux y 3 placas TDM400P una con cuatro canales FXO y 2 con cuatro canales FXS cada una. La afirmación es:

- CORRECTA
- INCORRECTA

3. Un canal FXS genera tono de discado, en cuanto un canal FXO recibe el tono desde la red pública o de algún otro PBX. La afirmación es:

- CORRECTA
- INCORRECTA

4. Marque las opciones correctas. Asterisk permite los siguientes recursos:

- Unidad de Respuesta Automática
- Distribución automática de llamadas
- Teléfonos IP
- Teléfonos Analógicos
- Teléfonos digitales de cualquier fabricante.

5. Para tener música en espera Asterisk necesita de un CD Player conectado en un ramal FXO. La afirmación es:

- CORRECTA
- INCORRECTA

6. Es responsable por la atención automática de clientes, normalmente suena un "prompt" y espera que usuario seleccione una opción. En algunos casos puede ser usada en conjunto con un banco de datos y conversión de texto para ser hablado.

Estamos hablando aquí de una:

- URA
- IVR
- DAC
- Unified Messaging

7 – Marque las opciones correctas, Un banco de canales es conectado al Asterisk a través de una interface:

- E1
- T1
- FXO
- FXS

8 – Marque a opción correcta. Un canal E1 soporta ____ canales de telefonía en cuanto un T1 soporta ____ canales.

- 12, 24
- 30, 24
- 12,12
- 1,1

9 – En las plataformas de telefonía convencional, normalmente URA, DAC y Buzón de voz están incluidos en el PBX. Esta afirmación es:

- CORRECTA
- INCORRECTA

10 – Marque las opciones correctas, Es posible intercomunicar, usando Asterisk, varias filiales a través de voz sobre IP reduciendo el uso de comunicaciones de larga distancia. En una filial:

- Asterisk puede ser la central telefónica para todos los usuarios.
- Asterisk puede integrar una central telefónica existente
- Pueden ser usados apenas teléfonos IP ligados a un Asterisk centralizado
- Redundancia y confiabilidad no son importantes cuando se conectan IP phones.

Bajando e instalando Asterisk

Este capítulo va a ayudarlo a preparar su sistema para la instalación de Asterisk. Este funciona en muchas plataformas y sistemas operativos, mas nosotros escogemos mantener las cosas simples y permanecer en una única plataforma y distribución de Linux. Vamos usar la distribución de Debian (Sarge 3.1) en este libro. Las instrucciones abajo pueden funcionar con otra distribución, pero esto no fue testado. Asterisk es conocido por funcionar en la mayoría de las distribuciones. Testeamos en Debian, Suse, Fedora, Ubuntu y Slackware.

2.1 Objetivos del capítulo

Objetivos

Para el final de este capítulo usted estará apto para:

- Dimensionar los requerimientos de hardware para Asterisk
- Instalar Linux con las dependencias necesarias.
- Bajar una versión estable de Asterisk (Asterisk stable) usando FTP.
- Compilar Asterisk
- Correr Asterisk y explica las opciones de ejecución.
- Aprender como arrancar Asterisk al inicio de su computadora.

2.2 Hardware Mínimo

Asterisk puede ser intensivo en el uso del procesador, pues el usa el propio procesador del PC para hacer el procesamiento de los canales de voz. Si usted estuviese construyendo un sistema complejo con carga elevada es importante entender este concepto. Para construir su primer PBX un procesador compatible con Intel que sea mejor que un Pentium 300Mhz con 256 MB RAM es lo suficiente. Asterisk no requiere mucho espacio en disco, cerca de 100 MB compilados, más código fuente, buzón de voz, grabaciones y todos requieren espacio.

Si usted usa apenas VOIP, ningún otro hardware es necesario. Se puede usar softphones como los de Counterpath (X-Lite) y hacer enlaces con operadoras de voz sobre IP. Una lista de las operadoras de voz sobre IP en Brasil puede ser encontrada en:

http://www.voipcharges.com/providers.php?_filter=1&country_sid=31

Cuidado!

Algunas aplicaciones de Asterisk como el "Meetme" requieren una fuente de clock para proveer la temporización. Normalmente la fuente de clock de Asterisk es una placa TDM. Si su sistema no tiene una placa TDM, usted puede usar un driver ztdummy.

2.2.1 Montando su sistema

El hardware necesario para Asterisk no es muy complicado. Usted no precisa de una placa de vídeo sofisticada o periféricos. Puertos seriales, paralelos y USB pueden ser completamente deshabilitados. Una buena tarjeta de red es esencial. Si usted estuviese usando una de las placas de Digium, seria bueno verificar las instrucciones de su placa-madre para determinar si los Slots PCI soportan estas placas. Muchas placas-madres compartirán interrupciones en slots PCI. Conflicto de interrupciones es una fuente potencial de problemas de calidad de audio en Asterisk. Una manera de liberar IRQs es deshabilitar desde el BIOS todo lo que no fuera necesario.

2.2.2 Problemas con IRQ compartidas

Muchas placas de telefonía como la X100P pueden generar grandes cantidades de interrupciones, atenderlas toma tiempo. Los drivers pueden no conseguir hacerlo a tiempo si otro dispositivo estuviese procesando la misma IRQ compartida y la línea de IRQ no podrá recibir otra interrupción. Tiende a funcionar mejor en sistemas multiprocesador. En sistemas monoprocesados usted puede tener muchas perdidas de interrupción y clock desfasado. Cualquiera de las placas de Digium y otras placas de telefonía pueden estar sujetas al mismo problema. Como la entrega precisa de IRQs es una necesidad primaria en telefonía, usted no debe compartir IRQs con nada. No siempre esto ocurre, pero usted debe prestar atención al problema. Si usted está usando un computador dedicado para Asterisk, deshabilite el mayor número de dispositivos que usted no va a usar.

La mayoría de las BIOS permite que usted manualmente designe las IRQs. Váyase hasta la BIOS y observe en la sección de IRQs. Es posiblemente cierto que usted consiga configurar las interrupciones manualmente por ranuras.

Una vez iniciado el computador, vea en `/proc/interrupts` las IRQs designadas.

```
#cat /proc/interrupts
CPU0
0: 41353058 XT-PIC timer
1: 1988 XT-PIC keyboard
2: 0 XT-PIC cascade
3: 413437739 XT-PIC wctdm <-- TDM400
4: 5721494 XT-PIC eth0
7: 413453581 XT-PIC wcfxo <-- X100P
8: 1 XT-PIC rtc
9: 413445182 XT-PIC wcfxo <-- X100P
12: 0 XT-PIC PS/2 Mouse
14: 179578 XT-PIC ide0
15: 3 XT-PIC ide1
NMI: 0
ERR: 0
```

Arriba usted puede ver las tres placas de Digium cada una en su IRQ. Si este fuera el caso, usted puede ir en frente e instalar los drivers de hardware. Si no fuera el caso, vuelva a la BIOS hasta que las placas no están compartiendo las IRQs.

2.3 Escogiendo una distribución de Linux.

Asterisk fue originalmente desarrollado para correr en Linux, Ahora puede ser usado en BSD y OS X. Todavía esto, las placas PSTN de Digium fueron diseñadas para trabajar con Linux i386. Si usted fuese nuevo con Asterisk procure usar Linux.

2.3.1 Requisitos de Linux

Varias distribuciones como RedHat, Mandrake, Fedora, Debian, Slackware y Gentoo fueron usadas con Asterisk.

2.3.2 Paquetes necesarios.

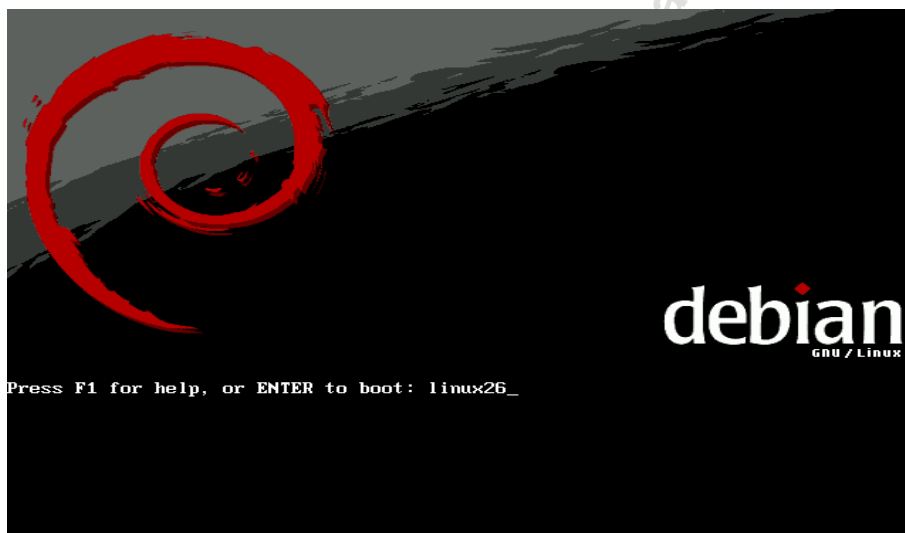
Antiguamente existían algunos paquetes que eran necesarios para instalar Asterisk como el `readline` y `readline-devel` que no son necesarios hoy. No existe hardware especial tal como una placa de sonido y el único paquete necesario es el propio Asterisk. Si usted estuviera usando hardware de Digium o `ztdummy`, usted va a precisar del paquete `zaptel`. El paquete `zaptel` es necesario para que algunas aplicaciones sean incluidas en tiempo de compilación. Si usted escoge compilar Asterisk y no el `zaptel`, pero descubre que esta faltando una aplicación relacionada al paquete `zaptel` (Como el `Meetme()`), usted tendrá que compilar el `zaptel` y entonces sera necesario recompilar Asterisk para que la aplicación sea incluida. Para interfaces T1 y E1 el paquete `libpri` es necesario. *Bison* es necesario para compilar el Asterisk. Los paquetes de desarrollo `ncurses` y

ncurses-development son necesarios si usted quisiera construir nuevas herramientas (Como el astman). Las bibliotecas zlib y zlib-devel son necesarias ahora para compilar. Esto se debe a la adhesión del protocolo DUNDi (Distributed Universal Number Discovery).

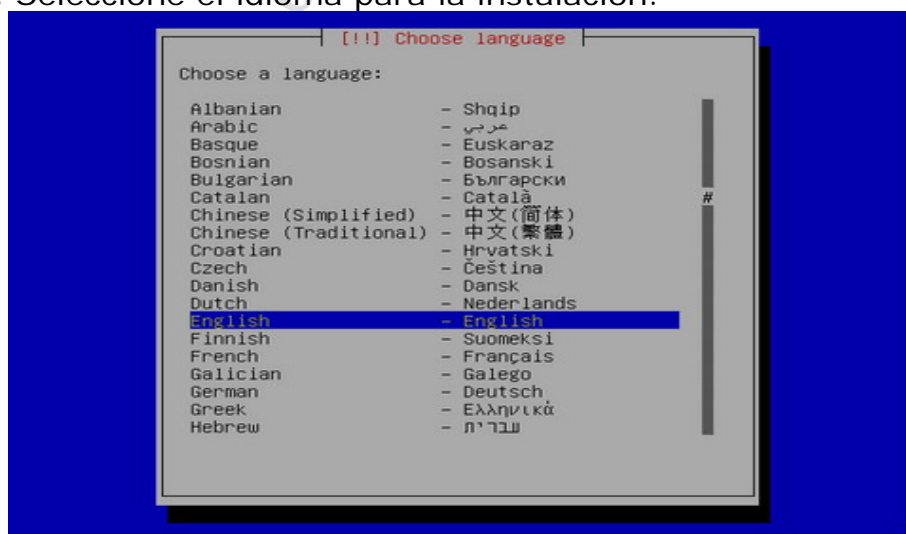
2.4 Instalando Linux para atender al Asterisk.

Usamos Debian con Kernel 2.6 para la instalación de Asterisk con Linux. Escogemos esta distribución por la gran aceptación que tiene y por ser una de las distribuciones soportadas por la Digium. Abajo siguen las instrucciones de instalación de Asterisk a partir de cero.

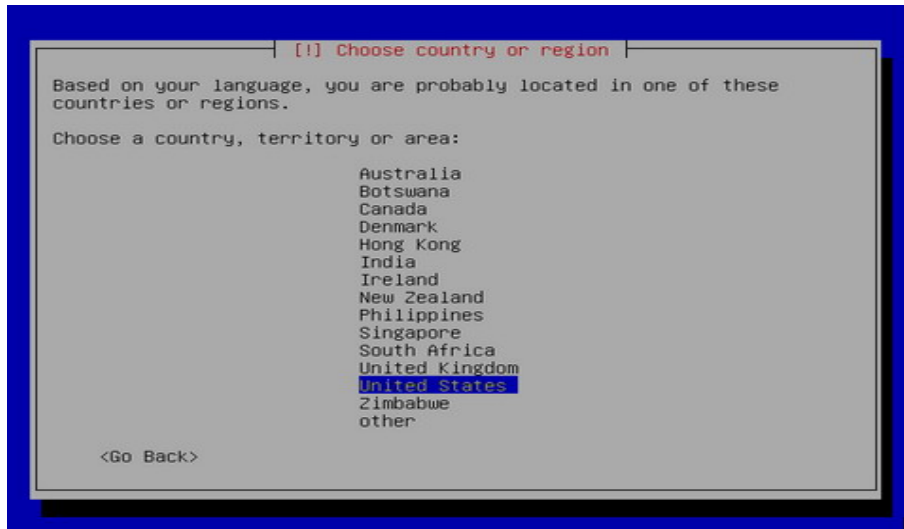
Paso 1: Coloque el CD de Debian en la lectora de cdrom e inicialice el booteo de su PC desde allí con opción linux26



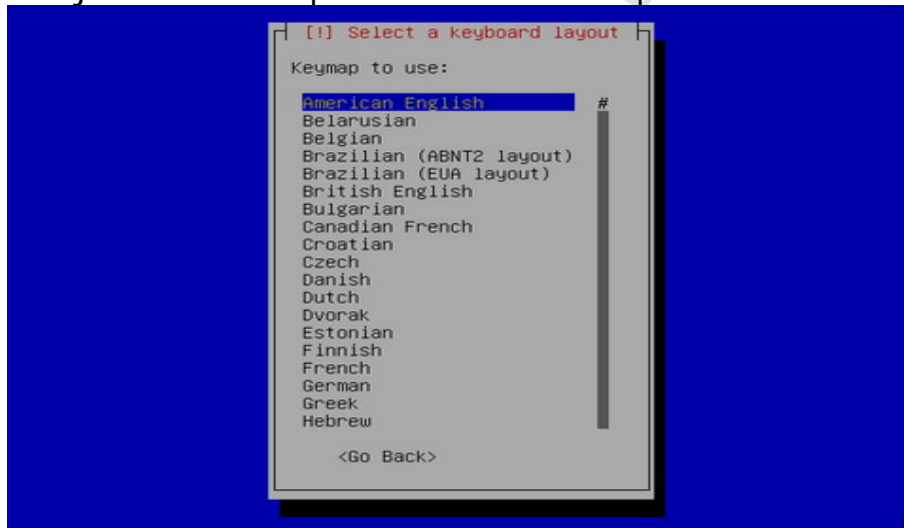
Paso 2 : Seleccione el idioma para la instalación.



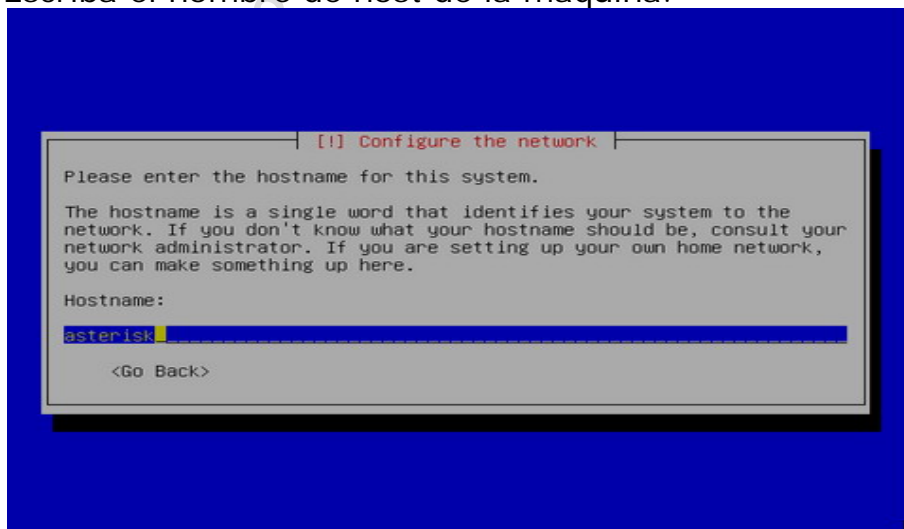
Paso 3: Seleccione el país de idioma seleccionado anteriormente.



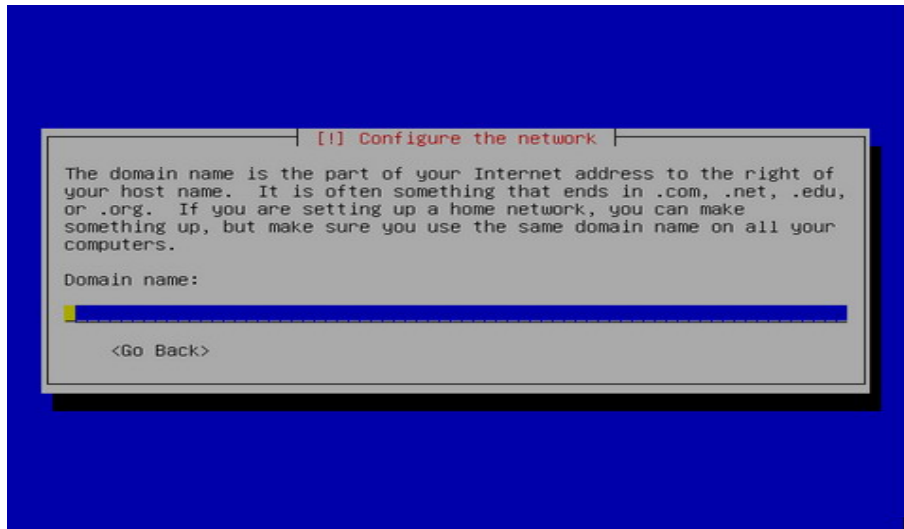
Paso 4: Escoja ahora el mapa de teclado correspondiente.



Paso 5: Escriba el nombre de host de la máquina.



Paso 6: Escriba el dominio al cual este equipo es parte.



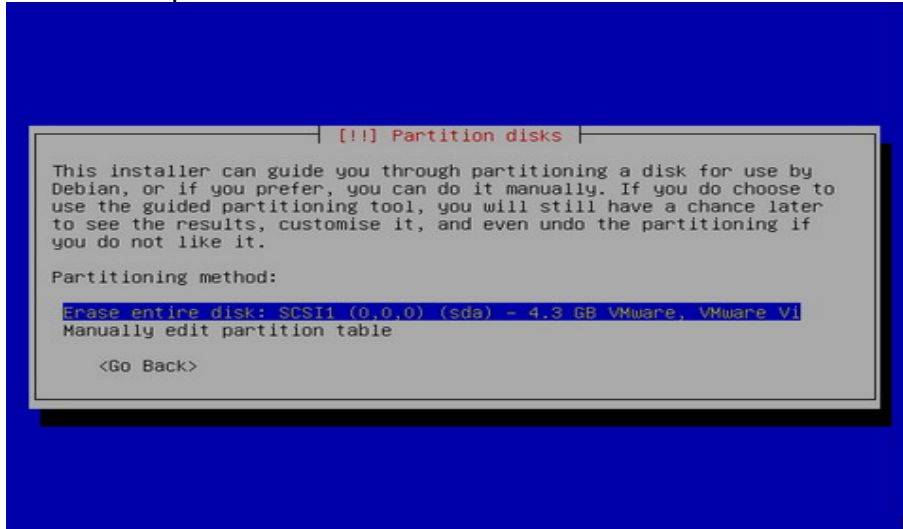
Paso 7: Será hecha la configuración de particionamiento de disco del equipo, apenas confirme las opciones presentadas.

Cuidado !

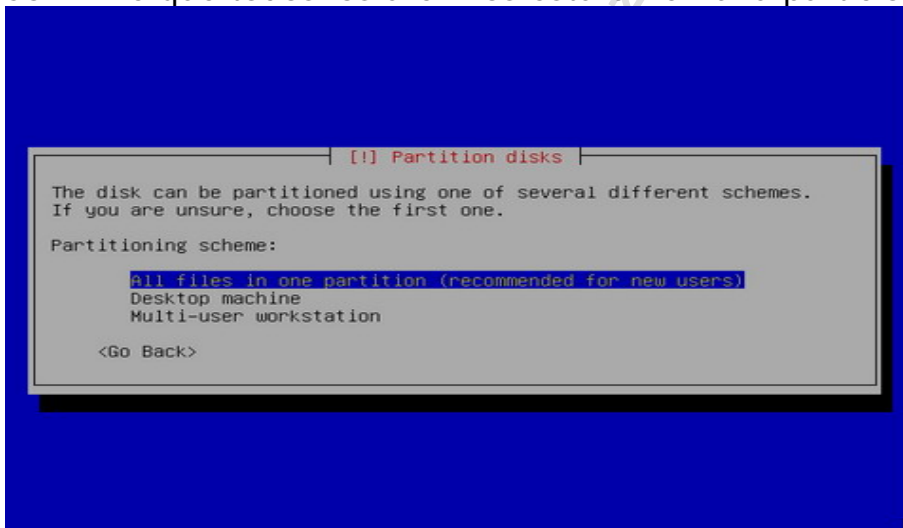
Todos los datos de su disco rígido serán perdidos. Instale en un PC nuevo o que usted pueda formatear !!!

Exclusivo para Vicente Miguel Perez Are

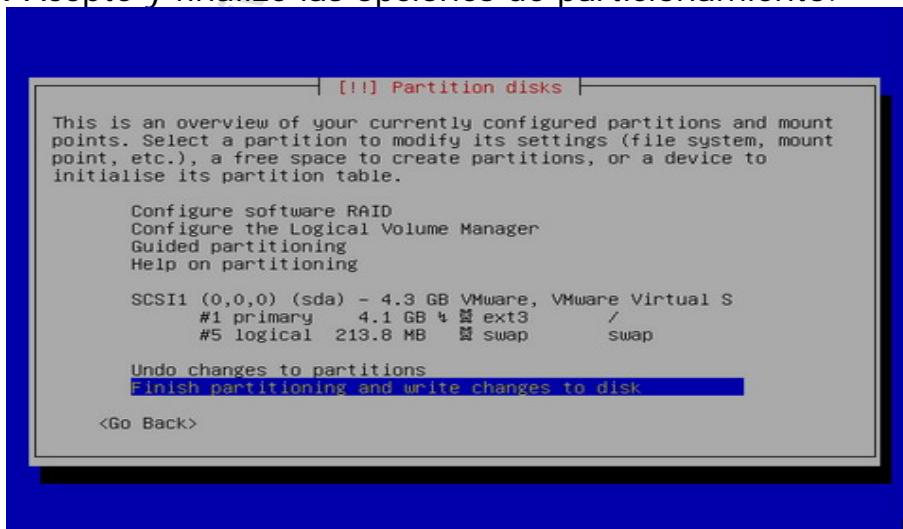
Paso 8: Confirme que usted va a formatear todo el disco.



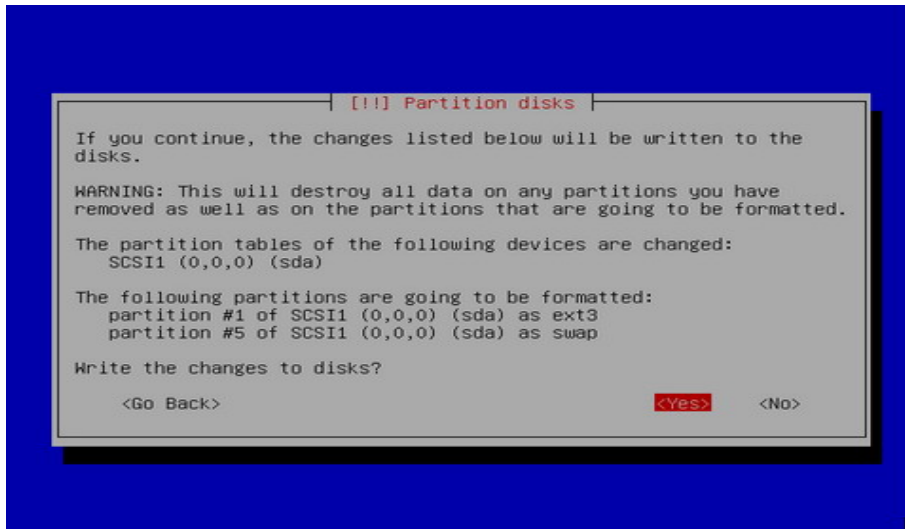
Paso 9: Confirme que todos los archivos estaran en una particion.



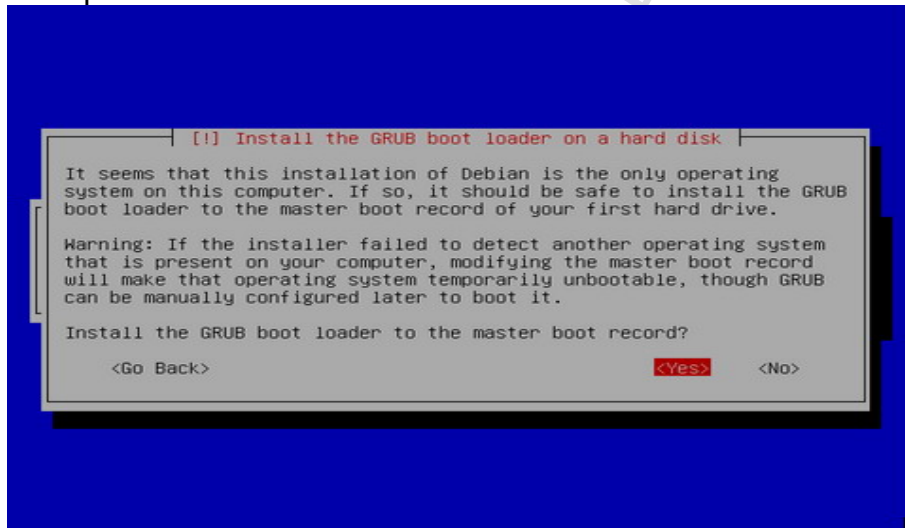
Paso 10: Acepte y finalice las opciones de particionamiento.



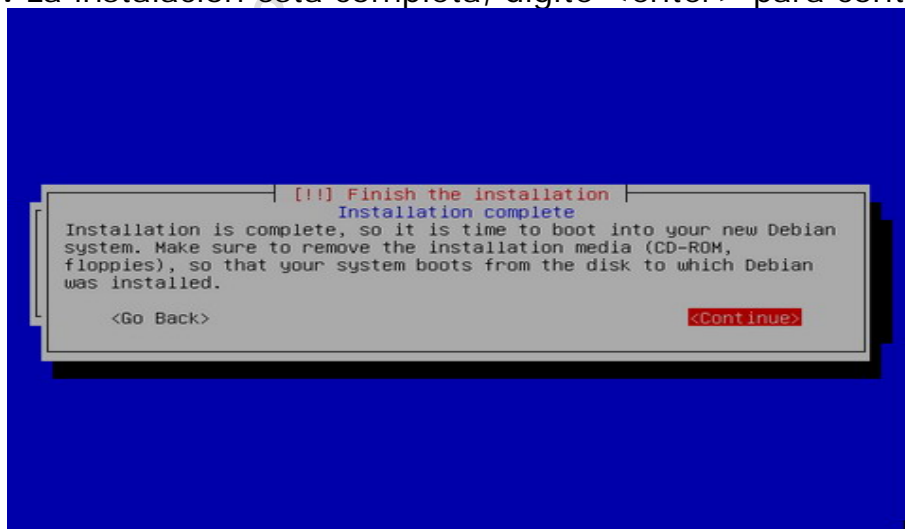
Paso 11: Confirme nuevamente.



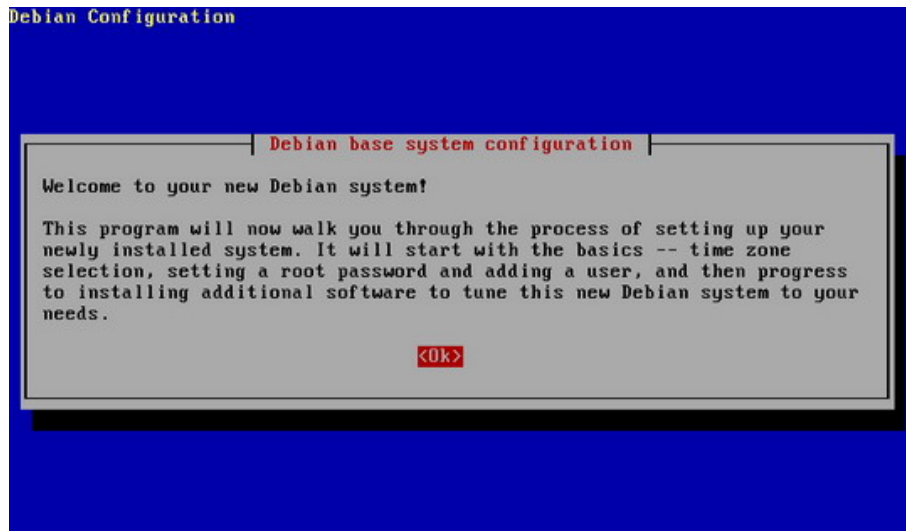
Paso 12: Acepte la instalación de GRUB.



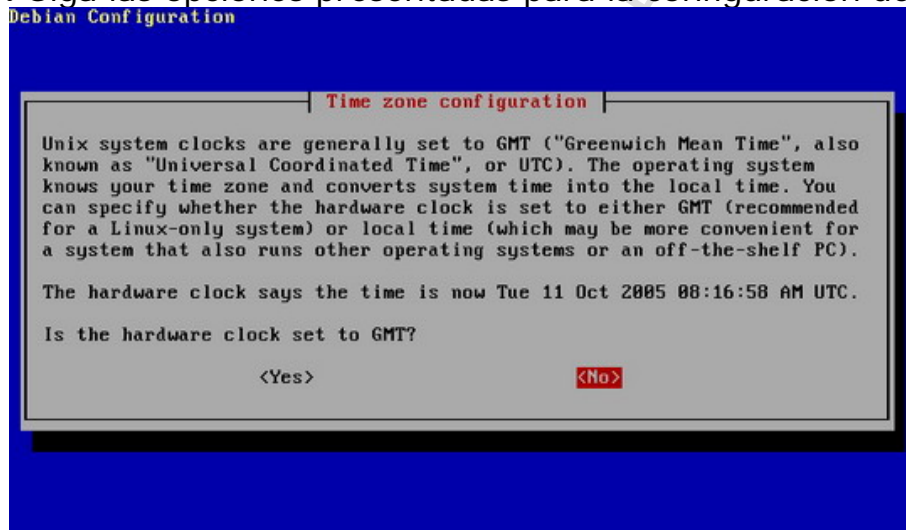
Paso 13: La instalación está completa, digite <enter> para continuar.



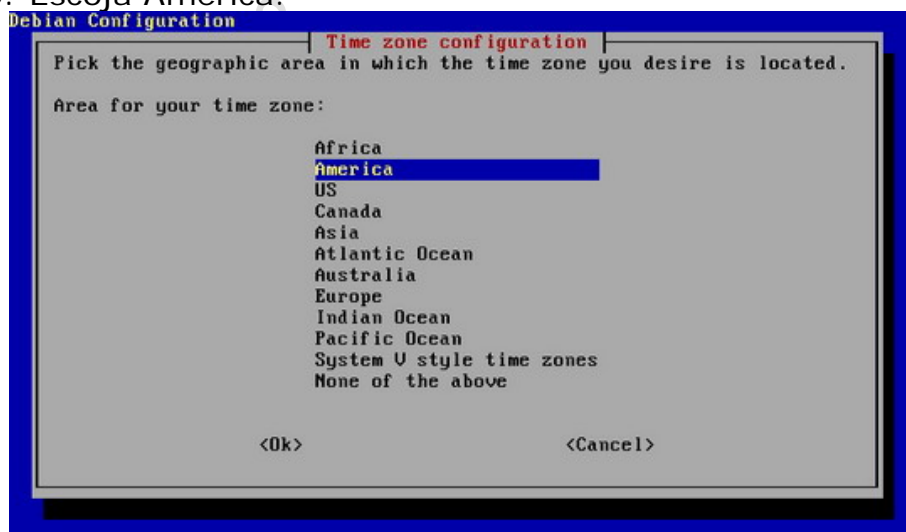
Paso 14: Ahora la máquina se reiniciara, remueva el CD.



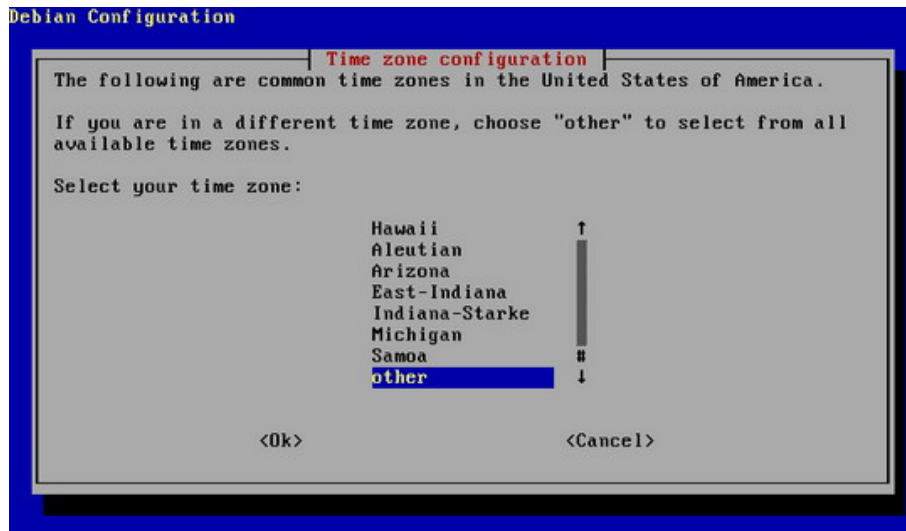
Paso 15: Siga las opciones presentadas para la configuración del horario.



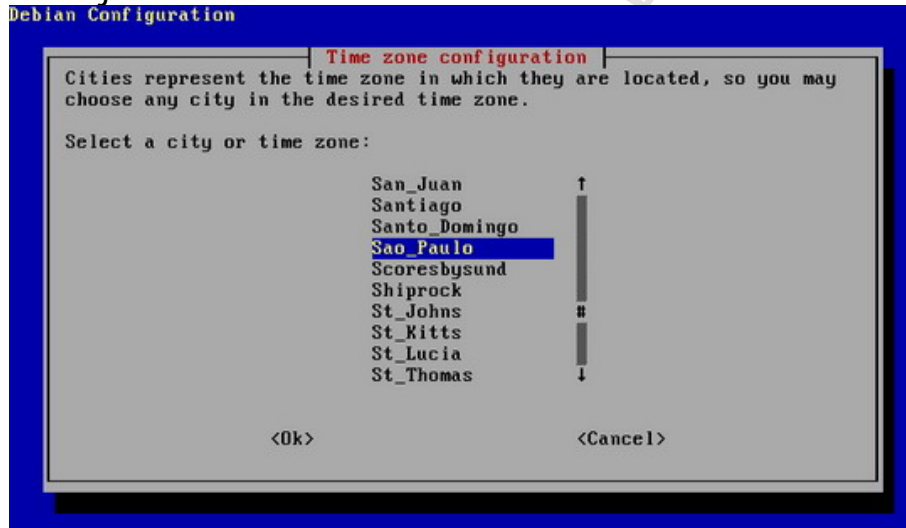
Paso 16: Escoja América.



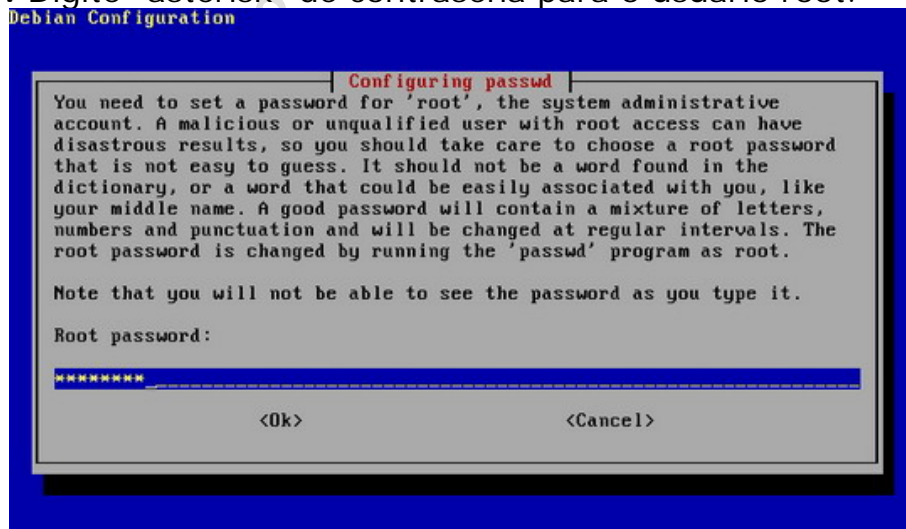
Paso 17: Escoja "Other".



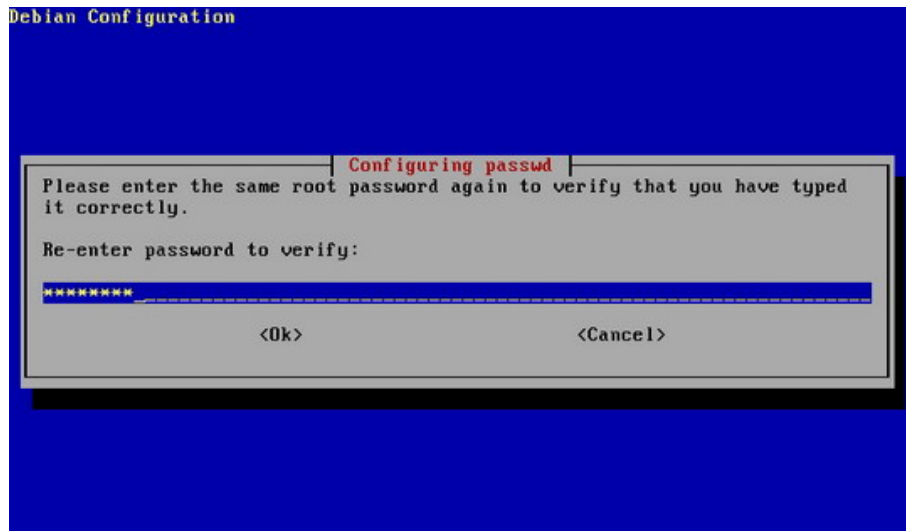
Paso 18: Escoja su ciudad.



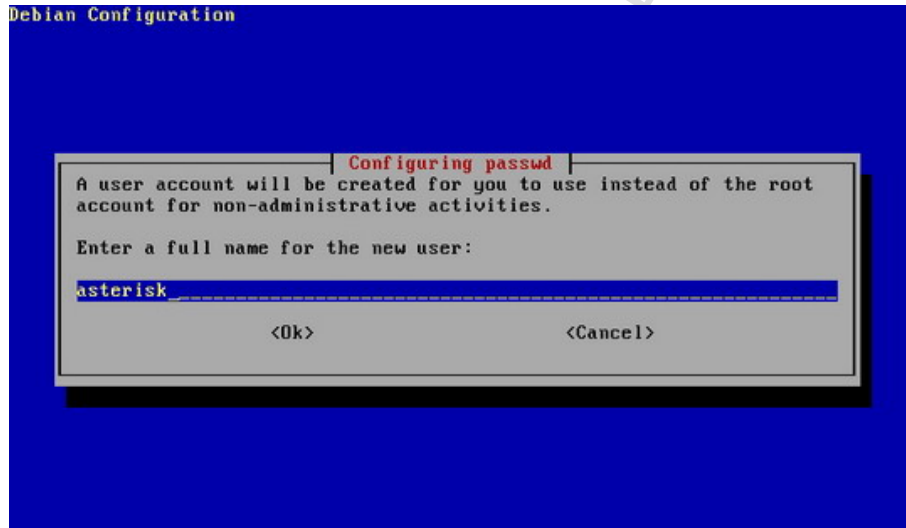
Paso 19: Digite "asterisk" de contraseña para o usuário root.



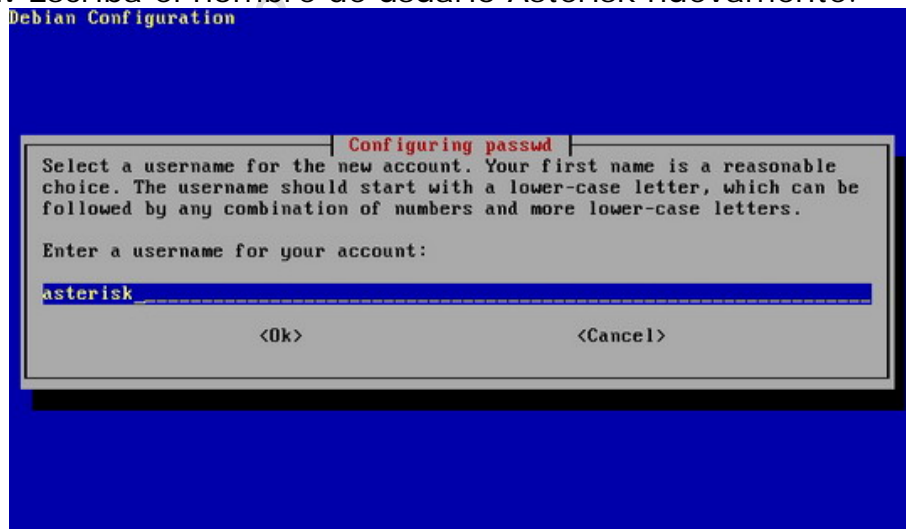
Paso 20: Reescriba la contraseña para confirmar.



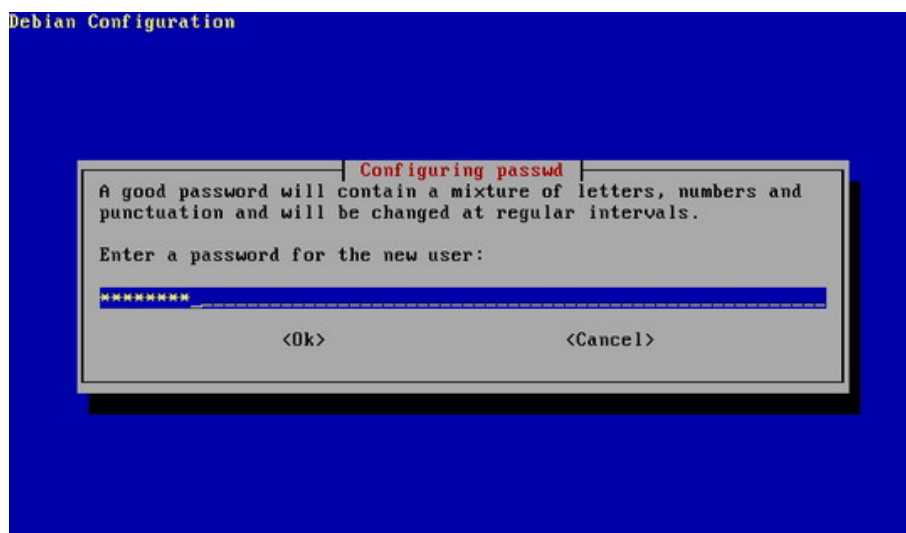
Paso 21: Cree un usuario llamado asterisk.



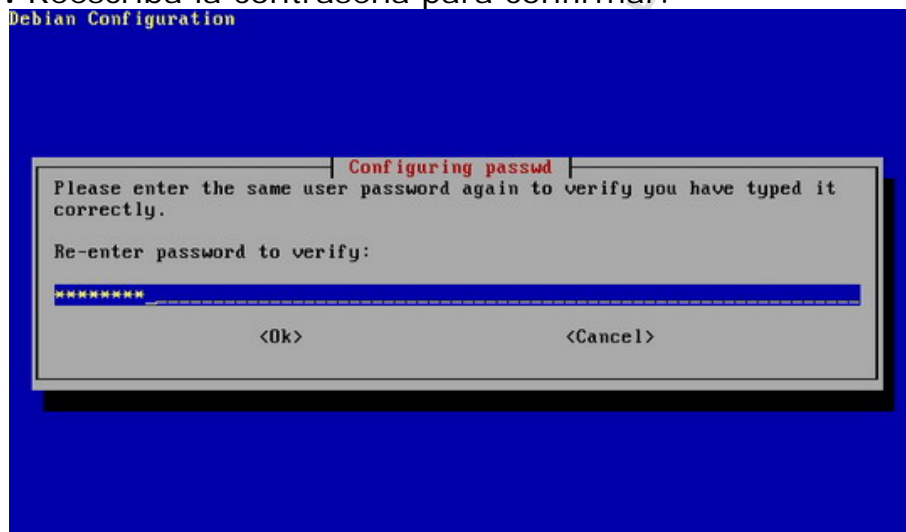
Paso 22: Escriba el nombre de usuario Asterisk nuevamente.



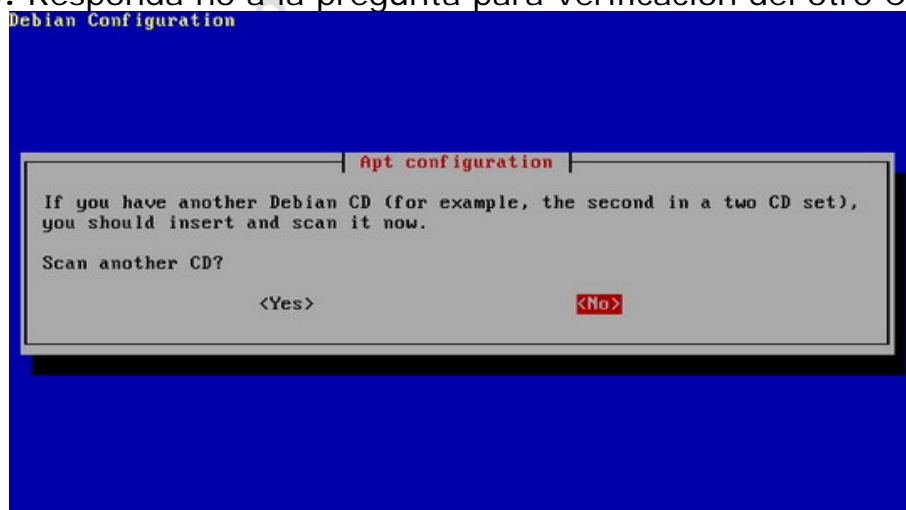
Paso 23: Escriba "asterisk" como contraseña para el usuario asterisk.



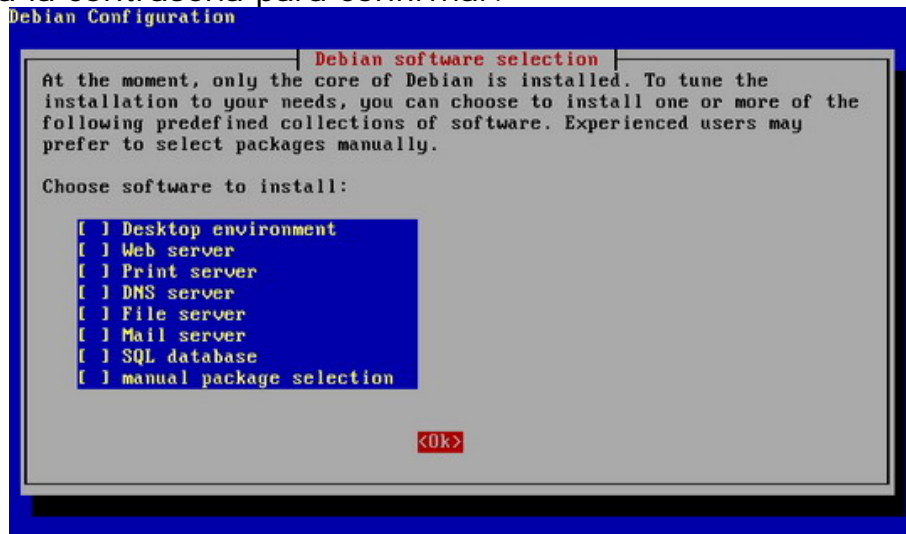
Paso 24: Reescriba la contraseña para confirmar.



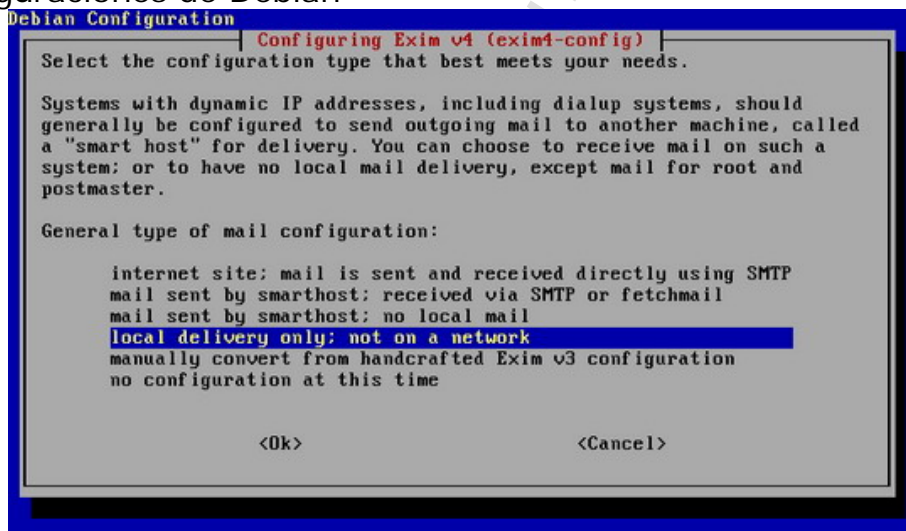
Paso 25: Responda no a la pregunta para verificación del otro CD.



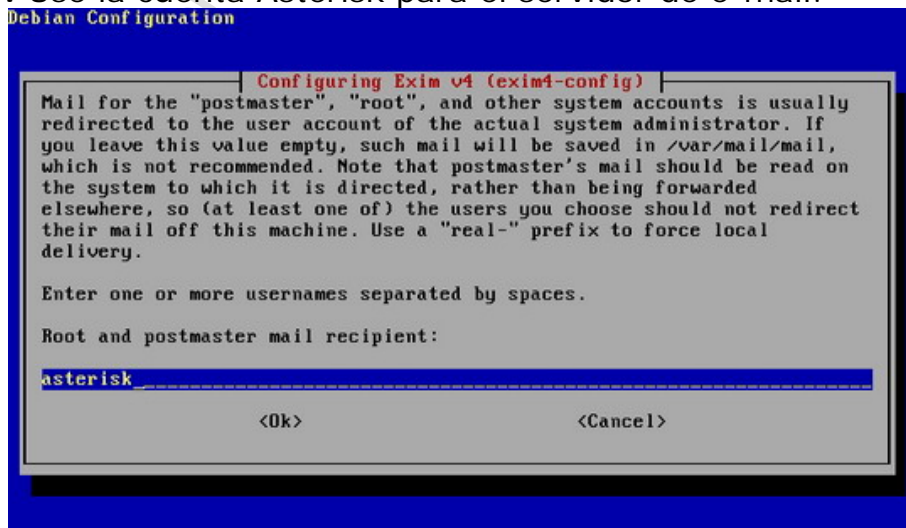
Paso 26: Ya que utilizaremos esta máquina como servidor Asterisk PBX, no es necesario seleccionar ninguna opción de las presentadas aquí. Reescriba la contraseña para confirmar.

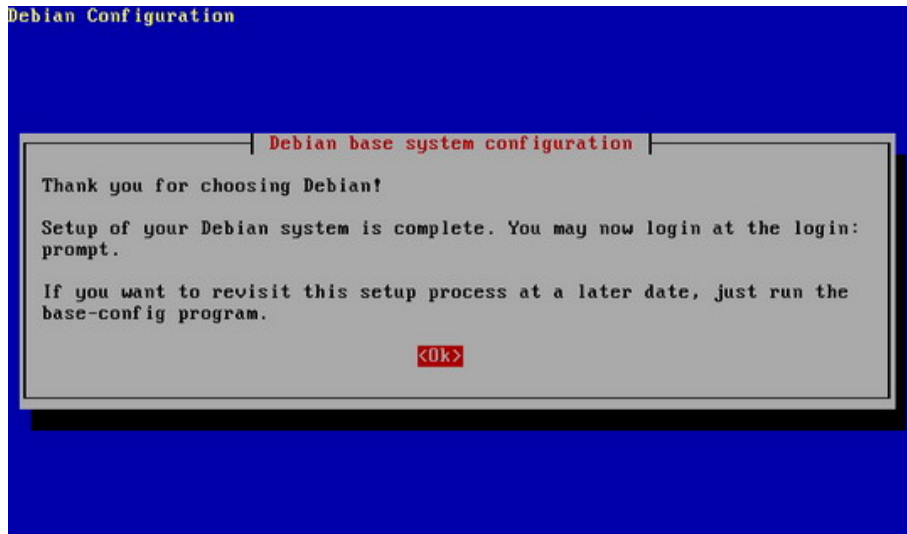


Paso 27: Apenas confirme la opción presentada para la terminación de las configuraciones de Debian



Paso 28: Use la cuenta Asterisk para el servidor de e-mail:





2.5 Preparando Debian para Asterisk.

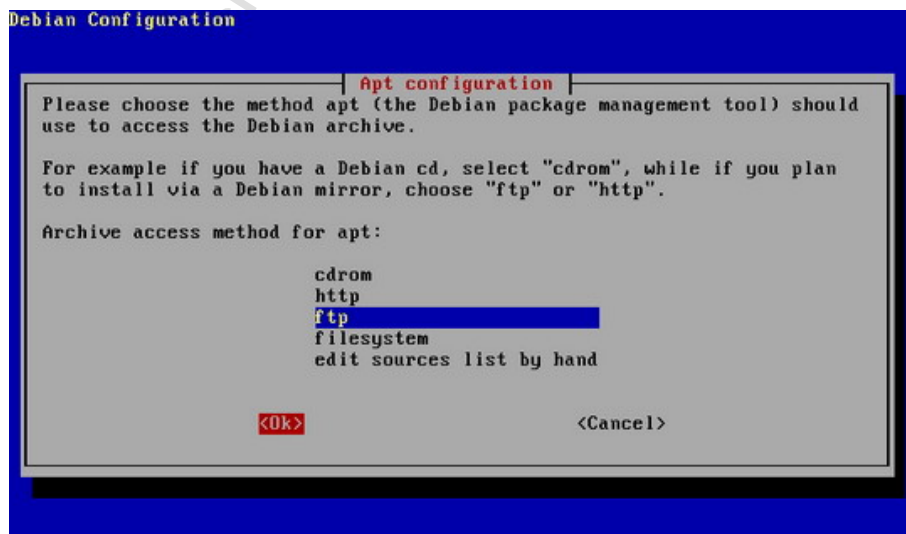
La instalación de Debian está completa, vamos ahora instalar los paquetes necesarios para la instalación/compilación de los drivers de las placas zaptel y del propio Asterisk.

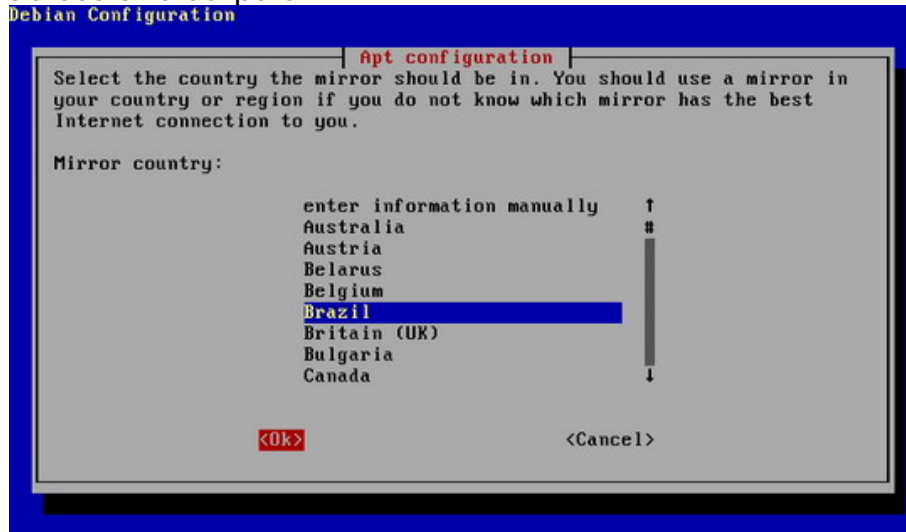
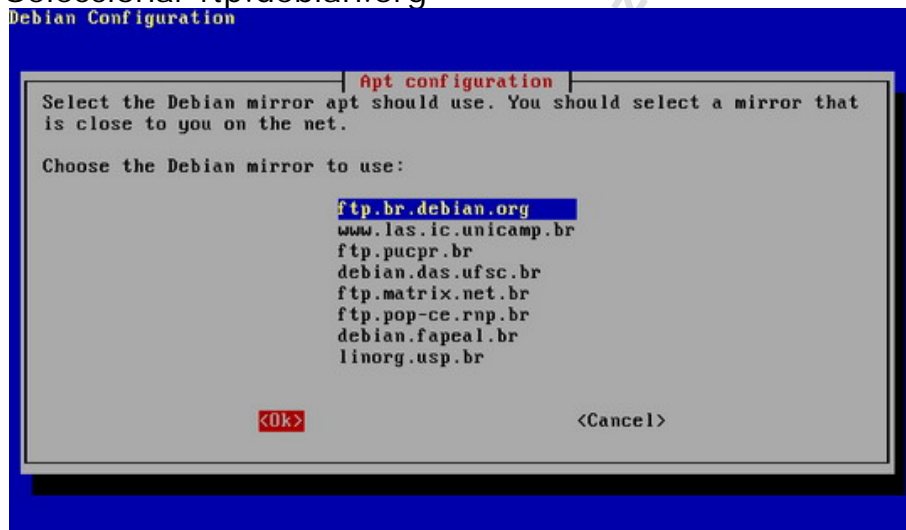
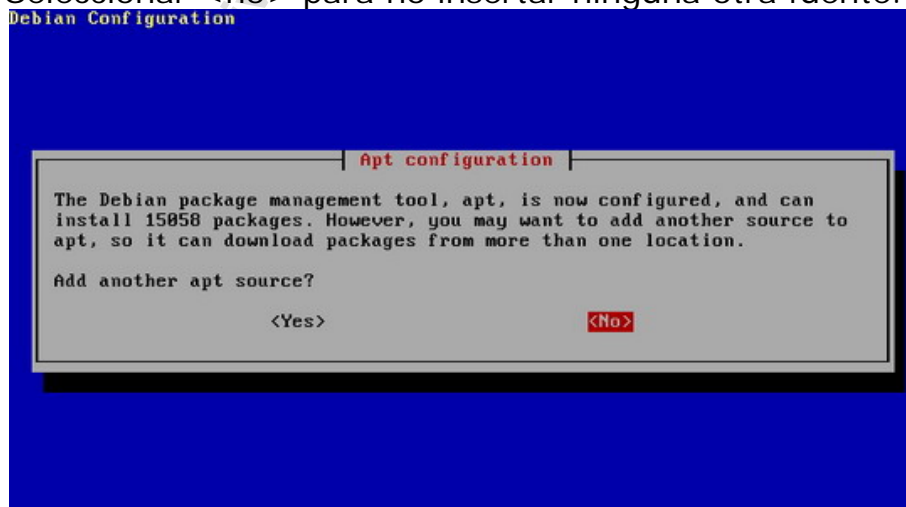
Paso 1: Haga login como root

Paso 2: Agregar fuente para download de paquetes (/etc/apt/sources.list))

```
#apt-setup
```

Paso 3: Seleccionar ftp.



Paso 4: Seleccione su país.**Paso 5:** Seleccionar ftp.debian.org**Paso 6:** Seleccionar <no> para no insertar ninguna otra fuente.**Paso 7:** Instalar los Headers del Kernel en ejecución:

```
#apt-get install kernel-headers-`uname -r`  
#ln -s /usr/src/kernel-headers-`uname -r` /usr/src/linux
```

Paso 8: Instalación de los paquetes necesarios para Asterisk:

```
#apt-get install bison openssl libssl-dev libasound2-dev libc6-dev libnewt-dev  
libncurses5-dev zlib1g-dev gcc g++ make
```

2.6 Obteniendo y compilando Asterisk

Ahora que usted ya instaló Linux y las bibliotecas necesarias, vamos a hacer la instalación de Asterisk.

2.6.1 Obteniendo las fuentes de Asterisk

Para obtener las fuentes de Asterisk y drivers de Zaptel para el uso con hardware de digium, usted debe bajar los paquetes de Digium. Baje los archivos usando el comando `wget`. Cree el directorio `/usr/src` si este no existe. En el momento de finalización de este libro estas eran las versiones corrientes, substituya los comandos abajo con los archivos de las versiones más actuales.

```
# cd /usr/src  
#wget http://ftp.DIGIUM.com/pub/zaptel/zaptel-1.4.x.tar.gz  
#wget http://ftp.DIGIUM.com/pub/libpri/libpri-1.4.x.tar.gz  
#wget http://ftp.DIGIUM.com/pub/ASTERISK/ASTERISK-addons-1.4.x.tar.gz  
#wget http://ftp.DIGIUM.com/pub/ASTERISK/ASTERISK-1.4.x.tar.gz
```

Descomprima los archivos usando:

```
# tar xzvf ASTERISK-1.4.x.tar.gz
# tar xzvf libpri-1.4.x.tar.gz
# tar xzvf ASTERISK-addons-1.4.x.tar.gz
# tar xzvf zaptel-1.4.x.tar.gz
```

2.6.2 Compilando los driver zaptel

Normalmente la compilación de los drivers zaptel es simple. En tanto que pueden ocurrir casos donde usted no posea ninguna placa TDM que use el driver zaptel. Asterisk precisa de una fuente de temporización que normalmente sea provista por una placa con driver zaptel. Si esta placa no existe va ha ser preciso compilar el módulo ztdummy

Compilando los drivers de Zaptel.

```
cd /usr/src/zaptel-1.4.x/
make clean
./configure
make menuselect
make install
make install-udev
make config
update-rc.d zaptel defaults
```

Use el comando `make menuselect` para escoger los módulos necesarios. Screenshot del comando "make menuselect":



2.6.3 Compilando Asterisk

Si usted ya compilo software antes, compilar Asterisk le va a parecer simple. Corra los siguientes comandos para compilar e instalar Asterisk después de que haya bajado este del servidor CVS.


```
cd /usr/src/libpri-1.4.x/
make clean
make
make install
```

```
cd /usr/src/asterisk-1.4.0
make clean
./configure
make menuselect
make
make install
make samples           ; use to create sample configuration files
make config            ; use para start asterisk at boot time
```

2.7 Iniciando y deteniendo Asterisk

Con esta configuración mínima, ya es posible iniciar Asterisk con éxito

```
/usr/sbin/asterisk -vvvc
```

Use el comando **stop now** para parar Asterisk. Vea los comandos disponibles en la interface de línea de comando de Asterisk.

```
CLI > stop now
```

2.7.1 Parametros de línea de comando de Asterisk.

El proceso de ejecutar Asterisk es fácil. Si Asterisk fuese corrido sin argumentos, el es lanzado como un "daemon" (Proceso que espera conexiones en un puerto TCP o UDP).

```
#!/sbin/asterisk
```

Usted puede acceder a la consola de un proceso de Asterisk que ya este en ejecución usando el comando escrito debajo. Más de una consola puede ser conectada al Asterisk simultáneamente.

```
#!/sbin/asterisk -r
```

2.7.2 Los parámetros disponibles

Si puede verificar las opciones de inicialización con:

```
debian:/usr/src/asterisk-1.4.0# asterisk -h
Asterisk 1.4.0, Copyright (C) 1999 - 2006, Digium, Inc. and others.
Usage: asterisk [OPTIONS]
Valid Options:
  -V           Display version number and exit
```

-C <configfile>	Use an alternate configuration file
-G <group>	Run as a group other than the caller
-U <user>	Run as a user other than the caller
-c	Provide console CLI
-d	Enable extra debugging
-f	Do not fork
-F	Always fork
-g	Dump core in case of a crash
-h	This help screen
-i	Initialize crypto keys at startup
-l	Enable internal timing if Zaptel timer is available
-L <load>	Limit the maximum load average before rejecting new calls
-M <value>	Limit the maximum number of calls to the specified value
-m	Mute the console from debugging and verbose output
-n	Disable console colorization
-p	Run as pseudo-realtime thread
-q	Quiet mode (suppress output)
-r	Connect to Asterisk on this machine
-R	Connect to Asterisk, and attempt to reconnect if disconnected
-t	Record soundfiles in /var/tmp and move them where they belong after they are done.
-T	Display the time in [Mmm dd hh:mm:ss] format for each line of output to the CLI.
-v	Increase verbosity (multiple v's = more verbose)
-x <cmd>	Execute command <cmd> (only valid with -r)

2.8 Directorios de instalación

Asterisk es instalado en diversos directorios. Usted puede alterar los directorios editando lo archivo asterisk.conf

asterisk.conf

```
[directories]
astetcdir => /etc/asterisk
astmoddir => /usr/lib/asterisk/modules
astvarlibdir => /var/lib/asterisk
astdatadir => /var/lib/asterisk
astagidir => /var/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk
astrundir => /var/run
astlogdir => /var/log/asterisk

; Changing the following lines may compromise your security.
;[files]
;astctlpermissions = 0660
;astctlowner = root
;astctlgroup = apache
;astctl = asteriskctl
;[options]
;internaltiming = yes
```

2.9 Los archivos de log e la rotación de ellos

Asterisk hace los logs en el directorio /var/log/asterisk. Lo archivo que controla los logs es /etc/asterisk/logger.conf.

```

;
; Logging Configuration
;
; In this file, you configure logging to files or to
; the syslog system.
;
; "logger reload" at the CLI will reload configuration
; of the logging system.

[general]
; Customize the display of debug message time stamps
; this example is the ISO 8601 date format (yyyy-mm-dd HH:MM:SS)
; see strftime(3) Linux manual for format specifiers
; dateformat=%F %T
;
; This appends the hostname to the name of the log files.
; appendhostname = yes
;
; This determines whether or not we log queue events to a file
; (defaults to yes).
; queue_log = no
;
; This determines whether or not we log generic events to a file
; (defaults to yes).
; event_log = no
;
;
; For each file, specify what to log.
;
; For console logging, you set options at start of
; Asterisk with -v for verbose and -d for debug
; See 'asterisk -h' for more information.
;
; Directory for log files is configured in asterisk.conf
; option astlogdir
;

[logfiles]
;
; Format is "filename" and then "levels" of debugging to be included:
;   debug
;   notice
;   warning
;   error
;   verbose
;   dtmf
;
; Special filename "console" represents the system console
;
; We highly recommend that you DO NOT turn on debug mode if you are simply
; running a production system. Debug mode turns on a LOT of extra
messages,
; most of which you are unlikely to understand without an understanding of
; the underlying code. Do NOT report debug messages as code issues, unless
; you have a specific issue that you are attempting to debug. They are
; messages for just that -- debugging -- and do not rise to the level of
; something that merit your attention as an Asterisk administrator. Debug
; messages are also very verbose and can and do fill up logfiles quickly;
; this is another reason not to have debug mode on a production system
unless
; you are in the process of debugging a specific issue.
;
; debug => debug
; console => notice, warning, error

```

```

; console => notice, warning, error, debug
messages => notice, warning, error
; full => notice, warning, error, debug, verbose

; syslog keyword : This special keyword logs to syslog facility
;
;
; syslog.local0 => notice, warning, error
;
;

```

Existen algunas instrucciones relativas a los logs.

Channel	Type	Status	Configuration
/var/log/asterisk/messages	File	Enabled	- Warning Notice
Error	Console	Enabled	- Warning Notice

```

CLI> logger rotate
== Parsing '/etc/asterisk/logger.conf': Found
Asterisk Event Logger restarted
Asterisk Queue Logger restarted

```

Usted puede programar la rotación de los logs usando lo logrotate daemon.

/etc/logrotate.d/asterisk.

```

/var/log/asterisk/messages /var/log/asterisk/*log {
    missingok
    rotate 5
    weekly
    create 0640 asterisk asterisk
    postrotate
        /usr/sbin/asterisk -rx 'logger reload'
    endscrip t
}

```

Más informaciones relativas a logrotate pueden ser obtenidas en:

```
#man logrotate
```

2.10 Iniciando Asterisk usando un usuario diferente de root.

Es más seguro ejecutar Asterisk con un usuario diferente del usuario root. En caso de falla de seguridad o ataque de tipo "buffer overflow" es más seguro que Asterisk haya sido iniciado con un usuario menos privilegiado.

Para alterar el usuario de ejecución de Asterisk:

- 1) Edite el archivo asterisk: vi /etc/init.d/asterisk

2) Retire los comentarios de las siguientes líneas del archivo removiendo el # del inicio de cada línea:

3)

```
AST_USER="asterisk"
AST_GROUP="asterisk"
```

4) Para alterar los derechos de usuario asterisk en las tareas utilizadas para la ejecución de Asterisk, digite:

```
#cd /
#chown --recursive asterisk:asterisk /etc/asterisk
#chmod --recursive u=rwX,g=rX,o= /etc/asterisk
#chown --recursive asterisk:asterisk /var/lib/asterisk
#chown --recursive asterisk:asterisk /var/log/asterisk
#chown --recursive asterisk:asterisk /var/run/asterisk
#chown --recursive asterisk:asterisk /var/spool/asterisk
#chown --recursive asterisk:asterisk /dev/zap
#chmod --recursive u=rwX,g=rX,o= /var/lib/asterisk
#chmod --recursive u=rwX,g=rX,o= /var/log/asterisk
#chmod --recursive u=rwX,g=rX,o= /var/run/asterisk
#chmod --recursive u=rwX,g=rX,o= /var/spool/asterisk
#chmod --recursive u=rwX,g=rX,o= /dev/zap
```

5) Testeé las alteraciones realizadas escribiendo: /etc/init.d/asterisk start

2.11 Consideraciones sobre la instalación de Asterisk

2.11.1 Sistemas en producción

Si Asterisk fue instalado en un ambiente de producción, se debe prestar atención en el proyecto del sistema. El servidor debe ser optimizado de forma que las funciones de telefonía tengan prioridad sobre otros procesos de sistema. En la mayoría de los casos Asterisk no debe correr otros procesos, principalmente si fueran intensivos en CPU. Si fueran necesarios procesos que utilizan mucha CPU como las bases de datos, por ejemplo, estos deben ser instalados eventualmente en un servidor separado.

De una forma general Asterisk es un sistema sensible a variaciones en el performance de la máquina. Esto significa que en un sistema en producción lo ideal es no usar interfaces gráficas como es KDE o GNOME.

2.11.2 Consideraciones sobre la red

Si usted va a usar teléfonos IP, lo que es muy probable es importante que usted preste atención a algunas cuestiones sobre la red. Los protocolos de voz sobre IP son muy buenos y resistentes a pérdidas de paquetes, atrasos y variaciones de atrasos. Entretanto si usted abusa, la calidad de voz no será buena. Solo es posible garantizar la calidad de voz utilizando

QoS extremo-a-extremo, lo que es inviable principalmente en telefonía sobre Internet. De esta forma es que están las siguientes recomendaciones.

Implemente QoS extremo-a-extremo siempre que pueda. Lo mismo en switches de 100Mbps donde es raro tener un congestionamiento, vale la pena, un virus o una condición de red inesperada puede colocar todo a perder.

Sea conservador, use, por ejemplo, una conexión de Internet exclusiva para softfones y teléfonos IP. En la mayoría de las veces los backbones tienen bajo tráfico, pero una conexión de acceso es congestionada por el propio usuario con descargas, navegación, e-mail entre otras cosas.

Evite concentradores de 10 y 100 Mbps, las colisiones en estos equipos, causan variación de atraso (*jitter*). Jitter es uno de los peores enemigos de telefonía IP.

Oriente a los usuarios de telefonía sobre Internet donde no es posible garantizar la calidad. Mantener las expectativas en un nivel realista evita problemas futuros y comentarios como "Si yo hubiese sabido que era así....".

Cuando usa una red IP privada con equipos que soportan QoS extremo-a-extremo, si la calidad de voz estuviese baja, verifique inmediatamente, es probable que exista algún problema en su red. Con QoS bien implementado la calidad de voz es perfecta, "sin disculpas".

2.12 Sumario

En este capítulo usted aprendió que el requisito mínimo de hardware de Asterisk es un Pentium 300 Mhz con 256 MB RAM y poco más de 100 MB de espacio libre en disco para instalar Asterisk. Por fin aprendimos a inicializar Asterisk con un usuario diferente de "root".

2.13 Cuestionario

1. Cual es la configuración mínima para Asterisk.

2. Las placas de telefonía para Asterisk tienen un procesador propio (DSP), no siendo necesario de mucha CPU del servidor.

- Correcto
- Incorrecto

3. Para que la telefonía IP funcione con perfección es necesario que la red posea QoS extremo-a-extremo.

- Correcto
- Incorrecto

4. Es posible obtener una buena calidad de voz en una red que no este congestionada con switches de 100 Mbps.

- Correcto
- Incorrecto

5. Liste abajo las bibliotecas necesarias para compilar Asterisk.

6. Si usted no tiene una placa zaptel, usted precisa una fuente de tiempo. El driver ztdummy desempeña este papel aprovechando una biblioteca USB. Esto es necesario, pues algunas aplicaciones como _____ y _____ precisan de una referencia de tiempo.

7. Cuando usted hace una instalación de Asterisk, lo mejor es no instalar los paquetes gráficos como KDE y GNOME, pues Asterisk es sensible a cuestiones de CPU y las interfaces gráfica roban muchos ciclos de CPU al servidor.

- Correcto
- Incorrecto

8. Los archivos de configuración de Asterisk se encuentran en _____.

9. Para instalar los archivos de configuración de ejemplo usted precisa ejecutar los siguientes comandos.

10. Porque es importante iniciar Asterisk con un usuario diferente de root.

Exclusivo para Vicente Miguel Perez Paredes

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

Primeros pasos en Asterisk

En este capítulo usted va a aprender a configurar Asterisk dentro de una configuración PBX simple. El objetivo aquí es que usted pueda tener una primera experiencia con Asterisk, configurando uno o dos teléfonos IP, discando entre ellos o para una extensión con un mensaje inicial. Mas adelante, mostraremos, con detalles, los canales SIP, IAX y ZAPATA.

3.1 Objetivos del capítulo

Objetivos

Para el final de este capítulo usted estará apto para:

- Entender y editar archivos de configuración
- Instalar softphones basados en SIP
- Configurar un simple dialplan
- Discar entre teléfonos
- Discar para mensajes originados por Asterisk
- Discar sobre Sip o Zap Trunk.

3.2 Gramática de los archivos de configuración

Asterisk es controlado a través de archivos de configuración localizados en el directorio /etc/asterisk. El formato de los archivos de configuración de Asterisk es semejante a la de los archivos (.ini) de Windows. El punto y coma es el caracter para los comentarios. El signo (=) y el signo (=>) pueden ser usados de forma idéntica, las líneas en blanco serán ignoradas. Archivo de ejemplo:

```
;
; La primera línea sin comentario debe ser el título de una sección.
;
;
[sección]
lave = valor ; Designación de variable
[sección2]
```

objeto => valor ; Declaración de objeto

El interprete de Asterisk toma (=) y (=>) de forma idéntica. La sintaxis sirve para tornar el código más legible. Ahora los archivos comparten la misma sintaxis, existen por lo menos tres tipos distintos de gramática.

3.3 Gramática

Gramática	Creación de los objetos:	Conf. File	Ejemplo
Grupo Simples	Todos en la misma línea	extensions.conf	exten =>4000,1,Dial(SIP/4000)
Herencia de Opciones	Las opciones son definidas primero. Los objetos heredan estas opciones	zapata.conf	[channels] context=default signalling=fxs_ks group=1 channel => 1
Entidad Compleja	Cada entidad recibe un contexto	sip.conf, iax.conf	[cisco] type=friend secret=mysecret host=10.1.30.50 context=trusted [xlite] type=friend secret=xlite host=dynamic

3.3.1 Grupo simples

El formato de grupo simple es la más básico y usado por archivos de configuración donde los objetos son declarados con todas las opciones en la misma línea. Los archivos extensions.conf, meetme.conf y voicemail.conf siguen este formato.

```
[sección]
objeto1 => op1, op2, op3
objeto2 => op1b, op2b, op3b
```

En este ejemplo, el objeto1 es creado con opciones op1, op2 y op3 en cuanto el objeto 2 es creado con op1b, op2b y op3b.

3.3.2 Formato de objeto con herencia de opciones

Este formato es usado por el zapata.conf y otras interfaces donde hay muchas opciones. Entretanto, la mayoría de las interfaces y objetos comparten el mismo valor para opciones con otros. En esta clase de archivo de configuración, típicamente existen una o más secciones que

contienen declaraciones de uno o más canales u objetos. Las opciones para el objeto son especificadas encima de la Declaración de objeto y pueden ser cambiadas para la Declaración de otro objeto. Es un concepto difícil de entender, más muy fácil de usar. Considere el ejemplo expresado aquí abajo:

```
[sección]
op1 = bas
op2 = adv
objeto=>1
op1 = int
objeto => 2
```

Las primeras dos configuran el valor de opción op1 y op2 para "bas" y "adv" respectivamente. Cuando el objeto 1 es inicializado, este es creado con su opción 1 siendo "bas" y su opción 2 siendo "adv". Después de declarar el objeto 1, cambiamos el valor de opción 1 en "int". Y entonces creamos el objeto 2, ahora el objeto 2 es creado con su opción 1 siendo "int" y su opción 2 permaneciendo "adv".

3.3.3 Objeto entidad compleja

El formato objeto entidad compleja es usado por el iax.conf , sip.conf y otras interfaces las cuales existen numerosas entidades con muchas opciones y que típicamente no comparten un gran volumen de configuraciones comunes. Cada entidad recibe su propio contexto (A veces existe un contexto reservado tal como [general] para las configuraciones globales). Las opciones entonces son especificadas en la Declaración de contexto. Considere esto:

```
[entidad1]
op1=val or1
op2=val or2
[entidad2]
op1=val or3
op2=val or4
```

La entidad [entidad1] tiene valores valor1 y valor2 para opciones op1 y op2 respectivamente. La entidad [entidad2] tiene valores valor3 y valor4 para las opciones op1 y op2.

3.4 Configurando una interface con la red pública o un PBX.

Para intercomunicarse con la red pública de telefonía es necesaria una interface de tipo FXO (Foreign Exchange Office) y una línea telefónica común. Un ramal de una central telefónica analógica existente puede ser usado también. Usted puede obtener una placa FXO comprando una placa Digium TDM400P.

En términos generales, una placa FXO es usada para unir la red pública a un PBX, esta placa recibe así tono. Una placa FXS en cambio puede ser usada para unir un aparato telefónico común, una línea FXS dará tono.

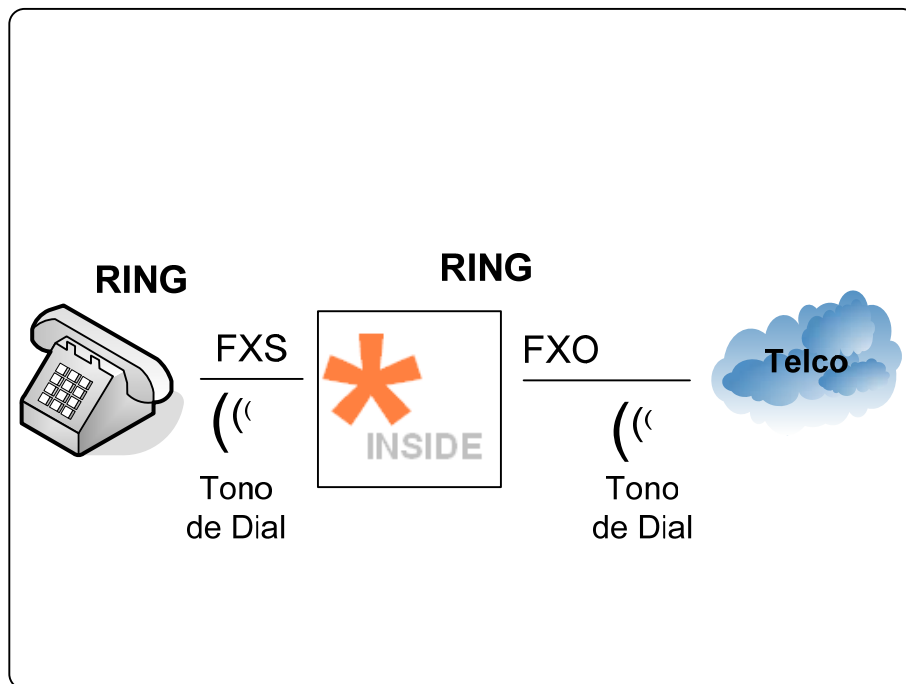


Figura 3.1 Diferencia entre interface FXS y FXO

Una pista: Otra forma es conseguir una placa FXO clon de una X100P. Algunas placas de modem con chipset Motorola y Intel pueden ser usadas. Ellas son difíciles de ser encontradas, pues ya no son fabricadas. Los chipsets que funcionan son, Motorola 68202-51, Intel 537PU, Intel 537 PG, Intel Ambient MD3200

No hay ninguna garantía de que estas placas funcionen con su PC y que así mismo funcionando no presenten problemas como ECO debido a diferencias de impedancia. Use las por su propia cuenta y riesgo, si usted no quisiera correr riesgos las placas de Digium son una excelente opción.

3.4.1 Instalando la placa X100P clon

Antes de instalar una placa X100P en su computadora, es recomendable que usted deshabilite todas las interrupciones y hardware que no está siendo usadas en este momento, como puertos seriales, paralelos, interrupción para vídeo etc...

Para instalar su placa X100P usted debe encajarla en un slot PCI y configurar dos archivos de configuración:

- **"zaptel.conf "** en el directorio /etc que es el archivo de configuración de la placa zaptel.

- **“zapata.conf”** en el directorio /etc/asterisk que es el archivo de configuración de los canales zapata de Asterisk.

No se preocupe en este momento en entender todos los detalles de los archivos de configuración, tenemos un capítulo entero sobre canales zapata. En este momento queremos apenas integrar Asterisk a la red pública usando una placa FXO. Vamos a mostrar abajo el archivo completo y las instrucciones modificadas se presentaran en forma sombreada. A pesar de ocupar varias páginas es la manera más simple de entender la edición del archivo.

zaptel.conf

```
fxsks=1
loadzone=br
defaultzone=br
channels=1
```

zapata.conf

```
[channel s]
context=default
signaling=fxs_ks
group=1
channel => 1
```

Luego de configurar los archivos cargue los drivers zaptel.

```
modprobe zaptel
ztcfg -vvvvv
modprobe wcfxo
asterisk -vvvvvgc
```

3.5 Configuración de los teléfonos IP SIP

En este capítulo vamos a dar una visión general de como configurar los teléfonos SIP. El objetivo en este momento es que usted pueda configurar un PBX simple. Más adelante usted va a tener una sección entera dedicada a SIP y podremos ver en detalles la configuración.

SIP es configurado desde el archivo /etc/asterisk/sip.conf y contiene parámetros relacionados con la configuración de los teléfonos y operadoras SIP. Los clientes SIP deben estar configurados antes que puedan hacer y recibir llamadas.

3.5.1 Sección general [general]

El archivo SIP es leído de encima para abajo. La primera sección contiene las opciones globales [general]. Estas opciones son:

- **allow**: Permite que un determinado codec sea usado.
- **bindaddr**: Dirección IP donde Asterisk irá a esperar por las conexiones SIP. El comportamiento general es esperar en todas las interfaces y direcciones secundarias.
- **context**: Configura el contexto general donde todos los clientes serán colocados, a menos que sea sobrescrito en la definición de entidad.
- **disallow**: Prohíbe un determinado codec.
- **bindport**: Puerto que Asterisk debe esperar por conexiones de entrada SIP. El más general o usado es el puerto 5060.
- **maxexpiry**: Tiempo máximo para registro en segundos.
- **defaultexpiry**: Tiempo por defecto para registrarse en segundos.
- **register**: Registra Asterisk con otro host. El formato es una dirección SIP opcionalmente seguido por una barra normal (/) y una extensión.

Ejemplo:

```
[general ]
bindport = 5060
bindaddr = 10. 1. 30. 45
context = default
disallow = all
allow = ulaw
maxexpiry = 120
defaultexpiry = 80
```

3.5.2 Opciones para cada teléfono

Después de la sección general, siguen las definiciones de las entidades generales SIP. Es bueno recordar que en esta sección vamos apenas a dar una introducción a el archivo sip.conf. Tenemos una sección específica para detallar los otros parámetros. Las entradas son divididas en tres categorías:

- **[name]**: Es la parte "username" de SIP URI.
- **peer**: Entidad para la cual Asterisk envía llamadas.
- **user**: Entidad que hace llamadas a través de Asterisk.
- **friend**: las dos interiores al mismo tiempo.
- **type**: Configura la clase de conexión, las opciones son peer, user y friend.
- **host**: Configura la dirección IP o el nombre de host. Se puede usar también la opción 'dynamic' donde se espera que el teléfono se registre, es la opción más común.
- **secret**: Un secreto compartido usado para autenticar los peers y users haciendo una llamada.

Ejemplo:

```
[ci sco]
type=fri end
secret=mysecret
host=10. 1. 30. 50
context=trusted
```

```
[xl i te]
type=fri end
secret=xl i te
host=dynami c
```

3.6 Introducción al plan de discado

El plan de discado es el corazón de Asterisk, en la medida, que este define como Asterisk irá a gestionando las llamadas. Este consiste de una lista de instrucciones o pasos que Asterisk debería seguir. Esas instrucciones son disparadas a partir dos dígitos recibidos de un canal o aplicación. Es fundamental para configurar Asterisk, que se entienda el plan de discado.

La mayor parte del plan de discado está contenida en el archivo `extensions.conf` en el directorio `/etc/asterisk`. El archivo puede ser separado en cuatro partes:

- Aplicaciones
- Contextos
- Extensiones
- Prioridades

En este capítulo vamos a ver como crear un plan de discado básico, pero que atiende a nuestras necesidades actuales. Más adelante usted tendrá la oportunidad de conocer el plan de discado más a fondo.

Si usted instaló los archivos de ejemplo, ya existe el `extensions.conf`. En nuestro caso será más interesante comenzar de cero. Esto va a ayudar al aprendizaje y podremos abordar paso a paso cada parte del archivo de plan de discado.

3.6.1 Extensiones

Dentro de cada contexto serán definidas diversas extensiones. En Asterisk, una extensión es un string que dispara un evento. Una extensión puede ser una de estos tres tipos: literal, estándar, o especial.

Formato de las extensiones

8000	Numérica
Alexander	Alfanumérica
4321/1234	Numérica con callerID
_4XXX	Patrones
s	Nombres predefinidos

Patrones

_ (underscore)	Inicio
. (dot)	Cualquier
[13-9]	13456789
X	0-9
Z	1-9
N	2-9

Vea este ejemplo:

```
exten=>8580, 1, Di al (SIP/8580, 20)
exten=>8580, 2, voi cemaí l (u8580)
exten=>8580, 101, voi cemaí l (b8580)
```

A instrucción "exten=>" describe cual es el próximo paso para la llamada. El "8580" es un conjunto de dígitos que fue recibido (número discado). El "1", "2" y "101" son las prioridades que determinan la orden de ejecución de los comandos. En este ejemplo, discando "8580" sonará el teléfono IP registrado como "8580", el cual de no ser atendido en 20 segundos será desviado para la prioridad 2 en la casilla de buzón de voz con un mensaje "no atendida". Si estuviese ocupado es desviado para la prioridad 101 y va para el buzón de voz con el mensaje, "ocupado".

Las extensiones determinan el flujo de las llamadas. Ahora las extensiones podrán ser usadas para especificar las extensiones, ellas pueden ser usadas para más que esto en Asterisk. Una extensión puede ser creada con la sintaxis definida abajo:

```
exten=> número (nombre), pri oridad, apl i cación
```

El comando "exten=>" es seguido por un número de extensión, una coma, la prioridad, otra coma y finalmente la aplicación.

3.6.2 Prioridades

Prioridades son pasos numerados de ejecución de cada extensión. Cada prioridad llama a una aplicación específica. Normalmente estos números de prioridad comienzan con 1 y aumentan de uno en uno en cada

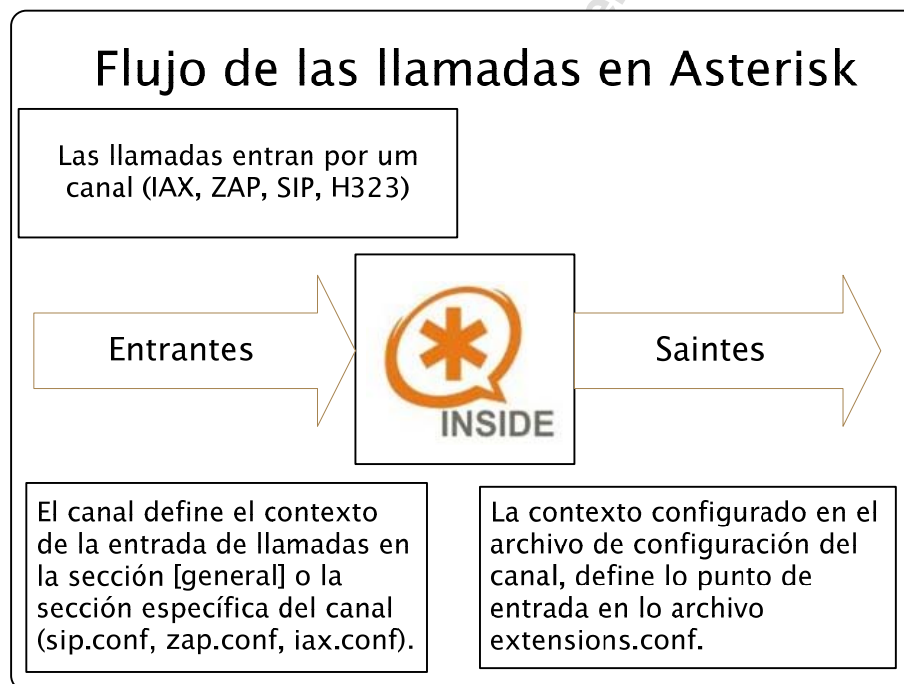
extensión. Los números de prioridad como usted vio arriba no siempre son consecutivos. Las prioridades son corridas en orden numérica.

3.6.3 Aplicaciones

Las aplicaciones son parte fundamental de Asterisk, ellas tratan al canal de voz, tocando sonidos, aceptando dígitos o cortando una llamada. Las aplicaciones son llamadas con opciones que afectan a su forma de funcionamiento. Usted puede usar **core show applications** en la interface de línea de comando de Asterisk. En la medida en que usted construye su primer plan de discado usted va a aprender a usar las aplicaciones de forma apropiada. Usted puede ver las opciones de aplicación instaladas en Asterisk usando el comando:

```
CLI>core show applications
```

3.6.4 Contextos



Los contextos tienen un papel importante en Asterisk en lo que respecta a organización y seguridad del plan de discado. Los contextos también definen el objetivo y permiten separar diferentes partes del plan de discado. Un punto que llama la atención es que los contextos están ligados directamente a los canales. Cada canal existe dentro de un contexto. Cuando una llamada entra al Asterisk por un canal ella es procesada dentro de un contexto.

Ejemplificando, vamos a suponer que usted tenga dos clases de ramales, aquellos que pueden hacer llamadas de larga distancia y aquellos que no. Usted puede crear dos contextos, [gerente] y [funcionario]. Dentro del

contexto [gerentes] cuando el dígito "0" es discado, se oye el tono de discado de la red pública. Dentro del contexto [funcionario] cuando el dígito "0" es discado es recibida, por ejemplo, por una grabación "llamada no autorizada".

Por otro lado, una llamada es recibida dentro del contexto del canal. Con esto diferentes canales pueden ser recibidos en diferentes teléfonos dependiendo del contexto seleccionado. Esto puede ser útil para tener una recepción diferente para cada compañía compartiendo un mismo servidor Asterisk.

Contextos también son usados para crear menús de voz que dan al usuario una lista de extensiones para escoger presionando las teclas de un teléfono multi frecuencial. Esta funcionalidad es normalmente conocida como auto-atención de llamadas. Auto-atención de llamadas será visto en capítulos posteriores.

Los contextos reciben su nombre dentro de llaves ([]s). Por ejemplo, si nosotros hiciéramos la creación de un contexto para la entrada de llamadas, podríamos definir esta como sigue:

[entrada]

Todas las instrucciones colocadas después de esta definición son partes del contexto. Para iniciar un nuevo contexto, simplemente escriba el nuevo contexto [nuevocontexto]. En el inicio del archivo extensions.conf existe un contexto llamado [globals]. El contexto globals es donde las variables son definidas y pueden ser usadas por todo el plan de discado.

3.6.5 Creando un ambiente de teste

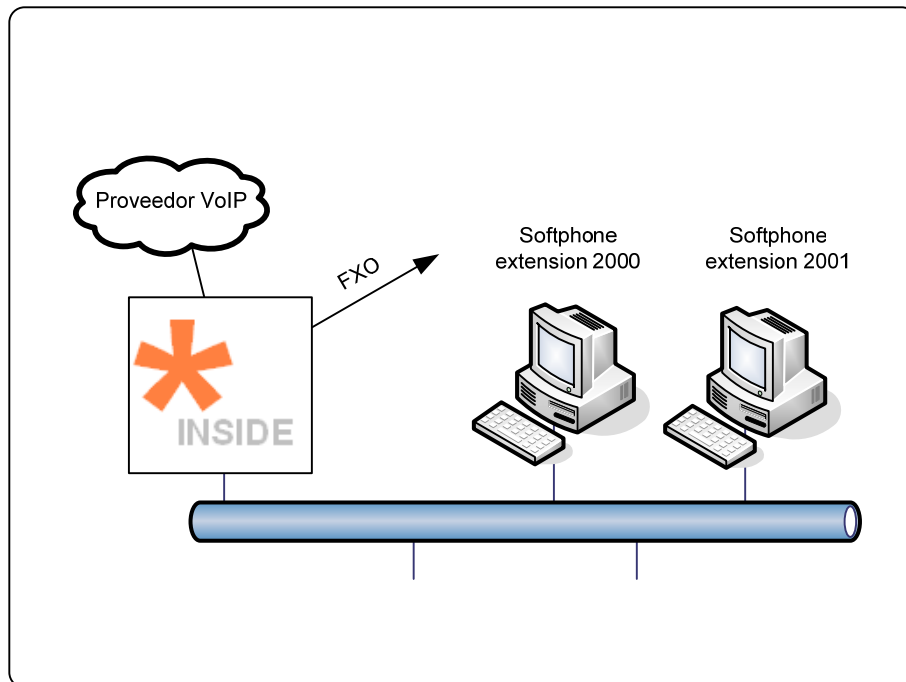


Figura 3.2 Laboratorio de voz sobre IP

La manera más económica de testear Asterisk es conectarse a un proveedor (en www.voipcharges.com usted puede encontrar una lista de proveedores que actúan en cualquier país del mundo), vamos a ver como se conecta a un proveedor en el capítulo relativo a los canales SIP. Otra forma es usar una placa FXO. Para hacer el laboratorio descrito usted va a precisar de dos PCs, Pentium 300 Mhz o mayores con 256 MB RAM por lo menos. Si usted tuviera un único PC puede usar el vmware (no es freeware) y usted precisara por lo menos 512 MB RAM. Vamos a usar el softphone gratuito de counterpath el xlite. Usted puede bajarlo de www.counterpath.com. Existen otros softphones gratuitos como el Sjphone de SJLabs www.sjlabs.com. Ciertamente usted encontrará otros buscando en Internet. Puede ser también que los dos citados arriba no existan más cuando usted estuviese leyendo este material. Usted puede también crear este laboratorio usando un equipamiento para Asterisk y un adaptador de telefonía analógico de dos canales.

Paso 1: Edite el archivo sip.conf y agregue la configuración de un ramal con número 2000.

```
[general]
bindport=5060
bindaddr=10.1.30.45 ; Coloque aquí la dirección IP de su servidor
context=default
allow=all
[2000]
type=friend
secret=2000
```

```
host=dynami c
canrei nvi te=no
```

Paso 2: Repita el paso 1 ahora para el ramal 2001

Paso 3: Configure los softphone xlite de ramal 2000 para Asterisk.

- a) Ejecute el programa de instalación
- b) Presione "next" en la primera pantalla
- c) Acepte el contrato de licencia
- d) Acepte las próximas ventanas en general hasta terminar la instalación. En otras palabras NEXT->NEXT->FINISH.
- e) Entre en el menu de X-LITE presionando el icono mostrado abajo:



- f) En la siguiente pantalla escoja "system settings".
- g) En la próxima escoja SIP PROXY.
- h) Escoja Default
- i) Complete los siguientes campos:

```
Di spl ay Name: 2000
Username: 2000
Authorizati on User: 2000
Password: 2000
Domai n/Real m: Di recci ón IP del servi dor
SIP Proxy: Di recci ón IP del servi dor
```

- j) Cierre el XLITE y ábralo de nuevo.
- k) Confirme que el teléfono fue registrado por Asterisk con el comando **sip show peers**.

Paso 4: Repita la configuración de XLITE para el softphone 2001.

3.6.6 Creando un plan de discado simple

Ahora, estamos listos para nuestro primer ejemplo de plan de discado. Por favor, presten atención a la manera en que cada prioridad llama una

aplicación. Note que en este ejemplo tenemos apenas una extensión creada con SIP. En estos ejemplos se asume que usted tiene una placa FXO. Vamos a mostrar los ejemplos y explicar paso a paso.

Ahora estamos listos para crear el `extensions.conf` de la forma mas simple posible. En este ejemplo todo lo que Asterisk irá a hacer es responder a una llamada, tocar un sonido que dice "adiós" y colgar.

La extensión especial "s"

Vamos a comenzar viendo la extensión especial 's' que significa inicio (start). De una forma general la llamada inicia un contexto al cual el canal pertenece. La extensión 's' es inmediatamente ejecutada al entrar en un canal. En nuestro ejemplo vamos a crear un plan de discado con la extensión 's'

Las aplicaciones `Answer()`, `Hangup()` y `Playback()`

Si vamos a responder a una llamada, es mejor que conozcamos las aplicaciones que van a hacer esto. La aplicación `Answer()` es usada para responder a un canal que está sonando. Esta hace la configuración inicial de llamada y puede hacer otras funciones. Pocas aplicaciones no requieren que necesariamente se responda (`Answer()`) al canal antes de hacer cualquier otra cosa

La aplicación `Playback()` es usada para hacer sonar un archivo de sonido previamente grabado sobre un canal. Cuando la aplicación `Playback()` está siendo ejecutada, cualquier dígito presionado por el usuario es simplemente ignorado. El formato es `Playback(nombredearchivo)`, este hace sonar el archivo con la extensión `.gsm` en el directorio de sonido por defecto.

La aplicación `Hangup()` hace exactamente lo que su nombre dice. Ella cuelga un canal activo. Usted debería usarla en el fin de contexto una vez que usted quiera colgar a quien no precise estar conectado al sistema.

Ahora, estamos listos para nuestro primer ejemplo de plan de discado. Por favor, presten atención a la manera que cada prioridad llama una aplicación. Note que en este ejemplo tenemos apenas una extensión creada con SIP. En estos ejemplos asumimos que usted tiene una placa FXO. Vamos a mostrar los ejemplos y explicar paso a paso.

```
[entrada]
exten=>s, 1, answer()
exten=>s, 2, pl ayback(goodbye)
exten=>s, 3, hangup()
```

Una llamada que entre por la FXO es enviada dentro del contexto [entrada] (Este contexto [entrada] debe estar configurado en el archivo zapata.conf para el canal FXO) y es enviada para la extensión 's'. Tenemos tres prioridades en el contexto, 1, 2 y 3. Cada prioridad llama un aplicativo. Vamos a observar de cerca las prioridades:

La prioridad 1 llama a la aplicación Answer(), Asterisk toma cuenta de la línea y configura la llamada. Después de responder la línea, Asterisk va para la próxima prioridad.

Prioridad 2, Asterisk llama a la aplicación Playback() para tocar el archivo de sonido goodbye.gsm (por tanto vamos usar los mensajes en ingles), este mensaje dará un saludo de adiós al usuario.

Por fin la prioridad 3 irá a desconectar al usuario.

Un ejemplo más útil

Ahora que ya comenzamos con algo simple, vamos a incrementar aprendiendo un poco de los aplicativos background() y goto(). Estas dos aplicaciones van a permitirnos crear planes de discado, con mucho más funcionalidad.

La llave para sistemas interactivos basados en Asterisk está en la aplicación Background(). Ella permite que usted grabe un archivo de sonido, pero cuando la origen presiona una tecla eso interrumpe la grabación y manda para la extensión correspondiente a los dígitos discados.

Otra aplicación muy útil es el goto(). Como su nombre lo explica, este salta de un contexto actual, extensión y prioridad para un contexto específico y prioridad específica. La aplicación goto() torna fácil el movimiento entre diferentes partes del plan de discado. El formato del comando goto() precisa un contexto de destino y prioridad como argumentos.

```
exten=>extensi on, pri ori dad, Goto(contexto, extensi on, pri ori dad)
```

Formatos válidos del comando Goto() son :

```
Goto(contexto, extensi on, pri ori dad)
Goto(extensi on, pri ori dad)
Goto(pri ori dad)
```

En este ejemplo, vamos asumir que somos una empresa de soporte, entrenamiento y ventas. Vamos a crear un sistema interactivo que permita al usuario seleccionar para que área de la empresa esta quiera ser redireccionado. En primer lugar vamos a usar el comando

Background() para tocar un mensaje “llame 1 para soporte, 2 para entrenamiento y 3 para ventas”, en este momento todavía no vamos a explorar el tratamiento de dígitos inválidos. En cada contexto vamos a tocar una grabación como “usted fue redireccionado para soporte (entrenamiento o ventas)”.

```
[entrada]
exten=>s, 1, Answer()
exten=>s, 2, Background(saludos)
exten=>s, 3, hangup()
exten=>1, 1, pl ayback(soporte)
exten=>1, 2, goto(soporte, s, 1)
exten=>2, 1, pl ayback(entrenamiento)
exten=>2, 2, goto(entrenamiento, s, 1)
exten=>3, 1, pl ayback(ventas, )
exten=>3, 2, goto(ventas, s, 1)
```

Vamos a seguir este ejemplo paso a paso. Cuando alguien se comunica en la interface FXO (configurada para el contexto [entrada]), la comunicación es pasada para la extensión ‘s’ dentro del contexto [entrada]. La extensión ‘s’ atiende la comunicación y usando el comando Background() hace sonar un saludo y aguarda por el discado de un dígito. Después de discar el dígito ‘1’ por ejemplo, el sistema va para la extensión ‘1’ en la prioridad ‘1’ y toca un mensaje del archivo soporte.gsm (Algo como, “usted se comunico con el área de soporte técnico” lo atendemos en breve). Enseguida tenemos el comando goto() que manda la comunicación para el contexto (soporte) donde ella será tratada.

Intercomunicando canales con la aplicación Dial()

Nosotros vamos a adicionar a nuestro ejemplo la aplicación Dial(). En vez de mandar la comunicación para otro contexto, vamos atenderla en un ramal específico.

```
[entrada]
exten=>s, 1, Answer()
exten=>s, 2, Background(saludo)
exten=>s, 3, hangup()
exten=>1, 1, pl ayback(soporte)
exten=>1, 2, Dial(SIP/8000)
exten=>2, 1, pl ayback(entrenamiento)
exten=>2, 2, Dial(ZAP/1)
exten=>3, 1, pl ayback(ventas)
exten=>3, 2, Dial(IAX/8002)
```

Al comparar con el ejemplo anterior, apenas creamos un atajo. En vez de enviar para otro contexto para el tratamiento de la llamada, enviamos directamente a un canal SIP del ramal 8000 o a un canal Zaptel (FXS - Analógico) o también a un canal IAX del ramal 8002.

En este punto usted ya debe estar entendiendo el uso de varias aplicaciones como el Answer(), Background(), Goto(), Hangup() y

Playback() y lo básico del comando Dial(). Esto es fundamental para el aprendizaje de aquí en adelante. Si algo no está claro todavía vuelva y lea de nuevo, es fundamental que haya entendido el proceso antes de seguir adelante.

Con un entendimiento básico de las extensiones, prioridades y aplicaciones es simple crear un plan de numeración básico. En los próximos capítulos vamos a hacer un plan de discado más poderoso.

3.7 Laboratorio

En este laboratorio vamos a crear una pequeña central telefónica capaz de discar entre ramales, para la red pública por la placa PSTN o por un proveedor VoIP y recibiendo llamadas usando atención automática de llamadas. Para crear este PBX vamos a hacer algunas consideraciones:

- Los ramales van desde 2000 hasta 2100
- Para salir para la red pública se disca el número 0 (cero)
- Para salir para la red pública a través de un proveedor se disca 1 (uno)
- 8000 será el ramal para grabación de menú de atención automática de llamadas.

3.7.1 Comunicando se entre teléfonos.

En el archivo extensions.conf en la sección default entre con los siguientes comandos:

```
[default]
exten=>2000,1,Dial(SIP/2000)
exten=>2001,1,Dial(SIP/2001)
```

3.7.2 Comunicando se para la red pública usando la placa zaptel

Coloque las siguientes líneas en el archivo extensions.conf. Para testear marque 0 (cero) seguido de un número externo.

```
[default]
exten=>0,1,Dial(ZAP/1,20,r)
```

3.7.3 Recibiendo comunicaciones usando atención automática de llamadas.

Escriba los comandos que están abajo, estos van a crear una interface para que usted pueda grabar los mensajes del menú de atención automática de llamadas. El mensaje puede ser algo como "Usted se comunico con xyz, llame el extensión deseado o aguarde y será atendido". Cuando termine de hablar el mensaje tecleé "#" para que este

sea grabado. Usted va a oírlo enseguida cuando el comando "Playback" fuese ejecutado.

```
[default]
exten=>8000, 1, Wait(2)
exten=>8000, 2, Record(menu:gsm)
exten=>8000, 3, Wait(2)
exten=>8000, 4, Playback(menu)
exten=>8000, 5, Hangup()
```

Abajo el comando Background va a tocar un mensaje que usted grabó y esperará que usted digite algo. Digite 2001 y usted va a ver que la comunicación será desviada para el ramal 2001. Si nada fuera discado la comunicación será automáticamente transferida para el ramal 2000.

```
[default]
exten=>s, 1, Background(menu)
exten=>s, 2, Dial(SIP/2000)
exten=>2000, 1, Dial(SIP/2000)
exten=>2001, 1, Dial(SIP/2001)
```

Exclusivo para Vicente Miguel Pe...

3.8 Sumario

En este capítulo usted aprendió que los archivos de configuración están en `/etc/asterisk`. Para usar Asterisk es preciso en primer lugar configurar los canales (Ej. `sip.conf` y `zapata.conf`). Existen básicamente tres formatos, el grupo simple, herencia de opciones, entidad compleja.

El plan de discado es creado en el archivo `extensions.conf`, en el son creados contextos, aplicaciones, extensiones y prioridades. Las aplicaciones que usamos han sido `Playback()`, `Background()`, `Dial()`, `Goto()`, `Hangup()` e `Answer()`.

`Background()` es un comando importante en la creación de una URA. `Dial()` es el principal comando del plan de discado

3.9 Cuestionario

1. Son ejemplos de archivos de configuración de canales Asterisk.

- `zaptel.conf`
- `zapata.conf`
- `sip.conf`
- `iax.conf`

2. Es importante definir el contexto de los archivos de canales, pues cuando nos llega una comunicación de este canal (`sip`, `iax`, `zap`) al Asterisk este será tratado en el archivo `extensions.conf` en este contexto.

- Correcto
- Incorrecto

3. El parámetro `switchtype` del archivo `zapata.conf` define El tipo de PBX al cual Asterisk está ligado. Esto es válido para conexiones no Estándar E1 con señalización ISDN PRI. Normalmente en Brasil y en Europa este Estándar debe ser definido como **National**.

- Correcto
- Incorrecto

4. Todavía sea considerado una línea digital, el E1 puede ser configurado con señalización asociada al canal (CAS) en este caso cada timeslot puede comportarse como un canal analógico FXS o FXO, por ejemplo. Esto es útil para la conexión a channel-banks (bancos de canales).

- Correcto
- Incorrecto

5. SIP Session Initiated Protocol es el protocolo de la ITU usado para conexiones de voz sobre IP. Este es bastante antiguo y viene siendo substituido recientemente por el H.323.

- Correcto
 Incorrecto

6. Dado la configuración de abajo del archivo sip.conf, en la sección [general] está definido la dirección IP 10.1.30.45, donde SIP estará esperando por conexiones. Si fuera necesario que todas las placas de red de la máquina esperasen por una conexión SIP, bindaddr debería estar configurado para: _____.____.____.____

```
[general]
bindport = 5060
bindaddr = 10.1.30.45
context = default
disallow = speex
disallow = ilbc
allow = ulaw
maxexpiry = 120
defaultexpiry = 80
```

7. En el archivo de abajo, los teléfonos 8000 y 8001 fueron definidos con la opción *canreinvite=no*. Con esto, cuando una comunicación es hecha de un teléfono para el otro, el audio va directamente de un teléfono para otro sin pasar por Asterisk.

```
[8000]
type=friend
secret=8000
host=dynamic
canreinvite=no

[8001]
type=friend
secret=8000
host=dynamic
canreinvite=no
```

- Correcto
 Incorrecto

8. La principal diferencia entre el comando Playback() y el comando Background() es que el Playback() simplemente toca un mensaje y pasa al comando siguiente, en cambio el Background aguarda que usted digite algo y desvía para algún lugar en el plan de discado basándose en los dígitos discados.

- Correcto
 Incorrecto

9. Cuando una comunicación entra en Asterisk por una interface de telefonía (FXO) sin identificación de llamada, esta comunicación es desviada para la extensión especial:

- '0'
- '9'
- 's'
- 'i'

10. Los formatos válidos para el comando Goto() son:

- Goto(contexto,extension,prioridad)
- Goto(prioridad, contexto, extension)
- Goto(extension,prioridad)
- Goto(prioridad)

Exclusivo para Vicente Miguel Perez Paredes

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

Canales Analógicos y Digitales

En esta sección vamos a mostrar como definir los canales de integración de comunicación las redes de telefonía POTS³. También haremos una revisión sobre conceptos de telefonía y aprenderemos la configurar el archivo zapata.conf para diversos escenarios.

4.1 Objetivos

Objetivos

Para el final de este capítulo usted estará apto para:

- Reconocer los principales terminos de telefonía and acronismos.
- Entender cuando usar circuitos digitales o analogicos.
- Recognize
- Reconocer la diferencia entre FXS y FXO
- Configurar Asterisk para FXS y FXO
- Configurar Asterisk para un E1/T1 digital usando ISDN
- Configurar Asterisk para un E1 digital usando MFC/R2

4.2 Conceptos básicos

La mayor parte de las implementaciones de telefonía analógica usa un par de hilos metálicos (tip and ring). Cuando un loop es Cerrado, el teléfono recibe el tono de discado de la central telefónica sea esta pública (operadora) el privada (PBX). Decimos que este tipo de señalización es de tipo "loop-start". Existen otras señalizaciones como ground-start, por ejemplo, pero son menos comunes.

Existen básicamente tres señalizaciones:

- Señalización de supervisión

³ POTS, Plain Old Telephony System, sistema de telefonía convencional basado en redes de par metálico y líneas TDM.

- Señalización de direccionamiento
- Señalización de información

4.2.1 Señalización de supervisión

Podemos destacar las señales **on-hook** (colgado), **off-hook** (descolgar) y **ringing** (sonando).

On-Hook - Cuando el usuario cuelga el teléfono, el PBX interrumpe y no permite que la corriente sea transmitida. En este caso el circuito esta en estado "on-hook". Cuando el teléfono está en esta posición apenas el "ringer" (campanita) está activo.

Off-Hook – El usuario que desea hacer una llamada telefónica debe pasar para el estado "off-hook" (descolgar), descolgando el teléfono. Este estado cierra el loop eléctrico, el cual indica al PBX que el usuario desea hacer una llamada telefónica. El PBX entonces, luego de recibir esa indicación, genera el tono de discado, indicando la el usuario que está listo para recibir la dirección de destino (número de teléfono).

Ringng – EL usuario al realizar una comunicación, envía un voltaje al "ringer" (campanita) que avisa al otro usuario la recepción de una llamada. La compañía telefónica también manda un tono de vuelta avisando a quien disco el progreso de la llamada.

Existen diferencias en la señalización de tono de discado, tono de ocupado, tono de campanilla (ringng). Usted puede personalizar los tonos de Asterisk para el estándar de su país alterando el archivo **indications.conf**.

```
[br]
descri pti on=Brazi l
ri ngcadance=1000, 4000
di al =425
busy=425/250, 0/250
ri ng=425/1000, 0/4000
congesti on=425/250, 0/250, 425/750, 0/250
cal l wai ti ng=425/50, 0/1000
```

4.2.2 Señalización de direccionamiento

Podemos usar dos tipos de señalización para el discado, el multi frecuencial (dtmf) el pulso (usado en los antiguos teléfonos de disco). Los usuarios que tienen un teclado para discado tienen asociado a cada botón un conjunto de frecuencias altas y bajas. La combinación de estos dos tonos indica para la central cual es el dígito. Esto es conocido como dtmf (dual tone multifrequency).

4.2.3 Señalización de información

La señalización de información muestra el progreso de la llamada y los diferentes eventos. Estos eventos, pueden ser:

- Tono de discado
- Señal de ocupado
- Tono de retorno (ringback)
- Congestionamiento (congestion)
- Número inválido
- Tono de confirmación

4.3 Interfaces de acceso a la red pública

En la mayoría de los casos usted se comunica con su PBX a la red pública de telefonía. En los capítulos posteriores vamos a mostrar como hacer esto. Básicamente existen tres opciones de líneas telefónicas hoy en el mercado. Las líneas analógicas, que son las más comunes y son entregadas usando un par metálico de hilos. Las líneas Digitales son usadas cuando son necesarias muchas líneas analógicas. La línea digital es normalmente entregada a través de un MODEM HDSL de 2 Mbps a través de una línea de fibra óptica con un MUX Digital. La tercera opción, la más reciente, es la entrega de la línea telefónica usando voz sobre IP normalmente a través del protocolo SIP. En Brasil por ejemplo, hasta este momento, Embratel, GVT y Brasil Telecom ya entregan números de teléfono a través del protocolo SIP.

4.4 Usando Interfaces FXS, FXO y E+M

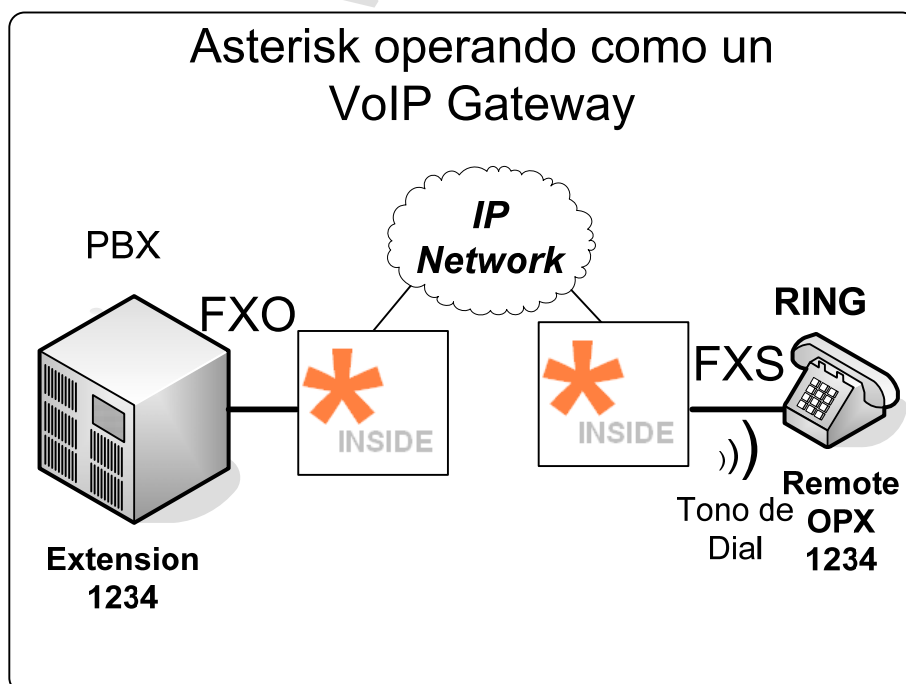


Figura 4.1 Asterisk operando como un VOIP gateway

4.4.1 Interfaces FX (Foreign eXchange)

Son interfaces analógicas. El termino "Foreign eXchange" es aplicado para troncos con acceso a un centro de conmutación de la red pública.

FXO (Foreign eXchange Office)

Básicamente utilizadas para la comunicación con una central de conmutación pública el una puerta de extensión de un PBX. Una puerta FXO en Asterisk se comunica directamente con la red pública el PBX, esta comunicación requiere tono de discado, indicación de campanilla y proveer indicadores de llamadas en progreso. Las interfaces FXO conectan el PBX a otro conmutador (PBX, Red Pública, gateway de voz sobre IP). Es muy común conectar una interface FXO de una central telefónica (ramal) a un gateway VoIP y transportar la voz empaquetada para otro gateway donde una interface FXS conecta un teléfono. Esta operación es conocida como OPX (Off-Promises Extension) el ramal remoto.

FXS (Foreign eXchange Station)

Son las conocidas líneas residenciales estándar. Pueden ser utilizadas para conectar dispositivos básicos: teléfonos, MODEM y faxes. Debe proveer voltaje, generar tono de campanilla, detección de "descolgado" e indicar llamadas en progreso.

4.4.2 Interfaces E & M

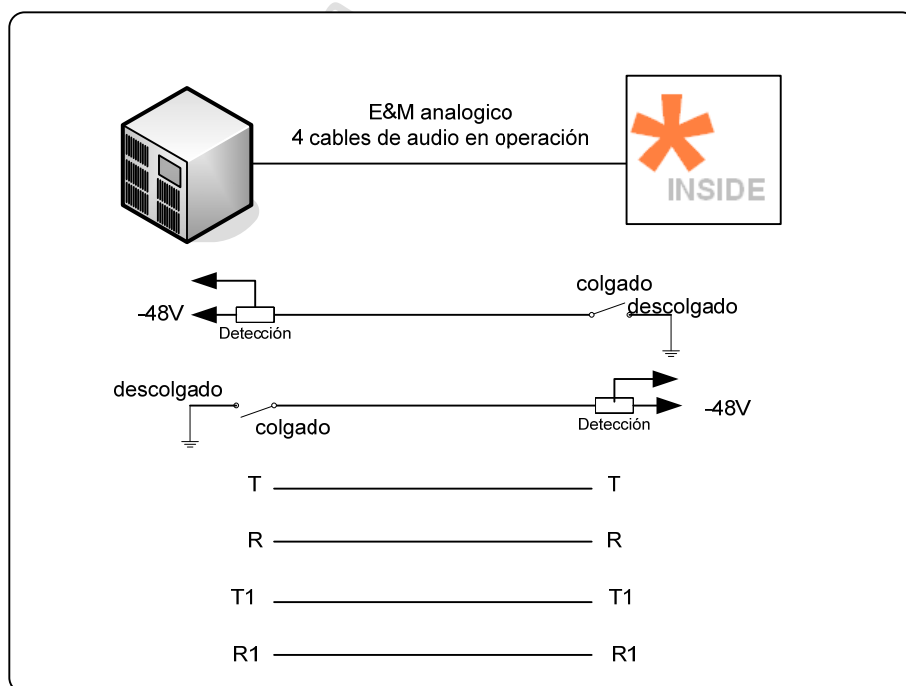


Figure 4.2 Interfaces E&M

Son también interfaces analógicas. El termino "E & M" viene de "Ear (receive) and Mouth (transmit)". Usadas principalmente en las comunicaciones entre PBXs el entre Switches de red-a-red.

Placas E&M para Asterisk no están disponibles. Estas placas son conocidas en el mercado de telefonía como "tie-lines" analógicas. La mayoría de las centrales no vienen con este tipo de interface, mucha de las centrales de marcas conocidas poseían E&M como un opcional. Las placas E&M permiten una comunicación bi-direccional, pueden dar y recibir tono. Si usted precisa usar una interface E&M con Asterisk la mejor opción es la integración de un gateway de voz.

4.4.3 Señalización en los enlaces

- Loop-Start
- Ground-Start
- Kool-Start

Es casi un estándar en Asterisk el uso de Señalización "koolstart". Koolstart no es un estándar de supervisión conocido en la industria, lo que "koolstart" hace es adicionar inteligencia dando a los circuitos la habilidad de monitorear lo que del otro lado se está haciendo. Como "koolstart" incorpora el "loopstart" e el "groundstart", es prácticamente solo el que se usa.

- **Loopstart:** es usada por prácticamente todas las líneas analógicas digitales. Permite al teléfono indicar los estados de "on hook/offhook", al "switch" indicar los estados de "ring/no ring". es el que usted tiene en casa. Cada línea viene en un par separado de hilos, pudiendo ser utilizada tanto para hacer como recibir llamadas. Posee este nombre pues es una línea abierta y la manera para iniciarse las llamadas es cerrándose un "loop", así la central telefónica le proporciona el "dial tone". De la misma manera una llamada entrante es señalizada por 100 V "ringing voltaje" a través de el par abierto, y para responder la llamada, el "loop" Debe ser cerrado.
- **Groundstart:** bastante semejante al "Loopstart". Cuando usted quiere hacer una llamada, uno de los lados del circuito es colocado en corto, cuando la Central identifica este estado, revierte el voltaje la través de el par abierto, solamente de es que el "loop" es cerrado. De esta forma, la línea primeramente se torna ocupada en la Central antes de ser proporcionada para la realización de una llamada.
- **Koolstart:** adiciona inteligencia en habilidad de los circuitos de monitorear lo que del otro lado está haciendo. Desde que

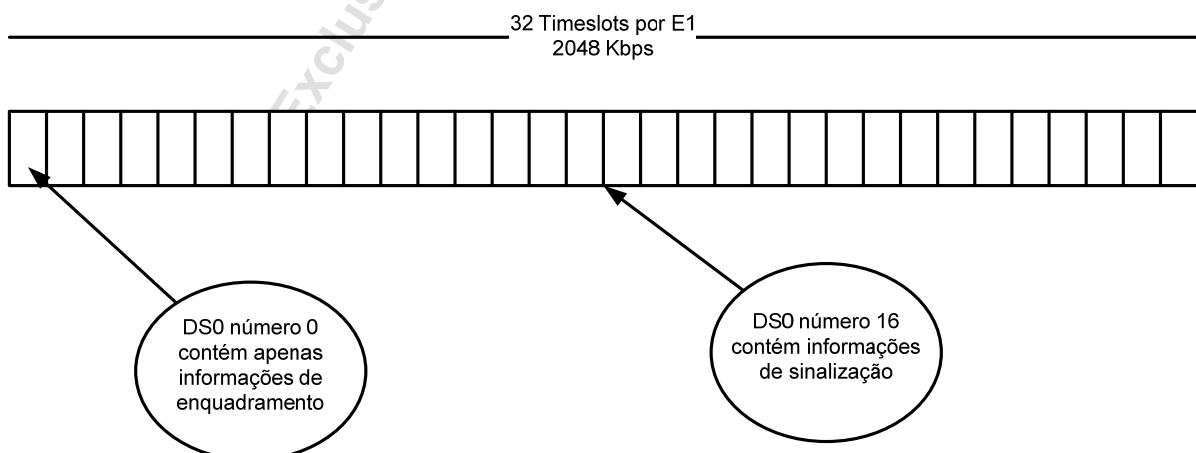
“koolstart” incorporó las ventajas “loopstart” y “groundstart” y siendo superior a ambos, usted probablemente irá a utilizarlo a no ser que haya algún problema de compatibilidad. “Koolstart” se tornó, informalmente, como el estándar para el uso con Asterisk.

4.5 Líneas Digitales E1/T1, Señalización CAS y CCS.

Cuando el número de líneas telefónicas solicitadas por un cliente pasa a ser muy grande, la compañía telefónica normalmente entrega un canal digital. En Brasil el más común es una línea estándar E1 (2 Mbps) con Señalización CAS (Channel Associated Signaling). Normalmente son comercializados 10, 15 o 30 Canales (líneas telefónicas). Algunas compañías ya entregan el canal E1 (2 Mbps) con CCS (Common Channel Signaling) en el estándar ISDN PRI que es más confiable con el Asterisk.

- **ISDN** (Red Digital Integrada de Servicios): es una nueva forma de entroncamiento (desde 1990) Un simple par de Hilos pueden transportar dos líneas y más un circuito de datos de 16kbps usados como señalización. ISDN permite una forma bastante elegante de meter mano a las llamadas. Por ejemplo, servicios como: Caller-ID, llamada en espera, servicios de SMS, entre otros fueron originalmente desarrollados para ISDN.
- **MFC/R2** es una Señalización definida para la ITU (Q.421/Q.441), usada principalmente en América Latina y Asia. La Señalización usa CAS, ahora pasa las señalizaciones de cada canal por el canal 16. El R2 posee variaciones específicas para cada país. Es la señalización de línea digital más común en Brasil.

4.5.1 Señalización y encuadramiento E1



En un encuadramiento E1, 30 de los 32 Canales disponibles son usados para voz, las informaciones de frame usan el primer DS0 y las

informaciones de señalización siguen por el DS0 16. El formato de señalización es conocido como CAS pues los bits del canal 16 son exclusivamente reservados para el propósito de señalar cada canal.

4.5.2 Señalización CAS, E1-R2-Brasil

En Brasil el tipo más común de Señalización CAS es el R2 Digital Brasil. Es posible implementar un canal R2 usando una placa Digium E1 y el driver unicall que puede ser bajado de <http://www.soft-switch.org/unicall/installing-mfcr2.html>. Este driver ha sido usado con éxito en Brasil y en V.Office ya hicimos algunas implementaciones con éxito. Ahora podrá ser usado, siempre que le sea posible use un canal ISDN, es más simple, tiene mejor desempeño y es parte integral de Asterisk. Cabe recordar que el driver para R2 es experimental y no tiene soporte de Digium.

4.5.3 Señalización CCS, E1-ISDN-PRI.

La Señalización ISDN está disponible para Asterisk en diversas variaciones. Estas variaciones son configuradas en el archivo zapata.conf. La red pública en Brasil cuando proporciona el ISDN normalmente usa *switchtype* como EuroISDN. Varias operadoras venden líneas ISDN en Brasil dependiendo de la central telefónica instalada en la ciudad. En centrales telefónicas Digitales, es posible usar ISDN, muchos fabricantes exigen upgrade de hardware y software para esto.

4.6. Configurando un canal de telefonía en Asterisk

Para configurar una placa de telefonía varios pasos son necesarios. En este capítulo vamos a mostrar tres casos comunes.

- Conexión analógica usando una puerta FXS y otra puerta FXO
- Conexión digital usando Señalización ISDN
- Conexión digital usando Señalización R2/Brasil

4.6.1 Procedimientos generales de configuración válidos en los tres casos

Antes de escoger un hardware para el Asterisk usted debe considerar el número de llamadas simultáneas y los Servicios y codecs que serán instalados. Asterisk es intensivo en el uso de hardware, por eso recomendamos una máquina exclusiva para el Asterisk.

El número de placas que pueden ser instalados en Asterisk es limitado por el número de interrupciones disponibles. Es mejor instalar una placa con 4 E1s que cuatro placas de 1 E1. Conflictos de interrupción son comunes,

y el uso del Kernel 2.6 de Linux con motherboards que soportan APIC ayuda bastante en estos casos.

Evite hardware incompatible con las placas ZAPTEL. De una ojeada en la lista de la página de Digium. Una lista puede ser encontrada en el link de abajo.

<http://www.digium.com/index.php?menu=compatibility>

4.7 Ejemplo 1 - Instalación de una puerta FXO y otra FXS

Usaremos como ejemplo la configuración de una placa Digium TDM400P con una interface FXO y otra interface FXS.

Pasos necesarios:

1. Instalación de una placa analógica FXO, FXS o ambas
2. Configuración del archivo `zaptel.conf`
3. Carga del driver de la placa `zaptel`
4. Ejecución del utilitario `zttest`
5. Ejecución del utilitario `ztcfg`
6. Configuración del archivo `zapata.conf`
7. Carga de Asterisk

4.7.1 Instalando una placa TDM400P con una porta FXS y otra puerta FXO.

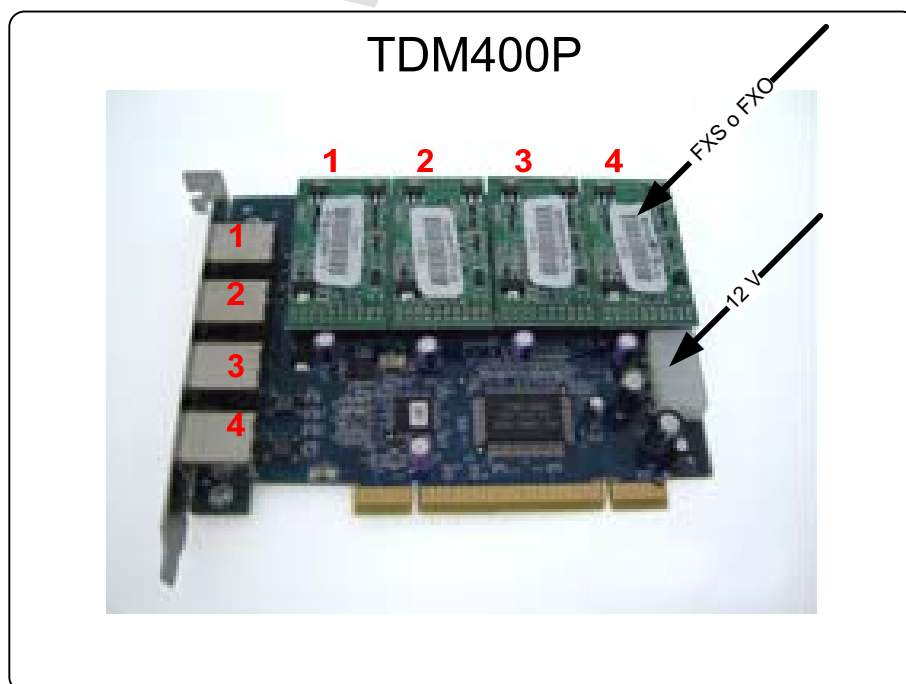


Figura 4.3 placa TDM400P

La placa TDM 400P contiene módulos FXS y FXO de una puerta. Cuando existiesen placas FXS, es preciso alimentar directamente la placa con una fuente de energía (semejante al conector de hard disk). Use protección electrostática para evitar daños en la placa o la de su computador. Conecte la placa a un slot PCI y conecte su máquina. Conecte los módulos FXS (S110M) y FXO (X100M) a la placa TDM400P. Si la placa tiene interfaces FXS alimente el conector de 12V usando un de los cabos de la fuente del PC.

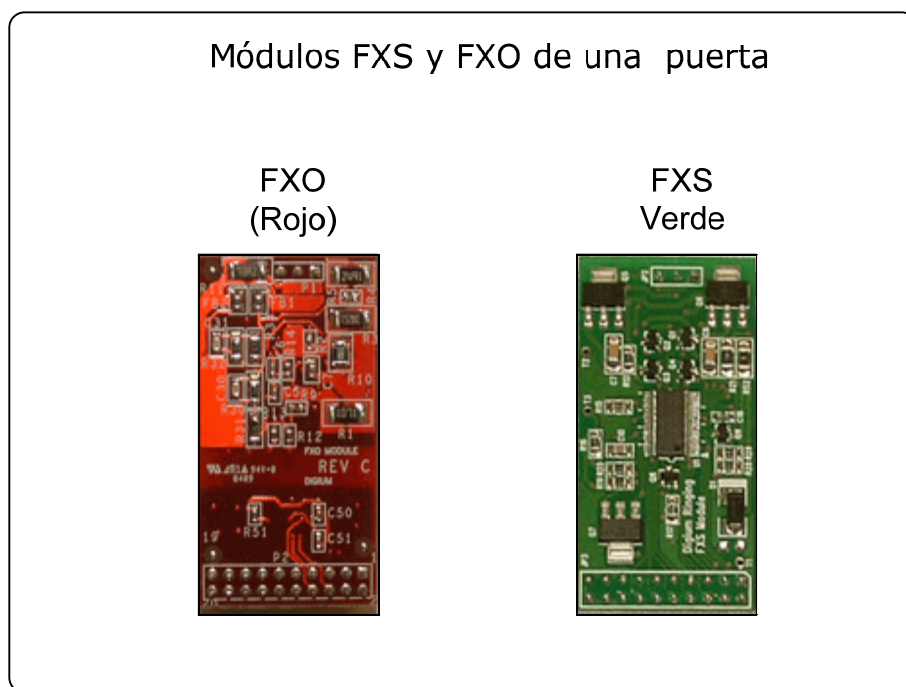


Figura 4.4 En verde el módulo FXS (S110M) y en rojo el FXO (X100M)

4.7.2 Configuración del archivo zaptel.cfg

Para cada placa es preciso configurar el archivo zaptel.conf. Este archivo se encuentra en el directorio /etc al contrario de todos los otros que se encuentran en el directorio /etc/asterisk.

En la TDM400P los módulos FXS son verdes y los módulos FXO son rojos.

```
fxsks=1      # una puerta FXO, certifique que el módulo rojo está más
             # próximo al tornillo de fijación de la placa.
fxoks=2      # una puerta FXS
defaultzone=br
loadzone=br
```

Nota: La señalización de telefonía analógica puede ser una fuente de confusión. Canales FXS son señalizados con Señalización FXO y Canales FXO son señalizados con FXS. Asterisk habla con los dispositivos internos como si estuviese del lado opuesto.

4.7.3 Cargar los drivers del kernel

Como usted ya vio en el capítulo 2, usted debe cargar el módulo zaptel y un módulo correspondiente a la placa que usted está instalando.

Tabla de los drivers de Digium

Placa	Driver	Descripción
TE410P	wct4xyp	4xE1/T1-3.3V PCI
TE405P	wct4xyp	4xE1/T1-5V PCI
TDM400P	Wctdm	4 FXS/FXO
T100P	wct1xyp	1 T1
E100P	Wct1xyp	1 E1
X100P	Wcfxo	1 FXO

```
#modprobe zaptel
#modprobe wctdm
```

4.7.4 Usando el utilitario zttest.

Un utilitario importante es el zttest que puede ser usado para verificar conflictos de interrupción y interrupt misses (interrupciones perdidas). Es muy común problemas en la calidad de audio por causa de conflictos y pérdidas de interrupción. Usted puede usar el comando:

```
#cat /proc/interrupts
```

Para verificar en que interrupción está cada una de las placas.

El zttest permite verificar la cantidad de interrupciones perdidas. un número abajo de 99.98% indica posibles problemas.

4.7.5 Usando el utilitario ztcfg

Luego de cargar el driver, usted debe configurar los canales usando ztcfg.

El comando ztcfg es usado para configurar la señalización usada para la interface física FX. Ztcfg irá a usar la configuración de señalización en zaptel.conf. Para ver la salida del comando usted debe usar -vv para colocar el programa en modo verbo.

```
#!/sbin/ztcfg -vv
Zaptel Configuration
=====
Channel map:
Channel 01: FXS Kewlstart (Default) (Slaves: 01)
Channel 02: FXO Kewlstart (Default) (Slaves: 02)
2 channels configured.
```

Si los canales se cargasen con éxito, usted va a ver una salida similar al recuadro de arriba. Un error común es invertir la señalización entre los canales. Si esto sucede usted va a ver algo parecido como:

ZT_CHANCONFIG failed on channel 1: Invalid argument (22)
 Did you forget that FXS interfaces are configured with FXO signalling
 and that FXO interfaces use FXS signalling?

Luego que los canales hayan sido configurados con éxito, usted está listo para iniciar el uso de su hardware con Asterisk.

4.7.6 Configuración del archivo zapata.conf

Luego de configurar el archivo zaptel.conf, usted tiene el hardware configurado. Este hardware puede ser usado con Asterisk también como otros softwares como Open SS7 y lo mismo para rateo PPP o HDLC. Para usar Asterisk con un hardware zaptel, usted debe ahora configurar el archivo de canales zapata.conf. Este va a crear los canales lógicos TDM que serán usados en su plan de discado.

```

signal ling=fxs_ks;
group=1;           número de grupo de Canales
context=entrada ; contexto de entrada de Los Canales
channel => 1;      número de Los Canales
signal ling=fxo_ks; Señalización de Los módulos fxs
group=2;           número de grupo de Canales
context=ramás;     contexto de Los Canales
channel => 2       número de Los Canales
  
```

4.8 Ejemplo 2 - Carga de dos Canales E1-ISDN

Placa E1 (TE205P)

Pasos necesarios:

1. Instalación de una placa digital TE205P o TE210P
2. Configuración del archivo zaptel.conf
3. Carga de driver de placa zaptel
4. Ejecución del utilitario zttest
5. Ejecución del utilitario ztcfg
6. Configuración del archivo zapata.conf
7. Carga y teste con Asterisk

4.8.1 Instalación de la placa TE205P

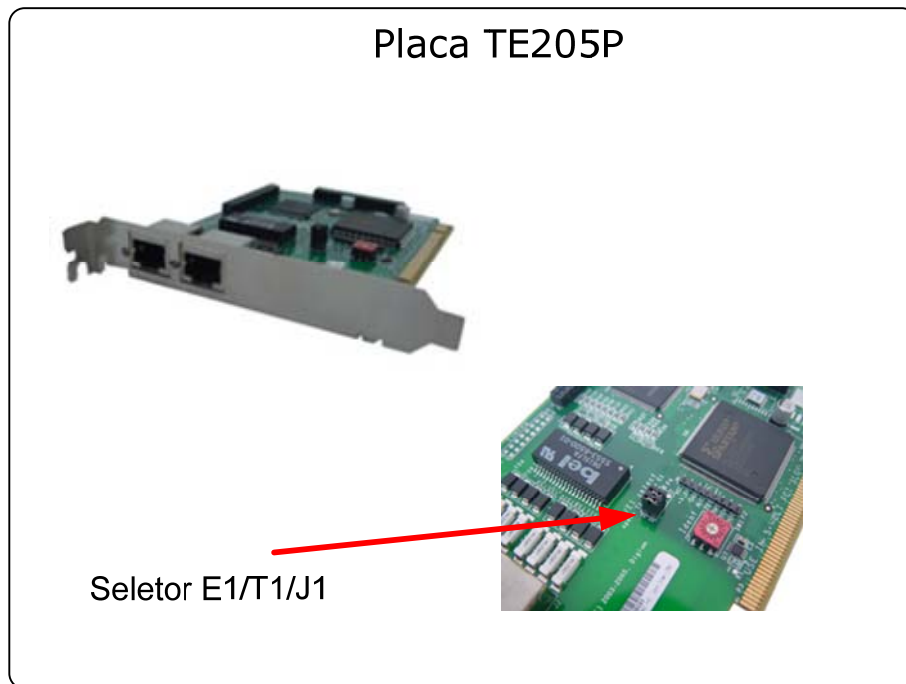


Figura 4.5 Placa TDM 205P

En primer lugar es importante entender las implicaciones de escoger una placa Digium TE205P o TE210P. La placa TE210P es una placa que usa un bus de datos de 64 bits de 3.3 volts. Este bus solo es encontrado en PCs servidores de red. Cuidado al especificar esta placa, certifíquese que su hardware va operar normalmente. La placa TE205P usa un bus PCI común de 5 volts y el hardware disponible es mucho mayor. Escogemos una placa con dos interfaces E1 para el ejemplo, pues es fácil reducirlo para los casos de una interface E1 apenas.

Cuidado al especificar: Certifíquese de estar comprando la placa correcta. Es muy común que las personas especificasen las placas de 3.3V (TE210P, TE410P, TE211P e TE411P) y devuelvan estas placas por no encontrar una motherboard compatible. Como ya dijimos estos slots de 3.3V 64 bits PCI-X solo son encontrados en motherboard para servidores.

4.8.2 Configuración del archivo zaptel.conf

Para configurar la placa E1 es un poco diferente todo. Primero precisamos definir el span (pin) y después los canales. Los spans son numerados a partir de la secuencia de reconocimiento de las placas en el hardware. En otras palabras no es necesario saber si usted tiene más de una placa cual es el span. Normalmente usamos un cabo con la señal ISDN y colocamos en una de las placas, en la consola Asterisk, tenemos el mensaje *"primary span X UP"*. En mi opinión esta es la forma más simples de detectar en que span esta la placa configurada.

```
span=1, 1, 0, ccs, hdb3, crc4 # no siempre crc4 es necesario, ver operadora
span=2, 0, 0, ccs, hdb3, crc4
bchan=1-15, 17-31
dchan=16
bchan=33-47, 49-63
dchan=48
default tzone=br
loadzone=br
```

4.8.3 Carga del driver de la placa zaptel

Como usted ya vio en el capítulo 2, usted debe cargar el módulo zaptel y un módulo correspondiente a la placa que usted está instalando.

Tabla de los drivers de Digium

Placa	Driver	Descripción
TE410P	wct4xxp	4xE1/T1-3.3V PCI
TE405P	wct4xxp	4xE1/T1-5V PCI
TE210P	wct2xxp	2xE1/T1-3.3V PCI
TE205P	wct2xxp	2xE1/T1-5V PCI
TDM400P	Wctdm	4 FXS/FXO
T100P	wct1xxp	1 T1
E100P	Wctlxxp	1 E1
X100P	Wcfxo	1 FXO

```
#modprobe zaptel
#modprobe wct2xxp
```

4.8.4 Usando el utilitario zttest

Un utilitario importante es el zttest que puede ser usado para verificar conflictos de interrupción e interrupt misses. Es muy común problemas de calidad de audio por causa de conflictos y pérdidas de interrupción. Usted puede usar el comando zttest dentro de /usr/src/asterisk. el número de interrupciones no debe ser peor que 99.987793%.

```
Opened pseudo zap interface, measuring accuracy...
99.987793% 100.000000% 100.000000% 100.000000% 100.000000% 100.000000%
100.000000%
100.000000% 100.000000% 100.000000% 100.000000% 100.000000% 100.000000%
100.000000% 100.000000%
100.000000% 100.000000% 100.000000% 100.000000% 99.987793% 100.000000%
100.000000% 100.000000%
100.000000% 100.000000% 100.000000%
--- Results after 26 passes ---
Best: 100.000000 -- Worst: 99.987793 -- Average: 99.999061
```

4.8.5 Verificando conflictos de interrupción

Para verificar en que interrupción está cada una de las placas use:

```
#cat /proc/interrupts
CPU0
0: 3675537      XT-PIC timer
1:    2        XT-PIC keyboard
2:    0        XT-PIC cascade
3: 410705      XT-PIC eth1
4: 1694897    XT-PIC eth0
5: 36737213   XT-PIC t1xxp
7:  14703     XT-PIC libata
8:    4       XT-PIC rtc
10: 36742553  XT-PIC wcfxo
11: 36742513  XT-PIC wcfxo
14:  55631    XT-PIC ide0
NMI:    0
LOC: 3675478
ERR:    0
MIS:    0
```

4.8.6 Usando el utilitario ztcfg

Esta es la salida del comando `ztcfg -vvv` de un servidor Asterisk con una placa E1 configurada para quince canales más dos placas FXO.

```
Zaptel Configuration
=====

SPAN 1: CCS/HDB3 Build-out: 0 db (CSU)/0-133 feet (DSX-1)

Channel map:

Channel 01: Clear channel (Default) (Slaves: 01)
Channel 02: Clear channel (Default) (Slaves: 02)
Channel 03: Clear channel (Default) (Slaves: 03)
Channel 04: Clear channel (Default) (Slaves: 04)
Channel 05: Clear channel (Default) (Slaves: 05)
Channel 06: Clear channel (Default) (Slaves: 06)
Channel 07: Clear channel (Default) (Slaves: 07)
Channel 08: Clear channel (Default) (Slaves: 08)
Channel 09: Clear channel (Default) (Slaves: 09)
Channel 10: Clear channel (Default) (Slaves: 10)
Channel 11: Clear channel (Default) (Slaves: 11)
Channel 12: Clear channel (Default) (Slaves: 12)
Channel 13: Clear channel (Default) (Slaves: 13)
Channel 14: Clear channel (Default) (Slaves: 14)
Channel 15: Clear channel (Default) (Slaves: 15)
Channel 16: D-channel (Default) (Slaves: 16)
Channel 32: FXS Kewlstart (Default) (Slaves: 32)
Channel 33: FXS Kewlstart (Default) (Slaves: 33)

18 channels configured.
```

4.8.7 Configuración de los Canales "zapata.conf"

```

cal leri d=<4830258580>
swi tchtype=euroi sdn
si gnalling = pri _cpe
group = 1
channel => 1-15; 17-31
group =2
channel => 32-46; 48-62

```

4.8.8 Comandos para verificación

zap show status: Muestra el status de las placas zaptel

```

vtsvoffi ce*CLI> zap show status
Description                    Al arms      IRQ          bpvi ol
CRC4
Di gi um Wi l dcard E100P E1/PRA Card 0    OK           0            0            0
Wi l dcard X100P Board 1                OK           0            0            0
Wi l dcard X100P Board 2                RED          0            0            0

```

pri show span: Permite verificar los datos de link ISDN.

```

vtsvoffi ce*CLI> pri show span 1
Primary D-channel: 16
Status: Provi sioned, Up, Acti ve
Swi tchtype: EuroI SDN
Type: CPE
Wi ndow Length: 0/7
Sentrej: 0
Sol i ci tFbi t: 0
Retrans: 0
Busy: 0
Overl ap Di al : 0
T200 Ti mer: 1000
T203 Ti mer: 10000
T305 Ti mer: 30000
T308 Ti mer: 4000
T313 Ti mer: 4000
N200 Counter: 3

```

zap show channel: Muestra los Canales zap configurados.

```

vtsvoffi ce*CLI> zap show channels
Chan Extensi on  Context      Language  Musi cOnHol d
pseu do         fax          en         defaul t
1               entrada     en         defaul t
2               entrada     en         defaul t
3               entrada     en         defaul t
4               entrada     en         defaul t
5               entrada     en         defaul t
6               entrada     en         defaul t
7               entrada     en         defaul t
8               entrada     en         defaul t
9               entrada     en         defaul t
10              entrada     en         defaul t
11              entrada     en         defaul t
12              entrada     en         defaul t
13              entrada     en         defaul t
14              entrada     en         defaul t
15              entrada     en         defaul t
32              defaul t    en         defaul t
33              fax         en         defaul t

```

zap show channel x: Muestra un canal específico.

```
vtsvoffice*CLI> zap show channel 1
Channel: 1*CLI>
File Descriptor: 21
Span: 1
Extension:
Dialing: no
Context: entrada
Caller ID: 4832341689
Calling TON: 33
Caller ID name:
Destroy: 0
InAlarm: 0
Signalling Type: PRI Signalling
Radio: 0
Owner: <None>
Real: <None>
Callwait: <None>
Threeway: <None>
Confno: -1
Propagated Conference: -1
Real in conference: 0
DSP: no
Relax DTMF: no
Dialing/CallwaitCAS: 0/0
Default law: alaw
```

debug pri span x: Habilita un debugging detallado de las llamadas ISDN. Note que es posible verificar exactamente que números están siendo discados y el identificador de llamada. Abajo sigue el debug de una llamada que procedió normalmente, esta puede ser útil en la comparación con una comunicación con problemas. Una pista es desconectar el modo verbo (set verbose=0) antes de emitir el comando para tener apenas los mensajes Q.931 de ISDN.

```
-- Making new call for cr 32833
> Protocol Discriminator: Q.931 (8) len=57
> Call Ref: len= 2 (reference 65/0x41) (Originator)
> Message type: SETUP (5)
> [04 03 80 90 a3]
> Bearer Capability (len= 5) [ Ext: 1 Q.931 Std: 0 Info transfer
capability: Speech (0)
> Ext: 1 Trans mode/rate: 64kbps, circuit-
mode (16)
> Ext: 1 User information layer 1: A-Law
(35)
> [18 03 a9 83 81]
> Channel ID (len= 5) [ Ext: 1 IntID: Implicit, PRI Spare: 0, Exclusive
Dchan: 0
> ChanSel: Reserved
> Ext: 1 Coding: 0 Number Specified Channel
Type: 3
> Ext: 1 Channel: 1 ]
> [28 0e 46 6c 61 76 69 6f 20 45 64 75 61 72 64 6f]
> Display (len=14) @h@>[ Flavio Eduardo ]
> [6c 0c 21 80 34 38 33 30 32 35 38 35 39 30]
> Calling Number (len=14) [ Ext: 0 TON: National Number (2) NPI:
ISDN/Telphony Numbering Plan (E.164/E.163) (1)
> Presentation: Presentation permitted, user
number not screened (0) '4830258590' ]
> [70 09 a1 33 32 32 34 38 35 38 30]
> Called Number (len=11) [ Ext: 1 TON: National Number (2) NPI:
ISDN/Telphony Numbering Plan (E.164/E.163) (1) '32248580' ]
```

```

> [a1]fi ce*CLI >
> Sending Complete (len= 1)
< Protocol Discriminator: Q.931 (8) len=10
< Call Ref: len= 2 (reference 65/0x41) (Terminator)
< Message type: CALL PROCEEDING (2)
< [18 03 a9 83 81]
< Channel ID (len= 5) [ Ext: 1 IntID: Implicit, PRI Spare: 0, Exclusive
Dchan: 0
<
< ChanSel: Reserved
< Ext: 1 Coding: 0 Number Specified Channel
Type: 3
< Ext: 1 Channel: 1 ]
-- Processing IE 24 (cs0, Channel Identification)
< Protocol Discriminator: Q.931 (8) len=9
< Call Ref: len= 2 (reference 65/0x41) (Terminator)
< Message type: ALERTING (1)
< [1e 02 84 88]
< Progress Indicator (len= 4) [ Ext: 1 Coding: CCITT (ITU) standard (0)
0: 0 Location: Public network serving the remote user (4)
< Ext: 1 Progress Description: Inband
information or appropriate pattern now available. (8) ]
-- Processing IE 30 (cs0, Progress Indicator)
< Protocol Discriminator: Q.931 (8) len=64
< Call Ref: len= 2 (reference 5720/0x1658) (Originator)
< Message type: SETUP (5)
< [04 03 80 90 a3]
< Bearer Capability (len= 5) [ Ext: 1 Q.931 Std: 0 Info transfer
capability: Speech (0)
< Ext: 1 Trans mode/rate: 64kbps, circuit-
mode (16)
< Ext: 1 User information layer 1: A-Law
(35)
< [18 03 a1 83 82]
< Channel ID (len= 5) [ Ext: 1 IntID: Implicit, PRI Spare: 0, Preferred
Dchan: 0
<
< ChanSel: Reserved
< Ext: 1 Coding: 0 Number Specified Channel
Type: 3
< Ext: 1 Channel: 2 ]
< [1c 15 91 a1 12 02 01 bc 02 01 0f 30 0a 02 01 01 0a 01 00 a1 02 82 00]
< Facility (len=23, codeset=0) [ 0x91, 0xa1, 0x12, 0x02, 0x01, 0xbc, 0x02,
0x01, 0x0f, '0', 0x0a, 0x02, 0x01, 0x01, 0x0a, 0x01, 0x00, 0xa1, 0x02,
0x82, 0x00 ]
< [1e 02 82 83]
< Progress Indicator (len= 4) [ Ext: 1 Coding: CCITT (ITU) standard (0)
0: 0 Location: Public network serving the local user (2)
< Ext: 1 Progress Description: Calling
equipment is non-ISDN. (3) ]
< [6c 0c 21 83 34 38 33 32 32 34 38 35 38 30]
< Calling Number (len=14) [ Ext: 0 TON: National Number (2) NPI:
ISDN/Telephony Numbering Plan (E.164/E.163) (1)
< Presentation: Presentation allowed of network
provided number (3) '4832248580' ]
< [70 05 c1 38 35 38 30]
< Called Number (len= 7) [ Ext: 1 TON: Subscriber Number (4) NPI:
ISDN/Telephony Numbering Plan (E.164/E.163) (1) '8580' ]
< [a1]
< Sending Complete (len= 1)
-- Making new call for cr 5720
-- Processing Q.931 Call Setup
-- Processing IE 4 (cs0, Bearer Capability)
-- Processing IE 24 (cs0, Channel Identification)
-- Processing IE 28 (cs0, Facility)
Handle Q.932 ROSE Invoke component
-- Processing IE 30 (cs0, Progress Indicator)
-- Processing IE 108 (cs0, Calling Party Number)
-- Processing IE 112 (cs0, Called Party Number)
-- Processing IE 161 (cs0, Sending Complete)
> Protocol Discriminator: Q.931 (8) len=10

```

```

> Call Ref: len= 2 (reference 5720/0x1658) (Terminator)
> Message type: CALL PROCEEDING (2)
> [18 03 a9 83 82]
> Channel ID (len= 5) [ Ext: 1 IntID: Implicit, PRI Spare: 0, Exclusive
Dchan: 0
>
> ChanSel: Reserved
> Ext: 1 Coding: 0 Number Specified Channel
Type: 3
> Ext: 1 Channel: 2 ]
> Protocol Discriminator: 0.931 (8) len=14
> Call Ref: len= 2 (reference 5720/0x1658) (Terminator)
> Message type: CONNECT (7)
> [18 03 a9 83 82]
> Channel ID (len= 5) [ Ext: 1 IntID: Implicit, PRI Spare: 0, Exclusive
Dchan: 0
>
> ChanSel: Reserved
> Ext: 1 Coding: 0 Number Specified Channel
Type: 3
> Ext: 1 Channel: 2 ]
> [1e 02 81 82]
> Progress Indicator (len= 4) [ Ext: 1 Coding: CCITT (ITU) standard (0)
0: 0 Location: Private network serving the local user (1)
> Ext: 1 Progress Description: Called
equipment is non-ISDN. (2) ]
< Protocol Discriminator: 0.931 (8) len=5
< Call Ref: len= 2 (reference 5720/0x1658) (Originator)
< Message type: CONNECT ACKNOWLEDGE (15)
< Protocol Discriminator: 0.931 (8) len=9
< Call Ref: len= 2 (reference 65/0x41) (Terminator)
< Message type: PROGRESS (3)
< [1e 02 84 82]
< Progress Indicator (len= 4) [ Ext: 1 Coding: CCITT (ITU) standard (0)
0: 0 Location: Public network serving the remote user (4)
< Ext: 1 Progress Description: Called
equipment is non-ISDN. (2) ]
-- Processing IE 30 (cs0, Progress Indicator)
< Protocol Discriminator: 0.931 (8) len=5
< Call Ref: len= 2 (reference 65/0x41) (Terminator)
< Message type: CONNECT (7)
> Protocol Discriminator: 0.931 (8) len=5
> Call Ref: len= 2 (reference 65/0x41) (Originator)
> Message type: CONNECT ACKNOWLEDGE (15)
NEW_HANGUP DEBUG: Calling q931_hangup, ourstate Active, peerstate Connect
Request
> Protocol Discriminator: 0.931 (8) len=9
> Call Ref: len= 2 (reference 65/0x41) (Originator)
> Message type: DISCONNECT (69)
> [08 02 81 90]
> Cause (len= 4) [ Ext: 1 Coding: CCITT (ITU) standard (0) 0: 0
Location: Private network serving the local user (1)
> Ext: 1 Cause: Unknown (16), class = Normal Event (1) ]
< Protocol Discriminator: 0.931 (8) len=5
< Call Ref: len= 2 (reference 65/0x41) (Terminator)
< Message type: RELEASE (77)
NEW_HANGUP DEBUG: Calling q931_hangup, ourstate Null, peerstate Release
Request
> Protocol Discriminator: 0.931 (8) len=9
> Call Ref: len= 2 (reference 65/0x41) (Originator)
> Message type: RELEASE COMPLETE (90)
> [08 02 81 90]
> Cause (len= 4) [ Ext: 1 Coding: CCITT (ITU) standard (0) 0: 0
Location: Private network serving the local user (1)
> Ext: 1 Cause: Unknown (16), class = Normal Event (1) ]
NEW_HANGUP DEBUG: Calling q931_hangup, ourstate Null, peerstate Null
NEW_HANGUP DEBUG: Destroying the call, ourstate Null, peerstate Null
< Protocol Discriminator: 0.931 (8) len=9
< Call Ref: len= 2 (reference 5720/0x1658) (Originator)
< Message type: DISCONNECT (69)
< [08 02 82 90]

```



```

< Cause (len= 4) [ Ext: 1 Coding: CCITT (ITU) standard (0) 0: 0
Location: Public network serving the local user (2)
< Ext: 1 Cause: Unknown (16), class = Normal Event (1) ]
-- Processing IE 8 (cs0, Cause)
NEW_HANGUP DEBUG: Calling q931_hangup, ourstate Disconnect Indication,
peerstate Disconnect Request
> Protocol Discriminator: 0.931 (8) len=9
> Call Ref: len= 2 (reference 5720/0x1658) (Terminator)
> Message type: RELEASE (77)
> [08 02 81 90]
> Cause (len= 4) [ Ext: 1 Coding: CCITT (ITU) standard (0) 0: 0
Location: Private network serving the local user (1)
> Ext: 1 Cause: Unknown (16), class = Normal Event (1) ]
< Protocol Discriminator: 0.931 (8) len=5
< Call Ref: len= 2 (reference 5720/0x1658) (Originator)
< Message type: RELEASE COMPLETE (90)
NEW_HANGUP DEBUG: Calling q931_hangup, ourstate Null, peerstate Null
NEW_HANGUP DEBUG: Destroying the call, ourstate Null, peerstate Null

```

4.9 Configurando MFC/R2

4.9.1 Entendiendo el problema

La placa usada para la Señalización R2 es la misma usada para la señalización ISDN. El propio Digium posee dentro del código de canal zapata señalización R2, más de acuerdo con Mark Spencer está lejos de estar implementado. Para usar la señalización R2 vamos usar el driver Unicall desarrollado por Steve Underwood y esta disponible en www.soft-switch.org. Lo ideal es usar ISDN que es más estable y posee soporte de Digium, la dificultad es que no siempre está disponible. La gran mayoría de las líneas en América del Sur usan Señalización R2, también conocida como R2-Digital.

4.9.2 Entendiendo el protocolo MFC/R2

En el protocolo MFC/R2 el direccionamiento telefónico es encaminado a través de un conjunto de tonos. Entretanto la Señalización del canal es pasada a través de un timeslot específico. En este timeslot (16) son transferidos los bits ABCD de cada canal de voz a través de los cuales es hecho el control de la llamada.

Los bits C e D raramente son utilizados. En algunos países estos pueden ser usados para Metering (medición de pulsos para tarificación). En una conversación normal tenemos dos lados operando, el lado de origen de la llamada y el lado receptor de la llamada. El lado de origen es referido como señalización hacia adelante y el receptor con señalización para atrás. Vamos a designar de aquí en adelante Af y Bf para los bits de adelante y At y Bt para los bits de atrás.

Vea la tabla abajo:

Estado	ABCD p/ Frente	ABCD p/ Trás
Idle/Released (libre)	1001	1001

Seized (Ocupado)	0001	1001
Seize Ack (Confirmación de ocupado, Ring)	0001	1101
Answered (En conversación)	0001	0101
ClearBack (Desconexión por la parte llamada)	0001	1101
ClearFwd (Antes de Clear-Back)	1001	0101
ClearFwd (Después de Clear-Back)		
Blocked (Bloqueado)	1001	1101

Diferencias para el estándar brasileño.

A pesar de R2 ha sido definido por la ITU existen variaciones en relación a la implementación hecha en cada país. Siguen abajo las tablas referentes al Brasil.

Estado	ABCD p/ Frente	ABCD p/ Trás
Idle/Released (libre)	1001	1001
Seized (Ocupado)	0001	1001
Seize Ack (Confirmación de ocupado, Ring)	0001	1101
Llamada en progreso	0001	1101
Atención de la Llamada	0001	0101
Answered (En conversación)	0001	0101
Tarificación	0001	1101
ClearBack (Desconexión por la parte llamada)	0001	1101
ClearFwd (Antes de Clear-Back)	1001	0101
ClearFwd (Confirmación de desconexión)	1001	1001
Desconexión forzada	0001	0001
Blocked (Bloqueado)	1001	1101
Confirmación de desconexión forzada	1001	0001
Falla	1101	1001

Señalización entre registradores.

La Señalización MFC usa combinaciones de dos tonos conforme la tabla a continuación. El cuadro de abajo presenta los significados para la norma brasileña.

Grupo de señales I (adelante)

Señales	Descripción	Señal enfrente
1	Algoritmo 1	I-1
2	Algoritmo 2	I-2
3	Algoritmo 3	I-3
4	Algoritmo 4	I-4
5	Algoritmo 5	I-5
6	Algoritmo 6	I-6
7	Algoritmo 7	I-7
8	Algoritmo 8	I-8
9	Algoritmo 9	I-9
10	Algoritmo 0	I-10
11	Intersección semi-supresor de eco	I-11
12	Pedido recusado o indicación de tránsito internacional	I-12
13	Acceso equipamiento de test	I-13
14	Inserción de semi-supresor de eco de destino o indicación de tránsito internacional	I-14
15	Fin de número o indicación de que la llamada cursó enlace vía satélite	I-15

Grupo de señales II (enfrente)

Señales	Descripción	Señal à frente
1	Suscriptor común	II-1
2	Suscriptor con tarificación especial	II-2
3	Equipamiento de manutención	II-3
4	Teléfono público local	II-4
5	Telefonista	II-5
6	Transmisión de datos	II-6
7	Teléfono público interurbano	II-7
8	Llamada a cobrar	II-8
9	Suscriptor común – servicio entrante internacional	II-9
10	Reservado	II-10
11	Indicativo de llamada transferida	II-11
12	Confuso o Inestable	II-12
13	Confuso o Inestable	II-13
14	Confuso o Inestable	II-14
15	Confuso o Inestable	II-15

Grupo de señales A (de atrás)

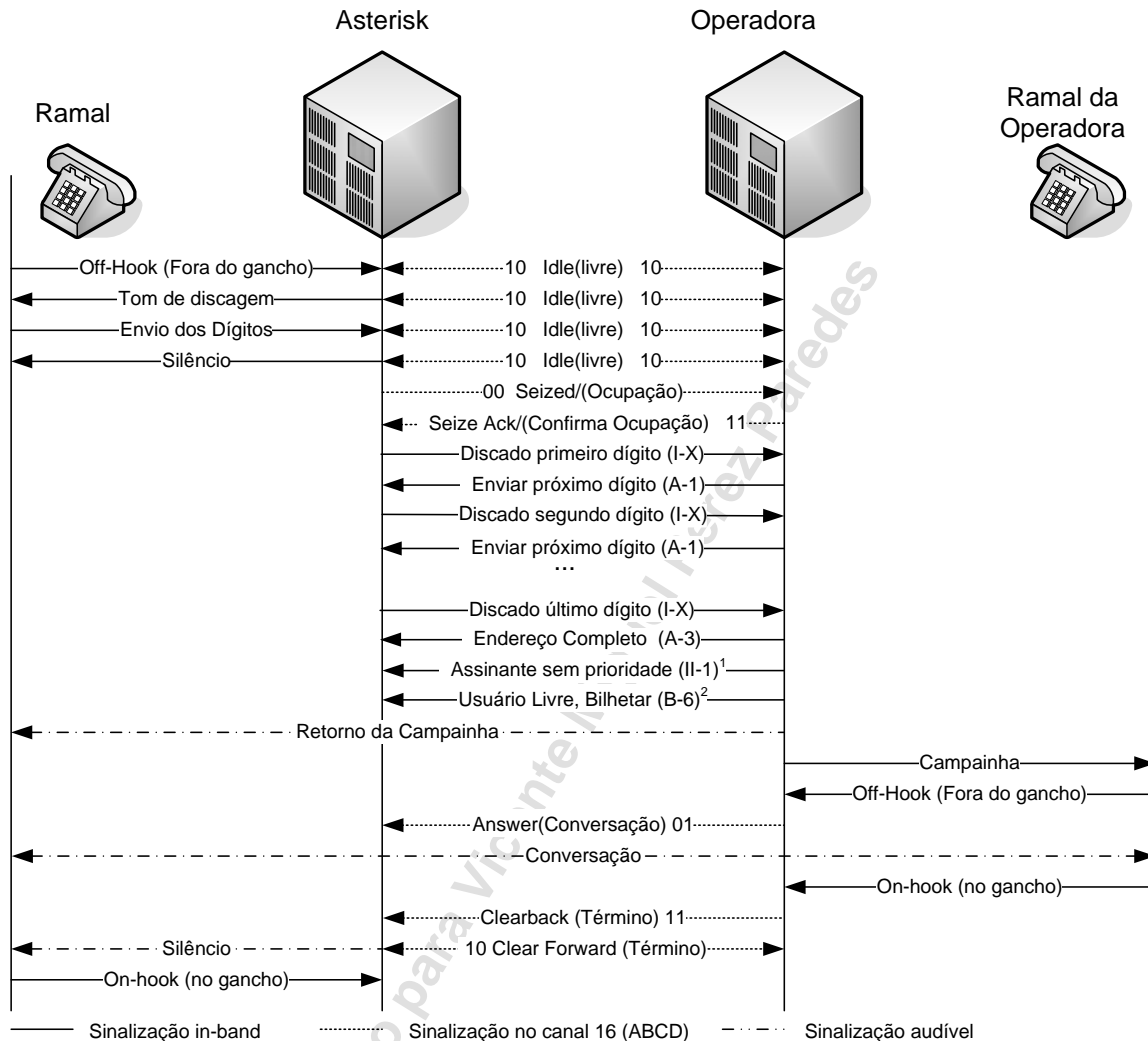
Señales	Descripción	Señal delante
1	Enviar el próximo Algoritmo (n+1)	A-1
2	Necesidad de semi-supresor de eco en destino o enviar el primero Algoritmo enviado	A-2
3	Preparar recepción de señales de grupo B	A-3
4	Congestionamiento de red nacional	A4
5	Enviar categoría y identidad de Suscriptor llamado	A5
6	Reservado	A6
7	Enviar Algoritmo N-2	A7
8	Enviar Algoritmo N-3	A8
9	Envía Algoritmo N-1	A9
10	Inestable para uso nacional	A10
11	Reservado	A11
12	Reservado	A12
13	Reservado	A13
14	Reservado	A14
15	Reservado	A15

Grupo de señales B (de atrás)

Señales	Descripción	Señal p/atrás
1	Línea de suscriptor libre con tarificación	B1
2	Línea de suscriptor ocupada	B2
3	Línea de suscriptor con número cambiado	B3
4	Congestionamiento	B4
5	Línea de suscriptor libre sin tarificación	B5
6	Línea de suscriptor libre con tarificación y colocar retención sobre control de suscriptor llamado	B6
7	Número confuso	B7
8	Línea de suscriptor fuera de servicio para tráfico terminado	B8
9	Reservado	B9
10	Reservado	B10
11	Reservado	B11
12	Reservado	B12
13	Reservado	B13

14	Reservado	B14
15	Reservado	B15

4.9.3 Secuencia de una llamada MFC/R2



La secuencia de arriba es de una llamada de un ramal de un PBX para una Terminal de red pública que enseguida desconecta el teléfono y finaliza la comunicación.

4.9.4 el driver Unicall

El driver unicall fue desarrollado por Steve Underwood y es distribuido gratuitamente. Este no es parte de Asterisk y no es soportado por Digium. Este fue desarrollado con base al canal zaptel (chan_cap.c).

El conjunto de recursos usados por el canal zapata sigue abajo

PSTN->PLACA ZAPTEL->DRIVER ZAPTEL->LIBPRI->CHAN_ZAP->ASTERISK

En el driver Unicall la secuencia es un poco diferente y el conjunto de recursos es como sigue abajo:

PSTN->PLACA ZAPTEL->DRIVER ZAPTEL->LIMFCR2->LIBUNICALL->CHAN_UNICALL->ASTERISK

4.9.5 Configurando la placa zaptel para operar con MFC/R2

Asumiendo que la Instalación y compilación del driver zaptel ya fue hecha en el capítulo 3, vamos a comenzar por la configuración del driver zaptel para el uso con el MFC/R2. En primer lugar vamos a colocar los canales en modo Bloqueado.

```
# MFC/R2 normalmente no usa CRC4
span=1, 1, 0, cas, hdb3
cas=1-15: 1101 ; Dependiente del variante R2 (bits ABCD de modo blocked)
cas=17-31: 1101
span=2, 0, 0, cas, hdb3
cas=33-47: 1101
cas=49-63: 1101
loadzone=br
defaultzone=br
```

Colocar los bits en bloqueado es importante para que la operadora no entienda que en la carga de la placa de Asterisk esta listo para recibir llamadas. La placa simplemente esta conectada, pero no lista para recibir llamadas.

Ejecutar ZTCFG.

```
bash# ztcfg -v
```

```
Zaptel Configuration
```

```
=====
```

```
SPAN 1: CAS/HDB3 Build-out: 0 db (CSU)/0-133 feet (DSX-1)
```

```
SPAN 2: CAS/HDB3 Build-out: 0 db (CSU)/0-133 feet (DSX-1)
```

```
62 channels configured.
```

4.9.6 Instalando y compilando las bibliotecas

Para instalar el R2 es preciso bajar y compilar las bibliotecas spandsp, libsupertone, libmfc2 y libunicall a partir de ftp.soft-switch.org. En la versión 1.4 Steve Underwood no está mas manteniendo lo código. Existe un patch en Internet que compila en la versión 1.4, (gracias a Moisés Humberto Silva Salmerón).

Instalando las dependencias

```
#apt-get install libtiff4-dev
#apt-get install libxml2-dev
```

spandsp

```
#cd /usr/src
#wget http://www.moythreads.com/unicall/soft-switch/r1b1/spandsp/spandsp-0.0.3pre27.tar.gz
#tar -xzvf spandsp-0.0.3pre27.tar.gz
#cd spandsp-0.0.3
# ./configure --prefix=/usr
#make
#make install
```

libsupertone

```
#cd /usr/src
#wget http://www.moythreads.com/unicall/soft-switch/r1b1/unicall/libsupertone-0.0.2.tar.gz
#tar -xzvf libsupertone-0.0.2.tar.gz
#cd libsupertone-0.0.2
# ./configure --prefix=/usr
#make
#make install
```

libunicall

```
#cd /usr/src
#wget http://www.moythreads.com/unicall/soft-switch/r1b1/unicall/libunicall-0.0.3_1.4.tar.bz2
#tar -xvf libunicall-0.0.3_1.4.tar.bz2
#cd libunicall-0.0.3_1.4
# ./configure --prefix=/usr
#make
#make install
```

libmfcr2

```
#cd /usr/src
#wget http://www.moythreads.com/unicall/soft-switch/r1b1/unicall/libmfcr2-0.0.3_1.4.tar.bz2
#tar -xvf libmfcr2-0.0.3_1.4.tar.bz2
#cd libmfcr2-0.0.3_1.4
# ./configure --prefix=/usr
#make
#make install
```

4.9.7 Integrating Unicall to Asterisk

Baje los archivos relativos al chan_unicall.

```
#wget http://www.moythreads.com/unicall/soft-switch/r1b1/unicall/asterisk-1.4.0/chan_unicall.c
#wget http://www.moythreads.com/unicall/soft-switch/r1b1/unicall/asterisk-1.4.0/channels_Makefile.patch
#wget http://www.moythreads.com/unicall/soft-switch/r1b1/unicall/asterisk-1.4.0/unicall.conf.sample
```

Copie los archivos para la estructura donde el asterisk será compilado

```
# cp chan_unicall.c channels_Makefile.patch /usr/src/asterisk/channels
```

Aplique el patch del canal de Asterisk

```
#cd /usr/src/asterisk/channels
#patch < channels_makefile.patch
```

Enseguida compile el Asterisk

```
cd /usr/src/asterisk/
make clean
./configure
make menuselect
make
make install
```

4.9.8 Configurando el canal unicall

Así como para configurar los Canales zaptel, usted editaba un archivo llamado zapata.conf, ahora para los Canales R2 usted editará el archivo unicall.conf. Este archivo tiene sintaxis y atributos casi idénticos al zapata.conf de forma que no será difícil entenderlo. Vea abajo el archivo editado.

```
;
;
; Uni call telephony channel driver
;
; Sample configuration file
;
; $Id: unicall.conf.sample,v 1.1 2005/05/28 11:17:02 steveu Exp $
;
[channel s]
;
; Default language
;
language=br
;
; Default context
;
context=default
;
; Whether or not to use caller ID
;
usecallerid=yes
;
; Whether or not to hide outgoing caller ID
;
hidecallerid=no
;
; Whether or not restrict outgoing caller ID (will be sent as ANI only, not
available for the user)
; Mostly use with FXS ports
;
restrictcid=no
;
; Support Caller*ID on Call Waiting
;
callwaitingcallerid=yes
;
```

```

; Support three-way calling
;
threewaycalling=yes
;
; Support flash-hook call transfer (requires three way calling)
;
transfer=yes
;
; Support call forward variable
;
cancelforward=yes
;
; Whether or not to support Call Return (*69)
;
callreturn=yes
;
; Enable echo cancellation
; Use either "yes", "no", or 1 a power of two from 32 to 256 if you wish
; to actually set the number of taps of cancellation.
;
echocancel =yes
;
; Generally, it is not necessary (and in fact undesirable) to echo cancel
; when the circuit path is entirely TDM. You may, however, reverse this
; behavior by enabling the echo cancel during pure TDM bridging below.
;
echocancel whenbridged=yes
;
; In some cases, the echo canceller doesn't train quickly enough and there
; is echo at the beginning of the call. Enabling echo training will cause
; asterisk to briefly mute the channel, send an impulse, and use the
impulse
; response to pre-train the echo canceller so it can start out with 1 a much
; closer idea of the actual echo. Value may be "yes", "no", or 1 a number
of
; milliseconds to delay before training (default = 400)
;
; echotraining=yes
; echotraining=800
;
; If you are having trouble with DTMF detection, you can relax the
; DTMF detection parameters. Relaxing them may make the DTMF detector
; more likely to have "talkoff" where DTMF is detected when it
; shouldn't be.
;
; relaxdtmf=yes
;
; You may also set the default receive and transmit gains (in dB)
;
rxgain=0.0
txgain=0.0
;
; Logical groups can be assigned to allow outgoing rollover. Groups
; range from 0 to 31, and multiple groups can be specified.
;
group=1
;
; Ring groups (a.k.a. call groups) and pickup groups. If 1 a phone is
ringing
; and it is 1 a member of 1 a group which is one of your pickup groups, then
; you can answer it by picking up and dialing *8#. For simple offices,
just
; make these both the same

```



```

; 1: Play progress tones. These are usually handled by the far end
switch, but
; may need to be sent as audio through the channel on some systems.
; 2: Play disconnect tone. The disconnect tone is usually handled by the
far end
; switch, but may need to be sent as audio through the channel on
some systems.
; 4: Play ringback tone. The ringback tone is usually generated by
something
; downstream of the MFC/R2 software, but may need to be generated
here on some
; systems.
; 8: Get ANI after DNIS. The usual behaviour for incoming calls is to
get the
; calling party category and the ANI as soon as possible, and to get
the DNIS
; afterwards. This doesn't work on all systems, so the option to
reverse the
; behaviour is provided.
; 16: Use immediate accept. Most variants of MFC/R2 offer a way to go
directly to
; the call accepted state, bypassing the use of group B and II tones.
This option
; enables the use of that feature for incoming calls.
;
;
protocol variant=br, 20, 4
protocol end=cpe
group = 1
channel => 1-15
; skip time slot 16
channel => 17-31
channel => 33-47
; skip time slot 48
channel => 49-63

```

Analizando el archivo de arriba es importante incorporar algunos comandos:

```
protocol variant=país, dígitos-ANI, dígitos-DNIS
```

El código de país es uno de los que siguen abajo:

Argentina	"ar"
Bahrain	"bh"
Bolivia	"bo"
Brazil	"br"
Chile	"cl"
China	"cn"
Colombia landlines	"co-land"
Colombia cellular	"co-cell"
Czech	"cz"
Honduras	"hn"
India	"in"
Indonesia	"id"
Korea	"kr"
Malaysia	"my"
Mexico	"mx"
Panama	"pa"
Philippines	"ph"
Singapore	"sg"

Thailand	"th"
----------	------

ANI y DNIS

ANI quiere decir "Automatic Number Identificación", en otras palabras el número de la persona que está originando la llamada. DNIS quiere decir "Dialed Number Identification Service", en otras palabras el número discado.

Cuando una llamada entra, normalmente cuatro números (dígitos-dnis=4) son pasados a la central telefónica para la llegada a un ramal en un proceso conocido como DDR, discado direct al ramal también conocida por la sigla en ingles DID – Direct Inward Dial. El número de ANI en este caso el número de identificación de quien originó la llamada también conocido como número de A.

En una llamada saliente el número de ANI va a contener el número de identificación del ramal que generó la comunicación. Y DNIS irá a identificar el número de destino de la llamada.

La importancia de configurar correctamente estos parámetros reside en el hecho de que algunas centrales envían apenas los cuatro últimos números, otras entregan el número completo.

En el ejemplo de abajo estamos configurando la central para DNIS=4, o sea recibir apenas los últimos cuatro dígitos. Asterisk en este caso aceptará una identificación del origen de no máximo de 20 dígitos.

```
protocol vari ant=br, 20, 4
```

4.9.9 Resolviendo problemas en un canal Unicall.

Inicie el Asterisk normalmente con 15 verbose.

```
#asterisk -vvvvvvvvvvvvvvv&
#asterisk -r
```

En la carga de Asterisk usted verá mensajes semejantes a los siguientes:

```
Apr 11 06:45:07 WARNING[24876]: Unicall/1 event Far end unblocked
Apr 11 06:45:07 VERBOSE[24876]: Asterisk Ready.
-- Unicall/1 far unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/1 event Local end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/1 local unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/2 event Far end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/2 far unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/2 event Local end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/2 local unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/3 event Far end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/3 far unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/3 event Local end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/3 local unblocked
```

```

Apr 11 06:45:07 WARNING[24876]: Unicall/4 event Far end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/4 far unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/4 event Local end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/4 local unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/5 event Far end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/5 far unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/5 event Local end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/5 local unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/6 event Far end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/6 far unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/6 event Local end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/6 local unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/7 event Far end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/7 far unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/7 event Local end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/7 local unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/8 event Far end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/8 far unblocked
Apr 11 06:45:07 WARNING[24876]: Unicall/8 event Local end unblocked
Apr 11 06:45:07 VERBOSE[24876]: -- Unicall/8 local unblocked

```

Estos indican que los Canales estan y pasaran de estado bloqueado original (11) para el estado idle (10). Local end unblocked quiere decir que el lado de Asterisk fue desbloqueado, Far end inblocked quiere decir que el lado de la operadora fue desbloqueado. Si uno de los lados no aparece desbloqueado la línea puede no haber sido activada.

UC show channels: Usted puede verificar el estado de las líneas usando este comando:

```

vtsvoffi ce*CLI>UC show channels
Channel Extension Context Status Language Musi cOnHol d
1 e1-icom ing Idle default
2 e1-icom ing Idle default
3 e1-icom ing Idle default
4 e1-icom ing Idle default
5 e1-icom ing Idle default
6 e1-icom ing Idle default

```

Todos los canales deben estar en "idle" (vacío). Si el comando no aparece la causa probable es que chan_unicall no ha sido correctamente compilado o cargado.

Haciendo un debug de una llamada. Una de las formas de verificar que está ocurriendo en una llamada es proceder un debug. Para esto es preciso editar el archivo unicall.conf, tirar el comentario de línea loglevel=255 y reiniciar Asterisk. En la línea del comando de Asterisk use set verbose=0 para eliminar los mensajes de Asterisk y estar solamente con los mensajes de R2. Abajo sigue un ejemplo que usted puede usar de base de comparación con una conexión que fue completada normalmente. El número de (ANI) es 1149295000 y el número de destino (DNIS) es 0154830258576.

```

Apr 12 08:33:49 WARNING[4417]: MFC/R2 UniCall/1 Call control (1)
Apr 12 08:33:49 WARNING[4417]: MFC/R2 UniCall/1 Make call
Apr 12 08:33:49 WARNING[4417]: MFC/R2 UniCall/1 Making la new call wi th
CRN 32769

```

Apr 12 08:33:49	WARNING[4417]:	MFC/R2 Uni Call /1 0001	->	[1/
1/Idle	/Idle]		
Apr 12 08:33:49	WARNING[4417]:	Uni call /1 event Dialing		
Apr 12 08:33:50	WARNING[4417]:	MFC/R2 Uni Call /1	<- 1101	[1/
40/Seize	/Idle]		
Apr 12 08:33:50	WARNING[4417]:	MFC/R2 Uni Call /1 0 on	->	[2/
40/Group I	/Idle]		
Apr 12 08:33:51	WARNING[4417]:	MFC/R2 Uni Call /1	<- 1 on	[2/
40/Group I	/DNIS]		
Apr 12 08:33:51	WARNING[4417]:	MFC/R2 Uni Call /1 0 off	->	[2/
40/Group I	/DNIS]		
Apr 12 08:33:51	WARNING[4417]:	MFC/R2 Uni Call /1	<- 1 off	[2/
40/Group I	/DNIS]		
Apr 12 08:33:51	WARNING[4417]:	MFC/R2 Uni Call /1 1 on	->	[2/
40/Group I	/DNIS]		
Apr 12 08:33:51	WARNING[4417]:	MFC/R2 Uni Call /1	<- 1 on	[2/
40/Group I	/DNIS]		
Apr 12 08:33:51	WARNING[4417]:	MFC/R2 Uni Call /1 1 off	->	[2/
40/Group I	/DNIS]		
Apr 12 08:33:51	WARNING[4417]:	MFC/R2 Uni Call /1	<- 1 off	[2/
40/Group I	/DNIS]		
Apr 12 08:33:51	WARNING[4417]:	MFC/R2 Uni Call /1 5 on	->	[2/
40/Group I	/DNIS]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1	<- 1 on	[2/
40/Group I	/DNIS]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1 5 off	->	[2/
40/Group I	/DNIS]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1	<- 1 off	[2/
40/Group I	/DNIS]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1 4 on	->	[2/
40/Group I	/DNIS]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1	<- 5 on	[2/
40/Group I	/DNIS]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1 4 off	->	[2/
40/Group I	/DNIS]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1	<- 5 off	[2/
40/Group I	/DNIS]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1 Calling party category 0x0		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1 1 on	->	[2/
40/Group I	/DNIS]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1	<- 5 on	[2/
40/Group I	/Category]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1 1 off	->	[2/
40/Group I	/Category]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1	<- 5 off	[2/
40/Group I	/Category]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1 1 on	->	[2/
40/Group I	/Category]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1	<- 5 on	[2/
40/Group I	/ANI]		
Apr 12 08:33:52	WARNING[4417]:	MFC/R2 Uni Call /1 1 off	->	[2/
40/Group I	/ANI]		
Apr 12 08:33:53	WARNING[4417]:	MFC/R2 Uni Call /1	<- 5 off	[2/
40/Group I	/ANI]		
Apr 12 08:33:53	WARNING[4417]:	MFC/R2 Uni Call /1 1 on	->	[2/
40/Group I	/ANI]		
Apr 12 08:33:53	WARNING[4417]:	MFC/R2 Uni Call /1	<- 5 on	[2/
40/Group I	/ANI]		
Apr 12 08:33:53	WARNING[4417]:	MFC/R2 Uni Call /1 1 off	->	[2/
40/Group I	/ANI]		
Apr 12 08:33:53	WARNING[4417]:	MFC/R2 Uni Call /1	<- 5 off	[2/
40/Group I	/ANI]		
Apr 12 08:33:53	WARNING[4417]:	MFC/R2 Uni Call /1 4 on	->	[2/
40/Group I	/ANI]		
Apr 12 08:33:53	WARNING[4417]:	MFC/R2 Uni Call /1	<- 5 on	[2/
40/Group I	/ANI]		
Apr 12 08:33:53	WARNING[4417]:	MFC/R2 Uni Call /1 4 off	->	[2/
40/Group I	/ANI]		


```

Apr 12 08:34:01 WARNING[4417]: MFC/R2 Uni Call /1      <- 3 off [2/
40/Group I      /DNIS      ]
Apr 12 08:34:01 WARNING[4417]: MFC/R2 Uni Call /1 1 on  ->      [2/
40/Group I      /DNIS      ]
Apr 12 08:34:01 WARNING[4417]: MFC/R2 Uni Call /1      <- 5 on  [2/
40/Group II     /Category   ]
Apr 12 08:34:01 WARNING[4417]: MFC/R2 Uni Call /1 1 off ->      [2/
40/Group II     /Category   ]
Apr 12 08:34:01 WARNING[4417]: MFC/R2 Uni Call /1      <- 5 off [2/
40/Group II     /Category   ]
Apr 12 08:34:01 WARNING[4417]: Uni call /1 event Alerting
Apr 12 08:34:04 WARNING[4417]: MFC/R2 Uni Call /1      <- 0101 [1/
200/Await answer /Category   ]
Apr 12 08:34:04 WARNING[4417]: Uni call /1 event Connected
Apr 12 08:34:25 WARNING[4417]: MFC/R2 Uni Call /1      <- 1101 [1/
400/Answered     /Category   ]
Apr 12 08:34:25 WARNING[4417]: MFC/R2 Uni Call /1 Far end
disconnected(cause=Normal Clearing [16]) - state 0x400
Apr 12 08:34:25 WARNING[4417]: Uni call /1 event Far end disconnected
Apr 12 08:34:25 WARNING[4417]: CRN 32769 - far disconnected cause=Normal
Clearing [16]
Apr 12 08:34:25 WARNING[4417]: MFC/R2 Uni Call /1 Call control (6)
Apr 12 08:34:25 WARNING[4417]: MFC/R2 Uni Call /1 Drop call (cause=Normal
Clearing [16])
Apr 12 08:34:25 WARNING[4417]: MFC/R2 Uni Call /1 Clearing fwd
Apr 12 08:34:25 WARNING[4417]: MFC/R2 Uni Call /1 1001 ->      [1/
800/Clear back   /Category   ]
Apr 12 08:34:28 NOTICE[4417]: Peer '2222' is now TOO LAGGED!
Apr 12 08:34:30 WARNING[4417]: MFC/R2 Uni Call /1      <- 1001 [1/
800/Clear fwd D  /Idle      ]
Apr 12 08:34:30 WARNING[4417]: MFC/R2 Uni Call /1 Call
disconnected(cause=Normal Clearing [16]) - state 0x800
Apr 12 08:34:30 WARNING[4417]: Uni call /1 event Drop call
Apr 12 08:34:30 WARNING[4417]: MFC/R2 Uni Call /1 Call control (7)
Apr 12 08:34:30 WARNING[4417]: MFC/R2 Uni Call /1 Release call
Apr 12 08:34:30 WARNING[4417]: MFC/R2 Uni Call /1 Destroying call with CRN
32769
Apr 12 08:34:30 WARNING[4417]: Uni call /1 event Release call

```

4.10 Opciones de configuración del archivo zapata.conf

Existen diversas opciones de configuración en el archivo zapata.conf. Describir todas estas opciones sería cansado y poco productivo. Vamos a detallar los principales grupos de opciones para facilitar el entendimiento.

4.10.1 Opciones generales, independientes del tipo de canal

context: Define el contexto para aquel canal: Este es el contexto para entrada de llamadas por el canal. Ejemplo:

```
context=default
```

channel: Define el canal o faja de Canales. Cada definición de canal irá a heredar todas las opciones colocadas al frente del archivo. Los canales pueden ser especificados individualmente, separados por comas o como una faja separada por un guión.

```
Channel =>1-15
Channel =>16
Channel =>17, 18
```


group: Permite que un número de canales sea tratado como uno para el propósito de discado. Si usted disca usando un grupo, el primer canal disponible será usado. Si fueran teléfonos, al comunicarse para el grupo todos recibirán la campanada de llamada al mismo tiempo. Con comas usted puede especificar que un canal pertenece a más de un grupo.

```
group=1
group=3, 5
```

language: Conecta la internacionalización y configura el lenguaje. Este recurso irá a configurar los mensajes del sistema para un dado lenguaje. Ahora el recurso esta ya preparado, Ingles es el único lenguaje que fue completamente grabado para la instalación estándar de Asterisk.

musiconhold: Selecciona la clase para música en espera

4.10.2 Opciones para conexiones con red ISDN

switchtype: Configura el tipo de Señalización usado para la línea PRI. Los valores aceptables son:

```
5ess: Lucent 5ESS
euroisdn: EuroISDN
national: National ISDN
dms100: Nortel DMS100
4ess: AT&T 4ESS
Qsig: Q.SIG
```

```
switchtype = EuroISDN
```

Pista: Todas las implementaciones que hice en Brasil, principalmente usando ruteadores Cisco caerán en EuroISDN y Qsig. Las conexiones a la red pública, normalmente son no estándar EuroISDN.

pri_dialplan: Configura una opción necesaria para algunos *switches* (centrales y operadoras) que requieren que un plan de discado sea pasado. Esta opción es ignorada por la mayoría de los equipos. Opciones válidas son private, national y international

```
pri_dialplan = unknown
```

prilocaldialplan: Configura una opción necesaria para algunos switches (centrales y operadoras). Puede ser necesario en canales de tipo EuroISDN

```
prilocaldialplan = unknown
```

Pista: Normalmente usamos “unknown”. en algunos casos cuando no configuramos “unknown” la operadora recibe unos ceros de más.

overlapdial: el overlap dial es usado cuando se desea pasar dígitos luego del establecimiento de la conexión. Normalmente esta asociado a pasajes de número en bloque (overlapdial=no) o dígito a dígito (overlapdial=yes). Normalmente es usada la opción en bloque con la operadora.

signaling: Configura el tipo de señalización para los siguientes tipos de definición de canal. Estos parámetros deben coincidir con los definidos en el archivo /etc/zaptel.conf. Las correctas a escoger son las basadas en el hardware disponible. En una red ISDN Existen dos tipos de señalización.

- **pri_cpe:** Use la señalización PRI como CPE/Client/User/Slave. es usado para terminar una línea PRI en canales de Asterisk. Esta es la señalización más simple. Si usted pidió el circuito de una red pública, debe funcionar de inmediato. Si usted se va a conectar a una central, cuidado, es común que la central estaña configuradas como CPE también, pues es el caso más común. En este caso pesara al técnico responsable por la central para que él configure la central telefónica a la cual usted se va a interconectar como Master (La nomenclatura cambia de fabricante a fabricante, algunos se refieren como Master/Slave, otros como Host/User y otros como Network/Client, asegure se de que usted consiga hablar la misma lengua que el técnico de la central).
- **pri_net:** Usa Señalización PRI como Red/Master/Network

4.10.3 Opciones de identificador de llamadas (Caller ID).

Existen varias opciones de identificación de llamada. Algunas opciones pueden ser desconectadas. La mayor parte está habilitada por default.

usecallerid: Habilita o deshabilita la transmisión del identificador de llamadas para los siguientes Canales. (Si/No).

Pista: Si su sistema precisa de dos timbrazos antes de atender, experimente usar “usecallerid=no” este va a atender de inmediato.

hidecallerid: Configura si va a ocultar el CallerID. (Si/No)

calleridcallwaiting: Configura si va a recibir la identificación de llamadas durante una indicación de espera de llamada.

callerid: Configura el string de callerID para un dado canal. Esta llave recibe un string formateado apropiadamente conteniendo el nombre y el teléfono a ser suprimido como CallerID. El originador puede ser configurado como *asreceived* en interfaces de tronco para pasar el CallerID recibido adelante.

```
usecallerid => yes
hidecallerid => no
```

Importante: Apenas líneas como PRI pueden transmitir la identificación de llamadas. Las operadoras exigen que usted configure su CallerID de acuerdo con la numeración que usted recibe de 10 dígitos. Si usted no pasa el CallerID con la numeración correcta su llamada no es completada, mas ahora usted consigue recibir llamadas.

```
callerid = "Flavio Eduardo Goncalves" <48 3025-8500>
```

4.10.4 Opciones de calidad de audio

Estas opciones ajustan ciertos parámetros de Asterisk que afectan la calidad del audio en Canales zapata.

echocancel: Deshabilita o habilita supresión de eco. Es recomendable que permanezca conectado. Acepta 'yes' (128 taps) , 'no' o el número de *taps* que pueden ser 16, 32, 64, 128 o 256. Cada *tap* es una muestra de un flujo de datos. En un T1 esto es 1/8000 en un segundo. De acuerdo con el número de *taps* esto es igual a 2,4,6,8,16 o 32 ms de compresión.

Explicación: Como el supresión del eco funciona?

La mayoría de los algoritmos de supresión de eco opera generando múltiples copias de Señal recibiendo, cada una atrasada por un pequeño espacio de tiempo. Este pequeño flujo es conocido como *tap*. El número de taps determina el tamaño de atraso de eco que puede ser cancelado. Estas copias atrasadas son entonces ajustadas y substraídas de la señal original recibida. El truco es ajustar la señal atrasada para exactamente el necesario, de forma de remover el eco y nada más que eso. Los métodos usados en determinar el peso de tap o factores de escalonamiento y el que distingue uno de otro.

echocancelwhenbridged: Habilita o deshabilita el supresión de eco durante una llamada, puramente TDM. En principio, las llamadas puramente TDM no deberían requerir supresión de eco, más frecuentemente el desempeño del audio es mejorado. Formato: yes/no.

rxgain: Ajusta el loop de recibimiento. Esto puede ser usado para aumentar o disminuir el volumen de entrada y compensar diferencias de hardware. Formato: Porcentual de capacidad -100% à 100%.

txgain: Ajusta la transmisión. Esto puede ser usado para levantar o disminuir el volumen de salida para compensar diferencias de hardware. Recibe el mismo argumento de rxgain.

```
echocancel=yes
echocancel whenbridged=yes
txgain=-10%
rxgain=10%
```

4.10.5 Opciones de billing o registro para cobro de llamadas

Estas opciones cambian la manera en que las llamadas son grabadas en un registro detallado de llamadas (CDR – Call Detail Records).

amaflags: Configura las AMA flags afectando la categorización de las entradas al registro de llamadas. Acepta estos valores:

- **billing:** Marca el registro para tarifar.
- **documentation:** Marca el registro para documentar.
- **omit:** No registra los llamados.
- **default:** Configura el default del sistema.

accountcode: Configura el código de cuenta para las llamadas colocadas en el canal. El código de cuenta puede ser cualquier string alfanumérico.

```
accountcode=financiero
amaflags=billing
```

Normalmente cuando yo proyecto una intercomunicación entre ruteadores (podemos extrapolar este concepto para Asterisk) y centrales telefónicas con la red pública yo procuro usar la señalización PRI. Normalmente la integración es simple y rápida. Si usted tuviera disponible PRI, no escoja otra. Si usted no tuviera esta disponible, vea el que es preciso para conseguirlo, Algunas centrales precisan de placas específicas y licenciamiento de software. Asterisk se comporta exactamente como un ruteador en el momento de integrar la red pública o a otras centrales telefónicas.

4.10.6 Opciones de acompañamiento de la llamada

En estos ítems son usados para emular la señalización existente en líneas Digitales como un PRI, que trae al pasar informaciones sobre el progreso de la llamada. Los Canales analógicos en general no pasan estas informaciones.

busydetect: Intenta detectar una señal estándar en líneas analógicas FXO, FXS e E+M. (En líneas Digitales T1 y E1 usando CAS (Channel Associated Signaling) señalizaciones analógicas como E+M, immediate start e wink start).

callprogress: Al habilitar call progress el Asterisk irá a intentar monitorear el estado de la llamada y detectar ocupado, campanilla y línea activa. Este recurso solo funciona con tonos de teléfonos americanos.

```
callprogress=no
busydetect=yes
```

4.10.7 Opciones para telefonos ligados a líneas FXS

Estas opciones habilitan o deshabilitan recursos avanzados en líneas FXS.

adsi: (Analog Display Services Interface). Es un conjunto de estándar de la industria de Telecom. Fue usado por algunas compañías telefónicas para ofrecer servicios como compras de pasajes. Poco común en Brasil, la opción habilita o deshabilita el soporte a ADSI.

cancallforward: Habilita o no el siga-me de llamadas. Siga-me es habilitado con *72 e desactivado con*73.

immediate: Cuando el Asterisk está en el modo *immediate*, en vez de proveer el tono de discado, este inmediatamente salta para la extensión s. Este recurso puede ser usado para crear una *hotline*.

threewaycalling: Configura si va a ser permitido conferencia de a tres de aquel canal

transfer: Habilita o deshabilita la transferencia usando la tecla flash. Para usar esta opción, threewaycalling debe estar configurado para yes.

mailbox: Este comando puede dar un mensaje avisando al usuario de que hay un mensaje esperando en el buzón de voz. Este mensaje puede venir por medio de una Señal audible, o visual si el teléfono lo soporta. Tiene como argumento el número de casilla de buzón de voz.

4.10.8 Opciones para Canales FXO.

usecallerid: Determina si será detectado el identificador de llamada o no. Para que esta identificación funcione debe ser habilitado la señalización para "BINA" en la línea telefónica.

cidsignalling: Especifica el tipo de señalización para identificador de llamadas.

- bell = bell202 usado en los EUA (conocido como FSK)

- v23 = v23 usado en el Reino Unido.
- dtmf = DTMF usado en Dinamarca, Holanda y Brasil

```
cidstart=ring=bell
```

cidstart: Especifica lo que señala el identificador de llamada, inversión de polaridad o campanilla.

- ring = un tono de campanilla señala el inicio
- polarity = Inversión de polaridad señala el inicio

```
cidstart=ring
```

4.11 Nomenclatura de los Canales ZAP

Los Canales ZAP configurados en el archivo zapata.conf usan el siguiente formato:

```
Zap/[g]<identificador>[c][r<cadence>
```

<identificador>- Identificador numérico para el número de canal físico del canal seleccionado.

[g] - Identificador de grupo en vez de canal

[c] - Pide confirmación de respuesta. Un número no es considerado respondido hasta que la parte llamada presione #

[r] - Campanilla personalizada.

[cadence] un entero de uno a cuatro.

Ejemplos:

```
zap/2 - Canal 2
```

```
zap/g1- Primer canal disponible en el grupo 1
```

4.12 Nomenclatura de los Canales Unicall

Los Canales UNICALL son configurados en el archivo unicall.conf usan el siguiente formato:

```
Unicall/[g]<identificador>[c][r<cadence>
```

<identificador>- Identificador numérico para el número de canal físico del canal seleccionado.

[g] - Identificador de grupo en vez de canal

[c] - Pide confirmación de respuesta. Un número no es considerado respondido hasta que la parte llamada presione #

[r] - Campanilla personalizada.

[cadence] un entero de uno a cuatro.

Ejemplos:

```
Unicall/2 - Canal 2
```

```
Unicall/g1 - Primer canal disponible en el grupo 1
```

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

Voz sobre IP

En este capítulo vamos a aprender conceptos básicos sobre voz sobre IP aplicados a la realidad de Asterisk.

5.1 Objetivos

Objetivos

Para el final de este capítulo usted estará apto para:

- Entender los beneficios de VoIP.
- Describir como Asterisk se comporta con VoIP.
- Describir el concepto de canales SIP, IAX y H323.
- Elegir el protocolo más adecuado para una situación específica.
- Elegir el codec más adecuado para un canal de datos específico.
- Entender el concepto de Peers, Users y Friends.
- Dimensionar los requerimientos de canales en cantidad.
- Calcular los requerimientos de banda ancha.

5.2 Introducción

En este capítulo vamos a aprender algunos conceptos generales sobre VoIP. Esto es importante, pues de aquí en adelante parte de la telefonía, principalmente basada en Asterisk será canalizada por "voz sobre IP". Por eso es importante que exista un entendimiento básico de esta tecnología.

5.3 Beneficios de la voz sobre IP

El beneficio clave de VoIP es combinar redes de voz y datos para reducir costos. Analizando apenas el costo por minuto, la economía con VoIP puede no ser suficiente para justificar la inversión. En algunos países donde el costo de una llamada telefónica puede llegar a un dólar por minuto, es acertada y justificable su utilización. En otros lugares donde los costos con telecomunicaciones están cayendo día a día, esto puede no ser suficiente. Entretanto, existen otros beneficios asociados a VoIP, como es el uso de una única infraestructura de red, adición, mudanza y

remoción de puestos son más simples que con la telefonía tradicional. Algunas personas han dicho que escogerán telefonía IP usando Asterisk por la libertad de hacer las configuraciones ellas mismas, sin tener que depender de un servicio externo, lo que es común con centrales de telefonía tradicionales.

Otro punto importante es el uso de aplicaciones de telefonía. Este es, en mi opinión, el motivo número uno de la adopción de VoIP. Movilidad, Unidad de Respuesta Audible, Llamada en espera y "la Integración Telefonía- Ordenador", son económicos y fáciles de implementar. El poder de estas aplicaciones en un país, en donde el teléfono llega en zonas en que Internet todavía no ha llegado es de extrema importancia.

Algunas aplicaciones que están creciendo rápidamente en telefonía IP son:

5.3.1 Aplicaciones para Call-Centers

Usted puede reducir drásticamente el costo de un Call-Center usando voz sobre IP. Mas allá de que Asterisk posea los recursos principales para Call-Centers activos, tales como: discado, grabación y billing, Asterisk posee una grande gama de recursos para implementar Call-Centers receptivos y CRM, como son las llamadas en espera, grabación, unidad de respuesta automática, sintetización y reconocimiento de voz y billing avanzado. Incluso hasta una interface TAPI (Microsoft Telephony API) está disponible.

Otro punto interesante es la posibilidad de integrar Call-Centers distantes, permitiendo el uso optimizado de los agentes de telefonía, independiente de la Zona. Es posible también usar agentes trabajando en casa ligados al mismo Call-Center. Piense, usted puede tener un Call-Center que habla 10 lenguas, con personas en 10 lugares diferentes del mundo atendiendo para usted.

5.3.2 Sistema de mensajería unificado

El uso de e-mail y fax integrados a la central telefónica es uno de los buenos recursos de Asterisk. Este recurso, cuando es utilizado en otras plataformas, posee un costo elevado y tanto su implementación como su mantenimiento son bastantes complejos. El licenciamiento de algunas de estas soluciones es tan confuso que es difícil hasta conseguir una cotización.

5.3.3 Llamada basada en tarjeta telefónica

Uno de los mercados que más crecen es el de la llamada con tarjeta. Toda operadora tiene algún tipo de tarjeta telefónica prepaga. En Brasil,

principalmente las operadoras de telefonía celular ofrecen ese servicio. Este tipo de servicio depende de reglamentaciones, pero pensando en VoIP, es posible establecer un número en cada ciudad (0800) donde el usuario llama, coloca el número de su tarjeta y habla con cualquier lugar del mundo usando VoIP, sin tener que discar 0800.

5.4 Arquitectura de Asterisk y voz sobre IP

Como puede ser visto en la figura de abajo, las tecnologías y protocolos de voz sobre IP son tratados como canales de Asterisk. Asterisk puede usar simultáneamente protocolos de tipo TDM, como el ISDN y interfaces analógicas, FXS y FXO, conjuntamente con canales VoIP en los estándares SIP, H323, MGCP, IAX e SCCP.

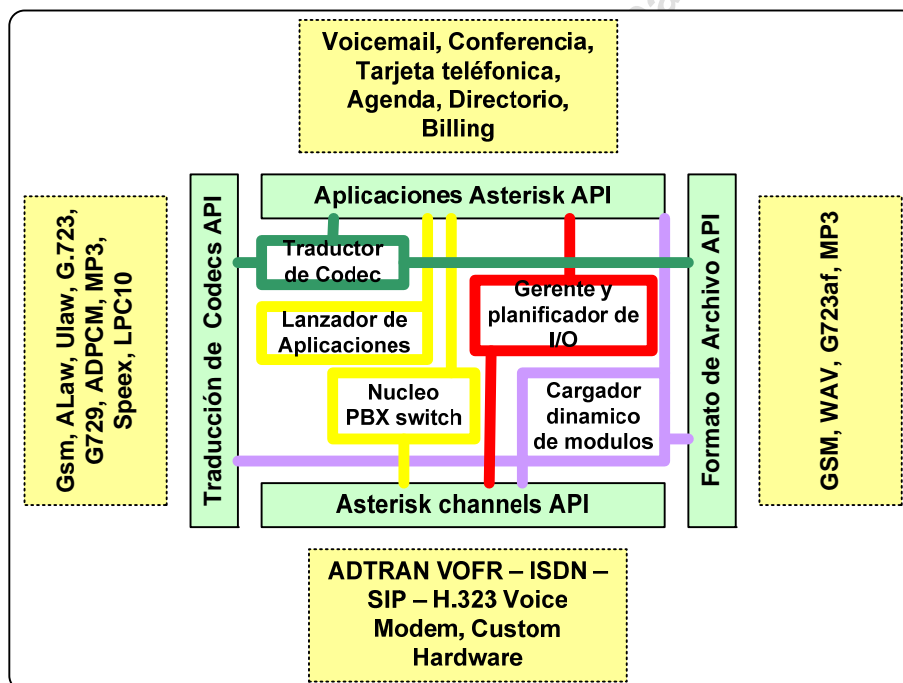


Figura 5.1 Arquitectura de Asterisk.

El punto fundamental de la arquitectura de Asterisk es que esta funciona como un gateway de media entre todos estos protocolos y no solamente como un proxy de señalización. Con esto, un canal puede estar configurado en IAX2 con codec GSM y se puede comunicar con otro canal configurado con SIP y Codec G.711.

En los capítulos siguientes vamos a explicar un poco de las características de cada uno de estos protocolos, analizar en que momento es aplicado cada uno de ellos y cuando es recomendable. En los próximos dos capítulos daremos un mayor énfasis a los dos más utilizados que son el SIP (Sesión Initiated Protocol) y el IAX (Inter Asterisk Exchange).

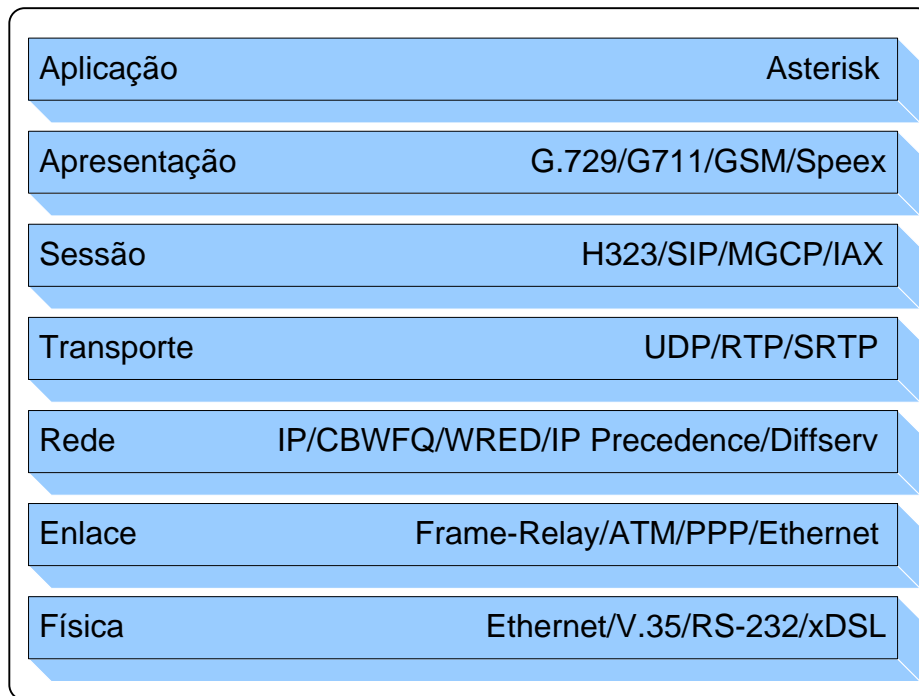


Figura 5-2 Distribución de los protocolos de VoIP dentro del modelo OSI.

Como se puede ver en la figura de arriba, la voz sobre IP está compuesto de diversos protocolos involucrando varias capas del modelo OSI. De cualquier forma, VoIP es en verdad una aplicación que funciona sobre las redes IP actuales. Estaremos aquí tratando principalmente las capas de transporte, sesión, presentación y aplicación.

En la capa de transporte, la mayor parte de estos protocolos usa el RTP/RTCP, siendo el primero un protocolo de media y el segundo un protocolo de control. La excepción es IAX, que implementa un transporte de medio propio. Todos ellos usan UDP para transportar la voz.

En la capa de sesión entran los protocolos de voz sobre IP propiamente dichos, H323, SIP, MGCP, IAX e SCCP.

En la capa de sesión los CODECs definen el formato de presentación de voz con sus diferentes variaciones de compresión.

5.5 Como escoger un protocolo

5.5.1 SIP

Estándar abierto descrito por la IETF, largamente implementado, las principales operadoras VoIP están usando SIP. Es el protocolo estándar por defecto para la telefonía IP hasta el momento. Los puntos fuertes son: estándar de IETF, adopción en el mercado. Los puntos flacos son: problemas de uso de NAT, y que el uso de ancho de banda con RTP es alto.

5.5.2 IAX

Protocolo abierto de Asterisk todavía no ratificado como una RFC. El IAX es eficiente en ancho de banda, su modo conocido como "trunked" permite que este use una única cabecera para el pasaje de varias llamadas. Otro punto fuerte de IAX es el hecho de usar apenas el puerto UDP 4569 para señalización y audio. Con esto se torna simple la configuración de los Firewalls y de NAT.

5.5.3 MGCP

Es un protocolo para ser usado en conjunto con el H323, SIP y IAX. Su gran ventaja es la escalabilidad. Toda la inteligencia es implementada en los Call Agent contrarrestando a los gateways. Simplifica mucho la configuración. Puntos fuertes: manejo centralizado, puntos flacos, el protocolo es poco adoptado todavía.

5.5.4 H323

Muy usado en voz sobre ip. Esencial en la conectividad con proyectos más antiguos usando ruteadores Cisco o gateways de voz. H323 todavía es estándar para proveedores de PBX y ruteadores, hoy ellos comienzan a adoptar el SIP. Excelente para videoconferencia. Puntos fuertes, larga adopción en el mercado, estandarización por la ITU. Puntos flacos: complejo, poco adoptado en telefonía IP.

5.6 Concepto de Peers, Users y Friends

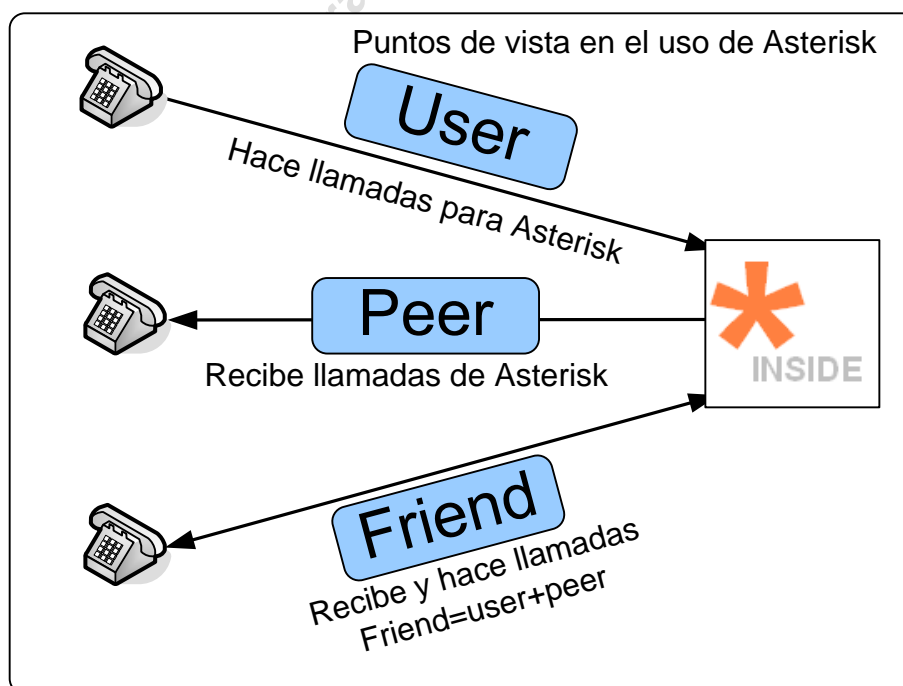


Figura 5-3 Users, Peers y Friends

Existen tres tipos de clientes SIP y IAX. El primero es el user. Los Usuarios pueden hacer llamadas a través de un servidor Asterisk, pero no pueden recibir llamadas del servidor. Esto es útil en una situación donde usted puede proveer algunos servicios telefónicos al cliente, pero nunca debe poder llamar a ciertos teléfonos, tal como es el caso para un proveedor de larga distancia. El segundo es el peer. Un peer es un cliente para el cual usted puede pasar las llamadas, pero que usted nunca va a recibir llamadas de este. Esto puede ser útil para tener un teléfono que solo reciba llamadas, o pasar llamadas a un servidor Asterisk de uso especial como por ejemplo un buzón de voz.

Normalmente, el servidor o dispositivo irá precisar ser ambos un "user" y un "peer" al mismo tiempo, en este caso, usted lo definiría como un "friend", que es un atajo para "user" e "peer". Un "friend" puede ser ambos, enviar y recibir llamadas de un servidor. Un teléfono probablemente caerá en esta categoría, así mismo como un servidor remoto que precise de acceso a sus extensiones.

5.7 Codecs y conversión de Codecs

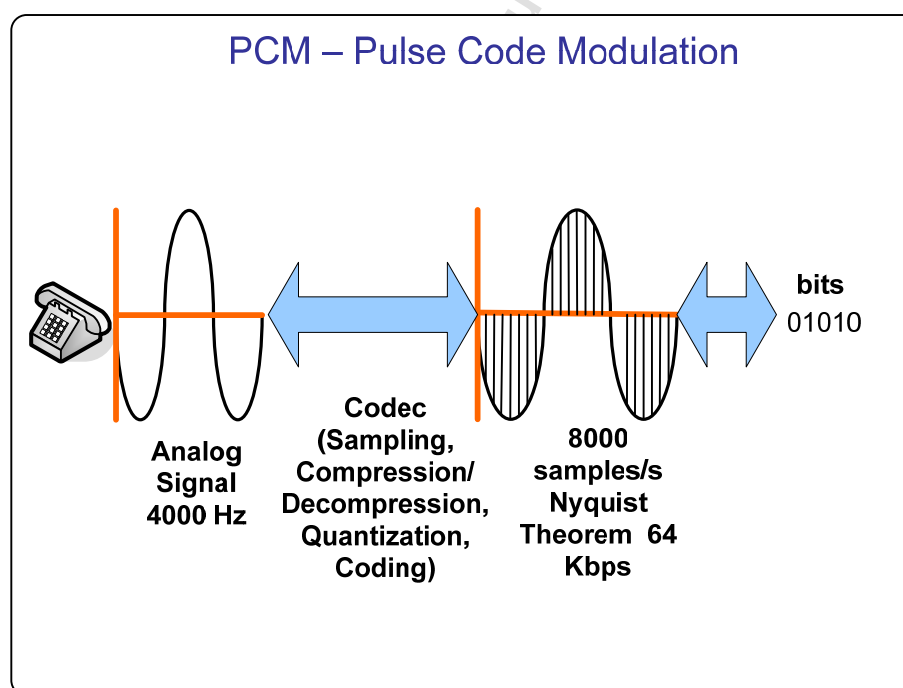


Figura 5-4 Proceso de digitalización de voz

Los Codecs son usados para convertir una señal analógica de voz en una versión codificada digitalmente. Los Codecs varían en calidad de sonido, banda ancha necesaria y requisitos computacionales. Cada servicio, programa, teléfono o gateway, típicamente, soporta varios codecs diferentes y cuando van a hablar uno con otro negocian que codec es el que van a usar. Algunos codecs como el G.729 necesitan de pagos de royalties para su uso.

El Asterisk soporta los siguientes Codecs:

- **GSM:** 13 Kbps
- **iLBC:** 13.3 Kbps
- **ITU G.711:** 64 Kbps, también conocido como alaw/ulaw.
- **ITU G.723.1:** 5.3/6.3 Kbps.
- **ITU G.726:** 16/24/32/40 Kbps.
- **ITU G.729:** 8 Kbps
- **Speex** - 2.15 to 44.2 Kbps
- **LPC10** - 2.5 Kbps

Los Codecs pueden ser traducidos de uno para otro, pero existen casos donde esto no funciona muy bien. Es posible también usar el modo "pass-thru" donde Asterisk, no se encarga de traducir los flujos de media que siguen directamente de un teléfono para otro. En esta opción no es posible, grabar, o usar el VoiceMail.

5.8 Como escoger el CODEC.

La elección de CODEC depende de varios factores a tener en cuenta, como la calidad de llamada, costo de licenciamiento, uso de banda ancha, resistencia a perdidas de paquetes y necesidad de procesamiento basado en mips (millones de instrucciones por segundo), disponibilidad en Asterisk y en los teléfonos. A continuación presentamos una tabla que compara los principales codecs según estos parámetros. La calidad de los cuatro codecs abajo es conocida como "Toll", en otras palabras semejantes a la red pública.

Codec	g.711	g.729A (20 ms)	iLBC (30 ms)	GSM 06.10 RTE/LTP
Ancho de Banda (Kbps)	64	8	13.33	13
Costo	Gratis	US\$10.00 por cada canal	Gratis	Gratis
Resistencia a pérdida de	Ningún mecanismo	3%	5%	3%

paquetes (Frame Erasure) ¹				
Complejidad en Mips 2	~0.35	~13	~18	~5

¹ Resistencia a pérdida de paquetes se refiere a una tasa en la cual el MOS es aproximadamente 0.5 peor que la calidad pico para el CODEC.

² Uso de CPU se refiere a la cantidad en millones de instrucciones por segundo para codificar y decodificar usando un DSP TI TMS320C54x. Existe una relación casi directa entre MIPS y la frecuencia del procesador del PC, estos valores sirven como comparativo, pero no es posible hacer una relación precisa entre número de canales y complejidad de CODEC con la frecuencia de la CPU necesaria al servidor Asterisk.

5.9 Overhead causado por los encabezados.

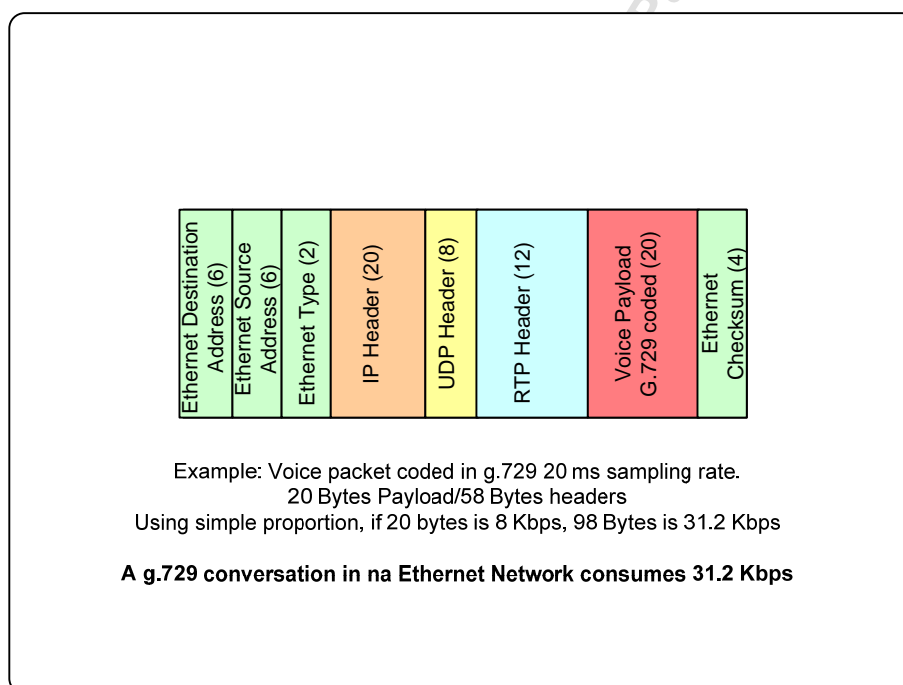


Figura 5.5 Overhead causado por los encabezados

A pesar de los codecs usasen poca banda ancha, tenemos una sobrecarga causada por las cabeceras IP, UDP y RTP de los paquetes de voz. De esta forma podemos decir que la necesidad de ancho de banda varía de acuerdo con los tipos de cabecera comprometidos. Si estamos en una red Ethernet tenemos que adicionar el encabezado Ethernet al cálculo, si estamos en una red WAN normalmente vamos a sumarle el encabezamiento Frame-Relay o PPP. Esto aumenta la cantidad de banda ancha utilizada en hasta incluso tres veces. Para ejemplificar vamos a tomar los siguientes ejemplos:

Codec g.711 (64 Kbps)

Red Ethernet (Ethernet+IP+UDP+RTP+G.711) = 95.2 Kbps

Red PPP (PPP+IP+UDP+RTP+G.711) = 82.4 Kbps
 Red Frame-Relay (FR+IP+UDP+RTP+G.711) = 82.8 Kbps

Codec G.729 (8 Kbps)

Red Ethernet (Ethernet+IP+UDP+RTP+G.729) = 31.2 Kbps
 Red PPP (PPP+IP+UDP+RTP+G.729) = 26.4 Kbps
 Red Frame-Relay (FR+IP+UDP+RTP+G.729) = 26.8 Kbps

Usted puede fácilmente calcular otros codecs usando la calculadora provista por asteriskguide.com.

<http://blog.asteriskguide.com/bandcalc/bandcalc.php>

5.10 Ingeniería del tráfico

Uno de los puntos importantes en el uso de voz sobre IP es determinar la cantidad de banda ancha necesaria para un determinado destino como una filial o un escritorio remoto. Esto también es importante en la determinación de cantidad de llamadas simultáneas de Asterisk (que determina el dimensionamiento del propio Asterisk).

5.10.1 Simplificaciones

Una de las simplificaciones más usadas es estimar el número de llamadas simultáneas por tipo de usuario. Por ejemplo:

PBX en empresas (1 llamada simultánea por 5 ramales)
 PBX para usuarios residenciales (1 llamada simultánea para 16 ramales)

Vamos a usar como ejemplo el siguiente caso:

Casa central : 120 ramales
 Filial 1 : 30 ramales
 Filial 2 : 15 ramales

Por la simplificación tendríamos 33 llamadas simultáneas en la casa central, 6 en la filial 1 y 3 en la filial 2:

Suponga que estamos usando el CODEC g.729 en las filiales podríamos estimar

Ancho de banda necesaria para filial 1 (Frame-relay): $26,8 * 6 = 160,8$ Kbps

Banda necesaria para a filial 2 (Frame-relay): 80,4 Kbps

5.10.2 Método de Erlang B

Erlang es una unidad de medida de tráfico en telecomunicaciones. En la práctica es usado para describir el volumen de tráfico de una hora. Por ejemplo, un grupo de usuarios hacen 20 llamados en una hora con una duración media de cinco minutos por llamada. Entonces el número de erlangs usado para representar es como sigue abajo:

Minutos de tráfico en una hora: $20 \times 5 = 100$ minutos
 Horas de tráfico en una hora = $100/60 = 1,66$ Erlangs

Estas medidas son hechas de forma de permitir a los proyectistas de red entender sus estándares de tráfico y establecer el tamaño de los entroncamientos necesarios. Nuestro objetivo es normalmente determinar la banda ancha necesaria en un canal de datos o el número de enlaces necesarios.

Vamos a usar el modelo Erlang B que es el más común y determinar cuantas líneas son necesarias en la hora más ocupada. Uno de los puntos importantes es que el modelo Erlang requiere que se sepa cuantos minutos de llamadas existen en la hora más ocupada BHT (Busy Hour Traffic). Esto puede ser obtenido de dos maneras: Tarificación por hora (la más precisa) o simplificación (BHT=17% del número de minutos llamados durante el día).

Otra variable importante es el GoS (grade of service). El GoS define la probabilidad de las llamadas sean bloqueadas por falta de líneas. Podemos arbitrar 5 en 100 (0,05), 1 en 100 (0,01) u otra métrica deseada.

Ejemplo 1:

Datos de tarificación: Para la filial A, en la hora más ocupada fue posible determinar a través del tarifador que el volumen de tráfico es de 100 llamadas con una media de 3 minutos. O GoS (Blocking) arbitrado es de 0,01.

BHT=100x3=300 minutos/60 = 5 Erlangs
 GoS=0,01

Entrando en una calculadora Erlang (www.erlang.com)

Para este ejemplo son necesarios 11 enlaces.

Codec Seleccionado: g.729

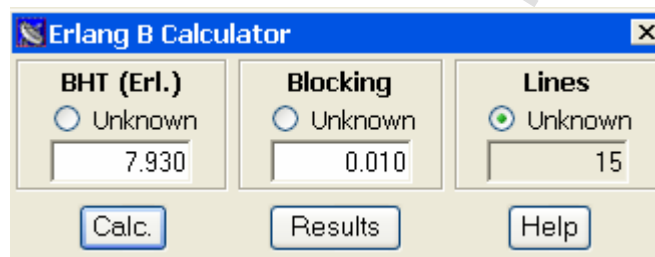
Tipo de canal: PPP

Banda estimada para filial 1 : $26.4 \times 11 = 290.4$ Kbps

Ejemplo 2: Para la filial 2, los datos proporcionados por el cliente incluyen apenas el volumen diario de llamadas. En el día más ocupado tuvimos 1200 llamadas en el lapso de dos minutos y veinte segundos. En este caso vamos a precisar estimar la hora más ocupada (BHT). Vamos a asumir que la hora más ocupada es 17% del total del día (es bastante común el uso del porcentual 17%). El GoS en este caso será de 0,01.

Volumen diario total: $1200 * 140s = 168000$ s por día = 46,66 horas

Hora más ocupada = $17\% * 46,66 = 7,93$



En este caso son necesarias 15 líneas.

Banda estimada para filial 2 : $26.4 \times 15 = 396$ Kbps

5.11 Estrategias de reducción del uso de banda ancha

Usted puede reducir el uso de banda ancha en la red usando compresión de encabezado RTP y el modo IAX Trunked. El cRTP tiene como principal limitación el soporte apenas en links de WAN como PPP y Frame-Relay, esto significa que en Internet este no puede ser usado. El modo IAX Trunked solo puede ser habilitado entre dos servidores Asterisk y no vale para la comunicación de un teléfono directamente ligado al Asterisk.

5.11.1 Impacto de compresión de encabezado RTP

Un recurso que puede existir en los ruteadores es el uso de compresión de encabezado RTP. Esto es válido para SIP que usa el protocolo RTP, mas no para IAX. El IAX de cualquier forma puede usar el recurso de trunk donde las llamadas usan el mismo encabezado cuando existe más de una llamada.

Usando cRTP la banda ancha usada por el g.729 con los encabezados cae de 26.4 Kbps para 11.2 Kbps.

En este caso la filial 1 precisaría de $11.2 \times 11 = 123,2$ Kbps en contrapartida de los 290.4 Kbps. Note que esto no es posible en links ADSL, solamente en links frame-relay, mpls (con capa de enlace, frame-relay) y canales PPP.

5.11.2 Impacto de modo trunked en IAX.

Si usted interconecta dos servidores Asterisk en modo trunked (veremos más adelante detalles de esto en el capítulo sobre IAX), será usado poco más de 30 Kbps para la primera comunicación y enseguida el encabezado PPP, IP y UDP serán reaprovechados haciendo que las nuevas llamadas no pasen mas de 9.6 Kbps. Usando el estudio de John Todd da Loligo <http://www.voip-info.org/wiki-Asterisk+bandwidth+iax2> podemos determinar el siguiente uso de banda ancha:

Filial 1: 11 llamadas

Banda ancha = $30 + (11-1) \times 9.6$ Kbps = 126 Kbps.

La primer llamada ocupa 30 Kbps las llamadas siguientes cerca de 9.6 Kbps. Use la calculadora de banda ancha.

<http://blog.asteriskguide.com/bandcalc/bandcalces.php>

5.12 Sumario

En este capítulo usted aprendió que Asterisk trata la voz como un canal, así como un canal TDM. Aprendió como todo encaja dentro del modelo OSI. Aprendió a comparar los protocolos H.323 el más viejo, IAX2 el más eficiente, y todavía poco adoptado y SIP el más aceptado, pero con problemas serios con NAT. Entendió como diferenciar users (client), peers(server) y friends(ambos). Entendió el concepto de codecs y traducción de codecs.

5.13 Cuestionario

1. Cite por lo menos cuatro beneficios del uso de voz sobre IP

2. Convergencia es la unificación de las redes de voz, vídeo y datos en una única red y su principal beneficio es la reducción con los costos de manutención de redes separadas.

- Correcto
- Incorrecto

3. Asterisk no puede usar simultáneamente recursos de PSTN (Red pública de telefonía y de voz sobre IP, pues los codecs no son compatibles).

- Correcto
- Incorrecto

4. La Arquitectura de Asterisk es de un SIP proxy con posibilidad de uso de otros protocolos.

- Correcto
- Incorrecto

5. Dentro del modelo OSI, los protocolos SIP, H.323 e IAX2 están en la capa de:

- Presentación
- Aplicación
- Física
- Sesión
- Enlace

6. SIP es hoy el protocolo más abierto (IETF) siendo implementado por la mayoría de los fabricantes.

- Correcto
- Incorrecto

7. H.323 es un protocolo sin expresión, poco usado fue abandonado por el mercado en favor de SIP.

- Correcto
- Incorrecto

8. IAX2 es un protocolo propietario de Digium, a pesar de tener poca adopción por fabricantes de teléfono el IAX es excelente en las cuestiones de:

- Uso de banda ancha
- Uso de vídeo
- Pasaje por redes que poseen NAT
- Estandarizado por órganos como la IETF e ITU.

9. "Users" pueden recibir llamadas

- Correcto
- Incorrecto

10. Sobre los codecs asignarle lo que es verdadero

- G711 es el equivalente al PCM (Pulse Code Modulation) y usa 64 Kbps de banda ancha.
- G.729 es gratuito por esto es el más utilizado, usa apenas 8 Kbps de banda ancha.
- GSM viene creciendo, pues ocupa 12 Kbps de banda ancha y no precisa de licencia.
- G711 u-law es común en los EUA en tanto a-law es común en Europa y en Brasil.
- G.729 es leve y ocupa poca CPU en su codificación.

Exclusivo para docente Miguel Perez Paredes

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

El Protocolo IAX

En este capítulo vamos a aprender un poco sobre el protocolo IAX, sus puntos fuertes y puntos flacos. Detalles como interfaces trunked e integración entre dos Asterisks también serán vistos.

6.1 Objetivos del Capítulo

Objetivos

Para el final de este capítulo usted estará apto para:

- Identificar intensidad y debilidades del protocolo IAX
- Describir escenarios de uso del protocolo IAX
- Describir las ventajas del modo trunk
- Configurar `iax.conf` para teléfonos
- Configurar `iax.conf` para conexiones con proveedores VoIP.
- Configurar `iax.conf` para interconexiones Asterisk
- Debuggear conexiones en IAX
- Comprender la autenticación en IAX

6.2 Introducción

Todas las referencias al IAX en este documento corresponden a la versión 2, normalmente llamado IAX2. El IAX2 substituyó al IAX y como tal nos vamos a referir a él como apenas IAX de aquí en adelante.

Inter-Asterisk eXchange Protocol proporciona control y transmisión de voz sobre redes IP. El IAX puede ser usado con cualquier tipo de medio como voz y vídeo, pero fue pensado principalmente para llamadas de voz. Los objetivos del proyecto de IAX derivarán de la experiencia con los protocolos de voz sobre IP como el SIP (Sesión Initiated Protocol) y el MGCP (Media Gateway Control Protocol) para control y el RTP para el flujo-multimedia (streaming media) y son:

- Minimizar el uso de banda ancha para el tráfico de ambos, media y control con énfasis específica en llamadas de voz individuales.
- Proveer transparencia a NAT (Network Address Translation).
- Tener la posibilidad de transmitir informaciones sobre el plan de discado.
- Soportar la implantación eficiente de recursos de paging y intercomunicación.

6.3 Teoría de operación

IAX es un protocolo de media y señalización "peer-to-peer". Eso significa que los dispositivos mantienen conexiones asociadas con las operaciones de protocolo. Con respecto a los componentes de señalización de IAX, este tiene más parecido con el SIP que con el MGCP, que es un protocolo de control de tipo maestro-esclavo.

El abordaje del proyecto básico de IAX multiplexa la señalización y múltiples flujos sobre una única asociación UDP entre dos hosts de Internet. En esta faceta del proyecto, este se torna de los protocolos, el primero es el protocolo de señalización de las sesiones, el segundo el protocolo para transportar los flujos de media. Este abordaje difiere de la arquitectura general de los protocolos basados en la IETF con los protocolos separados para, señalización (MGCP e SIP) y flujo de media (RTP/RTCP). Como el IAX usa el mismo protocolo para señalización y media en un mismo puerto UDP, este no sufre de los problemas de atravesar dispositivos que realizan NAT (Network Address Translation), como, por ejemplo, ruteadores ADSL. (Característica fundamental para operadoras de telefonía IP). El IAX usa el puerto UDP 4569 para comunicar todos los paquetes. El IAX entonces usa un número de llamada de 15 bits para multiplexar los flujos sobre una única asociación UDP.

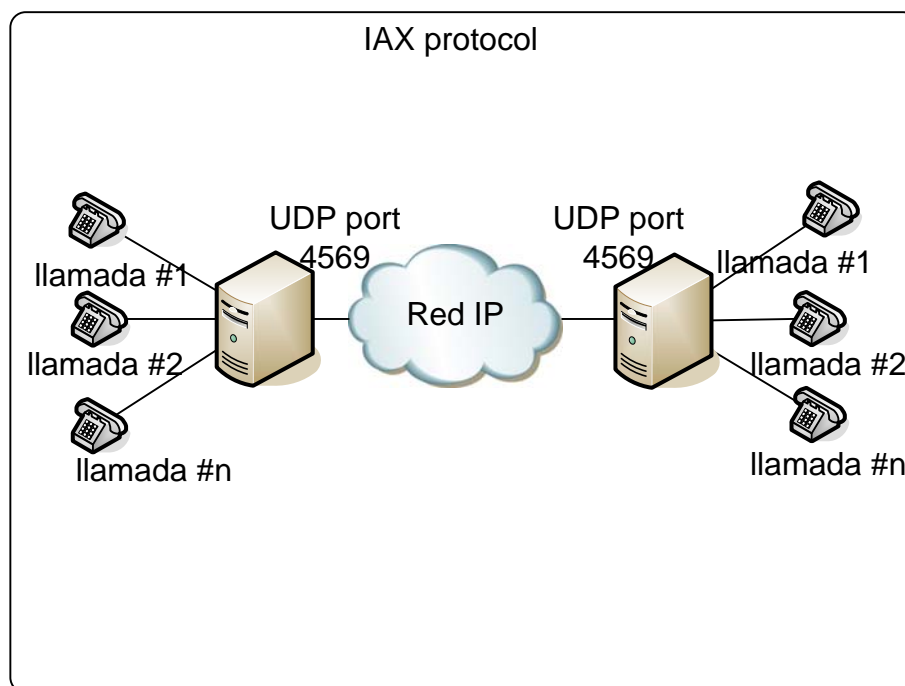


Figura 6.1: Múltiples llamadas sobre una única asociación de puerto UDP.

El valor 0 es un número especial de llamada reservado en cada host. Cuando intenta hacer una llamada, el número de llamada de destino todavía no es conocido. Un número de llamada cero es usado en esta situación. IAX es un protocolo binario. El diseño fue hecho para mejorar la eficiencia en el uso de banda ancha. Más allá de eso, el protocolo es optimizado para hacer un uso eficiente de banda ancha para cada llamada individual.

El protocolo IAX emplea un proceso similar de registro y autenticación al que usa SIP.

6.4 Formato de los Frames

Los mensajes IAX son llamadas frames. Existen varios tipos básicos de frames. Cada uno de los tipos es descrito en esta sección. Un bit F es usado para indicarse que el frame está completo (Full) o no lo está. El valor 0 indica que está completo. Un número de llamada de 15 bits es usado para identificar el punto final de flujo de media. Un valor de 0 indica que el punto final no es conocido. Una llamada tiene dos números de llamada asociados con ella en cualquier de las dos direcciones. El horario (timestamp) puede ser un campo de 32 o 16 bits. De cualquier forma el campo ocupa 32 bits.

6.4.1 Frame completo

Un frame completo puede ser usado para enviar señalización, audio y vídeo de forma confiable. El frame completo es el único tipo de frame que

es transmitido de forma confiable. Esto significa que el recipiente debe retornar algún tipo de mensaje al emisor después del recibimiento.

										1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
F	Número Originador de Llamada															R	Número de Destino de Llamada															
Timestamp																																
OSeqno								ISegno								Frame Type								C	Subclase							

Figura 6.2 Formato binario de un frame completo

La figura de arriba ilustra el formato binario de un frame completo. La tabla de abajo describe cada uno de los campos. El bit R es marcado para indicar que un frame está siendo retransmitido. La retransmisión ocurre después de un período de timeout y las retransmisiones son intentadas varias veces, dependiendo del contexto. El número de secuencia del flujo de salida "OSeqno" inicia con 0 y se incrementa de uno en uno. El campo "OSeqno" es usado para identificar el orden de los frames de media. ISegno es lo mismo solo que en el sentido de entrada (Inbound).

Campo	Descripción
F	Marcado en 1 indica que es un frame completo
Source Call Number	Número de llamada originador del lado de transmisión del frame
R	Marcado en uno indica que el frame está siendo retransmitido y el valor 0 que es transmisión inicial.
Destinación Call Number	Número de llamada de destino del lado receptor del frame.
Timestamp	Timestamp completo 32-Bits
OSeqno	Número de secuencia del flujo de salida
ISegno	Número de secuencia del flujo de entrada
Frame Type	Tipo de Frame
C	Formato de valor de la subclase
Subclase	Subclase

Figura 6-3 Descripción de los campos de Frame Completo

6.4.2 MiniFrame

El miniframe es usado para enviar el audio o vídeo (media) con un mínimo de sobrecarga de protocolo. El formato del miniframe se describe abajo.

										1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
F	Número originador de llamada															Timestamp															
Datos																															

Figura 6-4 Formato binario de un mini-frame

El timestamp del Mini Frame esta truncado. El cliente generalmente mantiene el timestamp completo de 32 bits. Cuando es enviando el mini frames, los 16 bits de orden más bajo son enviados en el campo

timestamp. Cuando el timestamp de 16 bits da la vuelta (imprevisto) un frame completo es enviado para permitir que el otro lado sincronice.

Campo	Descripción
F	Marcado en 0 indica que es un frame incompleto
Source Call Number	Número de llamada originador del lado de transmisión del frame completo
Timestamp	Timestamp 16-Bits
Datos	Datos

Figura 6-5 Descripción de los campos de Mini Frame

Una descripción completa del protocolo IAX puede ser encontrada en <http://www.cornfed.com/iax.pdf>

6.5 Uso de banda ancha

El uso de banda ancha en voz sobre IP es modificado por una serie de factores. Desde el CODEC, hasta cuestiones como compresión de encabezados.

El IAX permite el uso del modo Trunked. En este caso, cuando más de una llamada es hecha, el overhead de los encabezados IP es disminuido, encaminando múltiples paquetes de voz de diferentes llamadas en un único paquete. Con esto la necesidad de banda ancha es reducida. Uno de los buenos estudios sobre la banda ancha usada por el IAX fue publicada por John Todd da Loligo en <http://www.voip-info.org/wiki-Asterisk+bandwidth+iax2> y es reproducido abajo:⁴

Resultados de los tests:

G. 711 (ulaw)

Una llamada: 164333.75 bps /94.26 pps (82.1 kbps)

Dos llamadas: 296171.60 bps / 101.46pps (148 Kbps)

De este modo:

Para cada llamada adicional: 131837 bps (65.9 Kbps)

Sobrecarga estimada de IP/IAX (1 llamada) 32495 bps (16.0 Kbps)

Número de llamadas por megabit (15)

ILBC: (veja nota abajo)

Una llamada: 56134.91 bps/67.45 pps (28 kbps)

Dos llamadas: 98679.11 bps/102.41 pps (49.3 Kbps)

De este modo:

Para cada llamada adicional: 42544 bps (21.2 Kbps)

Sobrecarga estimada IP/IAX2 (1 llamada): 13590 bps (6.7 Kbps)

Número estimado de llamadas por megabit: 47

G. 729

Una llamada: 60124.33 bps/101.26 pps (30.0 Kbps).

Dos llamadas: 79496.23 bps/102.85 pps (39.7 Kbps).

De este modo:

Para cada llamada adicional: 19732 bps (9.6 Kbps)

⁴ Reproducido con el permiso del autor.

Sobrecarga estimada IP/IAX2 (1 llamada): 40572 bps (20.3 Kbps)
 Número estimado de llamadas por megabit: 103

GSM

Una llamada: 70958.16 bps/102.13 pps (35.4 kbps)
 Dos llamadas: 100455.23 bps/102.63 pps (50.2 Kbps)

De este modo:

Para cada llamada adicional: 29497 bps (14.7 Kbps)
 Sobrecarga estimada IP/IAX2 (1 llamada): 41461 bps
 Número estimado de llamadas por megabit: 68

LPC10

Una llamada: 43855.44 bps/89.94 pps (21.9 kbps).
 Dos llamadas: 56059.18 bps/100.81 pps (28.0 kbps)

De este modo:

Para cada llamada adicional: 12203 bps (6.1 Kbps)
 Sobrecarga estimada IP/IAX2 (1 llamada): 31561 bps (15.8 kbps)
 Número estimado de llamadas por megabit: 164

SPEEX:

Una llamada: 74817.18 bps/101.06 pps (37.14 kbps)
 Dos llamadas: 109692.68 bps/102.18 pps (54.8 kbps)

De este modo:

Para cada llamada adicional: 34875 bps (17.4 kbps)
 Sobrecarga estimada IP/IAX2 (1 llamada): 39941 bps (19.9 kbps)
 Número estimado de llamadas por megabit: 57

Conclusiones sobre el estudio:

- El uso de banda ancha es influenciado por muchas variables como el número de llamadas simultáneas, tipo de protocolo de capa 2, el hecho de que la red sea half o full-duplex.
- Una medición de 74817.18 bps es reportada como 37.15 kbps pues el link Ethernet del servidor donde esto fue medido es Ethernet Full-Duplex (posee un canal de entrada y salida). En una Wan que normalmente es full-duplex también la banda ancha va a variar un poco por causa del cambio de encabezado de capa 2. Normalmente en Wan el encabezado es un poco menor que en una LAN.
- Los datos relativos al ILBC difieren un poco de la realidad, porque el ILBC envía frames cada 30 ms y el trunking cada 20 ms. Corrigiendo los resultados (vea más detalles en el estudio original) se llega a los siguientes números: Para cada llamada adicional 14.6 Kbps, sobrecarga estimada con los encabezados IP/IAX2: 13.4 Kbps, número de llamadas por megabit 67.
- El mejor uso de banda ancha fue obtenido por el LPC10 sin tener en cuenta la calidad de sus llamadas. Ya el G729 mostró que tiene un excelente uso de banda ancha manteniendo una calidad de voz impecable.

6.6 Nomenclatura de los canales

6.6.1 Formato de una conexión de salida.

```
IAX2/[<user>[: <secret>]@]<peer>[: <portno>][/<exten>[@<context>][/<options>]
```

<user>	Identificación de usuario en punto remoto, o nombre de cliente configurado en iax.conf (opcional)
<secret>	Contraseña (opcional). Como alternativa puede ser el nombre de archivo de una llave RSA sin la extensión (.key, ou .pub), y dentro de [llaves] como por ej. [elnombredearchivo]
<peer>	Nombre de servidor al cual se conecta
<portno>	Número de puerto para la conexión en el servidor(opcional).
<exten>	Extensión en el servidor Asterisk remoto(opcional) <context>: Contexto para usar en el servidor asterisk remoto (opcional)
<options>	La única opción disponible es 'a' que significa pedido para auto-responder.

6.6.2 Ejemplos de canales de salida:

IAX2/mark:asdf@myserver/6275@default	Llame "myserver" usando "mark" como nombre de usuario y asdf como contraseña y pida la extensión 6275 en contexto default.
IAX2/iaxphone/s/a	llame al "iaxphone" pidiendo respuesta inmediata
IAX2/john:[johnrsa]@somewhere.com	Llame somewhere.com, usando john como nombre de usuario y una llave RSA como autenticación. El formato de un nombre de canal

	IAX usado para una conexión de entrada es simplemente:
--	--

6.6.3 Formato de una conexión de entrada

```
IAX2[[<username>@]<host>]/<callno>
```

<username>	El nombre do usuario
<host>	El host para conectarse
<callno>	El número de llamada local

6.6.4 Ejemplo de canales de entrada

IAX2[mark@192.168.0.1]/14	llame al número 14 de usuario "mark" en la ip 192.168.0.1
IAX2[192.168.10.1]/13	llame al número 13 de ip 192.168.10.1

6.7 Escenarios de uso

Los escenarios de uso nos van a permitir mostrar el archivo de configuración de IAX con más detalles y enseña a configurar el Asterisk en cada una de las situaciones. Las configuraciones relacionadas al IAX son ejecutadas en el archivo `iax.conf` del directorio `/etc/asterisk`. Las configuraciones que vamos a mostrar son:

- Conectando un softphone IAX al Asterisk
- Conectando el IAX a un proveedor VoIP basado en IAX
- Conectando dos servidores a través de IAX
- Conectando dos servidores a través de IAX2 en modo trunking
- Debugando una conexión IAX

6.7.1 Conectando un softphone IAX al Asterisk

Asterisk soporta teléfonos IP basados en IAX como el SNOM y el AIXy y también softfones como el idéfisk. Para configurar un teléfono IAX basta con configurar el archivo `iax.conf` y el propio teléfono.

Vamos a abordar en este capítulo el IDEFISK que es gratuito y puede ser bajado de Internet a partir de www.asteriskguru.com.

```
[general]
bindport=4569
bindaddr=0.0.0.0
bandwidth=high
disallow=all
allow=ulaw
```

```
[guest]
type=user
context=default
callerid="Guest IAX User"
;
; Trust Caller*ID Coming from iaxtel.com
;
;
[iaxtel]
type=user
context=default
auth=rsa
inkeys=iaxtel
;
; Trust Caller*ID Coming from iax.fwdnet.net
;
;
[iaxfwd]
type=user
context=default
auth=rsa
inkeys=freeworddialup

[2003]
type=friend
context=default
secret=contraseña
host=dynami c
```

Los parámetros alterados en la configuración que aparecen destacados son:

```
bandwidth=high
```

Este comando hace que codecs de banda alta como el G711 puedan ser seleccionados. Si usted usa `bandwidth=low` y selecciona como codec `ulaw` (g.711 ulaw) el Asterisk dará un mensaje de que ningún codec está disponible.

```
disable=al l
allow=ul aw
```

En los comandos de arriba deshabilitamos todos los codecs y habilitamos apenas el `ulaw` (g.711 ulaw). En redes locales es el codec más indicado pues consume poca cpu de Asterisk. La cuestión de banda ancha con 100 Mbps es secundaria cuando se considera una red local. En una red WAN difícilmente seleccionaríamos "ulaw"

```
[2003]
type=friend
context=default
secret=contraseña
host=dynami c
```

En los comandos de arriba definimos un friend de nombre "[2003]", el contexto es el default (por el momento vamos a usar apenas el default).

“Host=dynamic” va a indicar que el teléfono se va a registrar para informar que dirección IP está usando.

Hecho esto basta configurar el softphone.

6.7.2 LAB - Instalando un softfone IAX

Paso 1: Configure el archivo iax.conf, como el ejemplo de arriba.

Paso 2: Haga un download del software idefisk para su plataforma favorita.

http://www.asteriskguru.com/tools/idefisk/idefisk133_installer.exe

Cuidado: URLs frecuentemente cambian y el software probablemente no estará en este exacto lugar cuando usted decida hacer el download. Nada que usted no pueda resolver navegando un poco.

Otros softphones o teléfonos IP basados en IAX pueden ser usados para hacer este LAB.

Paso 3: Instale el idefisk ejecutando el utilitario de instalación.

Paso 4: Configure la cuenta de Asterisk haciendo click con el botón derecho sobre el icono de idefisk en la barra de herramientas, o usando ALT+A.

Usted va a ver una pantalla semejante a la de abajo:

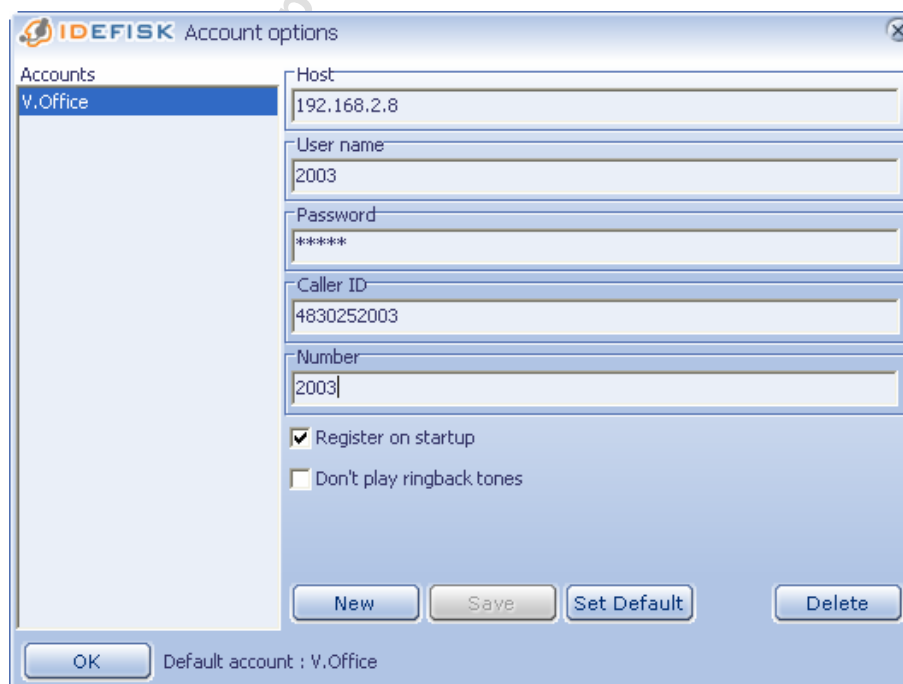


Figura 6-6 Configuración de cuenta de cliente

Paso 5: Configure el archivo `extensions.conf` de forma que usted pueda testear su cliente IAX

```
[default]
exten=>2000, 1, Dial(SIP/2000)
exten=>2001, 1, Dial(SIP/2001)
exten=>2003, 1, Dial(IAX2/2003)
```

Ahora usted puede discar entre teléfonos basados en SIP y teléfonos basados en IAX2.

Exclusivo para Vicente Miguel Perez Paredes

6.7.2 Conectándose a un proveedor basado en IAX2

Todavía existen pocos proveedores basados en IAX2, pero estos ciertamente son una buena opción principalmente si usted está atrás de NAT. Dos proveedores conocidos son el freeworlddialup (www.freeworlddialup.com) y el iaxtel (www.iaxtel.com). El proveedor freeworlddialup necesita una configuración especial para habilitar el protocolo IAX. Entre en contacto con su proveedor para tener instrucciones de como habilitar el IAX2.

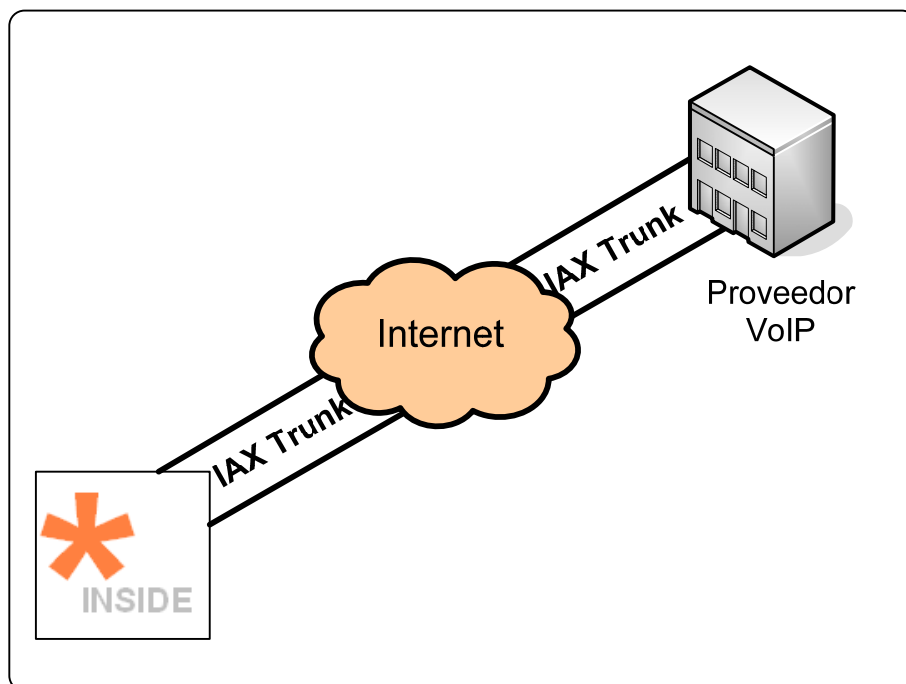


Figura 6.2 Conectándose a un proveedor en IAX2

6.7.3 LAB - Conectándose al freeworlddialup usando IAX2

Paso 1: Abra una cuenta en su proveedor favorito. Use el freeworlddialup si usted no conoce ninguno, este es gratuito.

Paso 2: Habilite el protocolo IAX2 si es necesario en su proveedor

En el freeworld dialup entre en "extra features" (Recursos extras) y haga click en el cuadro de texto de IAX como abajo se muestra:

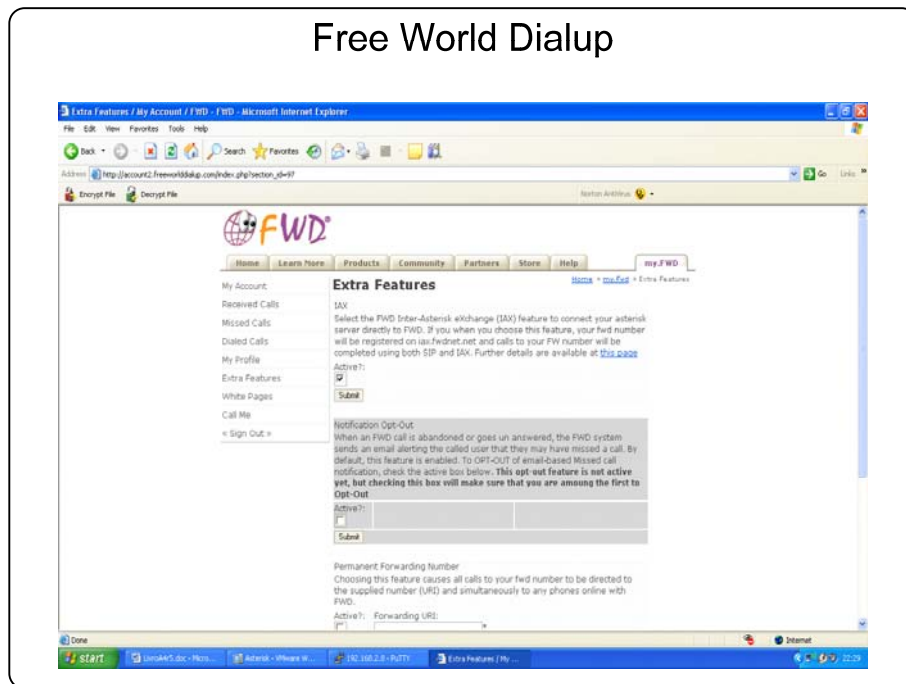


Figura 6.3 habilitando iax2 en el freeworlddialup

Aviso: Las pantallas y urls frecuentemente cambian de layout y posición. Navegue para hallar el lugar correcto para habilitar el IAX2.

Paso 3: Configure el archivo `iax.conf` para registrar su Asterisk en el proveedor. Adicione las siguientes líneas al archivo. La línea de abajo debe ser colocada dentro de la sección `[general]`.

```
[general]
register=>621538:contraseña@iax2.fwdnet.net/2003
```

En la línea de arriba usted se estará registrando con el proveedor con su nombre y contraseña. En el momento de recibir una llamada esta será encaminada para el ramal 2003.

```
[621538] ; su número en el proveedor
type=peer
secret=contraseña ; su contraseña en el proveedor
host=iax2.fwdnet.net
```

En las líneas de arriba estamos registrando al proveedor permitiendo el discado para él.

```
[iaxfwd]
type=user
context=default
auth=rsa
ikeys=freeworlddialup
```

Retire el comentario de la sección [iaxfwd] del archivo `iax.conf`. Esto es necesario para que ocurra la autenticación. A través del uso de llave pública usted puede tener certeza de que la llamada está siendo recibida de "freeworlddialup". Si alguien intenta usar este mismo camino las llaves de autenticación no van a dar.

Paso 4: Testeando la conexión con el proveedor.

Para testear la conexión vamos a usar un servicio de proveedor FreeWorldDialup llamado hora correcta. Discando 612 en el proveedor podremos oír una URA que dirá la hora exacta (en inglés).

Para esto vamos a configurar el archivo `extensions.conf` y adicionar la siguiente línea en el contexto `default`.

```
exten=>612, 1, Dial(IAX2/621538:contraseña@iax2.fwdnet.net/612, 20, r)
```

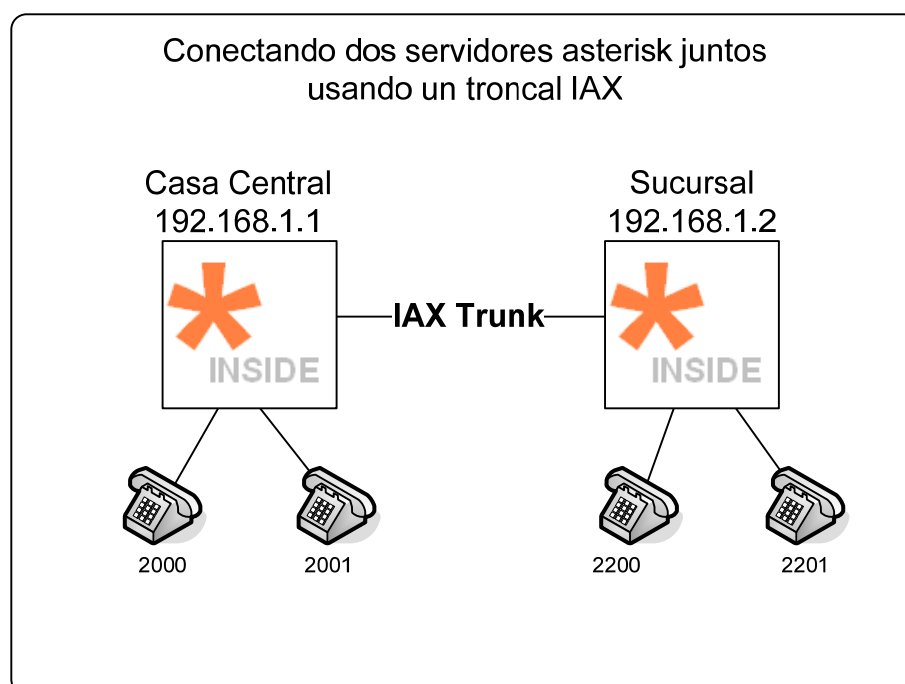
Entre a la interfase de línea de comando de Asterisk, haga un reload y enseguida use el comando:

```
CLI>reload  
CLI>iax2 show register
```

Para verificar si el sistema se registró adecuadamente en el proveedor.

Para llamar el proveedor simplemente marque 612 a partir del teléfono.

6.7.4 Conectando servidores Asterisk usando IAX2 Trunk.



Dos servidores Asterisk pueden ser conectados con el protocolo IAX2.

Para configurar un enlace IAX, vamos a usar el concepto de maestro-esclavo para tornar más fácil el aprendizaje.

Los siguientes pasos tienen que ser hechos para que el servidor esclavo pueda hacer y recibir llamadas del servidor maestro. Para mostrar un ejemplo colocamos todas las llamadas en el contexto [default]. Más tarde en este material usted va a aprender a colocar las llamadas en contextos más apropiados.

Paso 1: Configurar el archivo `iax.conf` del servidor esclavo para:

- 1.1. Registrarse en el maestro.
- 1.2. Recibir llamadas del maestro.

Paso 2: Configurar el archivo `iax.conf` del servidor maestro para:

- 1.3. Recibir llamadas del esclavo.
- 1.4. Registrarse en el esclavo.

Paso 3: Configurar el archivo `extensions.conf` del esclavo:

- 1.5. Para discar para el maestro.

2. Configurar el archivo `extensions.conf` del maestro

- 2.1. Para discar para el esclavo.

Vamos a suponer para el ejemplo de abajo que **todos los ramales en el servidor maestro comienzan con 20xx y todos los ramales del esclavo comienzan con 22xx.**

Paso 1 - iax.conf del servidor esclavo

1.1 Registrarse en el maestro

En el archivo iax.conf

```
register=>maestro: contraseña@maestro.net.br
```

1.2 Recibir llamadas del maestro

En el archivo iax.conf

```
[maestro]
type=user
auth=plaintext
context=default
secret=contraseña
host=dynamic
callerid='master'
trunk=yes
notransfer=yes

[maestro]
type=user
auth=plaintext
context=default
secret=contraseña
callerid='master'
trunk=yes
notransfer=yes

[2000]
type=friend
auth=plaintext
context=default
secret=contraseña
callerid='master'

[2001]
..... ; Definición de los otros canales IAX (Ramas)
```

Paso 2. Configurar el archivo iax.conf del maestro

2.1 Recibir y hacer llamadas del esclavo y el registro

```
[esclavo]
type=user
auth=plaintext
context=default
secret=contraseña
host=dynamic
callerid='esclavo'
trunk=yes
notransfer=yes

[esclavo]
```

```

type=user
auth=pl ai ntext
context-default t
secret=contraseña
calleri d=' esclavo'
trunk=yes
notransfer=yes

[2200]
type=fri end
auth=pl ai ntext
context-default t
secret=contraseña
calleri d=' master'

[2201]
..... ; Defi ni ción de los otros canales IAX (Ramas)

```

2.2 Registrarse en el esclavo

```
regi ster=>escl avo: contraseña@escl avo. net. br
```

Paso 3 – Configurar el archivo `extensions.conf` en el servidor esclavo.

```

[defaul t]
exten => _20XX, Di al (IAX2/master/${EXTEN})

exten=>_22XX, Di al (IAX2/${EXTEN})

```

Paso 4 – Configurar el archivo `extensions.conf` en el servidor maestro.

```

[defaul t]
exten => _22XX, Di al (IAX2/master/${EXTEN})

exten=>_20XX, Di al (IAX2/${EXTEN})

```

6.8 Autenticación en el IAX

Vamos a analizar la autenticación del IAX desde el punto de vista práctico y aprender a escoger las mejores opciones dependiendo de las necesidades de seguridad de la configuración.

6.8.1 Conexiones de entrada

Cuando Asterisk recibe una conexión de entrada, la información de la llamada inicial puede incluir un nombre de usuario (De campo **username**) o no. Más allá de esto, la conexión de entrada tiene una dirección IP que Asterisk usa para la autenticación también.

Si el nombre de usuario es proporcionado, Asterisk hace lo siguiente:

1. Busca el `iax.conf` para una entrada `"type-user"` (o `"type=friend"`) con la sesión nombre (`[username]`); si no la encuentra, recusa la conexión.
2. Si la entrada encontrada tiene configuraciones de tipo `"deny/allow"` (negar/permitir), compara la dirección IP del originador de la llamada. Si la conexión no fuese permitida, recusa la conexión.
3. Hace el chequeo de la contraseña (`secret`) (`plaintext`, `md5` o `rsa`); si falla, recusa a conexión.
4. Acepta la conexión y envía para el originador para el contexto específico en la configuración `context=` de la entrada en el archivo `iax.conf`.

Si un nombre de usuario no es proporcionado, Asterisk hace lo siguiente:

1. Busca por un `"type=user"` (o `type="friend"`) en archivo `iax.conf` sin un secreto especificado y también en las restricciones de tipo `"deny/allow"`. Si una entrada es encontrada, acepta la conexión y usa el nombre de la entrada del `iax.conf` como el nombre de usuario que se esta conectando.
2. Busca por un `"type=user"` (o `type="friend"`) en el `iax.conf` con un `secret` o llave RSA especificado y también verifica restricciones `"deny/allow"`. Si una entrada es encontrada, intenta autenticar al que llama usando el secreto especificado o la llave, y si está es correcta, acepta la conexión y usa el nombre encontrado en el `iax.conf` como nombre de usuario.

Suponga que su archivo `iax.conf` tenga las siguientes entradas:

```
[guest]
type=user
context=guest

[i axtel ]
type=user
context=i ncomi ng
auth=rsa
i nkeys=i axtel

[i ax-gateway]
type=fri end
al low=192. 168. 0. 1
context=i ncomi ng
host=192. 168. 0. 1

[i ax-fri end]
type=user
secret=thi s_i s_secret
```



```
auth=md5
context=incoming
```

Si una llamada de entrada tiene un nombre de usuario especificado como:

guest

iaxtel

iax-gateway

iax-friend

Entonces el Asterisk intentará autenticar la conexión usando la entrada correspondiente en `iax.conf`. Si cualquier otro nombre de usuario fuese especificado, la conexión será recusada.

Si ningún usuario ha sido especificado, Asterisk autenticará la conexión como usuario `guest` (invitado). Entretanto si usted no tuviese un usuario `guest` en su archivo `iax.conf`, el usuario que se estuviese conectando, puede especificar el secreto asociado con cualquier otro `iax.conf` que no tenga una dirección IP restringida. En otras palabras, si usted no tiene una entrada `guest` en su `iax.conf`, usted tiene varias entradas con contraseñas que pueden ser adivinadas y que van a permitir que el usuario se conecte en el sistema.

Usar llaves RSA como sus secretos es una forma de evitar este problema sin crear un usuario `guest`. Entradas RSA no son "adivinables", otro método es usar restricciones de IP en tantas entradas como sea posible.

6.8.2 Conexiones de salida

Las conexiones de salida pueden tomar sus informaciones de autenticación de:

- La descripción del canal IAX2 pasando por el comando `Dial()`.
- Una entrada "type=peer o friend" en el archivo `iax.conf`.
- Una combinación de los dos.

Suponga que su archivo `iax.conf` tenga las siguientes entradas:

```
[i axtel -outbound]
type=peer
username=i axtel_username
secret=i axtel_secret
host=i axtel.com
```

```
[i ax-gateway]
type=friend
allow=192.168.0.1
context=incoming
host=192.168.0.1
```

Entonces el comando:

```
Di al (IAX2/iaxtel-outbound/1234)
```

Conectaría al host iaxtel.com, usando el usuario y contraseña especificados en el archivo iax.conf para la autenticación.

El comando:

```
Di al (IAX2/user2:secret2@iaxtel-outbound/1234)
```

También se irá a conectar con el host iaxtel.com, pero especificará user2:secret2 como nombre y contraseña respectivamente y esto sobrescribe las entradas en el iax.conf.

```
Di al (IAX2/iaxtel.com/1234)
```

Nuevamente la conexión sería hecha al host iaxtel.com, aunque ningún nombre de usuario fue especificado. Como ninguna de las entradas en el archivo iax.conf da con la descripción del canal iaxtel.com la conexión es rechazada.

```
Di al (IAX2/iax-gateway/5678)
```

Si se conectasen al host 192.168.0.1, sin especificar ningún nombre, y si el host pide un secreto, ningún secreto será proporcionado. Presumiblemente este tipo de entrada sería usada para conexiones entre hosts con alto grado de confianza.

6.9 Configuración del archivo iax.conf

El archivo iax.conf posee una gran cantidad de configuraciones y discutir las una por una sería tedioso y poco útil. Vamos a discutir algunas opciones importantes de la sesión [general] las sesiones por peer, user o friend.

El archivo de configuración de IAX es una colección de secciones, cada cual representa una entidad dentro del objetivo de IAX (a excepción de la sesión general).

La primera sesión es típicamente la sesión general. En esta área, un número de parámetros pueden ser configurados afectando todo el

sistema. Específicamente los codecs default, puertos, direcciones, ancho del jitter, bits de TOS y registros.

6.9.1 Configuración del direccionamiento

El direccionamiento es configurado en la sesión **[general]**

Direccionamiento del servidor:

bindport = <portnum>	Esto configura el puerto en el que el IAX se va a conectar. El puerto default es 5036. Es recomendable mantener este valor.
bindaddr = <ipaddr>	Esto permite conectar el IAX a una dirección IP específica en vez de ligar el * a todas las direcciones.

6.9.2 Selección de los codecs:

La selección de los codecs puede ser hecha en la sesión [general] o en cada entidad individualmente.

bandwidth = [low medium high]	La selección de banda ancha inicializa la selección de codecs a valores apropiados para una dada banda. Escogiendo "High" habilita todos los codecs y es recomendado para conexiones de 10Mbps para arriba. Escogiendo "medium" elimina los codecs U-law A-law dejando apenas los codecs con 32Kbps o menos (Como MP3 como un caso especial). Esto puede ser usado en conexiones de banda ancha si se desea. El "low" elimina el ADPCM y MP3, dejando apenas el G723, GSM y LPC10.
allow/disallow=[gsm lpc10 g723.1 adpcm ulaw alaw mp3 linear all]	Los comandos "allow" y "disallow" permiten hacer una sintonía fina en la selección de codecs.

6.9.3 Configuración del buffer de jitter y marcación de paquetes:

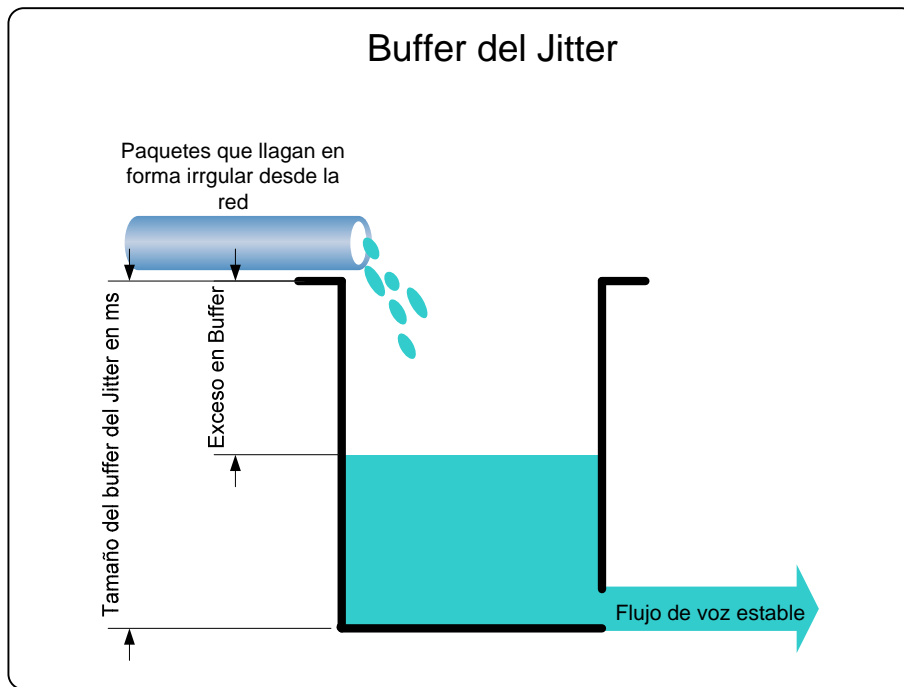


Figura 6.5 Buffer del jitter

Jitter también conocido como variación de atraso es uno de los grandes responsables en la reducción de calidad de voz. Un buffer de jitter es usado para compensar estas variaciones de atraso. Un jitter pequeño de menos de un frame de compresión (en ms) normalmente no es un problema, pero un jitter excesivo puede sin un buffer de jitter causar fallas en el audio. El buffer de jitter permite reducir estas variaciones a causa de un atraso mayor. De una forma general es importante mantener el atraso abajo de 150 ms y el delay abajo de 20 ms. Cuando usted configura un buffer de jitter el naturalmente va a atrasar el paquete en el exacto tamaño del buffer. Es importante pensar en un embotellamiento de atraso que usted puede consumir en la configuración de un buffer de jitter.

Los parámetros para regular el buffer de jitter siguen abajo:

- **jitterbuffer**=<yes/no> – Habilita o deshabilita el buffer de jitter.
- **Dropcount**=<número> - Esto representa el número máximo de frames que pueden estar atrasados en los últimos dos segundos. Se recomienda setear este número abajo de 3 (1,5% dos frames dropeados).
- **Maxjitterbuffer**=<número> - El tamaño de buffer del jitter. Normalmente esta abajo de 100 ms.
- **Maxexcessbuffer**=<número> - Si el atraso de la red mejora después de un período de jitter alto, es posible que el buffer sea mayor que lo necesario. Entonces si el buffer estuviese por encima del maxexcessbuffer, Asterisk empezará a reducirlo

- **Minexcessbuffer=<número>** - Este parámetro representa el espacio mínimo reservado para el buffer del jitter, si el buffer bajara de este número, Asterisk comenzará a aumentarlo.

El parámetro de abajo permite que los paquetes de Asterisk sean marcados para tratamiento especial en switches y ruteadores. Este parámetro es parte del campo TOS del encabezado IP que es seteado bit a bit. Para setear el TOS usted usa una combinación de los siguientes, por ejemplo:

<pre>tos=0x10 = low delay tos=0x08 = high throughput tos=0x04 = high reliability tos=0x02 = ECT bit set tos=0x01 = CE bit set</pre>

Ejemplo:

<pre>tos=0x05</pre>

Usted está diciendo que quiere alta confiabilidad junto con el bit CE habilitado.

6.9.4 Parámetros adicionales para tipo "User":

<pre>permit = <ipaddr>/<netmask> deny = <ipaddr>/<netmask></pre>	<p>Las reglas para permitir y negar (permit y deny) pueden ser aplicadas a usuarios, permitiendo a ellos conectarse de determinadas direcciones IP y no de otras. Las reglas son interpretadas en secuencia y son todas evaluadas en una dada dirección IP, con el resultado final siendo la decisión (Diferente de las listas de control de acceso Cisco y de la mayor parte de los Firewalls).</p> <p>Ejemplo:</p> <pre>permit=0.0.0.0/0.0.0.0 deny=192.168.0.0/255.255.255.0</pre> <p>Esto negará a cualquier en una 192.168.0.0 con mascara de 24 bits (clase C)</p> <pre>deny=192.168.0.0/255.255.255.0 permit=0.0.0.0/0.0.0.0</pre>
--	---

	Esto no negará a nadie ya que la regla final permite todos, de este modo sobrepasa la negación. Si ninguna regla estuviese listada, cualquiera puede conectarse de cualquier lugar.
callerid = <callerid>	Usted puede sobrescribir el identificador de llamada pasado por el usuario para usted (Si el elige enviarlo) de manera que el siempre este correcto desde el punto de vista de su servidor.
auth = [md5 plaintext rsa]	Usted puede seleccionar que métodos de autenticación son permitidos. Múltiples métodos pueden ser especificados, separados por comas. Si md5 o texto simple (plaintext) es seleccionado un secreto (secret) debe ser proporcionado. Si la autenticación RSA fuera especificada, entonces uno o más nombres de llaves deben ser especificados con "inkeys". Si ningún secreto y ningún método de autenticación fuera especificado, entonces ninguna autenticación es necesaria.
Secret = <secret>	La línea "secret" especifica el secreto compartido para los métodos md5 o autenticación en texto simple. No sugerimos la autenticación en texto simple excepto para debugging
inkeys = key1[:key2...]	La línea "inkeys" especifica que llaves nosotros podemos usar para autenticar un "peer" remoto. Los archivos de llave están en /var/lib/asterisk/keys/<name>.pub y son llaves públicas.

Parámetros de billing:

accountcode = <code> amaflags = [default omit billing documentation]:	Estos parámetros afectan la generación de detalles de llamadas. El primero configura el código de contabilización para registros recibidos con el IAX. El código de contabilización puede sobrescribir la base por usuario para llamadas entrantes. El "amaflags" controla como el registro es etiquetado
---	---

("omit" hace que ningún registro sea escrito. "billing" y "documentación" etiquetan los registros como registros para cobranza y documentación respectivamente y "default" selecciona el default del sistema).

6.10 Comandos de depuración de IAX2.

iax2 show netstats

```
vtsvoffi ce*CLI > iax2 show netstats
----- LOCAL -----
--- REMOTE ---
Channel          Del  Lost  % Drop  000  RTT  Jit  Del  Lost  % Drop  000  Kpkts  Jit
Del  Lost  % Drop  000  Kpkts
IAX2/8590-1      110   3   0   0   0   16  -1   0   -1  -1   0   -1    1   60
```

iax2 show channels

```
vtsvoffi ce*CLI > iax2 show channels
Channel          Peer          Username      ID (Lo/Rem)  Seq (Tx/Rx)
Lag      Jitter  JitBuf  Format
IAX2/8590-2  8.8.30.43  8590        00002/26968  00004/00003
00000ms -0001ms 0000ms unknow
```

iax2 show peers

```
vtsvoffi ce*CLI > iax2 show peers
Name/Username    Host          Mask          Port          Status
8584             (Unspeci fi ed) (D) 255.255.255.255 0
UNKNOWN
8564             (Unspeci fi ed) (D) 255.255.255.255 0
UNKNOWN
8576             (Unspeci fi ed) (D) 255.255.255.255 0
UNKNOWN
8572             (Unspeci fi ed) (D) 255.255.255.255 0
UNKNOWN
8571             (Unspeci fi ed) (D) 255.255.255.255 0
UNKNOWN
8585             (Unspeci fi ed) (D) 255.255.255.255 0
UNKNOWN
8589             (Unspeci fi ed) (D) 255.255.255.255 0
UNKNOWN
8590             8.8.30.43    (D) 255.255.255.255 4569          OK (16
ms)
3232             (Unspeci fi ed) (D) 255.255.255.255 0
UNKNOWN
9 iax2 peers [1 online, 8 offline, 0 unmonitored]
```

iax2 debug

Observe en el debug destacado el inicio y el fin de una llamada. Entre las llamadas observe las comunicaciones de registro de canal y el sistema de control da atraso y jitter (poke, pong) de donde vienen las estadísticas vistas en show channels.

```

vtsvoffice*CLI> iax2 debug
IAX2 Debugging Enabled

Rx-Frame Retry[ No] -- OSeqno: 000 ISeqno: 000 Type: IAX      Subclass:
REGREQ
  Timestamp: 00003ms  SCall: 26975  DCall: 00000 [8.8.30.43:4569]
  USERNAME           : 8590
  REFRESH            : 60

Tx-Frame Retry[000] -- OSeqno: 000 ISeqno: 001 Type: IAX      Subclass:
REGAUTH
  Timestamp: 00009ms  SCall: 00003  DCall: 26975 [8.8.30.43:4569]
  AUTHMETHODS        : 2
  CHALLENGE          : 137472844
  USERNAME           : 8590

Rx-Frame Retry[ No] -- OSeqno: 001 ISeqno: 001 Type: IAX      Subclass:
REGREQ
  Timestamp: 00016ms  SCall: 26975  DCall: 00003 [8.8.30.43:4569]
  USERNAME           : 8590
  REFRESH            : 60
  MD5 RESULT         : f772b6512e77fa4a44c2f74ef709e873

Tx-Frame Retry[000] -- OSeqno: 001 ISeqno: 002 Type: IAX      Subclass:
REGACK
  Timestamp: 00025ms  SCall: 00003  DCall: 26975 [8.8.30.43:4569]
  USERNAME           : 8590
  DATE TIME          : 2006-04-17 16:03:00
  REFRESH            : 60
  APPARENT ADDRES   : IPV4 8.8.30.43:4569
  CALLING NUMBER     : 4830258590
  CALLING NAME       : Flavi o

Rx-Frame Retry[ No] -- OSeqno: 002 ISeqno: 002 Type: IAX      Subclass: ACK
  Timestamp: 00025ms  SCall: 26975  DCall: 00003 [8.8.30.43:4569]
Tx-Frame Retry[000] -- OSeqno: 000 ISeqno: 000 Type: IAX      Subclass: POKE
  Timestamp: 00003ms  SCall: 00006  DCall: 00000 [8.8.30.43:4569]
Rx-Frame Retry[ No] -- OSeqno: 000 ISeqno: 001 Type: IAX      Subclass: ACK
  Timestamp: 00003ms  SCall: 26976  DCall: 00006 [8.8.30.43:4569]
Rx-Frame Retry[ No] -- OSeqno: 000 ISeqno: 001 Type: IAX      Subclass: PONG
  Timestamp: 00003ms  SCall: 26976  DCall: 00006 [8.8.30.43:4569]
  RR_JITTER          : 0
  RR_LOSS            : 0
  RR_PKTS            : 1
  RR_DELAY           : 40
  RR_DROPPED         : 0
  RR_OUTOFORDER      : 0
Tx-Frame Retry[-01] -- OSeqno: 001 ISeqno: 001 Type: IAX      Subclass: ACK
  Timestamp: 00003ms  SCall: 00006  DCall: 26976 [8.8.30.43:4569]

Rx-Frame Retry[ No] -- OSeqno: 000 ISeqno: 000 Type: IAX      Subclass: NEW
  Timestamp: 00003ms  SCall: 26977  DCall: 00000 [8.8.30.43:4569]
  VERSION            : 2
  CALLING NUMBER     : 8590
  CALLING NAME       : 4830258590

```



```

FORMAT          : 2
CAPABILITY     : 1550
USERNAME       : 8590
CALLED NUMBER  : 8580
DNID           : 8580

```

```

Tx-Frame Retry[000] -- 0Seqno: 000 ISeqno: 001 Type: IAX      Subclass:
AUTHREQ

```

```

Timestamp: 00007ms SCall: 00004 DCall: 26977 [8.8.30.43:4569]
AUTHMETHODS      : 2
CHALLENGE        : 190271661
USERNAME         : 8590

```

```

Rx-Frame Retry[Yes] -- 0Seqno: 000 ISeqno: 000 Type: IAX      Subclass: NEW

```

```

Timestamp: 00003ms SCall: 26977 DCall: 00000 [8.8.30.43:4569]
VERSION          : 2
CALLING NUMBER   : 8590
CALLING NAME     : 4830258590
FORMAT           : 2
CAPABILITY       : 1550
USERNAME         : 8590
CALLED NUMBER    : 8580
DNID             : 8580

```

```

Tx-Frame Retry[-01] -- 0Seqno: 000 ISeqno: 001 Type: IAX      Subclass: ACK

```

```

Timestamp: 00003ms SCall: 00004 DCall: 26977 [8.8.30.43:4569]

```

```

Rx-Frame Retry[ No] -- 0Seqno: 001 ISeqno: 001 Type: IAX      Subclass:
AUTHREP

```

```

Timestamp: 00063ms SCall: 26977 DCall: 00004 [8.8.30.43:4569]
MD5 RESULT       : 57cc5c48affba14106c29439944413a1

```

```

Tx-Frame Retry[000] -- 0Seqno: 001 ISeqno: 002 Type: IAX      Subclass:
ACCEPT

```

```

Timestamp: 00054ms SCall: 00004 DCall: 26977 [8.8.30.43:4569]
FORMAT           : 1024

```

```

Tx-Frame Retry[000] -- 0Seqno: 002 ISeqno: 002 Type: CONTROL Subclass:
ANSWER

```

```

Timestamp: 00057ms SCall: 00004 DCall: 26977 [8.8.30.43:4569]

```

```

Tx-Frame Retry[000] -- 0Seqno: 003 ISeqno: 002 Type: VOICE   Subclass: 138

```

```

Timestamp: 00090ms SCall: 00004 DCall: 26977 [8.8.30.43:4569]

```

```

Rx-Frame Retry[ No] -- 0Seqno: 002 ISeqno: 002 Type: IAX      Subclass: ACK

```

```

Timestamp: 00054ms SCall: 26977 DCall: 00004 [8.8.30.43:4569]

```

```

Rx-Frame Retry[ No] -- 0Seqno: 002 ISeqno: 003 Type: IAX      Subclass: ACK

```

```

Timestamp: 00057ms SCall: 26977 DCall: 00004 [8.8.30.43:4569]

```

```

Rx-Frame Retry[ No] -- 0Seqno: 002 ISeqno: 004 Type: IAX      Subclass: ACK

```

```

Timestamp: 00090ms SCall: 26977 DCall: 00004 [8.8.30.43:4569]

```

```

Rx-Frame Retry[ No] -- 0Seqno: 002 ISeqno: 004 Type: VOICE   Subclass: 138

```

```

Timestamp: 00210ms SCall: 26977 DCall: 00004 [8.8.30.43:4569]

```

```

Tx-Frame Retry[-01] -- 0Seqno: 004 ISeqno: 003 Type: IAX      Subclass: ACK

```

```

Timestamp: 00210ms SCall: 00004 DCall: 26977 [8.8.30.43:4569]

```

```

Rx-Frame Retry[ No] -- 0Seqno: 003 ISeqno: 004 Type: IAX      Subclass: PING

```

```

Timestamp: 02083ms SCall: 26977 DCall: 00004 [8.8.30.43:4569]

```

```

Tx-Frame Retry[000] -- 0Seqno: 004 ISeqno: 004 Type: IAX      Subclass: PONG

```

```

Timestamp: 02083ms SCall: 00004 DCall: 26977 [8.8.30.43:4569]

```

```

RR_JITTER       : 0
RR_LOSS         : 0
RR_PKTS         : 1
RR_DELAY        : 40
RR_DROPPED      : 0
RR_OUTOFORDER   : 0

```

```

Rx-Frame Retry[ No] -- 0Seqno: 004 ISeqno: 005 Type: IAX      Subclass: ACK

```

```
Timestamp: 02083ms SCall: 26977 DCall: 00004 [8.8.30.43:4569]
Rx-Frame Retry[ No] -- OSeqno: 004 ISeqno: 005 Type: IAX Subclass:
HANGUP
Timestamp: 08693ms SCall: 26977 DCall: 00004 [8.8.30.43:4569]
CAUSE : Dumped Call
```

Para desconectar el debug use:

```
vtsvoffice*CLI>i ax2 no debug
```

6.11 Sumario

En este capítulo usted aprendió a diferenciar los puntos fuertes y flacos del IAX. Fue posible demostrar los escenarios de uso de Asterisk como cliente usando un softphone y entroncamiento de dos PBX con IAX trunked. Fue demostrado que el modo trunked economiza banda ancha enviando diversas llamadas en un mismo paquete evitando la creación de nuevos paquetes y encabezados. Fue posible ver que la cantidad de banda ancha usada depende de los codecs, como también de las configuraciones. Aprendió a configurar el archivo `iax.conf` para conectarse a teléfonos y proveedores y pudo usar algunos de los comandos de consola relacionados al Asterisk.

Exclusivo para Vicente Miguel

6.12 Cuestionario

1. Podemos citar como principales beneficios de IAX la economía de banda ancha y facilidad de pasar por Firewalls con NAT.

- Correcto
- Incorrecto

2. En el protocolo IAX los canales de señalización y media pasan separados. Esta afirmación es:

- Correcta
- Incorrecta

3. El IAX emplea los siguientes tipos de frames

- Frame Completo
- Frame Incompleto
- Mini-Frame
- Trunked Frame

4. La banda ancha usada por el protocolo IAX es la suma de la carga de voz (payload) más los encabezados (Marque todas las que se aplican)

- IP
- UDP
- IAX
- RTP
- cRTP

5. Es importante tener la misma configuración para el payload de codec (20 à 30 ms) que la sincronización de los frames en el modo trunked (20 ms – estándar)

- Verdadero
- Falso

6. Cuando el IAX es usado en modo trunk, apenas un encabezado es usado para transmitir múltiples llamadas. La afirmación de arriba es:

- Correcta
- Incorrecta

7. El protocolo IAX2 es el más común para conectar proveedores de telefonía IP, pues pasa fácil a través de NAT. La afirmación de arriba es:

- Correcta

Incorrecta

8. En un canal IAX como el de abajo, la opción <secret> puede ser tanto una contraseña como una _____.

```
IAX/[<user>[: <secret>]@]<peer>[: <portno>][ /<exten>[@<context>][ /<opti ons>]]
```

9. El contexto es adicionado para cada cliente IAX, esto permite que diferentes clientes posean diferentes contextos. Se puede pensar en un contexto como una clase de ramal donde el cliente será colocado. La afirmación es:

Correcta

Incorrecta

10. Es comando IAX2 show registry muestra informaciones sobre:

Los usuarios registrados

Los proveedores a los cuales el Asterisk se conecta.

Exclusivo para Vicente Miguel Perez Paredes

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

El protocolo SIP

7.1 Objetivos

Objetivos

Para el final de este capítulo usted estará apto para:

- Entender la teoría de operación SIP
- Entender las virtudes y debilidades de SIP
- Configurar Asterisk para conectarlo a un proveedor SIP
- Integrar dos Servidores Asterisk using SIP
- Configurar Asterisk en un escenario NAT
- Configurar un cliente detrás de NAT

7.2 Visión general

SIP (Sesión Initiated Protocol) es un protocolo basado en texto, similar al HTTP y SMTP, diseñado para iniciar, mantener y terminar sesiones de comunicación interactiva entre usuarios. Tales sesiones incluyen, voz, vídeo, chat, juegos interactivos y realidad virtual. Fue definido por la IETF y se viene tornando el estándar de facto en telefonía IP.

7.2.1 Teoría de Operación de SIP

El SIP es un protocolo de señalización de voz sobre IP que posee los siguientes componentes:

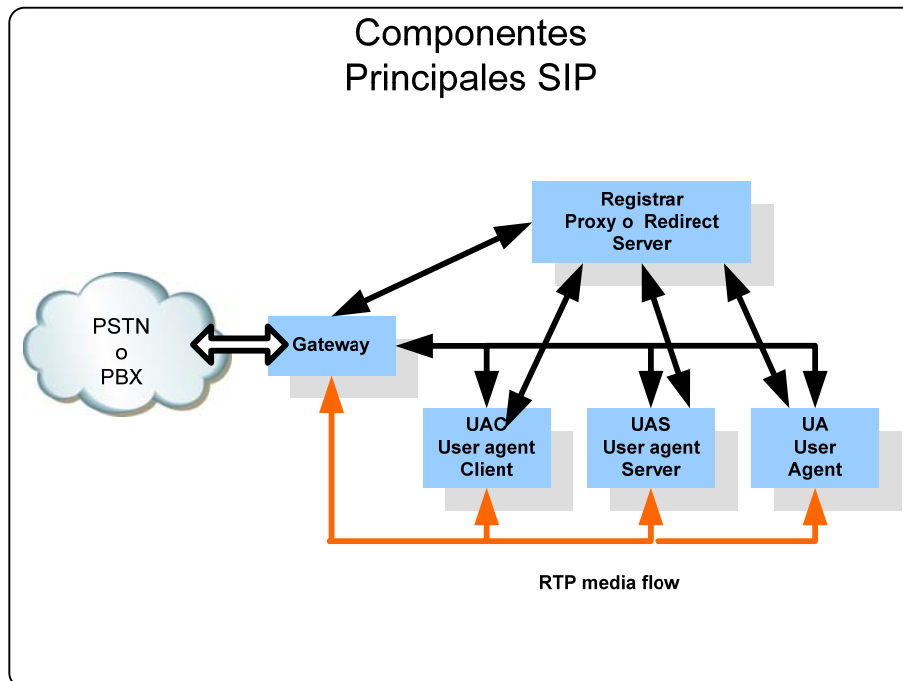


Figura 7.1 Componentes principales SIP

- **UAC (user agent client)** – cliente o terminal que inicia la señalización SIP.
- **UAS (user agent server)** – servidor que responde la señalización SIP de un UAC.
- **UA (user agent)** – terminal de red SIP (teléfonos SIP, o gateway para otras redes), contiene UAC y UAS.
- **Servidor Proxy** – Recibe pedidos de conexión de un UA y transfiere este para otro servidor proxy si la estación en particular no está en su administración.
- **Servidor de Redireccionamiento** – Recibe pedidos de conexión y los envía de vuelta al emisor incluyendo los datos de destino en vez de enviarlos directamente él a la parte llamada.
- **Servidor de localización** – recibe pedidos de registro de un UA y actualiza la base de datos de terminales con ellos.

Todas las sesiones del servidor (Proxy, Redirect y Location) están típicamente disponibles en una única máquina física llamada proxy server, que es responsable por la manutención de la base de datos de clientes,

establecimiento de conexiones, manutención y término y redireccionamiento de llamadas.

7.2.2 Proceso de Registro de SIP

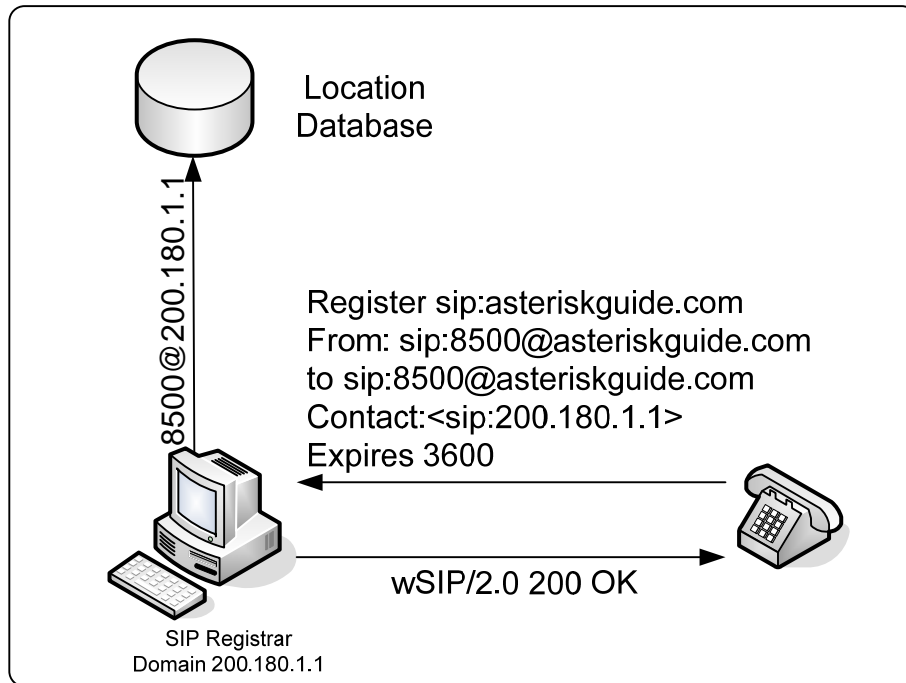


Figura 7.2 Proceso de registro SIP

Antes que un teléfono pueda recibir llamadas, este precisa registrarse en una base de localización. Es en este lugar donde el nombre será asociado a la dirección IP donde el teléfono se encuentra. En nuestro caso usamos como nombre el ramal 8500. Podría ser también una dirección con formato sip:flavio@voffice.com.br.

7.2.3 Operación de SIP en modo proxy.

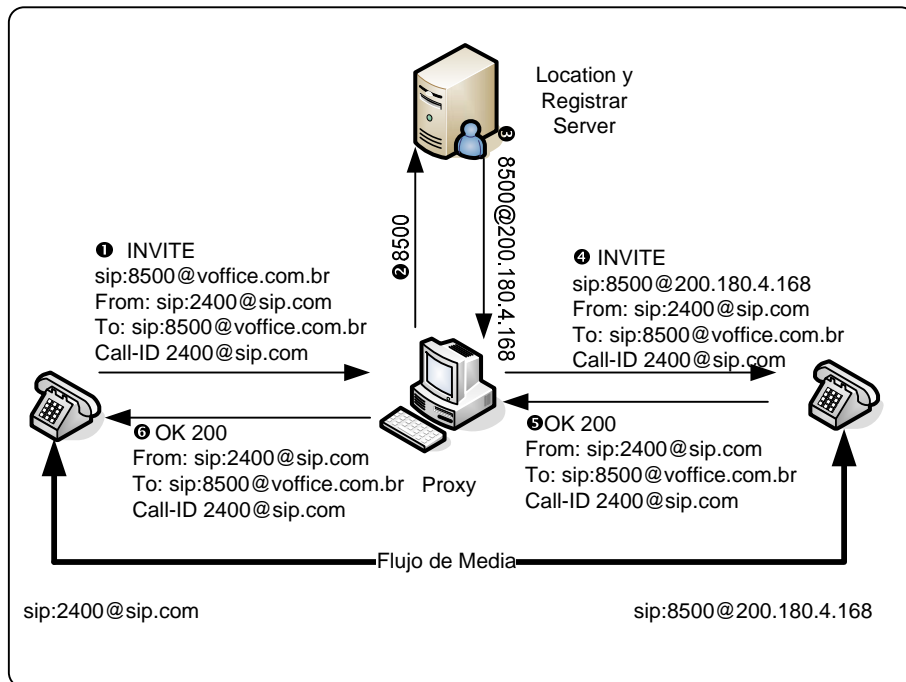


Figura 7.3 Operación en modo proxy

7.2.4 Operación en modo de redirect.

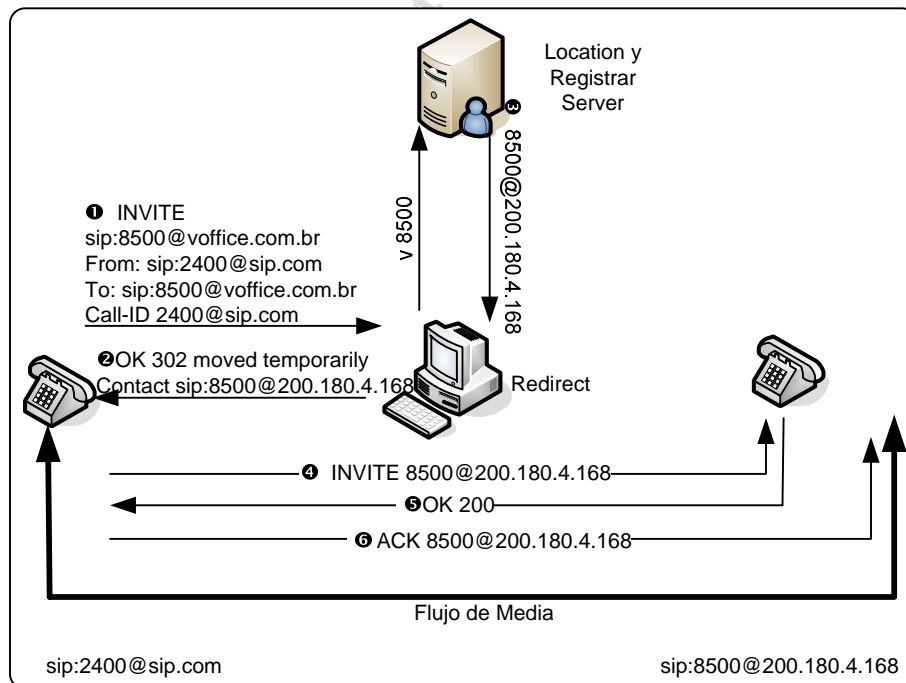


Figura 7.4 Operación en modo redirect

7.2.5 SIP en modo Asterisk

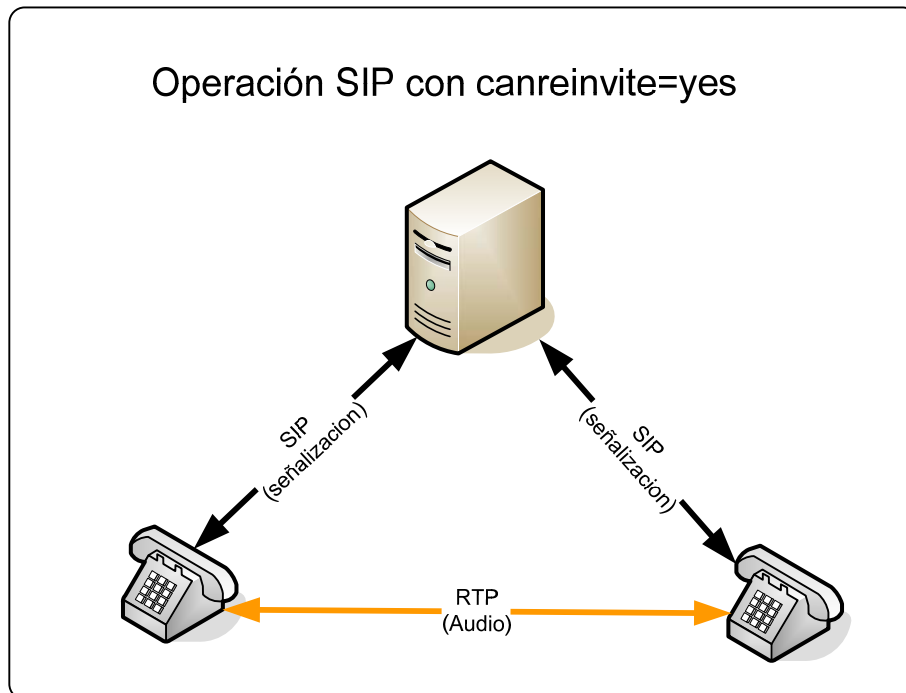


Figura 7.5 Operación en modo Asterisk con `canreinvite=yes` (`sip.conf`)

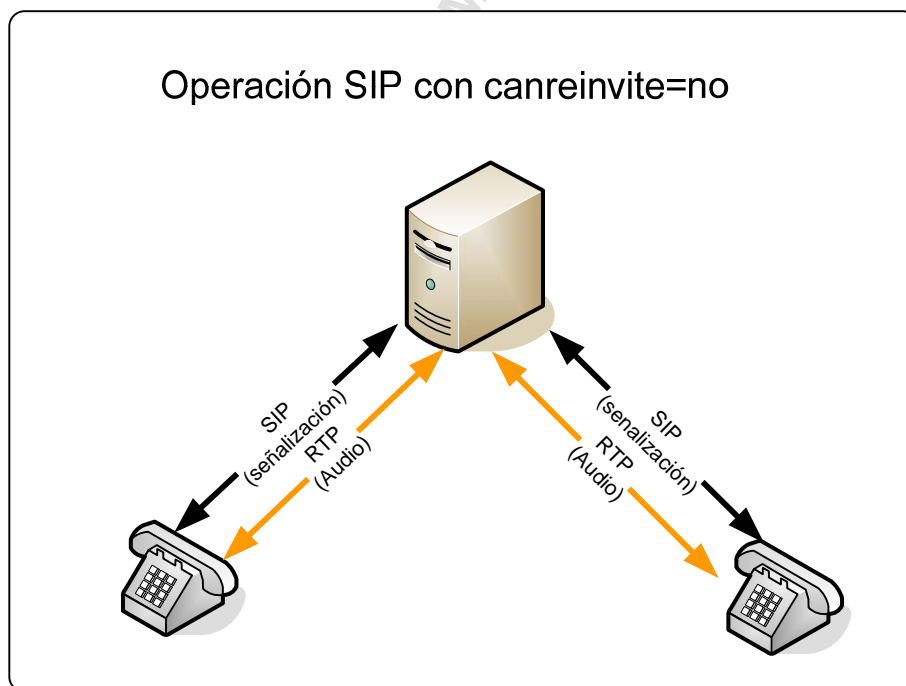


Figura 7.6 Operación en modo Asterisk con `canreinvite=no`

Es importante resaltar que Asterisk no es en sí un SIP Proxy ni un SIP Redirector. Asterisk es un Media Gateway. Él podría ser más bien descrito como un B2BUA, (back-to-back user agent). En otras palabras él conecta dos canales SIP como si fuesen canales de un PBX. Es posible usar en conjunto con Asterisk un SIP Proxy como es el SIP Express Router <http://www iptel.org/ser/>.

Los mensajes básicos enviados en un ambiente SIP son:

- INVITE – pedido de establecimiento de conexión.
- ACK – reconocimiento de INVITE por el receptor final del mensaje.
- BYE – término de la conexión.
- CANCEL – término de una conexión no establecida.
- REGISTER – registro de un UA en el SIP proxy.
- OPTIONS – pedido de opciones del servidor

Las respuestas a los mensajes de SIP son en formato texto como en el protocolo http. Aquí están las respuestas más importantes.

- 1XX – mensajes de información (100–intentando, 180–campanilla, 183–en progreso).
- 2XX – pedido completado con éxito (200 – OK).
- 3XX – encaminamiento de llamada, el pedido debe ser diseccionado para otro lugar. (302 – Moved temporarily movido, 305 – usa proxy).
- 4XX – error (403 – Prohibido).
- 5XX – error de servidor (500 – Error interno del servidor, 501 – No implementado).
- 6XX – falla global (606 – No aceptable).

7.3 Archivo de configuración sip.conf

Vamos a discutir la configuración del archivo sip.conf en varias partes. Vamos a mostrar como configurar los parámetros más genéricos de la sesión [general] y luego dentro de un contexto de uso vamos a ir presentando los otros parámetros.

Cada cliente de sip es identificado por un bloque de texto que se parece al ejemplo que se muestra aquí abajo:

```
[xxx]  
Type=xxx  
Parâmetro1=val or  
Parâmetro2=val or
```

Donde xxx es el nombre de usuario asociado con el cliente SIP, o es un nombre arbitrario usado por otros archivos de configuración para referirse, a este dispositivo SIP. Típicamente si un teléfono SIP tiene un número de extensión de 123, entonces su entrada correspondiente en este archivo irá a iniciarse con [123]. Note que usted todavía tiene que habilitar la extensión 123 en su plan de discado para alcanzar este teléfono.

La otra manera en que las llamadas SIP de entrada se inicien con las sesiones [xxx] de este archivo, es examinar la dirección IP de pedido que está llegando y ojear la sesión peer [xxx] que tenga el valor de host correspondiente. Si host=dynamic, entonces ninguna correspondencia será encontrada hasta que el cliente SIP este registrado.

7.3.1 Configuración de la sesión general [general]

Abajo sigue una explicación sobre los parámetros más comunes en la sesión [general] del archivo sip.conf.

```
bindaddr=0.0.0.0  
bindport=5060
```

Los parámetros "bindaddr" y "bindport" son opcionales y controlan el puerto y la interfase IP donde el Asterisk irá a recibir sus conexiones. Si usted configura bindaddr=0.0.0.0 el servidor recibirá conexiones en todas sus interfaces.

```
realm=voffi ce. com. br
```

Esta opción configura el parámetro "realm" para la computación del "digest".

```
srvookup=yes
```

Este parámetro habilita el uso de registros SRV en el DNS. Asterisk usa apenas el primer registro retornado. Para acceder servidores por el nombre de dominio usted precisará colocar este parámetro en "yes".

```
useragent=Asterisk PBX 1.2.7
```

Este comando altera el campo useragent de encabezado SIP identificando el tipo de agente usuario usado.

```
tos=lowdelay
```

Los parámetros lowdelay, throughput, reliability, mincost y none están obsoletos porque setean el tipo de servicio de acuerdo con el antiguo modelo de precedencia definidos en las RFC 791 y RFC 1349.

Los parámetros TOS setean bytes correspondientes al tipo de servicio en el encabezado del paquete IP. Este campo es responsable de informar a los ruteadores como se debe tratar estos paquetes en la red y aplicar QoS (calidad de servicio). Este campo ya sufrió varias modificaciones. Más recientemente la nomenclatura conocida como DSCP (Differentiated Services Code Point RFC 2474 y RFC 2475) ha sido la más usada. Usando valores numéricos (decimales) usted puede setear cualquier valor en el sistema antiguo o nuevo.

Para simplificar los valores más usados para los paquetes de voz son:

Precedencia 5 (RFC 791, obsoleta)

```
tos=160
```

DSCP EF – Express Forwarding (RFC 2474, sobrepone a la anterior)

```
tos=160
```

Como usted puede notar existe una compatibilidad de la nueva RFC con la antigua. Si usted setea auto-qos voip en un switch Cisco (que soporte el recurso) el automáticamente priorizará paquetes con este marcado en los encabezados.

Existe una excelente explicación de los valores de tos en: <http://www.cisco.com/warp/public/105/dscpvalues.html>

7.4 Configurando un cliente SIP

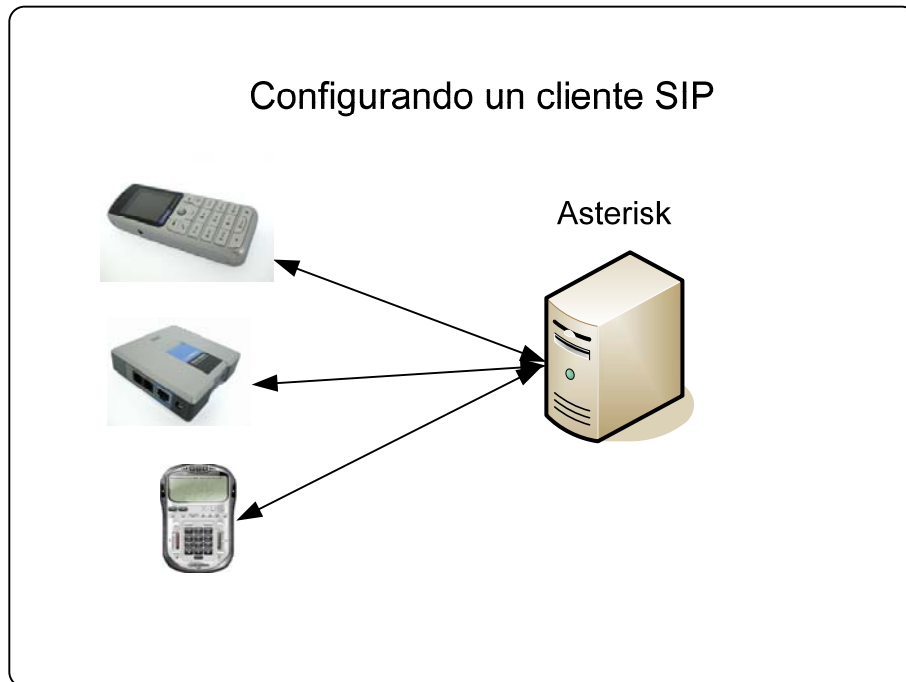


Figura 7.7 Configurando un cliente SIP

Configurando un cliente SIP con el SIP Server: Esto significa que clientes SIP (teléfonos, softones) se registran para el servidor Asterisk y configuran sesiones SIP con el servidor, llamadas y respuestas a llamadas.

- Paso 1 – Configurar una entrada en el archivo sip.conf para cada teléfono.
- Paso 2 – Configurar el teléfono.
- Paso 3 – Configurar la extensión en el plan de discado.

7.4.1 Paso 1 - Configurando el sip.conf

Teléfono Típico

```
[4101]
type=friend
context=default
callerid=Flávio E. Gonçalves<8550>
host=dynamic
canreinvite=yes
dtmfmode=info
mailbox=1234@default
display=alil
allow=ulaw
allow=g729
secret=contraseña
```

```
deny=0.0.0.0/0.0.0.0
permi t=192.168.1.0/255.255.255.0
musi ccl ass=default t
```

7.4.2 Paso 2 Configurando el teléfono

Siga las instrucciones de su teléfono favorito para configurarlo. Recuerde que nombre de usuario, contraseña, contexto y codec son los parámetros más importantes.

Ejemplo 1: Softfone X-Lite

Ejemplo 2: Linksys PAP2

7.4.3 Paso 3 - Definir la extensión en el archivo sip.conf

Después que usted defina las cuentas de clientes sip en sip.conf, usted está apto a loguearse en el servidor Asterisk dos clientes y hacer y recibir llamadas. Usted precisará configurar las extensiones en extensions.conf.

Ejemplo:

```
exten=> _41XX,1, Dial(SIP/${EXTEN},10,t)
```

Si alguien llama a una extensión que empieza con 41 y dos dígitos más, el cliente sip logueado será discado de forma de recibir la llamada.

7.4.4 Opciones de configuración para los teléfonos

```
type=fri end
```

Siempre es usada para teléfonos que hacen y reciben llamadas.

```
context=default t
```

Es el contexto usado para recibir llamadas de este teléfono. Este teléfono solo podrá llamar a números incluidos. En este contexto de plan de discado.

```
calleri d=Fl avio E. Goncal ves <8550>
```

Usted puede configurar un identificador de llamadas para cada teléfono lo que puede ser muy útil a la hora de leer los registros de billing y en la identificación del que llama en los displays de los teléfonos.

```
host=dynami c
```

Para teléfonos usted probablemente definirá el host como dinámico. Esto permite que el usuario se registre informando su dirección IP actual.

Usted puede informar el parámetro " defaultip=a.b.c.d" si desea que sea posible contactar un mismo teléfono antes de registrarse. Host=a.b.c.d es usado normalmente apenas para conexión a proveedores.

```
canrei nvi te=yes
```

El protocolo SIP intenta conectar los puntos finales directamente. Entretanto si Asterisk quisiera detectar el DTMF por ejemplo este deberá estar en el medio del flujo RTP entre los user agents. Varios recursos como transferencia funcionan apenas con canreinvite=no.

```
dtmfmode=i nfo
```

El dtmfmode es esencial para que el teléfono pase el DTMF necesario a operaciones como transferencia, captura de llamada entre otras cosas. Este puede ser seteado como uno de los siguientes parámetros:

dtmfmode=info. Cuando es configurado de esta forma las informaciones de DTMF son pasadas por la señalización SIP. Este método es definido por la RFC 3265 (eventos de señalización). Esta RFC define varias formas de pasajes de eventos por la señalización SIP como DTMF y MWI (Indicador de mensajes en espera) por ejemplo.

dtmfmode=inband. Cuando es configurado de esta forma el DTMF pasa como una señal de audio normal. Solamente funciona adecuadamente con CODECs sin compresión como el G711 (ulaw o alaw).

dtmfmode=rfc2833. Este rfc permite que los tonos DTMF sean pasados de parte a parte usando el protocolo RTP en la forma de eventos nombrados. Varias señales como DTMF, FAX, Línea entre otros. Abajo esta la tabla de eventos relacionados a DTMF.

Codificación do evento(decimal)

0--9	0--9
*	10
#	11
A--D	12--15
Flash	16

Tabla 1 - DTMF named events

```
mai l box=1234@defaul t
```

El comando mailbox permite que usted asocie una cuenta de buzón de voz al teléfono. Cuando esto es hecho el buzón de voz envía a los teléfonos (agentes usuario de una forma más genérica) un evento MWI de

mensajes en espera. En muchos teléfonos esto va hacer que se encienda una luz avisando al usuario de que un mensaje en esta en el buzón.

```
disallow=all
```

El comando `disallow=all` deshabilita todos los codecs. Este es normalmente seguido por la autorización de los codecs necesarios (Ex. `allow=ulaw`). En general es importante controlar que codecs están en uso limitándolos apenas a los necesarios.

```
secret=contraseña
```

Este parámetro configura la contraseña para la autenticación. Si usted no quisiera que las contraseñas aparezcan en el archivo de texto puede usar el comando **`md5secret="contraseña criptográfica en MD5"`**.

Para generar la contraseña en MD5 usted puede usar el comando de línea:

```
#echo -n "username: realm: secret" | md5sum
```

Use entonces el resultado en un comando similar a:

```
md5secret=0b0e5d467890...
```

No olvide el parámetro `-n` sino el retorno de carro `"\n"` será incluido en el cálculo de MD5.

```
deny=0.0.0.0/0.0.0.0
permit=192.168.1.0/255.255.255.0
```

Los comandos de arriba van a negar todas las direcciones IP y permitir el registro de teléfonos solo de la red local `192.168.1.0/24`.

```
musicclass=default
```

El parámetro de arriba permite establecer que directorio de música MP3 será ejecutado para música en espera. Las clases son definidas en el archivo `musiconhold.conf`. Tenga **cuidado** en el hecho que en la definición de los peers este parámetro puede estar como `musiconhold` y no en `musicclass` como en ejemplo del archivo `sip.conf`. Ver bug:

<http://bugs.digium.com/view.php?id=5521>

```
videosupport=yes/no
```

Este parámetro habilita llamadas entre dos usuarios con capacidad de vídeo.

```
language=br
```

El código de lenguaje definido en `indications.conf` – Define el lenguaje para los “prompts” y señales locales de telefonía.

```
rtptimeout=60
```

Termina las llamadas si no se tuviese actividad RTP por x segundos, cuando no estuviera en espera (hold).

```
rtpholdtimeout=120
```

Termina la llamada si no hubiera actividad RTP cuando esta en espera (hold) (debe ser mayor que el `rtptimeout`).

```
qualify=yes|no o tiempo en milisegundos
```

Envía periódicamente un mensaje para medir el tiempo de respuesta de peer. Cuando sea mayor que 2000 ms o el tiempo es especificado en el comando, considera al peer fuera de alcance.

7.4.5 Limitaciones y cuidados con clientes SIP

Asterisk no soporta llamadas SIP sobre TCP o TLS, apenas las soporta sobre UDP.

Asterisk usa la entrada de un flujo RTP como fuente de sincronización de tiempo para enviar el flujo de salida. Si el flujo de entrada es interrumpido debido a la supresión de silencio entonces la música en espera tendrá cortes. En resumen, **usted no puede usar supresión de silencio en teléfonos SIP.**

7.5 Conectándose a un proveedor SIP.

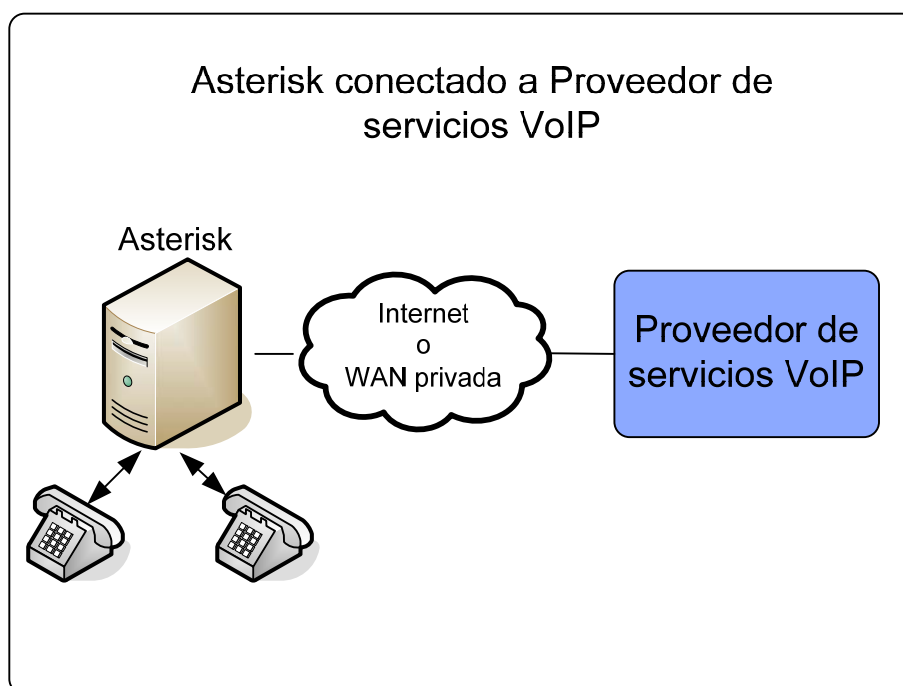


Figura 7.8 Asterisk conectado a proveedor de servicios VoIP

SIP client: Esto significa que Asterisk se registra como un cliente para otro servidor SIP y recibe y coloca llamadas para este servidor. La recepción de llamadas es ruteada para una extensión de Asterisk.

Para configurar un proveedor SIP son necesarios tres pasos:

- Paso 1 – Dentro del archivo sip.conf, en la sesión [general] colocar una línea de registro en el proveedor SIP.
- Paso 2 – Crear una entrada de tipo [peer] para el proveedor de modo de simplificar el discado.
- Paso 3 – Colocar una ruta de salida en el plan de discado.

7.5.1 Paso 1: Registrar el proveedor (sip.conf)

Esto va a permitir que el proveedor localice su Asterisk. En esta instrucción usted está diciéndole que quiere recibir cualquier llamada del primer proveedor en la extensión 4100 y del segundo proveedor en la extensión 8573. En la sesión [general] coloque LAS LÍNEAS DE ABAJO.

```
regi ster=>621538: password@fwd. fwdnet. net/4100
regi ster=>i p1140623535: password@gvt. com. br/8573
```

7.5.2 - Paso 2: Configurar el [peer] (sip.conf)

Crear una entrada del tipo [peer] para el proveedor de modo de simplificar el discado (sip.conf). Observe la línea "insecure=very" necesaria si usted quiere recibir llamadas del FWD. Si usted no colocase esta línea, Asterisk mandara para el proveedor un pedido de contraseña (Challenge). Como su proveedor no tiene cuenta en su Asterisk, la llamada sería rechazada. Esto sucede con otros proveedores como por ejemplo GVT. Abajo mostramos un ejemplo funcional con a GVT:

```
[gvt]
context=entrada
type=friend
callerid="i p1140623535" <1140623535>
dtmfmode=inband
canreinvite=no
username=ip1140623535
secret=[omitido por seguridad]
host=gvt.com.br
fromuser=ip1140623535
fromdomain=gvt.com.br
insecure=invite
```

7.5.3 Paso 3: Crear una ruta de salida en el plan de discado.

En este ejemplo, vamos escoger el digito 010 como ruta de salida para el FWD. Para discar para el 610000, usted debe discar 010610000. (Como si fuese una nueva operadora, "marque '10' para llamar el fwd").

```
exten=>_010., 1, SetCIDNum(621538)
exten=>_010., 2, SetCIDName(Filavio Goncalves)
exten=>_010., 3, Dial(SIP/${EXTEN:3}@gvt)
exten=>_010., 4, Playback(invalid)
exten=>_010., 5, Hangup
```

7.5.4 Opciones de configuración específicas para proveedores

Abajo vamos a entrar en los detalles de parámetros específicos en la conexión a proveedores.

```
register=>621538:password@fwd.fwdnet.net/4100
```

El comando register en la sesión [general] del archivo sip.conf es usado para hacer el registro en el proveedor. En este es informado el nombre y contraseña del usuario, bien como el proveedor y el puerto UDP. Además de eso en el final después de "/" es colocado el ramal para recibir llamados de este proveedor. Técnicamente hablando este ramal es colocado en el campo "Contact" del paquete SIP de forma que el proveedor sepa como llamar al Asterisk. Por default este valor es la extensión "s" del contexto definido en la sesión general.

Los parámetros de registro pueden ser alterados en la sesión global usando:

```
regi sterti meout=20
```

Intenta nuevamente registrarse cada 20 segundos (estandar)

```
regi steratti enepts=10
```

Número de tentativas de registro antes de desistir. Colocando este parámetro en cero hará que Asterisk intente indefinidamente siempre hasta que este acepte el registro.

```
CLI>si p show registry
```

El comando SIP show registry permite que usted vea si su Asterisk se registró correctamente a su proveedor.

```
username=i p1140623535
```

Es el nombre usado en la creación del "digest" de autenticación. El "digest" es un valor computado a partir del username, secret y realm (dominio).

```
host=gvt.com.br
```

Host define el host a ser contactado.

```
fromuser=i p1140623535  
fromdomain=gvt.com.br
```

Fromuser y Fromdomain son parámetros a veces necesarios para autenticación en proveedores de voz sobre IP. Ellos son colocados en el campo "From" del paquete SIP.

```
i nsecure=very
```

Cuando usted se conecta a un proveedor de voz sobre IP y hace una llamada, el proveedor pide de usted sus credenciales. En el caso inverso, cuando el proveedor encamina una llamada para Asterisk, este pide autenticación del proveedor enviando un mensaje con código 407 (Proxy Authentication Required). Es simple de entender que un proveedor no podrá tener la contraseña de acceso para todos sus clientes Asterisk. Este comando es usado para evitar que Asterisk envíe el paquete (407 Proxy Authentication Required).

Si usted usa insecure=invite Asterisk no pedirá autenticación en las llamadas que llegan del proveedor. Si usa insecure=port, Asterisk

encontrará el [peer] por la dirección IP no importando el puerto. Usted puede también combinar los dos con "insecure=invite;port".

7.6 Autenticación de las llamadas SIP entrantes.

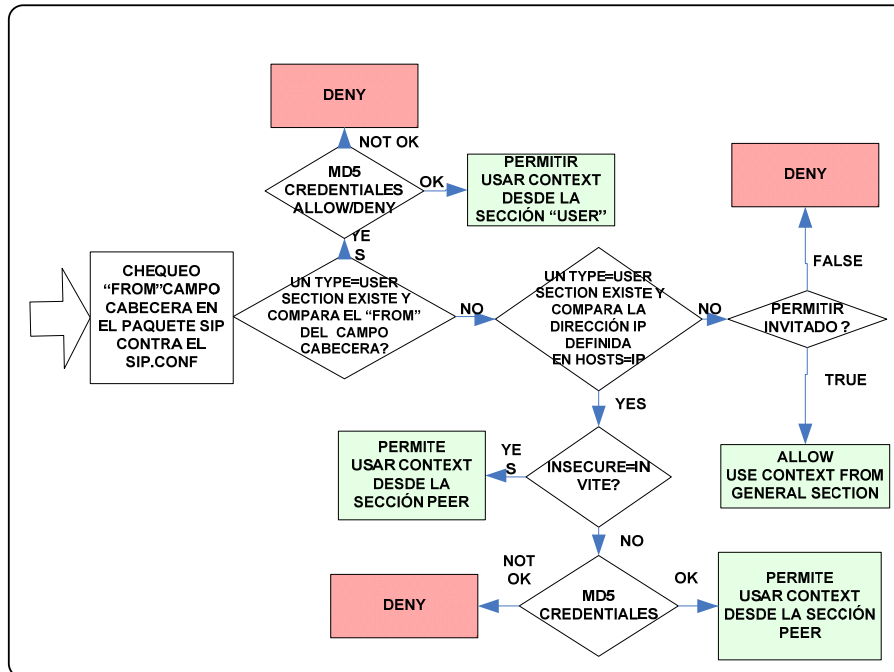


Figura 7.9 Autenticación de las llamadas SIP

Cuando Asterisk recibe una llamada SIP entrante en el módulo de canal SIP. Él sigue el flujo mostrado en la figura de arriba:

Tres parámetros de la sesión [general] que regulan la autenticación son:

`allowguest=yes/no`

Este comando controla si un usuario sin un "peer" correspondiente se puede autenticar sin un nombre y contraseña. Puede parecer extraño, pero la filosofía de protocolo SIP sería tener una red mundialmente direccionable que a través de DNS permitiese que cualquier agente-usuario SIP pudiese alcanzar cualquier otro agente sin necesidad de autenticación (Si así fuese no serían más necesarias operadoras de telefonía convencional o VoIP).

`insecure=invite;port`

Ya se discutía mas arriba, impide que Asterisk genere un mensaje " 407 Proxy Authentication Required".

`autocreatepeer=yes/no`

Este es uno de los comandos más difíciles de entenderse y es usado básicamente cuando Asterisk es integrado a un servidor SIP Proxy (Como SER por ejemplo).

Si esta opción estuviese habilitada, cualquier usuario-agente SIP se podrá registrar con su Asterisk PBX como un peer. Esta configuración de el "peer" estará basada en las opciones globales. El nombre de el [peer] estará basado en la parte usuario del campo "Contact" de la URL.

Cuidado: Por una cuestión de seguridad es muy importante colocar el contexto de la sesión general para un contexto que no pueda comunicarse para a red pública, o un usuario no autenticado podrá hacer llamadas externas sin ninguna restricción y causar un perjuicio considerable en su factura.

7.7 Nomenclatura de los canales SIP

El formato de el nombre de un canal SIP usado para una conexión de salida es:

El parámetro identificador puede ser hecho de tres partes.

```
SIP/[exten@]peer[:portno]
```

peer: El nombre de el peer al cual se conecta. Esto puede ser uno de los siguientes:

- Un "peer" o "friend" definido en sip.conf.
- Una dirección IP.
- Un nombre de dominio. Para nombres de dominio Asterisk primero revisará en los registros DNS SRV para aquel dominio.

portno: Es el puerto UDP a ser usada. Si es omitido Asterisk usará 5060.

exten: Si es definido, entonces el asterisk irá a pedir al "peer" que se conecte a la extensión "exten"

Ejemplos:

```
exten=>s, 1, Dial (SIP/iphone)
exten=>s, 1, Dial (SIP/info@voffice.com.br)
exten=>s, 1, Dial (SIP/192.168.1.8:5060, 20)
exten=>s, 1, Dial (SIP/8500@sip.com:9876)
```

7.8 SIP NAT Traversal

La traducción de direcciones IP (NAT) ha sido usada por la mayoría de los proveedores de servicio y empresas como una manera de contornear los problemas de la falta de direccionamiento IP. Normalmente las empresas reciben un pequeño bloque de direcciones IP que varía normalmente de 1 a 256 direcciones "válidas". Ya los usuarios domésticos reciben una dirección válida dinámica en sus ruteadores y usan direcciones inválidas atrás de estos ruteadores.

El NAT resuelve este problema mapeando las direcciones internas para direcciones públicas externas. Una dirección IP:Puerta interna es mapeada para una dirección IP:Puerta externa. Con este mapeamiento el ruteador sabe como encaminar de vuelta un paquete que ha venido de la red externa. Este mapeamiento es valido por un tiempo pré-determinado, después el cual en la ausencia de tráfico es descartado.

Existen cuatro tipos de NAT definidos como

- Full Cone
- Restricted Cone
- Port Restricted Cone
- Symmetric

Para una dada dirección interna, los tres primeros tipos de NAT mantienen un mapeamiento de su dirección interna que es independiente de la dirección de destino siendo visto. El cuarto tipo de NAT irá a localizar un nuevo mapeamiento para cada dirección de destino independiente. A menos que haya una tabla de mapeamiento estático. El mapeamiento que se abre cuando el primer paquete es enviado de un cliente a través de NAT puede ser válido apenas por cierta cantidad de tiempo, (típicamente algunos minutos), a menos que los paquetes continúen, siendo enviados y recibidos en un puerto IP.

7.8.1 Full Cone (Clon Completo)

En el caso de "Full Cone", el mapeamiento es bien establecido y cualquiera desde ip publica que quiera alcanzar un cliente atrás de NAT, precisará apenas saber el esquema de mapeamiento de forma a mandar paquetes para este.

Por ejemplo:

Un computador atrás de NAT con IP 10.0.0.1 enviando y recibiendo en el puerto 8000 es mapeado para la puerta externa IP en el NAT de

200.180.4.168:1234. Cualquiera desde Internet puede enviar paquetes para esta dirección y puerto IP y estos paquetes serán pasados para el cliente en la máquina de dirección ip 10.0.0.1:8000.

Es el caso de Firewalls sin control de sesión. Normalmente implementado a través de filtrado de paquetes y es el tipo más inseguro de Firewall y cada vez menos común en los días de hoy.

7.8.2 Restricted Cone (Cono Restrito)

En el caso de cono restringido, el par IP/Puerto externo solamente es abierto una vez que el computador interno envíe datos para la dirección de destino IP específico. Por ejemplo:

En el caso donde el cliente envía un paquete para un computador externo, el NAT mapea al cliente 10.0.0.1:8000 para 200.180.4.168:1234. Así el computador externo puede enviar paquetes de vuelta. Entretanto el NAT bloqueará paquetes que llegan de otros computadores externos.

Note que en este caso el Firewall tiene control sobre la sesión, esperando paquetes pertenecientes a una sesión, pero una vez abierto, aquel computador puede iniciar cualquier sesión independiente del puerto (200.210.1.1:3000, 200.210.1.1:3001...).

7.8.3 Port Restricted Cone (Cono restringido por puerto)

Un NAT de tipo "port restricted" es casi idéntico al "Restricted Cone", pero en este caso el NAT bloqueará todos los paquetes a menos que el cliente tenga enviado previamente un paquete para la IP y puerto que está enviando para el NAT. De esta forma, si el cliente enviara para un computador externo para el puerto 1010, el NAT apenas permitirá paquetes de vuelta si estos viniesen de 200.180.4.168 en el puerto 1010.

En este caso el Firewall tiene un control mayor de sesión, solo permitiendo que paquetes pertenecientes a aquella sesión puedan retornar, al final de la sesión, si el computador de destino resuelve enviar paquetes de un puerto diferente (200.210.1.1:10000) estos no serán aceptados.

7.8.4 Simétrico

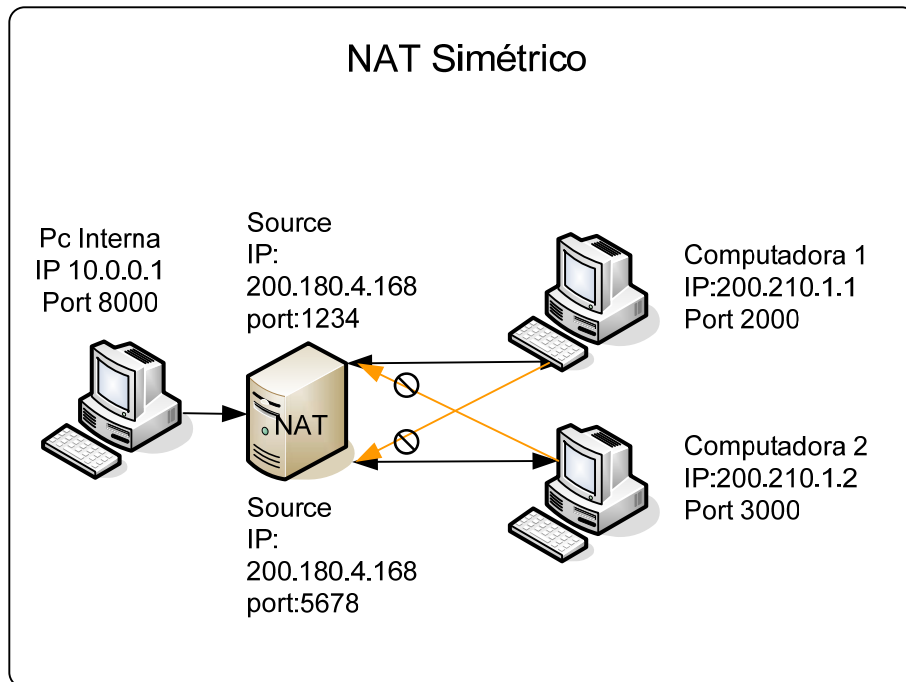


Figura 7.10 NAT Simétrico

El último tipo de NAT es el simétrico, este es diferente de los tres primeros. Un mapeamiento específico de IP:Puerto para un NAT público IP:Puerto es dependiente de la dirección de destino para el cual el paquete es enviado. Entonces por ejemplo, si un cliente envía de 10.0.0.1:8000 para el computador B, este puede ser mapeado como 200.180.4.168:1234. Si enviara del mismo puerto para una dirección IP diferente, este es mapeado de forma diferente 200.180.4.168:5678. El computador A y B pueden responder apenas para este mapeamiento. Si cualquiera intentara enviar para otro puerto mapeado, estos paquetes son descartados como en el caso de Cono Restringido. El par externo IP:puerto es abierto solo cuando el computador interno envía datos para un destino específico.

7.8.5 Resumen de los tipos de Firewall

	Precisa enviar datos para abrir a entrada	Puerto IP bien determinado para retorno	Restringe la entrada a la dirección IP:Puerto de Destino
Full Cone	No	Si	No
Restricted Cone	Si	Si	Solo la IP
Port Restricted Cone	Si	Si	Si
Simétrico	Si	No	Si

7.9 NAT en el pasaje de la señalización SIP

Existen dos partes de una llamada basada en SIP. La primera es la señalización, que es un protocolo de mensajes para establecer una

llamada. La segunda es realmente el flujo de media. Los paquetes de RTP viajan directamente entre los dispositivos finales.

La señalización SIP puede atravesar el NAT de una forma bastante directa, desde que exista un proxy, a un salto de distancia de NAT, que reciba los mensajes SIP del cliente (a través de NAT) y entonces retorne los mensajes para el mismo lugar. El proxy precisa retornar los paquetes de SIP para el mismo puerto de donde este recibió los paquetes (En el puerto SIP 5060). El SIP tiene etiquetas (tags) que le dicen al proxy para hacer esto – La etiqueta recibida dice al proxy para retornar un paquete para una IP específica y la etiqueta "rport" guarda el puerto a donde retornar. La mayoría de los proxys todavía no implementan la etiqueta "rport", y algunos clientes no van a procesar los mensajes SIP correctamente.

Si estas etiquetas estuviesen presentes, por lo menos en principio el mecanismo existe para atravesar el NAT. Otro modo simple de atravesar el NAT es usar TCP para la señalización SIP entre el cliente y el proxy. Desde que la conexión TCP es abierta a través de NAT directamente de el cliente para el proxy. La señalización procederá sin bloqueo. Nuevamente, muchos proxys todavía no implementarán todavía la opción TCP y trabajan apenas usando UDP. Note que la señalización SIP debería estar apta a atravesar cualquiera de los cuatro tipos de NAT si el proxy retorne los mensajes de SIP en el mismo puerto fuente que él recibió el mensaje inicial. El mensaje inicial SIP, enviado para el proxy IP:Port, abre el mapeamiento de NAT, y el proxy retorna los paquetes de NAT para el mismo IP:Puerto. Esto es permitido en cualquier escenario de NAT.

Registrar un cliente que está atrás de un NAT requiere o un **Registrar** que pueda salvar el IP:Puerto en la información de registro basada en el puerto e IP que la ve como fuente de mensajes SIP o un cliente que sepa de su dirección mapeada externamente y puerto y pueda así insertarlo en la información de contacto como IP:Puerto de forma de recibir los mensajes SIP. Es preciso tener cuidado en usar un intervalo de registro menor que el "keepalive" para el mapeamiento de NAT.

7.10 NAT en el flujo de media RTP

El RTP para atravesar un NAT no tiene una solución tan fácil como la señalización SIP. En este caso de RTP, el cuerpo del mensaje SIP contiene informaciones sobre los puntos finales, necesarias para permitir la comunicación de uno con el otro. Esta información es contenida en el mensaje SDP. Los dispositivos rellenan esta información de acuerdo con lo que ellos saben sobre sí mismos.

Un cliente situado atrás de un NAT conoce apenas su puerto interno IP:Puerto y es esto lo que él coloca en el cuerpo SDP del mensaje SIP. Cuando el punto de destino final quiere enviar paquetes para el punto

originador, el usará la información SDP recibida conteniendo la dirección IP interna del originador y los paquetes nunca van a llegar allá. Aquí va un ejemplo de un "trace" de un mensaje INVITE de un cliente SIP atrás de un NAT como recibida por el gateway. Existe un proxy en el medio del camino.

```

001 INVITE sip:12125551212@211.123.66.222 SIP/2.0
002 Via: SIP/2.0/UDP 211.123.66.223:5060;branch=a71b6d57-507c77f2
003 Via: SIP/2.0/UDP 10.0.0.1:5060;received=202.123.211.25;rport=12345
004 From: <sip:2125551000@211.123.66.223>;tag=108bcd14
005 To: sip:12125551212@211.123.66.222
006 Contact: sip:2125551000@10.0.0.1
007 Call-ID: 4c88fd1e-62bb-4abf-b620-a75659435b76@10.3.19.6
008 CSeq: 703141 INVITE
009 Content-Length: 138
010 Content-Type: application/sdp
011 User-Agent: HearMe SoftPHONE
012
013 v=0
014 o=del tathree 0 0 IN IP4 10.0.0.1
015 s=del tathree
016 c=IN IP4 10.0.0.1
017 t=0 0
018 m=audio 8000 RTP/AVP 4
019 a=ptime:90
020 a=x-ssrc:00aea3c0

```

En el trace de arriba, la Dirección IP en la línea 003 de encabezado SIP es la dirección IP donde el cliente piensa que está (10.0.0.1). Pero el Proxy sabe que dirección IP el realmente recibió el paquete. Entonces el adicionará las etiquetas "received" y "rport" con la dirección IP y el puerto después el mapeamiento de NAT. Estas etiquetas permiten al proxy encaminar los mensajes SIP de vuelta al cliente vía NAT.

Más información que es usada de forma de pasar los datos de voz (A conexión RTP) es mantenida más abajo en el mensaje de las líneas 014 e 016. El cliente espera recibir en el puerto 8000 (m=) en la IP 10.0.0.1 (c=), que es puerto que el ve propio, y como existe un segundo punto final retornará los paquetes. El resultado es que una vez que a llamada este establecida (La señalización SIP pasa) el audio no es recibido.

Si el cliente estuviese atrás de uno de los tres primeros tipos de NAT, entonces la solución de atravesar el NAT es simple. El cliente debe descubrir como su IP:Puerto aparece para el mundo y entonces debe colocar esta información en los mensajes SDP en vez de la información de su IP:Puerto interno. Existen dos métodos para un cliente de determinar la dirección públicamente mapeado para el IP:Puerto. el primero es preguntar al ruteador con NAT, el segundo es preguntar a alguien fuera de NAT en la red pública.

7.11 Formas de pasajes por el NAT

Existen innumerables mecanismos creados para el pasaje por el NAT. La mayoría funciona para los NATs de tipo Full Cone, Restricted Cone y Port Restricted Cone, entretanto apenas el RTP Relay funciona para los NATs de tipo simétrico. Felizmente Asterisk puede actuar como un RTP Relay usando la opción "canreinvite=no" para aquella extensión en el archivo sip.conf.

Podemos dividir los métodos de pasaje de NAT en Near-End-Nat Traversal (Soluciones en los clientes) y Far-End-Nat-Traversal (Soluciones en el servidor).

Soluciones Near-End-Nat-Traversal

- UPnP
- ALG
- STUN
- Configuración Manual
- ICE

Soluciones Far-End-Nat-Traversal

- Comedia (Conexión Oriented Media)
- TURN – Traversal of UDP using Relay NAT

7.11.1 UPnP

Un cliente puede preguntar al NAT como es que está mapeado para un par IP:Puerto a través de un protocolo llamado Universal Plug and Play. Esta es una solución que está siendo promovida por Microsoft (entre otros). El cliente pregunta al NAT vía UPnP que mapeamiento el debe usar si el quisiera recibir en el puerto x. NAT responde con el par IP:Puerto que alguien en la red pública debería usar para alcanzar el cliente en aquel puerto. Muchos fabricantes de dispositivos NAT ya incluyen UPnP en sus productos. Un problema es que el UPnP no va a funcionar en el caso de NATs cascadeados.

7.11.2 STUN - Simple Traversal of UDP NAT

En la ausencia de un mecanismo para comunicarse con el dispositivo NAT, el mejor medio para que el cliente determine su par IP:Puerto externo es preguntar al servidor situado en la Internet Pública como el ve su dirección. En este escenario existe un servidor que permanece esperando estos paquetes. Cuando el recibe un paquete el retorna un mensaje del mismo puerto para la fuente del paquete recibido conteniendo el par IP:puerto que el ve en el encabezado del paquete enviado. En todos los casos (todos los 4 casos de NAT), el cliente irá a recibir un paquete de retorno. El cliente entonces va a determinar:

1. Si el está atrás de un NAT (El par IP:Puerto contenido es diferente del par IP:Puerto que el piensa que está)
2. Cual par IP:Puerto publico él debería usar para colocar en el mensaje SDP de forma que el punto final lo alcance

Por ejemplo, si el cliente quiere ser alcanzado en 10.0.0.1:8000, el irá primero va enviar una consulta a NAT por el puerto 8000. NAT irá realmente a recibir una consulta del paquete 200.180.4.168:1234 y así va a responder para el par IP:puerto con el paquete contenido 200.180.4.168:1234. El cliente entonces podrá colocar esto en su SDP "m=AUDIO 1234" y "c=200.180.4.168", el cliente continua escuchando en el puerto 10.0.0.1:8000.

Esto irá a funcionar en las siguientes situaciones:

1. El cliente debe enviar y recibir el RTP en el mismo puerto.
2. El cliente debe enviar un mensaje SIP logo después de enviar la consulta para la prueba de NAT. Si existe un largo atraso NAT puede tener un timeout.
3. En el caso de Restricted Cone y Port Restricted Cone, el cliente debe enviar el paquetes para el punto final antes que el NAT permita paquetes del punto final para el cliente. Esto no va a funcionar en el caso de NAT simétrico, porque la dirección de la prueba de NAT es diferente de aquel punto final y de este modo el mapeamiento de NAT que se ve es diferente de aquel que el punto final usa para enviar paquetes hasta el cliente en aquel par IP:Puerto.

STUN - Simple Traversal de UDP a través de NAT (Travesia simple de UDP sobre el NAT). es un protocolo para configurar el tipo de prueba NAT como fue descrito. El realmente hace un poco más que apenas retornar el par IP:Puerto público, el puede también determinar el tipo de NAT del que usted está detrás. Clientes que usan el protocolo STUN ya existen como el XTEN, por ejemplo. Los pedidos de STUN especifican los siguientes parámetros:

RESPONSE-ADDRESS - El servidor STUN enviará su respuesta para el par IP:Puerto especificado en el atributo RESPONSE-ADDRESS. Si este campo no estuviese presente, entonces el servidor envía su respuesta en el par IP:Puerto de donde el recibió el pedido. Si ambas "flags" **Change IP** y **Change Port** no estuviesen seteadas, el STUN responde de el par IP:Puerto que el paquete inicial fue enviado. Si el **Change IP** estuviese seteado, el servidor responde de un IP diferente y si el **Change Port** estuviese seteado entonces el responde de un puerto diferente.

La respuesta del STUN contiene las siguientes informaciones:

MAPPED-ADDRESS – El par IP:Puerto del cliente como visto en el primer servidor STUN fuera de NAT para recibir el pedido.

CHANGED-ADDRESS – La Dirección IP que debería ser la fuente de respuesta retornada si el pedido fue hecho con el “flag” **Change IP** seteado.

SOURCE-ADDRESS – El Par IP:Puerto de donde la respuesta STUN fué enviada.

Usando una combinación de diferentes pedidos el servidor STUN, un cliente puede determinar si el está en la Internet abierta o si está atrás de un Firewall que bloquea el UDP o si el está atrás de un NAT y de que tipo.

7.11.3 ALG - Application Layer Gateway

Esta técnica se vale de la instalación de un Firewall/NAT mejorado llamado un gateway de capa de aplicación (ALG) que entiende la relación entre los flujos de media y los mensajes de señalización. El ALG procesa los flujos de media y señalización de forma a reflejar la dirección pública y puertos en la comunicación para fuera del Firewall, en otras palabras toda la traducción necesaria es hecha en el gateway. Ruteadores Cisco más recientes con IOS/Firewall y el Pix Firewall permiten estos recursos.

Pista: Varios ruteadores ADSL ya poseen ALG. Tuve la infelicidad de tomar algunas implementaciones con bugs. En este caso el síntoma fue que el ruteador se trabó y fue preciso reiniciarlo. El problema fue corregido deshabilitando el ALG por la interfase telnet del ruteador. Por la interfase Web no tenia esta opción. Dos equipamientos con chipset GlobeSpan Virata nos presentaron este problema en nuestros laboratorios.

7.11.4 Configuración manual

En este método el cliente es manualmente configurado con los detalles de las direcciones públicas IP y puertos que el NAT irá a usar para la señalización y media. En este caso el NAT debe ser configurado manualmente con mapeamientos estáticos en el ruteador.

Asterisk permite ser configurado de forma manual cuando está atrás de un NAT. En el archivo sip.conf en la sesión general, las instrucciones:

<pre> Externip=Dirección IP Externa Localnet=Dirección da Red Local Interna </pre>
--

Permiten que cuando Asterisk está enviando paquetes SIP para afuera de la red la dirección sea substituida por la dirección definida en el comando *Externip*. La línea *Localnet* define cual es direcciones pertenecen a la red local. Todas las redes que no estuviesen en la franja definida en *localnet* son externas. Con esto el Asterisk sabe cuando debe sustituir las direcciones de los encabezados dependiendo del peer de destino.

En el archivo RTP.CONF es posible definir en que puertos RTP el Asterisk va a trabajar.

```

; RTP Configurati on
;
;
[general ]
;
; RTP start and RTP end configure start and end addresses
;
;
rtptest=10000
rtptest=20000

```

7.11.5 COMEDIA Conexión Oriented Media

La solución de arriba funciona bien (Servidor STUN) para los tres primeros tipos de NAT. El cuarto caso (NAT simétrico) no va a permitir este esquema, pues este tiene diferentes mapeamientos dependiendo de la dirección IP objeto. De esta forma el mapeamiento que el NAT designó entre el cliente y la prueba NAT es diferente de aquella entre el cliente y el gateway. En el caso de NAT simétrico el cliente deberá enviar y recibir el RTP de vuelta de la misma dirección IP. Cualquier conexión RTP entre un punto final fuera de NAT y uno dentro de NAT debe ser establecido punto a punto y así (Mismo se una conexión SIP ya fue establecida) el punto final fuera de NAT debe esperar hasta recibir un paquete de un cliente antes que él pueda saber para donde responder. Esto es conocido como **"Media orientada a conexión"**.

Si es deseado que ambos, los UACs que están atrás de NATs y los UACs en la Internet abierta se comuniquen, entonces ellos deben saber si pueden confiar en el mensaje SDP que ellos reciben en el mensaje SIP y cuando ellos precisan esperar recibir un paquete directamente antes que el cliente abra un canal de vuelta para el par IP:puerto fuente de aquel paquete. Una propuesta para informar el punto final para esperar un paquete de entrada y adicionar una línea en el mensaje SDP (Viniendo de clientes detrás de NAT).

a=direction:active

Cuando el dispositivo esta en línea, el entiende que el cliente esta iniciando e irá activamente a establecer el par IP:Puerto para el cual el

dispositivo debe retornar el RTP, y que el par IP:Puerto encontrado en el mensaje SDP debe ser ignorado. La mayoría de los clientes SIP no soportan la directiva "a=". Hasta estos soportes deben existir en algún tipo de traductor en el medio de flujo SIP.

7.11.6 TURN - Traversal using Relay NAT.

Si un dispositivo soporta media orientada a conexión, entonces el problema de atravesar un NAT simétrico está resuelto. Dos escenarios todavía son problemáticos.

1. Si el punto final soporta la directiva a=direction:active tag.
2. Si ambos de los puntos finales están atrás de NATs simétricos.

En cualquiera de los dos casos, una solución es tener un Relay de RTP en el medio del flujo entre los puntos finales. El Relay RTP actúa como un segundo punto final para el cual los dispositivos reales se intentan comunicar uno con el otro. Típicamente, existiría un servidor en el medio del flujo que va a manipular el SDP de forma de instruir los puntos finales para enviar el RTP para el Relay en vez de directamente desde uno hacia el otro. El Relay establecerá su propio mapeamiento de una sesión, guardando el par IP:puerto de cada punto final para donde el debería enviar los paquetes RTP.

7.11.7 ICE - Interactivity Connectivity Establishment

El ICE está siendo desarrollado por la IETF en el grupo de trabajo MMUSIC y experimenta la forma de unificar varias técnicas de atravesar NAT. Esto va a permitir que el cliente VoIP atravesase con éxito una gran variedad de firewalls que existiesen entre el usuario remoto y la red.

ICE define una estandarización para los clientes SIP de forma de determinar que tipo de firewall existe entre ellos y los servidores y determinar una dirección IP en la cual ellos se puedan comunicar. Usando mecanismos como STUN, TURN, RSIP direcciones localmente configuradas que van a proveer una dirección donde el cliente se podrá comunicar. La gran ventaja del ICE es la unificar de los métodos de pasajes por NAT. El ICE usa un proceso interactivo donde es hecho el descubrimiento del mejor método a ser usado.

7.12 Soluciones Prácticas para el Asterisk

Lo más difícil en lo concerniente a NAT en Asterisk es entender que existen diversas situaciones de proyecto y cada una debe ser tratada

individualmente. En primer lugar vamos a tratar esto como dos soluciones separadas.

- Asterisk detrás de NAT
- Clientes detrás de NAT

Obviamente existen diversas situaciones intermediarias y la cosa se complica pensando que tenemos diferentes tipos de NAT (Full Cone, Restricted Cone, Port Restricted Cone e Simétrico). Para aumentar la complejidad tenemos clientes que soportan diferentes tipos de soluciones para NAT (TURN, STUN, ICE, ALG). Esto puede tornar el problema realmente complejo.

7.12.1 Parámetros de Asterisk usados para atravesar NAT

NAT

- nat=yes (puede ser true, t, y, 1 y on)
 - Es la combinación de los modos route + modo rfc3581
- nat=route
 - El Asterisk enviará el audio para el puerto y la ip de donde él recibió el audio en vez de confiar en las informaciones contenidas en los encabezados SIP y SDP. Esto solo va a funcionar si el teléfono detrás de NAT envía y recibe el audio del mismo puerto (RTP).
- nat=rfc3581
 - Este es el estándar, el Asterisk va a adicionar el "rport" al encabezado SIP informando al cliente en que puerto recibió el pedido y va a encaminar el flujo de los mensajes para el cliente en la dirección de donde vino y no en la dirección descrita en los encabezados. El cliente en este punto tiene condiciones de saber su dirección externa (campo Via received) y ahora su puerto externo. Ver rfc3581.
- nat=never
 - En este caso el Asterisk no va a adicionar el rport en la línea VIA del encabezado como en la RFC3581

QUALIFY

Esta opción tiene dos funciones, mantener el NAT abierto y certificarse que el Asterisk no intente enviar una llamada para un teléfono que está inalcanzable.

- Qualify=yes
 - Esta opción usa el valor estándar de 2 segundos.

- Qualify=no
 - Deshabilita el chequeo del peer
- Qualify=x ms
 - Setea el tiempo en ms entre los chequeos

EXTERNIP

Esta es la opción usada dentro de la sesión [general] del archivo sip.conf y puede ser colocada o como una ip o como un hostname apuntando para la dirección externo de su dispositivo NAT.

Ex: externip=200.180.4.168

Usted solo precisa usar esta opción si su Asterisk está detrás de NAT intentándose comunicar con dispositivos fuera de NAT.

LOCALNET

Usada dentro de la sesión [general] del archivo sip.conf. Indica la red privada interna donde el Asterisk no va a usar la dirección externa provista por el parámetro externip.

Ex: localnet=10.1.0.0/255.255.0.0

7.12.2 Escenarios de Asterisk con NAT

Para simplificar, vamos a usar dos situaciones que son las más típicas. Asterisk está detrás de un Firewall sub dominio del área técnica de la empresa. Los clientes son externos y no tienen dominio sobre la configuración de los firewalls de estos clientes

Asterisk detrás de NAT

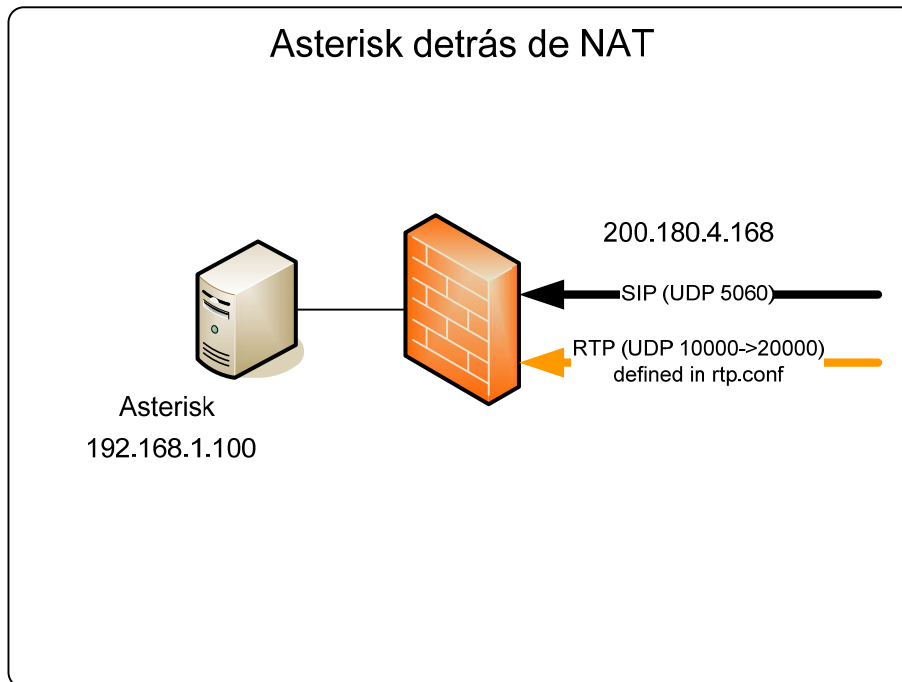


Figura 7.11 Asterisk detrás de NAT

Cuando el Asterisk está detrás de NAT podemos usar las configuraciones `localnet` y `externip` en el archivo `sip.conf` más allá de redireccionar los puertos en el Firewall. Supongo que la dirección IP externa fuese 200.184.7.1 y que la red local interna fuese 192.168.1.0/24. Esto quedaría así:

```
[general ]
nat=yes
externip = 200.84.7.1
localnet = 192.168.1.0/255.255.255.0
```

Más allá de esto, es preciso redireccionar los puertos UDP 5060 y RTP de 10000 a 20000 en el Firewall. Si usted quisiera reducir esta franja puede editar el archivo `rtp.conf`.

Cliente detrás de NAT

Cuando un cliente está detrás de un NAT, normalmente este NAT es dinámico, principalmente cuando es en uso doméstico. Con esto, solo restan las opciones de que el cliente soporte STUN o UPnP para que pueda aprender la dirección de una fuente externa o a partir del ruteador respectivamente.

Otra forma de operar con un cliente, que esta detrás de un NAT, y el Asterisk es el uso de un túnel basado en PPTP, IPIP o IPSec, esto puede ser hecho a través de un ruteador (Cisco o Linux).

Cuando opera un cliente detrás de un NAT y configura STUN en el cliente, se coloca los siguientes parámetros en la configuración del cliente en el archivo sip.conf. Los clientes tienen obligatoriamente que registrarse.

```
nat=yes           ; Ignora el encabezado VIA y usa la dirección de donde
llega el paquete.
canreinvite=no    ; Fuerza el flujo de media por el Asterisk.
qualify=500       ; Fuerza a que un paquete exploratorio que mantiene el
NAT abierto
```

7.13 Consideraciones finales sobre el NAT

Varios proveedores usan soluciones de tipo RTP Relay que permiten una flexibilidad mejor, esto es hecho usando el SER (SIP Express Router) y el rtpproxy del PortOne o el MediaProxy de ag-projects.de. A pesar de ser flexibles ellas adicionan mucha complejidad al ambiente. En ambientes corporativos la solución de VPN me parece ser la más simple y segura. Para proveedores de VoIP talvez no haya mucho para escoger. El uso de IAX también es una opción para quien se quiere librar de los problemas con NAT.

Exclusivo para Vicente Miguel Ferrer

7.14 Cuestionario

1. El SIP es un protocolo de tipo texto similar al _____ e _____.

- IAX
- HTTP
- H323
- SMTP

2. El SIP puede tener sesiones de tipo: (marque todos los que se aplican)

- Voz
- Buzón de voz
- Vídeo
- Chat
- Juegos

3. Podemos citar como componentes de SIP a: (marque todos los que se aplican)

- User Agent
- Media gateway
- PSTN Server
- Proxy Server
- Registrar Server

4. Antes que un teléfono pueda recibir llamados, este se debe _____.

5. El SIP puede operar en modo PROXY y en modo REDIRECT, a diferencia entre ellos es que en el caso de PROXY la señalización Siempre pasa por el computador intermediario (SIP Proxy) en tanto en modo REDIRECT los clientes señalarán directamente.

- Correcto
- Incorrecto

6. En el modo PROXY el flujo de media y la señalización pasan por el "SIP proxy" y no directamente de un cliente para otro.

- Correcto
- Incorrecto

7. El Asterisk actúa como un SIP Proxy.

- Correcto
- Incorrecto

8. La opción canreinvite=yes/no es de importancia fundamental, pues va a definir si el flujo de media va a pasar por el Asterisk o no. Esta afirmación es:

- Correcta
- Incorrecta

9. Asterisk soporta sin problemas la supresión de silencio en canales SIP. La afirmación es:

- Correcta
- Incorrecta

10. El tipo más difícil de NAT para transportar es el:

- Full Cone
- Restricted Cone
- Port Restricted Cone
- Symmetric

Exclusivo para Vicente Miguel Perez Parales

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

Recursos básicos del plan de discado

El plan de discado es sin duda el área de configuración de Asterisk más importante y es el responsable por el funcionamiento de la conmutación de las llamadas. Así como en una central convencional usted disca 0 (cero) para usar una línea externa o disca un ramal dentro de los ramales programados en la central para acceder a otra terminal en el Asterisk para cada número que usted disca, es posible programar una acción y esto es determinado en el plan de discado. El Archivo que contiene el plano de numeración esta localizado en `"/etc/asterisk/extensions.conf"` y en este capítulo usted va a aprender a configurarlo.

8.1 Objetivos del capítulo

Al final de este capítulo usted debe estar apto para:

- Estructurar el archivo `extensions.conf` en sus secciones (general, globales y contextos).
- Crear un plan de discado simple incluyendo contextos, extensiones y prioridades.
- Entender los estándares de extensión y como usarlos.
- Saber como definir y usar variables, funciones y expresiones.
- Usar las aplicaciones `Dial()`, `Answer()` y `Hangup()`.
- Configurar a recepción de llamadas.
- Configurar el discado de llamadas externas.

8.2 Introducción

El plan de discado es lo más importante en la configuración de Asterisk y este es configurado en el archivo `/etc/asterisk/extensions.conf`. Este controla como todas las llamadas de entrada y salida son encaminadas y configuradas.

En el archivo `extensions.conf` usted controla el comportamiento de todas las conexiones a través de su PBX.

8.3 Estructura del archivo `extensions.conf`

El archivo `extensions.conf` es separado en sesiones, la primera es la sección "general" y esta comienza con `[general]` y es seguida de algunos comandos. Cuando usted encuentra `[globals]` la sesión general tendrá terminado e iniciado la sesión global y así en adelante. Estas dos sesiones

son especiales, las restantes son denominadas contextos y también comienzan con "[]". Para algunos esta sintaxis puede parecer extraña, en la versión 1.2 existe el concepto de AEL "application extension language" o lenguaje de extensiones y aplicaciones que se asemejan mucho a los lenguajes de programación y puede ser usada en vez del `extensions.conf`. En este libro no abordaremos en detalle al AEL, visto que su utilización práctica aun es muy limitada.

8.3.1 Sesión [general]

En la cima del archivo `extensions.conf`, usted configura algunas configuraciones generales en la sesión con encabezado `[general]`. Aquí están algunas opciones relacionadas al plan de discado:

static: En la práctica, esta opción afecta apenas la operación del comando `save dialplan`. El valor estándar es "no", pero el archivo ejemplo de `extensions.conf` instalado con el Asterisk explícitamente coloca `static=yes`.

writeprotect: Si las opciones `writeprotect=no` y `static=yes` estuvieran configuradas, entonces usted puede salvar el plan de discado actual con el comando `save dialplan`. Definiciones en las variables globales en la categoría `[globals]` permanecen sin modificación. El valor estándar es "no".

autofallthrough: Este recurso es nuevo en la versión 1.2. Si el `autofallthrough` estuviera configurado para `yes`, al final de un conjunto de comandos para una extensión (después la ejecución de la última prioridad) este terminará la ejecución con un `BUSY`, `CONGESTION` o `HANGUP` dependiendo de que el Asterisk encuentre más conveniente (es la configuración recomendada). Caso contrario, si el Asterisk no tuviese más nada que procesar después de la última prioridad, el Asterisk aguardara que una nueva extensión sea discada (Este era el comportamiento de la versión anterior 1.0).

clearglobalvars: Si este recurso estuviese cargado, las variables globales serán limpias y recargadas cada vez que ocurra una recarga (reload) de Asterisk. Caso contrario las variables van a persistir así mismo que hayan sido borradas del plan de discado. Es un recurso nuevo en la versión 1.2.

priority jumping: Si usted lo configura con "si", entonces las aplicaciones que soportan saltar prioridades (normalmente `n+101`) fundamentado en el resultado de la operación correrán de forma normal (este comportamiento es el compatible con la versión 1.0). Aplicaciones individuales pueden hacer esto pasando un argumento "j" como opción.

Nota:

“save dialplan” sobrescribirá su archivo extensions.conf con uno nuevo generado por el plano de numeración actual. Una copia de su archivo extensions.conf viejo no será guardada. Todos los comentarios serán perdidos. El archivo ejemplo de Asterisk viene con esta configuración peligrosa, writeprotect=no, static=yes

8.3.2 Sesión [globals]

Enseguida en la sesión [globals], usted puede definir las variables globales (o constantes) y sus valores iniciales.

En verdad las variables globales de Asterisk son normalmente usadas como constantes y no como variables. Ellas son usadas para simplificar mudanzas futuras en la configuración del PBX. El valor actual de las variables globales puede ser cambiado usando el comando Set. El valor de la variable global puede ser referenciado, usando la sintaxis: \${nombredevariable}. Las variables globales no son sensibles a mayúsculas y minúsculas.

Que extensiones deben sonar cuando una llega una llamada

```
ENTRADA=>Zap/3&Zap/4
```

Cuanto tiempo debe sonar antes de llegar al buzón de voz

```
RINGTIME=>3
```

Que archivo de audio debe sonar como anuncio de buzón de voz

```
VMANNOUNCE=>mysounds/my-vm-announce
```

Definir los canales a los cuales nuestras extensiones están ligadas

```
COCINA=>Zap/3
SALA =>Zap/4
CUARTO =>Zap/5
```

Cuando nosotros queremos hacer una llamada externa que líneas debemos usar

La creación de estas definiciones, por si solas no tienen efecto alguno. Asterisk no sabe por si solo que hacer con estas variables. Es de su responsabilidad saber que hacer con ellas en su plan de discado.

En este ejemplo, las variables globales, fueron todas escritas con nombres en mayúscula. Nombres de variables globales no diferencian mayúsculas de minúsculas. Ellas son colocadas en mayúsculas, por conveniencia, para diferenciarse de las variables de canal que normalmente tienen mayúsculas y minúsculas.

Las variables globales pueden ser configuradas dentro del plan de discado usando el comando:

```
exten => 4000, 1, Set(global(teste)=0)
```

8.4 Contextos y Extensiones

Después de las secciones [general] y [globals], el resto del archivo extensions.conf es usado en la definición del plan de discado. El plan de discado consiste de una colección de contextos. Cada contexto consiste de una colección de extensiones.

8.4.1 Introducción a contextos y extensiones

Un plan de discado consiste de una colección de contextos. Estas definiciones de contexto son la parte más importante del archivo extensions.conf y la parte más importante de la configuración de Asterisk.

Un contexto es apenas una colección de extensiones. Ejemplo de diagrama de contextos y extensiones.

Contexto "default"

Extensión	Descripción
8580	Flavio
8581	Daniel
8582	Juliano
8583	Verificar el buzón
8585	Sala de conferencia
0	Telefonista

En este ejemplo, que fue dado el nombre de "default", las primeras tres extensiones serán asociadas a teléfonos. La cuarta extensión (8583) será asociada con la aplicación de buzón de voz. La quinta extensión (8585) será asociada a una sala de conferencia. Finalmente el 0 será asociado con el operador.

Aquí está otro ejemplo de un contexto:

Context "menú":

Extensión	Descripción
s	Bienvenido a mensajes e instrucciones

1	Ventas
2	Soporte
3	Contabilidad
9	Directorio
#	Colgar

En este ejemplo de contexto, con el nombre de "menú" tenemos apenas extensiones de un dígito. La extensión "s" es la extensión de inicio, donde el usuario inicia. Esta extensión irá a tocar un mensaje del tipo "Gracias por comunicarse con nuestra empresa", presione 1, ventas, 2, soporte, 3, contabilidad, 9 para en directorio de la empresa o # para colgar. Cada opción del menú es de hecho una extensión y podría o discar una extensión real o hacer algo como enviar para quien disco hacia otro menú.

Los contextos pueden ser usados para programar un número importante de recursos incluyendo:

- Seguridad: Permitir ciertas llamadas de algunos teléfonos.
- Ruteamiento: Rutear llamadas basadas en una extensión.
- Atención automática: Recibe quien disco y pide para entrar a las extensiones.
- Menús multicamada: Menús para ventas, soporte, etc.
- Autenticación: Pedir contraseña para ciertas extensiones.
- Callback: Reducir las tarifas para llamadas de larga distancia.
- Privacidad: Colocar en lista negra, personas a las cuales usted no quiere atender.
- PBX Multihost: Usted puede tener hosts virtuales en su PBX.
- Daytime/Nighttime: Usted puede variar el comportamiento dependiendo del horario.
- Macros: Crear scripts para funciones normalmente usadas.

8.4.2 ¿Cómo son usados los contextos?

Cuando Asterisk recibe una llamada, de entrada o de salida, esta llamada pertenece a un contexto. Cual contexto pertenece a que llamada depende de que canal vea la llamada. Cuando usted configura los canales que usted tiene en su PBX, una de las cosas que usted hace es definir en

que contexto una llamada de aquel canal va a ser colocada, usando una definición como:

```
context=i ncomi ng
```

Entonces la primer forma en que los contextos son usados es hacer que el Asterisk se comporte de forma diferente dependiendo de donde este viniendo la llamada. Con certeza usted va a tener por lo menos un contexto definido. En este contexto usted va a definir que hacer con la llamada, si va ha sonar una de las extensiones, entrar en el buzón de voz o tocar un anuncio de voz. Si usted quiere que Asterisk trate las conexiones de sus extensiones internas de forma diferente, por ejemplo, "poder hacer ddi", usted puede definir que diferentes canales entren en diferentes contextos.

8.5 Extensiones

Una extensión puede ser una de estos tres tipos: literal, estándar, o especial

8.5.1 Literales

Una extensión literal puede ser un número, como el 123, y este puede también contener símbolos estándar como * y # que aparecen en teléfonos normales. De forma que 12#89* es una extensión válida. Algunos teclados de teléfono multi - frecuencial tienen teclas especiales A, B,C,D y las extensiones pueden ser definidas con estas letras también. De hecho, el nombre de la extensión puede contener cualquier letra o número como así algunos caracteres especiales. Note que muchos teléfonos VoIP consiguen discar números de extensión que pueden ser un string arbitrario, tales como Escritorio en Asterisk. ,

Las extensiones llevan en consideraciones minúsculas o mayúsculas? Bien, la respuesta es si y no. Ellas consideran minúsculas o mayúsculas en el sentido de que, cuando Asterisk está intentando encontrar la extensión que el usuario discó encuentra las extensiones que están en el contexto, la extensión debe decir inclusive considerando mayúsculas y minúsculas. Entonces si el usuario disca la extensión "ESCRITORIO" usando su teléfono VoIP, el Asterisk no ejecutará los comandos que usted definió para la extensión "Escritorio". Por otro lado el Asterisk no permite que usted defina extensiones diferentes con el mismo nombre diferenciando solo por mayúscula y minúscula.

8.5.2 Estándares

Cuando usted define las extensiones dentro de un contexto, usted puede no solamente usar números literales, ni nombres alfanuméricos, pero también puede definir extensiones que correspondan a conjuntos de

números discados usando estándares. Un nombre de extensión es un estándar si este inicia con un carácter subrayado "_". Los siguientes caracteres tienen un significado especial:

X	corresponde a cualquier dígito de 0-9
Z	corresponde a cualquier dígito de 1-9
N	corresponde a cualquier dígito de 2-9
[1237-9]	corresponde a cualquier dígito o letra entre llaves (En este ejemplo, 1,2,3,7,8,9)
.	punto, corresponde a uno o más caracteres

Considere el contexto "routing"

Extensión	Descripción
_61XX	Escritorio São Paulo
_63XX	Escritorio Río de Janeiro
_62XX	Escritorio Salvador
_7[1-3]XX	Escritorio Recife
_7[04-9]XX	Escritorio Curitiba

Este contexto, tiene el nombre "routing", y envía llamadas para varios servidores de acuerdo con su extensión. Esta organización decidió que todas las extensiones de teléfono tendrán cuatro dígitos. Si un usuario disca una extensión comenzando con 61 o 63, esto será enviado al escritorio de São Paulo; 62 para el escritorio de Río, 71,72 y 73 para Recife y 70, 74, 75, 76, 77,78, 79 van para el escritorio de Curitiba.

Más ejemplos de estándares

_NXXXXXX	Corresponde a un número de teléfono de 7 dígitos
_021XXXXXXXX	Corresponde a una salida a través de Embratel

8.5.3 Especiales

Asterisk usa algunos nombres de extensión para propósitos especiales.

- i : Invalid (Inválido).
- s :Start (Inicio).
- h: Hangup (Colgó).
- t : Timeout (tiempo de espera).

- T: AbsoluteTimeout (tiempo de espera absoluto).
- o: Operator (Telefonista).
- a: Llamado cuando el usuario presiona "*" durante un mensaje inicial del buzón de voz.
- fax: usado para detección de fax en canales ZAP.
- talk: usado en conjunto con la aplicación BackgroundDetect.

Descripciones:

s: Start. Usado para planos de discado que entran en un contexto sin otras informaciones como el identificador de llamada. Así mismo si usted conoce el identificador de llamada, usted aún tiene un lugar para comenzar.

t: Timeout. Usado cuando las llamadas estén inactivas después un "prompt" haya sido tocado. También usado para colgar una línea que estuviese ociosa.

T: AbsoluteTimeout. Usado para llamadas que hayan sido colgadas debido a que el AbsoluteTimeout() las haya alcanzado. Es útil, por ejemplo, para hacer sonar una notificación con Playback().

h: Hangup. Usado para limpiar una llamada. Puede ser usado para hacer sonar un mensaje de adiós antes de colgar.

i: Invalid. Usado cuando discando para una extensión desconocida en el contexto o entrada desconocida en un menú de URA (Unidad de Respuesta Audible).

Los usos de estas extensiones pueden alterar el contenido de sus registros de facturación (CDR – Call Detail Record). En particular el campo dst no contendrá más que el número discado! Para perfeccionar el problema usted puede ver dentro del Dial() con la opción "g" ("va en el contexto") y considerar el uso del ResetCDR(w) y/o NoCdr().

hint: mapea una extensión para un canal (usado para la luz de ocupado). En Asterisk, un canal puede estar ocupado o sonando, pero una extensión es apenas un string de números que activan una o más aplicaciones.

o: Extensión del operador, usado para la salida presionando 0 en el voicemail.

8.5.4 Definiendo extensiones

Al contrario de un PBX tradicional, donde las extensiones están asociadas con teléfonos, interfaces, menús y así en adelante, en Asterisk una extensión es definida como una lista de comandos a ejecutar. Los comandos son generalmente ejecutados en el orden definido por su etiqueta de prioridad, por otro lado, algunos comandos como el "Dial" y el "Gotoif" tienen la capacidad de redireccionar para otro lugar basándose en alguna condición.

Cuando una extensión es discada, el comando etiquetado con prioridad de 1 es ejecutado, seguido por el comando con prioridad 2 y así sucesivamente. Esto ocurre hasta que la llamada sea colgada, o un comando retorne con código de resultado 1 (indicando falla) o hasta que ningún comando de mayor prioridad exista o la llamada sea ruteada para una nueva extensión.

En cada contexto usted puede definir una o más extensiones. Para cada extensión, usted define un conjunto de comandos. Para cada comando es dada una línea separada con el siguiente formato:

```
exten=>extensi on, pri ori ty, command(parameters)
```

Extensión: Nombre de la extensión, o el nombre literal de la extensión o un estándar. Usted repite exactamente el mismo nombre para cada comando para la extensión

Priority: Número interior. Llamarlo prioridad no es totalmente correcto, es apenas el número del comando, usado para numerar los comandos relacionados a una extensión. Asterisk comenzará la ejecución en la prioridad 1, si no existiese una línea con prioridad 1, entonces la extensión no irá a dar número discado. Después de ejecutar el comando de prioridad 1, Asterisk ejecutará el comando definido con prioridad 2. Si no existiese comando definido con prioridad 2, entonces asterisk termina el procesamiento para esta extensión, Así será lo mismo si existiese otro comando con prioridad 3 o más. Entretanto, algunos comandos, cuando son ejecutados, hacen al Asterisk saltar para una nueva prioridad en vez de ir para la próxima.

Command: Nombre del comando (también llamado aplicación) a ejecutarse.

Parameters: Los parámetros dependen del comando. Algunos comandos no tienen parámetros.

Ejemplo:

```
exten=>123, 1, Answer
exten=>123, 2, Playback(tt-weasel s)
```

```
exten=>123, 3, Voi cema1 I (44)
exten=>123, 4, Hangup
```

Esta es la definición de una extensión con el nombre "123". Cuando una llamada es hecha para la extensión 123, Asterisk responderá la llamada él mismo, hará sonar un archivo de audio llamado tt-weasels, dando al usuario la oportunidad de dejar un mensaje de voz para la casilla 44 y entonces colgar.

Note que Asterisk no da importancia al orden en que usted coloca las líneas en el archivo extensions.conf. Usted puede mezclar las líneas en un orden diferente como en el siguiente ejemplo, y esto no da diferencia porque Asterisk usa la prioridad de cada línea para determinar el orden de ejecución.

```
exten=>123, 1, Answer
exten=>123, 2, Pl ayback(tt-weasel s)
exten=>123, 3, Voi cema1 I (44)
exten=>123, 4, Hangup
```

Otras opciones para definir extensiones incluyen la opción normalmente llamada de lógica de la ex-novia. La lógica dirá la extensión discada viniendo de afuera o de adentro, basándose en la identificación de llamada (callerid) de la persona que está llamando. Por ejemplo:

```
exten=>123/100, 1, Answer()
exten=>123/100, 2, Pl ayback(tt-weasel s)
exten=>123/100, 3, Voi cema1 I (123)
exten=>123/100, 4, Hangup()
```

Esto dirá en la extensión 123 y ejecutará las siguientes opciones solamente si el identificador de llamada del usuario es 100. Esto también puede ser hecho con un estándar en vez de hacerlo con un literal.

```
exten=>1234/_256NXXXXXX, 1, Answer()
```

y así de aquí en adelante.

Esto dirá en el 1234 si el CallerID comienza con 256. Esto es muy útil para evitar que usuarios locales llamen al 0800 y usted tenga que pagar por esto.

Un punto importante por ser entendido es que cada canal VoIP (SIP, H323) o PSTN (Zap, ISDN) es configurado dentro de un contexto. Por lo tanto, cada canal pertenece a un único contexto. Que extensiones o que servicios va ha usar este canal dependerá de las instrucciones contenidas en el plan de discado para aquel contexto.

8.6 Variables

En Asterisk las variables pueden ser globales, variables asociadas a un canal y variables de ambiente. Las variables pueden ser vistas en la consola de Asterisk usando el comando NoOP. Aunque el comando no ejecute nada con la variable, su resultado es mostrado en consola.

Asterisk puede hacer uso de variables globales o específicas por un canal como argumentos para los comandos. Variables son referenciadas en el plan de discado (`extensions.conf`) usando la sintaxis.

```
${varname}
```

Donde "varname" es el nombre de la variable. Un nombre de variable puede ser un string alfanumérico iniciado con una letra. Los nombres de variables definidos por el usuario no son sensibles a mayúsculas y minúsculas. `${VarName}` y `${varname}` se refieren a la misma variable.

Entretanto las variables definidas por Asterisk diferencian mayúsculas y minúsculas. `${EXTEN}` funciona, pero `${exten}` no.

Existen tres tipos de variables:

- Variables globales
- Variables de canal
- Variables de ambiente

8.6.1 Variables globales pueden ser configuradas en la categoría [globals] del archivo `extensions.conf` o usando el comando `Set(global(var)=variable)GlobalVar`. Una vez definido, estas pueden ser referenciadas por cualquier canal a cualquier hora.

8.6.2 Variables de canal son configuradas usando el comando `Set()`. Cada canal recibe su propio espacio de variables, de forma que no hay chancee de colisiones entre diferentes llamadas, y la variable es automáticamente destruida cuando el canal es desconectado.

8.6.3 Variables de ambiente proporcionar un medio de acceder variables de ambiente Unix de dentro de Asterisk. Existe una lista más abajo en esta página.

Si usted define una variable de canal con el mismo nombre de una variable global (recuerde variables definidas por los usuario, no se importan con mayúsculas y minúsculas), las referencias al nombre de la variable van devolver el valor de la variable de canal.

Hay variables de canal predefinidas donde existen algunas variables de canal configuradas por Asterisk que usted puede referírseles como

definiciones del plan de discado. Variables definidas por Asterisk diferencian cajas alta y baja.

- `${CDR(accountcode)}`: Código de contabilización
- `${ANSWEREDTIME}`: Horario cuando la llamada fue atendida
- `${CALLERID(all)}`: El identificador de llamada (nombre y número)
- `${CALLERID(name)}`: El nombre de id de la llamada
- `${CALLERID(num)}`: El número de id de la llamada
- `${CALLINGPRES}`: Variable de presentación PRI Call ID para llamadas de entrada
- `${CHANNEL}`: Nombre del canal actual
- `${CONTEXT}`: Nombre del contexto actual
- `${DATETIME}`: Fecha y hora en formato YYYY-MM-DD_HH:MM:SS.
- `${DIALEDPEERNAME}`: Nombre de quien fue llamado
- `${DIALEDPEERNUMBER}`: Número de quien fue llamado.
- `${DIALEDTIME}`: Hora en que el número fue discado.
- `${DIALSTATUS}`: Status de la llamada.
- `${DNID}`: Identificador del número discado.
- `${EPOCH}`: Época al estilo de Unix (Número de segundos desde 1970).
- `${EXTEN}`: La extensión actual.
- `${HANGUPCAUSE}`: El último código de término de llamada en un canal Zap conectado a una interface PRI.
- `${INVALID_EXTEN}`: La extensión pedida cuando es redireccionada para a extensión i (inválida).
- `${LANGUAGE}`: La lengua actual.
- `${MEETMESECS}`: El número de segundos que un usuario participó de una conferencia en una sala de "MeetMe".
- `${PRIORITY}`: La prioridad actual.
- `${RDNIS}`: El redireccionamiento actual DNIS, Caller ID que redireccionó la llamada.
- `${SIPDOMAIN}`: Dominio de destino SIP de una llamada de entrada.
- `${SIP_CODEC}`: Usada para Setear o codec SIP para una llamada.
- `${SIPCALLID}`: El encabezado SIP dialog Call-ID.
- `${SIPUSERAGENT}`: El encabezado SIP user agent.
- `${TIMESTAMP}`: Fecha y hora en formato: YYYYMMDD-HHMMSS.
- `${TXTCIDNAME}`: Resultado de la aplicación TXTCIDName.
- `${UNIQUEID}`: Identificador único de llamada actual.

8.6.5 Variables específicas de aplicaciones

Algunas aplicaciones usan entradas de datos extras y proporcionan la salida en la forma de variables de canal.

- **ChanIsAvail** retorna `${AVAILCHAN}`: El primer canal disponible.

- Dial recibe input de `${VXML_URL}`: Envía una url XML para un Cisco 7960.
- Dial recibe entrada de `${ALERT_INFO}`: Configura la campanilla para teléfonos Cisco.
- Dial retorna `${CAUSECODE}`: Si en el discado algo se produce, este es el mensaje de error.
- Dial retorna `${DIALSTATUS}`: Texto con el código de status de la última tentativa de discado.
- EnumLookup retorna `${ENUM}`: El resultado de la búsqueda.
- MeetMe recibe input de `{MEETME_AGI_BACKGROUND}`: un script de AGI para correr.
- MeetMe retorna `${MEETMESECS}`: El número de segundos que el usuario este en conferencia.
- Hangup `${PRI_CAUSE}` variable para configurar los códigos de retorno PRI.
- TXTLookup retorna `${TXTCIDNAME}`: El resultado de un DNS lookup.

8.6.6 Variables específicas para Macros

Cuando en un contexto de macro, algunas variables adicionales de canal están disponibles.

- `${ARG1}`: El primer argumento pasado por la macro.
- `${ARG2}`: El segundo argumento pasado por la macro y así es igual en adelante.
- `${MACRO_CONTEXT}`: El Contexto de la extensión que disparó la macro.
- `${MACRO_EXTEN}`: La extensión que disparó la macro.
- `${MACRO_OFFSET}`: Configurado por una macro para influenciar la prioridad de ejecución al salir de la macro.

- `#{MACRO_PRIORITY}`: La prioridad en la extensión donde esta macro fue disparada.

8.6.7 Variables de ambiente

Usted puede acceder variables de ambiente Unix de la siguiente forma:

- `#{ENV(nombredelavariabile)}`.
- `#{ENV(ASTERISK_PROMPT)}`: El prompt actual de la línea de comando CLI .
- `#{ENV(RECORDED_FILE)}`: El nombre del archivo grabado por última vez con el comando Record.

8.7 Expresiones

El uso de expresiones puede ser muy útil para la configuración del plan de discado. Las expresiones combinan variables, valores y operadores para llegar a un resultado. Expresiones son usadas para manipular strings, y realizar operaciones matemáticas y lógicas. Una expresión sigue la sintaxis definida abajo:

```
#[expresión]
```

Vamos a suponer que tengamos una variable llamada SUMA. Y que queremos adicionar un valor a esta variable.

```
#[#{SUMA}+100]
```

Cuando Asterisk encuentra una expresión en el plan de discado, esta substituye la expresión entera con el valor resultante.

8.7.1 Operadores

Los siguientes operadores pueden ser usados en la construcción de expresiones.

Operadores matemáticos

- Suma (+)
- Resta(-)
- Multiplicación(*)
- División(/)
- Porcentaje(%)

Operadores lógicos

- Operador "Y" lógico (&)
- Operador "O" lógico (|)
- Operadores de comparación (=, >, >=, <, <=, !=)

Operador para expresiones regulares

- Expresión regular (:).

Vale la pena explicar un poco que es una expresión regular. En una expresión regular comparamos un string con un estándar, esta comparación puede ser falsa o verdadera, dependiendo si el string encaja en el estándar o no. Expresiones regulares son muy comunes en lenguajes de scripts como Perl y PHP y son muy usadas para manipular strings. Es importante que antes de usar expresiones regulares usted pueda leer algún tutorial, sobre el asunto, varios están disponibles en Internet. Explicar detalladamente las expresiones regulares escapa de este libro.

Ejemplo

Usted puede usar el comando Set que substituye el antiguo comando SetVar (versión 1.0x) para testear sus variables. Vamos a usar la aplicación NoOP() como una forma de mandar los resultados para la pantalla de la consola.

```
exten=1000, 1, Set(TEST=10)
exten=1000, 2, Set(SUMA=${TEST}*5)
exten=1000, 3, NoOP(${SUMA})
```

8.8 Funciones

A partir da versión 1.2 las funciones ganarán importancia y algunas aplicaciones están siendo substituidas por funciones. Ellas permiten procesar las variables de una forma todavía mas avanzada que los operadores. Usted puede verificar que funciones están disponibles en Asterisk usando el comando de consola:

```
CLI>core show functi ons
```

8.8.1 Compresión de String

`${LEN(string)}` retorna la compresión del string nombredelavariabile.

```
Ej empl o:
exten=>100, 1, Set(Fruta=pera)
exten=>100, 2, NoOp(${LEN(Fruta)})
exten=>100, 3, NoOp(${LEN(${Fruta}}))
```


El primer NoOp debería mostrar un valor igual a 5 (La Compresión del string "fruta"). La segunda operación NoOp debería mostrar el valor de 4 (La compresión del string "pera").

Esta es una buena manera de verificar por un string vacío o nulo.

8.8.2 Substrings

```
${string:offset:length}
```

offset: punto de inicio

length: compresión

Retorna un substring de string, iniciando en la posición definida por "offset" y retornando la compresión de caracteres definidos en "length".

Si el offset es negativo, esto es tomar de la derecha hacia la izquierda a partir del final del string.

Si la compresión es omitida, o es negativa, entonces todo el resto del string a partir del punto de inicio (offset) es retornado.

```
Ejemplos:
${123456789:1}-retorna el string 23456789
${123456789:-4}-retorna el string 6789
${123456789:0:3}-retorna el string 123
${123456789:2:3}-retorna el string 345
${123456789:-4:3}-retorna el string 678
```

Ejemplos de uso:

```
exten=>_NXX.,1,Set(areacode=${EXTEN:0:3})
```

Toma los primeros tres dígitos de la variable \${EXTEN}

```
exten=>_516XXXXXXX,1,Dial(${EXTEN:3})
```

Toma todos excepto los primeros tres dígitos de \${EXTEN}

```
exten=>100,1,Set(whichVowel=4) exten=>100,2,Set(foo=AEIOU:${whichVowel}:1)
```

Setea \${foo} para una única letra U

8.8.3 Concatenación de Strings

Para concatenar dos strings, simplemente escribalas juntas.

```
${foo}${bar}
555${el numero}
${PrefijoLargaDistancia}555${ONumero}
```

8.9 O plan de discado en la práctica

Vamos a usar los conceptos aprendidos arriba para ejecutar varias configuraciones que son comunes en el universo de centrales telefónicas. De esta forma podremos introducir varias aplicaciones dentro de un contexto práctico. Cada aplicación será presentada dentro del primer escenario en que esta fuera necesaria.

8.9.1 Discando entre ramales.

Para habilitar el discado entre ramales podremos usar el concepto de la variable de canal {EXTEN} que significa la extensión discada. Por ejemplo si una parte del ramal de la central va desde 4000 hasta 4999 y todos los ramales son SIP podremos usar el siguiente comando de abajo:

```
exten=_4XXX, 1, Dial (SIP/${EXTEN})
```

El comando Dial() usado arriba posee la siguiente sintaxis:

```
; Discando un canal
Dial (ti po/i denti fi cador, ti meout, opc i ones, URL)
```

```
; Discado para múltiplos canales
Dial (ti po1/i denti fi car1&ti po2/i denti fi car2/ti pon/i denti fi cadorn, ti meout, opc i ones, URL)
```

Parámetros:

Tipo: especifica el tipo de canal. Debe ser uno de los canales registrados como "Zap", "SIP", "IAX2" y así en adelante.

Identificador: especifica el número de teléfono a ser discado en este canal. El formato de "número de teléfono" depende del canal y puede contener parámetros adicionales (Ej. un toque de campanilla personalizado).

Si usted desea especificar más de un canal para que el comando Dial intente – recuerde que este discará todos simultáneamente – Sepárelos con el símbolo &.

Timeout: Este parámetro es opcional. Si no fuese especificado, el comando Dial esperará indefinidamente.

Opciones: Este parámetro, que es opcional, es un string conteniendo cero o más de una de las siguientes "flags".

- **A(x):** Toca un anuncio (x.gsm) para la persona llamada.

- **C:** Resetea el CDR (Registros de Facturación). Esto es como usar el comando NoCDR.
- **d:** Permite que una extensión de un dígito sea discada en tanto la otra parte no atiende al llamado. Esto permite que usted salte hacia otra prioridad en el medio de la comunicación.
- **D(dígitos):** Después aparte de responder la llamada, envía dígitos como un flujo de DTMF, entonces conecta la llamada al canal originador.
- **f:** Fuerza el callerid para ser configurado como una extensión de la línea que esta haciendo o redireccionando la llamada. Por ejemplo, algunas operadoras no permiten callerids de otras extensiones de aquellas que están designadas para usted.
- **g:** Cuando la persona llamada cuelga, sale a ejecutar más comandos en el contexto actual.
- **G(contexto^extensión^prioridad):** Si una llamada es atendida, este comando transfiere ambas comunicaciones para un contexto y extensión específicos. La parte originadora es transferida para la prioridad x y la parte llamada para la prioridad x+1. Esto permite al plan de discado distinguir entre el origen y destino de la llamada (versión 1.2).
- **h:** Permite al usuario llamado colgar discando *.
- **H:** Permite al usuario originador colgar discando *.
- **j:** Salta n+101 prioridades de todos los canales solicitados si estuviesen ocupados. (comportamiento estándar de la versión 1.0). es un recurso nuevo de la versión 1.2.
- **L(x[:y][:z]):** Limita la llamada a 'x' ms, enviando cuando restan 'y' ms, repetidos a cada 'z' ms). Solo 'x' es necesario. Las siguientes variables especiales son opcionales para limitar las llamadas .
- **LIMIT_PLAYAUDIO_CALLER** – yes|no (default yes) – Toca un sonido para el originador.
- **LIMIT_PLAYAUDIO_CALLEE** – yes|no (default yes) – Toca un sonido para el que fue llamado.
- **LIMIT_TIMEOUT_FILE** – Archivo para tocar cuando el tiempo expira.
- **LIMIT_CONNECT_FILE** – Archivo para tocar cuando la llamada inicia.
- **LIMIT_WARNING_FILE** – Archivo para tocar como aviso si 'y' está definido. Si LIMIT_WARNING_FILE no estuviese definido, entonces una macro de sonido especial para decir automáticamente cuanto tiempo resta ("Usted tiene [XX] minutos [YY] segundos").
- **m:** Provee música en espera para el originador hasta que el usuario llamado responda. Esto es mutuamente exclusivo con la opción r.

- **M(x)**: Ejecuta la macro(x) en la conexión de la llamada.
- **P(x)**: Usa el **PrivacyManager**, usando el X como banco de datos.
- **r**: Genera un tono de campanilla para el origen, no pasando ningún audio de canal llamado hasta que alguien responda.
- **S(n)**: Cuelga la llamada n segundos después que la persona llamada atiende.
- **t**: permite al usuario llamado transferir la llamada.
- **T**: permite al usuario de origen transferir la llamada.
- **o**: restaura el comportamiento de la identificación de llamadas de Asterisk 1.0 (enviar el identificador de llamada original), en Asterisk 1.2 (envía este número de extensión).
- **w**: Permite al usuario de destino iniciar una grabación presionando *1 o el que estuviese definido en el archivo features.conf (1.2). Requiere el siguiente comando, Set(DYNAMIC_FEATURES=automon)
- **W**: Permite a la parte que originó la llamada iniciar la grabación después de presionar *1 o el que estuviese definido en features.conf. (1.2). Requiere el siguiente comando, Set(DYNAMIC_FEATURES=automon)

El parámetro opcional URL también será enviado a la parte llamada cuando de conexión con éxito. Si la tecnología de canal soporta el envío de URLs de esta forma.

Otro ejemplo del comando Dial:

```
exten=_4XXX, 1, Dial (SIP/${EXTEN}, 20, tTm)
```

En el ejemplo de arriba, el comando Dial discará el canal SIP correspondiente a la extensión. El origen y el destino podrán transferir (tT) y si hubiera música en espera en vez de tono de campanilla en el discado.

8.9.2 Discado para un destino externo.

Para discar una ruta externa (dentro de la red pública de telefonía PSTN) normalmente se usa en Brasil el número 0. En algunos lugares el número 9 es usado para discado externo. Para salir para la red pública es necesario un canal Zapata (analógico o digital ISDN), uncall (digital R2) o sip (proveedor de voz sobre IP).

```
exten=_0ZXXXXXXXX, 1, Dial (Zap/1/${EXTEN:1}, 20, tT)
```

El comando de arriba permite el discado para un número local (O Z no permite el discado de un segundo 0). Observe que la comunicación podrá

salir apenas por el primer canal de la placa zaptel. Si hubiese más canales disponibles y el primer canal estuviese ocupado la comunicación no se completará. Observe también la expresión `/${EXTEN:1}` que es igual al número discado menos el primero dígito.

Si colocásemos el comando como de abajo:

```
exten=_0ZXXXXXXXX, 1, Di al (Zap/g1/${EXTEN:1}, 20, tT)
```

El parámetro "g1" busca el grupo de canales 1 permitiendo así el uso de todos los canales.

Usando:

```
exten=_0ZZXXZXXXXXXXX, 1, Di al (Zap/g1/${EXTEN:1}, 20, tT)
```

El usuario podría hacer un interurbano dentro del país.

8.9.3 Discando 0 para tomar una línea externa.

Si usted no tiene ninguna restricción en relación a la que el usuario puede discar usted puede simplemente dar al usuario el tono de línea externa.

```
exten=0, 1, Di al (Zap/g1, 20, tT)
```

8.9.4 Forzando una única operadora

Usando un poco los conceptos de concatenación y substrings es posible forzar una operadora cuando da ejecución de un DDD. Si yo quisiese forzar una ficticia operadora nacional (código 55).

```
exten=_0ZZ. , 1, Di al (Zap/g1/055${EXTEN:3}, 20, tT)
```

Observe que en este ejemplo yo use un nuevo concepto, el estándar "." (punto) que permite un número variable de dígitos (algunos números en Brasil aún poseen 7 dígitos).

8.9.5 Recibiendo una llamada en la telefonista.

Suponga que el ramal de la telefonía sea un teléfono SIP de número 4000 y que la comunicación este siendo recibida en un canal del tipo FXO con el contexto configurado para "entrada" (configurado en el Zapata.conf por ejemplo). En este caso la comunicación será procesada en la extensión especial "s" ya que un número de identificación del destino (ramal) no puede ser proporcionado.

```
[entrada]
exten = s, 1, Answer()
```

```
exten = s, 2, Di al (SIP/4000, 115, tT)
exten = s, 3, Hangup()
```

8.9.6 Recibiendo una comunicación con DDR

Si usted posee una línea digital (E1 con señalización RDSI (ISDN) o R2) usted puede usar un sistema conocido como discado directo al ramal que normalmente dispensa la telefonista. Con este sistema la operadora pasa los últimos cuatro números del teléfono para su central. Suponga que su conjunto de DDR va desde 4830258550 hasta 4830258599. La configuración quedaría así:

```
[entrada]
exten => _85[5-9]X, 1, Answer()
exten => _85[5-9]X, 2, Di al (SIP/${EXTEN}, 15, tT)
exten => _85[5-9]X, 3, Hangup()
```

8.9.7 Tocando varias extensiones

Frecuentemente es deseado que una extensión dada, en primer lugar toque un teléfono, y entonces si no existiese respuesta, toque otro teléfono (o conjunto de teléfonos).

```
exten => 9, 1, Di al (Zap/1, 15, tT)
exten => 9, 2, Di al (Zap/1&Zap/2&Zap/3, 15)
exten => 9, 2, Hangup()
```

En este ejemplo, cuando alguien disca 9 para la telefonista, primero es intentado con el canal Zap/1 que es el teléfono de la recepcionista. (Si el canal estuviese ocupado o no responde después de 15 segundos, intentamos tocar un grupo de teléfonos, incluyendo el de la telefonista) por más de 15 segundos.

8.9.8 Evitando el telemarketing

```
exten => 100/, 1, Zapateller
exten => 100, 1, Wait(0)
exten => 100, 2, Di al (Zap/1)
```

En este ejemplo, si una llamada es recibida sin identificación de llamada, entonces es ejecutada la aplicación Zapateller (Esta toca un tono familiar especial, aquel que usted oye cuando llama a un número que no está en servicio, frecuentemente haciendo que discadores automáticos, 'normalmente usados en tele marketing' desconecten).

Si el identificador de llamadas CallerID fuera proporcionado, entonces el aplicativo .Wait() es ejecutado por 0 segundos (en otras palabras no hace nada). En cualquier caso, el canal Zap/1 toca entonces indefinidamente (Sin timeout).

8.9.9 Ruteamiento por el originador de la llamada

En este ejemplo, ya conocido como extensión Ante ex-novia, muestra como Asterisk puede rutear basándose no solamente en el número de quien fue llamado, así también con el número de quien llama.

```
exten => 8590/482518888, 1, Congesti on
exten => 8590, 1, Di al (Zap/1, 20)
exten => 8590, 2, Voi cemaí l (u8590)
exten => 8590, 102, Voi cemaí l (b8590)
```

En este ejemplo, hecho en base al ejemplo anterior, fue adicionando una regla especial, donde si el originador es 482518888 (ruteamiento por el callerid es indicado colocando un ./ y el identificador de la llamada) este recibe inmediatamente el tono de ocupado. Otros originadores son atendidos normalmente. Un ejemplo más común de ruteamiento por el identificador de llamada es:

8.9.10 Usando variables en el plan de discado

Asterisk puede hacer uso de variables globales y de canal para argumentos de determinadas aplicaciones. Variables son expresada en el plan de discado usando `${teste}` donde `.teste.` es el nombre de la variable. Una variable puede ser cualquier string alfanumérico comenzando con una letra, pero existen algunas variables con significados especiales como ya hemos visto anteriormente.

Considere el siguiente ejemplo:

```
[gl obal s]
Fl avi o => Zap/1
Dani el => Zap/2&SI P/pi ngtel
Ana => Zap/3
Cri sti ano => Zap/4

[mai nmenu]
exten => 1, 1, Di al (${Dani el }&${Fl avi o})
exten => 2, 1, Di al (${Ana}&${Cri sti ano})
exten => 3, 1, Di al (${Ana}&${Fl avi o})
```

Organizando el plan de discado de esta forma se torna fácil la mudanza de interfaces físicas para cualquier usuario en particular. Esto muda todas las referencias en el plan de discado instantáneamente.

8.10 El estilo de la versión 1.2

Varios nuevos recursos se tornaran disponibles a partir de la versión 1.2. Una de las principales modificaciones fue el fin de salto de 101 posiciones. Los planos de discado de la versión 1.2 se caracterizan principalmente por

el uso de la prioridad n (próximo) y por el uso de etiquetas (labels) en las prioridades. Vea el ejemplo de abajo de instrucción dial:

```

exten=2000, 1, Dial (SIP/2000)
exten=2000, n, goto(${DIALSTATUS})
exten=2000, n, hangup()
exten=2000, n(BUSY), voi cema I (b2000)
exten=2000, n, hangup
exten=2000, n(NOANSWER), voi cema I (u2000)
exten=2000, n, hangup
exten=2000, n(CHANUNAVAILABLE), hangup
exten=2000, n(CANCEL), hangup
exten=2000, n(CONGESTION), hangup

```

Usted puede observar arriba diversas construcciones nuevas. En primer lugar surge la prioridad "n" de next (próximo), ella permite que usted no tenga que estar numerando su plan de discado. En segundo lugar usted puede observar las etiquetas (labels). Ellas permiten discar un punto en el plan de discado y por fin el comando "goto" salta para las etiquetas de acuerdo con el retorno de la variable \${DIALSTATUS}. Usted puede observar que los resultados de la variable \${DIALSTATUS} permiten un control mucho más rápido del estado de la llamada.

En la versión 1.2 usted puede usar también la variable \${HANGUPCAUSE} que permite retornar las causas de conexión de acuerdo con el protocolo q.931 usado en canales ISDN.

8.11 Creando un plan de discado simple

La idea de esta sesión es mostrar como podría ser creado el plan de discado en el archivo /etc/asterisk/extensions.conf en dos situaciones bastante comunes.

8.11.1 Central de cuatro puertas FXO y 16 ramales SIP.

Vamos a suponer que si fuese a construir un PBX con cuatro puertas analógicas y 16 teléfonos IP SIP. Consideremos para esto el ejemplo de abajo:

```

Instalada la placa TDM400P en un PC corriendo Asterisk
zaptel.conf configurado correctamente
zapata.conf configurado correctamente
sip.conf configurado correctamente
Grupo de canales = 1 (group=1 en zapata.conf)
Contexto de las FXO = entrada (context=entrada en zapata.conf)
Contexto de los canales SIP = ramales
Numeración de los ramales de 20 a 35
Telefonista en el ramal 20 o discando 9

```



```

; Recepción de llamadas en la telefonista
[entrada]
exten=s, 1, Answer()
exten=s, 2, Dial(SIP/20, 20, tT)
exten=s, 3, Hangup()

[ramales]
; discado para otros ramales
exten=_[2-3]X, 1, Dial(SIP/${EXTEN}, 20, tT)
exten=_[2-3]X, 2, Hangup()

; comunicación externa
exten=0, 1, Dial(Zap/g1, 20, tT)

; comunicación para la telefonista
exten=9, 1, Dial(SIP/20, 20, tT)

```

8.11.2 Central con un E1 de 15 canales o 50 ramales SIP

Vamos a suponer que si fuese a construir un PBX con quince canales digitales R2 y 50 teléfonos IP SIP. Consideramos para esto el ejemplo de abajo:

Instalada la placa TE110P en el PC corriendo Asterisk
 zaptel.conf configurando correctamente
 unicall.conf configurando correctamente
 sip.conf configurando correctamente
 Grupo de canales = 1 (group=1 en zapata.conf)
 Contexto de R2 = entrada (context=entrada en unicall.conf)
 Contexto de los canales SIP = ramales
 Numeración de los ramales de 4000 hasta 4049
 Telefonista en el ramal 4000 o discando 9
 DDR habilitado

```

; Recepción de llamadas en la telefonista
[entrada]
exten=_40[0-4]X, 1, Answer()
exten=_40[0-4]X, 2, Dial(SIP/${EXTEN}, 20, tT)
exten=_40[0-4]X, 3, Hangup()

[ramales]
; discado para otros ramales
exten=_40[0-4]X, 1, Dial(SIP/${EXTEN}, 20, tT)
exten=_40[0-4]X, 2, Hangup()

; comunicación externa
exten=_0., 1, Dial(Unicall/g1/${EXTEN:1}, 20, tT)

; comunicación para telefonista
exten=9, 1, Dial(SIP/4000, 20, tT)

```

8.12 Sumario

En este capítulo usted aprendió que el plan de discado es la pieza principal de configuración del Asterisk. Este está formado de contextos

extensiones y prioridades. Usted aprendió a conectar dos aparatos de PBX usando el comando Switch. Aprendió a usar variables de tipo global de canal y de ambiente. Entendió como funciona el procesamiento de las extensiones y el sistema de inclusión de contextos. Aprendió a crear macros para simplificar el trabajo de configurar las extensiones. Y por fin entendió como funcionan y para que sirvan las extensiones estándares y las prioridades.

8.13 Cuestionario

1. En la configuración de la sesión [general] el valor estándar de la opción `writprotect=no`. Siendo así si usted emite el comando `save dialplan` en la línea de comando CLI `>`. (Marque todas que se aplican).

- Asterisk sobrescribirá el `extensions.conf` con la configuración actual.
- Todos los comentarios serán perdidos.
- Será hecho un backup de `extensions.conf` en `extensions.bak`.
- La opción `static=yes` debe estar configurada para poder salvar el plan de discado.

2. Normalmente las variables globales van a estar escritas en mayúsculas en tanto las variables de canal estarán comenzando con mayúscula y restantes en minúsculas. Esto no es obligatorio, pero facilita la lectura.

- Verdadero
- Falso

3. Y extensión 's' es usada como extensión de inicio dentro de un contexto, este es normalmente usado en los siguientes casos.

- En la entrada de una llamada sin CallerID
- Como entrada de un menú llamado por comando `Background()`.
- En la entrada de una llamada con CallerID
- Como entrada de un contexto que fue direccionado por el `Goto()`.

4. Cite cuatro situaciones donde contextos pueden ser utilizados:

5. Para usar una variable en el plan de discado en Asterisk usted debe usar el siguiente formato:

- `#[varname]`
- `{varname}`
- `$(varname)`

`${varname}`

6. Las variables disponibles en el Asterisk son: (Escoja 3)

- Constantes
- Variables públicas
- Variables de ambiente
- Variables globales
- Variables privadas
- Variables de canal

7. Para obtener la compresión de un string usted puede usar la función _____.

8. Para concatenar strings basta colocar una al lado de la otra como en el ejemplo abajo. La afirmación está:

```
${foo}${bar}  
555${elnumero}  
${PrefijoLargaDistancia}555${elNumero}
```

- Correcta
- Incorrecta

9. Suponga que usted tenga una central telefónica analógica. Coloque en el espacio de abajo los comandos necesarios para direccionar la recepción de las llamadas para una telefonista en el ramal SIP/4000. Si el ramal SIP/4000 no atiende, El sistema debe direccionar después de 15 segundos para los ramales SIP/4000 y SIP/4001 simultáneamente.

10. Suponga que usted tenga una central telefónica digital. Coloque en el espacio de abajo los comandos necesarios (`extensions.conf`) para permitir el discado externo solo para números DDD en el contexto `[ramales]`. Coloque también los comandos necesarios para forzar el discado por la operadora 55 desconsiderando la operadora escogida por el usuario.

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

Recursos avanzados del plan de discado

Ahora que usted ya aprendió lo básico del plan de discado, vamos a sofisticar nuestra configuración con nuevos comandos y recursos.

9.1 Objetivos

AL final de este capítulo, el lector deberá estar apto para:

- Limitar comunicaciones externas usando el concepto de inclusión de contextos.
- Alterar la prioridad de como Asterisk encuentra una extensión.
- Usar el concepto de Macros para evitar la digitalización repetitiva de comandos.
- Configurar un menú de URA para recepción de llamadas.
- Programar seguimiento incondicional y seguimiento para ocupado usando la base de datos de Asterisk AstDB.
- Usar contextos basados en horarios y el comando gotoiftime() para crear mensajes fuera del horario de expediente.
- Programar el recurso de lista negra de teléfonos que no deben ser atendidos.
- Programar una central más sofisticada con menú de URA, seguimiento y seguimiento incondicional, restricción de DDD y DDI, autenticación de los usuarios y lista negra.
- Entender los principales pasos en la configuración de un PBX basado en Asterisk.
- Usar el concepto de Switches para pasar comunicaciones para otro servidor Asterisk.

9.2 Inclusión de contextos

Un contexto de extensiones puede incluir el contenido de otro. Por ejemplo, considere los siguientes contextos.

Contexto default

Extensión	Descripción
4101	Juan da Silva
4102	Jose da Silva
0	Teléfono

Contexto local

Extensión	Descripción
_9NXXXXXX	Llamadas locales

include=>	"default"
-----------	-----------

Contexto larga distancia

Extensión	Descripción
_91NXXNXXXXXX	Llamadas de larga distancia
include=>	"local"

Aquí definiremos tres extensiones:

El contexto default permite discar tres extensiones: Juan, José y el telefonista. El contexto local tiene un estándar de extensiones para permitir el discado de números de siete dígitos (llamadas locales). El contexto larga distancia tiene un estándar de extensiones para permitir un discado de larga distancia, y este también incluye el contexto "local", de este modo permitiendo al usuario hacer llamadas locales y también discar las extensiones de Juan, José o la telefonista.

Usando contextos de extensión usted puede cuidadosamente controlar quien tiene acceso a los servicios de discado.

Si más de un estándar corresponde al número discado, el Asterisk puede no usar el que usted espera. Vea la sección, correspondiente al orden de búsqueda de las extensiones del asterisk.

Cuando el Asterisk recibe una llamada de entrada en un canal, el Asterisk mira en el contexto definido por el canal por comandos diciendo al Asterisk lo que este debería hacer. El contexto define diferentes conjuntos de comandos dependiendo de que extensión el usuario discó. Por ejemplo, el contexto puede dar un conjunto de comandos para definir cual usar si el usuario discó "123" y otro conjunto de comandos si el usuario disco "9" y otro si el usuario discó cualquier número iniciado con "555".

Para cada contexto, usted debe definir una o más extensiones. El Asterisk las usa para comparar contra el número discado. Para cada extensión, usted dice al Asterisk que hacer listando un conjunto de comandos.

Las dos reglas de oro sobre contextos:

Una extensión solo puede discar para números que están dentro del mismo contexto.

El contexto de una comunicación es definida por el canal de entrada de comunicación (zapata.conf, iax.conf, sip.conf).

9.2.1 Limitando el discado para números DDD y DDI

Usted puede observar en las líneas de abajo que tenemos cuatro contextos. Usted puede configurar el contexto de un teléfono SIP usando

la instrucción "context=" en la definición del teléfono en el archivo sip.conf. Si usted define el contexto como "ramales" el usuario solo podrá discar para otros ramales. Si se define "local" el usuario podrá discar para los ramales y para números locales porque el contexto ramales está incluido en el contexto local. Las instrucciones [DDD] y [DDI] complementan el concepto.

```
[ramal es]
exten => _4XXX, 1, Di al (SIP/${EXTEN})
exten => _4XXX, 2, Congesti on()

[l ocal ]
ignorepat => 0
exten => 0, 1, Di al (Zap/g2/)
exten => 0, 2, Congesti on()
i ncl ude => ramal es

[DDD]
ignorepat =>0
exten => _00XXXXXXXXXXXX. , 1, Di al (Zap/g2/${EXTEN: 1})
exten => _00XXXXXXXXXXXX. , 2, Congesti on()
i ncl ude => l ocal

[DDI ]
ignorepat => 0
exten => _00. , 1, Di al (Zap/g2/${EXTEN: 1})
exten => _00. , 2, Congesti on()
i ncl ude => DDD
```

Siempre es bueno mantener en un PBX nuevo la misma experiencia que en un PBX anterior. Cuando usted disca 0 en su PBX actual, este da el tono de línea externa. Cuando usted disca 0 en Asterisk, este permanece mudo esperando que usted disque todo el resto. Para resolver esto tenemos un comando muy útil, la instrucción ignorepat.

La línea ignorepat manda a los canales del Asterisk a no dar el tono de discado hasta que el estándar este siendo discado, de la misma forma después de discar 0, este todavía recibe el tono de discado como si estuviese tomando una línea externa.

9.3 Como el plan de discado encuentra la extensión

El discado "encuentra en tanto disca" ocurre en dos situaciones:

- Menús de la unidad de respuesta audible. Algo como: disque 1 para..., 2 para
- Aparatos telefónicos estándar conectados por interfaces FXS. Esta es una situación donde usted levanta el teléfono, recibe la tonalidad y comienza a discar el número para llamar.

Que es el "encuentra en tanto disca"?

Considere una línea telefónica tradicional conectada en la compañía telefónica. Cuando usted levanta el tubo y comienza a discar, cada dígito es enviado y procesado inmediatamente, (“encuentra en cuanto disca”). En contraste, considere un teléfono celular, usted digita el número de teléfono y al final presiona el botón “envía” y el teléfono envía todo de una vez.

Una vez que la llamada tenga completado, ambos teléfonos funcionan de la misma forma. Si por ejemplo, usted disca para una compañía y es recibido por el sistema de buzón de voz, cualquier dígito que usted disque, ya sea usando un teléfono celular o uno común son enviados a medida que usted marca.

Para Asterisk, el estilo “todo de una vez” de discar es más fácil de procesar: Asterisk solo verifica si el número discado condice con cualquier estándar de extensión en el contexto actual del plan de discado.

El estilo “encuentra en cuanto disca” es un poco más difícil porque no está claro cuando usted va a terminar el discado. De un lado usted no quiere que Asterisk permanezca esperando; usted quiere que este procese el número lo más rápido posible, por otro lado Asterisk tiene que esperar si algún dígito adicional va a afectar como esta gerencia su llamada.

Entender como el “encuentra en tanto usted disca” funciona, lo ayudara a decidir como colocar sus números de extensión y como definir los padrones de extensión usados.

9.3.1 Proceso “encuentra en tanto usted disca”.

Cuando el Asterisk está procesando el número discado, la primer cosa que Asterisk considera es:

Cuando usted teclea otro dígito, existirán padrones de extensión que podrán dar con el número? Asterisk considera el número que usted discó hasta el momento y busca los padrones de extensión definidos en aquel contexto. Si la respuesta a la cuestión es que si, entonces el Asterisk va a esperar para ver si usted va a teclear otro dígito, lo mismo si existe un número de teléfono completo que corresponda al que usted ya digitó.

Por ejemplo, vamos a decir que usted tenga:

```
exten=>123, 1, DoSomething(...)
exten=>_XXX., 1, DoSomethingElse(...)
```

Si usted digitó 123, Asterisk esperará para ver si usted va a digitar otro dígito, porque no está claro que usted quiere la extensión 123. Por

ejemplo, la extensión 123999, podría corresponder al segundo estándar definido.

Asterisk verificará todos los estándares de extensión definidos para el contexto actual. Si existe por lo menos un estándar posible, que puede ocurrir en el caso que usted tecleé un nuevo dígito, Asterisk esperará.

Cuánto tiempo Asterisk esperará? Eso va a depender de dos factores. Primero, si es una nueva llamada de entrada o una conexión establecida.

Nuevas llamadas de entrada son aquellas en que Asterisk no inicio el procesamiento de cualquier comando del plan de discado. Para canales Zap, esto significa que alguien levanto el tubo y comenzó a discar y Asterisk está procesando cada dígito en la medida en que este es discado.

Para nuevas llamadas, el segundo factor que afecta cuanto tiempo Asterisk espera por más dígitos antes de desistir es el tipo de canal de donde la llamada esta viniendo. Cada tipo de canal determina su propio período de timeout. Un canal Zap tiene un tiempo fijo de 3 segundos y esto no puede ser mudado a no ser modificando el código fuente y re-compilando el módulo Zap.

Conexiones establecidas son aquellas donde Asterisk ya hizo la conexión inicial, fue para la extensión (al igual que la extensión s), inició el procesamiento de comandos y permaneció sin comandos para procesar y está esperando ahora que algo sea discado. Típicamente, uno de los comandos que el Asterisk podría haber ejecutado es el comando Background para tocar un mensaje grabado, presentando un menú de URA y Asterisk quedara esperando que sea discada la elección del menú. Para conexiones establecidas, el factor que afecta el período de timeout es el valor DigitTimeout. El valor estándar de digittimeout es de 5 segundos, pero esto puede ser cambiado usando el comando del mismo nombre.

Lo que enseguida hace Asterisk

Si asterisk esperó, pero no recibió ningún dígito en el período de timeout, o el Asterisk no encontró cualquier estándar de extensión, si usted disco mas dígitos, podría discar con su número. Es entonces que Asterisk va a considerar la cuestión: el número se marca con cualquier extensión en el contexto actual?

Si: Salta para la primer extensión que marca. Si más de una extensión se marca, asterisk salta para la primera que este encuentra. Asterisk no considera los estándares de extensión en el orden en que usted los define, de forma que la extensión encontrada puede no ser la que usted desea.

No: Si el número que usted discó es inválido en el contexto actual, entonces la acción que asterisk toma depende en que situaciones ella esta.

1. Si esta es una conexión de entrada nueva, entonces Asterisk retornará una señal apropiada de "número inválido" para quien discó. Si fuera un canal Zap el usuario recibirá un tono de discado inválido, si fuera un teléfono IP podrá eventualmente recibir un mensaje de discado inválido. En esta situación, ningún comando del plan de discado fue ejecutado y no existe control sobre el comportamiento de Asterisk en el modo que el gerencia un número inválido. Asterisk no salta para la extensión *i* en esta situación.

2. Si por otro lado, esta una conexión establecida, Asterisk salto para una extensión, inició la ejecución de comandos (ej. tocando un mensaje de un menú de ura) y permaneció sin comandos, esperando que el usuario disque alguna cosa. En este caso el discado de un número inválido hace que Asterisk salte para la extensión *i*. Si la extensión *i* no estuviese definida en este contexto, asterisk colgará.

9.3.2 Ejemplo

Una empresa quiere que sus llamadas telefónicas sean respondidas con un mensaje de voz dando la bienvenida a quien llamo e invitando a escoger una extensión. La empresa tiene seis extensiones que son 1,2,21,22,31,32. Entonces el contexto creado para las llamadas entrantes es:

```
[i ncomi ng]
exten=>s, 1, Background(bi enveni do a l a empresa)
exten=>1, 1, Di al (Zap/1)
exten=>2, 1, Di al (Zap/2)
exten=>21, 1, Di al (Zap/3)
exten=>22, 1, Di al (Zap/4)
exten=>31, 1, Di al (Zap/5)
exten=>32, 1, Di al (Zap/6)
```

Cuando usted llama a la empresa, asterisk tocará el archivo bienvenido-a-la-empresa.gsm. Después de esto, permanece sin comandos para ejecutar y espera que usted disque algo. Esto es lo que este hará si usted disca varias opciones.

Número discado	Acción de Asterisk
1	Inmediatamente hace el discado Dial(Zap/1)
2	Espera por el timeout y entonces hace Dial(Zap/2)
21	Inmediatamente hace el Dial (Zap/3)
22	Inmediatamente hace el Dial (Zap/4)
3	Espera por el timeout entonces cuelga
31	Inmediatamente disca para Dial(Zap/5)

```
32      Inmediatamente hace el Dial(Zap/6)
4      Inmediatamente cuelga
```

Note que cuando alguien intenta discar la extensión 2, este no es conectado inmediatamente. Asterisk espera para ver si quien disco va a teclear más dígitos para determinar cual extensión es deseada 2,21 o 22. Como todo el mundo le gustaría ser atendido inmediatamente, lo ideal es no usar un plan de discado ambiguo.

Si la compañía no quisiera perder llamadas si fuera discado un número inválido, el ideal es definir una extensión i (invalida) para este contexto y t (timeout).

9.3.3 Orden de busca de los estándares de extensión

Cada contexto definido en el plan de discado dirá al Asterisk como procesar los números de teléfono en aquel contexto. Como usted puede usar estándares para definir extensiones, un estándar de extensión puede con un número de teléfono dado. Asterisk no marca los números contra los estándares en el orden que usted los define, los estándares son ordenados primero.

Problema ejemplo:

Vamos a decir que para el contexto ejemplo, usted desearía que los números iniciados con 918 salgan por una línea analógica conectada en Zap/1, y todos los otros números salgan por la Zap/2. Entonces usted escribe algo como sigue:

```
[ejemplo]
exten=>_918.,1,Dial(Zap/1/${EXTEN})
exten=>_.,1,Dial(Zap/2/${EXTEN})
exten=>h,1,Hangup
```

Solo que no funciona. Usted descubre que no importa el número que usted disca, todos los números son enviados vía Zap/2. Esto ocurre porque Asterisk ordena las extensiones y toma la primera que corresponda. Para ver el orden por el cual Asterisk ordena las extensiones, digite el comando dialplan show ejemplo en la consola. Usted va a descubrir que el orden es:

```
_918
h
```

Note que este es un orden diferente del cual usted definió en su archivo extensions.conf. La línea con "_" es ahora la primera, incluyendo aquellas

que comienzan con 918. Note también que la extensión h no puede ser alcanzada, porque ella también corresponde al _.

9.3.4 Controlando el ordenamiento

Entonces como hace para que Asterisk disque los estándares de extensión en el orden que usted quiere. Usando la palabra clave *include* para incluir otro contexto de estándares de extensiones dentro del contexto actual.

```
[ej empl o]
include=>ej empl o-sub
exten=>h, 1, Hangup
exten=>_918., 1, Dial (Zap/1/${EXTEN})

[ej empl o-sub]
exten=>_., 1, Dial (Zap/2/${EXTEN})
```

Asterisk (cuando en el contexto "ejemplo") procesará los números discados en el siguiente orden.

```
_918
h
_
```

Note lo que Asterisk hace:

Las entradas "exten=>" dentro del contexto ejemplo son ordenadas y testeadas primeras. El contenido de cada contexto incluido es ordenado y testado enseguida. Los contextos incluidos son testados en el orden de las líneas incluidas en extensions.conf.

Reloading

Si usted quisiera recargar el plan de discado después de cambios, sin recargar toda la configuración de Asterisk. Use el comando de consola dialplan reload.

Un archivo grande o varios archivos pequeños?

Con la instrucción `#include <nombre de archivo>` en extensions.conf, otros archivos son incluidos. De esta forma usted puede configurar el sistema donde extensions.conf es el archivo principal, users.conf contiene los usuarios locales, services.conf contiene varios servicios como conferencia. De esta forma el plan de discado puede ser más fácil de mantener, dependiendo del tamaño de su setup. No confunda el `#include <nombre de archivo>` con la instrucción `include <context>`. El `#include` funciona en todos los archivos de configuración de Asterisk.

9.4 Recibiendo una llamada en un menú de voz.

Para mostrar los menús de voz precisamos entender los comandos `Background()` y el comando `Goto()`. El comando `playback` se puede ver en el capítulo tres. Abajo tenemos una descripción que cita los comandos `Background` y `Goto()`.

El comando `Background()`

Toca un archivo de audio, pero retorna el control inmediatamente.

Sintaxis:

```
Background(nombre del archivo)
```

Comienza tocando un archivo de audio dado, pero inmediatamente retorna, permitiendo al archivo de audio tocar de fondo entre tanto los próximos comandos (cualesquiera sean) se ejecuten. (Comparando con el comando `Playback`, que también toca un archivo de audio, pero que espera hasta que el audio este terminando antes de retornar el control). Si usted hiciese una segunda llamada al aplicativo `Background`, en tanto el primer todavía estuviese sonando, el segundo archivo será alineado para tocar después del término del primero.

Si el Asterisk permanece ejecutar comandos para una extensión dada, pero la conexión todavía estuviese activa (El usuario todavía no descolgó), El Asterisk terminará de tocar el audio en `background` antes de activar el temporizador `ResponseTimeout`.

Un uso del comando **`Background()`** es tocar un mensaje de menú (Presione 1 para ventas, 2 para soporte,..."), sin tener ningún comando adicional. De este modo el Asterisk, no teniendo comandos para ejecutar, espera por cualquier dígito DTMF que el usuario pueda discar en tanto esta continua tocando el archivo de audio. Si el usuario disca un número de extensión que es valido en el contexto actual, el archivo de audio será interrumpido y Asterisk saltará para la extensión discada. Si el usuario no disca una extensión, entonces el Asterisk terminará de tocar el archivo de audio de fondo y esperará silenciosamente por un tiempo en segundos definido por la variable **`ResponseTimeout`** hasta que el usuario disque algo. Si termina el tiempo de control va para la extensión `t` (defendiéndose en este contexto, señal colgar). Si el usuario disca una extensión que es invalida en el contexto actual, el control salta para la extensión `i` (Definida en este contexto, señal colgar).

Ejemplo:

```
[i ncomi ng]
exten=>s, 1, Answer
```

```

exten=>s, 2, ResponseTimeout(5)
exten=>s, 3, Background(mimenu)
exten=>s, 4, Background(escojaalgo)
exten=>t, 1, Goto(s, 2)
exten=>i, 1, Playback(pbx-invalid)
exten=>i, 2, Goto(s, 2)
exten=>1, 1, DoSomething
exten=>2, 1, DoSomethingElse
exten=>3, 1, Dial(...)

```

En este ejemplo, una llamada llegada de un contexto de entrada es inmediatamente respondida por Asterisk (paso s,1). Nosotros configuramos el tiempo de respuesta en 5 segundos (paso s,2). Entonces comenzamos a tocar el audio del archivo "mimenu.gsm" (paso s,3). Entonces alineamos hacia el audio del archivo "escojaalgo.gsm" para tocar una vez que el primer audio haya terminado. (paso s,4). Como no existe paso s,5, El Asterisk continua tocando mimenu.gsm y escojaalgo.gsm hasta estar terminado y entonces silenciosamente espera cinco segundos más por una respuesta. Si en cualquier momento, con el audio sonando o no, el usuario disca una extensión entonces el Asterisk para de tocar el audio y salta para la extensión. Si esta disca "1", "2" o "3", entonces Asterisk saltará para la extensión apropiada.

Si el usuario disca cualquier otro número, Asterisk salta para la extensión "i" y toca el audio del archivo "pbx-invalid.gsm" entonces vuelve para el inicio. Si el usuario no digita ninguna extensión de fin de 5 segundos de silencio, entonces Asterisk saltará para la extensión "t", y de esta siguen a **Goto** para volver al inicio nuevamente.

Note que si usted usa el comando **Background** en un macro este va siempre a llamar a una extensión fuera del contexto de quien llamó.

El comando goto()

Descripción:

Salta para una prioridad en particular, extensión o contexto.

Sintaxis:

```
Goto([[context|]extension|]priority)
```

Configura la prioridad para un valor específico opcionalmente, configurando opcionalmente también la extensión del contexto. La extensión BYEXTENSIÓN es especial, porque usa la extensión actual, de este modo permitiendo que usted va para un contexto diferente sin especificar una extensión dada.

Variantes válidas del comando:

Goto(context,extension,priority)

Goto(extension,priority)

Goto(priority)

Aviso: Insertando un espacio después de las comas separando los parámetros podrá producir resultados inesperados.

Ejemplo:

Goto(confexisting, 1, 1) buscare una extensión " 1", con un espacio en la frente.

Códigos de retorno:

Siempre retorna 0, lo mismo para un contexto dado, extensión, o la prioridad sea inválido.

Ejemplo:

exten=>1, 1, Goto(submenu, s, 1)
exten=>600, 4, Goto(s, 6)

Un menú de voz es típicamente programado dentro de su propio contexto.

```
[menuprincipal]
exten => s, 1, Background(bienvenido-menuprincipal)
exten => 1, 1, Goto(ventas, s, 1)
exten => 2, 1, Dial, Zap/2
exten => 9, 1, Directory(default)
exten => 0, 1, Dial, Zap/3
```

```
[ventas]
exten => s, 1, Background(bienvenido-ventas)
exten => 1, 1, Goto(default, 100, 1)
exten => 2, 1, Goto(default, 101, 1)
```

Un anuncio normalmente es tocado en la extensión s, al entrar en el menú. Entonces la aplicación Background toca un mensaje, mientras espera que el usuario disque una extensión. El ejemplo de arriba presenta dos menús, uno llamado "menú principal" y otro llamado "ventas". Cuando alguien entra en el contexto menú principal oirá un anuncio (Como presione 1 ventas, 2 soporte, 9 directorio y 0 telefonista). Al discar 1, el originador es transferido para ventas, discando 2 sonará Zap/2, hará sonar el Zap/3 y 9 presentará el directorio de la empresa.

Ejemplos de discado:

El tipo de extensión más común es aquella la que disca para otra interfase. El hecho de llamar con otra interfase es realizado con la aplicación Dial. En este ejemplo usamos el comando Dial en su forma más simple.

```
exten => 8590, 1, Dial (Zap/1, 20)
exten => 8590, 2, Voicemail (u8590)
exten => 8590, 102, Voicemail (b8590)
```

En este ejemplo, es una de las pocas excepciones para la ejecución de una extensión estando fuera de orden. Cuando esta extensión es discada, la primera cosa que Asterisk intenta discar es la interfase Zap/1, por un máximo de 20 segundos. Si la interfase está ocupada, esta saltará para la prioridad n+101 si tal prioridad existe en esta extensión. En este caso, tenemos nosotros prioridad (102), la cual envía el originador para la caja postal de buzón de voz 8590 con un mensaje de ocupado (busy). Si simplemente nadie atiende, entonces la ejecución continuará en el paso siguiente, donde el originador será colocado en el buzón de voz con el mensaje de "nadie atendió" (unavailable).

El comando gotoif()

El comando gotoif() permite un salto condicionando la resolución de una expresión. Vea el ejemplo de abajo:

Sintaxis:

```
Goto f(condición?label 1[: label 2])
```

Ejemplo:

```
; implementando un loop en la ura
exten=s, 1, set(counter = 3)
exten=s, n(inicio), Background(escoj asuopción)
exten=s, n, set(counter=${counter}-1)
exten=s, n, gotoif(${counter} = 0)?fin: inicio)
exten=s, n(fin), hangup

exten=1, 1, noop("ejecuta opción 1")
exten=2, 1, noop("ejecuta opción 2")
```

Cuidado en las expresiones pues se precisa que haya espacio antes y después de la señal de igual.

9.5 Switches, encaminando para otro Asterisk

Sintaxis:

```
[iaxprovider]
switch=> IAX2/user: [key]@server/context
```


Especifica el encaminamiento para otro servidor. El usuario y clave precisan ser definidos en el archivo `iax.conf` del servidor que es llamado. El contexto es el contexto del servidor de destino.

9.6 Macros

Un macro es un conjunto de comandos que van a ser ejecutados en secuencia. Los macros son usados primariamente para procesar las extensiones, de forma que no sea necesario en el plan de discado digitar varias líneas de comando para cada extensión.

Programación del Macro.

Descripción:

Macro(macroname, arg1, arg2...)

Ejecuta un macro usando el contexto `macro-<macroname>`, saltando para la extensión "s" de aquel contexto y ejecutando cada paso, así entonces para ir retornando al final.

Note que usted no puede usar cualquier otra extensión diferente de s para construir el macro, de forma que el control es retornado al contexto de la llamada cuando termina el procesamiento del macro.

La extensión que fue llamada, contexto, y prioridad son almacenadas en `${MACRO_EXTEN}`, `${MACRO_CONTEXT}` y `${MACRO_PRIORITY}` respectivamente. Los argumentos se tornan `${ARG1}` y `${ARG2}` etc en el contexto del macro. Si usted usa el comando Goto para afuera del contexto del Macro, este terminará y el control será retornado al local indicado por Goto (Ver comandos y aplicaciones). Si usted coloca `${MACRO_OFFSET}` en el término, el macro intentará continuar en la prioridad `MACRO-OFFSET+N+1`, si tal paso existe y N+1 en el caso contrario.

Códigos de retorno: El Macro retorna 1 si cualquier paso en el macro retorna -1, sino retorna 0.

Ejemplo:

```
[macro-stdexten]
; Macro extensión estándar
;
; ${ARG1}-Teléfono por sonar (ring)
exten=>s, 1, Dial (${ARG1}, 20, j)
exten=>s, 2, Voicemail (u${MACRO_EXTEN})
exten=>s, 3, Goto(default, s, 1) ; Si fuese presionado #, retorna al inicio
exten=>s, 102, Voicemail (b${MACRO_EXTEN}) ; Si esta ocupado, envía un
voicemail con o sin anuncio de busy
```

```
exten=>s, 103, Goto(default, s, 1) ; Si fuese presionado #, retorna al inicio
```

Para llamar esta macro usted usaría:

```
exten => 1234, 2, Macro(stdexten, sip/7960mark)
```

Veremos diversas macros en ejemplos del archivo extensions.conf.

9.7 La base de datos de Asterisk

La base de datos de Asterisk usa la versión 1 de Berkley DB que es parecido al registry de Windows. Este banco de datos puede ser usado por Asterisk para almacenar datos temporarios, y configuraciones. Un ejemplo de uso es la transferencia con consulta, donde si el teléfono está ocupado este guarda la extensión en una base de datos y permanece reintentando hasta conseguir.

Los datos son agrupados en familias e identificados con una llave que es única dentro de la familia.

9.7.1 Familias

cidname: usado por el comando LookupCIDName

blacklist: usado por el comando LookupBlacklist

9.7.2 Funciones

Existen funciones de acceso al banco de datos de Asterisk que usted puede usar dentro del plan de discado en el archivo extensions.conf:

- **DBdel():** Borra una clave de la base de datos.
- **DBdeltree():** Borra una familia o arbole de la base de datos.

9.7.3 Ejemplo de uso del Asterisk DB.

Sígame, Sígame cuando esta ocupado.

Si usted usa este ejemplo, un teléfono puede discar *21* <número> para hacer Sígame inmediato o *61* para hacer sígame si estuviese ocupado, y #21# o #61# para cancelar la configuración.

Abreviaciones usadas:

CFIM – familia de base de encaminamiento inmediato.

CFBS – familia de base de encaminamiento cuando esta ocupado.

```
[macro-stdexten]
; ${ARG1}-Extensi on
exten=>s, 1, gotoi f(${DB_EXI STS(CFIM/${ARG1})?cfi m)
exten=>s, n(di al ), Di al (SI P/${ARG1}, 20)
exten=>s, n, goto(${DI ALSTATUS})
exten=>s, n, hangup()
exten=>s, n(BUSY), gotoi f(${DB_EXI STS(CFBS/${ARG1})?cfbs: end)
exten=>s, n(cfbs), Di al (SI P/${DB(CFBS/${ARG1})}), 20)
exten=>s, n, hangup()
exten=>s, n(cfim), Di al (SI P/${DB(CFIM/${ARG1})}), 20)
exten=>s, n, hangup()
```

Notas:

- Este macro lee las variables almacenadas en la base de datos de Asterisk.
- El Macro ejecuta el Dial(SIP/\${temp}) si un número de redireccionamiento fuera encontrado.
- Si la función DB() no encuentra una llave, esta va hacia la prioridad 102.

```
[apps]
; Si game
exten=>_*21*X. , 1, Set(DB(CFIM/${CALLERID(num)})=${EXTEN: 4})
exten=>_*21*X. , 2, Hangup
exten=>#21#, 1, DBdel (CFIM/${CALLERID(num)})
exten=>#21#, 2, Hangup

; Sí game si esta ocupado o no di sponi ble
exten=>_*61*X. , 1, Set(DB(CFBS/${CALLERID(num)})=${EXTEN: 4})
exten=>_*61*X. , 2, Hangup
exten=>#61#, 1, DBdel (CFBS/${CALLERID(num)})
exten=>#61#, 2, Hangup
```

```
[pbx]
exten=7000, 1, Macro(stdexten, SI P/7000)
exten=7001, 1, Macro(stdexten, I AX/7001)
```

En el esquema de incluir la extensión apps en su contexto.

9.8 Programando el recurso de lista negra.

Para programar el recurso de lista negra usaremos la aplicación LookupBlackList(). Esta aplicación verifica un nombre/número de identificación de llamada de una base de datos de una lista negra. Si el número fuera encontrado y si existe la prioridad n+101, Asterisk saltará para esta prioridad, sino retornará cero. No hace nada si no recibe identificador de llamada.

Ejemplo:

```
[entrada]
exten => s, 1, LookupBlacklist(j)
exten => s, 2, Dial(SIP/4000, 20, tT)
exten => s, 3, Hangup()
exten => s, 102, Goto(bl_oqueado, s, 1)

[bl_oqueado]
exten => s, 1, Answer()
exten => s, 2, Playback(IIamadabl_oqueada)
exten => s, 3, Hangup()
```

Para insertar un número en la lista negra podemos usar el mismo recurso usado anteriormente en "sígame". Usando *31* seguido de la extensión. Para remover de la lista negra basta usar #31# seguido del número a ser removido de esta lista.

```
[apps]
exten=>_*31*X., 1, Set(DB(blacklist/${EXTEN}=1))
exten=>_*31*X., 2, Hangup()
exten=>_#31#, 1, DBdel(blacklist/${EXTEN})
exten=>_#31#, 2, Hangup()
```

Los números pueden ser colocados en banco de datos usando comandos de consola en la interfase de línea de comando CLI.

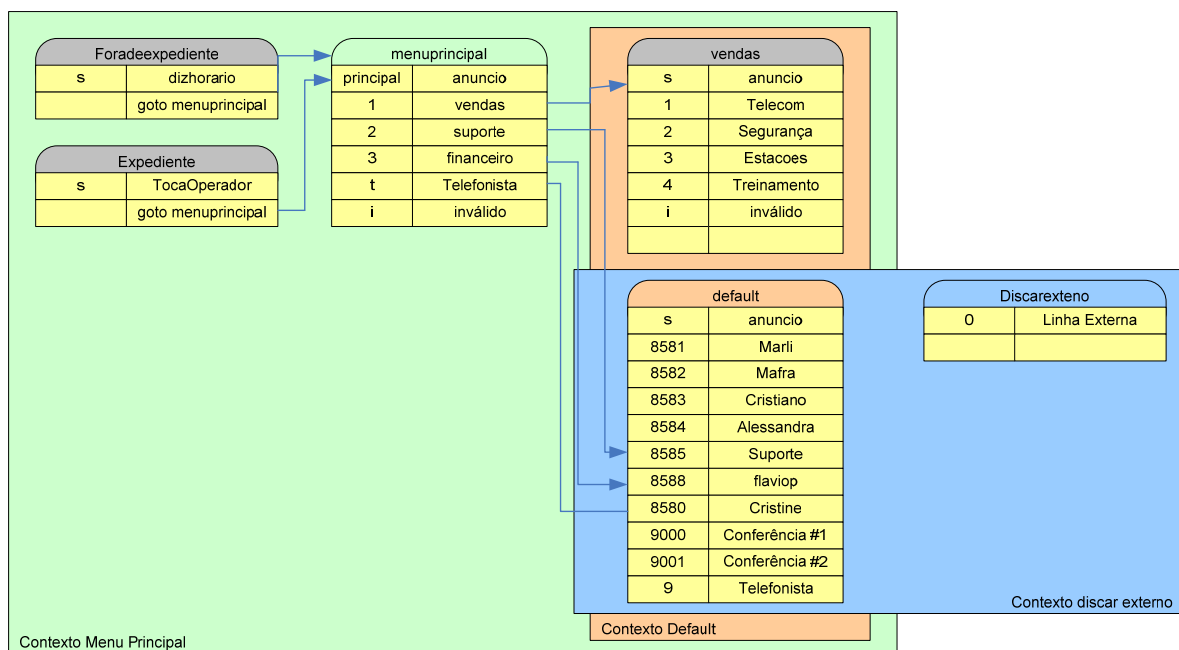
```
CLI>database put blacklist <nombre/número> 1
```

Note que el valor asociado a la clave es 1 y puede ser cualquier valor. La aplicación LookupBlackList buscará la llave y no el valor.

Para apagar use:

```
CLI>database del blacklist <nombre/número>
```

9.9 Contextos basados en horario



En la figura de arriba tenemos un plan de discado hecho de varios contextos. Cuando una llamada está entrando, esta viene por un canal. Este canal pertenece a un contexto. En el caso de arriba el canal pertenece al contexto "menuprincipal". Dentro del contexto menuprincipal, tenemos dos contextos que fueron incluidos "fueradeexpediente" y "expediente". Estos dos contextos fueron incluidos usando el comando:

```
include => context | <times> | <weekdays> | <mdays> | <months>
```

Por ejemplo:

```
; Este es el horario de expediente
include=>expediente | 08:00-18:00 | mon-fri | * | *
```

```
; Este es el horario fuera de expediente
include=>fueradeexpediente | 18:00-23:59 | * | * | *
include=>fueradeexpediente | 00:00-08:00 | * | * | *
include=>fueradeexpediente | * | sat-sun | * | * | *
```

De forma que uno de los dos va a tener la extensión s (Start) estos van a tocar la llamada apropiada y transferir para el menuprincipal. Después del anuncio del menú principal el sistema esperará por el discado de 1 dígito, pudiendo transferir para ventas o para cualquier ramal en el contexto default. Note que no es posible transferir para una línea externa, pues el contexto discar externo, no está incluido en el contexto menú principal.

Por otro lado si analizamos la salida de llamadas, vamos a imaginar que un canal haya sido colocado en el contexto [discarexterno] (Muy semejante a las clases de ramales de los PBX convencionales), este puede discar un número externo, y también discar cualquier ramal en el contexto default que también fue incluido en el contexto [discarexterno]. Es importante entender que el que define el contexto de un canal no es la entrada "exten=>" en el archivo extensions.conf, lo que define el contexto de un canal es la configuración del canal (sip.conf o zapata.conf dependiendo del tipo de canal). Por esto un canal que haya sido definido como contexto default puede discar apenas los otros ramales y uno que haya sido definido como [discarexterno], puede discar externamente y también para todos los ramales, pues el contexto default fue incluido en el contexto discarexterno.

Los mensajes dependientes de horario pueden también ser programados usando la aplicación:

GotoifTime().

El formato del comando GotoifTime sigue abajo:

```
GotoifTime(<timerange>|<daysofweek>|<daysofmonth>|<months>?[[context|]exten
sión|]pri)
```

Si la hora actual corresponde al tiempo especificado, el comando saltará para la extensión especificada. Cada uno de los ítems podrá ser rellenados con "*" (siempre) o como una faja. Si el tiempo actual no indica con una hora especificada el próximo comando deberá ser ejecutado.

Como especificar el tiempo:

```
<timerange> = <hour>:'<minute>'-'<hour>:'<minute> | "*"
<daysofweek> = <dayname>|<dayname>'-'<dayname> | "*"
<dayname> = "sun"|"mon"|"tue"|"wed"|"thu"|"fri"|"sat"
<daysofmonth> = <daynum>|<daynum>'-'<daynum> | "*"
<daynum> = un número de 1 a 31
<hour> = un número de 0 a 23
<minute> = un número de 0 a 59
<months> = <monthname>|<monthname>'-'<monthname> | "*"
<monthname> = "ene"|"feb"|"mar"|"abr"|"may"|"jun"|"jul"|"ago"|"sep"|"
"oct"|"nov"|"dic"
```

Nombres de día y de mes no diferencian mayúsculas y minúsculas.

```
exten=>s, 1, GotoifTime(8:00-18:00|mon-fri|*|*?expediente, s, 1)
```

Transfiere para el contexto expediente la extensión "s" prioridad 1 si la llamada llega entre las 08:00 hasta las 18:00, no verificando el día del mes o el mes.

9.10 Abriendo un nuevo tono de discado con DISA.

DISA o "direct inward system access" es un sistema que permite que el usuario al discar para el Asterisk reciba un segundo tono y posea discar a partir del servidor Asterisk. El formato del comando sigue abajo:

```
DISA(passcode[|context])
DISA(archivo de contraseñas)
```

Ejemplo de uso:

```
exten => s, 1, DISA(no-password|default)
```

Con este comando por ejemplo el usuario disca para la central y sin pasar una contraseña, recibe un tono de discado. Cualquier llamada colocada en DISA en este caso será procesada en el contexto "default".

Los argumentos para esta aplicación permiten una contraseña global o contraseñas individuales dentro de un archivo. Si ningún contexto fuese especificado será usado el contexto "disa". Si fuera usado un archivo de contraseña (especificado el camino completo del archivo) el archivo puede contener # y ; para comentarios. Mas alla de esos argumentos pueden tener un callerid adicionado a el para que un nuevo callerid pueda ser usado en esta llamada. Ejemplo:

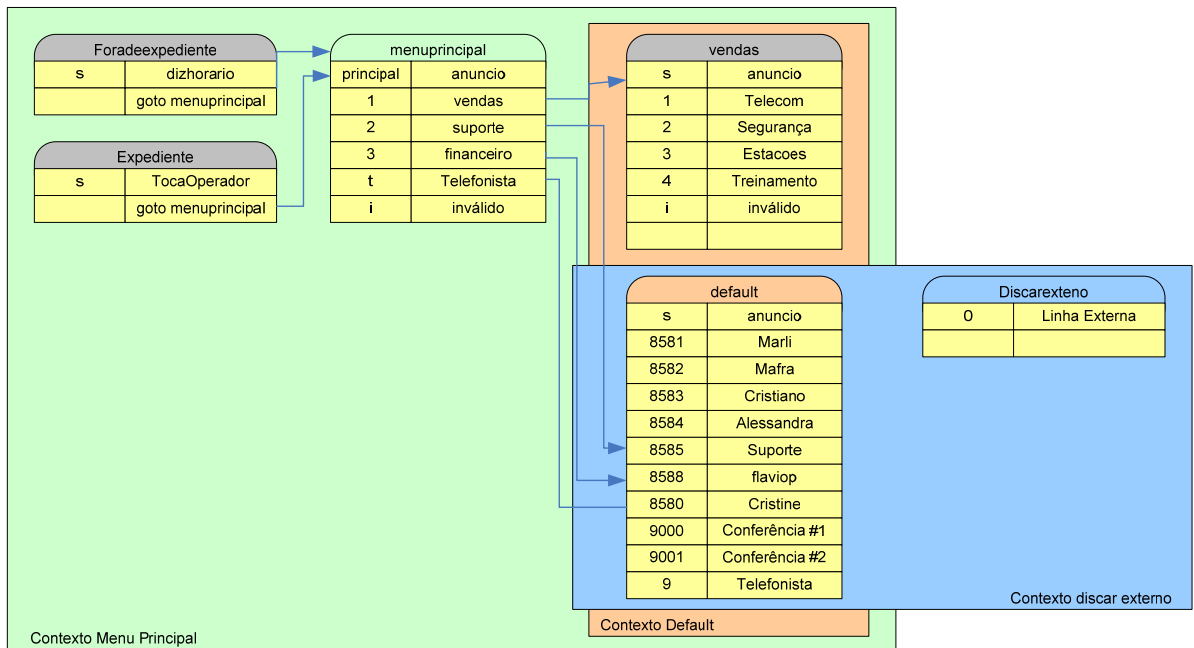
```
numeric-passcode|context|"Flavio" <(48) 30258590>
```

9.11 Limitando el número de llamadas simultaneas

La versión 1.2 incluye una serie de comandos que permiten colocar un canal dentro de un grupo y contar cuantos canales activos existen en un determinado grupo en un determinado momento. Vamos a usar el siguiente ejemplo para mostrar como funciona. Suponga que usted tiene una filial llamada "Rio" donde los teléfonos comienzan con "_214X" y que esta localidad es servida por un canal de datos donde apenas 64Kbps fuesen reservados para comunicaciones. En este caso usted va a querer que apenas dos comunicaciones al mismo tiempo puedan ser hechas para esta localidad. Vea el ejemplo de abajo:

```
exten=>_214X, 1, set(GROUP()=Rio)
exten=>_214X, n, Gotoi f($[${GROUP_COUNT()} > 1]?revento)
exten=>_214X, n, Dial(SIP/${EXTEN})
exten=>_214X, n, hangup
exten=>_214X, n(revento), playback(comunicacionesexcedidas)
exten=>_214X, n, hangup
```

9.12 Un plan de discado en la práctica



Vamos a construir un plan de discado en la práctica para una pequeña central telefónica. La central posee las siguientes características y requisitos:

- 4 enlaces analógicos.
- 16 ramales IP basados en SIP.
- 3 clases de ramales.
 - restricto (Interno, comunicaciones locales y 0800).
 - ddd (comunicaciones nacionales).
 - ddi (comunicaciones internacionales).
- Mensaje cuando se esta fuera de expediente
- Atención Automática de llamada

9.12.1 Paso 1 - Configurando los canales.

Enlaces analógicos (zapata.conf)

En primer lugar vamos a configurar los enlaces analógicos. Estos enlaces son interfaces de tipo fxo. Usted puede usar una placa Digium de tipo

T400P o cuatro placas modem Motorola o Intel (Aquellas de las que sí se compran cada 5 una sale buena).

Vamos asumir en este caso que los drivers del hardware de la placa de telefonía (zaptel.conf) ya están configurados y cargados. En el archivo zapata.conf usted configura los canales zap correspondientes a los enlaces de la central.

```
signal | i ng=fxs_ks
language=en
context=entrada
group=1
channel => 1-4
```

Canales SIP (sip.conf)

Vamos a escoger el rango de numeración del número 2000 al número 2100. Dos codecs van a ser utilizados, GSM y G711 ulaw. El primero es para teléfonos conectados a Internet o en líneas de baja velocidad, el segundo para los teléfonos locales. Vamos hacer una convención que ramales de los números 2000 hasta el 2039 son restrictos, de los ramales 2040 hasta el 2059 pueden hacer DDD y del número 2060 en adelante pueden hacer DDI.

```
[general ]
di sal low=al l
al low=gsm
al low=ul aw
bi ndport = 5060
bi ndaddr = 0.0.0.0
context = ramal es

[2000]
type=fri end
username=20
secret=contraseña
host=dynami c
mai l box=20
context=restri cto
canrei nvi te=yes

[2040]
type=fri end
username=20
secret=contraseña
host=dynami c
mai l box=20
context=puededdi
canrei nvi te=yes
dtmfmode=rfc2833

[2060]
type=fri end
username=20
secret=contraseña
host=dynami c
mai l box=20
```

```
context=puededdd
canreinvite=yes
dtmfmode=rfc2833
```

9.12.2 Paso 2 - Configurando el plan de discado

Ahora vamos a configurar el plan de discado para atender las especificaciones de la central. Todas las configuraciones de esta sección son hechas en el archivo extensions.conf.

Definiendo los ramales y salidas locales

```
; Este contexto permite conectar para ramales y comunicaciones locales
[ramales]
exten=>_20XX, 1, Dial(SIP/${EXTEN}, 20, r)

[local]
exten=>_0XXXXXXXX, 1, Dial(ZAP/g1/${EXTEN:1}, 20, Tt) ; comunicaciones locales
```

Definiendo la salida para DDD

```
[ddd]
; Discado para DDD al número de 7 dígitos (BrT)
exten=>_00XXXXXXXX, 1, Dial(Zap/g1/${EXTEN:1}, 20, Tt)

; Discado para DDD al número de 8 dígitos
exten=>_00XXXXXXXXXX, 1, Dial(Zap/g1/${EXTEN:1}, 20, Tt)
```

Definiendo la salida para DDI

```
[ddi]
; Discado para DDI
exten=>_000X., 1, Dial(Zap/g1/${EXTEN:1}, 20, Tt)
```

Definiendo las clases de ramal

```
[restricto]
include=>ramales
include=>local

[podeddd]
include=>ramales
include=>local
include=>ddd

[podeddi]
include=>ramales
include=>local
include=>ddd
include=>ddi
```

9.12.3 Recepción de las llamadas

Para recibir las llamadas, vamos a usar dos contextos uno para comunicaciones durante el expediente y otra para comunicaciones fuera del expediente. La comunicación va a ser atendida por un contestador automático y transferida directamente para un ramal o para algún otro menú.

Menús: Expediente y fuera de expediente

En los menús de abajo, el sistema tocará un mensaje avisando al originador de la llamada que la empresa se encuentra fuera de expediente, pero permitiendo que el usuario disque un ramal (eventualmente alguien está haciendo horas-extras).

```
[entrada]
i ncl ude=>expedi ente|08: 00-18: 00|mon-fri |*|*
; Este es el horario fuera de expediente
i ncl ude=>fuera deexpedi ente|18: 00-23: 59|*|*|*
i ncl ude=>fuera deexpedi ente|00: 00-07: 59|*|*|*
i ncl ude=>fuera deexpedi ente|*|sat-sun|*|*

[expedi ente]
exten=>s, 1, Goto(menupri nci pal , s, 1)

[fuera deexpedi ente]
exten=>s, 1, Background(fuera deexpedi ente)
exten=>s, 2, hangup()
exten=>i, 1, hangup()
exten=>t, 1, hangup()
i ncl ude=>ramal es
```

Menús: Principal y Ventas

Durante el expediente el usuario es recibido por el menú con contestador automático. El mensaje será algo como: *"Bienvenido, usted se comunica con la Nombre de la Empresa, Disque 1 para ventas, 2 para soporte y 3 para entrenamiento o el ramal deseado"*.

```
[gl obal s]
TELEFONI STA=SI P/2060
VENTAS=SI P/2035
SOPORTE=SI P/2004
ENTRENAMI ENTO=SI P/2036

[menupri nci pal ]
exten=> s, 1, Background(bi enveni do-menupri nci pal )
exten=>1, 1, Goto(ventas, s, 1)
exten=>2, 1, Goto(soporte, s, 1)
exten=>3, 1, Goto(entrenami ento, s, 1)
exten=>i, 1, Pl ayback(I nval i do)
exten=>i, 2, hangup()
exten=>t, 1, Di al (${TELEFONI STA}, 20, Tt)
i ncl ude=>ramal es

[ventas]
exten=>s, 1, Di al (${VENTAS}, 20, Tt)
```

```
[soporte]
exten=>s, 1, Di al (${SOPORTE}, 20, Tt)
```

```
[entrenamiento]
exten=>s, 1, Di al (${ENTRENAMIENTO}, 20, Tt)
```

Con esto la mayor parte de la funcionalidad de su plan de discado está lista y ahora usted puede comenzar a testear y mejorar su configuración. En la próxima sección vamos a ver más detalles sobre las aplicaciones más comunes.

Exclusivo para Vicente Miguel Perez Paredes

9.13 Cuestionario

1. Para incluir un contexto que depende del horario, usted puede usar:

include=> context|<times>|<weekdays>|<mdays>|<months>

El comando que sigue:

include=>expediente|08:00-18:00|mon-fri|*|*

- Ejecuta las extensiones de lunes a viernes desde las 08:00 hasta las 18:00.
- Ejecutan las opciones todos los días en todos los meses.
- El comando es inválido.

2. Cuando el usuario disca "0" para tomar la línea el Asterisk automáticamente corta el audio. Esto es malo, porque el usuario está acostumbrado a discar el "0" y oír el tono externo de discado. Para crear este comportamiento que el usuario está acostumbrado, se puede usar comando _____.

3. Los comandos:

```
exten => 8590/482518888, 1, Congesti on
exten => 8590, 1, Di al (Zap/1, 20)
exten => 8590, 2, Voi cema i (u8590)
exten => 8590, 102, Voi cema i (b8590)
```

Hace que un usuario que marco para la extensión 8590:

Reciba una señal de ocupado si el CallerID=482518888

Reciba una señal de ocupado independientemente del número discado

Va para el canal ZAP/1

Va para el VoiceMail() si el canal ZAP/1 estuviese ocupado o no atiende, excepto en el caso donde el CallerID fuera 482518888

4. Para concatenar varias extensiones basta separarlas con la seña ____.

5. Un menú de voz normalmente es creado con el comando inicial _____.

6. Usted puede incluir archivos dentro de sus archivos de configuración con el comando _____.

7. Asterisk permite que se use el plan de discado una base de datos basada en:

- Oracle
- MySQL
- Berkley DB
- PostgreSQL

8. Cuando usted usa el comando Dial(tipo1/identificar1&tipo2/identificar2) con varios identificadores, Asterisk disca para cada una en secuencia y espera 20 segundos o el tiempo de timeout antes de pasar para el otro número. La afirmación es:

- Falsa
- Verdadera

9. En el comando Background la música de fondo tiene que ser tocada enteramente antes que el usuario pueda digitar algo. La afirmación es:

- Falsa
- Verdadera

10. Los formatos válidos para el comando Goto() son:

- Goto (context,extension)
- Goto(context,extension,priority)
- Goto(extension,priority)
- Goto(priority)

11. Switches son usados para direccionar para otro PBX. La afirmación de arriba es:

- Correcta
- Incorrecta

12. Un macro puede ser usado para automatizar una serie de operaciones en secuencia para una extension específica. El primer argumento pasado por la llamada del macro es el:

- \${ARG1}
- \${ENV1}
- \${V1}
- \${X}

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

Configurando recursos avanzados

En este capítulo veremos algunos recursos avanzados del Asterisk PBX.

10.1 Objetivos

Al final de este capítulo usted debe estar apto para configurar los siguientes recursos.

- Gerenciamiento de llamadas.
 - Call Parking (Estacionamiento de llamadas).
 - Call Pickup (Captura de llamadas).
 - Call Transfer (Transferencia).
 - Call Conference (Conferencia simple).
 - Call Recording (Grabación de las llamadas)
- Sala de conferencia.
- Música en espera.

10.2 Soporte a los recursos de PBX

En esta sección vamos a ver una serie de recursos que fueron introducidos en la versión 1.0 de Asterisk para el tratamiento de llamadas. El archivo `features.conf` controla la mayor parte de estos recursos.

Es importante entender que algunos recursos son implementados por el teléfono y otros por Asterisk. Otro punto importante es el caso de teléfonos IP basados en SIP con la configuración `canreinvite=yes`. En este caso la comunicación de media ocurre entre los dos teléfonos, si el usuario presiona #, quien va a interpretar es el otro teléfono y no necesariamente el Asterisk.

10.2.1 Recursos con soporte para todo tipo de canal

- **Música en espera:** Vea la aplicación `MusicOnHold()`.

- **Estacionamiento de llamadas:** Soportado en la instalación estándar.
- **Captura de llamadas:** Soportado en la instalación estándar (*8).
- **Grabado de llamadas:** Usando la aplicación Monitor().
- **Conferencia:** Usando la aplicación MeetMe().

10.2.2 Recursos con soporte en teléfonos SIP

- **Llamada en espera:** Normalmente implementado por el teléfono.
- **Transferencia a ciegas:** Implementado en el * opcional del teléfono.
- **Llamada en espera con consulta:** Implementado en el teléfono
- **Seguimiento incondicional:** *
- **Seguimiento si no se atiende:** Creado en el plan de discado.
- **Seguimiento si estuviese ocupado:** Creado en el plan de discado.
- **Transferencia asistida:** *
- **Llamada de a tres:** Normalmente implementado por el teléfono.
- **Filtrado de las llamadas:** Creado en el plan de discado.
- **Rediscado automático:** Es posible usando AGI.
- **Rediscado manual:** Teléfono
- **No molestar:** *
- **Mensaje en espera:** *, pero debe ser soportado en el teléfono.
- **Indicación de llamada en espera:** *, debe ser soportado en el teléfono.

10.2.3 Para teléfonos analógicos (Zap).

- **Captura de llamadas:** Soportada en la instalación estándar.
- **Llamada en espera:** Normalmente implementado en el teléfono.
- **Transferencia a ciegas:** *
- **Llamada en espera con consulta:** Implementado en el teléfono.
- **Seguimiento incondicional:** *
- **Seguimiento si no se atiende:** Creado en el plan de discado.
- **Seguimiento si estuviese ocupado:** Creado en el plan de discado.
- **Transferencia asistida:** *
- **Llamada de a tres:** Normalmente implementado por el teléfono.
- **Filtrado de las llamadas:** Creado en el plan de discado.
- **Rediscado automático:** Es posible usando AGI.
- **Rediscado manual:** *
- **No molestar:** *
- **Mensaje en espera:** *, pero debe ser soportado en el teléfono.
- **Indicación de llamada en espera:** *, debe ser soportado en el teléfono.
- **Encuentreme:** *

10.3 Estacionamiento de llamadas

10.3.1 Descripción:

Es usado para estacionar una llamada. Esto nos auxilia cuando usted está atendiendo en un teléfono fuera de su sala. Estaciona la llamada en una determinada extensión y cuando se mueve de vuelta hacia su sala digita la extensión donde la llamada está estacionada.

Por default la extensión 700 es usada para estacionar una llamada. En el medio de una conversación, presione # para iniciar una transferencia, entonces disque 700. Asterisk ahora anunciará la extensión de estacionamiento, pero probablemente 701 o 702. Cuelgue, y el originador de la llamada será colocado en espera en la extensión anunciada. Vaya para un teléfono diferente y disque 701 (la extensión anunciada) y entonces podrá continuar conversando. Si el originador permanece estacionado por más tiempo que el límite entonces Asterisk hará sonar la extensión discada originalmente.

10.3.2 Lista de tareas para la configuración

1. Incluir la línea (include=>parkedcalls) en el archivo extensions.conf.
2. Testeé usando #700
3. Configure el archivo features.conf si usted quisiera alterar los estándares de funcionamiento del estacionamiento de las llamadas.

10.3.2.1 Habilitando llamadas en estacionamiento: (obligatorio)

Incluya en el archivo extensions.conf las líneas

```
include=>parkedcalls
```

10.3.2.2 Testeé el estacionamiento transfiriendo para el ramal 700 usando #700.

Notas:

- La extensión de estacionamiento no será mostrada en el comando dialplan show.
- Es preciso reiniciar el asterisk después de modificar el archivo features.conf. Un reload simple no funcionará.
- Es preciso poder transferir para poder estacionar una llamada, entonces verifique las opciones t y T del comando Dial().
- Las transferencias basadas en el Asterisk funcionan solamente si el Asterisk estuviera en el medio del camino (canreinvite=no en el sip.conf por ejemplo).

10.4 Captura de llamadas - Call-pickup

10.4.1 Descripción

La captura permite que usted empuje un llamado de un colega en el mismo grupo de llamadas. Esto evita que usted tenga que levantar para atender un teléfono de su vecino que no para de sonar.

Discando *8 usted puede capturar una llamada en su grupo de llamadas. Este recurso puede ser alterado por la vía del archivo features.conf.

10.4.2 Lista de tareas para habilitar la captura.

1. Configure el grupo para que sus extensiones pertenezcan dentro de uno de los siguientes archivos sip.conf, iax.conf, zapata.conf (obligatorio).

```
[4x00]
cal l group=1
pi ckupgroup=1, 2
```

2. Configure el archivo features.conf (opcional)

```
pi ckupexten=*8; Configura la extensión de captura. El estándar es *8
```

10.5 Transferencia de llamadas - Call Transfer

10.5.1 Descripción

La transferencia de llamadas puede ser hecha de dos formas. A transferencia a ciegas (blind transfer) y la transferencia asistida.

En la transferencia a ciegas usted usa los dígitos que estuviesen definidos en el parámetro **blindxfer** en el archivo features.conf. Por ejemplo, si el **blindxfer** fuese configurado como "#1", usted disca este número y el número de destino y cuelga el teléfono.

El **atxfer** permite la transferencia asistida o con supervisión. Funciona de la siguiente manera:

Durante una conversación, usted disca la secuencia de transferencia asistida #2, El asterisk pronuncia "transfer" y entonces le da a usted un tono de discado, en tanto coloca el originador en música de espera. Usted disca para quien va a transferir y habla con él para introducir la llamada, entonces cuelga, la transferencia queda así completa. Si la persona transferida no quisiese atender, ella simplemente cuelga el teléfono y usted recibirá la llamada.

10.5.2 Lista de tareas para configurar

1. Si el teléfono fuese estándar SIP asegúrese que la opción `canreinvite=no` está habilitada (obligatorio).
2. En la instrucción Dial certifíquese que las opciones `t` o `T` estén habilitadas.

10.6 Conferencia - Call Conference

La conferencia en Asterisk es hecha usando el aplicativo `MeetMe()`. El `MeetMe()` es un puente de videoconferencia, muy simple de usar con la ventaja de que funciona con cualquier tipo de canal. Algunos teléfonos estándar SIP, por ejemplo, tienen sus propios mecanismos de conferencia que también pueden ser usados, pero el método estándar para conferencia es el `MeetMe()`. Vamos a dar una ojeada más en profundo en el aplicativo.

10.6.1 Formato:

```
MeetMe(Confno[|options][|password])
```

10.6.2 Descripción:

Coloca al usuario en una sala de conferencia específica. Si el número de conferencia fuese omitido, será pedido uno al usuario.

El string da la opción de poder contener o no uno o más de los siguientes caracteres.

- 'm'—Configura el modo oír apenas.
- 't'—Configura el modo apenas habla
- 'i'— Anuncia la entrada y salida de usuario.
- 'p'—Permite al usuario salir de la conferencia presionando #.
- 'X'—Permite al usuario salir de la conferencia discando una extensión válida de un dígito.

La extensión es ejecutada en `_${MEETME_EXIT_CONTEXT}` o en el contexto actual si la variable no haya sido definida.

- 'v'—modo vídeo.

- 'q'—modo quieto (no toca los sonidos de entrada y salida).
- 'd'—adiciona dinámicamente la conferencia.
- 'D'—dinámicamente adiciona la conferencia, pidiendo por pin (personal identification number). Cuando oye el pedido del pin, si el usuario no quisiese un pin designado para la conferencia él deberá discar '#'.
- 'M'—Habilita música en espera cuando la conferencia tiene un único usuario.
- 'e'—Selecciona una conferencia vacía.
- 'E'—Selecciona una conferencia vacía sin pin.
- 'b'—Corre el script AGI especificado en `#{MEETME_AGI_BACKGROUND}` el estándar es `conf-background.agi` (para canales zap solamente, no funciona con canales que no sean zap en la misma conferencia).
- 'a'—Configura el modo de administración.
- 'A'—Configura el modo marcado.
- 's' —Presenta un menú (usuario o administrador) cuando '*' es recibido (envía para el menú).
- 'x'—Fecha la conferencia y cuelga en todos los otros cuando el ultimo usuario marcado sale.
- 'w'—Espera hasta que el usuario marcado entre en la conferencia antes de permitir que alguien hable. Todos los otros usuarios conectados van a oír música en espera hasta que el usuario marcado entre.

10.6.3 Códigos de retorno:

La aplicación siempre retorna -1.

10.6.4 Detalle de las opciones:

Ni la opción 's' ni la opción 'p' funcionan si son usados junto con 'X'.

- Usuario (sin la opción 'a' siendo configurada): Al presionar * suena el menú de voz "Por favor, presione 1 para enmudecer y volver".
- Admin (con la opción a sido configurada): Al presionar * toca el menu de voz "Por favor, presione 1 para enmudecer/desenmudecer, 2 para trabar/destrabar esta conferencia".

10.6.5 Lista de tareas de configuración

1. Escoja las extensiones para la sala de conferencia (obligatorio)
2. Incluya el comando MeetMe() en el archivo extensions.conf (obligatorio)
3. Edite el archivo meetme.conf para configurar las contraseñas (opcional)

10.5.6 Ejemplos

Ejemplo 1: Sala de conferencia simple

1. Coloque en el archivo extensions.conf la sala de conferencia 101.

```
exten=>500, 1, MeetMe(101|123456)
```

2. Defina en el archivo meetme.conf la contraseña (pin) 123456 en la extensión 500

Ejemplo 2 – Sala de conferencia autenticada

- 1 – Defina la sala de conferencia en el archivo extensions.conf

```
exten=> 18, 1, Answer
exten=> 18, 2, Wait(1)
exten=> 18, 1, Answer
exten=> 18, 2, Wait(1)
exten=> 18, 3, Authenticate(5678)
exten=>18, 4, MeetMe(18|p)
exten=>18, 5, Playback(vm-goodbye)
exten=>18, 6, Hangup
```

- 2 – Defina en el archivo meetme.conf las contraseñas para cada extensión.

Nota importante:

La aplicación MeetMe precisa de un temporizador para funcionar. Existen diferentes maneras de hacer un temporizador para que funcione, pero no va a funcionar por default si usted no tiene un hardware Digium Zaptel instalado. En este momento, apenas dispositivos zaptel pueden ser

usados. Si usted no tiene un dispositivo Zaptel use el driver ztdummy para temporización.

10.5.7 Archivo de configuración del MeetMe()

Este archivo es usado para configurar el modulo de conferencia del MeetMe.

Ejemplo:

```
;
; Archivo de configuración para las salas de conferencia del Meetme()
;
;
[rooms]
; Formato: conf=>confno, pincode
;
conf=>9000
conf=>9001, 123456
```

Actualmente no es necesario usar ni el comando reload, ni el comando 'restart' para que Asterisk note los cambios en el archivo meetme.conf.

La aplicación Meetme cuenta con dos aplicativos de apoyo:

```
MeetMeCount(confno[|var])
```

Toca el número de usuarios en una conferencia MeetMe. Si la variable fuese especificada, el mensaje no será tocado y la variable recibirá el valor de cuenta de usuarios.

```
MeetMeAdmin(confno, command, user):
```

Corre el comando admin para la conferencia especificada en confno.

- 'K' Tira todos los usuarios de la sala de conferencia.
- 'k' Tira un usuario de la sala de conferencia.
- 'L' Traba la conferencia.
- 'l' Destraba la conferencia.
- 'M' Enmudece la conferencia.
- 'm' Desenmudece la conferencia

Aplicaciones avanzadas. Si usted desea, es posible todavía usar la aplicación MeetMe2 que posee una interfase gráfica para el control de la

conferencia, o también el `app_conference` que no precisa de un temporizador y es más eficiente en términos de traducción de codecs.

10.7 Grabando una comunicación

Existen diversas formas de grabar una comunicación en Asterisk. Un recurso interesante es el uso de `Automon`. Este permite que simplemente discando *1 usted pueda accionar dinámicamente la grabación de llamadas.

Ejemplo:

```
exten=>_4XXX,1,Set(DYNAMIC_FEATURES=automon)
exten=>_4XXX,2,Dial(SIP/${EXTEN},20,jtTW);wW habilita la grabación.
```

Los canales de audio de entrada (IN) y salida (OUT) serán separados en dos archivos y estarán disponibles en su servidor Asterisk en el siguiente directorio.

```
/var/spool/asterisk/monitor
```

Los dos archivos pueden ser unidos usando el aplicativo `sox`.

```
soxmix *in.wav *out.wav output.wav
```

Si usted no quisiese usar la aplicación `Set` antes de cada comando `Dial()` usted puede usar la sección "globals".

```
[globals]
DYNAMIC_FEATURES=>automon
```

10.8 Música en espera - Music on hold

En la versión 1.2 permaneció bastante simplificado el uso de música en espera. Ya no es necesario usar el `mpg123` como en la versión 1.0.x. Siga los pasos de abajo para habilitar la música en espera.

Edite los archivos para habilitar música en espera

En el archivo `/etc/asterisk/zapata.conf`, adicione la línea:

```
[channel s]
music hold d=default
```

Una cosa importante es descompactar los archivos en MP3 para el formato RAW. Esto va a reducir bastante la carga de CPU usada por la música en espera porque no será necesaria decodificación MP3 en tiempo real.

Si usted compila el sox con soporte a MP3 usted puede usar los comandos de abajo.

```
sox -c 1 ejemplo.mp3 -t raw -r 8000 -c 1 -s -w ejemplo.raw
```

Si usted tuviese el sox sin soporte a mp3 usted puede usar:

```
$lame --decode ejemplo.mp3 ejemplo.wav
$ sox -V ejemplo.wav -r 8000 -c 1 -w ejemplo.raw
```

Edite el archivo /etc/asterisk/musiconhold.conf

```
[default]
mode=files
directory=/var/lib/asterisk/mohmp3
```

Usted debe reiniciar el asterisk porque ha alterado el archivo musiconhold.conf

Coloque sus archivos en formato .raw en el directorio /var/lib/asterisk/mohmp3

En el plan de discado, usted debe configurar la música en espera como en el ejemplo de abajo:

```
exten=>100, 1, SetMusicOnHold(default)
exten=>100, 2, Dial(Zap/2)
```

Ejemplos:

Configurando el archivo extensions.conf para testear MusicOnHold

```
[local]
exten => 6601, 1, WaitMusicOnHold(30)
```

10.9 Cuestionario

1. Con relación a Call Parking asigne las afirmaciones verdaderas.

- Por Default la extensión 800 es usada para Call Parking
- Cuando fuese para el otro teléfono la comunicación disque 700 para recuperar a llamada
- Por Default la extensión 700 es usada para Call parking
- Digite la extensión anunciada para recuperar la llamada

2. Para que el Call Pickup funcione es preciso que las extensiones estén en el mismo _____. En el caso de extensiones Zap esto es configurado en _____.

3. En el caso de transferencia de llamadas existen las transferencias _____, donde el ramal de destino no es consultado antes y la transferencia _____ donde es posible verificar si el usuario está en la extensión.

4. Para hacer una transferencia asistida usted usa el ___ en tanto para hacer una transferencia a ciegas (blind) usted usa ____.

- #1, #2
- #2, #1
- #3, #1
- #4, #2

5. Para hacer conferencia en Asterisk es necesario usar el aplicativo _____.

6. Si fuese necesario administrar una conferencia, usted puede usar el aplicativo _____ y tirar un usuario de la sala.

- MeetMe()
- MeetMeConsole()
- MeetMeAdministrator()
- MeetmeAdmin()

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

DAC Distribución automática de Llamadas.

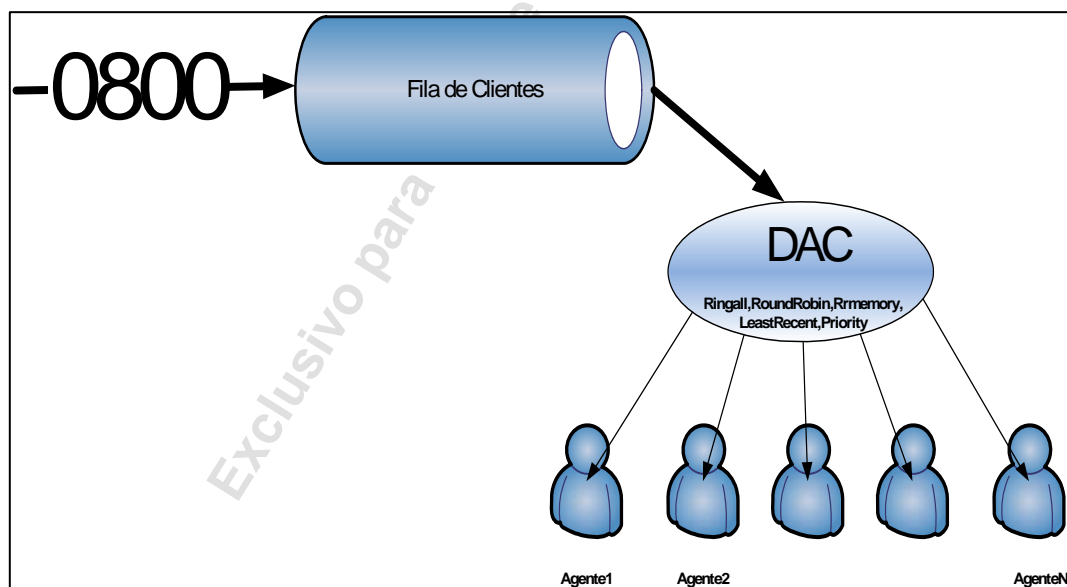
11.1 Objetivos

Al final de este capítulo usted deberá ser capaz de:

- Comprender la utilidad de DAC.
- Comprender su funcionamiento básico.
- Instalar y configurar DAC

11.2 Introducción

Las Filas de atención de llamadas permiten que estas llamadas de entrada en el PBX puedan ser tratadas de forma más eficiente. De una forma general los siguientes pasos son ejecutados en una fila de distribución de llamadas.



1. Las llamadas de entrada son colocadas en la fila.
2. Miembros de la fila atienden (extensiones que autenticarán como agentes).
3. Una estrategia de gerenciamiento y distribución de la fila es usada.

4. Se hace sonar música en tanto el usuario espera en la fila.
5. Son hechos anuncios para miembros y para quien está en la fila.

La principal aplicación para filas es la atención de llamadas dentro de una empresa o Call Center. Usando filas usted se evita de perder llamadas cuando sus agentes están ocupados y mejora la productividad en la medida en que solo se distribuye para aquellos que están listos para atender.

Otro punto clave de las filas de atención de llamadas y distribución automática de llamadas son las estadísticas que pueden ser generadas como TME (tiempo medio de espera), TMA (tiempo medio de atención de llamadas), llamadas abandonadas, que permiten que usted pueda dimensionar científicamente su atención.

Las filas son definidas en el archivos "**queues.conf**". Los agentes son personas que responden a las llamadas que fueron colocadas en una fila específica. Un agente se autentica indicando que está listo para recibir una llamada, los agentes son definidos en el archivos **agents.conf**. El Asterisk transfiere una llamada recibida para la fila, que es así transferida para un agente disponible.

11.3 Miembros

Miembros son aquellos canales que están activos y respondiendo a la fila. Los miembros pueden ser canales directos (ej. Teléfonos conectados al Asterisk). Usted puede definir también miembros como individuos que se autentican de cualquier conexión para recibir una llamada. Los Agentes son definidos en el archivo **agentes.conf**. El login de los agentes es hecho a través de la aplicación agentlogin.

11.4 Estrategias

Las llamadas son distribuidas entre los miembros de una fila de acuerdo con una entre varias estrategias.

- **ringall:** Hace sonar todos los canales disponibles hasta que uno los atiende.
- **roundrobin:** Distribuye igualmente las llamadas por las interfaces.
- **leastrecent:** Distribuye para la interfase que menos recibió llamadas.
- **fewestcalls:** Toca aquella con menos llamadas completadas.

- **random:** Toca una interface aleatoria.
- **rrmemory:** Round-robin con memoria, recuerda donde dejo la última llamada.

11.5 Menú para el usuario

Usted puede definir un menú para el usuario, mientras él este esperando. Para este menú, usted puede usar extensiones de un dígito. Para habilitar esta opción, defina el contexto para el menú en la configuración de la fila.

11.6 Nuevos recursos

Queue() ahora tiene opciones para **penalidad (penalty)**, **tiempo para ordenarse (wrapuptime)** y **prioridad (priority)**. Estos recursos fueron adicionados al sistema de filas de Asterisk.

Prioridad funciona de la siguiente manera:

Los nuevos recursos permiten que las filas actúen como filas de prioridad, mas allá del modo actual FIFO (primero que entra es el primero que sale). Esto da la habilidad de enfilear una llamada no en el fin de la fila, pero en cualquier lugar de esta de acuerdo con la prioridad de la llamada.

Ahora usted puede tener una única fila atendiendo las llamadas ordenadas desde las más importantes hacia las menos importantes. La prioridad de una llamada entrando en la fila es determinada por una variable especial, QUEUE_PRIO. Valores más altos de la variable significan mayor prioridad. Si la variable no recibe un valor, todas las llamadas tienen la misma prioridad, por default (FIFO).

Ejemplo:

Clientes importantes

```
exten=>111, 1, PI ayback(wel come)
exten=>111, 2, Set(QUEUE_PRI 0=10)
exten=>111, 3, Queue(support)
```

Clientes menos importantes

```
exten=>112, 1, PI ayback(wel come)
exten=>112, 2, Set(QUEUE_PRI 0=5)
exten=>112, 3, Queue(support)
```

11.7 Configuración

11.7.1 Lista de Tareas

1. Crear la fila de atención de llamadas (obligatorio).

2. Configurar los agentes (opcional).
3. Crear los agentes de atención de llamadas (opcional).
4. Colocar la fila en el plan de discado (obligatorio).
5. Configurar el grabado de los agentes (opcional).
6. Verificar el funcionamiento con agent show y queue show.

11.7.2. Crear la fila de atención de llamadas.

queues.conf

```
[tel emarketing]
music = default
; announce = queue-markq
; context = qoutcon
timeout = 2
retry = 2
maxlen = 0
member => Agent/300
member => Agent/301
[audi toria]
music = default
; announce = queue-markq
; context = qoutcon
timeout = 15
retry = 5
maxlen = 0
member => Agent/600
member => Agent/601
```

11.7.3 Definir parámetros de los agentes.

agents.conf

```
[agents]
; Define autologoff si es apropiado. Esto es cuanto tiempo el teléfono va ha
; tocar sin ; respuesta antes que el agente sea automáticamente desconectado
; de la fila (en ; segundos). Note que si este valor fuese mayor que el valor
; de timeout de la fila, los ; agentes no serán removidos.
autologoff=15
; Define ackcall para pedir una confirmación por '#' cuando un
; agente se loguea sobre agentcallpark. Por default es "yes".
; ackcall=yes
; Define el tiempo de wrapuptime. Esta es la cantidad mínima de tiempo que
; el agente tiene después de desconectar antes que posea recibir una nueva
; comunicación.
; Es el tiempo que el agente tiene por ejemplo para dar marcha al pedido
; del cliente. (En milisegundos).
; wrapuptime=5000
; Define la música en espera para los agentes.
; music hold => music_class
music hold => default
; Un beep opcional para hacer sonar para los agentes con conexión
; permanente
; custom_beep=beep
```



```
; Grupos para agentes (puede cambiar en el medio del archivo apenas);
group=1
```

11.7.4 Crear los agentes en el archivo

agents.conf

```
; agent => agentid, agentpassword, name
[agents]
agent => 300,300,Test Rep - 300
agent => 301,301,Test Rep . 301
agent => 600,600,Test Ver - 600
agent => 601,601,Test Ver . 601
```

11.7.5 Colocar la fila en el plan de discado.

extensions.conf

```
; Fila telemarketing.

exten=>_0800XXXXXXXX,1,Answer
exten=>_0800XXXXXXXX,2,SetMusicOnHold(default)
exten=>_0800XXXXXXXX,3,DigitTimeout,5
exten=>_0800XXXXXXXX,4,ResponseTimeout,10
exten=>_0800XXXXXXXX,5,Background(welcome)
exten=>_0800XXXXXXXX,6,Queue(telemarketing)

; Transfiere para la fila auditoria
exten => 8000,1,Queue,(Auditoria) ; Conecta a auditoria
exten => 8000,2,Playback(demo-echotest) ; Ningun auditor está disponible
exten => 8000,3,Goto(8000,1) ; Verifica el auditor nuevamente

; Login de los agentes para la fila telemarketing (0800) y auditoria (8000)
exten => 9000,1,Wait,1 ; Levemente pausada, no necesaria
exten => 9000,2,AgentLogin()
; recibirá 'login inválido' entre con el número de agente.
```

11.7.6 Configurar la grabación

queues.conf

```
; Habilita la grabación de llamadas direccionadas a los agentes.
recordagentcalls=yes

; Formato usado para grabar las llamadas (wav, gsm, wav49). El default es
; "wav".
recordformat=gsm

; Inserte en el campo usuario del CDR el nombre de la grabación creada
; Default, desconectado
; createlink=yes

; Texto a ser adicionado a la grabación, permite formar un link url.
; urlprefix=http://host.domain/calls/
; En directorio opcional para salvar las conversaciones
; El estandar es: /var/spool/asterisk/monitor
savecallsin=/var/calls
```

11.7.7 Aplicaciones de apoyo para las filas.

- **AddQueueMember**: Dinámicamente adiciona un miembro en la fila.
- **RemoveQueueMember**: Dinámicamente remueve un miembro de la fila.
- **AgentLogin**: Login de agente en la fila de llamadas.
- **AgentCallbackLogin**: Otro tipo de login.

Línea de comando en la interfase de línea de comando

- **Agent show**: Muestra los agentes.
- **Show queues**: Lista todos los agentes.
- **Queue show <name>**: Muestra datos de una fila específica.

11.8 Funcionamiento de las filas

11.8.1 Login del Agente

La aplicación de AgentLogin es usada para lograr los agentes. A través del uso de filas, usted puede enviar llamadas para quienes las atienden.

Sintaxis:

```
AgentLogin([agentnumber])
```

Nota: Si el número agente no es pasado, un mensaje de login inválido será tocado y pedirá el número de agente y contraseña.

Puede ser usado también el aplicativo AgentCallbackLogin(), en este caso el agente colocará el teléfono para colgarlo y cuando alguien estuviese en

la fila el teléfono del agente será llamado, en vez de estar todo el tiempo abierto como el AgentLogin().

Ejemplo: Discado de salida pasado a los agentes, basado en filas.

Parte 1: Login del agente

Un agente de la fila telemarketing descuelga el teléfono y disca #9000, él oye un mensaje de login inválido (y es pedido el número y la contraseña). La fila auditoria sigue el mismo proceso. La fila en la cual el termina depende de la separación en agentes.conf.

Parte 2: Fila

Una vez en la fila, el agente oirá una MOH (música en espera, si fue definida), o nada en caso contrario. Cuando una conexión ocurre, la llamada es conectada a un agente en la fila telemarketing. El agente oirá un beep (la MOH dejara de sonar si estuviese tocando), y será conectada a esa llamada.

Parte 3: Término de la llamada

Cuando un agente termina con la llamada el puede:

- Presionar * para desconectar y permanecer en la fila telemarketing.
- Colgar el teléfono, el agente será desconectado de la fila telemarketing.
- Presionar #8000 y la llamada será transferida para la fila auditoria.

Exclusivo para Vicente Miguel Perez Paredes

11.9 Cuestionario

1. Cite cuatro estrategias de ruteamiento del sistema de fila de atención de llamadas.

2. Es posible grabar la conversación de los agentes usando _____ en el archivo agents.conf.

3. Para loguear un agente se usa el comando Agentlogin([agentnumber]). Cuando el agente termina la llamada este puede presionar:

- * para desconectarse y permanecer en la fila
- Colgar el teléfono y ser desconectado de la fila
- Presionar #8000 y la llamada será transferida para la fila de auditoria
- Presionar # para colgar.

4. Las tareas obligatorias para configurar una fila de atención de llamadas son:

- Crear la fila
- Crear los agentes
- Configurar los agentes
- Configurar la grabación
- Colocar en el plan de discado

5. Cual es la diferencia entre los aplicativos AgentLogin() y AgentCallBackLogin().

6. Cuando en una fila de atención de llamadas, usted puede definir un determinado número de opciones que el usuario puede discar. Esto es factible incluyendo un _____ en la fila.

- Agente
- Menú
- Contexto
- Aplicativo

7. Las aplicaciones de apoyo `AddQueueMember()`, `AgentLogin()`, `AgentCallBackLogin` y `RemoveQueueMember()` deben ser incluidas en:

- el Plan de discado
- la interfase de línea de comando
- en el archivo `queues.conf`
- en el archivo `agents.conf`

8. Es posible grabar los agentes, pero para esto e preciso de un grabador externo.

- Verdadero
- Falso

9. "Wrapuptime" es el tiempo que el agente precisa después del término de una llamada para prepararse para la próxima o completar procesos en relación a la llamada atendida

- Verdadero
- Falso

10. Una llamada puede ser priorizada dependiendo del CallerID dentro de una misma fila. La afirmación es:

- Correcta
- Incorrecta

Exclusivo para Vicente Miguel Peres Paredes

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

El Buzón de voz

12.1 Objetivos

Al final de este capítulo usted debe ser capaz de:

- Comprender para que sirve el Buzón de voz.
- Comprender el funcionamiento básico.
- Instalar y configurar el Buzón de voz.
- Enviar el buzón de voz para el e-mail.

12.2 Introducción

El recurso de buzón de voz permite que una llamada ocupada o no atendida sea enviada para una secretaria electrónica o contestador automático. Un mensaje de aviso con audio anexo es enviado para el receptor del mensaje. Es uno de los recursos estándar de Asterisk. Es interesante que sistemas en una arquitectura de conmutación de circuitos fueran antes mucho más costosos y ahora con telefonía IP se tornan estándar.

12.3 Lista de tareas para la configuración

Para configurar el buzón de voz usted debe seguir los siguientes pasos:

1. Configure el archivo `voicemail.conf` para crear los usuarios y contraseñas del voicemail (obligatorio).
2. Configure el archivo `extensions.conf` para que una llamada ocupada o no atendida sea direccionada para el utilitario `VoiceMail()`. (Obligatorio).
3. Configure una extensión para que el usuario pueda oír su buzón de voz y administrarlo.

12.3.1 Configurando el archivo `voicemail.conf`

```
[general ]
cú di nternal contexts = house-admin, house-toll, house-local, house-intercom-
only

[default]
```

```
1234=>1234, SomeUser, email@address.com, pager@address.com, saycid=yes | dialout=
fromvm | callback=fromvm | review=yes | operator=yes
```

Veremos más opciones de aquí en adelante.

12.3.2 Configurando el archivo extensions.conf

Usted puede usar una macro para crear las extensiones.

```
[global]
PHONE1=Zap/1
PHONE2=SI P/6002

[macro-one-line]
exten=>s, 1, Dial (${ARG1}, 20, t)
exten=>s, 2, Voicemail (u${MACRO_EXTEN})
exten=>s, 3, Hangup
exten=>s, 102, Voicemail (b${MACRO_EXTEN})
exten=>s, 103, Hangup

[local]
exten=>6601, 1, Macro(one-line, ${PHONE1})
exten=>6602, 1, Macro(one-line, ${PHONE2})
```

12.3.3 Usando la aplicación VoiceMailMain()

```
exten=>9000, 1, VoicemailMain()
```

Cuando usted configura esta línea en los archivos extensions.conf, o VoiceMailMain() va a llevarlo para un menú de opciones donde usted podrá ejecutar una serie de opciones:

Menú

- 0 Opciones de casilla postal
 - 1 Graba su mensaje de no disponible
 - 2 Graba su mensaje de ocupado
 - 3 Graba su nombre
 - 4 Graba su mensaje temporáneo
 - 5 Cambia su contraseña
 - * Retorna al menú principal
- 1 Lee los mensajes de voicemail
 - 3 Opciones avanzadas
 - 1 Reply
 - 2 Call Back
 - 3 Envelope
 - 4 llamada de salida
 - 4 Toca el mensaje anterior
 - 5 Repite el mensaje actual
 - 6 Toca el próximo mensaje
 - 7 Apaga el mensaje actual
 - 8 Reencamina el mensaje para otra casilla
 - 9 Salva el mensaje para un folder (carpeta)

- * Ayuda
- * Salir
- 2 Cambios de pistas
- Ayuda
- # Salida

Después de grabar un mensaje (mensaje de entrada, ocupado, no disponible o nombre)

1 – Acepta

2 – Revisión

3 – Re-grabar

0 – Va hacia la telefonista

12.3.4 Sintaxis del Voicemail():

```
Voicemail ([flags]boxnumber[@context][&boxnumber2[@context]][&boxnumber3])
```

Graba el canal, salvando un archivo de audio en un número dado de casilla postal, que debe estar configurado en el archivo voicemail.conf. El número de casilla postal debe estar precedido por una o más opciones.

s: La letra s, si esta presente, hace que las instrucciones ("Please leave your message after the tone. When done, hang up, or press the pound key.") no sea ejecutada. (En Castellano "Por favor, deje su mensaje después del bip, cuando termina, cuelgue o presione la tecla sosteniendo")

u: La letra u, si esta presente, hace que el mensaje no disponible sea tocado. Por default, el mensaje dice, "La persona en la extensión 1234.... no está disponible" pero el propietario de la casilla postal puede grabar su propio mensaje de no disponible con el comando **Voicemailmain**.

b: La letra b, si esta presente, hace que el mensaje de ocupado sea tocado. Por default el mensaje dice: "La persona en la extensión 1234 está ocupada"

Usted no puede especificar ambos la "u" y la "b" juntos. Usted puede en tanto si, combinarlos con s, dando seis posibilidades.

- s: No hace sonar nada
- (Sin opciones): Toca instrucciones

- su: Toca mensaje de no disponible
- u: Toca mensaje de no disponible, entonces instrucciones
- sb: Toca un mensaje de ocupado
- b: Toca un mensaje de ocupado, entonces instrucciones.

En todos los casos, el archivo beep.gsm será tocado, antes de comenzar a grabar.

Los mensajes de buzón de voz serán grabados en el directorio inbox para aquellos números de caja postal de buzón de voz.

/var/spool/asterisk/voicemail/context/boxnumber/INBOX/

Si el originador presiona 0 (cero) durante el anuncio, este será movido para la extensión 'o'(Out) en el contexto actual del voicemail. Esto puede ser usado como salida para la recepcionista. Si usted no quiere esto, apenas configure la extensión para volver al voicemail, sino la llamada será colgada.

También durante el audio, si el originador presiona:

'*'—La llamada salta para la extensión 'a' en el contexto actual del buzón de voz.

'#'-El mensaje de bienvenida y/o instrucciones son paradas y la grabación comienza inmediatamente.

Cuando se usa el cero '0' y la estrella '*' es importante notar que el contexto que usted colocó la aplicación voicemail es irrelevante, el contexto donde fue colocada la casilla postal del buzón de voz será usada para saltar para las extensiones 'a' u 'o'.

Si la caja postal no existe, y existe la prioridad n+101, entonces esta prioridad será la próxima.

Si durante la grabación el originador presiona '#' o el limite definido de silencio que este definido, la grabación es parada y la llamada continua en la prioridad+1.

Certifíquese de hacer algo con el originador después que este vuelva del voicemail(), seria algo así:

```
exten=>somewhere, 5, PI ayback(Goodbye)
exten=>somewhere, 6, Hangup
```

12.3.5 Códigos de retorno

Retorna -1 en errores de casilla postal no encontrada, si el usuario cuelga. Sino retorna 0.

12.4 Archivos de configuración del VoiceMail.

12.4.1 Configuraciones de la sección [general].

Las configuraciones de la sección general se aplican para todos los usuarios. Las siguientes configuraciones están disponibles.

attach

hace que asterisk copie un mensaje de voicemail para un archivo de audio y lo envíe al usuario como un anexo en un e-mail. El estándar es no (no).

delete

Si se ha configurado 'yes' el mensaje será borrado de la casilla de voicemail (después de haber sido enviada por e-mail).

Nota: Esta configuración no siempre funciona como una configuración global. Es recomendado que usted coloque esto en cada casilla postal que el mensaje debe ser borrado después de ser enviado por e-mail.

mailcmd

Mailcmd permite al administrador, alterar el comando de e-mail por un comando definido por el usuario. Mailcmd recibe como comando un string con la línea de comando deseada. El estándar es '/usr/sbin/sendmail -t'

maxsilence

Maxsilence define cuanto tiempo Asterisk esperará por un período continuo de silencio antes de terminar una llamada para el voicemail. El valor estándar es 0, lo que significa que el detector de silencio permanece deshabilitado y el tiempo de espera es infinito. Maxsilence recibe un número entero que es el número de segundos que este esperará antes de desconectar.

envelope

Envelope controla si Asterisk tocará o no tocará un mensaje de envelope (data/hora) antes de tocar el mensaje de voicemail. (yes/no).

externnotify

Quiere correr un programa externo siempre que un llamador deja un mensaje en el buzón de voz para el usuario. Existe el comando `externnotify`. Recibe el valor de un string con el comando a ser ejecutado.

`silencethreshold`

Las veces cuando usamos el **maxsilence**, es necesario ajustar la detección de silencio para eliminar un disparo falso causado por ruido de fondo. Números mayores permiten más ruido de fondo. Es necesario experimentar hasta llegar a los valores apropiados.

`serveremail`

Esta configuración puede ser usada para identificar la fuente de una notificación de mensaje de voz. El valor es un string que puede ser codificado de una de dos formas. Si el string está en la forma *alguien@host.com*, entonces el string será usado con dirección fuente en las notificaciones por e-mail. Si el string estuviera en la forma de *alguien*, entonces el nombre del host de la máquina que corre asterisk será adicionado al final del string después de la inserción de @.

`maxmessage`

Define el tiempo máximo en segundos de un mensaje de entrada. Use cuando el espacio en disco fuese limitado. El estándar es 0 que significa sin límites.

`minmessage`

Esta configuración puede ser usada para eliminar mensajes que son más cortos que un determinado número de segundos. El estándar es 0 que significa sin tiempo mínimo definido.

`format`

Permite la selección de formato de audio usado para almacenar los mensajes de buzón de voz. El formato es un string y puede ser `wav49|gsm|wav`.

- **wav49**: En este formato el tamaño de los archivos es pequeño y

la calidad buena. Es una buena opción para mandar mensajes de notificación por e-mail, pues tiene más soporte en las estaciones con Windows.

- **gsm**: Básicamente el mismo tamaño y calidad de voz, pero es menos soportado por el cliente.
- **wav**: Formato no compactado, grande en el tamaño y no es recomendado.

maxgreet

Permite al administrador limitar la compresión de un saludo de usuario grabado por el usuario. El valor es un entero definiendo el tiempo máximo en segundos. El estándar es 0 que significa sin límite.

skipms

Define el intervalo en milisegundos para usar cuando se pasa adelante o hacia atrás un mensaje está siendo tocado. El valor ingresado debe ser positivo y entero. El estándar es 3000 (3 segundos)

maxlogins

Define el número de tentativas de login antes de que Asterisk desconecte el usuario. El valor debe ser entero y positivo. El estándar es 3.

cidinternalcontexts

Define los contextos internos usados para determinar el tipo de anuncio de voz a ser tocado cuando sea leído la identificación de llamada o cuando el recurso avanzado saycid estuviese habilitado. Esto altera el mensaje de "llamada de '12345678' por "llamada de la extensión.....".

review

A veces es interesante dejar al que llama reoír sus mensajes antes de pasar a los mensajes de su casilla postal. Los valores pueden ser 'yes/no'.

operator

Permite a los usuarios ir para la telefonista mientras el mensaje está siendo grabado o una vez que el mensaje haya sido enviado, si la opción *review* estuviese configurada. La telefonista debe ser configurada en la extensión 'o' en el contexto de voicemail en el archivo `extensions.conf`.

saycid

Vuelve a leer el número del llamado antes de tocar el mensaje de entrada, después de ser tocado el mensaje de fecha y hora.

dialout

Especifica el contexto a ser usado para que una llamada de salida pueda ser usada. Recibe un string con el contexto.

callback

Especifica el contexto a ser usado en el recurso "devolver la llamada". Recibe un string especificando el contexto de salida a ser usado.

dbuser

Especifica el nombre de usuario en la base de datos MySQL a ser usado en la aplicación de buzón de voz. El valor es un string, el estándar es 'test'. Note que Asterisk debe ser compilado con soporte para MySQL.

dbpass

Especifica la contraseña en la base de datos MySQL a ser usada para la aplicación de buzón de voz

dbhost

Especifica el servidor de base de datos MySQL. El valor es un string.

dbname

Especifica el nombre de la base de datos MySQL. El valor es un string.

dboption

Especifica la opción de base de datos Postgres. El valor es un string. Note que el Asterisk tiene que ser compilado para el Postgres para que esta opción funcione.

pbxskip

Esta configuración cambia la línea del asunto *Subject:* en un mensaje de notificación. Esta configuración recibe 'yes/no'. El estándar es 'no'. Cuando esta en 'yes' la línea del asunto *Subject:* quedará: "Subject: New message M em Mailbox B". Cuando este seteado en 'no' la línea del asunto quedará: "Subject: [PBX]: New message M in mailbox B".

fromstring

Esta configuración permite al administrador sustituir la porción de la línea "From:" en el mensaje de notificación. Recibe un string.

emailsubject

Esta configuración substituye completamente la línea asunto *Subject:* . El valor pasado es un string.

emailbody

Substituye el mensaje normal de texto vista en el cuerpo del mensaje de notificación del voicemail. También soporta la substitución de variables. para tornarla más legibles. Las variables son abordadas más adelante.

exitcontext

Contexto de salida para dejar al usuario después de tener presionado * o 0 para salir del voicemail.

nextaftercmd

Si es configurado en "yes" después de borrar el mensaje de voicemail, el sistema automáticamente tocará el próximo mensaje.

12.4.2 Variables para emailsubject y emailbody.

- VM_NAME
- VM_DUR
- VM_MSGNUM
- VM_MAILBOX
- VM_CIDNUM
- VM_CIDNAME
- VM_CALLERID
- VM_DATE

12.4.3. Configuraciones para las secciones [CONTEXT]

La parte final del voicemail.conf contiene uno o más contextos.

```
[context_secti on]
extensi ón_number =>
voicemail_password, user_name, user_email_address, user_pager_email_address, us
er_opti on(s)
```

Después, la definición en la sección contexto de una entrada para cada buzón debe existir en este contexto. Los parámetros están descriptos abajo:

- **extensión_number:** es el número de la extensión que será designado para esta casilla de buzón de voz. Existen cinco parámetros que definen la configuración de buzón de voz para esta entrada:
- **voicemail_password:** Contiene la contraseña numérica para esta casilla de buzón de voz.
- **user_name:** Campo alfanumérico con el primer y ultimo nombre de usuario.
- **user_email_address:** Dirección de e-mail para notificación.
- **pager_email_address:** puede ser configurado para la dirección de e-mail de un pager de forma que cuando reciba un buzón de voz sea notificado en un celular.
- **user_option(s):** Campo a ser usado para sustituir valores definidos en la sección general. Existen 9 pares del tipo "*configuración=valor*" que pueden ser especificados. Las opciones deben ser separadas por la barra vertical (|). Las nueve configuraciones son: **attach**, **serveremail**, **tz**, **saycid**, **review**, **operator**, **callback**, **dialout**, e **exitcontext**.

12.5 Interfase Web para el Buson de voz.

Existe un script en Perl en la distribución fuente llamado `vmail.cgi`, localizado en `/usr/src/asterisk/vmail.cgi`.

Note que el `make install` no instala esta interfase a menos que usted corra **make webvmail** separadamente. Este script requiere que el Perl y `Perl-suidperl` estén instalados en el Web Server.

Este script puede requerir alguna edición antes de funcionar en su instalación. Copie `/usr/asterisk/images/*.gif` para el path `/asterisk` en el árbol html. El script precisa pasar a ejecutable:

```
chmod +x vmail.cgi
```


12.6 Sumario

En este capítulo usted aprendió los pasos necesarios para la configuración de buzón de voz. Conoció también los archivos de configuración y los principales archivos relacionados.

Exclusivo para Vicente Miguel Perez Paredes

12.7 Cuestionario

1. Los archivos emparentados con la configuración del Buzón de voz son:

- sip.conf
- iax.conf
- asterisk.conf
- voicemail.conf
- vmail.conf
- extensions.conf

2. En la aplicación VoiceMail(), los parámetros "u"y"b" son _____ y _____ respectivamente. Estos sirven para determinar que tipo de mensaje el buzón de voz tocará para el usuario.

- Ocupado, Libre
- Ocupado, No atiende
- No atiende, Ocupado
- Libre, Ocupado

3. La aplicación VoiceMailMain() es usada para que el usuario que llamó deje su mensaje. La afirmación es correcta:

- Correcta
- Incorrecta

4. Para salir del VoiceMailMain usted debe digitar:

- 0
- *
- #
- 9999

5. Escriba abajo la sintaxis del comando VoiceMail()

6. En la sección [general] del archivo voicemail.conf, el parámetro attach=yes va hacer que una notificación por e-mail sea enviada al usuario con el archivo de audio anexado. La afirmación es:

- Correcta
- Incorrecta

7 La opción "delete" hace que todas los mensajes sean borrados de la casilla postal antes de que hayan sido enviadas para lo buzón del usuario.

- Correcta
- Incorrecta

8. El formato más adecuado para los mensajes de audio es el "WAV", pues tienen más soporte de estaciones Windows.

- Correcta
- Incorrecta

9. Es posible personalizar los mensajes de buzón con emailsubject y emailbody. Que variables pueden ser usadas para indicar el CallerID en el mensaje:

10. El nombre de CGI para instalar la interfase Web del voicemail es el _____.

Exclusivo para Vicente Miguel Pérez Parrales

Billing en Asterisk

13.1 Introducción

El Asterisk como en otras plataformas de telefonía, permite la contabilidad de las llamadas telefónicas. Existen diversos programas en el mercado que importan la contabilidad de llamadas generados por las centrales telefónicas y tratan estos con fines de correlacionarlos con la cuenta telefónica, controles internos y estadísticas entre otros cosas.

13.2 Objetivos:

Al final de este capítulo el lector deberá estar apto para:

- Entender donde y en que formato son generados los informes de contabilidad de llamadas
- Generar los informes de contabilidad de llamada de Asterisk en una base de datos MySQL
- Implantar un sistema de autenticación de discado

13.3 Formato de los informes de contabilidad de llamadas del Asterisk

El Asterisk genera los informes de contabilidad de llamadas, también conocidos como CDR (Call Detail Records, o, registro detallado de llamadas), para cada llamada. Los registros son almacenados en un archivo de texto separado por comas, también conocido como CSV (comma separated value), en el directorio `/var/log/asterisk/cdr-csv`.

Los principales campos CDR son:

CDR	Descripción	Tipo	Tam.
Accountcode	Número de la cuenta usado	String	20
Src	Identificación de origen	String	80
Dst	Extensión de destino	String	80
Dcontext	Contexto de destino	String	80
Clid	Texto de identificación de llamada	String	80
Channel	Canal usado	String	80
Dstchannel	Canal de destino	String	80
Lastapp	Última aplicación	String	80
Lastdata	Últimos parámetros de la aplicación	String	80
Start	Inicio de llamada	Date/Time	
Answer	Respuesta de llamada	Date/Time	

End	Fin de la llamada	Date/Time	
Duration	Tiempo total desde el discado, en segundos	Integer	
Billsec	Tiempo total contabilizado, en segundos	Integer	
Disposition	Lo que sucedió con la llamada: ANSWERED (atendida), NO ANSWER (no atendida), BUSY (ocupada), FAILED (falló)	String	20
Amaflags	Indicador, DOCUMENTATION (documentar), BILLING (Contabilizar), OMMIT (omitir), IGNORE (ignorar).	String	20
User field	Campo definido por el usuario	String	255

Sigue un ejemplo en dos líneas del archivo csv importado para una tabla.

AccountCode	CallerID No.	Extensión	Contexto	CallerID texto	Canal Origen	Canal Destino
1234	4830258576	*72*1234*8584	admin	"Alexandre Keller" <4830258576>	SIP/8576-5f30	SIP/8584-9153
1234	4830258576	*72*1234*8584	admin	"Alexandre Keller" <4830258576>	SIP/8576-96f5	SIP/8584-3312
1234	4830258576	*72*1234*8584	admin	"Alexandre Keller" <4830258576>	SIP/8576-74ac	SIP/8584-297b
1234	4830258576	2012348584	admin	"Alexandre Keller" <4830258576>	SIP/8576-2c5d	SIP/8584-9870
1234	4830258584	2012348576	default	"Luiz Eduardo Dagios" <4830258584>	SIP/8584-03fd	SIP/8576-645c

Aplicación	Parámetros	Inicio	Atendido	Fin	Dur.	Bil.	Disposición	Amaflags
Dial	SIP/8584 30 tT	27/3/2006 16:05	27/3/2006 16:05	27/3/2006 16:05	5	3	ANSWERED	DOCUMENTATION
Dial	SIP/8584 30 tT	27/3/2006 16:16	27/3/2006 16:16	27/3/2006 16:16	6	4	ANSWERED	BILLING
Dial	SIP/8584 30 tT	27/3/2006 16:22	27/3/2006 16:22	27/3/2006 16:22	9	5	ANSWERED	BILLING
Dial	SIP/8584 30 tT	27/3/2006 16:37	27/3/2006 16:37	27/3/2006 16:37	5	2	ANSWERED	BILLING
Dial	SIP/8576 30 tT	27/3/2006 16:37	27/3/2006 16:37	27/3/2006 16:37	9	5	ANSWERED	BILLING

13.4 Codigos de cuenta y contabilización automática.

Usted puede especificar códigos de cuenta y los indicadores de CDR (AMA – Automated Message Accounting) en cada uno de los canales (sip.conf, iax.conf y zapata.conf).

Para cada canal usted puede especificar los indicadores de CDR (amaflags)

- Default: Configura el estándar del sistema
- Omit: No registrar las llamadas
- Billing: Marca el registro para Contabilizar
- Documentation: Marca el registro para documentación

Así como usted puede configurar los amaflags, usted puede configurar para cada canal un código de contabilización. El código de la cuenta puede ser un string de hasta 20 caracteres. Con él usted puede contabilizar por departamento por ejemplo: ventas, producción, asistencia técnica.

Ejemplo: (sip.conf)

```
[8576]
amafl ags=default
accountcode=Soporte
type=fri end
username=8576
```

13.5 Cambiando el formato del CDR

Usted puede cambiar el formato de lo CDR, editando el archivo `cdr_custom.conf`.

```
;
; Mappings for custom config file
;
[mappings]
Master.csv =>
"${CDR(cli id)}", "${CDR(src)}", "${CDR(dst)}", "${CDR(dcontext)}", "${CDR(channe
l)}", "${CDR(dstchannel)}", "${CDR(lastapp)}", "${CDR(lastdata)}", "${CDR(start
)}", "${CDR(answer)}", "${CDR(end)}", "${CDR(durati on)}", "${CDR(bi ll sec)}", "${
CDR(di sposi ti on)}", "${CDR(amafl ags)}", "${CDR(accountcode)}", "${CDR(uni quei d
)}", "${CDR(userfi el d)}"
```

13.6 Métodos de almacenamiento del CDR

La contabilidad de llamadas puede ser almacenada de diversas formas. Las formas más importantes en nuestra opinión son los CSV, pues puede ser importada para planillas y para software de contabilidad de llamadas de forma fácil. El almacenamiento en MySQL también es muy común y permite el uso de interfaces gráficas de visualización de contabilidad. La tercera forma de almacenamiento de gran interés es el odbc, pues permite almacenar en diversos bancos de datos corporativos principalmente en Oracle.

13.6.1 Drivers de almacenamiento disponibles

`cdr_csv` – Archivos de texto separados con comas
`cdr_SQLite` – Banco de datos SQLite
`cdr_pgsq` – Banco de datos Postgres
`cdr_odbc` – Banco de datos soportado por unixODBC
`cdr_mysql` – Banco de datos MySQL
`cdr_FreeTDS` – Banco de datos Sybase e MSSQL
`cdr_yada` – Banco de datos yada
`cdr_manager` – Contabilidad para el Asterisk Manager
`cdr_radius` – Radius Server.

El Asterisk contabiliza para todos los módulos que estuviesen activos en el archivo `/etc/asterisk/modules.conf`. Si el archivo estuviese con la opción `autoload=yes`, este carga todos los módulos del directorio especificado.

13.6.2 Almacenando en CSV

Como ya dijimos el Asterisk coloca los datos en un archivo CSV por default. Esto es hecho por el modulo `cdr_csv.so`. Si usted no estuviese viendo los datos en el archivo verifique si el Asterisk está cargado usando el comando “`module show`” y si no lo estuviese, verifique la carga en `/etc/asterisk/modules.conf`.

13.6.3 Almacenando en una base de datos MySQL.

Asterisk también puede almacenar registros de CDR en un banco de datos MySQL. Debido al licenciamiento de las bibliotecas de MySQL la aplicación de billing en MySQL no es más parte integrante de Asterisk. Esto viene con el archivo `addons` que usted tendrá que descomprimir y compilar si desease usar esta aplicación.

Siga las siguientes instrucciones para compilar el soporte de MySQL para Asterisk

Instale los paquetes MySQL y MySQL-devel

```
#apt-get install mysql-server-4.1
#apt-get install libmysqlclient12-dev
#cd /usr/src
#wget http://ftp.digium.com/pub/asterisk/releases/asterisk-addons-1.4.1.tar.gz
#tar -xvzf asterisk-addons-1.4.1.tar.gz
#cd asterisk-addons-1.4.1
#make clean
#make
#make install
```

Un ejemplo de la configuración de archivos `cdr_mysql.conf` puede ser visto abajo:

```
[global]
hostname=localhost
dbname=asteriskdb
password=asterisk
user=asterisk
port=3306
; sock=/var/run/mysqld/mysqld.sock
; userfile=d=1
```

Edite el archivo `modules.conf` para incluir la carga del módulo `cdr_addon_mysql.so`

Creando el banco de datos para el `cdr_addon_mysql`

```
mysql -p
```

```
CREATE DATABASE asteriskdb;
```

```

GRANT INSERT ON asterisk.* TO asterisk@localhost IDENTIFIED BY 'asterisk';
USE asteriskdb;

CREATE TABLE `cdr` (
  `calldate` datetime NOT NULL default '0000-00-00 00:00:00',
  `clid` varchar(80) NOT NULL default '',
  `src` varchar(80) NOT NULL default '',
  `dst` varchar(80) NOT NULL default '',
  `dcontext` varchar(80) NOT NULL default '',
  `channel` varchar(80) NOT NULL default '',
  `dstchannel` varchar(80) NOT NULL default '',
  `lastapp` varchar(80) NOT NULL default '',
  `lastdata` varchar(80) NOT NULL default '',
  `duration` int(11) NOT NULL default '0',
  `billsec` int(11) NOT NULL default '0',
  `disposition` varchar(45) NOT NULL default '',
  `amaflags` int(11) NOT NULL default '0',
  `accountcode` varchar(20) NOT NULL default '',
  `userfield` varchar(255) NOT NULL default ''
);

ALTER TABLE `cdr` ADD INDEX ( `calldate` );
ALTER TABLE `cdr` ADD INDEX ( `dst` );
ALTER TABLE `cdr` ADD INDEX ( `accountcode` );

```

Un consejo de uso es copiar y pegar estos comandos SQL en un archivo de texto como cdr.sql , entonces ejecutar el siguiente comando.

```
mysql -p -u asterisk asteriskdb <cdr.sql
```

13.7 Aplicaciones y funciones

Diversas aplicaciones están relacionadas a billing (Contabilidad de llamadas).

13.7.1 CDR(accountcode)

Configura un código de cuenta antes de llamarse a una aplicación como el Dial().

Formato:

```
Set(CDR(accountcode)=accountcode)
```

El código de la cuenta puede ser verificado a través de la variable `${CDR(accountcode)}`

13.7.2 SetAMAFIags

```
Set(CDR(amaflags)=flags)
```


Configura el indicador de billing, las opciones son: default, omit, documentation, billing.

13.7.3 NoCDR()

Hace que ningún CDR sea grabado para una cierta llamada.

13.7.4 ResetCDR()

Apaga el CDR, opcionalmente almacenando el CDR actual antes de cerrarlo si es que la opción w fuese especificada.

13.7.5 Set(CDR(userfield)=Value)

Este comando configura el campo definido por el usuario de CDR. Si usted estuviese usando el cdr_addon_mysql certifíquese de tener userfield=1 en el cdr_mysql.conf. Si estuviera usando cdr_csv.c, edite el archivo fuente recompile para habilitar el campo de usuario. El comando no tiene efecto si no estuviese habilitado.

13.7.6 AppendCDRUserField(Value)

Adiciona datos al campo definido por el usuario (userfield) en el CDR.

13.8 Usando autenticación de los usuarios

Esta aplicación permite autenticar un usuario antes de hacer una comunicación. La autenticación puede ocurrir con una contraseña colocada directamente en el comando, con un archivo de contraseñas cuando es precedido por "/" o usando el AstDB.

Formato:

```
Authenticate(password[|options])
Authenticate(/passwdfile[|options])
Authenticate(</db-keyfamily|d>options)
```

Opciones:

- a - Configura el código de la cuenta en el CDR con la contraseña que fuese ingresada. Esto es útil si fuese deseado Contabilizar estas comunicaciones.
- d – Interpreta el camino como una familia de llaves en el banco de datos
- r – Remueve la llave de la base de datos después de una tentativa de autenticación con éxito (funciona apenas con la opción d).
- j – Empuja para la prioridad n+101 en el caso de autenticación inválida.

Ejemplo: (Autenticación para llamadas internacionales)

```
exten=_000. , 1, Authenticate(/contraseña|daj)
exten=_000. , 2, Dial (Zap/g1/${EXTEN: 1}, 20, tT)
exten=_000. , 3, Hangup()
exten=_000. , 102, Playback(no_authorized)
exten=_000. , 103, Hangup()
```

Para catastrar las contraseñas use la línea de comando en la consola

```
CLI> database put contraseña 123456 1
```

13.9 Usando las contraseñas del Buzón de voz

Hace lo mismo que el Authenticate, pero usa la base de datos del buzón de voz para la autenticación de los usuarios.

```
VMAuthenticate([mailbox] [@context] [|options])
```

Si una casilla postal fuera especificada, apenas la contraseña de la casilla postal será considerada válida. Si la casilla postal no fuese especificada, la variable de canal AUTH_MAILBOX será configurada con la casilla postal autenticada. Si la opción contiene 's' ningún prompt será ejecutado.

Ejemplo: (Autenticación para llamadas internacionales)

```
exten=_000. , 1, VMAuthenticate(${CALLERID}@ramales|ajs)
exten=_000. , 2, Dial (Zap/g1/${EXTEN: 1}, 20, tT)
exten=_000. , 3, Hangup()
exten=_000. , 102, Playback(no_authorized)
exten=_000. , 103, Hangup()
```

13.10 Sumario

En este capítulo aprendimos como atribuir códigos de contabilización a los canales (ramales y enlaces) de Asterisk. Aprendimos como configurar el billing de llamadas para ser almacenada en archivos de texto y también en bases de datos. Por fin entendemos los mecanismos de autenticación que permiten un control mayor de la contabilización en Asterisk.

13.11 Cuestionario

1. Por default Asterisk contabiliza las llamadas en /var/log/asterisk/cdr-csv.

- Falso
- Verdadero

2. Asterisk solo permite contabilizar en las bases de datos:

- MySQL
- Oracle nativo
- MSSQL
- Archivos texto CSV
- Bases de datos por unix_ODBC

3. Asterisk contabiliza apenas en un tipo de almacenamiento

- Falso
- Verdadero

4. Los indicadores disponibles para billing (amaflags) son:

- Default
- Omit
- Tax
- Rate
- Billing
- Documentation

5. Si usted desea asociar un departamento o un código de contabilización al CDR, usted debe usar el comando _____. El código de la cuenta puede ser verificado por la variable_____.

6. La diferencia entre las aplicaciones NoCDR() y ResetCDR() es que el NoCDR() no genera ningún billing, en tanto que el ResetCDR() cierra el tiempo de la llamada en el registro.

- Falso
- Verdadero

7. Para usar un campo definido por el usuario (UserField) es necesario recompilar el Asterisk.

- Falso
- Verdadero

8. Los tres métodos de autenticación disponibles para la aplicación Authenticate() son:

- Contraseña
- Archivos de contraseñas
- Banco berkley DB (dbput e dbget)
- VoiceMail

9. Las contraseñas de buzón son especificadas en otra parte del archivo voicemail.conf y no son las mismas del buzón de voz.

- Falso
- Verdadero

10. La opción del comando Authenticate ___ hace que la contraseña sea colocada en el código de contabilización del cdr.

- a
- j
- d
- r

Exclusivo para Vicente Miguel Pérez Paredes

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

Integración de Asterisk usando AGI y AMI

14.1 Introducción

En muchas situaciones será necesario extender la funcionalidad de Asterisk usando aplicaciones externas. En centrales telefónicas convencionales, esto era normalmente hecho a través de una interfase de integración de telefonía-computador conocida como CTI. El Asterisk como ya es construido sobre un computador, tiene diversas formas de integración y no esta limitado apenas a una interfase de CTI. El punto negativo es que una interfase CSTA estandarizada todavía no está disponible. En este capítulo además de la interfase de CTI de Asterisk conocida como AMI (Asterisk Manager Internase), veremos el uso de AGIs, del comando "asterisk -rx" y del aplicativo System(). A partir de estos recursos es posible construir aplicaciones de URA, discadores, "popup" de pantalla, monitorización e informes. Consideraciones sobre escalabilidad y seguridad también serán abordadas.

14.2 Objetivos de este capítulo

Al final de este capítulo el lector deberá estar apto para:

- Describir las opciones de acceso al Asterisk por programas externos
- Usar el comando asterisk -rx para ejecutar un comando de consola
- Usar la aplicación System() para llamar a programas externos en el plan de discado
- Explicar que es y como funciona el AMI
- Configurar los archivos manager.conf y habilitar la interfase de gerenciamiento.
- Ejecutar un comando en el Asterisk a través de AMI usando PHP.
- Instalar y configurar el Asterisk Manager Proxy
- Describir los diversos tipos de AGI (DeadAGI, AGI, EAGI, FastAGI)
- Ejecutar un AGI simple.

14.3 Principales formas de extender el Asterisk

Asterisk posee diversas formas de interfasear con programas externos y en muchos casos este hace uso de estas interfaces en aplicativos como el buzón de voz por ejemplo donde ejecuta un comando externo de tipo sendmail. Vamos abordar las siguientes formas de extender el Asterisk:

- Línea de comando de consola
- Aplicación System()

- Asterisk Manager Internase - AMI
- Asterisk Gateway Internase - AGI

14.4 Extendiendo Asterisk a través de la consola.

Una aplicación puede fácilmente comandar el Asterisk a través de la consola usando el comando:

```
asterisk -rx <comando>
```

Por ejemplo:

```
#asterisk -rx "stop now"
```

O lo mismo con un comando que tenga un output como:

```
asterisk: ~# asterisk -rx "sip show peers"
Name/username      Host                Dyn Nat ACL Port      Status
4000/4000          10.1.1.6           D          5060     Unmonitored
1 sip peers [1 online, 0 offline]
```

14.5 Extendiendo Asterisk usando el aplicativo System

La aplicación System() permite la llamada de una aplicación externa a partir del plan de discado.

```
asterisk*CLI> core show application system
asterisk*CLI>
  -= Info about application 'System' -=

[Synopsis]
Execute a system command

[Description]
System(command): Executes a command by using system(). If the command
fails, the console should report a fallthrough.
Result of execution is returned in the SYSTEMSTATUS channel variable:
  FAILURE      Could not execute the specified command
  SUCCESS      Specified command successfully executed
```

14.5.1 Ejemplo de uso de la aplicación System()

Ejemplo 1: Pop-up del CallerID de la llamada vía WindowsPopup.

```
exten => 200, 1, NoOp(${CALLERID(all)} ${DATETIME})
exten => 200, 2, System(/bin/echo -e "' Incoming Call From: ${CALLERID(all)}
\r Received: ${DATETIME}' " | /usr/bin/smbclient -M target_netbiosname)
exten => 200, 3, Dial, sip/sipuser|30|t
exten => 200, 4, Congestion
```

Ejemplo 2: Envío de la imagen recibida para un e-mail a través del fax2email

```
exten => fax, 1, Set(FAXFILE=/var/spool/asterisk/fax/${CALLERID(num)}.tif)
exten => fax, n, Set(FAXFILENOEXT=/var/spool/asterisk/fax/${CALLERID(num)})
exten => fax, n, rxfax(${FAXFILE})
exten => fax, n, System(' /usr/bin/fax2mail ${CALLERID(num)}
"${CALLERID(name)}" FaxNum RecipName email@address.com ${FAXFILENOEXT} p')
```

14.6 Lo que representa AMI

El AMI (Asterisk Manager Interfase) permite a un programa cliente conectarse a una instancia de Asterisk y emitir comandos o leer eventos del PBX sobre un flujo TCP/IP. Integradores van encontrar esos recursos particularmente útiles cuando intentasen rastrear el estado de los teléfonos dentro del Asterisk. Un protocolo de línea simple basado en el concepto de llave: el valor es utilizado sobre un socket tcp. Las líneas son terminadas usando CRLF. El Asterisk en si no está preparado para recibir muchas conexiones sobre la interfase del Manager, por esto si usted posee dos o más clientes de AMI considere usar un proxy de AMI como AstManProxy.

14.6.1 Que lenguaje utilizar ?

Hoy es muy difícil seleccionar un lenguaje de programación, pues estos son muchos. Existen ejemplos de programación de AMI en Java, Perl, Python, PHP, C, C# entre muchas otras. Es posible programar el AMI en cualquier lenguaje que soporte una interfase sockets o que simule Telnet. Para este documento, escogemos PHP por la popularidad y por la facilidad de integración con un Web-browser.

14.6.2 Comportamiento del protocolo

- Antes de emitir cualquier comando para Asterisk, usted debe establecer una sesión de gerenciamiento.
- La primer línea de un paquete tendrá una llave "Action" cuando es enviada a un cliente.
- La primer línea de un paquete tendrá una llave "Response" o "Event" cuando es enviada a partir de Asterisk.

- Paquetes que pueden ser transmitidos en cualquier dirección después de la autenticación.

14.6.3 Tipos de paquetes

El tipo de paquete es determinado por la existencia de las siguientes llaves:

- Action: Un paquete enviado por un cliente conectado al Asterisk solicitando que una acción sea hecha. Existe un conjunto finito de acciones disponibles para el cliente, determinada por los módulos actualmente cargados en el Asterisk. Solo una acción será procesada por vez. El paquete contiene el nombre de la operación y sus parámetros.
- Response: Es la respuesta enviada por el Asterisk a la última acción solicitada por el cliente.
- Event: Datos pertenecientes a un evento generado de dentro del núcleo de Asterisk o por un módulo.

Al enviar paquetes de tipo Action para Asterisk los clientes incluyen un parámetro ActionID. Como no existe ninguna garantía en la orden de las respuestas, el ActionID es usado para garantizar que una respuesta corresponde a una determinada acción.

Los paquetes de eventos son usados en dos contextos diferentes. De un lado ellos informan al cliente sobre cambios en el Asterisk (como nuevos canales siendo creados y desconectados o agentes entrando y saliendo de una fila). La segunda forma de uso de los eventos es transportar las respuestas a pedidos de los clientes.

14.7 Configurando usuarios y permisos

Para acceder al AMI es necesario establecer una conexión TCP/IP escuchando una puerta TCP (normalmente 5038). Para esto es necesario configurar el archivo `/etc/asterisk/manager.conf` y crear una cuenta de usuario. Además de la cuenta es necesario configurar un conjunto de permisos.

Existe un conjunto finito de permisos: "read" para lectura, "write" para escritura o ambos. Estos permisos son definidos en el archivo `/etc/asterisk/manager.conf`.

```
[general]
enabled=yes
port=5038
bindaddr=127.0.0.1

[admin]
secret=contraseña
```

```
read=system,call,log,verbose,command,agent,user
write=system,call,log,verbose,command,agent,user
deny=0.0.0.0/0.0.0.0
permit=127.0.0.1/255.255.255.255
```

14.7.1 Logueandose en AMI

Para hacer el login y autenticarse en AMI, usted precisa enviar una acción de tipo "Login" con su nombre de usuario y contraseña. Algo como lo que esta abajo:

```
Action: Login
Username: admin
Secret: contraseña
```

14.7.2 Logueandose en AMI via PHP.

```
<?php
$socket = fsockopen("127.0.0.1", "5038", $errno, $errstr, $timeout);
fputs($socket, "Action: Login\r\n");
fputs($socket, "Username: admin\r\n");
fputs($socket, "Secret: contraseña\r\n\r\n");
?>
```

Si usted no quisiera recibir los eventos, puede usar:

```
<?php
$socket = fsockopen("127.0.0.1", "5038", $errno, $errstr, $timeout);
fputs($socket, "Action: Login\r\n");
fputs($socket, "Username: admin\r\n");
fputs($socket, "Secret: contraseña\r\n\r\n");
fputs($socket, "Events: off\r\n\r\n");
?>
```

14.7.3 Paquetes de tipo Action

Cuando usted envía al Asterisk una acción, algunas llaves extras pueden ser proporcionadas como parámetros. Por ejemplo, un número a ser llamado. Esto es hecho en la forma de pasar los pares llave:valor después el "Action". Es posible también pasar variables para el plan de discado.

```
Action: <action type><CRLF>
<Key 1>: <Value 1><CRLF>
<Key 2>: <Value 2><CRLF>

Variable: <Variable 1>=<Value 1><CRLF>
Variable: <Variable 2>=<Value 2><CRLF>
...
<CRLF>
```

14.7.4 Comandos de tipo Action

Usted puede usar el comando **manager show commands** para descubrir las acciones disponibles en AMI. En la versión 1.2.7 los comandos eran los siguientes.

Action	Privilege	Sinopse
AbsoluteTimeout	call,all	Set Absolute Timeout
AgentCallbackLogin	agent,all	Sets an agent as logged in by callback
AgentLogoff	agent,all	Sets an agent as no longer logged in
Agents	agent,all	Lists agents and their status
ChangeMonitor	call,all	Change monitoring filename of a channel
Command	command,all	Execute Asterisk CLI Command
DBGet	system,all	Get DB Entry
DBPut	system,all	Put DB Entry
EventsControl	<none>	Event Flow
ExtensiónState	call,all	Check Extensión Status
Getvar	call,all	Gets a Channel Variable
Hangup	call,all	Hangup Channel
IAXnetstats	<none>	Show IAX Netstats
IAXpeers	<none>	List IAX Peers
ListCommands	<none>	List available manager commands
Logoff	<none>	Logoff Manager
MailboxCount	call,all	Check Mailbox Message Count
MailboxStatus	call,all	Check Mailbox
Monitor	call,all	Monitor a channel
Originate	call,all	Originate Call
ParkedCalls	<none>	List parked calls
Ping	<none>	Keepalive command
QueueAdd	agent,all	Add interface to queue.
QueuePause	agent,all	Makes a queue member temporarily unavailable
QueueRemove	agent,all	Remove interface from queue.
Queues	<none>	Queues
QueueStatus	<none>	Queue Status
Redirect	call,all	Redirect (transfer) a call
SetCDRUserField	call,all	Set the CDR UserField
Setvar	call,all	Set Channel Variable
SIPpeers	System,all	List SIP peers (text format)
SIPshowpeer	System,all	Show SIP peer (text format)
Status	call,all	Lists channel status
StopMonitor	call,all	Stop monitoring a channel

Si usted precisa de un comando específico:

Manager show command originate:

```

asterisk*CLI>manager show command originate
Action: Originate
Synopsis: Originate Call
Privilege: call,all
Description: Generates an outgoing call to a Extensión/Context/Priority or
Application/Data
Variables: (Names marked with * are required)
  *Channel: Channel name to call
  Exten: Extensión to use (requires 'Context' and 'Priority')
  Context: Context to use (requires 'Exten' and 'Priority')
  Priority: Priority to use (requires 'Exten' and 'Context')
  Application: Application to use
  Data: Data to use (requires 'Application')
  Timeout: How long to wait for call to be answered (in ms)
  CallerID: Caller ID to be set on the outgoing channel
  Variable: Channel variable to set, multiple Variable: headers are
allowed
  Account: Account code
  Async: Set to 'true' for fast origination

```

14.7.5 Paquetes de tipo Event

Evento de tipo Link:

Descripción:

Disparado cuando dos canales de voz son conectados y el cambio de datos de voz se inicia.

Notas:

Varios eventos de link pueden ser vistos para una única llamada. Esto puede ocurrir cuando el Asterisk falla en configurar un puente nativo para una llamada. En estos casos Asterisk esta entre los dos teléfonos y configura normalmente una traducción de Codecs.

Ejemplo:

```

Event: Link
Channel 1: SIP/101-3f3f
Channel 2: Zap/2-1
Uniqued1: 1094154427.10
Uniqued2: 1094154427.11

```

Evento de tipo unlink

Descripción:

Disparado cuando un link entre dos canales de voz es descolgado, un poco antes de completar la llamada.

Ejemplo:

```
Event: Unlink
Channel 1: SIP/101-3f3f
Channel 2: Zap/2-1
Uniqueid1: 1094154427.10
Uniqueid2: 1094154427.11
```

14.7.8 Eventos disponibles en el Asterisk

AbstractAgentEvent	HoldEvent	PeerStatusEvent
AbstractParkedCallEvent	JoinEvent	QueueEntryEvent
AbstractQueueMemberEvent	LeaveEvent	QueueEvent
AgentCallbackLoginEvent	LinkageEvent	QueueMemberAddedEvent
AgentCallbackLogoutEvent	LinkEvent	QueueMemberEvent
AgentCalledEvent	LogChannelEvent	QueueMemberPausedEvent
AgentCompleteEvent	ManagerEvent	QueueMemberRemovedEvent
AgentConnectEvent	MeetMeEvent	QueueMemberStatusEvent
AgentDumpEvent	MeetMeJoinEvent	QueueParamsEvent
AgentLoginEvent	MeetMeLeaveEvent	QueueStatusCompleteEvent
AgentLogoutEvent	MeetMeStopTalkingEvent	RegistryEvent
AgentsCompleteEvent	MeetMeTalkingEvent	ReloadEvent
AgentsEvent	MessageWaitingEvent	RenameEvent
AlarmClearEvent	NewCallerIdEvent	ResponseEvent
AlarmEvent	NewChannelEvent	ShutdownEvent
CdrEvent	NewExtenEvent	StatusCompleteEvent
ChannelEvent	NewStateEvent	StatusEvent
ConnectEvent	OriginateEvent	UnholdEvent
DBGetResponseEvent	OriginateFailureEvent	UnlinkEvent
DialEvent	OriginateSuccessEvent	UnparkedCallEvent
DisconnectEvent	ParkedCallEvent	UserEvent
DNDStateEvent	ParkedCallGiveUpEvent	ZapShowChannelsCompleteEvent
ExtensiónStatusEvent	ParkedCallsCompleteEvent	ZapShowChannelsEvent
FaxReceivedEvent	ParkedCallTimeoutEvent	
HangupEvent	PeerEntryEvent	
HoldedCallEvent	PeerlistCompleteEvent	

14.8 Asterisk Manager Proxy

Asterisk no fue hecho para gerenciar un gran número de conexiones a la interfase del Manager. Por ejemplo, si cada supervisor o agente de un call-center usa el AMI para login y logout de agentes, esto puede tornar al Asterisk inestable. Fue desarrollado un Proxy llamado astmanproxy para resolver este problema. Las principales ventajas de Astmanproxy son:

- Una conexión única y persistente en el Asterisk
- Una interfase TCP más segura (no root)
- Habilidad de filtrar entrada / salida
- Menor número de conexiones y carga para el Asterisk
- Múltiples medios de acceso (estándar, http, xml, csv)
- Soporte para SSL
- Conexión a múltiples servidores Asterisk

- Formatos de I/O seleccionables en una base por cliente

Usted puede usar el Astmanproxy como base para que una aplicación basada en Web o XML permita acceso por aplicaciones Windows usando HTTP POST o HTTP GET y recibiendo la salida en HTML. Usted puede usar el Astmanproxy como un alimentador XML para un programa .NET que mantiene el registro del estado de Asterisk.

14.8.1 Instalando y configurando el astmanproxy

Para colocar el Astmanproxy para correrlo usted va a precisar:

Paso 1: Bajar el Astmanproxy

Use el sistema subversión para bajar el astmanproxy.

svn checkout <http://svncommunity.digium.com/svn/astmanproxy/trunk>

Paso 2: Compilar e Instalar

```
make
make install
```

Paso 3 – Editar el archivo de configuración

```
vi /etc/asterisk/astmanproxy.conf
```

```
; astmanproxy.conf
; Asterisk Manager Proxy Configuration Sample
; (C) 2005-2006 David C. Troy - dave@popvox.com

; List of asterisk host(s) you want to proxy
; host = ip_addr, port, user, secret, events, use_ssl
host = localhost, 5038, admin, contraseña, on, off

; Server reconnect interval (in seconds); how often to retry
; Connecting to an asterisk server whose connection was lost
retryinterval = 2

; Number of times to retry connecting to a given server
; use 0 for infinitely, or some finite number
maxretries = 10
```

Paso 4 – Opcionalmente editar otros archivos de configuración

```
vi /etc/asterisk/astmanproxy.users
vi /etc/asterisk/ssl.conf
```

Paso 5 – Inicie el programa

```
astmanproxy
```

Paso 6 – Para ver la salida del output, inicie el astmanproxy en el modo debug

```
astmanproxy -ddddd
```

Paso 7 – Para iniciar el astmanproxy en tiempo de inicialización:

Coloque el siguiente archivo en /etc/rc.d/rc.local

```
/usr/local/sbin/astmanproxy
```

Mas informaciones sobre el Astmanproxy pueden ser encontradas en el archivo README de la distribución de astmanproxy.

14.9 Asterisk Gateway Interface

AGI es una interfase de gateway para el Asterisk similar al CGI usado por los servidores Web. Este permite el uso de diferentes lenguajes de programación como el Perl, PHP, C a su elección. AGI es llamado a partir del plan de discado definido en extensions.conf. La aplicación principal para el AGI es URA con acceso a banco de datos.

Existen tres tipos de AGIs.

- AGI Normal: llamado directamente en el servidor Asterisk.
- FastAGI: que llama al AGI en otro servidor a partir de una interfase sockets.
- EAGI: da a la aplicación la posibilidad de acceder y controlar el canal de sonido mas allá de la interacción con el plan de discado.
- DEADAGI: que da acceso a un mismo canal después de hangup().

Formato de la aplicación:

```
asterisk*CLI>core show application agi
asterisk*CLI>
  -= Info about application 'AGI' -=

[Synopsis]
Executes an AGI compliant application

[Description]
[E|Dead]AGI (command|args): Executes an Asterisk Gateway Interface
compliant
program on a channel. AGI allows Asterisk to launch external programs
written in any language to control a telephony channel, play audio,
read DTMF digits, etc. by communicating with the AGI protocol on stdin
and stdout.
Returns -1 on hangup (except for DeadAGI) or if application requested
hangup, or 0 on non-hangup exit.
```

Using 'EAGI' provides enhanced AGI, with incoming audio available out of band on file descriptor 3

Use the CLI command 'agi show' to list available agi commands

Usted puede verificar los comandos agi existentes usando:

```

asterisk*CLI> agi show
  answer      Answer channel
  channel status Returns status of the connected channel
  database del Removes database key/value
  database del tree Removes database keytree/value
  database get Gets database value
  database put Adds/updates database value
  exec        Executes a given Application
  get data    Prompts for DTMF on a channel
  get full variable Evaluates a channel expression
  get option  Stream file, prompt for DTMF, with timeout
  get variable Gets a channel variable
  hangup     Hangup the current channel
  noop       Does nothing
  receive char Receives one character from channels supporting it
  receive text Receives text from channels supporting it
  record file Records to a given file
  say alpha   Says a given character string
  say digits  Says a given digit string
  say number  Says a given number
  say phonetic Says a given character string with phonetics
  say date    Says a given date
  say time    Says a given time
  say datetime Says a given time as specified by the format given
  send image  Sends images to channels supporting it
  send text   Sends text to channels supporting it
  set autohangup Autohangup channel in some time
  set callerid Sets callerid for the current channel
  set context Sets channel context
  set extensión Changes channel extensión
  set music   Enable/Disable Music on hold generator
  set priority Set channel dialplan priority
  set variable Sets a channel variable
  stream file Sends audio file on channel
  control stream file Sends audio file on channel and allows the listener
  to control the stream
  tdd mode    Toggles TDD mode (for the deaf)
  verbose     Logs a message to the asterisk verbose log
  wait for digit Waits for a digit to be pressed

```

Para propósitos de debugging usted puede usar "agi debug".

14.9.1 Usando el AGI

En el ejemplo de abajo vamos a usar el PHP-CLI, en otras palabras el php por línea de comando. Certifíquese de que el php-cli este instalado apropiadamente.

Siga los siguientes pasos para usar scripts AGI en php

Paso 1: Los scripts de agi son colocados en el directorio /var/lib/asterisk/agi-bin.

Paso 2: Coloque permisos de ejecución en sus archivos php.

```
chmod 755 *.php
```

Paso 3: Usando una interfase con el shell (específico para el php)

Las primeras líneas del script tienen que ser:

```
#!/usr/bin/php -q
<?php
```

Paso 4: Abriendo los canales de entrada y salida:

```
$stdin = fopen('php://stdin', 'r');
$stdout = fopen('php://stdout', 'w');
$stderr = fopen('php://stderr', 'w');
```

Paso 5: Gerenciando la entrada que llega del ambiente de Asterisk

El Asterisk siempre envía un conjunto de informaciones cada vez que el AGI es llamado.

```
agi_request: testphp
agi_channel: Zap/1-1
agi_language: en
agi_type: Zap
agi_callerid:
agi_dnid:
agi_context: default
agi_extensión: 4000
agi_priority: 1
```

Salve estas informaciones en su script

```
while (!feof($stdin)) {
    $temp = fgets($stdin);
    $temp = str_replace("\n", "", $temp);
    $s = explode(":", $temp);
    $agivar[$s[0]] = trim($s[1]);
    if (($temp == "") || ($temp == "\n")) {
        break;
    }
}
```

Esto va a dejar un array llamado \$agivar. Las opciones disponibles son:

- agi_request – El nombre del archivos AGI

- agi_channel – El canal que originó el AGI
- agi_language – El lenguaje configurado en el momento
- agi_type – El tipo de canal que originó el agi (Ex. Sip, Zap)
- agi_uniqueid – Identificación única de llamada
- agi_callerid – Identificador de llamada (Ex. Flavio <8590>)
- agi_context – Contexto de origen
- agi_extensión – Extensión llamada
- agi_priority – Prioridad
- agi_accountcode – Código de la cuenta del canal de origen (billing)

Para llamar una variable use \$agivar[agi_extensión].

Paso 6 – Usando el canal AGI

En este punto usted puede comenzar a hablar con el Asterisk. Use el comando fputs para enviar comandos agi. Usted puede usar el comando "echo" también.

```
fputs($stdout, "SAY NUMBER 4000 '79#' \n");
fflush($stdout);
```

Notas con relación al uso de comillas

- Las opciones del comando en el AGI no son opcionales, ellas deben aparecer en el string de comando.
- Algunas opciones deben ser pasadas en comillas <escape digits>.
- Algunas opciones no deben ser pasadas en comillas <digit string>
- Algunas opciones pueden ir de los dos modos
- Usted puede usar comillas simples cuando sea necesario

Paso 7 – Pasando variables

VARIABLES de canal pueden ser definidas dentro del AGI, pero las variables de canal no son pasadas al AGI. El ejemplo de abajo no funciona.

```
SET VARIABLE MY_DIALCOMMAND "SIP/${EXTEN}"
```

Este de abajo si funciona:

```
SET VARIABLE MY_DIALCOMMAND "SIP/4000"
```

Paso 8: Respuestas del Asterisk.

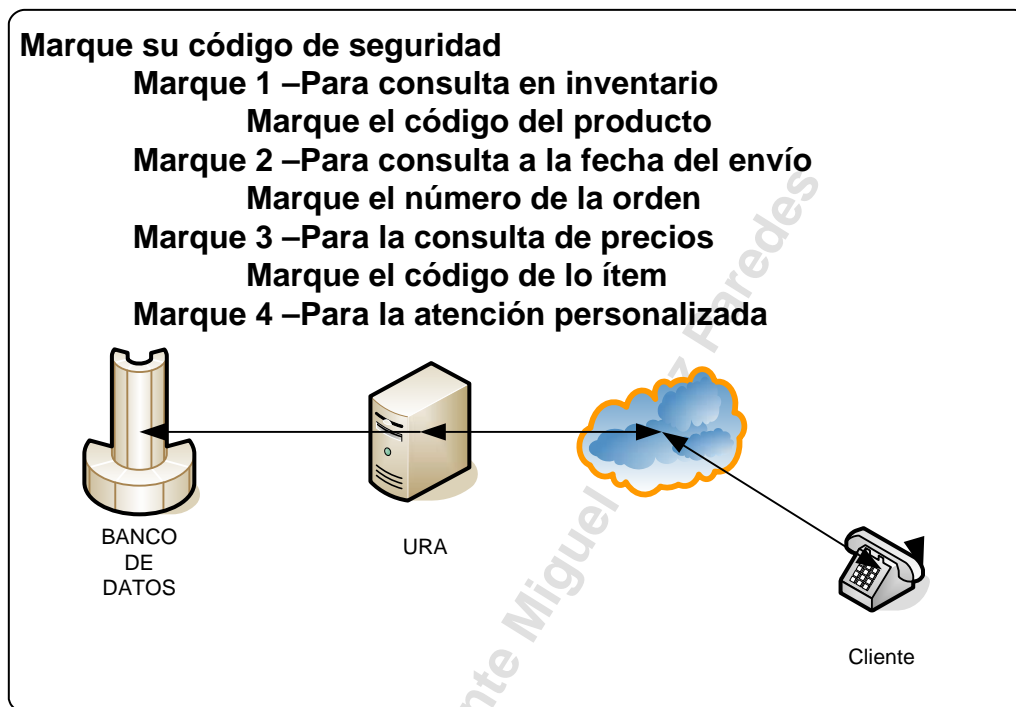
En seguida es necesario verificar las respuestas de Asterisk.

```
$msg = fgets($stdin, 1024);
fputs($stdout, $msg . "\n");
```

Paso 9: Mate los procesos que estén trabados;

Si su script falla por alguna razón, este va a permanecer trabado. Use el comando killproc para limpiarlo antes de testear de nuevo.

14.9.2 Ejemplo de AGI



El Agi de abajo captura el valor de una variable del plan de discado y utiliza este valor para realizar una consulta en un banco de datos MySQL. La consulta retorna una

```
#!/usr/bin/php -q
<?php
ob_implicit_flush(true);
set_time_limit(6);
error_reporting(0);
$in = fopen("php://stdin", "r");
$log = fopen("/var/log/asterisk/agi.log", "w");

// Habilita modo debugging (mas verbose)
$debug = true;

// Do function definitions before we start the main loop
function read() {
    global $in, $debug, $log;
    $input = str_replace("\n", "", fgets($in, 4096));
    if ($debug) fputs($log, "read: $input\n");
    return $input;
}

function write($line) {
    global $debug, $log;
    if ($debug) fputs($log, "write: $line\n");
    echo $line . "\n";
}
```

```

}

// Colocamos headers AGI dentro de un array
while ($env=read()) {
    $s = split(":", $env);
    $agi [str_replace("agi_", "", $s[0])] = trim($s[1]);
    if (($env == "") || ($env == "\n")) {
        break;
    }
}

// Función que conecta al banco de datos
function connect_db() {
    $db_connection = mysql_connect('localhost', 'root', 'senha') or die
(mysql_error());
    $db_select = mysql_select_db('estoque') or die (mysql_error());
}

// Programa Principal
connect_db();

// Almacena valor de la variable OPCA0 en
write("GET VARIABLE OPCA0");
$c = read();
$produto = substr($c, 14);
$produto = substr($call, 0, -1);

$query = "SELECT quantidade FROM produtos WHERE produto=' $produto' ";
$query_result = @mysql_query($query);
$result = mysql_fetch_row($query_result);

$val or = "$result[0]";

write("EXEC SayDigits \" $val or \"\n");
read();

fclose($in);
fclose($stdout);
exit;

?>

```

En `extensions.conf` podría ser hecho algo como se muestra abajo:

```

exten => s, 1, Set(OPCA0=4001)
exten => s, n, agi (nomedoagi . agi )
exten => s, n, HangUp

```

14.9.3 DeadAGI

Si usa el DeadAGI cuando usted no tiene un canal vivo, en general usted ejecuta un DeadAgi en la extensión `h` (hangup). En este caso usted debe usar el aplicativo `deadagi()` en vez de AGI.

14.9.4 FASTAGI

El Fast AGI implementa una conexión de AGI vía TCP por la puerta 4573 por default. Si usa el FastAGI en el formato (`agi://`).

Ejemplo:

```
exten => 0800400001, 1, Agi(agi://192.168.0.1)
```

Cuando la conexión TCP es perdida o desconectada, el comando AGI termina y la conexión TCP es cerrada siguiendo la desconexión de la llamada.

Este recurso puede auxiliarlo a reducir la carga de CPU de su servidor de telefonía corriendo scripts pesados en otro servidor linux.

Usted puede obtener más detalles de FastAGI en el directorio fuente, vea los archivos "agi/fastagi-test".

La empresa OrderlyCalls posee un servidor JAGI que implementa el FastAGI para el JAVA. Más informaciones en www.orderlycalls.com.

14.10 Alterando el código fuente

El Asterisk es desarrollado en lenguaje C (no C++). Enseñar como programar en C es como alterar el código fuente de Asterisk está fuera del escape de este libro. De cualquier forma para aquellos que le interese esto tienen algunas páginas que pueden ayudarlo.

Existe una buena documentación sobre el código fuente de Asterisk en:

www.asterisk.org/developers

En este sitio hay buenos consejos de como aplicar patches de corrección al Asterisk y principalmente tiene una documentación de las APIs generadas por el software Doxygen

<http://www.asterisk.org/doxygen/>

Para aquellos que están familiarizados con C puede ser interesante alterar aplicaciones existentes. No es preciso destacar los cuidados y peligros de esta práctica.

14.11 Sumario

En este capítulo usted aprendió como usar programas externos a través de interfaces con el Asterisk. Iniciamos con el comando "asterisk -rx" que permite pasar comandos para la consola. Enseguida vimos la aplicación System() que llama aplicaciones externas. AMI, Asterisk Manager Interface es uno de los mecanismos más poderosos de Asterisk y la que más se aproxima de una interfase de CTI existente en las centrales digitales. Para llamar una aplicación desde dentro del plan de discado,

usamos el AGI. Descubrimos también sus diferentes variaciones, tales como: Deadagi, para canales “muertos”, EAGI (que pasa el flujo de audio junto), FastAGI (uso de programas remotos) y el AGI simple que corre los programas en una misma máquina de Asterisk.

Exclusivo para Vicente Miguel Perez Paredes

14.12 Cuestionario

1. Cual de los siguientes no es un método de uso de interfase con el Asterisk

- AMI
- AGI
- Asterisk -rx
- System()
- External()

2. AMI, que quiere decir que Asterisk Manager Interfase permite que comandos puedan ser pasados al Asterisk vía puerto TCP. Este recurso es habilitado por default.

- Correcto
- Incorrecto

3. El AMI es muy seguro pues su autenticación es hecha usando MD5 Challenge/Response

- Verdadero
- Falso

4. Para compensar por la falta de seguridad y escalabilidad de la Interfase AMI podemos usar:

- El Ami no tiene ningún problema de seguridad y escalabilidad
- Astmanproxy
- Sysproxy

5. El FastAGI permite el uso de aplicaciones remotas vía socket TCP, normalmente en el puerto 4573.

- Falso
- Verdadero

6. El Deadagi es usado en canales activos de tipo Zap, Sip pero no puede ser usado con IAX.

- Falso
- Verdadero

7. Aplicaciones que usan el AGI solo pueden ser programadas usando PHP.

- Falso

Verdadero

8. El comando _____ muestra todas las aplicaciones disponibles para uso en AGIs

9. El comando _____ muestra todos los comandos disponibles para uso con AMI

10. Para hacer debug en un AGI usted debe usar el comando _____.

Exclusivo para Vicente Miguel Perez Paredes

Página dejada intencionalmente en blanco

Exclusivo para Vicente Miguel Perez Paredes

Asterisk Real-Time

15-1 Introducción

Como usted ya sabe la configuración de Asterisk es retirada de varios archivos de texto dentro de /etc/asterisk. A pesar de la facilidad proporcionada por la simplicidad de los archivos de configuración en textos simple, estos presentan muchas desventajas.

Podemos citar:

- Necesidad de recarga del sistema a cada alteración.
- Creación de gran volumen de usuarios.
- Dificultad en la creación de interfaces de aprovisionamiento.
- Imposibilidad de integración con las bases existentes.

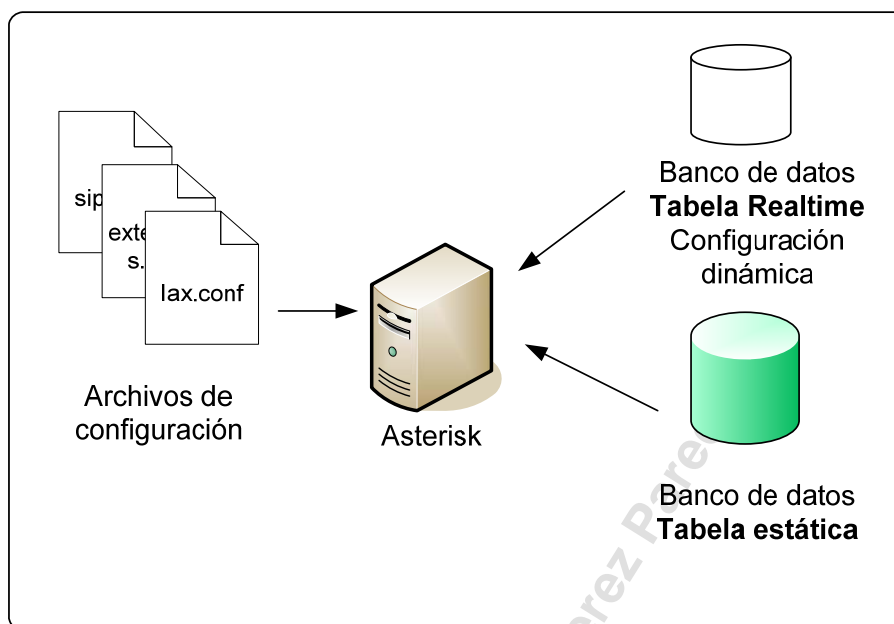
El ARA como es conocido el sistema Real-Time fue creado por Anthony Minessale II, Mark Spencer y Constantine Filin y fue proyectado para permitir una integración transparente entre la estructura de configuración interna y otras bases de datos SQL. Drivers para LDAP también están disponibles. Este sistema también es conocido como "Asterisk External Configuration" y es configurado en el archivo etc/asterisk/extconfig.conf, permitiendo a usted mapear cualquier archivo de configuración (mapeos estáticos) para una base de datos, o mapear entradas en tiempo real que permitan la creación dinámica de objetos (peers) sin la necesidad de hacer un reload.

15.2 Objetivos de este capítulo

Al final de este capítulo el lector debe estar apto para:

- Entender las ventajas y limitaciones de Asterisk Real Time.
- Instalar y configurar el ARA usando como base de datos MySQL.

15.3 Modo de funcionamiento de Asterisk Real Time



En la nueva arquitectura RealTime, todo el código específico para bancos de datos fue movido para drivers de canal. El canal apenas llama a una rutina genérica que hace una búsqueda en la base de datos. Esto en fin es mucho más simple y limpio desde el punto de vista del código fuente. El banco de datos es accedido con tres funciones:

- **STATIC:** Es usado para configurar una configuración estática cuando un módulo es cargado
- **REALTIME:** Es usado para buscar objetos durante una llamada, u otro evento.
- **UPDATE:** Es usado para actualizar los objetos.

El soporte de banco de datos en el canal no fue alterado. Existen los "peers/users" llamados normales o estáticos y los peers y users en banco de datos. Los estáticos, no importan si están cargados de un banco de datos o en un archivo de texto, son mantenidos en memoria y en el canal SIP podemos tener soporte en NAT para estos. Los "peers/users" de banco de datos no son mantenidos en la memoria. Estos solamente son cargados cuando nosotros tenemos una llamada y entonces esta es borrada, de forma que no existe soporte para NAT "keep-alives" o indicaciones de buzón de voz.

Usted puede habilitar Real Time Caching usando `rtcachefriends=yes` en el archivo `sip.conf` y permitir que sea usado NAT y MWI, de cualquier forma en este caso usted va a precisar hacer un reload si hubiera alguna alteración.

15.4 LAB 1 Instalando el Asterisk Real/Time

En este caso vamos a asumir que usted ya posee las bibliotecas MySQL instaladas en su Asterisk.

Paso 1: Baje de www.asterisk.org el paquete de add-ons.

```
#wget http://ftp.digium.com/pub/asterisk/releases/asterisk-addons-1.4.1.tar.gz
```

Paso 2: En Debian baje una de las dependencias de MySQL, se esto no ha sido hecho en lo capítulo 13.

```
#apt-get install libmysqlclient12-dev
```

Paso 3: Descomprima y compile los add-ons

```
#tar -xzvf asterisk-addons-1.4.1.tar.gz
#cd asterisk-1.4.1
#make
#. /configure
#make menuselect ; Seleccione apenas o módulo
realtime
#make install
#make samples
```

Confirme la instalación del módulo usando **module show**

15.5 Como configurar el Asterisk Real Time.

res_mysql.conf

```
[settings]
; Static configuration files:
;
; file.conf => driver,database[,table]
; queues.conf => odbc,asterisk,ast_config

; Realtime configuration engine
;
iaxusers => mysql,asteriskdb,iax_buddies
iaxpeers => mysql,asteriskdb,iax_buddies
sipusers => mysql,asteriskdb,sip_buddies
sippeers => mysql,asteriskdb,sip_buddies
voicemail => mysql,asteriskdb,voicemail
extensions => mysql,asteriskdb,extensions_table
```

El ARA es configurado en el archivo extconfig.conf. En este pueden ser notadas nítidamente dos secciones, una estática que substituye

los archivos de configuración. La segunda sección es la relacionada al Real Time que va a crear dinámicamente las entidades necesarias a su funcionamiento.

Formato del archivos extconfig.conf

```

;
; Static and real time external configuration
; engine configuration
;
; Please read doc/README.extconfig for basic table
; formatting information.
;
[settings]
;
; Static configuration files:
;
; file.conf => driver, database[, table]
;
; maps a particular configuration file to the given
; database driver, database and table (or uses the
; name of the file as the table if not specified)
;
; uncomment to load queues.conf via the odbc engine.
;
; queues.conf => odbc, asterisk, ast_config
;
; The following files CANNOT be loaded from Real time storage:
;   asterisk.conf
;   extconfig.conf (this file)
;   logger.conf
;
; Additionally, the following files cannot be loaded from
; Real time storage unless the storage driver is loaded
; early using 'preload' statements in modules.conf:
;   manager.conf
;   cdr.conf
;   rtp.conf
;
; Real time configuration engine
;
; maps a particular family of real time
; configuration to a given database driver,
; database and table (or uses the name of
; the family if the table is not specified)
;
; example => odbc, asterisk, alltable
; iaxusers => odbc, asterisk
; iaxpeers => odbc, asterisk
; sipusers => odbc, asterisk
; sippeers => odbc, asterisk
; voicemail => odbc, asterisk
; extensions => odbc, asterisk
; queues => odbc, asterisk
; queue_members => odbc, asterisk

```

La configuración estática es donde usted almacena los archivos *.conf regulares en la base de datos. Estas configuraciones son

leídas en la carga de Asterisk. Algunos módulos también releen esto cuando hace un reload. Formato de la configuración estática:

Ejemplo:

```
<conf filename> => <driver>, <database name>[, table_name]
queues.conf => odbc, asterisk, ast_config
sip.conf => mysql, asterisk
iax.conf => ldap, MyBaseDN, iax
```

Encima tenemos tres ejemplos. El primer ejemplo colará queues.conf en la tabla ast_config localizada en el contexto del banco de datos llamado asterisk usando un driver "odbc".

El segundo ejemplo irá a la tabla sip.conf (Como fue omitido el nombre de la tabla, esta tiene como estándar el nombre de archivo) en la base de datos asterisk usando un driver MySQL. La tercera irá a configurar el archivo iax.conf a la tabla iax.conf usando un driver LDAP. MyBaseDN es la base BN para la búsqueda.

Usando los ejemplos de arriba, cuando la aplicación app_queue.so carga, el driver ODBC RealTime ejecutará una consulta y tirará las informaciones necesarias. El mismo es verdadero para el chan_sip.so y así en adelante.

Ejemplo:

La configuración RealTime (segunda parte del archivos extconfig.conf) es donde configuramos la parte que va a ser cargada, actualizada y descargada en tiempo real. Por ejemplo, vamos a decir que usted tenga varios usuarios Sip definidos en su archivo sip.conf y usted quiere adicionar un tercero. Usted los adiciona al archivo y entonces ejecuta el comando "**sip reload**". Entonces el chan_sip.so relea el archivo y hace las modificaciones, así el tercer teléfono puede loguearse. Con el RealTime no es más necesario hacer el reload.

El formato de Realtime es el que sigue abajo:

```
<family name> => <driver>, <database name>[, table_name]
```

Ejemplo:

```
sippeers => mysql, asterisk, sip_peers
sipusers => mysql, asterisk, sip_users
queues => mysql, asterisk, queue_table
queue_members => mysql, asterisk, queue_member_table
voicemail => mysql, test
```

Arriba tenemos cinco líneas. En la primer línea, nosotros estamos conectando el nombre de la familia sippeers para la tabla sip_peers en la base de datos "asterisk" usando un driver MySQL. En la última línea ligamos la familia voicemail a la tabla voicemail en la base de datos test usando MySQL. Las familias sip_users y sip_peers se pueden referir a la misma tabla, si usted así lo desea.

15.6 Configuración de la base de datos

El Asterisk Real Time (también conocido como external configuration) es configurado en /etc/asterisk/extconfig.conf como vimos arriba. Ahora precisamos saber que forma precisamos crear las tablas en el banco de datos. De una forma general las columnas en su tabla deben tener los mismos campos de los archivos de configuración. Por ejemplo para un archivo de configuración de tipo SIP o IAX, la base podría ser como la de abajo:

```
[4000]
host=dynamic
secret=contraseña
context=default
context=ramales
```

Puede ser almacenado en una base de datos en una tabla como la de abajo

name	host	secret	context	ipaddr	port	regseconds
4000	dynamic	contraseña	default;ramales	10.1.1.1	4569	1765432

Para el uso con el IAX, la tabla tiene que poseer por lo menos los siguientes campos "name", "port", "regseconds". Para los otros campos usted puede configurar una columna para cada campo deseado. Si usted deseará colocar el CallerID, adicione la columna correspondiente a la tabla.

Una tabla SIP debe aparecerse como la de abajo para tener los campos necesarios.

name	Host	secret	context	ipaddr	port	regseconds	username
4000	dynamic	contraseña	default	10.1.1.1	5060	1765432	4000

Una tabla de voicemail se parecerá como a la de abajo:

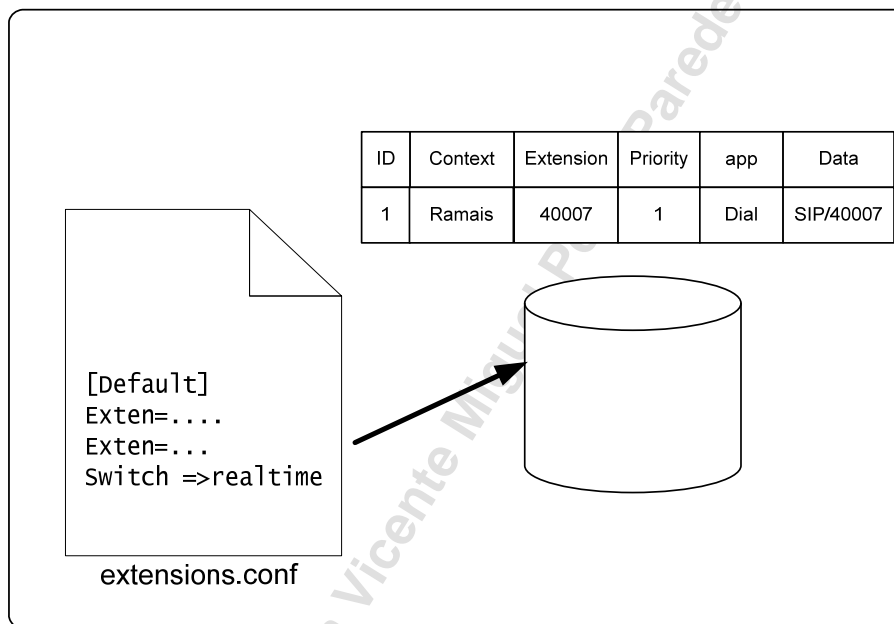
uniqueid	mailbox	context	password	Email	fullname
1	4000	default	4000	juan@silva.com	Juan Silva

El uniqueid debe ser único para cada usuario de buzón de voz y debe ser auto incrementado. El uniqueid no puede tener cualquier relación con lo buzón o el contexto.

Una tabla de extensiones se debe parecer como la que esta abajo:

context	Exten	priority	app	appdata
Ramales	4000	1	dial	SIP/4000&IAX2/4000

15.6.1 Creando un plano de marcación Realtime.



En el plan de discado usted tendrá que usar la conmutación para RealTime:

```
[ramal es]
swi tch => real ti me
```

o

```
[ramal es]
Swi tch =>real ti me/ramai s@extensi óns
```

15.7 Lab 2 - Instalando el banco y creando las tablas

En este laboratorio vamos a preparar el banco de datos para recibir los parámetros de Asterisk. Vamos a preparar apenas las tablas REALTIME. La parte estática de las configuraciones continuará en los archivos de texto.

15.7.1 Creando las tablas en MySQL.

Paso 1: Instale o servidor Mysql si no hay hecho esto en capitulo 13.

```
#apt-get install mysql-server-4.1
```

Paso 2: Entre en mysql usando

```
#mysql -u root -p
```

Paso 3: Cree una base de datos para el Asterisk

```
mysql>create database asteriskdb;
```

Paso 4: Cree un usuario con acceso a la base de datos de Asterisk.

```
mysql>grant all privileges on 'asteriskdb'.* to  
'asterisk'@'localhost' identified by 'asterisk';
```

Paso 5: Salga del mysql con el comando ´quit´ y entre nuevamente con el siguiente comando:

```
#mysql -u asterisk -p asteriskdb
```

Cuando sea solicitada la contraseña use "asterisk"

Paso 6: Vamos a crear las tablas necesarias:

```
#wget http://blog.asteriskguide.com/realtime.sql
```

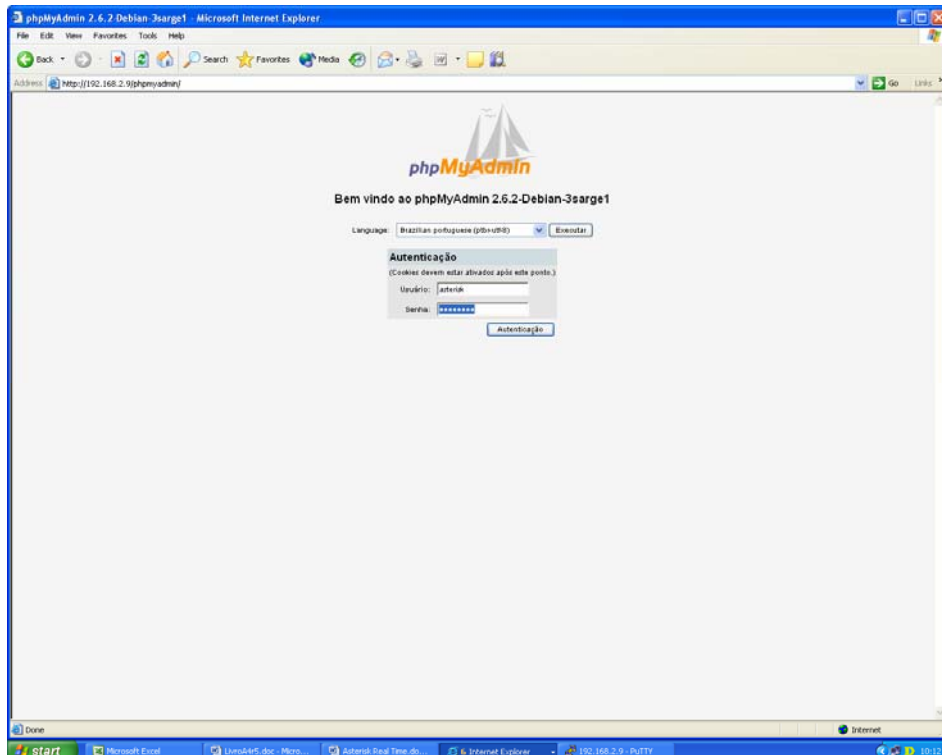
Execute os siguientes comandos no prompt do linux.

```
#mysql asteriskdb -u asterisk -p <realtime.sql
```

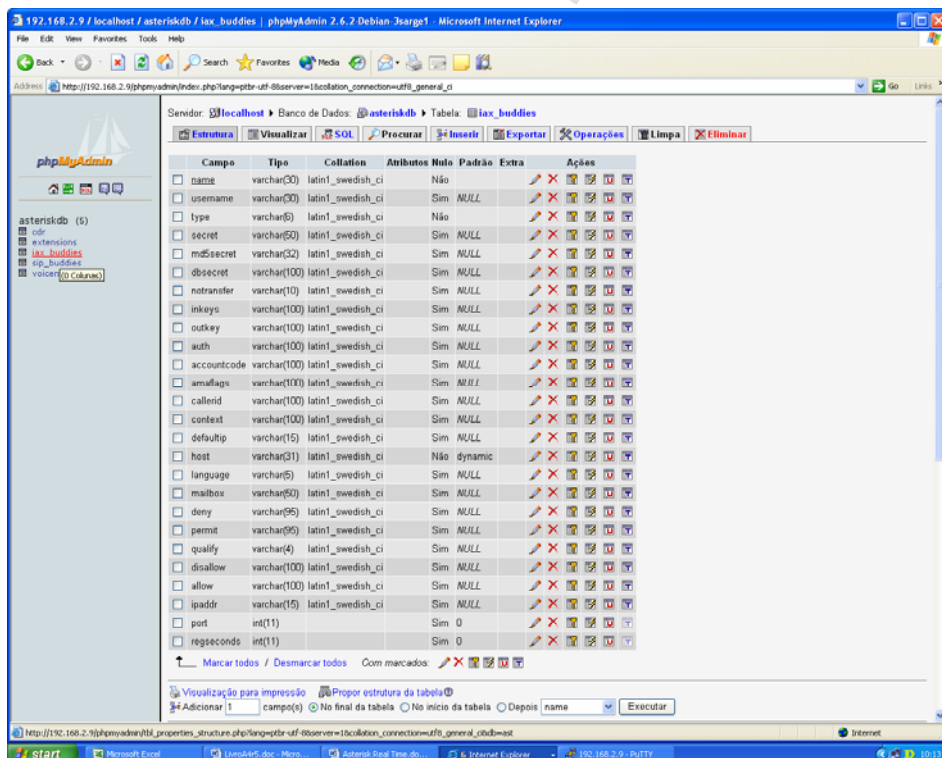
Paso 7: Instale el phpmyadmin para administrar las tablas e incluir datos.

```
#apt-get install phpmyadmin
```

Sigue abajo algunas de las pantallas de phpmyadmin:



Pantalla de login



Pantalla de una de las tablas

Paso 8: Configure el Asterisk para acceder al banco de datos creado:

Coloque los siguientes comandos en res_mysql.conf

```
[general]
dbhost = 127.0.0.1
dbname = asteriskdb
dbuser = asterisk
dbpass = asterisk
dbport = 3306
```

15.8 Lab 3 - Configurando y testeando el ARA

En este laboratorio vamos a configurar el archivo extconfig.conf para la configuración de nuestro banco de datos.

Paso 1: Configurar el extconfig.conf y reinicializar el Asterisk.

```
; Real time configuration engine
```

```
; example => odbc, asterisk, alltable
iaxusers => mysql, asteriskdb, iax_buddies
iaxpeers => mysql, asteriskdb, iax_buddies
sipusers => mysql, asteriskdb, sip_buddies
sippeers => mysql, asteriskdb, sip_buddies
voicemail => mysql, asteriskdb, voicemail
extensions => mysql, asteriskdb, extensions_table
```

Paso 2: Test de un ramal Real-Time

The screenshot shows the phpMyAdmin interface for the asteriskdb database, specifically the sip_buddies table. The table configuration is as follows:

Campo	Tipo	Funciones	Valor
id	int(11)		
name	varchar(80)		4001
accountcode	varchar(20)	<input checked="" type="checkbox"/>	
amaflags	varchar(13)	<input checked="" type="checkbox"/>	
callgroup	varchar(10)	<input checked="" type="checkbox"/>	
callend	varchar(80)	<input checked="" type="checkbox"/>	
canreinvite	char(3)	<input type="checkbox"/>	no
context	varchar(80)	<input type="checkbox"/>	default
defaultip	varchar(15)	<input checked="" type="checkbox"/>	
dtmfmode	varchar(7)	<input type="checkbox"/>	rfc2833
fromuser	varchar(80)	<input checked="" type="checkbox"/>	
fromdomain	varchar(80)	<input checked="" type="checkbox"/>	
fullcontact	varchar(80)	<input checked="" type="checkbox"/>	
host	varchar(31)	<input type="checkbox"/>	dynamic
insecure	varchar(4)	<input checked="" type="checkbox"/>	
language	char(2)	<input checked="" type="checkbox"/>	
mailbox	varchar(80)	<input checked="" type="checkbox"/>	
md5secret	varchar(80)	<input checked="" type="checkbox"/>	
nat	varchar(5)	<input type="checkbox"/>	no
deny	varchar(95)	<input checked="" type="checkbox"/>	
permit	varchar(95)	<input checked="" type="checkbox"/>	
mask	varchar(95)	<input checked="" type="checkbox"/>	
pickupgroup	varchar(10)	<input checked="" type="checkbox"/>	
port	varchar(5)	<input type="checkbox"/>	
qualify	char(2)	<input checked="" type="checkbox"/>	

Vamos a crear un ramal 4001 con contraseña 4001 en el ARA e intentar autenticar en la estación.

Paso 3: Testee una conexión de teléfono 4000 al 4001

Verifique con sip show peers el ramal. Usted va a notar que el ramal en tiempo real no aparece en la lista de peers como los estáticos. Si usted precisa usar soporte de NAT o indicador de mensajes de buzón de voz, use la instrucción "rtcachefriends=yes" en el archivo sip.conf

Paso 4: Coloque el comando "rtcachefriends=yes" en la sección [general] del archivo sip.conf.

Paso 5: Testeé nuevamente una comunicación de teléfono 4000 al 4001

Verifique el comando sip show peers. Porque este aparece ahora ?

Paso 6: Cree un nuevo ramal en la base de datos con número 4007

Haga el registro de teléfono en el 4007 sin hacer reload del Asterisk.

Paso 7: Cree las extensiones en lo banco de datos.

```
mysql -u asterisk -p
Enter password:
```

--> Utilice "asterisk" como contraseña.

Utilice o phpmyadmin para incluir la extensión en lo banco de datos o directamente en mysql como abajo.

```
use asteriskdb;
insert into extensions_table(id, context, exten, priority, app,
appdata) VALUES ('1', 'teste', '40007', '1', 'Dial', 'SIP/40007');
```

Paso 8: Cree las extensões real-time no plano de discagem

Cree la siguiente línea del contexto default.

```
swi tch => real ti me/teste@extensi ons
```

```
CLI>extensi ons rel oad
```

Passo 10: Reconfigure un dos teléfonos para la extensions 40007.

Passo 11: Disque 40007 de un teléfono existente

15.9 Sumario

En este capítulo usted aprendió que el Asterisk Real Time permite colocar las configuraciones de Asterisk en un banco de datos. Los bancos soportados son MySQL, cualquier base de datos que soporte unixODBC y LDAP. La configuración del ARA posee dos partes, estática, que substituye los archivos de configuración en texto y realtime que permite la creación dinámica de objetos en el Asterisk. Enseguida fue hecha la instalación, configuración y tests del sistema Realtime.

15.10 Cuestionario

1. El Asterisk Real Time es un recurso que hace parte de la distribución del Asterisk?

- Verdadero
- Falso

2 – Para compilar el Asterisk Real Time y usarlo con Mysql es preciso bajar que bibliotecas?

- Libmysqlclient12-dev
- Mysql-server-4.1
- Perl
- Php

3 – LA configuración de las direcciones y puertos de acceso al banco de datos MySQL es hecha en el archivo:

- extensions.conf
- sip.conf
- res_mysql.conf
- extconfig.conf

4 - El archivo exteconfig.conf es donde configuramos que tablas serán usadas por el RealTime. En este archivo tenemos dos áreas distintas para configurar

- Archivos de configuración estática
- Configuración del Realtime
- Configuración de las rutas de salida
- Configuración de los puertos y direcciones IP del banco de datos

5 – En la configuración estática, una vez hecha la carga de los objetos contenidos en la configuración estos son creados dinámicamente en el Asterisk cuando es necesario.

- Verdadero
- Falso

6 – Cuando un canal SIP (ramal) es configurado Realtime, no es posible usar recursos como “qualify” o “mwi – indicador de mensaje en espera” porque simplemente el canal no existe hasta ser usado. Esto causa los siguientes problemas:

- Este teléfono solamente podrá llamar, pero no va a recibir llamadas
- No es posible usar NAT, pues el qualify es usado para mantener el NAT abierto.
- No es posible indicar al teléfono mensajes en espera en el buzón de voz.
- No es posible usar el ramal pues la configuración del SIP es siempre estática

7 – Si yo quisiera usar la configuración Realtime con ramales SIP, pero quisiera tener soporte en NAT y MWI yo debo usar:

- El realtime no fue creado para usar con NAT
- Usted puede usar el comando `rtcachefriends=yes` en el archivo `sip.conf`
- Apenas MWI es posible
- Para usar NAT es preciso que la configuración sea estática.

8 – Usted puede continuar usando los archivos de texto de configuración, mismo después de configurar el Asterisk Real Time.

- Verdadero
- Falso

9 – El `phpmyadmin` es obligatorio en el uso de Asterisk Real Time

- Verdadero
- Falso

10 – El banco de datos tiene que ser creado obligatoriamente con todos los campos contenidos en el archivo de configuración correspondiente?

- Verdadero
- Falso

Respuesta de los ejercicios

Respuestas del capítulo 1

1. Disque las opciones correctas.

Asterisk tiene cuatro componentes básicos de arquitectura

- CANALES
- PROTOCOLOS
- AGENTES
- TELÉFONOS
- CODECS
- APLICACIONES

2. Si fuera necesario crear un PBX con 4 enlaces y ocho teléfonos, ud. puede usar una PC con Linux y 3 placas TDM400P una con cuatro canales FXO y 2 con cuatro canales FXS cada una. La afirmación es:

- CORRECTA
- INCORRECTA

3. Un canal FXS genera tono de discado, en cuanto un canal FXO recibe el tono desde la red pública o de algún otro PBX. La afirmación es:

- CORRECTA
- INCORRECTA

4. Disque las opciones correctas. Asterisk permite los siguientes recursos:

- Unidad de Respuesta Automática
- Distribución automática de llamadas
- Teléfonos IP
- Teléfonos Analógicos
- Teléfonos digitales de cualquier fabricante.

5. Para tener música en espera Asterisk necesita de un CD Player conectado en un ramal FXO. La afirmación es:

- CORRECTA
- INCORRECTA

6. Es responsable por la atención automática de clientes, normalmente suena un "prompt" y espera que usuario seleccione una opción. En algunos casos puede ser usada en conjunto con un banco de datos y conversión de texto para ser hablado.

Estamos hablando aquí de una:

- URA
- IVR
- DAC
- Unified Messaging

7 – Disque las opciones correctas, Un banco de canales es conectado al Asterisk a través de una interface:

- E1
- T1
- FXO
- FXS

8 – Disque a opción correcta. Un canal E1 soporta ____ canales de telefonía en tanto un T1 soporta ____ canales.

- 12, 24
- 30, 24
- 12,12
- 1,1

9 – En las plataformas de telefonía convencional, normalmente URA, DAC y Buzón de voz están incluidos en el PBX. Esta afirmación es:

- CORRECTA
- INCORRECTA

10 – Disque las opciones correctas, Es posible intercomunicar, usando Asterisk, varias filiales a través de voz sobre IP reduciendo el uso de comunicaciones de larga distancia. En una filial:

- Asterisk puede ser la central telefónica para todos los usuarios.
- Asterisk puede integrar una central telefónica existente
- Pueden ser usados apenas teléfonos IP ligados a un Asterisk centralizado
- Redundancia y confiabilidad no son importantes cuando se conectan IP fones.

Respuestas del capítulo 2

1. Cual es la configuración mínima para Asterisk.

Pentium 300Mhz
256MB RAM
100 MB de disco

2. Las placas de telefonía para Asterisk tienen un procesador propio (DSP), no siendo necesario de mucha CPU del servidor.

- Correcto
 Incorrecto

3. Para que la telefonía IP funcione con perfección es necesario que la red posea QoS extremo-a-extremo.

- Correcto
 Incorrecto

4. Es posible obtener una buena calidad de voz en una red que no este congestionada con switches de 100 Mbps.

- Correcto
 Incorrecto

5. Liste abajo las bibliotecas necesarias para compilar Asterisk.

gcc, ncurses, ncurses-devel, bison, Termcap, openssl, openssl-devel

6. Si usted no tiene una placa zaptel, usted precisa una fuente de tiempo. El driver ztdummy desempeña este papel aprovechando una biblioteca USB. Esto es necesario, pues algunas aplicaciones como MEETME() y MUSICONHOLD() precisan de una referencia de tiempo.

7. Cuando usted hace una instalación de Asterisk, lo mejor es no instalar los paquetes gráficos como KDE y GNOME, pues Asterisk es sensible a cuestiones de CPU y las interfaces gráfica roban muchos ciclos de CPU al servidor.

- Correcto
 Incorrecto

8. Los archivos de configuración de Asterisk se encuentran en /etc/asterisk.

9. Para instalar los archivos de configuración usted precisa ejecutar los siguientes comandos.

`#make samples`

10. Porque es importante iniciar Asterisk con un usuario diferente de root.

Seguridad

Exclusivo para Vicente Miguel Perez Paredes

Respuestas del capítulo 3

1. Son ejemplos de archivos de configuración de canales Asterisk.

- zaptel.conf
- zapata.conf
- sip.conf
- iax.conf

2. Es importante definir el contexto de los archivos de canales, pues cuando nos llega una comunicación de este canal (sip, iax, zap) al Asterisk este será tratado en el archivo extensions.conf en este contexto.

- Correcto
- Incorrecto

3. El parámetro switchtype del archivo zapata.conf define el tipo de PBX al cual Asterisk está ligado. Esto es válido para conexiones no estándar E1 con señalización ISDN PRI. Normalmente en Brasil y en Europa este estándar debe ser definido como **National**.

- Correcto
- Incorrecto

4. A pesar de ser considerado una línea digital, el E1 puede ser configurado con señalización asociada al canal (CAS) en este caso cada timeslot puede comportarse como un canal analógico FXS o FXO, por ejemplo. Esto es útil para la conexión a channel-banks (bancos de canales).

- Correcto
- Incorrecto

5. SIP Session Initiated Protocol es el protocolo de la ITU usado para conexiones de voz sobre IP. Este es bastante antiguo y viene siendo substituido recientemente por el H.323.

- Correcto
- Incorrecto

6. Dado la configuración de abajo del archivo sip.conf, en la sección [general] está definido la dirección IP 10.1.30.45, donde SIP estará esperando por conexiones. Si fuera necesario que todas las placas

de red de la máquina esperasen por una conexión SIP, bindaddr debería estar configurado para: **0.0.0.0**

```
[general]
bindport = 5060
bindaddr = 10.1.30.45
context = default
disallow = speex
disallow = ilbc
allow = ulaw
maxexpiry = 120
defaultexpiry = 80
```

7. En el archivo de abajo, los teléfonos 8000 y 8001 fueron definidos con la opción *canreinvite=no*. Con esto, cuando una comunicación es hecha de un teléfono para el otro, el Audio va directamente de un teléfono para otro sin pasar por Asterisk.

```
[8000]
type=friend
secret=8000
host=dynamic
canreinvite=no

[8001]
type=friend
secret=8000
host=dynamic
canreinvite=no
```

- Correcto
- Incorrecto

8. La principal diferencia entre el comando Playback() y el comando Background() es que el Playback() simplemente toca un mensaje y pasa al comando siguiente, en cambio el Background aguarda que usted digite algo y desvía para algún lugar en el plan de discado basándose en los dígitos discados.

- Correcto
- Incorrecto

9. Cuando una comunicación entra en Asterisk por una interfase de telefonía (FXO) sin identificación de llamada, esta comunicación es desviada para la extensión especial:

- '0'
- '9'
- 's'
- 'i'

10. Los formatos válidos para el comando Goto() son:

- Goto(contexto, extensión, prioridad)
- Goto(prioridad, contexto, extensión)
- Goto(extensión, prioridad)
- Goto(prioridad)

Exclusivo para Vicente Miguel Perez Paredes

Respuestas del Capítulo 5

1. Cite por lo menos cuatro beneficios del uso de voz sobre IP

2. Convergencia es la unificación de las redes de voz, vídeo y datos en una única red y su principal beneficio es la reducción con los costos de manutención de redes separadas.

- Correcto
- Incorrecto

3. Asterisk no puede usar simultáneamente recursos de PSTN (Red pública de telefonía y de voz sobre IP, pues los codecs no son compatibles).

- Correcto
- Incorrecto

4. La Arquitectura de Asterisk es de un SIP proxy con posibilidad de uso de otros protocolos.

- Correcto
- Incorrecto

5. Dentro del modelo OSI, los protocolos SIP, H.323 e IAX2 están en la capa de:

- Presentación
- Aplicación
- Física
- Sesión
- Enlace

6. SIP es hoy el protocolo más abierto (IETF) siendo implementado por la mayoría de los fabricantes.

- Correcto
- Incorrecto

7. H.323 es un protocolo sin expresión, poco usado fue abandonado por el mercado en favor de SIP.

- Correcto

■ Incorrecto

8. IAX2 es un protocolo propietario de Digium, a pesar de tener poca adopción por fabricantes de teléfono el IAX es excelente en las cuestiones de:

- Uso de banda ancha
- Uso de vídeo
- Pasaje por redes que poseen NAT
- Estandarizado por órganos como la IETF y ITU.

9. "Users" pueden recibir llamadas

- Correcto
- Incorrecto

10. Sobre los codecs asignarle lo que es verdadero

- G711 es el equivalente al PCM (Pulse Code Modulation) y usa 64 Kbps de banda ancha.
- G.729 es gratuito por esto es el más utilizado, usa apenas 8 Kbps de banda ancha.
- GSM viene creciendo, pues ocupa 12 Kbps de banda ancha y no precisa de licencia.
- G711 u-law es común en los EUA en tanto a-law es común en Europa y en Brasil.
- G.729 es leve y ocupa poca CPU en su codificación.

Exclusivo para Vente Digital Peres Paredes

Respuestas del capítulo 6

1. Podemos citar como principales beneficios de IAX la economía de banda ancha y facilidad de pasar por Firewalls con NAT.

- Correcto
- Incorrecto

2. En el protocolo IAX los canales de señalización y media pasan separados. Esta afirmación es:

- Correcta
- Incorrecta

3. El IAX emplea los siguientes tipos de frames

- Frame Completo
- Frame Incompleto
- Mini-Frame
- Trunked Frame

4. La banda ancha usada por el protocolo IAX es la suma de la carga de voz (payload) más los encabezados (Disque todas las que se aplican)

- IP
- UDP
- IAX
- RTP
- cRTP

5. Es importante tener la misma configuración para el payload de codec (20 à 30 ms) que la sincronización de los frames en el modo trunked (20 ms – estándar)

- Verdadero
- Falso

6. Cuando el IAX es usado en modo trunk, apenas un encabezado es usado para transmitir múltiples llamadas. La afirmación de arriba es:

- Correcta
- Incorrecta

7. El protocolo IAX2 es el más común para conectar proveedores de telefonía IP, pues pasa fácil a través de NAT. La afirmación de arriba es:

- Correcta
- Incorrecta

8. En un canal IAX como el de abajo, la opción <secret> puede ser tanto una contraseña como una **“digital key”**

```
IAX/[<user>[: <secret>]@]<peer>[: <portno>][/<exten>[@<context>][/<options>]]
```

9. El contexto es adicionado para cada cliente IAX, esto permite que diferentes clientes posean diferentes contextos. Se puede pensar en un contexto como una clase de ramal donde el cliente será colocado. La afirmación es:

- Correcta
- Incorrecta

10. Es comando IAX2 show registry muestra informaciones sobre:

- Los usuarios registrados
- Los proveedores a los cuales el Asterisk se conecta.

Exclusivo para Vicente Miguel Perez

Respuestas del capítulo 7

1. El SIP es un protocolo de tipo texto similar al _____ e _____.

- IAX
- HTTP
- H323
- SMTP

2. El SIP puede tener sesiones de tipo: (disque todos los que se aplican)

- Voz
- Buzón Electrónico
- Vídeo
- Chat
- Juegos

3. Podemos citar como componentes de SIP a: (disque todos los que se aplican)

- User Agent
- Media gateway
- PSTN Server
- Proxy Server
- Registrar Server

4. Antes que un teléfono pueda recibir llamadas, este se debe **Registrar**.

5. El SIP puede operar en modo PROXY y en modo REDIRECT, a diferencia entre ellos es que en el caso de PROXY la señalización Siempre pasa por el computador intermediario (SIP Proxy) en tanto en modo REDIRECT los clientes señalizarán directamente.

- Correcto
- Incorrecto

6. En el modo PROXY el flujo de media y la señalización pasan por el "SIP proxy" y no directamente de un cliente para otro.

- Correcto
- Incorrecto

7. El Asterisk actúa como un SIP Proxy.

- Correcto
- Incorrecto

8. La opción canreinvite=yes/no es de importancia fundamental, pues va a definir si el flujo de media va a pasar por el Asterisk o no. Esta afirmación es:

- Correcta
- Incorrecta

9. Asterisk soporta sin problemas la supresión de silencio en canales SIP. La afirmación es:

- Correcta
- Incorrecta

10. El tipo más difícil de NAT para transportar es el:

- Full Cone
- Restricted Cone
- Port Restricted Cone
- Symmetric

Exclusivo para Vicente Miguel Perez Parides

Respuestas del capítulo 8

1. En la configuración de la sesión [general] el valor estándar de la opción writeprotect=no. Siendo así si usted emite el comando save dialplan en la línea de comando CLI>. (Disque todas que se aplican).

- Asterisk sobrescribirá el extensions.conf con la configuración actual.
- Todos los comentarios serán perdidos.
- Será hecho un backup de extensions.conf en extensions.bak.
- La opción static=yes debe estar configurada para poder salvar el plan de discado.

2. Normalmente las variables globales van a estar escritas en mayúsculas en tanto las variables de canal estarán comenzando con mayúscula y las restantes en minúsculas. Esto no es obligatorio, pero facilita la lectura.

- Verdadero
- Falso

3. La extensión 's' es usada como extensión de inicio dentro de un contexto, esta es normalmente usado en los siguientes casos.

- En la entrada de una llamada sin CallerID
- Como entrada de un menú llamado por comando Background().
- En la entrada de una llamada con CallerID
- Como entrada de un contexto que fue direccionado por el Goto().

4. Cite cuatro situaciones donde contextos pueden ser utilizados:

Seguridad

Ruteamiento

Menús

Aislamiento

5. Para usar una variable en el plan de discado en Asterisk usted debe usar el siguiente formato:

- \$[varname]
- {varname}
- \$(varname)
- \${varname}

6. Las variables disponibles en el Asterisk son: (Escoja 3)

- Constantes
- Variables públicas
- Variables de ambiente
- Variables globales
- Variables privadas
- Variables de canal

7. Para obtener la compresión de un string usted puede usar la función `#{LEN(string)}`.

8. Para concatenar strings basta colocar una al lado de la otra como en el ejemplo abajo. La afirmación está:

```
#{foo}#{bar}
555#{elnumero}
#{PrefijoLargaDistancia}555#{elNumero}
```

- Correcta
- Incorrecta

9. Suponga que usted tenga una central telefónica analógica. Coloque en el espacio de abajo los comandos necesarios para direccionar la recepción de las llamadas para una telefonista en el ramal SIP/4000. Si el ramal SIP/4000 no atiende, El sistema debe direccionar después de 15 segundos para los ramales SIP/4000 y SIP/4001 simultáneamente.

```
exten=4000, 1, di al (SIP/4000, 15)
exten=4000, 2, di al (SIP/4000&SIP/4001, 15)
```

10. Suponga que usted tenga una central telefónica digital. Coloque en el espacio de abajo los comandos necesarios (`extensions.conf`) para permitir el discado externo solo para números DDD en el contexto `[ramal s]`. Coloque también los comandos necesarios para forzar el discado por la operadora 55 desconsiderando la operadora escogida por el usuario.

```
[ramal s]
exten=_0XXXXXXXXX, 1, Di al (Zap/g1/_055#{EXTEN: 4}, 15)
```

Respuestas del capítulo 9

1. Para incluir un contexto que depende del horario, usted puede usar:

```
include=> context|<times>|<weekdays>|<mdays>|<months>
```

El comando que sigue:

```
include=>expediente|08:00-18:00|mon-fri|*|*
```

■ Ejecuta las extensiones de lunes a viernes desde las 08:00 hasta las 18:00.

Ejecutan las opciones todos los días en todos los meses.

El comando es inválido.

2. Cuando el usuario disca "0" para tomar la línea el Asterisk automáticamente corta el audio. Esto es negativo porque el usuario está acostumbrado a discar el "0" y oír el tono externo de discado. Para crear este comportamiento que el usuario está acostumbrado, se puede usar comando **ignorepat=>**.

3. Los comandos:

```
exten => 8590/482518888, 1, Congesti on
exten => 8590, 1, Di al (Zap/1, 20)
exten => 8590, 2, Voi cema i (u8590)
exten => 8590, 102, Voi cema i (b8590)
```

Hace que un usuario que marco para la extensión 8590:

■ Reciba una señal de ocupado si el CallerID=482518888

Reciba una señal de ocupado independientemente del número

■ Va para el canal ZAP/1

■ Va para el VoiceMail() si el canal ZAP/1 estuviese ocupado o no atiende, excepto en el caso donde el CallerID fuera 482518888

4. Para concatenar varias extensiones basta separarlas con la señal **&**.

5. Un menú de voz normalmente es creado con el comando inicial **Background()**

6. Usted puede incluir archivos dentro de sus archivos de configuración con el comando **#include**.

7. Asterisk permite que se use el plan de discado una base de datos basada en:

- Oracle
- MySQL
- Berkley DB
- PostgreSQL

8. Cuando usted usa el comando Dial(tipo1/identificar1&tipo2/identificar2) con varios identificadores, Asterisk disca para cada una en secuencia y espera 20 segundos o el tiempo de timeout antes de pasar para el otro número. La afirmación es:

- Falsa
- Verdadera

9. En el comando Background la música de fondo tiene que ser tocada enteramente antes que el usuario pueda digitar algo. La afirmación es:

- Falsa
- Verdadera

10. Switches son usados para direccionar para otro PBX. La afirmación de arriba es:

- Correcta
- Incorrecta

11. Un macro puede ser usado para automatizar una serie de operaciones en secuencia para una extensión específica. El primer argumento pasado por la llamada del macro es el:

- \${ARG1}
- \${ENV1}
- \${V1}
- \${X}

Respuestas del capítulo 10

1. Con relación a Call Parking asigne las afirmaciones verdaderas.

- Por Default la extensión 800 es usada para Call Parking
- disque 700 para recuperar la llamada
- Por Default la extensión 700 es usada para Call parking
- Digite la extensión anunciada para recuperar la llamada

2. Para que el Call Pickup funcione es preciso que las extensiones estén en el mismo **grupo**. En el caso de extensiones Zap esto es configurado en **zapata.conf**.

3. En el caso de transferencia de llamadas existen las transferencias **ciegas**, donde el ramal de destino no es consultado antes y la transferencia **asistida** donde es posible verificar si el usuario está en la extensión.

4. Para hacer una transferencia asistida usted usa el ___ en tanto para hacer una transferencia a ciegas (blind) usted usa ____.

- #1, #2
- #2, #1
- #3, #1
- #4, #2

5. Para hacer conferencia en Asterisk es necesario usar el aplicativo MEETME().

6. Si fuese necesario administrar una conferencia, usted puede usar el aplicativo _____ y tirar un usuario de la sala.

- MeetMe()
- MeetMeConsole()
- MeetMeAdministrator()
- MeetmeAdmin()

Respuestas del capítulo 11

1. Cite cuatro estrategias de ruteamiento del sistema de fila de atención de llamadas.

Ringall, Roundrobin, Leastrecent, fewest calls, random, rrmemory

2. Es posible grabar la conversación de los agentes usando record=yes en el archivo agents.conf.

3. Para loguear un agente se usa el comando Agentlogin([agentnumber]). Cuando el agente termina la llamada este puede presionar:

- * para desconectarse y permanecer en la fila
- Colgar el teléfono y ser desconectado de la fila
- Presionar #8000 y la llamada será transferida para la fila de auditoria
- Presionar # para colgar.

4. Las tareas obligatorias para configurar una fila de atención de llamadas son:

- Crear la fila
- Crear los agentes
- Configurar los agentes
- Configurar la grabación
- Colocar en el plan de discado

5. Cual es la diferencia entre los aplicativos AgentLogin() y AgentCallBackLogin().

6. Cuando en una fila de atención de llamadas, usted puede definir un determinado número de opciones que el usuario puede discar. Esto es factible incluyendo un _____ en la fila.

- Agente
- Menu
- Contexto
- Aplicativo

7. Las aplicaciones de apoyo AddQueueMember(), AgentLogin() , AgentCallBackLogin y RemoveQueueMember() deben ser incluidas en: _____

- el Plan de discado
- la interfase de línea de comando
- en el archivo queues.conf
- en el archivo agents.conf

8. Es posible grabar los agentes, pero para esto es preciso de un grabador externo.

- Verdadero
- Falso

9. "Wrapuptime" es el tiempo que el agente precisa después del término de una llamada para prepararse para la próxima o completar procesos en relación a la llamada atendida

- Verdadero
- Falso

10. Una llamada puede ser priorizada dependiendo del CallerID dentro de una misma fila. La afirmación es:

- Correcta
- Incorrecta

Exclusivo para Vicente Miguel Paredes

Respuestas del capítulo 12

1. Los archivos emparentados en la configuración del Buzón de voz son:

- sip.conf
- iax.conf
- asterisk.conf
- voicemail.conf
- vmail.conf
- extensions.conf

2. En la aplicación VoiceMail(), los parámetros "u"y"b" son _____ y _____ respectivamente. Estos sirven para determinar que tipo de mensaje el buzón de voz tocará para el usuario.

- Ocupado, Libre
- Ocupado, No atiende
- No atiende, Ocupado
- Libre, Preso

3. La aplicación VoiceMailMain() es usada para que el usuario que llamó deje su mensaje. La afirmación es correcta:

- Correcta
- Incorrecta

4. Para salir del VoiceMailMain usted debe digitar:

- 0
- *
- #
- 9999

5. Escriba abajo la sintaxis del comando VoiceMail()

```
VoiceMail ([flags]boxnumber[@context][&boxnumber2[@context]][&boxnumber3])
```

6. En la sección [general] del archivo voicemail.conf, el parámetro attach=yes va hacer que una notificación por e-mail sea enviada al usuario con el archivo de audio anexado. La afirmación es:

- Correcta
- Incorrecta

7 La opción "delete" hace que todas los mensajes sean borrados de la casilla postal antes de que hayan sido enviadas para lo buzón del usuario.

- Correcta
- Incorrecta

8. El formato más adecuado para los mensajes de audio es el "WAV", pues tienen más soporte de estaciones Windows.

- Correcta
- Incorrecta

9. Es posible personalizar los mensajes de buzón con emailsubject y emailbody. Que variables pueden ser usadas para indicar el CallerID en el mensaje:

VM_CallerID

10. El nombre de CGI para instalar la interfase Web del voicemail es el vmail.cgi.

Exclusivo para Vicente Miguel Perez Palades

Respuestas del capítulo 13

1. Por default Asterisk contabiliza las llamadas en /var/log/asterisk/cdr-csv.

- Falso
- Verdadero

2. Asterisk solo permite contabilizar en las bases de datos:

- MySQL
- Oracle nativo
- MSSQL
- Archivos texto CSV
- Bases de datos por unix_ODBC

3. Asterisk contabiliza apenas en un tipo de almacenamiento

- Falso
- Verdadero

4. Los indicadores disponibles para billing (amaflags) son:

- Default
- Omit
- Tax
- Rate
- Billing
- Documentation

5. Si usted desea asociar un departamento o un código de contabilización al CDR usted debe usar el comando Set(CDR(accountcode)=*accountcode*). El código de la cuenta puede ser verificado por la variable $\${CDR(accountcode)}$.

6. La diferencia entre las aplicaciones NoCDR() y ResetCDR() es que el NoCDR() no genera ningún billing, en tanto que el ResetCDR() cierra el tiempo de la llamada en el registro.

- Falso
- Verdadero

7. Para usar un campo definido por el usuario (UserField) es necesario recompilar el Asterisk.

- Falso

Verdadero

8. Los tres métodos de autenticación disponibles para la aplicación Authenticate() son:

- Contraseña
- Archivos de contraseñas
- Banco berkley DB (dbput e dbget)
- VoiceMail

9. Las contraseñas de buzón son especificadas en otra parte del archivo voicemail.conf y no son las mismas del buzón de voz.

- Falso
- Verdadero

10. La opción del comando Authenticate ___ hace que la contraseña sea colocada en el código de contabilización del cdr.

- a
- j
- d
- r

Exclusivo para Vicente Miguel Pez Paredes

Respuestas del capítulo 14

1. Cual de los siguientes no es un método de uso de interfase con el Asterisk

- AMI
- AGI
- Asterisk -rx
- System()
- External()

2. AMI, que quiere decir que Asterisk Manager Interfase permite que comandos puedan ser pasados al Asterisk vía puerto TCP. Este recurso es habilitado por default.

- Correcto
- Incorrecto

3. El AMI es muy seguro pues su autenticación es hecha usando MD5 Challenge/Response

- Verdadero
- Falso

4. Para compensar por la falta de seguridad y escalabilidad de la Interfase AMI podemos usar:

- El Ami no tiene ningún problema de seguridad y escalabilidad
- Astmanproxy
- Sysproxy

5. El FastAGI permite el uso de aplicaciones remotas vía socket TCP, normalmente en el puerto 4573.

- Falso
- Verdadero

6. El Deadagi es usado en canales activos de tipo Zap, Sip pero no puede ser usado con IAX.

- Falso
- Verdadero

7. Aplicaciones que usan el AGI solo pueden ser programadas usando PHP.

- Falso
- Verdadero

8. El comando **agi show** muestra todas las aplicaciones disponibles para uso en AGIs

9 El comando **manager show commands** muestra todos los comandos disponibles para uso con AMI

10. Para hacer debug en un AGI usted debe usar el comando **agi debug**.

Exclusivo para Vicente Miguel Perez Paredes

Respuestas del capítulo 15

1. El Asterisk Real Time es un recurso que hace parte de la distribución del Asterisk?

- Verdadero
- Falso

2 – Para compilar el Asterisk Real Time y usarlo con Mysql es preciso bajar que bibliotecas?

- Libmysqlclient12-dev
- Mysql-server-4.1
- Perl
- Php

3 – LA configuración de las direcciones y puertos de acceso al banco de datos MySQL es hecha en el archivo:

- extensions.conf
- sip.conf
- res_mysql.conf
- extconfig.conf

4 - El archivo exteconfig.conf es donde configuramos que tablas serán usadas por el RealTime. En este archivo tenemos dos áreas distintas para configurar

- Archivos de configuración estática
- Configuración del Realtime
- Configuración de las rutas de salida
- Configuración de los puertos y direcciones IP del banco de datos

5 – En la configuración estática, una vez hecha la carga de los objetos contenidos en la configuración estos son creados dinámicamente en el Asterisk cuando es necesario.

- Verdadero
- Falso

6 – Cuando un canal SIP (ramal) es configurado Realtime, no es posible usar recursos como “qualify” o “mwi – indicador de mensaje en espera” porque simplemente el canal no existe hasta ser usado. Esto causa los siguientes problemas:

- Este teléfono solamente podrá llamar, pero no va a recibir llamadas
- No es posible usar NAT, pues el qualify es usado para mantener el NAT abierto.
- No es posible indicar al teléfono mensajes en espera en el buzón de voz.
- No es posible usar el ramal pues la configuración del SIP es siempre estática

7 – Si yo quisiera usar la configuración Realtime con ramales SIP, pero quisiera tener soporte en NAT y MWI yo debo usar:

- El realtime no fue creado para usar con NAT
- Usted puede usar el comando rtcachefriends=yes en el archivo sip.conf
- Apenas MWI es posible
- Para usar NAT es preciso que la configuración sea estática.

8 – Usted puede continuar usando los archivos de texto de configuración, mismo después de configurar el Asterisk Real Time.

- Verdadero
- Falso

9 – El phpmyadmin es obligatorio en el uso de Asterisk Real Time

- Verdadero
- Falso

10 – El banco de datos tiene que ser creado obligatoriamente con todos los campos contenidos en el archivo de configuración correspondiente?

- Verdadero
- Falso