

UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA



**IMPLEMENTACIÓN DE UN SERVIDOR PBX PARA
INTERCONEXIÓN DE REDES DE VOZ SOBRE IP CON
LA RED DE TELEFONÍA PÚBLICA**

INFORME DE COMPETENCIA PROFESIONAL

PARA OPTAR EL TITULO PROFESIONAL DE:

INGENIERO ELECTRONICO

PRESENTADO POR:

ALEJANDRO LEONARDO BARRIOS REATEGUI

PROMOCIÓN

2002-II

LIMA-PERÚ

2007

**IMPLEMENTACIÓN DE UN SERVIDOR PBX PARA INTERCONEXIÓN DE REDES DE
VOZ SOBRE IP CON LA RED DE TELEFONÍA PÚBLICA**

Dedicado:

**A mis padres, que me educaron
con esfuerzo y a todos los que
me ayudaron en el desarrollo de
este trabajo.**

SUMARIO

El presente Informe de competencia profesional intenta reemplazar el uso de los PBX (Public Branch eXchange) tradicionales (Centrales Telefónicas Conmutadas) con un modelo más interesante de funcionamiento, usando un servidor Linux como Servidor PBX.

Entre las ventajas que tenemos de usar este nuevo tipo de sistema es que el servidor al estar conectado, además de la PSTN (Public Switch Telephone Network), a Internet, es capaz de conectarse a redes de voz sobre IP y de funcionar como gateway (puerta de enlace) entre ambas redes. El efecto que produce esta solución es el de gran escalabilidad para crecer hacia las nuevas tecnologías de telefonía Internet. De esta manera, el servidor PBX puede tener todo tipo de extensiones, ya sea un teléfono analógico, digital, un teléfono IP e incluso softphones (teléfonos basados en aplicaciones de software) instalados en computadores personales. El manejo y administración de la central es realizada completamente a través del servidor.

Como adicional, el servidor PBX, al estar basado en Linux, puede tener automatizado varias de las tareas típicas de un PBX clásico, pero con toda la potencia y flexibilidad de un sistema UNIX. De esta manera es posible generar reportes de llamadas, configurar planes de llamada, IVR (Interactive Voice Response - Respuesta de voz interactiva), y otros más a través de una interfaz web, a la vez que es posible ejecutar tareas como scripts dentro del sistema UNIX para cualquier propósito. La ventaja comparativa de usar este tipo de servidores frente a los antiguos PBX clásicos es inmensa y el coste de tal solución relativamente bajo, gracias a la disponibilidad de la tecnología de fuente abierta y su total compatibilidad con todos los estándares de telefonía y protocolos de Voz sobre IP como a las nuevas tecnologías emergentes.

INDICE

INTRODUCCION	1
CAPITULO I. ANÁLISIS DE LOS SISTEMAS DE TELEFONÍA	4
1.1 Telefonía Clásica	4
1.1.1 Partes de un Teléfono Analógico	7
1.1.2 Telefonía Digital	8
1.1.3 Arquitectura de una Central Telefónica	24
1.1.4 Procesamiento de Llamadas	25
1.1.5 Conexión entre Centrales	25
1.1.6 Ruteo, Señalización y Protocolos	26
1.1.7 Utilización de Protocolos	27
1.2 Voz sobre IP	27
1.2.1 Definición de Voz sobre IP	29
1.2.2 Funcionamiento de Voz sobre IP	30
1.2.3 Ejemplo de Conexión de VoIP usando SIP	34
1.2.4 Conexión de muchas computadoras usando VoIP	35
1.2.5 Ventajas y Desventajas de VoIP	36
1.2.6 Implementaciones	37
CAPITULO II. DESCRIPCIÓN Y VENTAJAS DE LOS SERVIDORES LINUX	39
2.1 Ventajas de usar un Servidor Linux	40
2.2 Ventajas del uso de tecnología de fuente abierta	42
2.3 Software de fuente abierta aplicado a telefonía	44
2.3.1 Asterisk	44
2.3.2 OpenPBX	45
2.3.3 Logger	46
2.3.4 Bayonne	47
2.3.5 OpenH323	47
2.3.6 CT Server	48

2.3.7 YCTA Server.....	48
2.4 Descripción de Asterisk como Servidor PBX.....	48
2.4.1 Asterisk cierra el lazo entre la Telefonía Tradicional y la Telefonía de Redes.....	50
2.4.2 El Proyecto de Telefonía Zapata.....	50
2.4.3 Cambios masivos requieren tecnología flexible.....	51
2.4.4 Asterisk como Software PBX de fuente abierta.....	52
CAPITULO III. SISTEMA PROPUESTO.....	54
3.1 Arquitectura del Sistema.....	54
3.2 Justificación del uso de la tecnología propuesta.....	58
3.3 Ventajas comparativas de esta solución en el mercado peruano.....	59
3.4 Tecnología usada para la conexión con la PSTN.....	62
3.4.1 Tarjetas de Interfaz Analógica.....	62
3.4.2 Tarjetas de Interfaz Digital.....	64
3.4.3 Banco de Canales.....	66
3.4.4 Otros tipos de Interfaz con la PSTN.....	66
3.5 Distintos tipos de Protocolos de Voz sobre IP.....	67
3.5.1 IAX: Inter Asterisk eXchange Protocol.....	67
3.5.2 SIP: Session Initiation Protocol.....	69
3.5.3 H.323.....	71
3.5.4 MGCP.....	74
3.5.5 Jingle.....	75
3.5.6 Protocolos Proprietarios.....	76
3.6 Tecnologías Soportadas para Codificación/Decodificación de voz.....	76
3.6.1 G.711.....	77
3.6.2 G.726.....	77
3.6.3 G.723.1.....	78
3.6.4 G.729A.....	79
3.6.5 GSM.....	79
3.6.6 iLBC.....	79
3.6.7 Speex.....	80
3.6.8 MP3.....	80
3.7 Implementación del Sistema propuesto.....	80
3.7.1 Calidad de Servicio.....	80
3.7.2 Eco.....	83

3.8 Caso de Estudio: Instalación y puesta en Marcha de un Servidor PBX en IT-Synergy para el servicio de Interconexión de la Red PBX de las oficinas del PNUD de Medio Oriente	85
3.8.1 Interconexión de los PBXs	86
3.8.2 Interconexión con Asterisk	88
3.8.3 Menu de voz Interactivo y Plan de Llamadas Local y de la Red del PNUD	93
3.8.4 Aplicaciones usadas en el Sistema Asterisk para la red de VoIP del PNUD...	95
CAPITULO IV. DESCRIPCIÓN DE LAS CARACTERÍSTICAS DEL USO DEL SISTEMA PROPUESTO	96
4.1 Plan de Llamadas	96
4.1.1 Contextos (Contexts)	97
4.1.2 Extensiones (Extensions).....	98
4.1.3 Prioridades.....	99
4.1.4 Aplicaciones.....	99
4.1.5 Un Plan de Llamadas Simple	99
4.1.6 Un Plan de Llamadas más Complejo	101
4.2 Sistema de Respuestas Interactivas (IVR)	101
4.2.1 Configuración de Festival.....	102
4.2.2 Usando Festival desde el Plan de Llamadas.....	103
4.3 Auto-Attendant	103
4.4 Manejo de los distintos protocolos de Voz sobre IP	104
4.5 Puente para Conferencias.....	105
4.6 Interfase de Manejo: Panel de Operador en Flash	106
4.7 Asterisk en la Industria de las Comunicaciones	107
CAPITULO V. EVALUACIÓN DE COSTOS	112
5.1. Costo de Instalación de un PBX Clásico	112
5.1.1 Precios de Implementación de Teléfonos de Oficina.....	113
5.1.2 Precios de Sistemas Telefónicos muy pequeños	115
5.1.3 Precios de Sistemas Telefónicos pequeños	115
5.1.4 Precios de Sistemas Telefónicos de mediano tamaño	116
5.2. Costo de Instalación del Sistema Propuesto	116
CONCLUSIONES	118
ANEXOS	120
Anexo A	120
Anexo B	122
Anexo C.....	127

Anexo D.....	135
BIBLIOGRAFIA	154

INTRODUCCIÓN

Desde 1876, cuando Alexander Graham Bell inventó el teléfono, hasta hace unas pocas décadas, la tecnología utilizada para comunicarse por teléfono no había sufrido muchas transformaciones. El sistema seguía siendo el mismo, donde la transmisión se daba por medio de un par de hilos de cobre con dos terminales. Con el advenimiento posterior de las centrales telefónicas conmutadas la telefonía pública dio un salto muy grande hacia la masificación del servicio. El siguiente paso fue la aparición de las centrales telefónicas digitales que hicieron posible la mejora de la calidad del servicio.

A fines de los años 80 aparecieron también los teléfonos celulares, que supuso el otro gran salto en los sistemas de telefonía. Esta tecnología vino a complementar la telefonía pública y darle movilidad al servicio, que hasta el momento no tenía.

Los sistemas de telefonía pública y telefonía móvil han sido servicios que han usado diferente tecnología a lo largo de los años.

Con la llegada de Internet y los cambios que impuso en los sistemas de comunicación con la banda ancha, se está imponiendo en estos momentos la comunicación de voz a través de la red de computadoras, en una tecnología conocida como Voz sobre IP, que traerá consigo durante el tiempo la sustitución de los actuales equipos de comunicación de telefonía tradicional o clásica por equipos de redes.

Esta transición que se está produciendo actualmente y que durará por unos cuantos años más, hasta que las actuales redes de comunicación de telefonía clásica sean reemplazadas completamente por equipos de voz sobre IP, trae consigo la necesidad de contar con equipos que puedan proveernos la interconexión entre ambas redes, de tal manera que la migración se produzca de una manera ordenada, mientras los antiguos equipos vayan siendo lentamente desechados como obsoletos.

Las empresas telefónicas tradicionales que dominaron el mercado durante todo el siglo XX están en época de actualización e incesante competencia con las empresas de mensajería Internet que están desarrollando tecnologías de comunicación de audio y video, cubriendo el mercado cada vez más intensa y agresivamente.

Con el próximo advenimiento de la tecnología Wi-Max y su próxima instalación en ciudades enteras, dando comunicación Internet sin cables de forma masiva, la propuesta de comunicación para el próximo siglo está sobre la mesa: Toda la comunicación será a través de la Internet. La aparición de grandes proyectos de investigación de las principales empresas de software que intervienen y marcan el destino de Internet como Microsoft, Google o Yahoo que proveen de comunicación por voz y video al software de mensajería instantánea que ya usan millones de usuarios en el mundo parece ser el siguiente paso en el proceso de evolución de las comunicaciones.

La creación de aparatos portátiles que usen Google Talk, que funcionen como dispositivos móviles de comunicación en ciudades enteras, conectándose a Internet por Wi-Max, necesitando sólo un nick o cuenta de usuario para poder establecer comunicación ya están en proceso de pruebas para salir al mercado en los años que vienen.

Sin embargo, en aquellos países en los que la tecnología toma un poco más de tiempo en llegar, todavía se estarán usando los antiguos equipos de telefonía clásica, además de que la infraestructura existente en el mundo de telefonía no va a eliminarse por los nuevos equipos que son diametralmente diferentes a los actuales.

Sistemas como Skype, que tratan de interconectar ambas tecnologías, proveyendo al usuario de una manera de establecer contacto desde un equipo de comunicación de Internet como una computadora con un teléfono publico están en completo auge.

Tomará un tiempo hasta que la telefonía Internet con Voz sobre IP alcance su masa crítica, pero hay un tremendo movimiento en esa dirección. Muchas organizaciones están no solo atraídas por la promesa de bajos costos del Voz sobre IP, sino también a su habilidad de mover datos, imágenes tráfico de voz sobre la misma conexión. Una sola llamada telefónica por Internet puede llevar la forma de compartir información a un completo nuevo nivel.

Esa es la razón por la que muchos administradores de Red y desarrolladores en el campo de Tecnologías de Información están buscando activamente configurar sistemas de VoIP privados dentro de su empresa. La eficiencia que aquellas redes de usuarios pueden alcanzar es casi ilimitada, y además de bajo costo, si el sistema es construido sobre software de fuente abierta como Asterisk. Hay opciones de VoIP comerciales, pero muchos de ellos son sistemas caros que se están volviendo obsoletos, que poseen código complicado y se ejecutan sobre un hardware obsoleto. Asterisk corre sobre Linux y puede interoperar con casi todos los estándares de equipos de telefonía, además de poder ser reprogramado a voluntad y con mucha facilidad a través de una interfaz web, que hace innecesaria la intervención de algún técnico experto.

Este es el punto de partida del presente trabajo, donde se detallará la manera como se puede construir un servidor PBX Asterisk de fuente abierta, de bajo costo, y que haga posible la interconexión entre ambas tecnologías – la telefonía tradicional y la de Voz sobre IP – que es una solución completamente eficiente y profesional para cualquier empresa que esté dispuesta a entrar con fuerza en el nuevo mundo de las telecomunicaciones a base de la conmutación de paquetes, sin dejar de lado la interconexión con los sistemas de telefonía clásica.

El presente informe detalla también un caso de estudio real, donde se pone a prueba los conceptos vertidos y explicados en los primeros capítulos. Este Caso de Estudio es un Proyecto que se realizó con el Programa de las Naciones Unidas para el Desarrollo, en sus distintas oficinas localizadas en Medio Oriente.

CAPITULO I

ANÁLISIS DE LOS SISTEMAS DE TELEFONÍA 1.1 Telefonía Clásica

El servicio telefónico es, junto con la red eléctrica, uno de los más confiables que conocemos y usamos, ya que todo es muy redundante y está pensado para funcionar siempre. Una central telefónica esta diseñada para minimizar los tiempos de interrupción del servicio.

Es una tecnología en que la interfaz es muy importante, la gente la conoce, espera que cuando levanta el auricular se escuche el tono, y si no es el mismo que el que esperaba escuchar, entonces algo puede andar mal; además es muy universal y difundida. Todo esto se tiene en cuenta a la hora de prestar el servicio telefónico.

En el principio, la red telefónica básica (RTB) fue creada para transmitir la voz humana. Tanto por la naturaleza de la información a transmitir, como por la tecnología disponible en la época en que fue creada, es de tipo analógico. Hasta hace poco se denominaba RTC (Red Telefónica Conmutada o PSTN por sus siglas en ingles), pero la aparición del sistema RDSI (digital pero basado también en la conmutación de circuitos), ha hecho que se prefiera utilizar la terminología RTB para la primitiva red telefónica (analógica), reservando las siglas RTC para las redes conmutadas de cualquier tipo (analógicas y digitales); así pues, la RTC incluye la primitiva RTB y la moderna RDSI (Red Digital de Servicios Integrados).

La conmutación de circuitos telefónicos supone que, en un determinado instante, se establecen conexiones entre una serie de líneas que comienzan en el emisor y terminan en el receptor, de tal forma, que mientras dura la llamada hay una continuidad entre ambos puntos, lo que hace posible la comunicación. Cuando esta se termina, los enlaces se rompen, y muchas de estas líneas son utilizadas de nuevo con otro esquema de

conexiones para transmitir entre otro par de puntos. El que una misma línea se utilice secuencialmente para muchas llamadas distintas es lo que hace posible la conocida "saturación en la línea", cuando demasiada gente pretende utilizar los mismos conductores.

Las clásicas líneas de RTB, la que tenemos en el teléfono de casa, tienen cada una un número (su dirección telefónica) y están físicamente construidas por dos hilos (conocidos como par de cobre), que se extienden desde la central telefónica hasta la instalación del abonado (se conoce también como bucle de abonado). Cada central atiende las líneas de abonado de un área geográfica determinada. A su vez, las centrales telefónicas están unidas entre sí. Esta unión de centrales constituye el sistema telefónico nacional que a su vez está enlazado con los restantes del mundo.

Puede darse el caso que un abonado disponga de varias líneas que responden a un solo número, lo que se consigue mediante un artificio denominado grupo de salto, en el que la compañía telefónica envía la llamada a una primera línea (número de cabecera), y si está ocupado, traspasa la llamada a la siguiente línea que haya incluido en el grupo de salto (cada línea tiene un número distinto, que es manejado internamente por la compañía).

La RTB original era de funcionamiento completamente analógico, primero de conmutación humana (telefonistas); después de conmutación automática (electromecánica). En cualquier caso, las antiguas conexiones puramente analógicas eran propensas al ruido, a las pérdidas de conexión, y no se prestaban fácilmente al establecimiento de conexiones de larga distancia. Por estas causas, a principios de los 60, el sistema telefónico fue transformándose gradualmente en un sistema digital basado en conmutación de paquetes, al mismo tiempo que fueron sustituyéndose gradualmente las primitivas y gigantescas centrales telefónicas convencionales por otras más modernas de funcionamiento digital (1).

No hay que confundir "línea analógica en central digital" con "línea digital". La primera, sigue siendo totalmente analógica, aunque esté conectada a una central digital donde los sistemas de conmutación ya no son de tipo electromecánico. En este caso la central digital solo proporciona algunas pequeñas ventajas adicionales; posibilidad de marcar por tonos, llamada en espera, facturación detallada, buzón de voz, etc. A estas líneas solo se pueden conectar dispositivos telefónicos de tipo analógico (teléfonos, módems, máquinas de fax de grupo III, etc.). La línea digital por contra, solo transporta ceros y

unos (mejor sería decir dos niveles de tensión o de luz) y por supuesto solo permite la conexión de dispositivos de este tipo.

La situación a inicios del siglo XXI para la RTB podríamos calificarla como híbrida; lo normal es que la transmisión sea todavía analógica en los bucles de abonado de ambos extremos y digital en su tráfico entre centrales (esto requiere una doble conversión, analógico-digital y digital-analógico). Para su digitalización, la señal analógica es muestreada a 8.000 veces por segundo (8 KHz.), el valor de cada muestra puede ser un valor entre 0 y 255 (puede ser representada por 1 byte -octeto-) lo que supone un flujo de datos de 8 KB/s o 64 Kbps, la que se denomina calidad de sonido "telefónico".

En realidad, la utilización de técnicas de muestreo de frecuencia variable y de compresión de datos, permiten que, en la práctica, el ancho de banda utilizado por una conversación telefónica normal pueda ser solo una fracción de esta cantidad (del orden de 1/8) sin que por ello se alcance una pérdida de la calidad que sea apreciable por los interlocutores.

Los bucles de abonado de cualquier tipo RTB o RDSI tienen dos partes: Externa e Interna. La primera, desde la central hasta el comienzo de la instalación del abonado, donde existe un dispositivo conocido como PTR (Punto de Terminación de Red). Esta parte externa de la instalación es responsabilidad de la compañía telefónica que se encarga de su conservación y mantenimiento. La parte interna constituye la parte de instalación en el interior del local del abonado y es propiedad de este, siendo también suya la responsabilidad de su instalación y conservación. Esta parte termina en las conocidas rosetas con conectores RJ-11 que se instalan en las habitaciones, a los que conectamos el cable del teléfono (estos conectores tienen capacidad para cuatro hilos, aunque en realidad solo se utilizan los dos contactos centrales).

Una característica de la instalación de abonado de los bucles RTB, es que dentro de ciertos límites, se pueden conectar varios dispositivos en paralelo (manteniendo una impedancia mínima), mientras que en las líneas RDSI esto no es tan sencillo.

En cualquier caso, la desventaja principal de la RTB es precisamente su carácter analógico (al menos en los bucles de abonado), ya que debido a su propia naturaleza, este tipo de señales tiende a degradarse, en especial las componentes de alta frecuencia. Además cada conversión supone una posibilidad adicional de distorsión de la señal.

A pesar de que los humanos pueden percibir vibraciones de sonido en el rango de 20-20000Hz, la mayoría de sonidos que hacemos cuando hablamos tienden a estar en el rango de 250-3000Hz. Dado que el teléfono ha sido diseñado para transmitir sonidos entre personas que hablan, fue diseñado con un ancho de banda en el rango de 300-3500Hz. Este limitado ancho de banda significa que alguna calidad en el sonido se perderá especialmente en las altas frecuencias.

1.1.1 Partes de un Teléfono Analógico

Un teléfono analógico está compuesto de 5 partes: el timbre, la unidad de marcación, el circuito de regulación de la transmisión, el conmutador de saltos y el auricular.

- **Timbre:** Es el elemento que produce una señal auditiva fuerte para avisar que hay una llamada entrante. Se activa con corriente alterna proveniente de la central telefónica a la que está conectado el aparato. La señal eléctrica se origina en esta central cuando su equipo conmutador recibe la señal procedente del aparato que inició la comunicación.
- **Unidad de Marcación:** La marcación en un aparato de teclado (teléfono digital) se lleva a cabo por medio de la suma de frecuencias, según la tabla a continuación:

TABLA No 1.1. Tabla de frecuencia de marcados

	1209 Hz	1336 Hz	1477 Hz	1633 Hz ^a
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Según la tabla anterior, por ejemplo, si el usuario marca el dígito "2", la señal que procesará el equipo conmutador de la central será la suma de 1 336Hz (vertical) y 697Hz (horizontal); consecuentemente la señal será de 2 033Hz. Así ocurrirá en cada caso según el dígito marcado.

La transmisión de un dígito en un teléfono analógico tarda 1,5 segundos, mientras que en un teléfono digital tarda tan solo 0,7 segundos.

- **Circuito de regulación de transmisión:** Su función es mantener la intensidad acústica dentro de límites predeterminados para que la calidad de la comunicación sea la adecuada. Consiste de una serie de elementos eléctricos o electrónicos varios, como condensadores, varistores y resistencias. El circuito de regulación de transmisión opera automáticamente, permitiendo que las señales en el teléfono siempre tengan una intensidad adecuada para escuchar normalmente, en forma independiente de la intensidad que provenga de la línea telefónica.
- **Conmutador de saltos:** Este dispositivo señala el estado del circuito telefónico a la central. Cuando se levanta el auricular, el conmutador de saltos cierra el circuito entre el usuario y la central, el cual es visto como un pedido de tono para marcar. Cuando se cuelga, el conmutador de saltos abre el circuito, indicando que ha terminado la llamada. El conmutador de saltos también puede ser usado para señalar propósitos. Algunos teléfonos electrónicos analógicos tienen un botón etiquetado como "Link" que causa un evento llamado flash. Se puede efectuar un flash manualmente presionando el conmutador de saltos por una duración de entre 200 a 1200 milisegundos. Si se deja abajo por un tiempo mayor, el carrier asumirá que se ha colgado.
- **El auricular:** Está compuesto por el receptor y el transmisor. Se encarga de efectuar la conversión entre la energía sonora que los humanos usan y la energía eléctrica que se usa en la red telefónica.

1.1.2 Telefonía Digital

La telefonía analógica está casi muerta. En la RTP, o en inglés PSTN, la famosa Última Milla es la pieza remanente de la red telefónica que usa tecnología que se inventó cien años atrás.

Uno de los desafíos primarios cuando se transmiten señales analógicas es que todo tipo de cosas pueden interferir con esas señales, causando bajo volumen, estática, y otros indeseados efectos. En lugar de tratar de preservar una onda analógica sobre distancias que pueden recorrer miles de kilómetros, se miden las características del sonido original y se envía la información al lado contrario. La onda original no llegaría hasta allá, pero si la información necesaria para reconstruirla.

Este es el principio de todo audio digital (incluyendo la telefonía): tomar una muestra de las características de la onda fuente, almacenar la información medida, y enviar los datos al otro extremo. Luego, al otro lado, usar la información transmitida para generar una

completa y nueva señal de audio que tiene las mismas características que la original. La reproducción es tan buena que el oído humano no puede notar la diferencia.

La principal ventaja del audio digital es que la información muestreada puede ser matemáticamente chequeada por errores a través de la ruta hacia el destino, asegurando un duplicado perfecto del original al otro extremo. Las distancias no afectarían mas la calidad, y la interferencia puede ser detectada y eliminada.

1.1.2.1 Modulación por código de Pulsos (PCM)

Hay muchas maneras para codificar digitalmente el audio, pero el método más común (y el único usado en los sistemas de telefonía) es conocido como Modulación por Código de Pulsos (PCM – Pulse Code Modulation). Para ilustrar cómo funciona vamos por unos ejemplos:

a) Codificando digitalmente una onda analógica

El principio del PCM es que la amplitud de la onda analógica es muestreada a intervalos específicos para que luego pueda ser recreada. La cantidad de detalle que es capturada depende de la resolución de bits de cada muestra y de que tan frecuente son las muestras tomadas. Una alta resolución de bits y una alta tasa de muestreo proveerán una exactitud mayor, pero mayor ancho de banda se requerirá para transmitir esta información mas detallada.

Veamos cómo funciona el PCM. Considerando la onda mostrada en la Figura 1.1.

Para codificar digitalmente una onda, esta debe de muestrearse en una base regular, y la amplitud de la onda en cada momento en el tiempo debe de ser medida. El proceso de partir una onda en momentos de tiempo y medir la energía en cada momento es llamado muestreo.

Las muestras necesitarán ser tomadas frecuentemente y necesitarán capturar suficiente información para asegurarse que el lado opuesto pueda recrear una onda suficientemente similar. Para lograr una muestra más exacta, se requieren tener más bits. Para explicar este concepto, comenzaremos con una resolución baja, usando cuatro bits para representar nuestra amplitud.

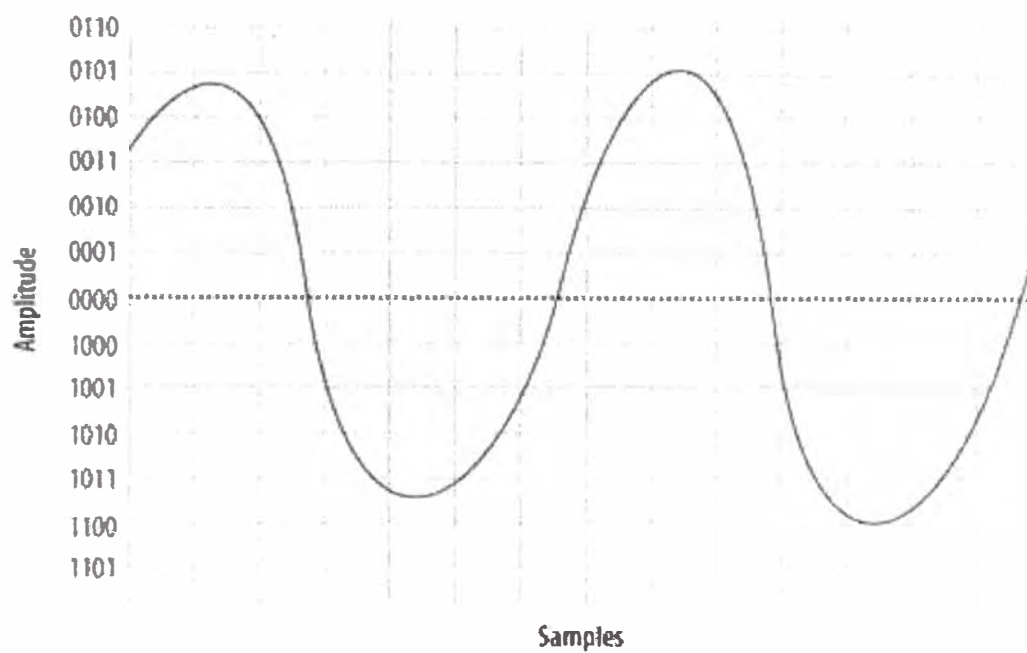


Figura No. 1.1. Onda de entrada

La siguiente figura muestra la información que será capturada cuando se muestrea la onda sinusoidal a una resolución de cuatro bits.

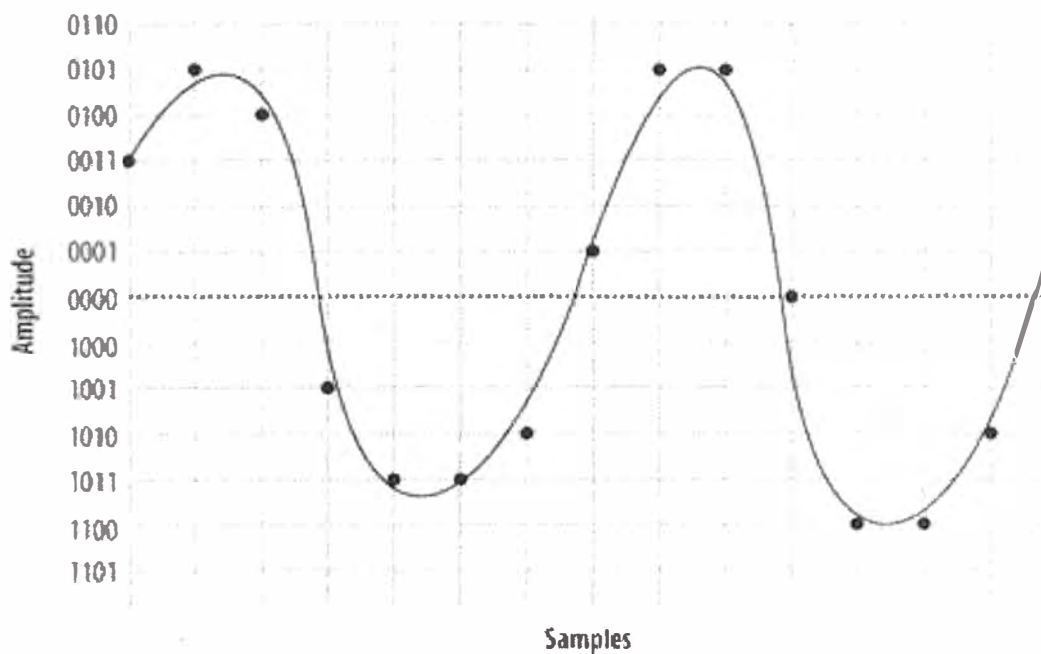


Figura No. 1.2. Puntos de muestreo

En cada intervalo, se mide la amplitud de la onda se graba la correspondiente intensidad. En otras palabras, se muestrea. Se puede notar que la resolución de cuatro bits limita la exactitud. El primer muestreo tiene que ser redondeado a 0011, y la siguiente cuantización envía una muestra de 0101. Luego viene 0100, 1011, y las demás. En total, tenemos 14 muestras (en realidad, muchas miles de muestras deben de ser tomadas por segundo). Si se encadenan juntos todos los valores, podemos enviarlo al otro extremo como:

```
0011 0101 0100 1001 1011 1011 1010 0001 0101 0101 0000 1100 1100 1010
```

Figura No. 1.3. Muestra digitalizada de la onda de entrada

En el cable, el código puede lucir algo como se muestra en la Figura 1.4:

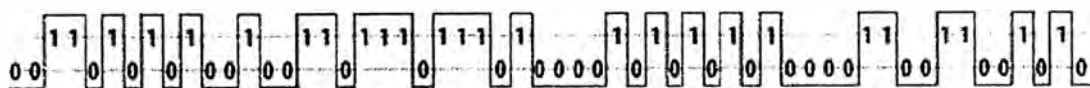


Figura No. 1.4. Muestra digitalizada en el cable

Cuando el convertidor digital/analógico (D/A) del otro extremo recibe la señal, esta puede usar la información para plotear las muestras, como mostradas en la figura que viene a continuación.

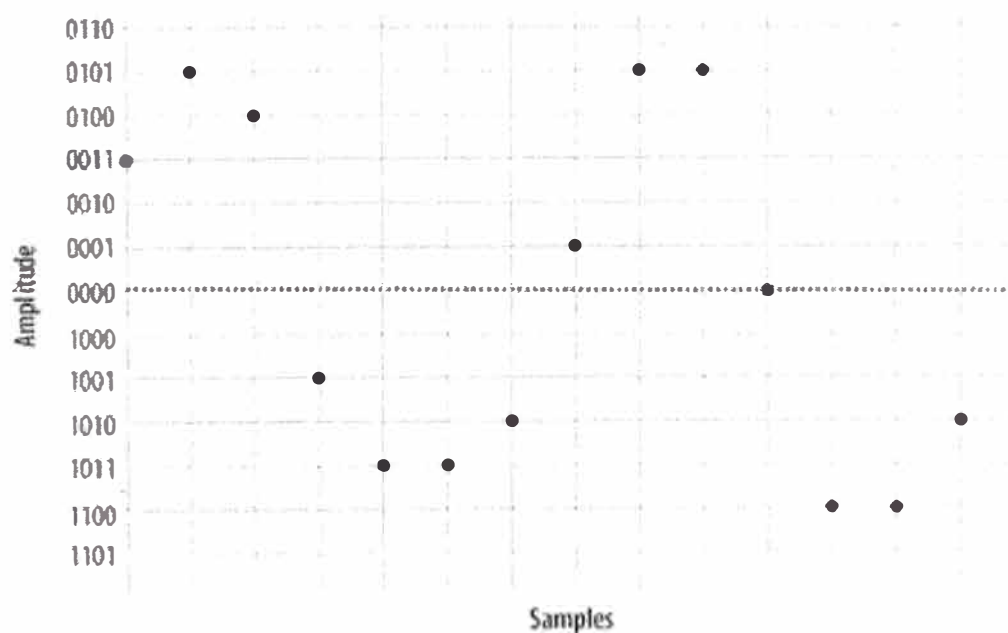


Figura No. 1.5. Puntos para reconstrucción de la onda de entrada

De esta información, la onda puede ser reconstruida de esta manera (Figura 1.6):

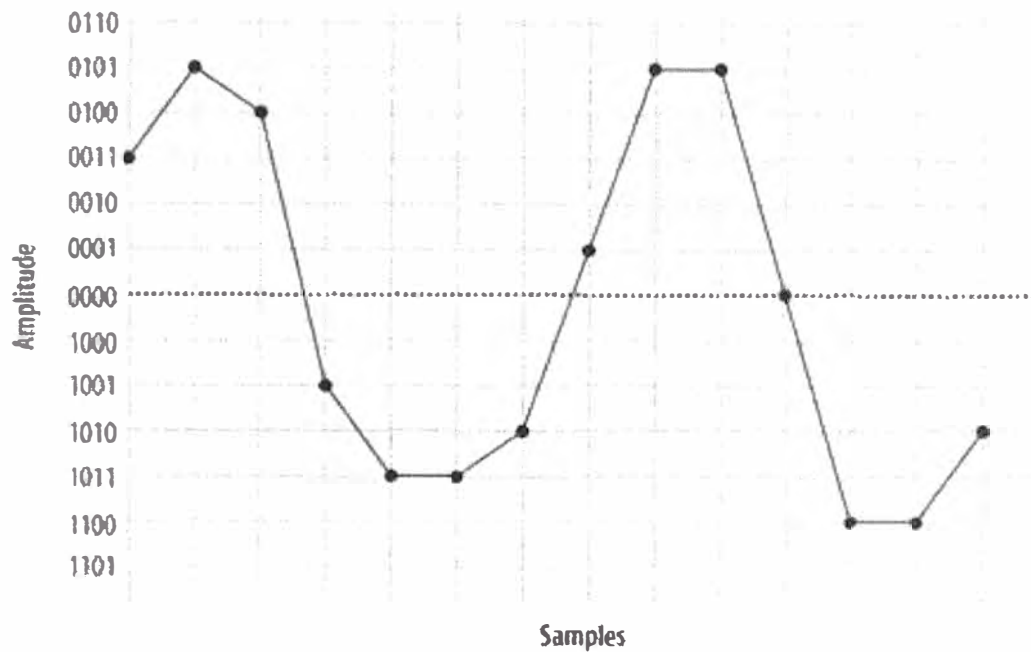


Figura No. 1.6. Onda de entrada reconstruida al otro extremo de la línea.

Como se puede apreciar en la figura anterior, la reconstrucción de la onda no es muy exacta. Esto fue hecho para demostrar un punto importante: La calidad de la onda digitalmente codificada es afectada por la resolución y la tasa a la cual es muestreada. Una tasa de muestreo demasiado baja, y con demasiada resolución de muestreo producirá una calidad de audio nada aceptable.

b) Incrementando la resolución y la tasa de muestreo

Tomando una mirada nuevamente a la onda original, esta vez usando cinco bits para definir los intervalos de cuantización, se procederá a un nuevo muestreo. Veamos nuevamente la onda de entrada en la Figura 1.7.

En realidad no existe tal como un PCM de cinco bits. En la red de telefonía, las muestras de PCM son codificadas usando 8 bits.

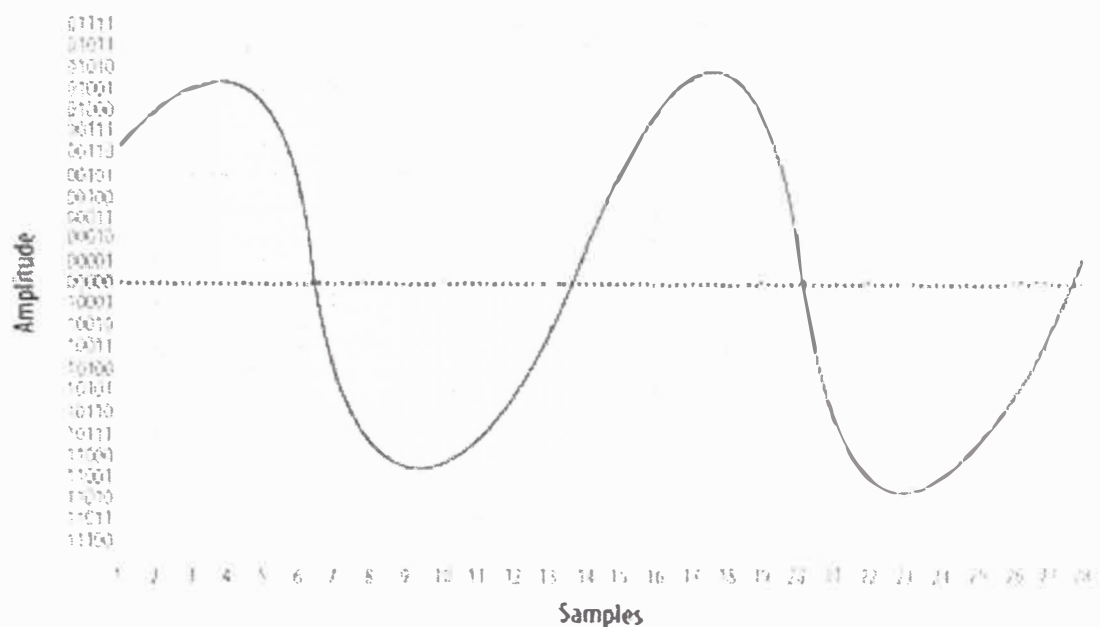


Figura No. 1.7. Onda de entrada lista para el muestreo.

Y doblando también nuestra frecuencia de muestreo, los puntos ploteados están mostrados en la siguiente figura:

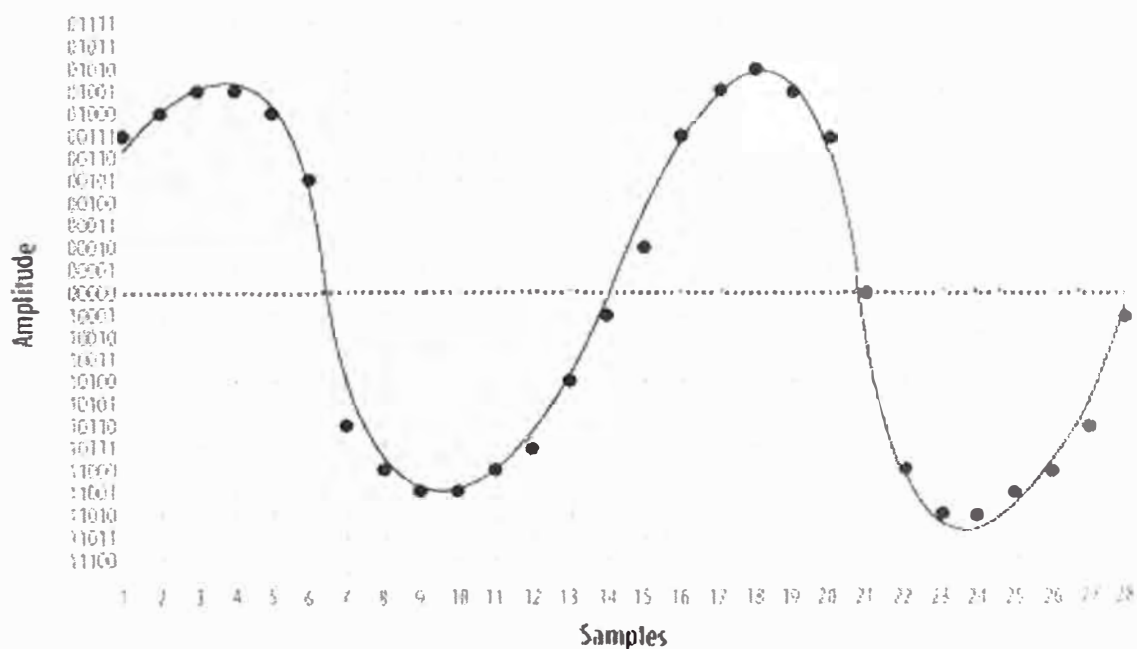


Figura No. 1.8. Recolección de los puntos de muestreo

Ahora tenemos el doble de número de muestras, al doble de resolución. Ellos son:

```

00111 01000 01001 01001 01000 00101 10110 11000 11001 11001 11000 10111
10100 10001 00010 00111 01001 01010 01001 00111 00000 11000 11010 11010
11001 11000 10110 10001

```

Figura No. 1.9. Formato digital de los puntos muestreados.

Cuando la información se reciba al otro extremo, podrá ser ploteada como se muestra en la siguiente figura:

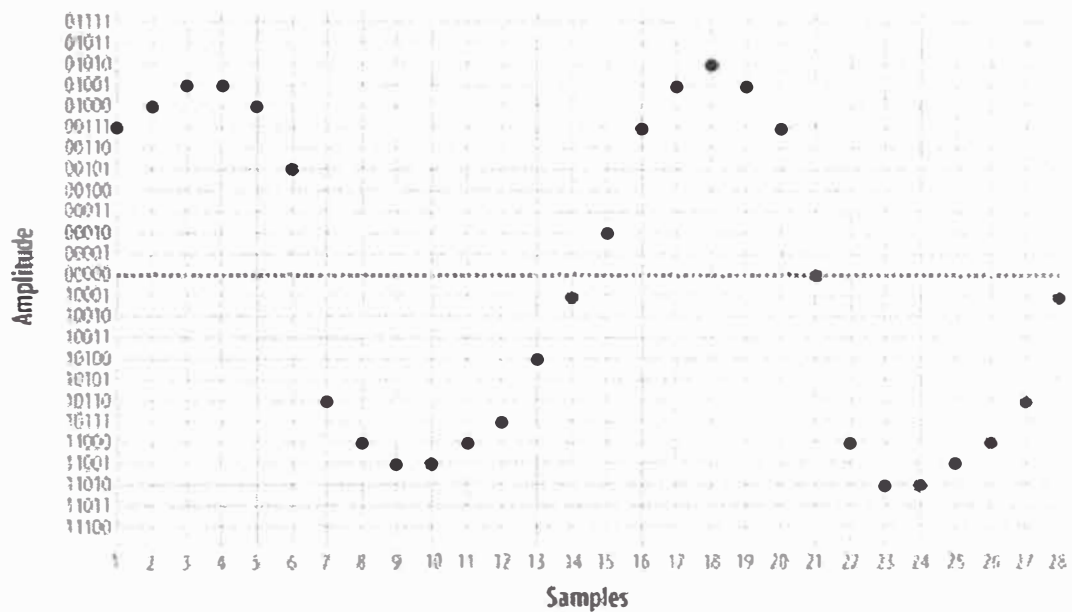


Figura No. 1.10. Puntos para reconstrucción de la onda de entrada

Con esta información de bits se puede generar la onda de la Figura 1.11.

Como se puede apreciar, la onda resultante es de lejos mucho más representativa de la onda original. Sin embargo, se puede también ver que aun hay lugar para mejoras.

Se debe también notar, que se requirió 40 bits para codificar la onda a una resolución de 4-bits, mientras que 156 bits para enviar la misma onda a una resolución de 5-bits (y además doblando la tasa de muestreo). Entonces hay un costo: A mayor calidad de audio que se desee codificar, mayor cantidad bits se requieren para hacerlo, y a mayor cantidad de bits que se desea enviar (en tiempo real, naturalmente), mayor ancho de banda que se consume.

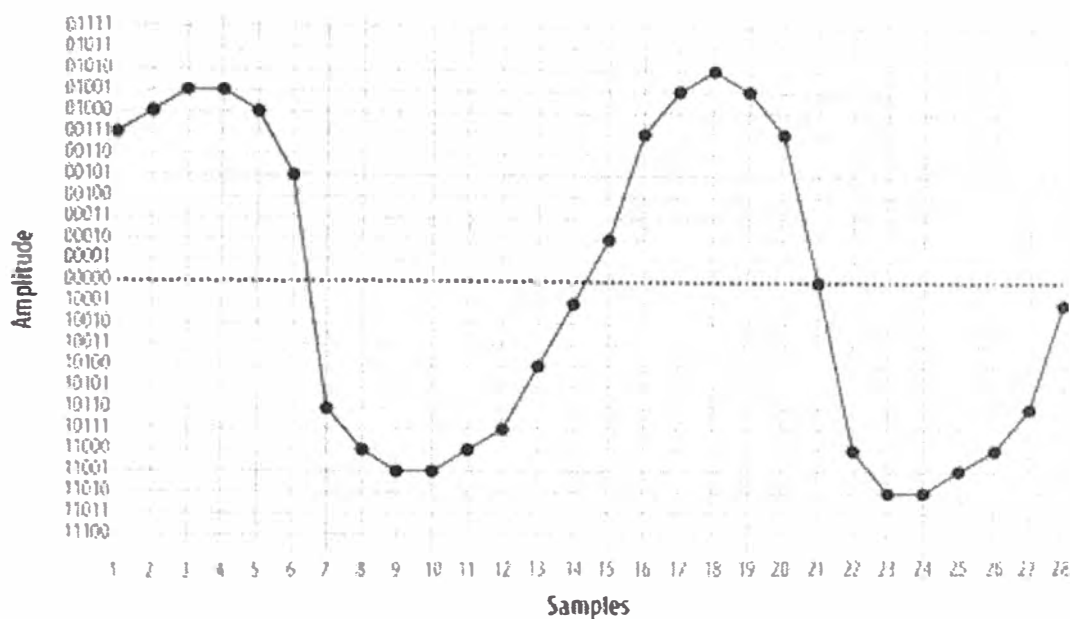


Figura No. 1.11. Onda de entrada reconstruida al otro lado de la línea.

c) Teorema de Nyquist

Entonces, cuánto muestreo es suficiente para digitalizar una señal analógica? Esta misma pregunta fue considerada en los años 1920s por un ingeniero eléctrico (y empleado de AT&T/Bell) llamado Harry Nyquist.

El Teorema de Nyquist enuncia: *“Cuando se muestrea una señal, la frecuencia de muestreo debe de ser mas grande que el ancho de banda de la señal de entrada para poder reconstruir la original perfectamente de la versión muestreada”.*

Nyquist publico dos trabajos (1): *“Certain Factors Affecting Telegraph Speed”* (1924) y *“Certain Topics in Telegraph Transmission Theory”* (1928), en los cuales postulaba lo que luego se conocería como el Teorema de Nyquist. Probado en 1949 por (2) Claude Shannon (*“Communication in the Presence of Noise”*), también es referido como el Teorema de Muestreo de Nyquist-Shannon.

En esencia, lo que quiere decir el teorema de muestreo es que para codificar con exactitud una señal analógica, se tiene que muestrear doblemente más frecuente como el total de ancho de banda que se desea reproducir. Como la red telefónica no lleva frecuencias menores a 300Hz ni mayores a 4000Hz, una frecuencia de muestreo de 8000 muestras por segundo será suficiente para reproducir cualquier frecuencia dentro del ancho de banda de un teléfono analógico.

d) Companding Logarítmico

Hemos discutido que mayores intervalos de cuantización (por ejemplo mayores tasas de muestreo) dan una mejor calidad pero también requieren mayor ancho de banda. En el caso del teléfono, la mínima tasa de muestreo para exactamente medir el rango de frecuencias que se va a transmitir es de 8Khz. Esto va a provocar que se tenga que mandar muchos bits de datos en el cable.

Companding es un método para mejorar el rango dinámico de un método de muestreo sin perder exactitud importante. Esto funciona cuantizando las amplitudes mayores de una manera más gruesa que las bajas amplitudes. En otras palabras, si se grita en el teléfono, la señal no será muestreada tan limpiamente como será cuando se habla normalmente. Gritar no es además nada bueno para la presión sanguínea, así que es mejor evitarla.

Dos métodos de *companding* son comúnmente empleados: u-law en Norte América, y A-law en el resto del mundo. Ellos operan en la misma base de principios pero no son de ningún modo compatibles entre ambos.

El método de *companding* divide la onda en cuerdas, cada una de las cuales tiene muchos pasos. La cuantización involucra emparejar las amplitudes medidas a un paso apropiado dentro de una cuerda. El valor de la banda y el número de cuerdas (tanto como el signo positivo o negativo) se convierten en la señal. Los siguientes diagramas darán una idea visual de lo que el *companding* hace. Ellos no están basados en ningún estándar, pero en lugar de ello, se ha hecho con el propósito de ilustración (otra vez, el *companding* en la red telefónica será hecho a una resolución de 8-bits, no 5-bits).

La siguiente figura ilustra un *companding* de 5-bits. Como se puede apreciar, las amplitudes cercanas al punto de cruce en cero serán muestreadas mucho más exactamente que las amplitudes mayores (sean positivas o negativas). Sin embargo, ya que el oído humano, el transmisor, y el receptor también tienden a distorsionar las señales fuertes, esto no vendría a ser ningún problema.

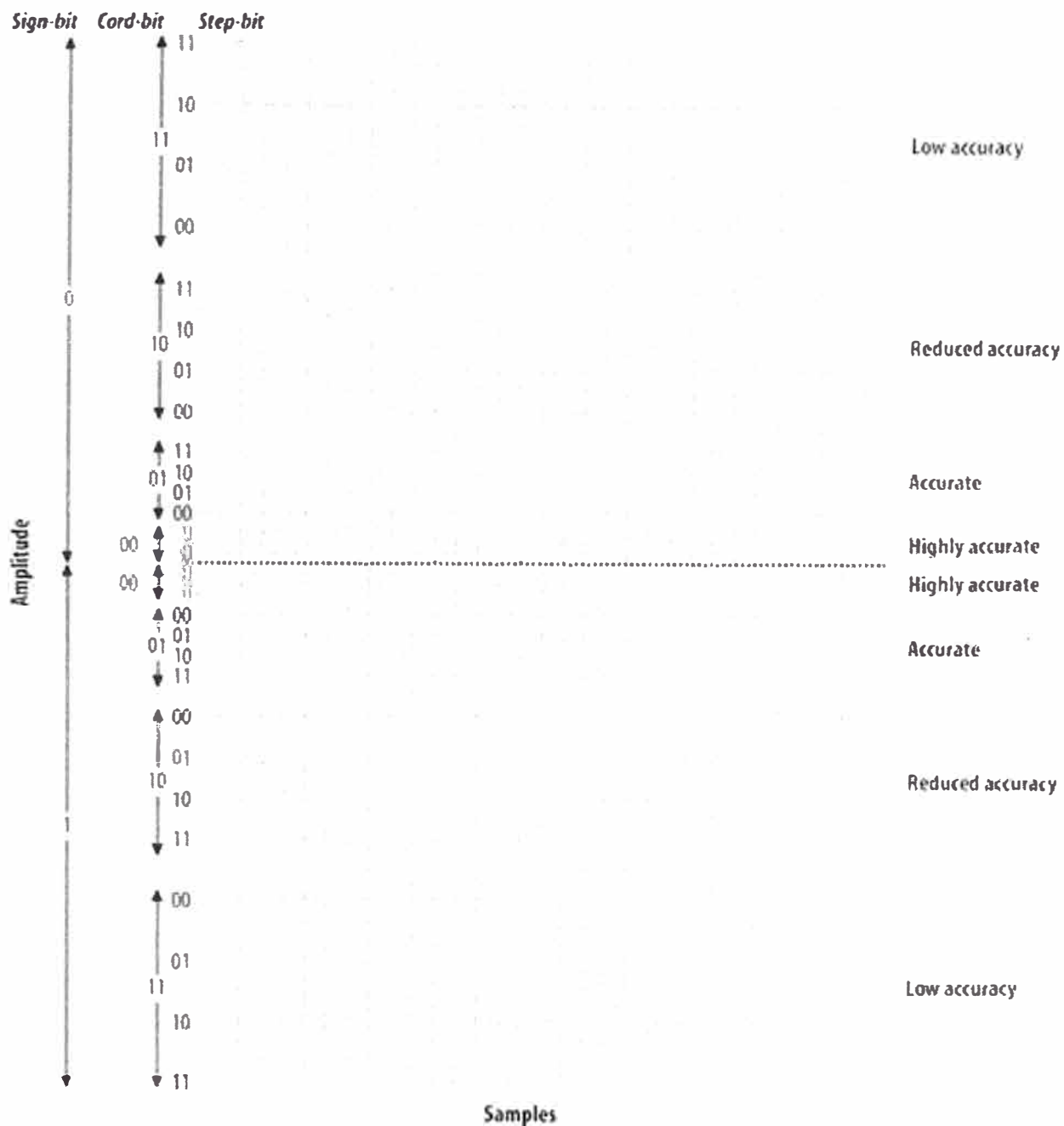


Figura No. 1.12. Companding de 5-bits.

La muestra cuantizada de la señal produciría la siguiente secuencia de bits:

```
00000 10011 10100 10101 01101 00001 00011 11010 00010 00001 01000 10011
10100 10100 00101 00100 00101 10101 10011 10001 00011 00001 00000 10100
10010 10101 01101 10100 00101 11010 00100 00000 01000
```

Figura No. 1.13. Señal digitalizada.

Y se vería de la siguiente manera:

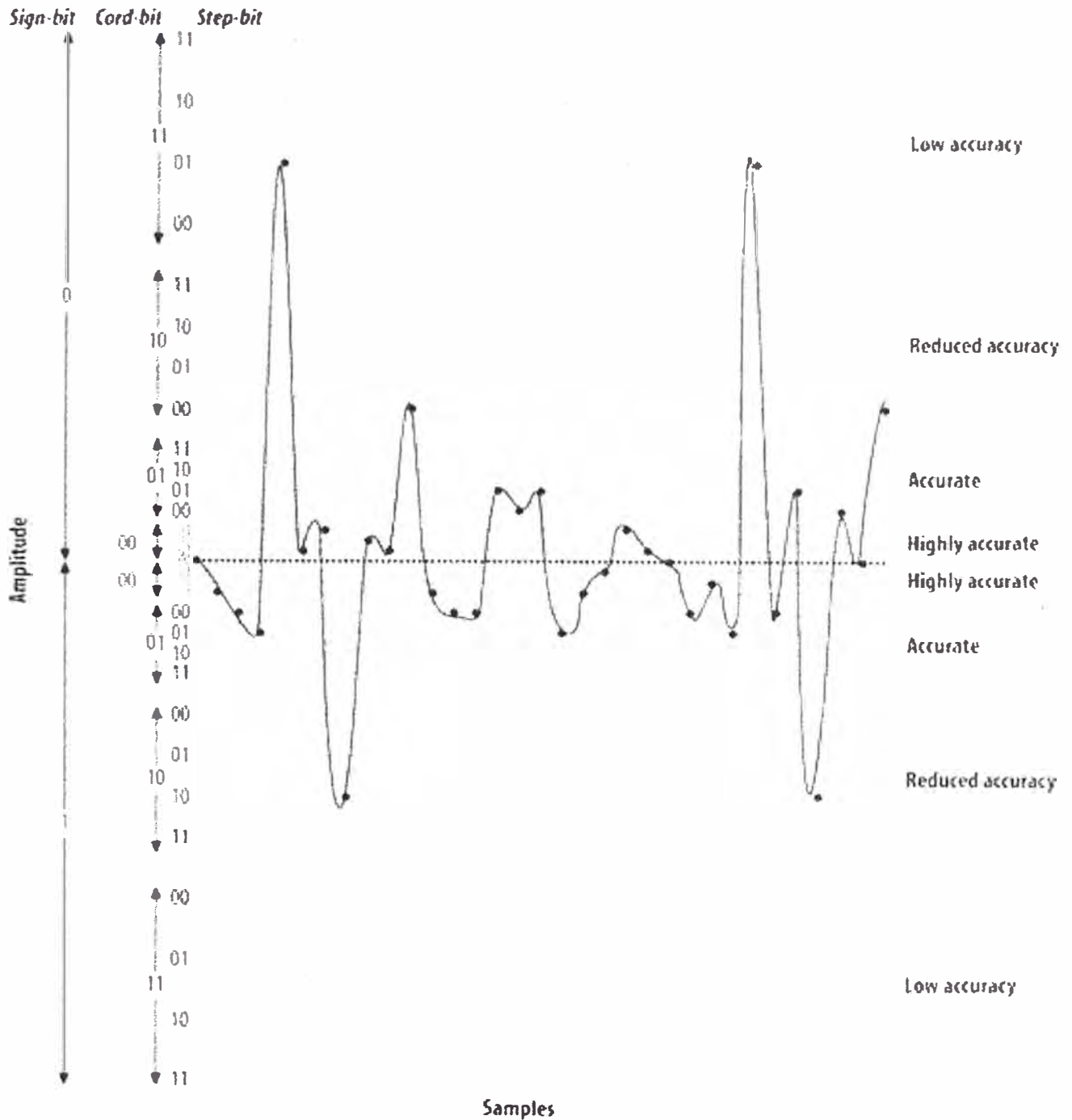


Figura No. 1.14. Señal cuantizada del Companding de 5-bits.

e) Aliasing

Si alguna vez se ha mirado las ruedas en un vagón girar hacia atrás en una vieja película del oeste, se ha visto los efectos del *aliasing*. La tasa de cuadros de la película no puede mantenerse con la frecuencia rotacional de las ruedas, y entonces una falsa rotación es percibida.

En sistemas de audio digital (del cual el moderno PSTN es uno de ellos), el *aliasing* ocurre si las frecuencias que son mayores que la mitad de la tasa de muestreo están presentes para el convertidor analógico/digital (A/D). En la red PSTN, esto es cualquier audio frecuencias por sobre los 4Khz (la mitad que la tasa de muestreo de 8Khz). Este problema es fácilmente corregido pasando el audio a través de un filtro pasa-bajo antes de presentarlo al convertidor A/D.

1.1.2.2 La Red Telefónica de Circuitos Digitales Conmutados

Por más de cien años, las redes telefónicas fueron exclusivamente de circuitos conmutados. Lo que esto quiere decir es que, para cada llamada telefónica hecha, una conexión dedicada era establecida entre los dos terminales, con una cantidad de ancho de banda fija, colocada en ese circuito.

Crear dicha red era costoso, y a juzgar por la distancia, usar la red era igualmente costoso también. Sin embargo, se está prediciendo el final de la red de circuitos conmutados. Muchas personas aún la usan hoy en día, y realmente funciona muy bien.

a) Tipos de Circuitos

En la red PSTN, hay muchos tamaños diferentes de circuitos sirviendo las varias necesidades de la red. Entra la oficina central y el subscritor, uno o más circuitos analógicos, o unas docenas de canales enviados sobre un circuito digital, generalmente suficiente. Entre las oficinas PSTN (y con muchos consumidores), los circuitos de fibra óptica son generalmente usados.

- **El DS-0, el fundamento de todo:** Desde que el método estándar para digitalizar una llamada telefónica es grabar una muestra de 8-bits 8000 veces por segundo, podemos ver que un circuito telefónico codificado en PCM necesitará un ancho de banda de 64Kpbs. El canal de 64Kpbs es referido como un DS-0. El DS-0 es el bloque fundamental de todos los circuitos digitales de telecomunicaciones.

Incluso los viejos circuitos análogos son muestreados a DS-0 tan pronto como sea posible. Algunas veces esto pasa donde están los terminales de la oficina central, y a veces mucho antes.

- **Circuitos de T-Carrier:** La venerable T-1 es uno de los más reconocidos términos de telefonía digital. Un T-1 es un circuito digital consistente de 24 DS-0s multiplexados en un flujo de bits de 1.544-Mbps. Este flujo de bits está propiamente definido como un DS-1. La voz es codificada en un T-1 usando el algoritmo u-law companding.

Los 24 DS-0 usan 1536Mbps y el remanente 0.008Mbps es usado por los cuadros de bits.

La versión europea del T-1 fue desarrollada por la Conferencia Europea de Administración Postal y de Telecomunicaciones (CEPT), y fue referida como un CEPT-1. Es ahora llamado E-1.

El E-1 está comprimido a 32 DS-0s, pero el método de codificación PCM es diferente. E-1 usa companding A-law.

Esto significa que la conexión entre una red basada en E-1 y una red basada en T-1 siempre requerirá un paso de transcodificación.

Un E-1, a pesar que tiene 32 canales, también considera un DS-1.

Tabla No. 1.2. Circuitos T-Carrier.

Carrier	Equivalent data bit-rate	Number of DS-0s	Data bit-rate
T-1	24 DS-0s	24	1.544 Mbps
T-2	4 T-1s	96	6.312 Mbps
T-3	7 T-2s	672	44.736 Mbps
T-4	6 T-3s	4032	274.176 Mbps

A densidades sobre T-3, es muy poco común ver un circuito de un T-carrier. Para esas velocidades, los circuitos de carrier óptico (OC) pueden ser usados.

- **Circuitos SONET y OC:** La red de sincronía óptica (SONET) fue desarrollada como un deseo de llevar el sistema de T-Carrier al siguiente nivel de tecnología:

Las fibras ópticas. SONET está basado en el ancho de banda de T-3 (44.736 Mbps), con una poca carga haciéndolo 51.84 Mbps.

Esto es referido como un OC-1 o STS-1. Como la siguiente tabla muestra, todas las altas velocidades de circuitos OC son múltiples de esta tasa base.

Tabla No. 1.3. Circuitos OC.

Carrier	Equivalent data bit-rate	Number of DS-0s	Data bit-rate
OC-1	1 DS-3 (plus overhead)	672	51.840 Mbps
OC-3	3 DS-3s	2016	155.520 Mbps
OC-12	12 DS-3s	8064	622.080 Mbps
OC-48	48 DS-3s	32256	2488.320 Mbps
OC-192	192 DS-3s	129024	9953.280 Mbps

SONET fue creada como un esfuerzo de estandarizar los circuitos ópticos, pero debido a su alto costo, se empareja con el valor ofrecido por muchos otros esquemas nuevos, como la Múltiplexación por División de Ondas Densas (DWDM), hay muchas controversias alrededor de su futuro.

b) Protocolos de Señalización Digital

Como con cualquier circuito, no es suficiente para los circuitos usados en el PSTN llevar solamente voz entre ambos lados de la línea. Los mecanismos deben también proveer pasar información acerca del estado del canal entre ambos puntos. La desconexión y supervisión de respuesta son dos ejemplos de señalización básica que puede necesitarse; el Caller ID es un ejemplo de una forma más compleja de señalización.

c) Señalización de canales asociados (CAS)

También conocido como señalización robet-bit, CAS es lo que se usa para transmitir voz en un T-1 cuando no está disponible RDSI. En lugar de tomar ventaja de la potencia del circuito digital, CAS simula canales analógicos. La

señalización CAS trabaja robando bits del flujo de audio para propósitos de señalización. A pesar que el efecto en la calidad de audio no es notable, la falta de un más potente canal de señalización limita la flexibilidad.

Cuando se configura un CAS T-1, las opciones de señalización en cada lado pueden coincidir. La señal E&M (Ear & Mouth o recEive & transMit) es preferida generalmente, porque ofrece mejor supervisión.

CAS es usado muy raramente en circuitos PSTN, debido a la superioridad de RDSI-BRI. Una de las limitaciones de CAS es que no permite la asignación dinámica de canales a diferentes funciones. También la información de Caller ID (la cual no está ni siquiera soportada) tiene que ser enviada como parte del flujo de audio. CAS es comúnmente usado en el enlace T-1 en banco de canales, a pesar que PRI está a veces disponible (y preferible).

d) RDSI

La Red Digital de Servicios Integrados (RDSI) ha estado presente por alrededor de 20 años. Debido a que separa los canales que llevan tráfico (los canales portadores, o canales-P) de los canales que llevan la información de señalización (canales-D), RDSI permite la entrega de características mas diversas que CAS. En el principio, RDSI prometía entregar la misma funcionalidad que la Internet nos brinda ahora, incluyendo capacidades avanzadas de voz, video y transferencia de datos.

Desafortunadamente, en lugar de ratificar un estándar y plegarse a éste, las distintas empresas de manufactura de dispositivos de telecomunicaciones decidieron añadir sus propias personalizaciones al protocolo, en la creencia que sus versiones serían superiores y eventualmente dominarían el mercado. Como resultado, tenemos dos sistemas RDSI-compatibles y para conectarse uno con el otro, era una tarea frecuente y costosa. Los carriers que tenían que adoptar esa costosa tecnología luego cobraban caro por ella, que no fue rápidamente adoptada. Actualmente RDSI es raramente usado por mucho más que entroncamiento básico.

e) RDSI-BRI/BRA

La interfaz de tasa básica (BRI – Basic Rate Interface) es una forma de RDSI, diseñada para servir pequeños puntos finales como estaciones de trabajo.

La especificación de esta forma de RDSI, llamada BRI es también referida simplemente como “RDSI”, pero esto puede ser una fuente de confusión, porque el RDSI es un protocolo, no un tipo de circuito (sin mencionar que los circuitos PRI son también correctamente referidos como RDSI!).

Un circuito de tasa básica RDSI consiste de dos canales B de 64Kbps controlados por un canal D de 16Kbps, por un total de 144 Kbps.

El RDSI de tasa básica ha sido una fuente de mucha confusión durante su existencia, debido a problemas con estándares de compatibilidad, complejidad técnica, y pobre documentación. Aún, en países europeos, los circuitos RDSI-BRI permanecen como una manera muy popular de conectarse al PSTN.

g) ISDN-PRI/PRA

La interfaz de tasa primaria (PRI - Primary Rate Access) es una forma de RDSI usada para proveer el servicio RDSI a conexiones de red muy largas. Un circuito RDSI de tasa primaria usa un solo canal DS-0 como enlace de señalización (El canal D); el resto de canales sirve como canales B.

En Norte América, el RDSI-PRI es comúnmente llevado en uno o más circuitos T-1. Ya que un T-1 tiene 24 canales, un circuito PRI norteamericano típicamente consiste de 23 canales B y un canal D. Por esta razón, PRI es frecuentemente referido como 23B+D.

En Europa, un circuito E-1 de 32 canales es usado, entonces un RDSI-PRI es referido como un 30B+D (El canal final es usado para sincronización).

El RDSI de tasa primaria es muy popular, debido a sus beneficios técnicos y generalmente a su precio competitivo. Si se requiere una docena de líneas PSTN, se debería de mirar primero a la alternativa de precio de un RDSI-PRI.

Desde una perspectiva técnica, RDSI-PRI es siempre preferible a CAS.

h) Señalización de Sistema 7

SS7 es un sistema de señalización usado por los proveedores. Es conceptualmente similar al RDSI y es instrumental en proveer mecanismos para los proveedores de transmitir información adicional que los puntos finales RDSI necesitan pasar. Sin embargo, la tecnología SS7 es diferente de aquella de RDSI. Ninguna diferencia es que SS7 corre en una red completamente separada de las actuales troncales que llevan las llamadas.

Soporte para SS7 en Asterisk está en el horizonte, ya que hay mucho interés en hacer a Asterisk compatible con los proveedores de red. Una versión de fuente abierta de SS7 existe (<http://www.openss7.org>), pero todavía se necesita mucho trabajo para la total compatibilidad SS7.

1.1.3 Arquitectura de una Central Telefónica.

Todos tenemos un teléfono en nuestra casa. Y, en general, sabemos que el cable del teléfono tiene una ficha (RJ-11) parecida a la del cable de red, y que adentro tiene dos cables de cobre, al que se denomina **par telefónico**. Ese par telefónico es el que va hasta la central telefónica, a una placa que se la suele denominar **placa de abonado**. Es la placa que controla nuestra línea.

En realidad, puede controlar muchas líneas, no una sola, y tiene una densidad de puertos que depende del fabricante, ronda entre los 8 y 16 abonados (a veces más, a veces menos). El valor exacto depende del equipo en particular.

La central telefónica es un conjunto de equipos relacionados. Todo este conjunto forma un equipo muy grande que puede llegar a ocupar varias habitaciones.

Como mencionamos, las centrales telefónicas suelen estar diseñadas para tener una muy alta disponibilidad (se suele decir que son *carrier class*, dado que se dice están disponibles el 99,999% del tiempo, que representa alrededor de 5 minutos al año de interrupción de servicio). Para lograr este objetivo, cuentan con redundancia en múltiples niveles (procesadores, enlaces, etc.); y en general se conectan a un sistema de energía ininterrumpida, que tiene un buen número de baterías que se conectan a un grupo electrógeno que se activa cuando se corta la luz.

1.1.4 Procesamiento de llamadas.

Hasta en la central, la voz va en forma analógica. Actualmente ya no existen centrales analógicas, todo lo que hay desde que llega la señal a la central y sale de la otra central hacia el otro abonado, es digital.

La placa de abonado es la que se encarga de hacer la conversión de una señal analógica a una digital y viceversa. La señal se convierte a un PCM de 64kbps, que es una señal digital **sin pérdida de información y sin compresión**, es el formato que se está utilizando desde prácticamente sus comienzos. También es la placa de abonado la que decodifica los tonos de discado (DTMF).

Es decir que, se utiliza el concepto de *señalización en banda*, comandar a la central utilizando la misma *banda* por la que se habla.

1.1.5 Conexión entre centrales.

La llamada que sale de nuestra central tiene que llegar hasta la central donde está la persona con la que queremos hablar. No hay doscientos millones de cables entre una y otra, sino que hay un enlace, el cual puede ser de diversos tipos. Este enlace se debe *multiplexar* para que todos los abonados de la central puedan hablar por teléfono.

Esta multiplexación es la que hace una diferencia a la hora de la calidad del servicio para el usuario. El sistema de multiplexación que utilizan las centrales telefónicas se llama **TDM: Time Division Multiplex**. Consiste en dividir el *stream* de datos en partes iguales de 64k (llamadas **time-slots**), de manera que los datos correspondientes al primer abonado van en el primer time-slot, los correspondientes al segundo en el segundo, y así.

Suponiendo un enlace de 2 Mbps de ancho de banda, como se transmiten 64k, podría haber hasta 32 abonados hablando a la vez. Con esta multiplexación en tiempo se separan y luego vuelven a unir los *streams* de voz que van de una central a otra, de manera transparente para el que lo está utilizando.

La ventaja de esta tecnología es que como se divide por un tiempo fijo, se puede *garantizar* el time-slot y saber que siempre lo que corresponde al primer abonado va en el primer time-slot y así sucesivamente para los siguientes abonados. Una vez establecida la comunicación, sea llamada local o internacional, está garantizado el ancho de banda necesario para poder hablar sin interrupciones.

Esto, en particular, es muy opuesto a lo que es IP, o cualquier enlace de paquetes en los que pueda haber colisiones, se pierdan paquetes, etc. Ya que en esos enlaces es muy difícil garantizar que la calidad inicial se mantenga a lo largo de toda la conversación, puede pasar que haya paquetes que lleguen antes que otros, que se sature la conexión y muchos otros factores que afectan a la calidad final del audio.

En definitiva, TDM es una de las diferencias esenciales entre la telefonía común y la de voz sobre IP, permite tener una red predictiva y garantizar calidad.

1.1.6 Ruteo, Señalización y Protocolos

Otro tema importante es el "ruteo" entre centrales, es decir, cómo sabe la central del abonado con que central se tiene que conectar.

Vamos a denominar **señalización** a la información relacionada con una llamada que se transmite entre dos equipos (la definición en sí es mas amplia, pero esto es en particular lo más relevante para el caso). Podemos dividirla en dos grupos: la que refiere al abonado y las llamadas en sí (levantó, marcó, cortó), y otra parte entre las centrales (que se le caiga algo y le quiera avisar, etc.).

A través de la señalización, la central puede ubicar a qué otra central tiene que llamar, a qué abonado dentro de esa central hay que llamar, saber que se cortó la comunicación, que dio ocupado, etc.

Las centrales entre sí se comunican utilizando diversos protocolos, los cuales generalmente son estándares públicos, aunque en muchos casos las especificaciones no son fáciles (o baratas) de conseguir. Los protocolos más comunes son tres: R2, PRI y SS7.

R2 es uno de los más viejos y tiene muchas variantes distintas, hay -incluso- una variante argentina, y pasa toda su información utilizando 4 bits. SS7 es, por otra parte, uno de los más nuevos y complejos.

Se necesita que las dos centrales que se están queriendo comunicar puedan *hablar* un mismo protocolo, de manera que si se quieren intercomunicar dos centrales que no soportan los mismos protocolos, es necesario que utilicen una central intermedia que *traduzca* la información.

Acerca del enlace por el cual se pasa tanto la señalización como la voz en sí, existen muchísimos tipos. Los más conocidos y comunes son E1 o E3 (europeos), con sus variantes T1 o T3 (utilizadas principalmente en los Estados Unidos). Son cables de cobre, muy parecidos al cable coaxial, que pueden ser de 75 o 120 ohms. El E1 tiene 2Mbps (32 canales de 64kbps), el E3 tiene 32Mbps (512 canales de 64kbps).

En muchos ámbitos cuando se habla de este tipo de enlaces se le da importancia solo al ancho de banda; sin embargo en nuestro caso también nos interesa el número de timeslots en el cual se puede dividir.

Sin embargo, no se pueden ocupar todos los canales para pasar todos los abonados. Es necesario poder avisar que hay llamadas y ese tipo de información. Por ejemplo, en el caso de una E1 se suelen utilizar 30 canales para el paso de la voz, 1 para *framing* (el 0) y 1 para señalización (el 15). En el de framing se suele encontrar (entre otras cosas) el CRC de los otros 31 (aunque depende de la configuración), de manera que si un determinado frame está corrupto, se lo puede notar y actuar en consecuencia.

1.1.7 Utilización de los protocolos.

Si se quiere que una computadora se comunique directamente con una central telefónica, es necesario hacer que la máquina *hable* el protocolo correspondiente, utilizando TDM y demás.

Un ejemplo muy común de este tipo de equipos son los denominados IVR, que son esos aparatos contestadores que nos dicen *marque uno para hablar ventas, dos para soporte técnico...*; y suelen hablar o R2 o PRI, según el caso.

1.2 Voz sobre IP

Hace 30 años Internet no existía, y las comunicaciones se realizaban por medio del teléfono a través de la red telefónica pública conmutada (PSTN), pero con el pasar de los años y el avance tecnológico han ido apareciendo nuevas tecnologías y aparatos bastante útiles que nos han permitido pensar en nuevas tecnologías de comunicación: PCS, teléfonos celulares y finalmente la popularización de la gran red Internet. Hoy por hoy podemos ver una gran revolución en comunicaciones: todas las personas usan los computadores e Internet en el trabajo y en el tiempo libre para comunicarse con otras personas, para intercambiar datos y a veces para hablar con mas personas usando

aplicaciones como NetMeeting o teléfono IP (Internet Phone), el cual particularmente comenzó a difundir en el mundo la idea que en el futuro se podría utilizar una comunicación en tiempo real por medio del PC: VoIP (Voice Over Internet Protocol).

Después de haber constatado que desde un PC con elementos multimedia, es posible realizar llamadas telefónicas a través de Internet, se podría pensar que la telefonía en IP es algo más que un juguete, pues la calidad de voz que se obtiene a través de Internet es muy pobre.

No obstante, si en una empresa se dispone de una red de datos que tenga un ancho de banda bastante grande, también se podría pensar en la utilización de esta red para el tráfico de voz entre las distintas delegaciones de la empresa. Las ventajas que se obtendrían al utilizar la red para transmitir tanto la voz como los datos son evidentes, ahorro de costos de comunicaciones, pues las llamadas entre las distintas delegaciones de la empresa saldrían gratis.

Realmente la integración de la voz y los datos en una misma red es una idea antigua, pues desde hace tiempo han surgido soluciones desde distintos fabricantes que, mediante el uso de multiplexores, permiten utilizar las redes WAN de datos de las empresas (típicamente conexiones punto a punto y Frame-Relay) para la transmisión del tráfico de voz, además es importante resaltar que el paquete de voz es indistinguible del paquete de datos, y por lo tanto puede ser transportado a través de una red que estaría normalmente reservada para transmisión de datos, donde los costos son frecuentemente más bajos.

Es innegable la implantación definitiva del protocolo IP desde los ámbitos empresariales a los domésticos y la aparición de un estándar, el VoIP, no podía hacerse esperar. La aparición del VoIP junto con el abaratamiento de los DSP's (procesador digital de señal), los cuales son claves en la compresión y descompresión de la voz, son los elementos que han hecho posible el despegue de estas tecnologías.

Para este auge existen otros factores, tales como la aparición de nuevas aplicaciones o la apuesta definitiva por VoIP de fabricantes como Cisco Systems o Nortel-bay Networks. Por otro lado los operadores de telefonía están ofreciendo o piensan ofrecer en un futuro cercano, servicios IP de calidad a las empresas.

1.2.1 Definición de Voz sobre IP

VoIP viene de Voice Over Internet Protocol. Como dice el término VoIP intenta permitir que la voz viaje en paquetes IP y obviamente a través de Internet.

La telefonía IP conjuga dos mundos históricamente separados: la transmisión de voz y la de datos. Se trata de transportar la voz, previamente convertida a datos, entre dos puntos distantes. Esto posibilitaría utilizar las redes de datos para efectuar las llamadas telefónicas, y yendo un poco más allá, desarrollar una única red convergente que se encargue de cursar todo tipo de comunicación, ya sea voz, datos, video o cualquier tipo de información.

Principio de Conmutación de Paquetes

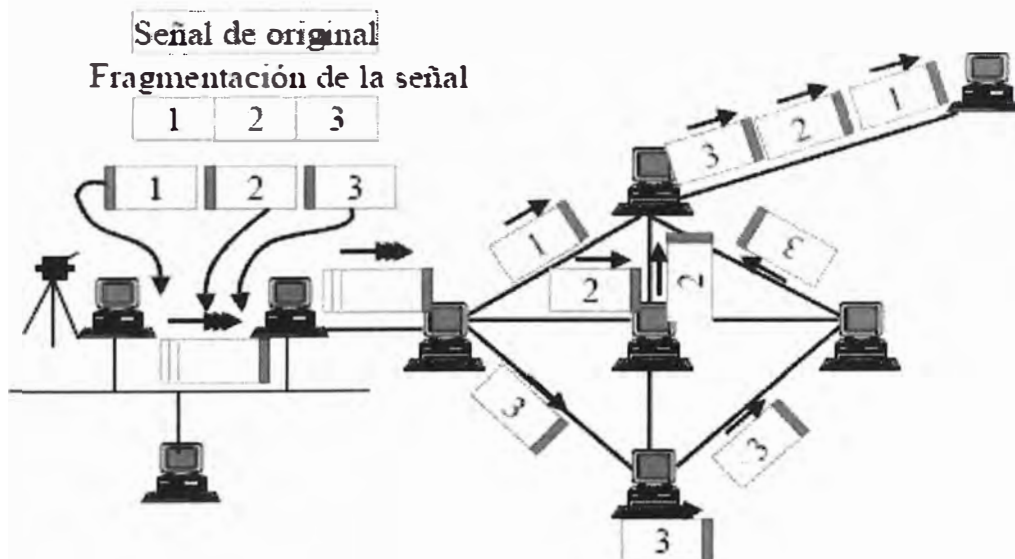


Figura No. 1.15. Conmutación de Paquetes.

La voz IP, por lo tanto, no es en sí mismo un servicio, sino una tecnología que permite encapsular la voz en paquetes para poder ser transportados sobre redes de datos sin necesidad de disponer de los circuitos conmutados convencionales PSTN (4), las redes desarrolladas a lo largo de los años para transmitir las conversaciones vocales, se basaban en el concepto de conmutación de circuitos, o sea, la realización de una comunicación que requiere el establecimiento de un circuito físico durante el tiempo que dura ésta, lo que significa que los recursos que intervienen en la realización de una llamada no pueden ser utilizados en otra hasta que la primera no finalice, incluso durante los silencios que se suceden dentro de una conversación típica.

En cambio, la telefonía IP no utiliza circuitos para la conversación, sino que envía múltiples de ellas (conversaciones) a través del mismo canal codificadas en paquetes y flujos independientes. Cuando se produce un silencio en una conversación, los paquetes de datos de otras conversaciones pueden ser transmitidos por la red, lo que implica un uso más eficiente de la misma.

Según esto son evidentes las ventajas que proporciona el segundo tipo de red, ya que con la misma infraestructura podrían prestar mas servicios y además la calidad de servicio y la velocidad serian mayores; pero por otro lado también existe la gran desventaja de la seguridad, ya que no es posible determinar la duración del paquete dentro de la red hasta que este llegue a su destino y además existe la posibilidad de pérdida de paquetes, ya que el protocolo IP no cuenta con esta herramienta.

1.2.2 Funcionamiento de Voz sobre IP

Años atrás se descubrió que enviar una señal a un destino remoto también podía hacerse de manera digital: antes de enviar la señal se debía digitalizar con un ADC (analog to digital converter), transmitirla y en el extremo de destino transformarla de nuevo a formato análogo con un DAC (digital to analog converter).

VoIP funciona de esa manera, digitalizando la voz en paquetes de datos, enviándola a través de la red y reconvirtiéndola a voz en el destino. Básicamente el proceso comienza con la señal análoga del teléfono que es digitalizada en señales PCM (pulse code modulación) por medio del codificador/decodificador de voz (codec).

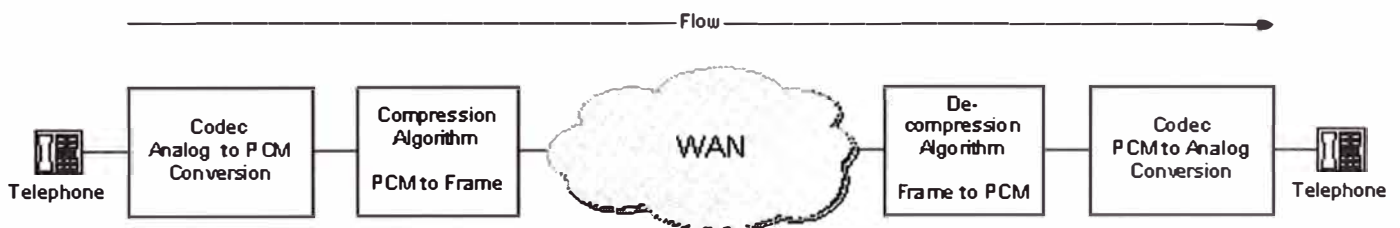


Figura No. 1.16. Flujo de un circuito de voz comprimido.

Las muestras PCM son pasadas al algoritmo de compresión, el cual comprime la voz y la fracciona en paquetes que pueden ser transmitidos para este caso a través de una red

privada WAN. En el otro extremo de la nube se realizan exactamente las mismas funciones en un orden inverso. El flujo de un circuito de voz comprimido es el mostrado en la Figura 1.16.

Dependiendo de la forma en la que la red esté configurada, el enrutador o el gateway pueden realizar la labor de codificación, decodificación y/o compresión.

Por ejemplo, si el sistema usado es un sistema análogo de voz, entonces el enrutador o el gateway realizan todas las funciones mencionadas anteriormente de la siguiente manera:

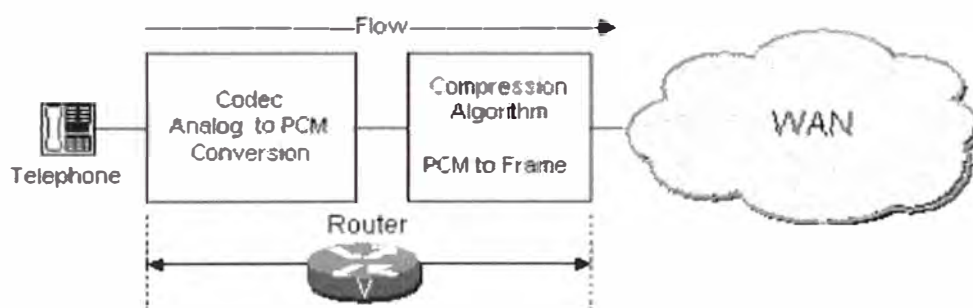


Figura No. 1.17. Enrutador realizando las funciones de codec.

Si, por otro lado, el dispositivo utilizado es un PBX digital, es entonces éste el que realiza la función de codificación y decodificación, y el enrutador solo se dedica a procesar las muestras PCM que le ha enviado el PBX.

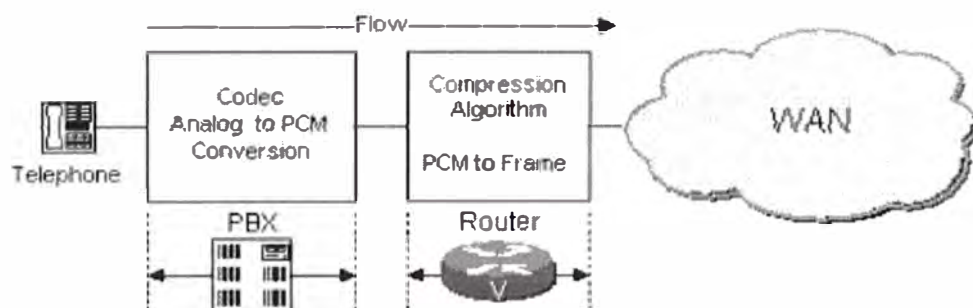


Figura No. 1.18. Ruteador procesando muestras PCM enviadas por un PBX.

Para el caso en el que el transporte de voz se realiza sobre la red pública Internet, se necesita una interfaz entre la red telefónica y la red IP, el cual se denomina gateway y es

el encargado en el lado del emisor de convertir la señal analógica de voz en paquetes comprimidos IP para ser transportados a través de la red, del lado del receptor su labor es inversa, dado que descomprime los paquetes IP que recibe de la red de datos, y recompone el mensaje a su forma análoga original conduciéndolo de nuevo a la red telefónica convencional en el sector de la última milla para ser transportado al destinatario final y ser reproducido por el parlante del receptor.

Es importante tener en cuenta también que todas las redes deben tener de alguna forma las características de direccionamiento, enrutamiento y señalización. El direccionamiento es requerido para identificar el origen y destino de las llamadas, también es usado para asociar clases de servicio a cada una de las llamadas dependiendo de la prioridad.

El enrutamiento por su parte encuentra el mejor camino a seguir por el paquete desde la fuente hasta el destino y transporta la información a través de la red de la manera más eficiente, la cual ha sido determinada por el diseñador. La señalización alerta las estaciones terminales y a los elementos de la red su estado y la responsabilidad inmediata que tienen al establecer una conexión.

Para telefonía IP hay muchos protocolos. Los vamos a separar en 3 partes: codificación de la voz, transmisión de la voz y señalización.

a) Codificación de la voz.

La transmisión ya no se va a hacer en PCM (protocolo G.711), como en la telefonía tradicional. La voz se puede comprimir aun más: si una persona se queda callada, por ejemplo, no es necesario transmitir el sonido completo del silencio. Hay muchos *codecs* de compresión. Como todo codec, cuanto más se comprime, más procesador se necesita. Hay codecs con pérdida que comprimen de 64k a 4k, incluso hasta 3.1k. Hay algunos que no consideran pérdida, pero la mayoría son con pérdida.

Hay muchos estudios al respecto, ya que lo más importante es la percepción que tiene la gente de lo que se escucha, y es muy difícil medir la percepción humana. Para la realización de estos estudios, se comprime el audio y se pide a grupos de personas que lo escuchen y que manifiesten si les parece que es de buena calidad o no, se les asignan puntajes, etc.

En general se elige un balance entre compresión y percepción. Hay muchos balances distintos. Hay muchos codecs que están patentados, para los que hay que pagar las licencias de uso (no la implementación, sino el uso en sí).

Un ejemplo de un buen codec es el GSM, utilizado en los teléfonos celulares. Es un codec libre, que se escucha bastante bien, comprime bastante bien, y consume muy poco procesador. Que consuma poco procesador es importante cuando se está trabajando a gran escala (200, 1000 líneas). En el caso de los celulares, la voz se comprime en el mismo aparato celular y se transmite ya comprimida. Para este protocolo, en GNU/Linux existe la libgsm que es una biblioteca de pequeño tamaño.

b) Envío de la voz.

Para el envío de la voz se utiliza UDP, dado que las características de TCP son muy contraproducentes: al ser un tráfico *real time*, nos interesa que si hay algún retraso, desorden o pérdida de paquetes, la información no sea retransmitida.

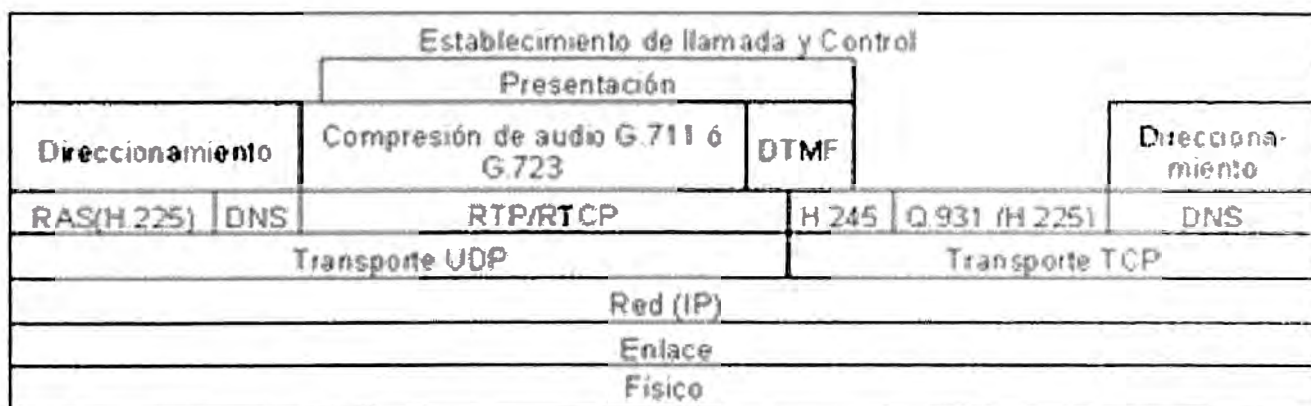


Figura No. 1.19. Corte transversal de las Capas OSI con los protocolos de VoIP.

Aunque UDP no ofrece integridad en los datos, el aprovechamiento del ancho de banda es mayor que con TCP. UDP provee a los usuarios acceso a los servicios IP. Los paquetes UDP son entregados como paquetes IP no orientados a conexión, los cuales pueden ser descartados antes de alcanzar su objetivo.

c) Señalización.

Tal como vimos anteriormente, es necesario tener un protocolo para poder indicar a qué máquina se quiere llamar y demás. Existen actualmente varios protocolos para señalización.

Uno que está cayendo muy en desuso es el H323, que no es fácil de programar, y tiene restricciones cuando trabaja sobre redes con NAT; pero es muy importante porque fue el primero con el que se empezó a usar en VO/IP de forma masiva. Actualmente se está dejando de usar, y probablemente en el futuro no se use más. Los programas NetMeeting, y su equivalente libre GnomeMeeting utilizan este protocolo.

El protocolo que más se está usando actualmente es SIP: *Session Initiation Protocol*. Se trata de un protocolo que tiene una característica muy particular: está estandarizado por la IETF (*Internet Engineering Task Force*, es decir, los que hacen las RFCs) y, en consecuencia, es muy abierto y de fácil acceso.

SIP es un protocolo de texto plano que se utiliza sobre TCP, ya que en el caso de la señalización es importante que no se pierda la información. Tiene una arquitectura que está muy bien pensada, no trata de meter todo el mundo telefónico en IP, ni todo IP en el mundo telefónico. Sin embargo, también tiene problemas para atravesar NAT.

Normalmente, cuando se usa SIP, el protocolo que se utiliza para enviar la voz es RTP (*Real Time Protocol*), que se usa sobre UDP.

Entre las aplicaciones que usan el protocolo sobre Linux, encontramos el programa linphone, que es un cliente SIP.

1.2.3 Ejemplo de conexión VoIP usando SIP.

A modo de ejemplo, vamos a considerar dos PCs, que están conectadas a través de Internet. Juan, que está conectado desde una PC, quiere hablar con María, que está conectada desde otra. A María le llega un *invite* que le indica que Juan quiere hablar con ella (equivalente a un RING), y si acepta la comunicación (equivalente a levantar el teléfono), puede hablar con Juan.

La conexión se establece usando SIP sobre TCP y luego la transmisión se hace usando RTP sobre UDP. Cuando se termina la conversación, se transmite la terminación de la conexión por SIP. Esto permite que dos usuarios de PC puedan *hablar por teléfono*, sin tener una central telefónica en el medio, utilizando la estructura IP existente para establecer una comunicación.

Durante la inicialización se pasan las IPs y los puertos a utilizar y por eso es que es difícil hacerlo a través de una red NAT.

1.2.4 Conexión de muchas computadoras usando VoIP

Si en lugar de 2 PCs, se quiere conectar un número importante de computadoras, que quieren hablar entre sí sin tener que estar transmitiéndose los números IP, y la entidad que les está proveyendo el servicio quiere poder tener un registro de las comunicaciones establecidas, se utiliza un **Server SIP** (que vendría a ser el equivalente a un *Gatekeeper* en H323). También se lo suele llamar Proxy SIP o Router SIP, que si bien teóricamente cumplen funciones específicas, en general se utilizan los términos de manera indistinta.

Teniendo un servidor, cuando Juan quiere hablar por teléfono, le envía una señal al server indicándole que quiere hablar con María, y este le avisa a María que Juan quiere hablar con ella. A partir de que se acepta la comunicación, se pasan algunos mensajes más a través del server (utilizando SIP) para negociar IPs, puertos, protocolo de compresión a utilizar, etc. Pero una vez que comienza la comunicación, el canal UDP ya no pasa por el server. Una vez terminada la conversación, se utiliza SIP para avisar que se terminó la conversación.

Esta es una de las mejores cosas que tiene la telefonía IP, porque por un lado separa la señalización de la transmisión de voz, y por el otro lado la transmisión se hace *peer to peer*. Pero el servidor debe confiar en la buena fe de los clientes para saber cuándo una comunicación se terminó realmente.

Un cliente que tenga DHCP tiene que avisarle al server en qué IP está, para poder autenticarse contra él, utilizando un nombre de usuario y una clave. De manera que el server puede saber que un determinado usuario *no está* y poner un contestador, dar ocupado, etc.

Con este principio se puede hacer que un teléfono VO/IP se enchufe en cualquier lugar del mundo donde haya banda ancha y siempre sigue siendo el mismo teléfono. Y de

hecho este servicio existe y se vende: se proporciona una línea en Lima, se lleva el teléfono VO/IP a cualquier lugar del mundo, se enchufa a un ADSL y se puede hablar o ser llamado como si se estuviera en Lima.

De la misma manera que con las centrales telefónicas, puede haber varios servers que se comuniquen entre sí, y solamente van a intercambiar la parte correspondiente al protocolo SIP, la parte de RDP/UDP se hace directo entre los dos puntos que se están comunicando.

La implementación de referencia del server SIP en fuente abierta se llama *ser*.

Por otro lado, uno puede querer hablar desde una computadora a teléfonos comunes, para esto se necesita un *gateway* (gw) que haga la conversión de una tecnología a otra.

1.2.5 Ventajas y Desventajas de VoIP.

Ventajas:

- Como la información está comprimida, se puede pasar más volumen en el mismo ancho de banda.
- El hardware es más sencillo, no se necesita tener conocimientos tan específicos como los que se necesitan para telefonía.
- Permite ahorrar dinero, es más barato. Todo lo que se habla utilizando VoIP es plata que no se gastó en hablar por teléfono.

Desventajas:

Trabajar con IP hace que uno no pueda contar con determinados anchos de banda, que haya colisiones de paquetes, etc.

Algo importante de GNU/Linux es el control de tráfico (tc), ya que mediante tc podemos hacer un control de ancho de banda a un nivel mucho más alto de lo que permiten algunos routers.

Pero más allá de eso, controlar tráfico masivo a nivel IP no es nada sencillo, hay muchas variables muy complejas, y es muy difícil de hacer a gran escala. Es muy fácil meter la pata y que salga mal. Además, si se está utilizando un enlace sobre el cual no se tiene control (Internet siendo un claro ejemplo) ya no se puede garantizar nada.

El problema de VoIP, entonces, no es el protocolo, sino la red IP. La red IP no está pensada para dar ningún tipo de garantías, es una red *mejor esfuerzo*. El peor problema no es solo el ancho de banda (ya que con buena compresión, se puede transmitir hasta por modem), sino la latencia. No se puede utilizar VO/IP con 200ms de latencia. Cuando el usuario está hablando y el otro lo está escuchando, no es aceptable tener 200ms de pausa. Y, como ya se explicó, las centrales telefónicas, con la tecnología TDM evitan este tipo de problemas.

Tabla No. 1.4. Conmutación de Circuitos Vs. Conmutación de Paquetes.

Conmutación de Circuitos-versus-Paquetes

Características	Conmutación de Circuito	Conmutación de Paquetes
Tiempo de establecimiento	<ul style="list-style-type: none"> •Aceptable para voz •Muy largo para datos 	<ul style="list-style-type: none"> •No existe fase de establecimiento.
Retardo de transmisión	<ul style="list-style-type: none"> •Despreciable 	<ul style="list-style-type: none"> •Existe en toda comunicación •Orden de mseg.
Asignación de circuitos	<ul style="list-style-type: none"> •Único y exclusivo para cada comunicación 	<ul style="list-style-type: none"> •Compartido por otras comunicaciones simultáneas.
Identificación del destino	<ul style="list-style-type: none"> •Sólo en la fase de establecimiento 	<ul style="list-style-type: none"> •Se incluye un identificador en cada paquete.
Necesidad de almacenar en la red	<ul style="list-style-type: none"> •No 	<ul style="list-style-type: none"> •Sí, en los nodos de la red.
Flexibilidad de la red	<ul style="list-style-type: none"> •Encaminamiento alternativo 	<ul style="list-style-type: none"> •Gran flexibilidad.

La única solución es que cuando se utiliza VoIP, hay que controlar la red sí o sí. Si se tiene un enlace dedicado o una LAN interna, por ejemplo, es posible controlar el uso de la red y garantizar la calidad requerida.

1.2.6 Implementaciones.

Cuando se habla de una implementación a nivel telefonía real se tiene que ir por un enlace controlado. Si se tiene un enlace de fibra de Lima a Arequipa, es posible pasar muchos más abonados por el mismo enlace E1 por el que se pasaban 30; pero es necesario poner controles en ambas puntas. Hay que tener mucha *inteligencia* en los equipos de control.

Una posibilidad para tener una red de VO/IP interna, por ejemplo, es tener unos auriculares y un micrófono en cada estación (PC). Y por otro lado, tener un equipo (gw) que se encarga de hacer la conversión a la telefonía tradicional. Para ello, es necesario tener algún contacto con la telefonía tradicional, una E1, 4 líneas telefónicas, etc.

En el caso de la E1, hará falta tener el hardware que se encargue de hacer TDM, para las líneas telefónicas, es necesario tener placas especiales que se conectan a líneas telefónicas.

CAPITULO II

DESCRIPCIÓN Y VENTAJAS DE LOS SERVIDORES LINUX

Linux es principalmente una de las tantas opciones de sistemas Operativos diseñados especialmente para trabajar en red. Es una plataforma UNIX con un entorno más amigable, sin embargo, podríamos decir que a pesar de algunas diferencias es casi un perfecto clon del mismo.

Linux es un sistema multiusuario, multitarea, y trabaja en 32 Bits REALES a diferencia de otros sistemas como Microsoft Windows 95 / 98 que dice hacerlo pero que en realidad trabajan a 16 Bits.

Hay muchas cualidades que diferencian a Linux de otros Sistemas Operativos (SOs). Linux es reconocido por su gran estabilidad, la excelente administración de recursos que realiza el sistema y su open source (código abierto) porque el usuario tiene total libertad para modificar el código fuente y recompilarlo creando una versión del sistema operativo. El kernel o núcleo de Linux se actualiza todo el tiempo, tapando viejos bugs o agujeros y ampliando la compatibilidad con diferentes dispositivos.

Linux es la denominación de un sistema operativo y el nombre de un núcleo. Es uno de los paradigmas del desarrollo de software libre (y de código abierto), donde el código fuente está disponible públicamente y cualquier persona, con los conocimientos informáticos adecuados, puede libremente estudiarlo, usarlo, modificarlo y redistribuirlo.

El término Linux estrictamente se refiere al núcleo Linux, pero es más comúnmente utilizado para describir al sistema operativo tipo Unix (que implementa el estándar POSIX), que utiliza primordialmente filosofía y metodologías libres (también conocido como GNU/Linux) y que está formado mediante la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos/grupos de software (libre o no libre).

El núcleo no es parte oficial del proyecto GNU (el cual posee su propio núcleo en desarrollo, llamado Hurd), pero es distribuido bajo los términos de la licencia GNU GPL.

La expresión Linux también es utilizada para referirse a las distribuciones Linux, colecciones de software que suelen contener grandes cantidades de paquetes además del núcleo. El software que suelen incluir consta de una enorme variedad de aplicaciones, como: entornos gráficos, suites ofimáticas, servidores web, servidores de correo, servidores FTP, etcétera. Coloquialmente se aplica el término Linux a éstas, aunque en estricto rigor sea incorrecto, dado que la distribución es la forma más simple y popular para obtener un sistema Linux.

La marca Linux (Número de serie: 1916230) pertenece a Linus Torvalds y se define como "un sistema operativo para computadoras que facilita su uso y operación".

Desde su lanzamiento, Linux ha incrementado su popularidad en el mercado de servidores. Su gran flexibilidad ha permitido que sea utilizado en un rango muy amplio de sistemas de cómputo y arquitecturas: computadoras personales, supercomputadoras, dispositivos portátiles, etc.

Los sistemas Linux funcionan sobre más de 20 plataformas diferentes de hardware; entre ellas las más comunes son las de los sistemas compatibles con PCs x86 y x86-64, computadoras Macintosh, PowerPC, Sparc y MIPS.

Asimismo, existen Grupos de Usuarios de Linux en casi todas las áreas del planeta.

2.1 Ventajas de usar un Servidor Linux

- **Es más seguro**
 - Ya que la gran mayoría de los ataques de hackers son dirigidos a servidores Windows al igual que los virus los cuales se enfocan principalmente a servidores con este sistema operativo.
 - La plataforma Linux es más robusta lo cual hace más difícil que algún intruso pueda violar el sistema de seguridad de Linux.

- **Es más rápido**

- Al tener una plataforma más estable, esto favorece el desempeño de aplicaciones de todo tipo tales como: bases de datos, aplicaciones XML, multimedia, etc.
- La eficiencia de su código fuente hace que la velocidad de las aplicaciones Linux sean superiores a las que corren sobre Windows lo cual se traduce en velocidad de su página.

- **Es más económico**

- Ya que requieren menor mantenimiento. En servidores windows es más costoso debido a que es necesaria una frecuente atención y monitoreo contra ataques de virus, hackers y errores de código, instalación y actualización de parches y service packs.
- El software Linux así como también un sin número de aplicaciones son de código abierto (gratuitos).
- No requieren supervisión tan estrecha ni pagos de pólizas de mantenimiento necesarias para obtener los Service Packs.

¿Qué ventajas tiene Windows sobre Linux?

- **Es más fácil**

- Al ser de mayor facilidad de uso, Windows en este momento continúa siendo el sistema operativo más comercial lo cual se refleja en la disponibilidad de aplicaciones, facilidad de mantenimiento así como soporte en el desarrollo de nuevas aplicaciones, puntos que pueden ser cruciales en la elección de servidores que corren aplicaciones web.

- **Aplicaciones desarrolladas en menor tiempo**

- Fruto de la inversión realizada por Microsoft y aunado a una comunidad de programadores cada vez más grande se ha logrado facilitar el desarrollo de aplicaciones y sistemas que corran sobre servidores Windows lo cual se ve reflejado en tiempos de desarrollo menores.
- La curva de aprendizaje en el sistema Windows es mucho menor.

La creciente popularidad de Linux se debe a las ventajas que presenta ante otros tipos de software. Entre otras razones se debe a su estabilidad, al acceso a las fuentes (lo que permite personalizar el funcionamiento y auditar la seguridad y privacidad de los datos tratados), a la independencia de proveedor, a la seguridad, a la rapidez con que incorpora los nuevos adelantos tecnológicos (IPv6, microprocesadores de 64 bits), a la escalabilidad (se pueden crear clusters de cientos de ordenadores), a la activa comunidad de desarrollo que hay a su alrededor, a su interoperabilidad y a la abundancia de documentación relativa a los procedimientos.

2.2 Ventajas del uso de la tecnología de fuente abierta

Código abierto (del inglés open source) es el término por el que se conoce al software distribuido y desarrollado en forma libre. Este término empezó a utilizarse en 1998 por algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original en inglés del software libre (free software).

Free en inglés puede significar diferentes cosas: gratuidad y libertad. Por ello, por un lado, permite pensar en "software por el que no hay que pagar" (software gratuito) y, por otro, se adapta al significado que se pretendió originalmente (software que posee ciertas libertades).

El término para algunos no resultó apropiado como reemplazo para el ya tradicional free software, pues eliminaba la idea de libertad (incluso hay algunos que usan —en inglés— el término libre software para evitar la ambigüedad de free).

Desde el punto de vista de una "traducción estrictamente literal", el significado obvio de "código abierto" es que "se puede mirar el código fuente", por lo que puede ser interpretado como un término más débil y flexible que el del software libre. Basado en ello se argumenta que un programa de código abierto puede ser software libre, pero también puede ser semi-libre o incluso completamente no libre. Sin embargo, por lo general, un programa de código abierto puede ser y de hecho es software libre, como igualmente un programa Software Libre es Open Source. Esto ocurre dado que ambos movimientos reconocen el mismo conjunto de licencias y tienen principios equivalentes.

Un término que pretende resolver posibles ambigüedades o confusiones que ambos términos generan es FOSS (Free Open Source Software).

En la actualidad open source es utilizado para definir un movimiento nuevo de software (la Open Source Initiative), diferente al movimiento del Software Libre, incompatible con este último desde el punto de vista filosófico, y completamente equivalente desde el punto de vista práctico, de hecho, ambos movimientos trabajan juntos en el desarrollo práctico de proyectos.

La filosofía del Open Source centra su atención en la premisa de que al compartir el código, el programa resultante tiende a ser de calidad superior al software propietario, es una visión meramente técnica. Por otro lado, el Software Libre funciona bajo el ideal que el software propietario, al no poder compartirse es antiético, dado que prohibir compartir entre seres humanos va en contra de las leyes naturales.

El movimiento Open Source tiene un decálogo que debe cumplir un código para poder llamarse "Open Source" (es de hacer notar que estas 10 premisas son completamente equivalentes con las 4 libertades o principios del Software Libre), éstas son :

1. Libre redistribución: el software debe poder ser regalado o vendido libremente.
2. Código fuente: el código fuente debe estar incluido u obtenerse libremente.
3. Trabajos derivados: la redistribución de modificaciones debe estar permitida.
4. Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches.
5. Sin discriminación de personas o grupos: nadie puede dejarse fuera.
6. Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos.
7. Distribución de la licencia: deben aplicarse los mismos derechos a todo el que reciba el programa
8. La licencia no debe ser específica de un producto: el programa no puede licenciarse solo como parte de una distribución mayor.
9. La licencia no debe restringir otro software: la licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.
10. La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.

2.3 Software de fuente abierta aplicado a telefonía.

Una de las fronteras para el software de fuente abierta son las telecomunicaciones. Es en este campo donde muy pocas empresas se han atrevido a entrar agresivamente, debido a que el mercado de las telecomunicaciones siempre estuvo a cargo de grandes empresas, las cuales son muy difíciles de hacerles frente, ya que poseen un capital enorme y hacen grandes inversiones en tecnología.

Sin embargo, es justamente el nicho que se necesita para que proyectos pequeños de fuente abierta comiencen a ser usados en reemplazo de los tradicionales. Es así que tenemos varias soluciones presentes para software de telefonía desarrollados con tecnología de fuente abierta. Aunque es todavía un mercado joven y no son muchas las opciones de donde escoger, es posible encontrar software que esté a la par que las desarrolladas por grandes empresas. Entre ellos, una compañía llamada Asterisk está haciéndose un nombre por sí misma.

Podemos enumerar los siguientes proyectos de fuente abierta que están dentro de las aplicaciones que pueden competir profesionalmente con soluciones comerciales:

2.3.1 Asterisk

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

Asterisk tiene licencia GPL. Mark Spencer de Digium inicialmente creó Asterisk y actualmente es su principal desarrollador junto con otros programadores han contribuido a corregir errores, añadir novedades y funcionalidades. Originalmente desarrollado para el sistema operativo Linux, Asterisk actualmente también funciona en BSD, MacOSX, Solaris y Microsoft Windows aunque la plataforma nativa (Linux) es la mejor soportada de todos.

Asterisk incluye muchas características anteriormente solo disponibles en caros sistemas propietarios PBX: buzón de voz, conferencias, IVR, distribución automática de llamadas, y otras muchas más. Los usuarios pueden crear nuevas funcionalidades escribiendo un plan de llamadas o dialplan en el lenguaje de script de Asterisk o añadiendo módulos escritos en lenguaje C o en cualquier otro lenguaje de programación soportado por Linux.

Para conectar teléfonos normales analógicos hacen falta unas tarjetas telefónicas FXS o FXO fabricadas por Digium o por otros fabricantes, ya que para conectar el servidor a una línea externa no funciona con un simple módem.

Quizá lo más interesante de Asterisk es que soporta muchos protocolos VoIP como pueden ser SIP, H.323, IAX y MGCP. Asterisk puede interoperar con terminales IP actuando como un registrador y como gateway entre ambos.

Las compañías de telecomunicaciones de todo el mundo empiezan a utilizar Asterisk como sistema nativo de VoIP junto con SER Sip Express Router en lugar de otras marcas que ofrecen PBX propietarios como Alcatel, Cisco, Avaya ó Nortel.

2.3.2 OpenPBX

Es un paquete completo y funcional de PBX para oficina, con administración web completa. Integra el PBX y otras funcionalidades detrás de la oficina como el correo electrónico, web y servidor ftp en una sola PC para construir una solución completamente funcional de PBX/IT. La interfaz de usuario basado en web significa que el PBX se puede configurar fácilmente, comparado al otros PBXs que tiene interfaces difíciles de configurar basados en "touch tone". Por ejemplo tu web browser se puede utilizar para tener acceso a tu correo de voz, en lugar de presionar botones en el teléfono.

El OpenPBX es un proyecto de fuente abierta sobre Linux que ha hecho un "fork" del desarrollo actual de Asterisk. Es un proyecto prometedor que aún esta formándose por un grupo de desarrolladores muy talentosos. Estas son algunas de las razones por las cuales decidieron crear un clon de Asterisk y desviar el desarrollo enfocado a generar el mejor software PBX libre del mercado:

- Dar a la comunidad Internet de telefonía el control y las entradas de la aplicación. Ninguna persona individualmente puede detener el progreso.
- Poder usar lo mejor del código que ya existe, como apr, rtp stacks, sip stacks, etc.
- No reinventar la rueda si no es necesario.
- No tener ningún compromiso con licencias duales.
- No permitir que las decisiones de negocio de una empresa afecten el desarrollo de software.
- Cualquiera puede hacer enmiendas al código fuente, sin tener que hacer negados de copyright.

- No se puede competir con los clientes porque todos están en el mismo nivel.

Entre las diferencias más importantes entre OpenPBX y Asterisk, podemos mencionar:

- OpenPBX tiene un servidor STUN, codificado dentro del sistema, como soporte natural para SIP NAT transversal.
- OpenPBX tiene fax y T.38 fax sobre soporte IP.
- OpenPBX tiene jitterbuffer universal para uso con cualquier tipo de canal.
- OpenPBX usa timers POSIX, que significa que no hay dependencias con los timers de las tarjetas Zaptel.
- OpenPBX tiene un plan de llamadas más rápido y eficiente, porque usa hashing.
- OpenPBX tiene compatibilidad de plataforma cruzada y es más confiable.
- OpenPBX evalúa la corrección e integridad de los datos de configuración.
- OpenPBX se ejecuta sobre una maquina virtual como Xen o VMware.

A pesar de todas las ventajas que pudiera tener sobre Asterisk, este joven proyecto esta aún en modo de pruebas y aún no llega a la etapa de producción, sin embargo es bastante prometedor. Es un candidato a tenerlo a corta vista para ver cómo progresa su desarrollo, ya que parece ser el siguiente salto de tecnología luego de que Asterisk haya regido por los últimos dos años.

2.3.3 Logger

Este paquete es una aplicación de registros con licencia LGPL, excelente para aplicaciones de registro de voz, y grabación de llamadas. La aplicación fue diseñada para usarse con tarjetas Voicetronix OpenLog y OpenPri, ejecutándose sobre Linux.

Las aplicaciones típicas de registro de voz cuestan alrededor de US\$1000/canal. Usando este paquete se puede construir 4 canales de registro de voz por el precio de una PC y una tarjeta OpenLog4. Para soluciones de mayor densidad, se puede usar la tarjeta OpenPri que permitiría la construcción de una solución de registro de voz por un costo menor a US\$60/canal.

Este paquete contiene código hecho en C++/perl/html abierto, que contiene las siguientes características:

- Un servidor de registro de voz en C++, que es configurado vía una interfaz de usuario web.

- Graba las llamadas de usuarios (u otro tipo de audio) al disco duro
- De fácil configuración a base de una interfaz web
- Tiene la habilidad de monitorear llamadas que están siendo grabadas en tiempo real vía una tarjeta sound blaster en la PC de registro.
- Puede registrar hasta ocho llamadas simultaneas en tiempo real (puede ser expandido a múltiplos de ocho instalando tarjetas OpenLog8i adicionales).
- Se distribuye enteramente el código fuente, libre para modificaciones futuras.
- Incluye un módulo para copias de seguridad sobre CDs.

2.3.4 Bayonne

Bayonne es el servidor de telefonía del Proyecto de Telefonía GNU, y del Proyecto GNU. Actualmente en la versión 2, GNU Bayonne soporta la construcción de IVR a través de scripts, usando hardware de Voicetronix, Dialogic, Aculab, CAPI drivers y Quicklink drivers sobre GNU/Linux. Bayonne puede integrar aplicaciones Perl y Python, y ha sido comercialmente usado en sistemas de producción por muchos años.

Actualmente soporta protocolos SIP, H.323, y drivers Voicetronix. Puede ser usado en sistemas Linux de 32 y 64 bits, varios sistemas BSD, Mac OS/X y Microsoft Windows. Aún tiene trabajo en progreso para soportar hardware Dialogic, Aculab y Synway. Otros drivers serán añadidos mientras la comunidad incluye nuevos desarrolladores.

La versión actual permite la ejecución de aplicaciones IVR escritas en lenguaje script GNU Bayonne, también como accesos, conversión y reproducción de audio de URLs remotos. También ejecuta funciones básicas de interconexión de switches, incluyendo detección de tonos, regeneración DTMF, que son necesarias para operaciones básicas de compuertas. También puede funcionar como SIP Proxy o registrarse a externos servidores SIP, los cuales pueden ser usados para construir sistemas telefónicos y compuertas de entrada. Se soporta la integración con código perl, python, php, C# y Java, la habilidad de hacer operaciones de Query XML y rendering de voz de documentos en formato BayonneXML desde un web site; como la construcción de un servidor web que ofrezca paginas basadas en XMLRPC.

2.3.5 OpenH323 PSTNGW

Es un gateway de 4/8/12/16 puertos de Voz sobre IP ejecutando H.323, usando compresión de voz GSM. Este software habilita la construcción de un gateway de VoIP

por el precio de una PC y una tarjeta OpenLine. El proyecto usa el paquete de software OpenH323 PSTNGW.

2.3.6 CT Server

Es una librería cliente/servidor para desarrollo rápido de aplicaciones de Telefonía por computadora sobre Perl. Usa hardware Voicetronix y se ejecuta sobre Linux. Algunas características de este software son:

- El cliente Perl se comunica con el servidor vía TCP/IP.
- Existe un proceso cliente por línea telefónica
- Un solo proceso servidor (ctserver) maneja múltiples líneas.
- Usa puertos TCP/IP 1200, 1201, 1202, etc. para 1, 2, 3, etc. de la tarjeta CT.
- Soporta tarjetas OpenSwitch para construcción de PBX PCs.

El modelo de desarrollo está basado en lenguaje script ccscript, desarrollado por David Sugar para el servidor IVR Bayonne. David ha desarrollado este nuevo lenguaje de scripts que habilita implementaciones elegantes de menús IVR.

2.3.7 YCTA Server

Es una Interfaz Programable de Aplicaciones (API) para tarjetas de voz VoiceTronix desarrollada por Younix Networks. Es un Perl wrapper de la librería driver VoiceTronix. Directamente mapea la API VoiceTronix escrita en C, para permitir a los usuarios construir aplicaciones de telefonía por computadora usando la potencia de Perl en Linux.

2.4 Descripción de Asterisk como Servidor PBX

Asterisk es una plataforma de telefonía de fuente abierta, la cual está diseñada para ejecutarse sobre Linux. Asterisk combina unos 100 años de conocimiento telefónico en un servidor robusto de aplicaciones de telecomunicaciones fuertemente integradas. El poder de Asterisk radica en su naturaleza personalizable, complementada por inigualables características conforme a estándares. Ningún otro PBX puede ser implementado en tantas maneras tan creativas.

Aplicaciones como correo de voz (voicemail), conferencias, colas de llamadas y agentes, música en espera, parqueo de llamadas y otras son características estándares provistas dentro del software. Y aún mas, Asterisk puede integrarse con otras tecnologías de

negocio en maneras que PBX propietarios y cerrados solamente podrían soñar en hacerlo.

Aún cuando Asterisk puede parecer un poco complejo para el usuario nuevo, debido a su característica de ser open source, puede encontrarse mucha información al respecto sobre su uso y configuración.

La industria de las telecomunicaciones ha sido discutidamente la mayor de las industrias electrónicas que había permanecido intocada por la revolución del software de fuente abierta. La mayoría de empresas de fabricación de equipos de telecomunicaciones aún construye ridículamente caros sistemas incompatibles, ejecutando código complicado, antiguo en hardware obsoleto.

Como un ejemplo, Nortel's Business Communications Manager vende juntos un servidor Windows NT4.0, con un viejo Key Telephone Switch VXWorks de 15 años de antigüedad y una PC de 700MHz. Todo eso puede tener un precio entre 5 y 15 mil dólares, sin incluir teléfonos. Si se quiere hacer algo realmente interesante con el equipo, se tienen que pagar licencias extras por una funcionalidad limitada y cerrada de aplicaciones separadas. La personalización de la solución es prácticamente nula y no está incluida en el plan. No es compatible con estándares y/o futura tecnología.

Todas las mayores empresas de fabricación de equipos de comunicaciones ofrecen similares productos, encerrando al cliente en sus ciclos de producción.

Asterisk cambia completamente todo este tema. Con Asterisk, nadie te dice cómo funciona tu sistema de telefonía, ni a qué tecnología está limitado. Si se requiere una característica, es fácil de obtener. Es completamente compatible con estándares, mientras que a la vez es totalmente libre al desarrollo de innovaciones propias.

La flexibilidad viene con un precio: Asterisk no es un sistema simple de configurar. Esto no sucede porque sea ilógico, confuso o críptico, sino por el contrario, es muy sensible y práctico. Y debido a ello hay muchas maneras de lograr un cierto resultado, pero el proceso naturalmente requiere esfuerzo extra.

2.4.1 Asterisk cierra el lazo entre la Telefonía Tradicional y la Telefonía de Redes

Mientras que la Voz sobre IP (VoIP) es frecuentemente referido como un método para obtener llamadas gratuitas a larga distancia, el valor real (así como el desafío) de VoIP es que permite a la voz convertirse en nada más que otra aplicación en la red de datos.

El propósito del teléfono es permitir a las personas comunicarse. Este simple objetivo, realmente, debería hacer posible muchas otras maneras de comunicación, mas creativas y flexibles que las actualmente disponibles a nosotros. Desde que la industria de las telecomunicaciones ha demostrado su desgano de perseguir esta meta, una larga comunidad de personas han tomado esta tarea en sus manos.

Este desafío viene del hecho que la industria de las telecomunicaciones ha cambiado muy poco en el ultimo siglo, lo que muestra el poco interés en buscar nuevas formas y más viables de comunicación.

2.4.2 El Proyecto de Telefonía Zapata

El Proyecto de Telefonía Zapata fue concebido por Jim Dixon, un ingeniero consultor de telecomunicaciones que fue inspirado por los increíbles avances en las velocidades de CPU que la industria de la computación había llegado a alcanzar. La creencia de Dixon era que sistemas de telefonía mucho más económicos podrían ser creados si existiera una tarjeta en la PC que tuviera nada más que los componentes electrónicos básicos requeridos para hacer interfase con los circuitos de telefonía. En lugar de tener componentes costosos en la tarjeta, el procesamiento digital de señales (DSP) sería manejado por software en el CPU. Mientras esto supondría una tremenda carga en el CPU, Dixon estaba seguro de que el bajo costo de los CPUs en relación con su rendimiento haría de ellos más atractivos que los costosos DSPs, y, más importante, que la tasa de precio/rendimiento continuaría mejorando mientras los CPUs mejoraran en cuanto a rendimiento.

Como muchos visionarios, Dixon creía que muchos otros verían también esta oportunidad, y que él meramente tendría que esperar a que alguien más creara lo que para él era un obvio paso para la mejora en la tecnología. Después de muchos años, él se dio cuenta que no solo nadie había creado esas tarjetas, pero también parecía que nadie más las iba a crear. En este punto, era claro que si él quería una revolución, tendría que comenzarla por sí mismo. Entonces, el Proyecto de Telefonía Zapata fue creado.

Dado que este concepto era tan revolucionario, y era claro que movería muchas olas en la industria, llamó al proyecto "Zapata", en relación al revolucionario mexicano Emiliano Zapata.

Todas las aplicaciones de telefonía por computadora tienen su inicio en este concepto, como resultado de la contribución de Jim Dixon.

2.4.3 Cambios masivos requieren tecnología flexible

El más exitoso sistema de marcado telefónico en el mundo tiene una limitación de diseño que ha sobrevivido 15 años de usuarios rogando lo que parecía un simple cambio: Cuando se determina el número de veces que un teléfono va a sonar antes de que sea reenviado a la casilla de voz, se puede elegir entre 2, 3, 4, 6 ó 10 ciclos de sonido. Para tener una idea de cuántas personas alguna vez quisieron cambiar por tener 5 ciclos de sonido, u otros, sabemos que las empresas de manufactura de equipos de telecomunicaciones no podían más que mover sus cabezas negativamente para resolver este simple problema. Esa es la forma en la que funciona, decían a los usuarios, y ellos tenían que arreglárselas.

Otro ejemplo, con el mismo sistema, es que el nombre con el que se programa el set puede solamente tener 7 caracteres de largo. En la última década de los 80s, cuando este sistema particular fue construido, la RAM era muy preciada y cara, y el almacenamiento de esos 7 caracteres para docenas de sets representaba un claro costo de hardware. Pero no existe excusa para hoy en día, y tampoco hay planes para cambiarlos. Difícilmente el tema es oficialmente mencionado como problema.

La realidad es que cada PBX que existe sufre de defectos. No importando qué tan rico en características sea, algo ha sido siempre dejado de lado, porque incluso en los PBX que poseen más ricas características siempre fallan en anticipar la creatividad del cliente. Un pequeño grupo de usuarios deseará una extraña pequeña característica que el equipo de diseño no lo pensó o no podía justificar el costo de construirlo, y, desde que el sistema es cerrado, los usuarios no podrían construirlo por sí mismos.

Si la Internet hubiera sido obstaculizado por los intereses y reglas comerciales, es indudable que no se hubiera desarrollado ni hubiera tenido la amplia aceptación que actualmente posee. Los cientos de miles de mentes que colaboraron en la creación de la Internet produjeron algo que ninguna corporación hubiera podido hacerlo sola.

Como con cualquier otro proyecto de fuente abierta, como Linux y la Internet, la explosión de Asterisk fue empujada por los sueños de personas que sabían que debería de haber algo mejor que lo que la industria estaba produciendo. La fuerza de la comunidad está en que no está compuesta de empleados asignados a tareas específicas, pero de personas que trabajan en todo tipo de industrias, con todo tipo de experiencias, y todo tipo de ideas acerca de lo que la flexibilidad significa, y lo que la apertura trae consigo. Estas personas sabían que si uno tomaba las mejores partes de los varios PBX y los separara en componentes interconectados a una caja de ladrillos LEGO, podrían empezar a concebir cosas que no podrían sobrevivir a un análisis de riesgo de proceso de una corporación tradicional.

Asterisk esta nutrido de un gran numero de personas, muchas más que las que se usaron en la creación de cualquier PBX. La mayoría de empresas de manufactura de telecomunicaciones dedicaron no más que unos cuantos desarrolladores a cualquier producto. Asterisk tiene cientos. La mayoría de PBXs tienen en el mundo un equipo de soporte de una docena de expertos reales; Asterisk tiene cientos de ellos.

La profundidad y maestría que envuelve a este producto es incomparable en la industria de las telecomunicaciones. Asterisk tiene ahora la atención de los viejos profesionales Telco y las compañías de telecomunicaciones y aún más los nuevos codificadores de comunicaciones de datos que ayudaron a construir la Internet. Esas personas comparten una idea en común: que la industria de las telecomunicaciones necesita una propia revolución, y Asterisk es el catalizador.

2.4.4 Asterisk como Software PBX de fuente abierta

La flexibilidad que Asterisk permite a los usuarios crea posibilidades que los mejores sistemas propietarios pueden solamente soñar.

Nunca en la industria de las telecomunicaciones ha habido un sistema encajado con las necesidades de negocio disponibles, a cualquier precio. Asterisk es la tecnología que lo posibilita, y, con Linux, se convertirá en algo muy raro encontrar a una empresa que no ejecute alguna versión de Asterisk, en alguna capacidad, en alguna red, resolviendo algún tipo de problema que solo se puede resolver con sistemas abiertos.

La aceptación parece llegar mucho más rápido de lo que sucedió con Linux, por varias razones:

- Linux ya ha marcado la ruta que llevó a las aplicaciones de fuente abierta a su aceptación, así que Asterisk solo tiene que seguir ese camino.
- La industria de las telecomunicaciones está lisiada, con ningún liderazgo proveniente de los grandes competidores. Asterisk tiene una visión excitante y realista a la vez que comprometedora.
- Los usuarios finales están cansados de funcionalidades limitadas e incompatibles, además de muy pobre soporte. Asterisk resuelve estos primeros dos problemas; y la comunidad ha mostrado una verdadera pasión por el segundo caso.

CAPITULO III

SISTEMA PROPUESTO

3.1 Arquitectura del Sistema

El sistema propuesto, tiene por finalidad interconectar los distintos tipos de tecnología usada para comunicaciones. Esto significa interconectar redes de voz y datos con la red telefónica tradicional.

Por un lado, se encuentra la telefonía tradicional, con teléfonos analógicos y digitales, mientras que por otro lado está la telefonía actual, obtenida por intermedio de voz sobre IP, usando distintos tipos de protocolos.

Hasta que las redes de voz e Internet lleguen masivamente a toda la población, estaremos usando la telefonía tradicional junto con la nueva de voz sobre IP, en un híbrido que permitirá la transición suave, mientras las nuevas tecnologías amplían su desarrollo.

La arquitectura del sistema es fundamentalmente simple (7), pero diferente a la mayoría de productos telefónicos. Esencialmente, Asterisk actúa como un middleware, conectando tecnologías telefónicas con aplicaciones, creando un entorno consistente para desarrollar un entorno de telefonía híbrido. Las tecnologías telefónicas pueden incluir: VoIP, servicios como SIP, H.323, IAX y MGCP (gateways y teléfonos), como tecnologías TDM tradicionales como T1, ISDN, PRI, POTS analógicos y servicios PSTN, Basic Rate ISDN (BRI), y más. Las aplicaciones telefónicas incluyen cosas como puenteo de llamadas, conferencias, correo de llamadas, auto respuesta, scripts de IVR personalizados, parqueo de llamadas, intercom, y muchos más.

El núcleo de Asterisk contiene muchos motores para controlar cada rol crítico de la operación de software. Cuando asterisk se inicia, el módulo de carga dinámico carga e

inicializa cada uno de los drivers que proveen los canales, formato de archivos, manejadores del registro detallado de llamadas.

El objetivo del sistema propuesto es el de cambiar un PBX tradicional por un servidor PBX, basado en software Asterisk, que sirva como central telefónica para interconectar la red de telefonía básica con la red de datos Internet.

El resultado es obtener un sistema, el cual pueda, usando las facilidades de Internet ser usado en la interconexión de puntos alejados. De esta manera se podrá, por ejemplo, configurar una extensión o anexo en cualquier otro lugar del mundo solamente configurándolo usando alguno de los protocolos de VoIP disponibles.

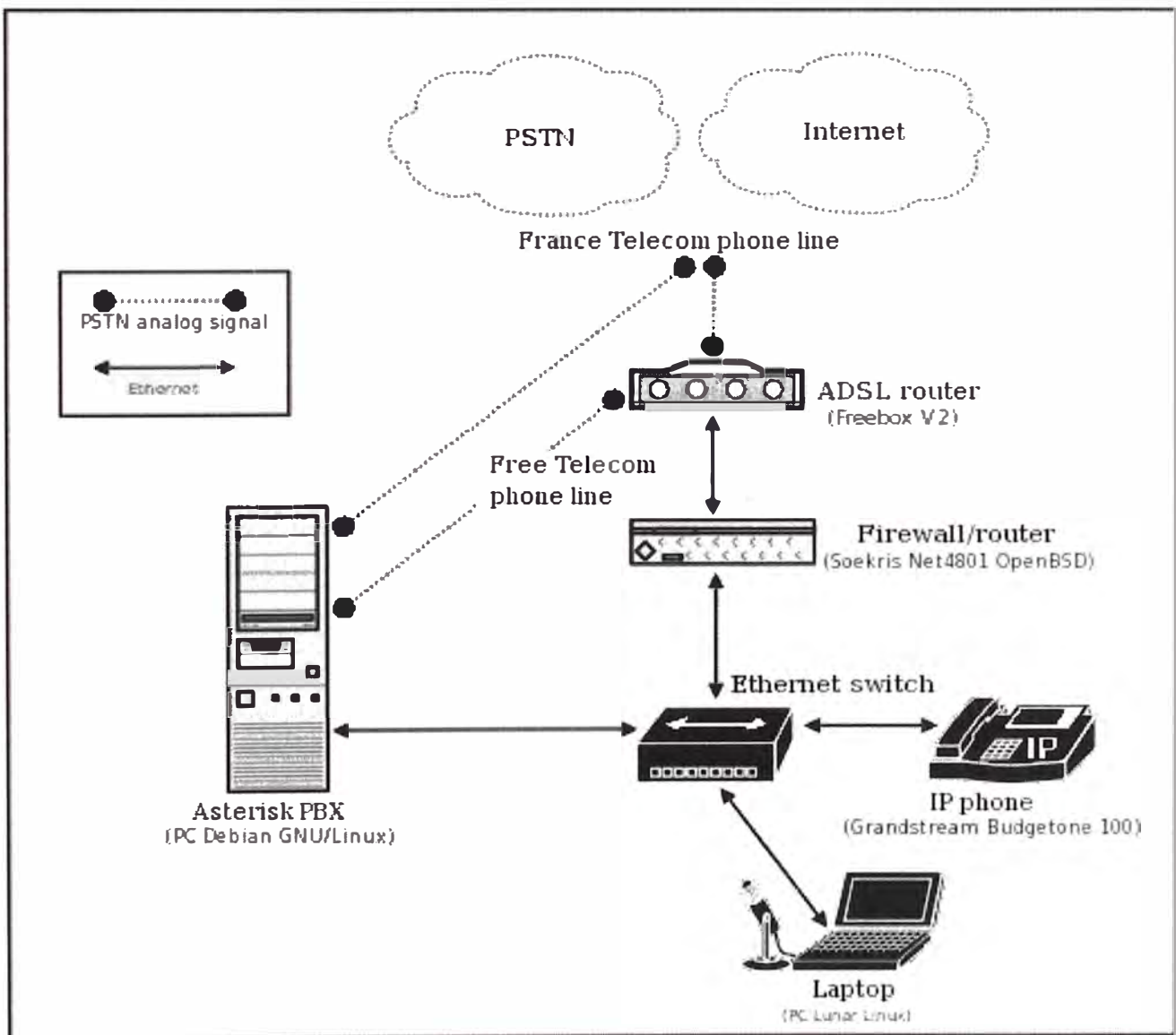


Figura No. 3.1. Servidor Asterisk sobre Linux como gateway entre las redes PSTN e Internet.

Además de las ventajas obtenidas en la interconexión de la red de datos con la red de telefonía, se tendría un servidor PBX con la capacidad necesaria para configurar cualquier tipo de aplicación que se requiera.

En la figura 3.1, podemos obtener una extensión, basada en softphone en la laptop que se encuentra conectada al switch ethernet. Igualmente, es posible, mediante la configuración necesaria, proveer extensiones a teléfonos IP (teléfonos SIP), softphones (aplicaciones telefónicas en la computadora) o dispositivos telefónicos que se encuentren en algún lugar en la Internet.

Para la puesta en práctica de la arquitectura del sistema, es necesario evaluar primero el rendimiento del hardware que se requiere usar.

La manera en que el sistema va a ser usado tiene un rol importante en seleccionar las especificaciones de la computadora que se necesite usar. Mientras no hay una forma de especificar claramente una matriz que refleje que tan potente tiene que ser el sistema para soportar las aplicaciones que se instalaran, sí hay ciertos puntos que son necesarios tenerlos en cuenta:

- El número máximo de conexiones concurrentes que se espere que soporte el sistema, ya que cada conexión incrementa la carga en el sistema.
- El porcentaje de tráfico intensivo que requerirá el procesador DSP de codecs comprimidos (como G.729, GSM, etc.). El trabajo que Asterisk realiza en software puede decaer por el número de conexiones concurrentes. Un sistema que soporta fácilmente 50 llamadas concurrentes en G.711 puede ser traído a tierra por una conferencia de 10 canales comprimidos G.729 .
- Si se proveerá Conferencias y qué nivel de actividad de conferencias se requerirá. Conferencias requiere convertir y mezclar las entradas de audio de un codec distinto a múltiples salidas. En tiempo real, eso puede suponer una enorme carga al CPU.
- Cancelación de Echo: Puede requerirse en cualquier llamada donde la interfaz hacia la PSTN esté involucrada. Como la cancelación de eco es una ecuación matemática, a más tenga que usarla el sistema, mayor la carga que soportará el CPU.
- Codecs y transcodificación: Un codec (abreviación de coder/decoder o compression/decompression) es un conjunto de ecuaciones matemáticas que definen cómo una onda analógica será digitalizada. Las diferencias entre los

varios codecs son debido en gran parte a los niveles de compresión y calidad que ofrecen. Generalmente hablando, a mayor compresión requerida, mayor trabajo que el DSP debe de hacer para codificar o decodificar la señal. Codecs no comprimidos, por tanto, ponen menos carga en el CPU pero requieren mayor ancho de banda. La selección de buenos codecs debe de ser un balance entre ancho de banda y carga del procesador.

- **Lógica de Scripts en el Dialplan:** Siempre que Asterisk tiene que pasar el control a un programa externo, hay una penalidad en rendimiento. El dialplan debe de construirse con tanta lógica como sea posible. Si se usan scripts externos deberían de diseñarse teniendo en cuenta eficiencia y rendimiento como metas finales.
- **Optimizaciones de Kernel:** Un kernel optimizado para mejorar el rendimiento de una aplicación específica es algo que muy pocas distribuciones Linux ofrecen por defecto.
- **Latencia de IRQ:** La latencia de petición de interrupción (IRQ) es el retardo entre el momento en el que una tarjeta periférica (como una tarjeta de interfaz telefónica) requiere al CPU que se detenga en lo que está haciendo para responder a la petición y manejar esa tarea. Los periféricos de Asterisk (especialmente las tarjetas Zaptel) son extremadamente intolerantes a la latencia de IRQ.
- **La estabilidad y calidad de un sistema Asterisk depende de los componentes que se seleccionen para su arquitectura.** Asterisk es una bestia grande y como tal, espera siempre que la alimenten muy bien. Los altos costos no son siempre sinónimos de calidad, pero es necesario darle un buen procesador con un buen ancho de banda.

En el sistema que fue preparado por IT Synergy para manejar la red de voz del PNUD (Programa de las Naciones Unidas para el Desarrollo) que interconectaba a todas las oficinas de medio oriente, se usó un servidor con chip Intel Pentium IV 3.4MHz con 1GB RAM, que será descrito y visto en detalle en los siguientes capítulos. Este modelo se usó para proveer la interconexión entre los PBX de varias oficinas del PNUD del medio oriente, de tal manera que las llamadas entre las oficinas y entre los países sean totalmente a través de Internet, sin ocasionar gastos por los altos costos de las llamadas internacionales.

3.2 Justificación del uso de la tecnología propuesta

El sistema PBX basado en Asterisk planteado en el presente informe, al ser un sistema computacional, necesita de tarjetas especiales para conectarse a la red de telefonía pública. Estas tarjetas que proveen canales FXS, FXO pueden ser obtenidas de varios fabricantes, como Digium, Voicetronix, Sagoma, etc que fabrican tarjetas compatibles con Asterisk. En particular, el costo asociado a esta tarjeta, como el de una computadora es mucho mas económico que el que se genera al comprar un PBX tradicional con algunas de las características que posee Asterisk, como correo de voz, integración con protocolos de voz sobre IP como SIP, auto respuestas, etc.

Las soluciones de telefonía por computadora están en crecimiento desde hace cinco años atrás y a la fecha se ha progresado, en el campo de tecnologías de fuente abierta, a limites muchos mas competitivos que las soluciones que proveen sistemas comerciales como Cisco, Nortel, Panasonic, con un sinnúmero de aplicaciones. El sistema planteado tiene la finalidad de ofrecer la mejor flexibilidad que es posible alcanzar en una central telefónica en sincronía con las actuales tecnologías de comunicación que puedan brindar no solo inter-conectividad entre los diferentes protocolos y medios, sino también pueda ser extensible a otras futuras aplicaciones en la industria de voz sobre IP.

Los protocolos de comunicación de voz sobre IP como SIP están en completo auge, y se prevé que sea el protocolo dominante de comunicaciones en los años que vienen. Igualmente se prevé que los servicios de tele-conferencia comiencen a desarrollarse en entornos profesionales, a base de Asterisk y con el soporte, cada vez más creciente, de la comunidad de fuente abierta. Las aplicaciones de tele-conferencia que proveen empresas de manufactura como Cisco son extremadamente costosas y todavía muy lejos de ser usados en pequeñas y medianas empresas. Sin embargo, esto podría cambiar con la contribución del desarrollo que se esta produciendo en la comunidad Asterisk en los últimos años.

En suma, cualquier funcionalidad que esté disponible en otros sistemas propietarios PBX, está también disponible en Asterisk, y con muchas más opciones y facilidad para su configuración y/o personalización. Es lo más recomendable en lo que a soluciones de PBX existen actualmente en el mercado, con perspectivas a soportar futuras tecnologías emergentes y además de menor costo de implementación.

3.3 Ventajas comparativas de esta solución en el Mercado Peruano

Para una empresa peruana, que esta entrando al mercado a operar un servicio específico o a vender un cierto producto, es primordial contar con un sistema telefónico que sirva de base para las comunicaciones efectuadas hacia dentro y fuera de la empresa. Para una pequeña empresa es tal vez suficiente uno o un par de números telefónicos, pero mientras mas grande sea la empresa, será necesario contar con una central telefónica que se encargue de manejar y rutear las llamadas a los canales destinados dentro de ella.

Para esta empresa existen muchas opciones de las cuales elegir, pero principalmente, si es que es una empresa cuyo foco de negocio no esta focalizado dentro de tecnologías de información, requerirá entonces de un sistema autosuficiente y de mínimo mantenimiento que le provea de estas facilidades. En caso contrario, y si es que la empresa cuenta con un departamento o área de IT con capacidad instalada en tecnologías de fuente abierta, las posibilidades comienzan a ser ilimitadas para la manera en la que puede llegar a crecer. En ambos casos, Asterisk provee de las características necesarias para su uso en distintos entornos y configuraciones, con mínimo mantenimiento.

En el caso específico del Perú, el número de profesionales en el campo de tecnologías de información de fuente abierta como Linux, y en mucha menor medida, Asterisk está en crecimiento constante desde hace unos cuantos años atrás. Actualmente ya se pueden encontrar muchos profesionales especializados en proveer soluciones Linux. Sin embargo, el uso de Asterisk como servidor PBX aun no es muy conocido en el mercado, y por consiguiente, aun existe un cierto temor al uso de estas nuevas tecnologías como soporte profesional de los sistemas de comunicación de las empresas.

Sin embargo, el mercado peruano, con los profesionales que están disponibles en la actualidad, especializados muchos de ellos en tecnologías de fuente abierta, podría fácilmente ser capaz de dar ese salto tecnológico y adaptar la tecnología Asterisk como base para el desarrollo de sus centrales telefónicas.

Esto da a entender que actualmente existe en el Perú, como en muchos otros países, una oportunidad de negocios para pequeñas empresas que estén interesadas en proveer soporte y soluciones de sistemas basados en Asterisk, que es lo que aun se esta comenzando a desarrollar en el Perú. Los profesionales existen, pero aun no hay

empresas formadas que provean soporte en este tipo de tecnologías. Las razones, entre otras, es que el hardware necesario para la implementación de este tipo de soluciones tiene que ser importado del extranjero, el cual no es muy conocido aun en el medio peruano. Aunque el costo del hardware es relativamente bajo (Una tarjeta analógica con un canal FXO y uno FXS puede ser obtenido comprándose por Internet por un precio alrededor de los 200 dólares), no hay una difusión de que estas soluciones podrían fácilmente reemplazar y modernizar la estructura tecnológica de las comunicaciones, dominadas por las empresas de manufactura de centrales telefónicas tradicionales.

Para las empresas que deseen instalar Asterisk como servidor PBX, pero que no poseen capacidad instalada dentro de ellas mismas para configurar, modificar y/o mantener la solución, necesitan hacer un out-sourcing a alguna otra empresa de soporte con experiencia y especialización en ese tipo de tecnologías para sentirse seguras de que el sistema que están adoptando este operativa y funcional durante el tiempo de vida que tenga. Estas empresas de soporte de tecnología Asterisk aun no están presentes en el mercado peruano, lo cual puede traducirse en una oportunidad única de negocios para los nuevos profesionales que salen de las universidades y estén dispuestos a crear pequeñas empresas de soporte de soluciones de telefonía basadas en fuente abierta.

La consecuencia de la aparición de este tipo de empresas que provean un soporte profesional en el mercado peruano para las centrales telefónicas de fuente abierta basadas en Asterisk traerá consigo una modernización de la infraestructura de las telecomunicaciones en todo el país, que tendría en el mediano plazo la capacidad de migrar a las comunicaciones de voz vía Internet y dejar de lado la infraestructura actual de telecomunicaciones tradicional.

Por otro lado, para las empresas que adopten Asterisk como plataforma de operación de sus centrales telefónicas, tendrían la capacidad de crecer a la par con las tendencias tecnológicas actuales, además de seguir usando las características básicas que proveen las soluciones tradicionales.

Otra de las ventajas de la adopción de este tipo de tecnologías es su bajo costo en relación a las soluciones de sistemas propietarios como Cisco, Nortel y otros en el mismo campo. Para una empresa mediana o grande, la inversión es mucho menor, sin embargo, las empresas pequeñas que no tienen capacidad instalada en el área de IT aun el costo es comparativamente mayor, si es que se considera que ellos mismos no serian

capaces de mantener el sistema y por consiguiente serian dependientes de un servicio que tenga que ser efectuado por una empresa externa. Esto sin embargo cambiara en los próximos años, ya que tanto el costo de la tecnología siempre tiende a la baja con el paso del tiempo.

Entre las nuevas características que tendrían disponibles las empresas que adopten este tipo de sistemas basados en Asterisk tenemos los siguientes:

- Utilización de los servidores internacionales de llamadas vía Internet e integrarlo al propio sistema PBX de la empresa para efectuar llamadas internacionales a costos bastante mas bajos que los costos usados en telefonía tradicional.
- Oficina móvil, ya que uno podría llevar su extensión telefónica en su laptop a cualquier sitio, usando un soft-phone.
- Integración de la central telefónica con el software de mensajería instantánea como Google Talk o Jabber, poniendo a la empresa al alcance de cualquiera que tenga un usuario en Google. De esta manera uno podría llamar a la empresa sin necesidad de marcar un numero, sino usando el software Google Talk desde su computadora, con un consiguiente ahorro en llamadas telefónicas de parte de los clientes.
- Puesta en marcha de un sistema de respuestas interactivo o también llamado recepcionista virtual, que se encargue de hacer la interfaz entre el usuario que llama a la empresa y los diferentes canales y estructura interna dentro de la empresa.
- Integración de la central telefónica con proveedores DID (Direct Inward Dialing), que es una característica desarrollada por AT&T para poder obtener un numero telefónico local en un país extranjero, y recibir las llamadas localmente.
- Poseer la tecnología necesaria para integrarse al sistema ENUM, que es un conjunto de protocolos que provee la integración de telefonía Internet y tradicional en un solo numero. Cuando una persona llama a ese número, el sistema primero verifica que pueda usar la conexión Internet, y si esta no esta disponible se usa la red telefónica tradicional.
- Primer paso a un futuro soporte de video-conferencias en base a Asterisk en el corto plazo, teniendo la infraestructura apropiada para efectuarlo.

Todas estas características, y muchas otras más, son posibles de adoptarse si se opta por usar el sistema Asterisk como base para la puesta en marcha de un servidor PBX.

3.4 Tecnología usada para la conexión con la PSTN

Para conectar Asterisk con cualquier equipo de telecomunicaciones, se necesita el hardware correcto. Este hardware requerido determinará lo que se quiere lograr con nuestro servidor asterisk.

Asterisk permite puentear el circuito de red de telefonía conmutada (casi siempre referido como la TDM network, que quiere decir: Time Division Multiplexing o en castellano, Multiplexación por división de tiempo, usado para llevar tráfico por la red PSTN – Public Switch Telephone Network), con una red de datos de paquetes conmutados (Packet-Switched Data Networks), que popularmente son llamados redes de Voz sobre IP, aunque las redes de voz sobre IP no son el único método de transmitir voz sobre una red de paquetes (Voz sobre frame relay era muy popular en los últimos años de la década de los 90).

Gracias a la arquitectura abierta de Asterisk (y el código de fuente abierta), es también posible conectarse a cualquier hardware de interfaz compatible con estándares de telefonía. La selección de tarjetas de interfaz de telefonía de fuente abierta es actualmente limitada, pero mientras el interés en Asterisk crece, eso cambiará rápidamente. Por el momento, una de las formas más populares y con buena relación costo-beneficio para conectarse a la PSTN es por el uso de tarjetas de interfaz que han evolucionado del trabajo del proyecto Zapata Telephony (<http://www.zapatatelephony.org>).

3.4.1 Tarjetas de Interfaz Analógica

A menos que se necesiten muchos canales de voz (o se tiene suficiente dinero para invertir cada mes en facilidades de telecomunicaciones), se puede decir que la interfaz PSTN consistiría de uno o más circuitos analógicos, cada uno de los cuales requerirían un puerto FXO (Foreign Exchange Office).

Digium, la compañía que promueve el desarrollo de Asterisk, produce las tarjetas de interfaces analógicas más populares, conocidas como TDM400P (TDM400P no es, en realidad, una tarjeta TDM por ningún sentido. Es analógica).

La TDM400P (ver Figura 3.2) es una tarjeta de 4 puertos base, para la inserción de hasta cuatro tarjetas hijas, las cuales ofrecen puertos FXS (Foreign eXchange Station) o FXO (Foreign Exchange Office). La TDM400P puede ser comprada con esas tarjetas pre-

instaladas, y Digium ha designado números de partes para describir esa configuración. La convención de nombres es TDM x y B, donde x e y son números que representan la cantidad de tarjetas FXS y FXO en la tarjeta respectivamente. FXO y FXS se refieren a los lados opuestos de un circuito analógico. El que se necesite se determinará con lo que se requiera conectar. Las tarjetas de interfaz analógica de Digium proveen conexión a la línea y estaciones de telefonía tradicional:

- **TDM400P:** Es una tarjeta half-length PCI 2.2-compliant que soporta interfaces FXS y FXO para conectar teléfonos analógicos y POTS a través de una PC.
- **TDM2400P:** Tarjeta full-length PCI 2.2-compliant para conexión con teléfonos analógicos y líneas POTS analógicas a través de un PC. Soporta una combinación de hasta 6 canales FXS y módulos FXO para un total de 24 líneas.
- **X100P:** Provee una sola conexión analógica con el PSTN.
- **S100U:** Provee una sola conexión POTS a través de USB.

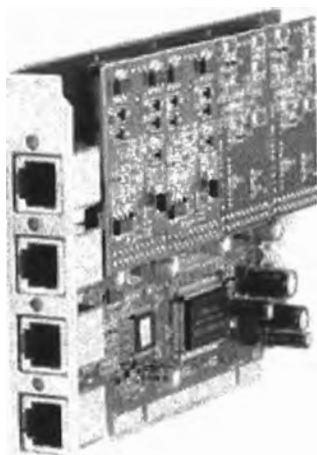


Figura No. 3.2. Tarjeta Analógica TDM400P

En la figura se muestra una tarjeta TDM400P, que toma el lugar de un costoso banco de canales, y trae el sistema a un precio muy bajo. Usando módulos S110M (tarjetas verdes pequeñas en la imagen anterior que proveen puertos FXS) y X100M (tarjetas rojas

pequeñas en la imagen anterior que provee puertos FXO) con el TDM400P, uno puede crear una solución con soporte para un buen rango de teléfonos.

Otra compañía que produce tarjetas analógicas compatibles con Asterisk es Voicetronix. Ellos tienen tres tarjetas en su línea analógica: OpenLine4, OpenSwitch6 y OpenSwitch12.

3.4.2 Tarjetas de Interfaz Digital

Si se requiere más de 10 circuitos, o se requiere conectividad digital, entonces se necesitará estar en el mercado con una tarjeta T1 o E1 (T1 y E1 son circuitos de telefonía digital). Téngase en mente, que el costo mensual de un circuito digital PSTN varía ampliamente. En algunos lugares, tantos pocos como 5 circuitos pueden justificar un circuito digital, mientras que en otros, la tecnología puede nunca ser justificada en costos. La mayor competición existente en el área trae consigo una mejor opción de encontrar una mejor configuración.

EL proyecto de telefonía Zapata originalmente produjo una tarjeta T1, la Tormenta, que es el antecesor de la mayoría de tarjetas T1 compatibles con Asterisk. La tarjeta original Tormenta es considerada obsoleta, pero aún trabaja con Asterisk. Actualmente la única compañía que se conoce que produce esas tarjetas es Varion.

Digium produce muchas tarjetas de interfaz de circuito digital. Las características en esas tarjetas son casi las mismas. Las diferencias están en si ellas proveen interfaces T1 o E1 y en cuantas interfaces se proveen. Aunque es técnicamente posible, el consenso general en la comunidad Asterisk es que no más de una de esas tarjetas debería ser empleada en un solo sistema.

Sangoma, que ha estado produciendo tarjetas WAN de fuente abierta por muchos años, ha añadido recientemente soporte Asterisk a sus tarjetas E1/T1. Las tarjetas Sangoma contienen poderosos circuitos FPGA (Field Programmable Gate Arrays), los cuales la hacen extremadamente flexible. En un entorno Asterisk, por ejemplo, ellos han sido programados para hacer de interfaz con el driver de canal de Zapata. Como nota, la tarjeta Sangoma Frame Relay figura en el desarrollo original de Asterisk. Además Sangoma tienen una larga historia como soporte de interfaces WAN de fuente abierta para Linux.

Las tarjetas de interfaz digital de Digium tienen un diseño innovador, y vienen en varios modelos. Entre los modelos más comunes, encontramos:

- **B410P:** Con DSP basado en cancelación de eco sobre la tarjeta, soportando modos NT y TE sobre una interfaz S/T.
- **TE412P:** Ofrece DSP basado en cancelación de eco, soportando entornos E1, T1 y J1, configurable a una por puerto o por tarjeta.
- **T100P:** Un solo entorno T1 o conexión PRI (permite mezcla de datos/voz)
- **E100P:** Un solo entorno E1 o conexión PRA (permite mezcla de datos/voz)
- **T400P:** Cuádruple entorno T1 o conexión PRI (permite mezcla de datos/voz)
- **E400P:** Cuádruple entorno E1 o conexión PRAI (permite mezcla de datos/voz)

De las tarjetas descritas anteriormente, la más usada o más popular en cuanto a uso general es la tarjeta TE412P, que se muestra en la figura siguiente. El TE412P es una tarjeta basada en DSP con cancelación de eco. Provee un algoritmo de grado proveedor certificado que ha sido probado con alto rendimiento para la cancelación de eco. Con este módulo, 16ms o 128taps de cancelación de eco es posible a través de 128 canales.

La tarjeta permite a los usuarios eliminar colas de eco de hasta 128ms o 1024 taps a través de todos los 128 canales en modo E1 o 96 canales en modo T1/J1. Además provee una calidad de voz superior a las anteriores tarjetas.

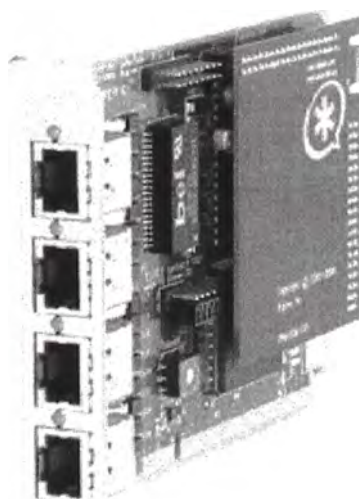


Figura No. 3.3. Tarjeta Digital TE412P

Digium ha diseñado el TE412P (ver Figura 3.3) para ser completamente compatible con las aplicaciones de software existentes y está completamente integrado con la plataforma de Asterisk Open Source PBX/IVR. Además, los drivers de fuente abierta proveen una API para personalización en el desarrollo de aplicaciones.

3.4.3 Banco de Canales

Un banco de canales está definido por un dispositivo que permite a un circuito digital ser de-multiplexado en varios circuitos analógicos (y vice-versa). Más específicamente, un banco de canales permite conectar teléfonos analógicos y líneas en un sistema a través de una línea T1. La siguiente figura ilustra como un banco de canales encaja dentro de un sistema telefónico en una oficina típica.

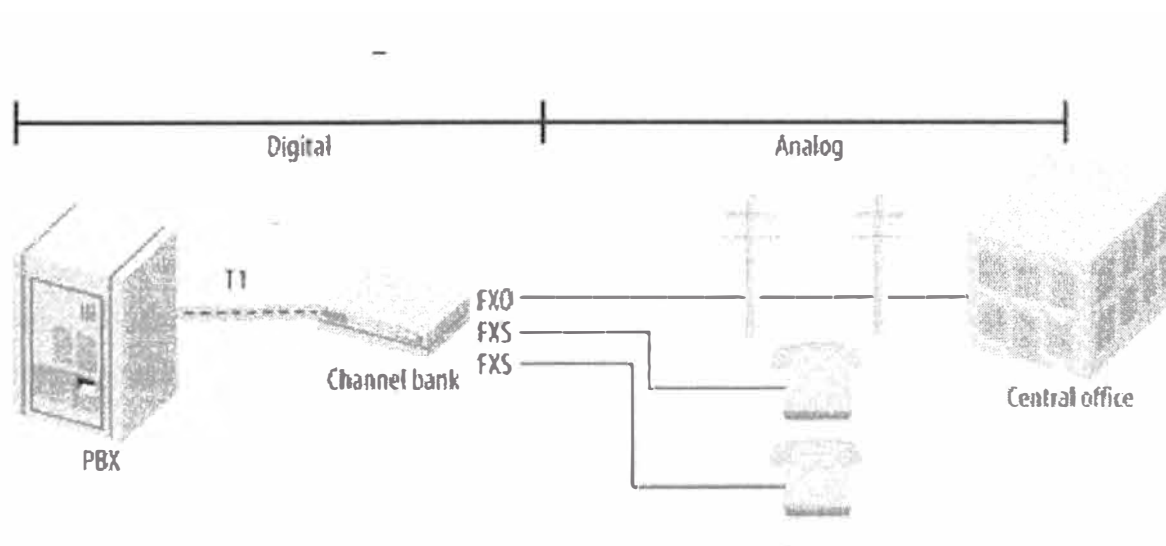


Figura No. 3.4. Estructura de funcionamiento de un Banco de Canales.

A pesar de lo caros que puedan llegar a costar, muchas personas sostienen fuertemente que la única manera de integrar circuitos analógicos y dispositivos con Asterisk sea a través de un Banco de Canales.

3.4.4 Otros tipos de Interfaz con la PSTN

Muchos VoIP gateways existentes pueden ser configurados para proveer acceso a la red de circuitos PSTN. Generalmente hablando, esos serán de mayor uso en un sistema pequeño (una o dos líneas). Ellos pueden también ser complicados de configurar, como la interacción entre las distintas redes y dispositivos requiere de un entendimiento sólido de los fundamentos de telefonía y VoIP. Vale la pena dar una mirada a como funcionan, y

unidades muy populares son desarrolladas por compañías como Supura, Grandstream, Digium, Multitech y otras.

Otras formas de conexión con la PSTN son a través del uso de circuitos ISDN de Interfaz de tarifa básico (Basic Rate Interface - BRI). BRI es un estándar de telefonía que especifica un circuito de dos canales que lleva 144Kbps de tráfico. Es muy poco usado en América y en la mayoría del resto del mundo, pero es muy popular en Europa. Debido a la variedad de las formas en que esta tecnología puede ser implementada, no se invertirá mucho en describir los detalles BRI en el presente trabajo.

3.5 Distintos tipos de Protocolos de Voz sobre IP

Para conectar el servidor Asterisk a la red de paquetes, no es necesario más que la interfaz de red, ya que es un servidor Linux.

La industria de las telecomunicaciones se extiende por cerca de 100 años, y Asterisk integra todas, sino casi todas las mayores tecnologías que se han usado por todo este largo tiempo. Para sacar el mayor provecho de la red de telecomunicaciones construida, se hace necesario entender toda la gama de diferencias entre varios codecs y protocolos de comunicación de voz.

El mecanismo para llevar una conexión de VoIP generalmente envuelve una serie de transacciones de señalización entre los lados opuestos (y gateways que se encuentren en el camino), culminando en dos flujos de datos (media streams) persistentes (uno para cada dirección) que llevan la conversación. Existen muchos protocolos existentes que manejan esto. En esta sección discutiremos aquellos importantes para la VoIP en general como para Asterisk específicamente.

3.5.1 IAX: Inter-Asterisk eXchange Protocol

Es un protocolo abierto, que cualquiera puede descargar y desarrollar para él, pero no es aún un estándar de ninguna clase. Oficialmente, la versión actual de IAX es IAX2, y el soporte para IAX1 ha sido dejado obsoleto. Si es que se habla de IAX, se estará hablando de IAX2.

Este protocolo está soportado por el módulo `chan_iax2.so` en el código fuente de Asterisk.

Este protocolo fue desarrollado por Digium con el propósito de efectuar comunicación entre servidores Asterisk. IAX es un protocolo (parecido a SIP) que usa un solo puerto UDP (4569) para el señalamiento de canal y como flujo de protocolo de transporte en tiempo real (RTP). Esto hace fácil la integración dentro de un firewall y mucho más fácil de funcionar detrás de un NAT.

IAX también tiene la habilidad única de truncar múltiples sesiones en un solo flujo de datos, lo cual puede ser una tremenda ventaja de ancho de banda cuando se envían muchos canales simultáneamente a un servidor remoto. El truncado permite múltiples flujos de datos a ser representados con una sola cabecera de datagrama, para bajar la sobrecarga asociada con los canales individuales. Esto ayuda a obtener poca latencia y reducir el poder de procesamiento y ancho de banda requerido, permitiendo al protocolo escalar mucho más fácil en un largo número de canales activos entre ambos lados de la conexión.

Dado que IAX ha sido optimizado para voz, ha recibido muchas críticas por no soportar mejor video, sin embargo, IAX tiene el potencial de llevar cualquier tipo de flujo de media deseado. Debido a que es un protocolo abierto, los tipos de media del futuro serán incorporados en la comunidad Asterisk cuando se desee.

IAX incluye la habilidad de autenticación de tres maneras: texto plano, MD5 hashing e intercambio de llaves RSA. Esto claro, no tiene nada que ver con la encriptación de las rutas de los datos o las cabeceras de entre ambos lados de la conexión. Muchas soluciones incluyen una red privada virtual (VPN) o software de encriptación del flujo en otra capa de tecnología, que requiere que los lados pre-establezcan un método que tengan esos túneles configurados y operacionales. En el futuro, IAX puede ser habilitado para encriptar flujos entre puntos finales con el uso de una llave RSA, o una llave dinámica en el seteo de llamada, permitiendo el uso de un automático cambio de llaves. Esto sería muy atractivo para crear enlaces seguros de conexión con una institución privada como un banco. Muchas organizaciones van a necesitar ese nivel de seguridad de acceso a las conexiones.

IAX fue deliberadamente diseñado para trabajar desde detrás de dispositivos que proveen NAT (como firewalls o Proxy servers) (8). El uso de un solo protocolo UDP para señalización y transmisión de datos también mantiene el número de puertos requeridos en el firewall al mínimo. Estas consideraciones han ayudado a hacer a IAX uno de los protocolos mas fáciles (sino, el mas fácil) de implementar en redes seguras.

3.5.2 SIP: Session Initiation Protocol

El protocolo de Inicio de Sesión (SIP), ha tomado el mundo de VoIP como una tormenta. Originalmente fue considerado un poco más que una idea interesante, ahora SIP parece haber destronado a H.323 como el protocolo escogido en los puntos finales de la red. La promesa de SIP es que cada fin de una conexión es un par (peer), y el protocolo negocia capacidades entre ellos. Lo que hace a SIP interesante es que es un protocolo relativamente simple, con una sintaxis similar a aquella de otras familias de protocolos como http o SMTP.

SIP es soportado en Asterisk con el modulo chain_sip.so.

SIP fue originalmente propuesto al Internet Engineering Task Force (IETF) en febrero de 1996 como "draft-ietf-mmusic-sip-00". El borrador inicial no se parecía en nada a lo que conocemos como SIP ahora, y contenía solamente una sola petición: un requerimiento de seteo de llamada. En marzo de 1999, después de 11 revisiones, SIP RFC 2543 nació como protocolo.

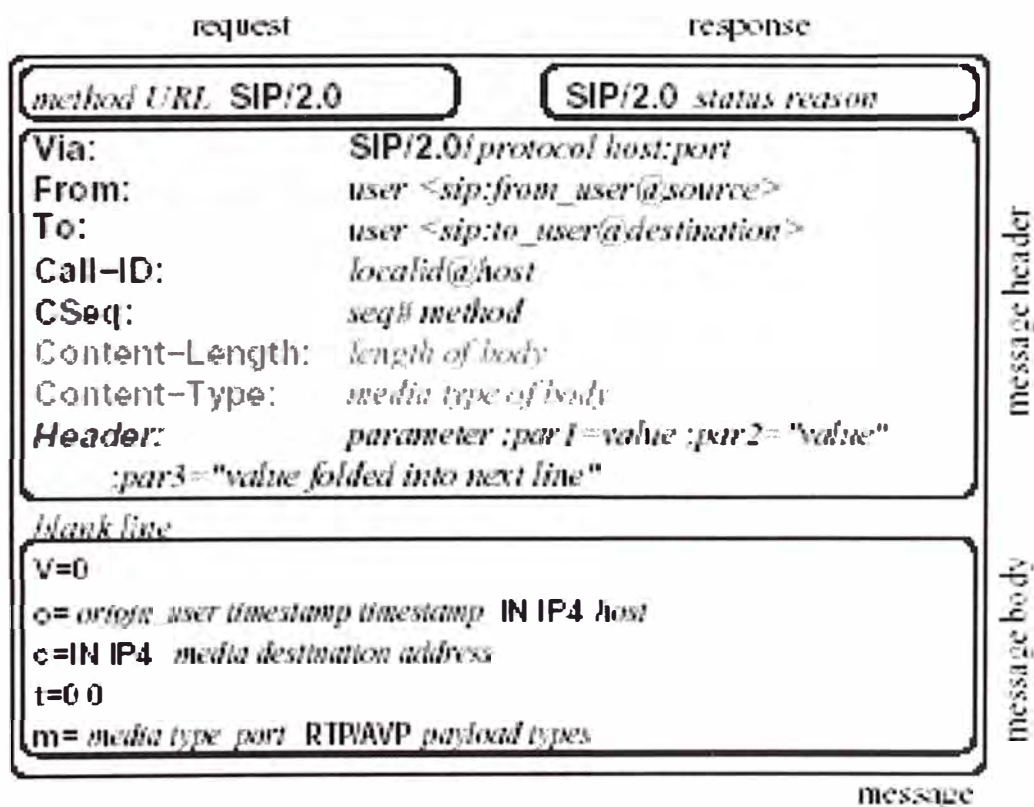


Figura No. 3.5. Estructura de un mensaje SIP.

Al principio, SIP fue ignorado, como H.323 era considerado el protocolo de elección para la negociación de transporte de VoIP. Sin embargo, creció su popularidad, en gran medida por su especificación de libre disponibilidad.

Una sesión SIP puede soportar desde una llamada telefónica hasta una multi-conferencia multimedia con elementos de colaboración. Está siendo desarrollado por el SIPWG del IETF (RFC 2543, 2543bis), con la misma filosofía de sencillez y mínimo esfuerzo de siempre. SIP está pensado como un mecanismo para el establecimiento, la terminación y la modificación de sesiones. Se trata de un protocolo basado en el paradigma de petición/respuesta (request-response), al igual que HTTP o SMTP.

En la Figura 3.5 se ilustra un mensaje tipo, con los campos más importantes de la cabecera y el cuerpo rellenos de forma genérica.

SIP maneja mensajes de petición: [que se estructuran en tres bloques] Request Line + Cabecera + Cuerpo, y mensajes de respuesta: Status Line + Cabecera + Cuerpo. En ambos casos el cuerpo es independiente de SIP y puede contener cualquier cosa. A efectos de estandarización se definen métodos para describir las áreas de especificación; SIP define los siguientes métodos: invite, bye, options, ack, register, cancel, info (rfc 2976), comet, prack, subscribe/, notify/, message.

Los mecanismos SIP necesarios para ofrecer una serie de servicios se pueden apreciar en la figura siguiente:

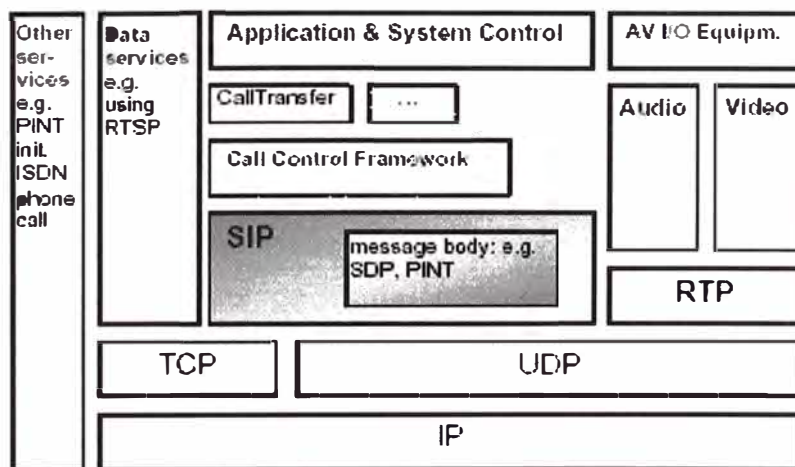


Figura No. 3.6. Corte transversal de las capas OSI para observar el protocolo SIP.

SIP se ha ganado un lugar como el protocolo estándar de VoIP. Todos los productos de usuario y corporativos tienen por defecto soporte SIP, cualquier producto existente sería una venta difícil a menos que un camino de migración a SIP sea ofrecido. Se espera que SIP provea muchas más capacidades que solamente VoIP, incluyendo la habilidad de transmitir video, música, y todo tipo de multimedia de tiempo real. SIP está posicionado para entregar la mayoría de nuevas aplicaciones en los siguientes años.

Probablemente, el más grande desafío técnico que SIP tiene que conquistar es el de llevar transacciones a través de una capa NAT. Debido a que SIP encapsula información de direcciones en sus flujos de datos, y NAT sucede a una capa de red mucho menor, la dirección de información no es modificada, y entonces, los media streams no tendrán la información correcta de dirección necesaria para completar la conexión cuando NAT está presente. Además, los firewalls normalmente integrados con NAT no consideran el flujo de media entrante como parte de la transacción SIP, y como es lógico, bloquean dicha conexión.

3.5.3 H.323

Este protocolo de Unión de Telecomunicación Internacional (ITU) fue originalmente designado para proveer un mecanismo de transporte IP para video-conferencia. Se convirtió luego en el estándar de equipos IP basados en video-conferencia, y brevemente tuvo la fama como el protocolo de VoIP también. Mientras hay mucho debate sobre si SIP o H.323 (o IAX) dominarán el mundo de protocolo de VoIP, en Asterisk, H.323 ha quedado largamente obsoleto a favor de IAX y SIP. H.323 no tiene mucho éxito entre usuarios o corporaciones, sin embargo aún sigue siendo el protocolo de VoIP más usado entre proveedores.

En una arquitectura H.323 (como la que se muestra en la Figura 3.7) se integran como componentes básicos los Terminales, Gateways (para interconexión con recursos PSTN/IN), Gatekeepers (Control de admisión, registro y ancho de banda) y MCUs (Multiconference Control Units).

Las dos versiones de H.323 soportadas en Asterisk son manejadas por los módulos `chan_h323.so` (suministrado con Asterisk), y `chan_oh323.so` (disponible como libre add-on).

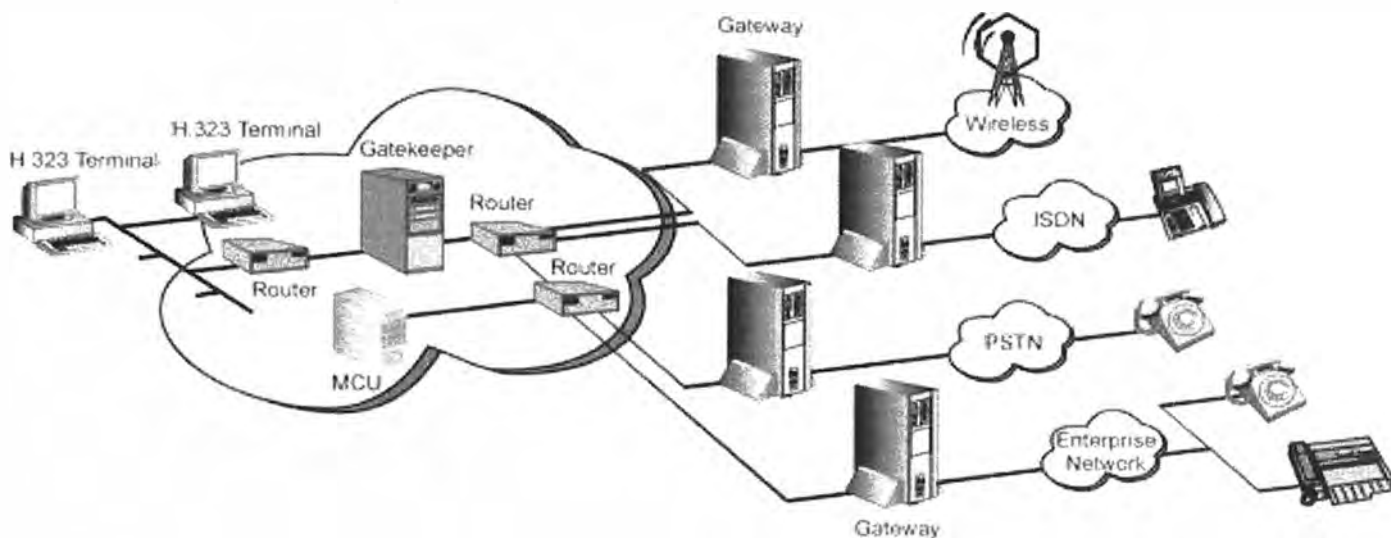


Figura No. 3.7. Arquitectura de una red H.323.

H.323 fue desarrollado por la ITU en mayo de 1996 como un medio de transmisión de voz, video, datos y comunicaciones fax a través de una red IP manteniendo la conectividad con la PSTN. Desde ese tiempo, H.323 ha evolucionado a través de versiones y anexos (las cuales han añadido funcionalidad al protocolo), permitiéndolo operar en redes de VoIP puras y más amplias redes distribuidas.

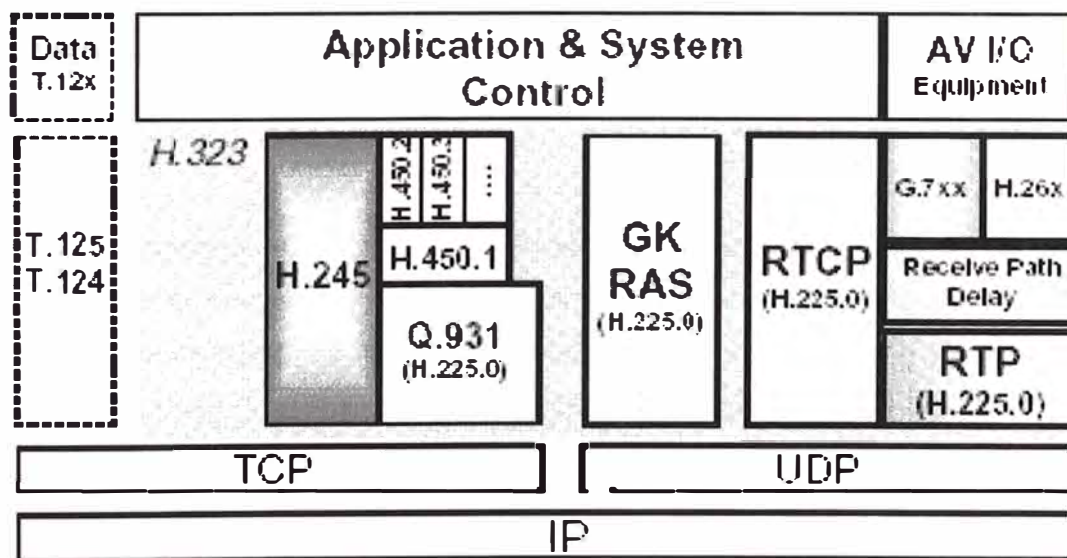


Figura No. 3.8. Pila de protocolos H.323.

Dentro de H.323 se incluyen todo un conjunto de protocolos perfectamente integrados (en la Figura 3.8 se ilustra la pila de protocolos H.323) que toman parte en el establecimiento y mantenimiento de conferencias multimedia: Q.931 para el establecimiento de llamada,

H.225 para la señalización, H.245 para la negociación de capacidades y el establecimiento de canales, H.450.x para la definición de servicios suplementarios (Call Park, Call Pickup, Call Hold, Call Transfer, Call Diversion, MWI, ...), RAS para el registro de terminales y el control de admisión, RTP/RTCP para el transporte y secuenciación de los flujos multimedia, G.711/G.712 para la especificación de los codecs, T.120 para colaboración y "data-conferencia"... Esto da una idea muy clara de una de las características menos agradables de este protocolo, y que siempre han argumentado sus detractores: su excesiva complejidad, frente a la sencillez del modelo Internet en que se basa SIP. De hecho SIP se podría comparar, grosso modo, con las partes de Q.931 y H.225 de H.323.

El futuro de H.323 está sujeto a debate. H.323 es comúnmente referido como técnicamente superior a SIP, pero como muchas otras tecnologías, eso últimamente puede no importar. Uno de los factores que hace a H.323 impopular es su complejidad, aunque muchos pueden discutir que el una vez simple SIP está comenzando a sufrir el mismo problema.

H.323 aún lleva de lejos la mayoría de tráfico de VoIP en los proveedores, pero las empresas están comenzando a ser menos dependientes de los proveedores tradicionales de telecomunicaciones. El futuro de H.323 se ve difícil de predecir con certeza. Mientras H.323 puede no ser el protocolo de preferencia para nuevas implementaciones, podemos con certeza esperar tener que tratar con interoperabilidad con proveedores H.323 por algún tiempo más.

El estándar H.323 usa el protocolo RTP para transporte de media entre los puntos terminales. Debido a esto, H.323 tiene los mismos problemas que SIP cuando se encuentra con redes que poseen topología que implementen NAT. El método más fácil es simplemente reenviar los puertos apropiados a través del dispositivo NAT al cliente interno.

Para recibir llamadas, se necesitara siempre reenviar TCP puerto 1720 al cliente. Además, se necesitará reenviar los puertos UDP por RTP media y RTCP control de flujos. Viejas aplicaciones clientes (como MS Netmeeting) también requerirán reenviar puertos TCP por H.245 tunneling.

Si se tiene un número de clientes detrás de un dispositivo NAT, se necesitará usar un gatekeeper corriendo en modo Proxy. El gatekeeper requerirá una interfaz enganchada a

la subred IP y la red pública Internet. EL cliente H.323 en la subred privada se registrará al gatekeeper, que hará las llamadas Proxy en vez del cliente. Igualmente, cualquier cliente externo que requiera llamar a la aplicación en la subred también requerirá registrarse con el servidor Proxy.

3.5.4 MGCP

El MGCP es, en esencia, un protocolo maestro/esclavo, donde se espera que los gateways ejecuten comandos enviados por el MGC. El Protocolo de Control de Media Gateway (MGCP) es usado para controlar los gateways de telefonía desde los elementos de control de llamadas externos llamados Media Gateways Controllers (MGC) o Gatekeepers.

Un gateway de telefonía es un elemento de red que provee conversión entre las señales de audio transportadas sobre los circuitos telefónicos y los paquetes de datos transportados sobre la internet o sobre otra red de paquetes.

MGCP asume una arquitectura de control de llamada, donde la inteligencia del control de la llamada está fuera de los gateways y manejada por un elemento de control de llamada externo. El MGCP asume que estos elementos de control de llamadas o MGC, se sincronizarán entre sí para enviar comandos coherentemente a los gateways que están bajo su control.

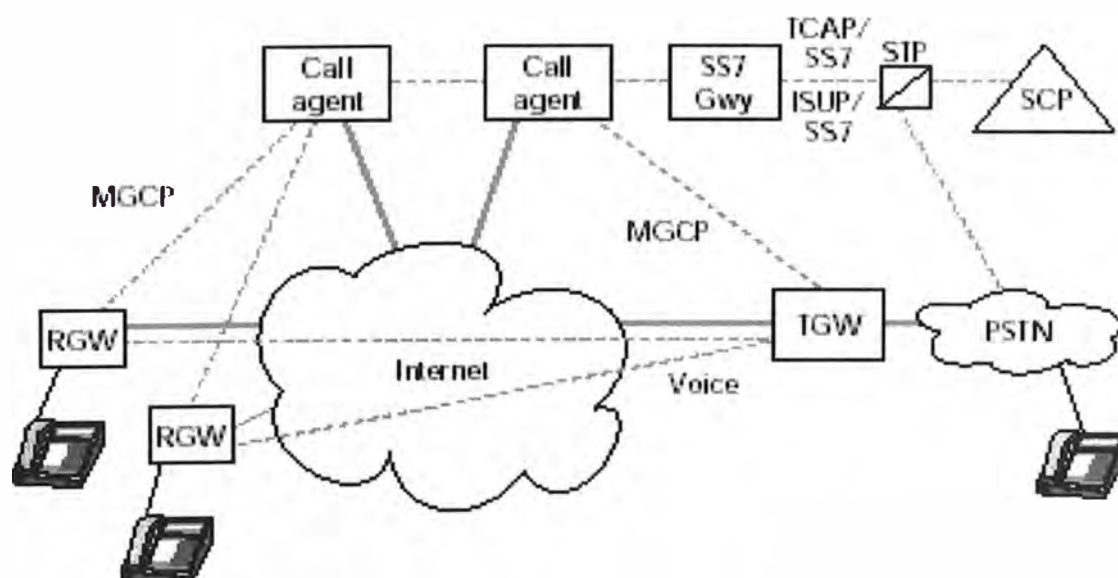


Figura No. 3.9. Integración MGCP-PSTN.

Lo que se propuso con MGCP fue sacar el control de la señalización del propio gateway (GW), llevándolo a otro elemento, el 'media gateway controller' MGC (que se conoce como 'softswitch') que se encargará del control de los media gateways'(MGW).

A nivel de sistemas lo que se ha hecho es desagregar el gatekeeper (GK) en sus equivalentes en el mundo SS7. Esta iniciativa surgió de varios fabricantes con el nombre de IPDC (Cisco, Alcatel, 3Com et al.) por un lado y SGCP (Telcordia) por otro; un esfuerzo que el IETF aglutinó bajo la denominación de MGCP y asignada a la responsabilidad del grupo de trabajo Megaco. MGCP es en la fecha de redacción de este documento un documento de trabajo. Tanto IETF como la ITU-T trabajan para llegar a un estándar, el primero bajo la responsabilidad de Megaco y como H.248 para el segundo.

En MGCP se puede decir que se ha separado la "inteligencia" (las funciones de control) de los datos (los contenidos: 'the media'). Que se trata de un protocolo Maestro/Esclavo. El maestro es el MGC ('softswitch' o 'call agent') y el esclavo es el MGW (que puede ser un GW de VoIP, un DSLAM, un router MPLS, un teléfono IP,...). Ésta es precisamente la característica que más choca con la filosofía (P2P) de SIP. Otra característica interesante es que intenta reproducir el modelo de la PSTN/IN sobre IP (en la Figura siguiente se ilustra el escenario típico para un despliegue tipo 'Internet Telephony' que es la aplicación para la que se pensó, al menos en principio esta solución), en contra del modelo distribuido que propone SIP.

En la Figura 3.9 se detalla un poco más lo que sería un escenario integrado con la PSTN, pensando en prestar el servicio de telefonía sobre Internet.

3.5.5 Jingle

Jingle es una extensión al protocolo Jabber/XMPP que permite la transferencia de información peer-to-peer (p2p). A través de este protocolo se pueden transmitir datos multimedia, permitiendo la adopción de servicios de Videoconferencia y de VoIP

Este protocolo fue diseñado inicialmente por Google junto con la Jabber Software Foundation y liberado (bajo licencia similar a la de BSD) tras la salida de Google Talk en 2006 para su uso en Jabber. Google Talk ya implementa este protocolo mientras que otros clientes Jabber como Psi o Jabbin están aún en proceso de implementación.

Asterisk soporta también esta tecnología en modo de pruebas, mientras que en la siguiente versión estable para sistemas en producción se espera que esté completamente disponible.

3.5.6 Protocolos Proprietarios:

Además de los protocolos estándares, hay dos protocolos propietarios soportados por Asterisk:

- **Skinny/SCCP:** El Protocolo de Control de Cliente Skinny (SCCP) es un protocolo propietario de equipos de Cisco VoIP. Es el protocolo por defecto para puntos terminales en un Cisco Call Manager PBX. Skinny es soportado por Asterisk, pero si se va a conectar teléfonos Cisco a Asterisk, es generalmente mejor si se obtiene las imágenes SIP para los teléfonos que los soportan y hacer la conexión por SIP.
- **UNISTIM:** Soporte para el protocolo propietario de VoIP de Nortel, UNISTIM, ha sido recientemente añadido a Asterisk. Esto significa que Asterisk es el primer PBX en la historia que soporta nativamente terminales IP propietarios con dos de las más grandes empresas de VoIP, Nortel y Cisco.

3.6 Tecnologías Soportadas para Codificación/Decodificación de voz

En lo que se refiere al mundo de los Codecs (Codificador/Decodificador), tenemos muchos soportados por Asterisk.

Los Codecs son entendidos generalmente como varios modelos matemáticos usados para codificar (y comprimir) digitalmente información de audio analógico. Muchos de estos modelos toman en consideración la habilidad del cerebro humano en formar una impresión incompleta de información.

Hemos visto todo tipo de ilusiones ópticas; de la misma manera, los algoritmos de compresión de voz toman ventaja de nuestra tendencia a interpretar lo que creemos que deberíamos oír, en lugar de lo que realmente escuchamos. El propósito de los distintos tipos de algoritmos de codificación de voz es para balancear la eficiencia y calidad de la voz. Veamos los tipos de codecs, el data bitrate y la licencia requerida para usarlos:

Tabla No. 3.1. Comparación de diversos tipos de CODECS.

Codec	Data bitrate (kbps)	Licencia requerida?
G.711	64 kbps	No
G.726	16, 24, o 32 kbps	No
G.723.1	5.3 o 6.3 kbps	Yes (no for passthrough)
G.729A	8 kbps	Yes (no for passthrough)
GSM	13 kbps	No
iLBC	13.3 kbps (30-ms frames) o 15.2 kbps (20-ms frames)	No
Speex	Variable (entre 2.15 y 22.4 kbps)	No

Originalmente el término CODEC se refería a COder/DECoder: un dispositivo que convertía entre señal analógica y digital. Ahora el término parece relacionarse más a COmpresión/DECompresión.

3.6.1 G.711

La ITU ha estandarizado la Modulación de Código de Pulso como G.711, permite una señal de audio de calidad tarifada con un ancho de banda de 3.4 KHz que ha de ser codificado para la transmisión de índices de 56 Kbps o 64 Kbps.

El G.711 utiliza A-law o μ -law para una compresión simple de amplitud y es el requisito básico de la mayoría de los estándares de comunicación multimedia de la ITU.

PCM es un método de codificación de señal de audio analógica más popular y es ampliamente utilizado por la red telefónica pública. Sin embargo, el PCM no soporta compresión de ancho de banda, por lo que otras técnicas de codificación como el ADPCM utilizan estimaciones basándose en dos muestras cuantificadas consecutivas para reducir el ancho de banda.

3.6.2 G.726

Este codec ha estado en el mercado por algún tiempo (Es usado en lugar de G.721 que está ahora obsoleto), y es uno de los codecs comprimidos originales. Es conocido como Modulación por Código de Pulso Adaptivo Diferencial (ADPCM), y puede correr a

diferentes bitrates. Los más comunes son 16Kbps, 24Kbps y 32 Kbps. Asterisk actualmente soporta solo el ADPCM-32 rate, el cual es de lejos el más popular bitrate para este codec.

G.726 ofrece calidad idéntica al G.711, pero usa solo la mitad de ancho de banda. Esto es posible porque en lugar de enviar el resultado de la medida de cuantización, envía solamente información suficiente para describir la diferencia entre la muestra actual y la anterior.

G.726 cayó en popularidad a mediados de los 90s por su inhabilidad de llevar señales MODEM y fax, pero a causa de su rendimiento ancho de banda/CPU está haciendo un regreso. G.726 es especialmente atractivo porque no requiere mucho trabajo de CPU en el sistema.

3.6.3 G.723.1

G.723.1 define cómo puede codificarse una señal de audio con un ancho de banda de 3.4 KHz para transmitirse a dos bit rates:

- 6.3 kbit/s (usando 24 byte chunks) usando un algoritmo MPC-MLQ.
- 5.3 kbit/s (usando 20 byte chunks) usando un algoritmo ACELP.

Requiere un índice de transmisión muy bajo ofreciendo una calidad de audio cercana a la tarifada. G.723.1 ha sido seleccionada por el VoIP Forum como el codec básico para aplicaciones de telefonía IP de bajo índice de bits.

El codificador de habla G.723.1 opera con tramas de 30ms de señales de habla en ancho de banda de teléfono digitalizadas y de muestreo a 8 kHz. Las tramas se dividen en cuatro subtramas de 7.5ms de 60 muestras cada una. Cada trama con 240 muestras de entrada se transforma en una palabra de 12 16 bits de datos comprimidos a alta velocidad o palabras de 10 16 bits de datos comprimidos a baja velocidad.

Las Detección de Actividad de Voz/Generación de Ruido Confortable (Voice Activity Detection/Comfort Noise Generation o VAD/CNG) especificado en el Anexo A se incorporan por completo al ITU-T G.723.1.

G.723.1 es uno de los codecs requeridos para ser compatibles con H.323, aunque otros codecs pueden ser también empleados con H.323.

Está protegido con patentes y por consiguiente requiere licencia si se va a hacer un uso para aplicaciones comerciales. Lo que significa que mientras se puede conmutar entre dos llamadas G.723.1 a través del sistema Asterisk, no está permitido decodificarlas sin una licencia.

3.6.4 G.729A

Considerando el poco ancho de banda que usa, G.729A entrega una impresionante calidad de sonido. Lo hace a través del uso de Algoritmos Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP). A causa de las patentes, no se puede usar G.729A sin una licencia; sin embargo, es extremadamente popular y es soportado por una amplia variedad de teléfonos y sistemas.

Para lograr este impresionante ratio de compresión, este codec requiere un esfuerzo del CPU igual de impresionante. En un sistema Asterisk, el uso de codecs de compresión pesada fuerza hacia abajo la potencia del CPU.

G.729A usa 8Kbps de ancho de banda.

3.6.5 GSM

Este codec no viene con requerimientos de licencia en la manera que lo hacen G.723.1 y G.729A y ofrece rendimiento sobresaliente con respecto a la demanda que pone en el CPU. La calidad de sonido es generalmente considerada de menor grado que la producida por G.729A, pero sin embargo ofrece buenos resultados.

El codec GSM opera a 13Kbps.

3.6.6 iLBC

El Internet Low Bitrate Codec (iLBC) provee una mezcla atractiva de bajo ancho de banda y calidad en enlaces de red de poco ancho de banda.

Naturalmente, Asterisk lo soporta, pero no es tan popular como los ITU codecs y por lo tanto puede no ser compatible con teléfonos IP comunes y sistemas de VoIP comerciales. iLBC está dentro de los estándares de IETF, con los RFC 3951 y 3952.

Debido a que iLBC usa algoritmos complejos para obtener sus altos niveles de compresión, necesita un alto costo de CPU en Asterisk.

iLBC opera a 13.3Kbps (30-ms cuadros) y 15.2kbps (20-ms cuadros).

3.6.7 Speex

Speex es un codec de bitrate variable (VBR), que significa que es capaz de modificar dinámicamente su bitrate en respuesta a los cambios en las condiciones de red. Se ofrece en versiones para redes de banda ancha y corta, dependiendo en la calidad del teléfono.

Speex es un codec totalmente libre, licenciado por Xiph.org, variante de la licencia BSD.

Speex opera en cualquier frecuencia dentro de 2.15 a 22.4kbps, debido a su bitrate variable.

3.6.8 MP3

MP3 es también un codec. Específicamente, se usa en Asterisk para Música de Espera (Music On Hold - MoH). MP3 no es un codec de telefonía, porque está optimizado para música, no voz; sin embargo, es muy popular con los sistemas de VoIP como un método para enviar Música de Espera.

3.7 Implementación del Sistema propuesto.

Para la implementación de un Sistema Asterisk, es necesario tener en cuenta varias variables que deben de asegurar la calidad y rendimiento del sistema. Veamos más de cerca estas variables:

3.7.1 Calidad de Servicio

Calidad de Servicio, o QoS, como es nombrado más popularmente, se refiere al desafío para proveer flujo de datos sensitivo en el tiempo a través de una red que fue diseñada para transmitir datos de manera ad-doc, con el mejor esfuerzo. Aunque no haya reglas absolutas establecidas, se acepta generalmente que si uno puede entregar sonido producido por el parlante al oído de la persona que escucha dentro de 300 milisegundos, un flujo normal de conversación es posible. Cuando el retardo excede 500 ms., éste se torna difícil de evitar la interrupción de cualquiera de los lados. Fuera de un segundo, la conversación normal se torna extremadamente mala.

Además de llevar el paquete a su destino a tiempo, es también esencial asegurar que la información transmitida llegue intacta. Muchos paquetes perdidos pueden prevenir que el lado opuesto pueda completar de reproducir la muestra de audio, y los huecos en el audio se escucharán como estáticos, y en severos casos, palabras completas u oraciones perdidas.

a) Protocolos de transporte de paquetes: TCP, UDP y SCTP

- **TCP:** El Protocolo de Control de Transmisión (TCP) casi nunca es usado para VoIP. Dado que tiene mecanismos para asegurar el envío, no tiene ninguna prisa para hacerlo. A menos que haya una extremada baja latencia de interconexión entre los dos lados, TCP va a tender a causar más problemas que resolverlos.

El propósito de TCP es garantizar el envío de paquetes. Para lograr eso, han sido implementados muchos mecanismos, tales como numeración de paquetes (para reconstruir bloques de datos), envío de acuse de recibos, y reenvíos de muchos paquetes. En el mundo de VoIP, llevar los paquetes al lado opuesto de la línea es crítico, pero los 20 años de redes celulares nos ha enseñado a tolerar un poco la pérdida de paquetes.

El alto grado de carga de procesamiento de TCP, el manejo de estados, y el trabajo del acuse de recibo funcionan bien para transmitir largas cantidades de datos, pero simplemente es ineficiente para comunicaciones de tiempo real.

- **UDP:** A diferencia de TCP, el Protocolo de Datagrama de Usuario (UDP) no ofrece ninguna garantía de envío de paquetes. Los paquetes son colocados en el cable tan pronto como sea posible y soltados en el mundo para que encuentren su camino a sus destinos finales, sin ninguna confirmación si es que llegaron a su destino o no.

Como UDP no ofrece ninguna garantía para la llegada de paquetes, adquiere una alta eficiencia al gastar muy poco esfuerzo en lo que está transportando.

- **SCTP:** Aprobado por el IETF como un estándar en RFC 2960, EL Protocolo de Transmisión de Control de Flujo (SCTP) es relativamente un

nuevo protocolo de transporte. Se diseñó para acabar con las deficiencias de TCP y UDP, especialmente como relación a los tipos de servicios que se tienen que entregar sobre redes de telefonía de circuitos conmutados.

Algunos de los objetivos de SCTP fueron:

Mejores técnicas de congestión-evasión (específicamente, evitando ataques de negación de servicio)

Estricta secuencia de envío de datos

Baja latencia para transmisiones mejoradas en tiempo real.

Mejorando las deficiencias de TCP y UDP, los desarrolladores de SCTP esperaban crear un protocolo robusto para la transmisión de SS7 y otros tipos de señalización PSTN sobre una red basada en IP.

b) Diferenciación de Servicio

La diferenciación de servicio, o también llamado DiffServ, no es mucho un mecanismo QoS como un método por el cual el tráfico puede ser marcado y dado un tratamiento específico. Obviamente, DiffServ puede ayudar a proveer QoS permitiendo a ciertos tipos de paquetes tomar precedencia sobre otros. Mientras esto aumentaría la opción de que un paquete de VoIP pase rápidamente a través de un enlace, no garantiza nada.

c) Garantía de Servicio

La última garantía de QoS es proporcionada por la PSTN. Para cada conversación, un canal de 64Kbps es completamente dedicado a la llamada, teniendo un ancho de banda garantizado. Similarmente, los protocolos que ofrecen servicio garantizado pueden asegurar que una cantidad requerida de ancho de banda es dedicada a la conexión a ser servida. Como cualquier tecnología de red de paquetes, estos mecanismos generalmente operan mejor cuando el tráfico es mejor a los niveles máximos. Cuando una conexión se aproxima a sus límites, es casi imposible eliminar la degradación.

- **MPLS:** Conmutación de Etiqueta Multiprotocolo (Multiprotocol Label Switching - MPLS) es un método para la ingeniería de patrones de tráfico de red independientes de capa 3 de tablas de ruteo. El protocolo trabaja

asignando pequeñas etiquetas (MPLS frames) a los paquetes de red, y últimamente a sus destinos finales. Tradicionalmente, los ruteadores hacen una decisión independiente basada en el IP lookup table en cada salto en la red.

En una red MPLS, esta búsqueda es hecha una vez, cuando el paquete entra a la nube MPLS al ingreso del ruteador. El paquete es asignado a un stream, referido a una etiqueta Label Switched Path (LSP), e identificado por una etiqueta. La etiqueta es usada como búsqueda en un índice en la tabla MPLS, y el paquete transversa el LSP, independientemente de las decisiones de ruteo de la capa 3. Esto permite a los administradores de largas redes calibrar mejor las decisiones de ruteo y hacer mejor uso de sus recursos de red. Adicionalmente, la información puede ser asociada con una etiqueta que priorice el reenvío de paquetes.

- **Mejor Esfuerzo:** La más simple, y menos costosa aproximación a QoS es no proveerla. El método de "Mejor Esfuerzo". Mientras esto pueda sonar una mala idea, puede en realidad funcionar bien. Cualquier llamada VoIP que sale a la red pública Internet es casi seguro que hará su mejor esfuerzo, ya que los mecanismos de QoS aún no están en este entorno.

3.7.2 Eco

El "Eco" ha sido un problema en el PSTN tan antiguo como los teléfonos. La industria de las telecomunicaciones ha invertido grandes cantidades de dinero en diseñar dispositivos caros de cancelación de eco.

El "Eco" ocurre en el mundo analógico. Cuando se escucha eco, no es el teléfono el que está causando el problema; es el extremo alejado del circuito. Conversacionalmente, el eco escuchado en el extremo opuesto es generado en este extremo. El eco es causado por el hecho que un circuito de lazo local analógico tiene que transmitir y recibir en el mismo par de cables. Si el circuito no está eléctricamente balanceado, o si es que hay un teléfono de baja calidad conectado en el otro extremo del circuito (desacople de impedancias), las señales que recibe pueden ser reflejadas de regreso, y de esa manera uno podría volver a escuchar las palabras que pronunció momentos atrás.

El oído humano puede percibir un eco después de un retardo de aproximadamente 40 milisegundos.

En un teléfono barato, es posible que el eco se genere en el cuerpo del handset. Esta es la razón por la que algunos teléfonos IP baratos pueden causar eco, aún cuando la conexión entera no contiene ningún circuito analógico. En realidad el handset de cualquier teléfono, sea tradicional o de VoIP es un circuito analógico.

En el mundo de VoIP, el eco es usualmente introducido por un circuito analógico en algún lugar de la conexión, o por terminales baratos que reflejan atrás la señal (por ejemplo, a través de handset de manos libres pobremente diseñados). Una regla de experiencia es mantener la latencia por menos de 250 milisegundos.

a) Controlador de Eco en Asterisk

En el archivo de configuración `zconfig.h`, se puede escoger de los muchos algoritmos de cancelación de eco, con el `MARK2` establecido por defecto. Experimentación con los muchos canceladores de eco de la red determinará el mejor para el entorno. Asterisk también tiene opción para hacer la cancelación de eco más agresiva. Se puede habilitar en la siguiente línea del archivo antes mencionado:

```
#define AGGRESSIVE_SUPPRESSOR
```

Una cancelación de eco agresiva, puede crear un walkie-talkie, con un efecto half-duplex. Esto debería ser habilitado solo si otros métodos de reducción de eco han fallado.

Cuando una cancelación de eco ha sido habilitada, el cancelador de eco aprende del eco en la línea escuchándolo durante la duración de la llamada. Consecuentemente, el eco puede ser escuchado al principio de la conversación, atenuándose luego de un periodo de tiempo. Para evitar esta situación, se puede emplear un método llamado "echo training", el cual enmudecerá la línea brevemente en el principio de la llamada, y luego enviará un tono con el cual la cantidad de eco de la línea puede ser determinada. Esto permite a Asterisk tratar con el eco más rápidamente. El "echo training" puede ser habilitado en Asterisk con la siguiente instrucción:

```
echotraining=yes.
```

3.8 Caso de Estudio: Instalación y puesta en Marcha de un Servidor PBX en IT-Synergy para el servicio de Interconexión de la Red PBX de las oficinas del PNUD de Medio Oriente

El sistema que se mostrará a continuación fue desarrollado en un proyecto para la interconexión de los PBX de las oficinas del Programa de las Naciones Unidas para el Desarrollo (PNUD) de Medio Oriente.

El proyecto a analizar, tenía como principal objetivo la interconexión, mediante una red de Voz sobre IP, todos los PBX de las oficinas del PNUD que se encuentran en diferentes países del Medio Oriente, con la finalidad de reducir los costos generados por las continuas llamadas internacionales que se hacen a diario entre esas oficinas.

Las oficinas de las Naciones Unidas de los siguientes países estaban incluidas en el proyecto:

- UNDP Cairo, Egipto;
- UNDP Nueva York, United States;
- UNDP Riyadh, Arabia Saudita;
- UNDP Tunis, Tunes;
- UNDP Trípoli, Libia;
- UNDP Jartoum, Sudan;
- UNDP Abu Dhabi, Emiratos Arabes Unidos;
- UNDP Rabat, Marruecos;
- UNDP Beirut, Líbano;
- UNDP Amman, Jordania;
- UNDP Yemen, Yemen;
- UNDP Bahrein, Dubai;
- IT-Synergy Headquarters - Cairo, Egipto.

Además de la interconexión de los PBX de las oficinas de las Naciones Unidas en los países antes mencionados, también se necesitaba ofrecer el servicio de extensiones u anexos móviles que puedan ser usados por determinadas personas que se encuentran en constante movimiento.

Para la implementación del sistema, se planeó usar Asterisk como servidor PBX para brindar todos los beneficios de un PBX a cada uno de los anexos, incluyendo brindar extensiones o anexos remotos (oficinas remotas).

3.8.1 Interconexión de los PBXs

Para la implementación de la interconexión de los PBXs, se utilizaron los productos de telefonía MultiVoIP de Multitech (<http://www.multitech.com>). Estos dispositivos funcionan como VoIP gateway entre los PBX y la red de datos, y proveen comunicaciones de voz y fax sobre la Internet o intranet. Integrando la voz y fax dentro la red de datos existente se pueden realizar sustanciosos ahorros en las llamadas a larga distancia.

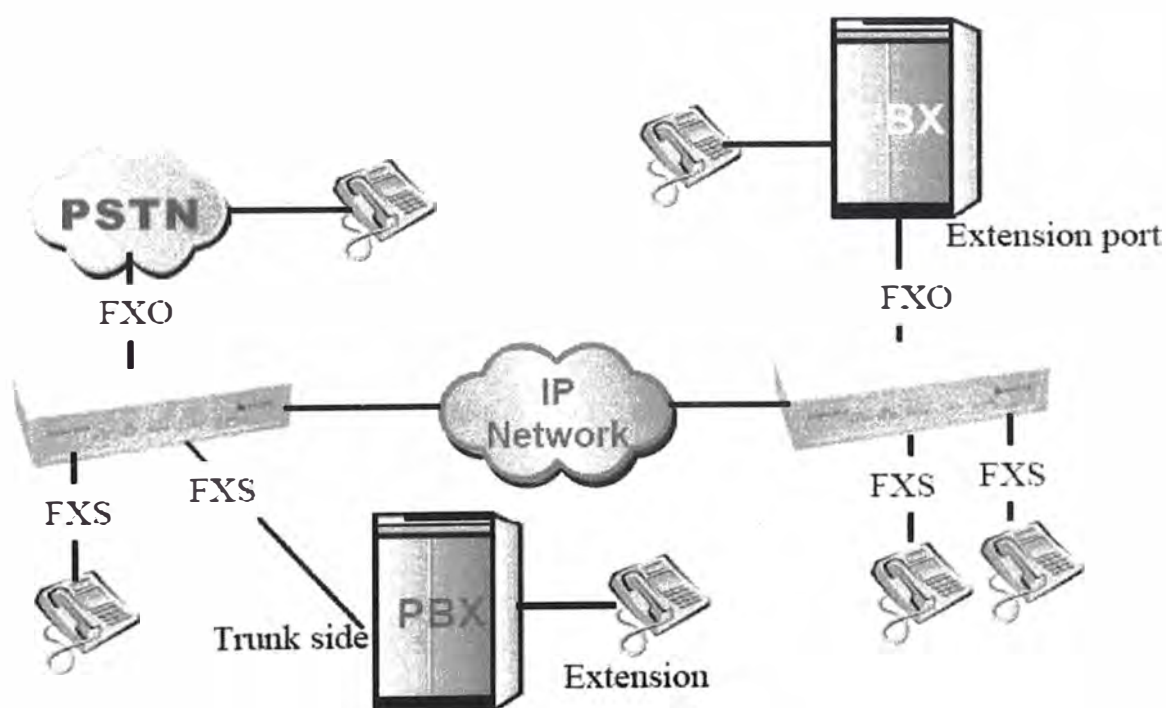


Figura No. 3.10. Interconexión de PBX remotos.

La figura 3.10 nos muestra la conexión planeada por el lado de las oficinas de UNDP, donde cada dispositivo MultiVOIP se encargaría de hacer el gateway entre los PBX de dichas oficinas y la red de datos con su proveedor.

De esta manera todas las oficinas de UNDP tendrían un dispositivo MultiVOIP que pueden ser usados para interconectar todos los PBX. Este es el primer paso para la interconexión.

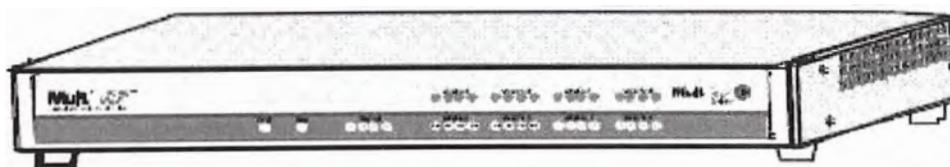
El modelo de dispositivo usado en las oficinas difería dependiendo de las necesidades explícitas de cada una de ellas. De acuerdo a las experiencias es a veces mejor conectar el dispositivo MultiVOIP como troncal que como anexo. Dependiendo de si es que se disponía de un técnico en cada país que pueda acceder directamente a la configuración del PBX local (cuya marca difería dependiendo del país) se conectaba como anexo o extensión FXO.

Los dispositivos MultiVOIP que se usaron en los distintos países fueron escogidos debido a las necesidades de contar con uno, dos o más canales de voz, que pudieran ser usados simultáneamente. Estos casos fueron analizados por separado, llegándose a usar los siguientes modelos:

- MVP810: Posee 8 puertos VOIP gateway (FXO/FXS)
- MVP410: Posee 4 puertos VOIP gateway (FXO/FXS)
- MVP210: Posee 2 puertos VOIP gateway (FXO/FXS)
- MVP130: Posee 1 puerto VOIP gateway (FXO/FXS)

Esas configuraciones dependían del tráfico previsto que habría en cada oficina.

La ventaja del uso de este tipo de dispositivos es que son fácilmente montables en los racks, gracias a lo pequeño de su diseño, como se puede apreciar en la siguiente figura:



MVP-410/810 Chassis



MVP-210 Chassis

Figura No. 3.11. Modelos de los dispositivos MultiVOIP de Multitech.

Mecánicamente, los MVP410 y MVP810 MultiVOIP son diseñados para un rack estándar industrial EIA de 19 pulgadas de cerrado. En contraste, los MVP130 y MVP210 son unidades de mesa pequeñas.

Estos dispositivos MultiVOIPs inter-operan con un switch telefónico PBX, actuando como dispositivos conmutadores que direccionan las llamadas de voz y fax sobre la red IP. Los MultiVOIPs tienen "Libros Telefónicos", que son directorios que determinan a quienes las llamadas serán hechas y las secuencias que deben de ser usadas para completar las llamadas a través del MultiVOIP. Los directorios o libros telefónicos permiten al usuario interactuar con el sistema de VoIP como si lo harían con un PBX ordinario o Switch Telco. Cuando los directorios telefónicos están configurados, las secuencias de marcado especial son minimizadas o eliminadas todas juntas. Cuando el destino de la llamada ha sido determinado, la configuración del directorio de llamadas determinan si el MultiVOIP destino debe de añadir o quitar dígitos para dar la impresión que el destino de la llamada es local.

Adicionalmente, estos dispositivos MultiVOIP soportan los protocolos de transmisión de voz H.323, SIP y SPP.

Los protocolos H.323 y SIP han sido descritos anteriormente. El protocolo SPP (Single Port Protocol) es un protocolo propietario (diseñado y desarrollado por Multitech) que ofrece ventajas de uso en ciertas situaciones, especialmente cuando se usan firewalls y cuando se necesita asignar un IP dinámico a las unidades antes mencionadas. Sin embargo, cuando se usa SPP, ciertas características del SIP y H.323 pueden no estar disponibles y SPP no podría operar con sistemas de Voz sobre IP que usen H.323 o SIP.

Los dispositivos MultiVOIP fueron conectados a las redes locales de las respectivas oficinas asignándoles un IP público, para que puedan ser interconectadas entre ellas.

3.8.2 Interconexión con Asterisk

Debido a que los dispositivos MultiVOIP soportan el protocolo SIP, la interconexión con Asterisk es posible, sirviendo los primeros como clientes del servidor PBX general (Asterisk).

Para la implementación del servidor Asterisk Open Source, se contó con un servidor Intel Pentium IV 3.4MHz con 1GB RAM, Sistema Operativo Linux Mandriva 2005, con una

tarjeta analógica Digium TDM400P, cuyas características han sido descritas anteriormente en el Capítulo 4.2.1. Esta tarjeta contaba con dos módulos FXS y dos FXO.

La razón del uso de esta tarjeta es que este servidor estaría situado en los Headquarters de IT-Synergy, que cuenta con dos líneas troncales analógicas, que han sido conectadas a los puertos FXO del servidor PBX (Asterisk). Uno de los puertos de entrada FXO se usa para recibir llamadas, mientras que el otro se usa para transmitir y recibir faxes. Igualmente, un puerto de salida FXS que viene de la tarjeta ha sido conectado directamente a la máquina de fax, mientras que el otro se usa como terminal de recepción. La razón de esta elección es que si por algún motivo haya una desconexión de la red local con Internet y todos los teléfonos SIP dejaran de funcionar, el teléfono en recepción y el fax aun estarían funcionando sin ningún problema y se podría usar para casos de emergencia.

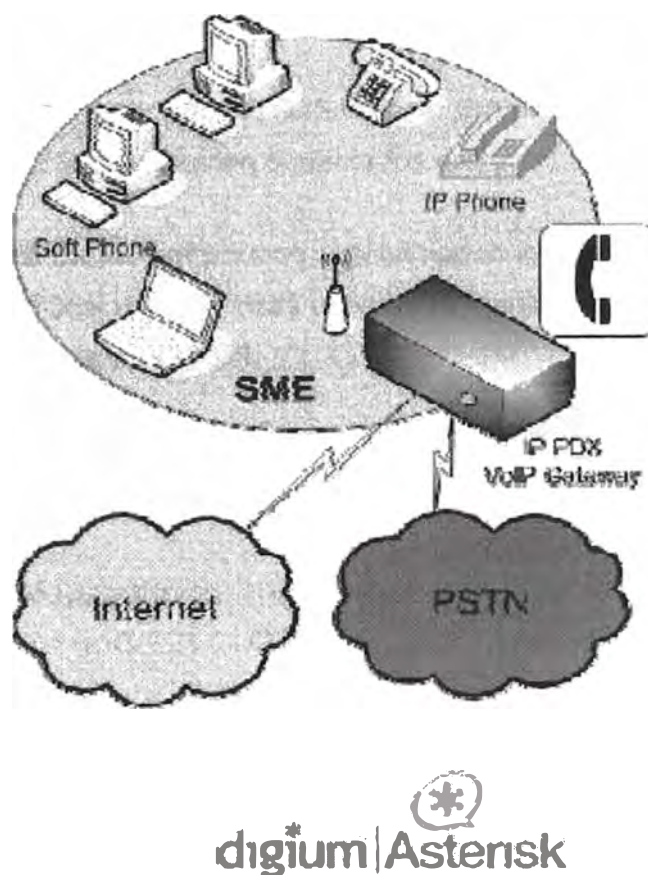


Figura No. 3.12. Estructura de conexión de Asterisk en una LAN.

Los detalles de configuración de los puertos FXS/FXO de la tarjeta TDM400P se encuentran en el archivo `zaptel.conf` del servidor Asterisk sobre Linux. Este archivo se describe con más detalle en el **Apéndice A**. Igualmente, la configuración de la interfaz de

telefonía de la tarjeta Zapata con Asterisk para definir la configuración de cada canal se define en el archivo `zapata.conf`, que se describe en detalle en el **Apéndice B**.

De esta manera, en los Headquarters de IT Synergy, queda instalado el servidor Asterisk, conectado a la red PSTN de Cairo, y a la vez conectado a la red local a través de la interfaz ethernet. Un IP público fue asignado a esta máquina, de tal manera que pudiera ser visible desde cualquier punto conectado a la red Internet. Esto haría posible que las demás oficinas de UNDP tengan conexión directa al Servidor Asterisk simplemente con tener salida a Internet.

La figura 3.12 muestra cómo el servidor Asterisk está conectado a la vez a la red local, como a la red PSTN. Para lograr optimizar aún más el ancho de banda usado por los teléfonos locales de la oficina de IT Synergy, se separó la red local en dos segmentos clase C: una donde solo estén conectados los teléfonos IP (SIP Phones), y la otra donde estén las Workstations, servidores y otros.

Téngase en cuenta que el servidor Asterisk tiene dos tarjetas de red, haciendo NAT interno con los teléfonos SIP que están conectados en la red local interna.

Para el caso específico de los softphones (aplicaciones telefónicas que usan protocolos de VoIP), se usa la red local donde estará conectada la máquina, laptop o servidor desde donde se está ejecutando la aplicación.

El protocolo SIP fue usado en los teléfonos SIP y en los soft-phones debido a que está convirtiéndose en un estándar muy usado en la industria actualmente.

Adicionalmente, los codecs seleccionados para usarse en la transmisión de información vía Voz sobre IP, fueron el G.723.1, G.729A y GSM, además de u-law cuando no sea posible usar alguno de los otros. Preferiblemente es más conveniente usar los primeros, dado la gran cantidad de ancho de banda que ahorran. Sin embargo, si es que se realizan muchas comunicaciones simultáneamente, es necesario cambiar de codec, ya que el rendimiento del CPU es traído abajo por la carga que la compresión y descompresión de voz que generan los primeros.

Con el sistema funcionando, los dispositivos MultiVoIP se conectan como clientes SIP al servidor Asterisk. Esto permite que toda la red del PNUD esté centralizada en el servidor Asterisk que corre sobre Linux, haciéndolo parecer como si todas pertenecieran a una misma oficina, con sus respectivas extensiones.

De esta manera, es posible llamar desde, por ejemplo, Nueva York a un celular en Beirut, donde la llamada viaja desde Nueva York hasta Beirut vía Internet usando voz sobre IP, desde allí ingresa al PBX de Beirut mediante MultiVOIP, para próximamente salir hacia la red pública para ubicar el número de teléfono celular. Ver Figura 3.13 para apreciar los distintos tipos de conexión que pueden efectuarse.

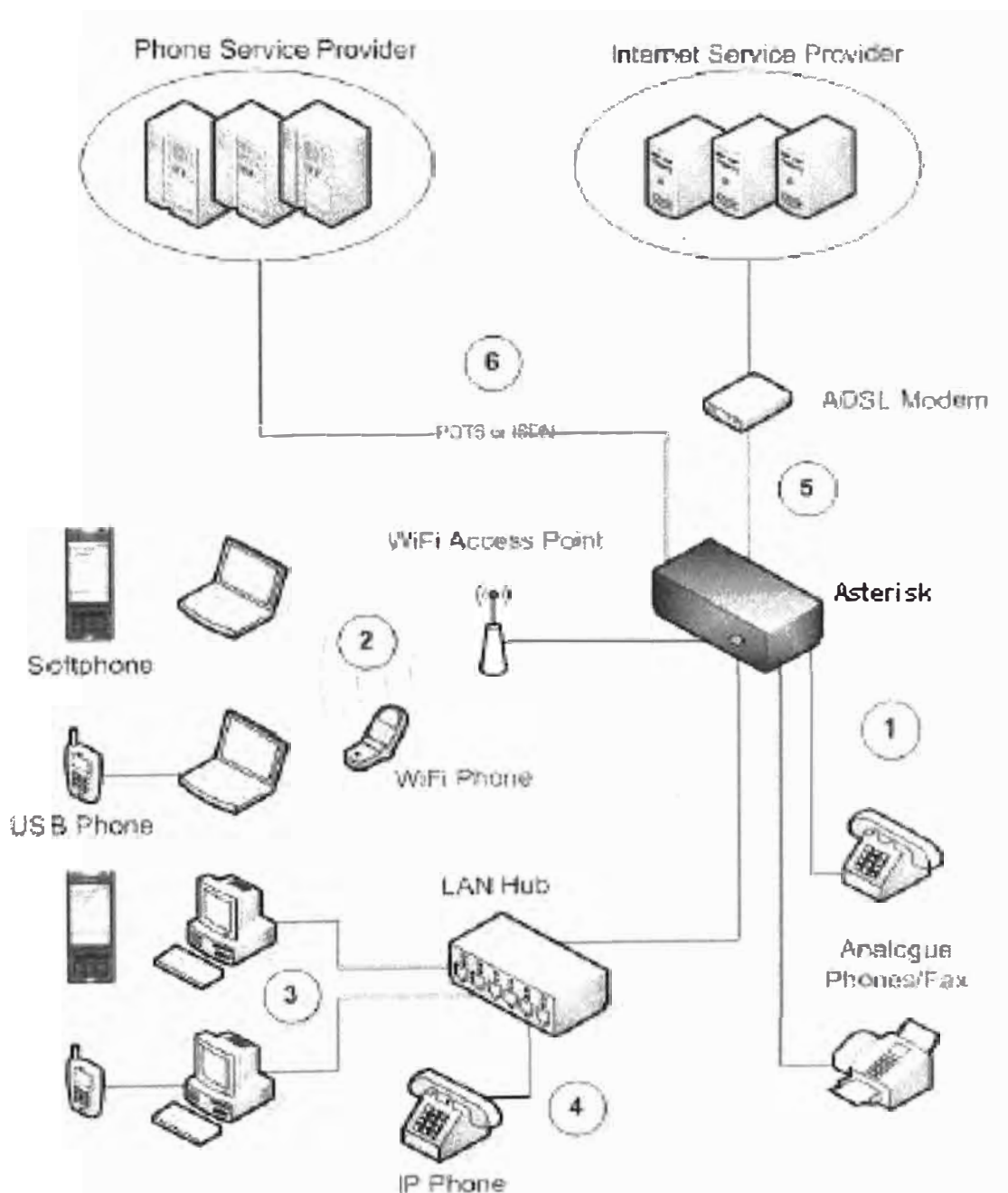


Figura No. 3.13. Conexión de diversos dispositivos de comunicación con el servidor Asterisk.

Las llamadas se producen de la misma manera sea cual sea la oficina del PNUD del país desde donde se llame, notándose claramente que una llamada internacional de Nueva

York a Beirut tiene el costo de una llamada local hecha en Beirut solamente, ya que el tramo Nueva York-Beirut ha sido completado usando Voz sobre IP.

Igualmente, Asterisk puede funcionar como cliente SIP, conectándose a un servidor SIP de algún proveedor de VoIP que provea interconexión con otros países a través de sus propios gateways, como Voicetrading (www.voicetrading.org). Esto nos proporciona una forma más de flexibilidad. Se puede configurar Asterisk de tal manera que marcando desde cualquier extensión, por ejemplo, el número 4444, se pueda tener acceso a la red del proveedor voicetrading, y desde allí hacer llamadas usando VoIP a otros países que no están dentro de la red del PNUD. El total de las llamadas internacionales se maneja a través de VoIP, usándose para los países de la red del PNUD de medio oriente la propia red del PNUD, y para cualquier otro país la red del proveedor de VoIP Voicetrading. La razón para esta separación es simplemente por motivos de seguridad, ya que las llamadas a los países que están dentro de la red del PNUD son ruteadas dentro de la misma red. Solamente para llamadas a países externos a la red, se rutean las llamadas por el proveedor de VoIP Voicetrading.

La configuración de los clientes y servidores SIP en Asterisk se realiza en el archivo sip.conf. Detalles de la configuración de este archivo se pueden encontrar en el **Apéndice C**.

SME with Remote Offices

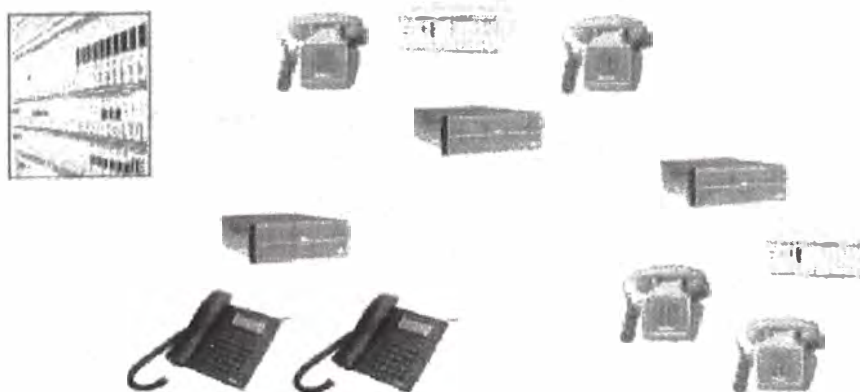


Figura No. 3.14. Conexión del Servidor Asterisk con oficinas remotas.

Por otro lado, también es posible dar el servicio de oficina remota, donde un usuario puede tener una extensión o anexo fijo en un softphone, sea cual sea el lugar donde se encuentre, siempre y cuando tenga acceso a Internet. Esta característica viene

proporcionada por Asterisk, como en el diagrama siguiente, que muestra cómo se puede construir múltiples oficinas con MultiVOIP gateways, enlazándolas a todas juntas por intermedio de Asterisk.

De acuerdo a la figura 3.14 anterior, una persona en Cairo, que usa una extensión 120 del Servidor PBX, puede viajar a Lima, y si se conecta al servidor desde su softphone, puede seguir usando la misma extensión y ser ubicado por los demás usuarios del sistema que se encuentran en Cairo, Beirut, Nueva York, Bahrein, Trípoli, etc. Estas son las facilidades que provee este Sistema.

3.8.3 Menú de voz Interactivo y Plan de Llamadas Local y de la Red del PNUD

Dado que el sistema Asterisk es completamente flexible, es posible separar lógicamente la red de voz sobre IP del PNUD y el sistema telefónico de IT Synergy que opera localmente en Cairo. De esta manera se pueden configurar distintas IVRs (Interactive Voice Responses) para cada una de las diferentes redes.

La separación se lleva a cabo, porque la Oficina de IT Synergy es llamada constantemente para materias administrativas desde Egipto por cualquier persona. Dado que no se desea permitir el acceso de cualquiera de estas personas a la red de voz sobre IP del PNUD, entonces se vio en la necesidad de separar ambos sistemas, pero usando un mismo servidor.

- **Plan de Llamadas de la Red del PNUD:** Las oficinas del PNUD poseen todas su propio PBX digital, que ha sido interconectado con la red de voz sobre IP mediante los gateways MultiVOIP. Dado que poseen un PBX local, éstos ya tienen consigo un Menú interactivo de respuestas configurado localmente. La interconexión de todos estos menús, es posible configurando un Menú general que ha sido configurado en el Plan de Llamadas del servidor Asterisk. De este modo, cualquiera puede entrar al menú general para luego a partir de ahí ingresar al menú local de país, si es que no recuerda la extensión a la cual desea llamar.
- **Plan de Llamadas Local de IT Synergy:** A diferencia del PNUD, IT- Synergy no poseía anteriormente un plan de llamadas local, ya que el Servidor Asterisk fue configurado para ese sentido. En el mismo servidor Asterisk se pueden configurar diferentes contextos donde el contexto [internal] define el punto de entrada para el plan de llamadas local y el contexto [undp] define el plan de llamadas del PNUD, el cual está completamente separado lógicamente del primero.

Para evitar que los teléfonos locales queden aislados totalmente de la red del PNUD, basta con marcar una combinación de teclas con una contraseña para poder ingresar del menú local al menú general e inicio del plan de llamadas de la red del PNUD. Esto se logra gracias al uso de los contextos en Asterisk.

El plan de llamadas en la oficina de IT-Synergy sigue una lógica muy simple, dado que es el teléfono de la empresa. Igualmente, el sistema de respuestas interactivo que se usa en la empresa es básicamente un saludo y la petición de marcado para una determinada extensión o el ingreso al directorio de la Empresa.

El directorio es parte de las aplicaciones que pueden desarrollarse con Asterisk, el cual marcando los números que corresponden a las tres primeras letras del nombre o apellido de una persona, el sistema responde con el nombre de la persona buscada y su extensión haciendo posible la ubicación de cualquier empleado de la empresa.

La configuración de una aplicación de directorio telefónico se puede hacer dentro del archivo extensions.conf.

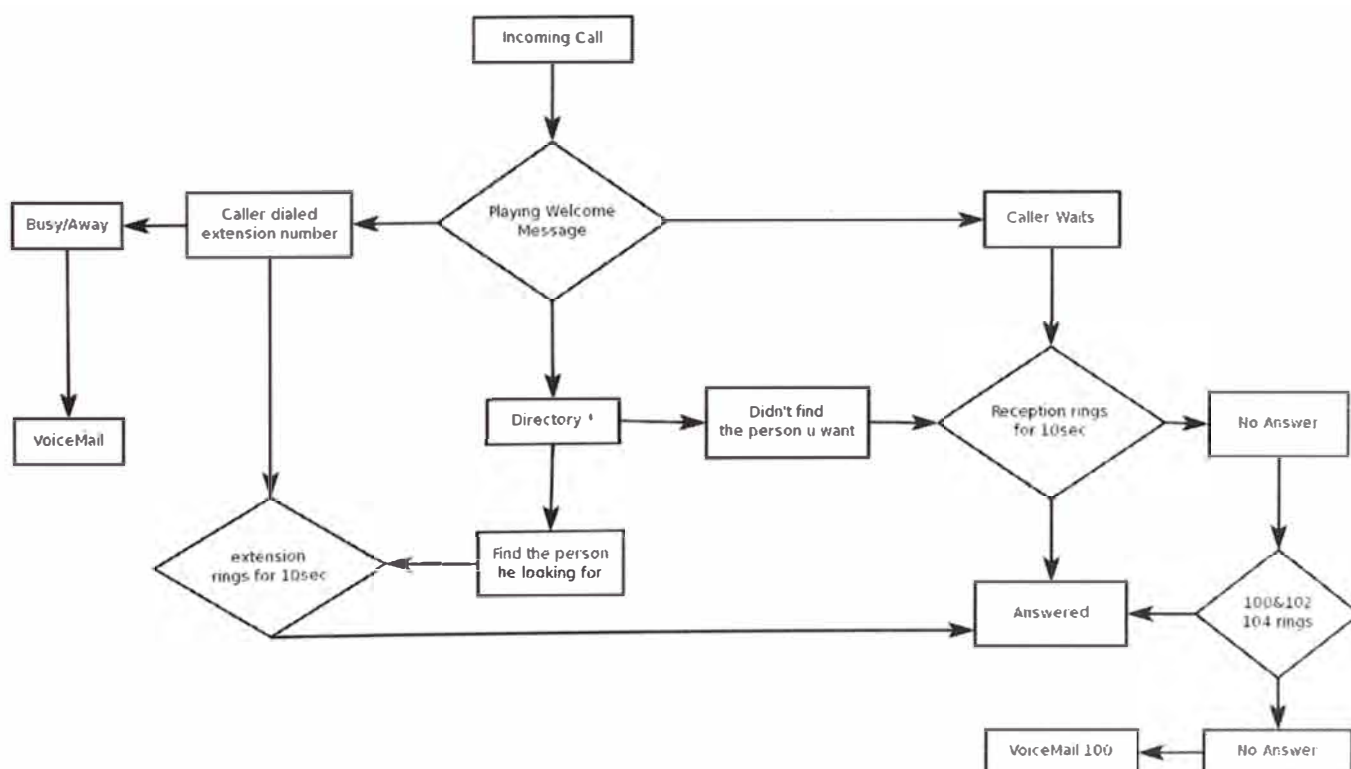


Figura No. 3.15. Flujo del Plan de llamadas de IT-Synergy configurado en el servidor Asterisk.

Este es el plan de llamadas configurado para IT Synergy en el servidor, aislado del plan de llamadas de la red del PNUD, como se puede apreciar en la figura 3.15 anterior.

Es necesario recordar que este plan es configurado en el archivo `extensions.conf`. Detalles de la configuración de este archivo puede encontrarse en el **Apéndice D**.

3.8.4 Aplicaciones usadas en el Sistema Asterisk para la red de VoIP del PNUD

El sistema implementado para la red de VoIP del PNUD posee además muchas aplicaciones desarrolladas exclusivamente para las Naciones Unidas. Entre estas aplicaciones, podemos enumerar las siguientes:

- Multi-Conferencia
- Casillas de Voz
- Traducción de Texto al Habla (Text-To-Speech Engine) a través de Festival
- Manejo y control de llamadas a través del Flash Panel Operador
- Interfaz de configuración web.
- Reportes y Registro Web de llamadas hechas a través del Sistema.
- Monitoreo de Dispositivos de VoIP por intermedio de Nagios Monitoring System (www.nagios.org). El site esta monitoreado en <http://www.undpvoip.net>.

CAPITULO IV

DESCRIPCIÓN DE LAS CARACTERÍSTICAS DEL USO DEL SISTEMA PROPUESTO

4.1 Plan de Llamadas

El Plan de llamadas es el corazón del sistema Asterisk, porque define cómo se manejan las llamadas entrantes y salientes (8). Consiste de una lista de instrucciones o pasos que Asterisk seguirá. A diferencia de los sistemas telefónicos tradicionales, el plan de llamadas de Asterisk es completamente personalizable. Para configurarlo satisfactoriamente, se necesitará solamente entender su sintaxis.

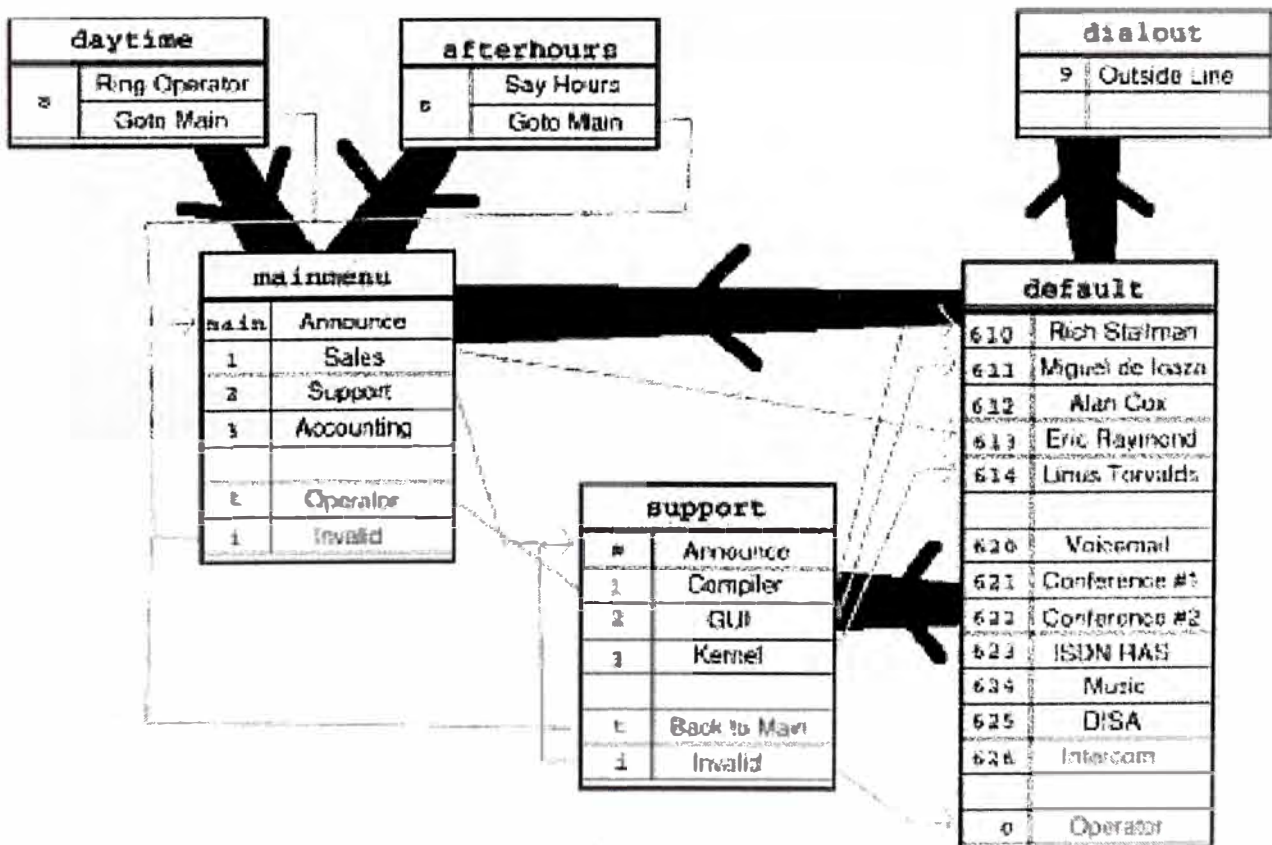


Figura No. 4.1. Diagrama de un Plan de Llamadas.

El plan de llamadas de Asterisk (como ejemplo, ver el diagrama anterior de un plan de llamadas) es quien rutea cada llamada al sistema desde su fuente, a través de sus varias aplicaciones, hasta su destino final. Todo, desde casillas de voz, conferencia, auto respuesta, menú de voz, etc. es manejado por un concepto lógico y consistente llamado Plan de Llamadas.

El plan de llamadas de Asterisk es especificado en el archivo de configuración llamado `extensions.conf`.

El plan de llamadas esta hecho de cuatro partes principales: Contextos, extensiones, prioridades y aplicaciones.

4.1.1 Contextos (Contexts)

El Plan de llamada está dividido en secciones llamadas Contextos (contexts). Los contextos son llamados grupos de extensiones. Poniéndolo simple, ellos mantienen diferentes partes del plan de llamadas de interactuar uno con el otro. Una extensión que es definida en un contexto, está completamente aislada de las extensiones en otro contexto, a menos que una interacción sea específicamente permitida.

Como un ejemplo simple, imaginemos que tenemos dos compañías compartiendo un servidor Asterisk. Si colocamos el menú de voz de cada compañía en su propio contexto, efectivamente se separarán uno del otro. Esto permite definir independientemente lo que sucede si, digamos, la extensión 0 es marcada: Las personas que marcan 0 en el menú de voz de la compañía A llegarán a la recepcionista de la compañía A. Las que marquen 0 en el menú de voz de la compañía B, llegaran a la recepcionista de la compañía B. Este ejemplo asume, que cuando le decimos a Asterisk que transfiera las llamadas a la recepcionista cuando la persona que llama marca 0.

Los contextos son denotados colocando el nombre del contexto entre corchetes.

[incoming]

Todas las instrucciones colocadas luego de la definición de contexto son parte del contexto hasta que el siguiente es definido. Al comienzo del plan de llamadas, hay dos contextos especiales llamados [general] y [globals].

Uno de los más importantes usos de los contextos es para reforzar la seguridad. Usando contextos correctamente, se puede dar acceso a ciertas llamadas a algunas características (como llamadas a larga distancia) que no están disponibles a otras personas. Si no se diseña cuidadosamente el plan de llamadas, se puede inadvertidamente permitir a otros usar el sistema fraudulentamente.

4.1.2 Extensiones (Extensions)

Dentro de cada contexto, definimos una o más extensiones. Una extensión es una instrucción que Asterisk seguirá, activada por una llamada entrante o por dígitos que han sido marcados en un canal. Las extensiones especifican lo que sucede con las llamadas cuando hacen su recorrido por el plan de llamadas. Aunque las extensiones pueden ser usadas para especificar las extensiones telefónicas en el sentido tradicional (por ejemplo, llamar a José en la extensión 153), ellas se pueden usar para mucho más en Asterisk.

La sintaxis para una extensión es la palabra `exten`, seguido por una flecha formada por un signo igual y un `>` mayor de:

```
exten =>
```

En Asterisk, una extensión puede ser una combinación de números y letras.

Una extensión completa está compuesta de tres componentes:

- El número de la extensión
- La prioridad (cada extensión puede incluir múltiples pasos; el número de pasos es llamado "prioridad")
- La aplicación (o comando) que ejecuta alguna acción sobre la llamada.

Estos tres componentes están separados por comas, como por ejemplo:

```
exten => name,priority,application( )
```

Este es un ejemplo de cómo puede verse una extensión real:

```
exten => 123,1,Answer( )
```

En el ejemplo anterior, el nombre de la extensión es 123, la prioridad es 1, y la aplicación es `Answer()`.

4.1.3 Prioridades

Cada extensión puede tener múltiples pasos, llamadas priorities. Cada prioridad es un número secuencial, que comienza en 1. Cada prioridad ejecuta una aplicación específica. Como ejemplo, la siguiente extensión respondería el teléfono (en prioridad número 1) y luego colgaría (en prioridad numero 2):

```
exten => 123,1,Answer( )
```

```
exten => 123,2,Hangup( )
```

4.1.4 Aplicaciones

Las aplicaciones son los caballos de trabajo del plan de llamadas. Cada aplicación ejecuta una acción específica en el canal actual, tal como tocar sonido, aceptar la entrada de tonos, o colgar la llamada.

Algunas aplicaciones, como las mencionadas anteriormente Answer() y Hangup(), no necesitan otras instrucciones para ser llamadas. Otras aplicaciones requieren información adicional. Estas piezas de información, llamadas argumentos, pueden ser pasadas a las aplicaciones para afectar su ejecución. Para pasar los argumentos a una aplicación, se deben colocarlos entre paréntesis, separados por comas.

4.1.5 Un Plan de Llamadas Simple

Vamos a diseñar un plan de llamadas, de tal manera que cuando una llamada llegue, Asterisk responda, toque un archivo de sonido, y luego cuelgue la llamada. Para este ejemplo, asumiremos que al menos un canal Zap ha sido creado y configurado y que todas las llamadas entrantes son enviadas al contexto [incoming].

Cuando una llamada entra a un contexto sin una extensión destino específico (por ejemplo, una línea FXO sonando), ésta es manejada automáticamente por la extensión s.

En el siguiente ejemplo, vamos a ejecutar tres acciones a la llamada (contestar, tocar un archivo de sonido y colgar), así que necesitamos crear la extensión llamada s con tres prioridades. Se colocarán las tres prioridades dentro de [incoming], ya que las llamadas deben de comenzar por ese contexto:

```
[incoming]
```

```
exten => s,1,application( )
```

```
exten => s,2,application( )
```

```
exten => s,3,application( )
```

La aplicación Answer() es usada para responder un Canal que está sonando. Este hace el seteo inicial para el canal que recibe la llamada entrante. Unas pocas aplicaciones no requieren que se responda el canal primero, pero respondiendo el canal antes de ejecutar alguna acción es una buena práctica. Answer() no tiene ningún argumento.

La aplicación Playback() es usada para tocar un sonido grabado previamente sobre un canal. Cuando se usa Playback(), la entrada del usuario es simplemente ignorada.

Para usar Playback(), se especifica un archivo (sin extensión) como argumento. Por ejemplo Playback(filename) tocará el archivo filename.gsm, asumiendo que está localizado en el directorio de sonidos por defecto. Se puede siempre incluir la ruta completa al archivo si se requiere:

```
Playback(/home/john/sounds/filename)
```

También es posible usar rutas relativas.

La aplicación Hangup() hace exactamente lo que su nombre implica. Esta corta el canal activo. La persona que llama recibiría una indicación de que la llamada ha sido terminada. Esta aplicación es comúnmente usada al final de cada contexto cuando se quiere finalizar la llamada actual, para asegurarse que los que llamen no continúen dentro del plan de llamada. Esta aplicación no toma ningún argumento.

Ahora que la extensión ha sido creada, se le ha dado tres diferentes prioridades, se pueden juntar todo para crear el primer plan de llamadas simple:

En la primera prioridad de la extensión se responderá la llamada. En la segunda, se tocará un archivo de sonido musica.gsm, y en el tercero se terminará la llamada. Así se configuraría este plan de llamadas:

```
[incoming]
```

```
exten => s,1,Answer( )
```

```
exten => s,2,Playback(música)
```

```
exten => s,3,Hangup( )
```

Si se tiene un canal o dos configurados,

4.1.6 Un Plan de Llamadas más Complejo

Un Plan de Llamadas más complejo luciría de esta manera:

[incoming]

```

exten => s,1,Answer( )
exten => s,2,Background(enter-ext-of-person)
exten => 101,1,Dial(Zap/1,10)
exten => 101,2,Playback(vm-nobodyavail)
exten => 101,3,Hangup( )
exten => 101,102,Playback(tt-allbusy)
exten => 101,103,Hangup( )
exten => 102,1,Dial(SIP/Jane,10)
exten => 102,2,Playback(vm-nobodyavail)
exten => 102,3,Hangup( )
exten => 102,102,Playback(tt-allbusy)
exten => 102,103,Hangup( )
exten => i,1,Playback(pbx-invalid)
exten => i,2,Goto(incoming,s,1)
exten => t,1,Playback(vm-goodbye)
exten => t,2,Hangup( )

```

[internal]

```

exten => 101,1,Dial(Zap/1,,r)
exten => 102,1,Dial(SIP/jane,,r)

```

En el ejemplo anterior se han usado las aplicaciones Dial(), Playback() y Goto().

Esta aplicación Dial() se encarga de conectar a las llamadas entrantes con los diversos canales, que se insertan como argumentos dentro de la aplicación. Por ejemplo Dial(Zap/1,10) enlaza la llamada en cuestión con el canal Zap/1, que puede ser un canal FXS, como un teléfono analógico, o un teléfono SIP, si es que se usa Dial(SIP/Jane,10). El segundo argumento es la cantidad de segundos que sonará ese canal.

4.2 Sistema de Respuestas Interactivas (IVR)

IVR es el acrónimo de Interactive Voice Response, que se traduce como Respuesta de Voz Interactiva.

Consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz. Es la típica máquina que nos responde para saber la hora o cuando llamamos al banco.

IVR (interactive voice response) o VRU (voice response unit) Sistema de respuesta interactiva, orientado a entregar y/o capturar información automatizada a través del teléfono permitiendo el acceso a los servicios de información y operaciones autorizadas, las 24 horas del día. puede realizar todo tipo de transacciones disponibles.

El modo de funcionamiento es el siguiente: El usuario realiza una llamada a un número de teléfono, el sistema de audio respuesta contesta la llamada y le presenta al usuario una serie de acciones a realizar, esto se hace mediante mensajes (menús de opciones). El usuario elige la opción a realizar ingresando un número en el teclado del teléfono y navega por los diferentes menús hasta encontrar la información solicitada.

Una de las características mas modernas y fascinantes de este tipo de sistemas es la capacidad TTS (Text To Speech): Capacidad que tiene para interpretar palabras de un archivo de texto, y mediante una librería de audio, reproducir las palabras del texto.

Asterisk soporta este tipo de configuración de Sistema de Respuestas Interactivas con Texto hablado, implementando el motor Festival, que es un motor Text-To-Speech.

La premisa básica para el uso de Festival con Asterisk es que el plan de llamadas pueda pasar un cuerpo de texto a Festival, el cual luego “hablará” el texto al que llama. Probablemente, el uso más obvio para Festival sería el de leer el correo electrónico cuando se llame a una determinada extensión mientras uno esté viajando en su camino.

4.2.1 Configuración de Festival

Una vez instalado y configurado Festival, el archivo de configuración que se debe de revisar es llamado festival.conf. Dentro de este archivo se especifica el nombre de host y el puerto por donde está ejecutándose el servidor Festival, así como otras opciones. Para la mayoría de las instalaciones, las opciones por defecto funcionarán bastante bien.

Para iniciar Festival en “modo de prueba”, deberá de ejecutarse:

```
[root@asterisk ~]# festival --server
```

Si se está seguro que Festival esta ejecutándose y aceptando conexiones, se puede activarlo con:

```
[root@asterisk ~]# festival_server 2>&1 >/dev/null &
```

4.2.2 Usando Festival desde el Plan de Llamadas

Para Usar Festival desde el plan de llamadas, se escribirán las siguientes líneas:

```
exten => 123,1,Answer( )
```

```
exten => 123,2,Festival(Asterisk y Festival están configurados y trabajando bien)
```

Siempre debe de llamarse a la aplicación Answer(), antes de llamar a Festival para asegurarse de que el canal esté establecido.

4.3 Auto-Attendant

Un sistema de Auto-Attendant o de respuesta automática en telefonía es un sistema que permite a los que llaman ser transferidos a la extensión de un usuario automáticamente sin la intervención de un recepcionista, que actúa como operador de teléfono. Esta característica está presente en todos los PBX modernos, y también esta en Asterisk.

También es posible configurar un directorio de “Marca por nombre”, donde las tres primeras letras del apellido o nombre de la persona son usadas como anexo para localizarlas.

Igualmente un teléfono puede ser reconfigurado, simplemente marcando una tecla, como “No molestar”, para que todas las llamadas que vengan a esa extensión vayan directamente al correo de voz sin que suene la extensión con la llamada entrante.

Muchas aplicaciones interesantes pueden ser desarrolladas usando este tipo de sistema, como por ejemplo un sistema de registro, donde los usuarios se registran en un curso llamando a un número y presionando un juego de teclas.

Muchas aplicaciones más son posibles si es que se enlazan las llamadas hechas por un usuario a un script del servidor que se puede usar para activar una rutina determinada que controle algo. Por ejemplo se puede resetear el servidor cada vez que se marque un número y se presione una combinación de teclas como contraseña.

4.4 Manejo de los distintos protocolos de Voz sobre IP

La premisa básica para los protocolos de VoIP es la paquetización de flujo de audio para transportarlos sobre las redes basadas en protocolo de Internet. No solo debe la señal llegar en esencia de la misma manera en que fue transmitida, pero también tiene que hacerlo en menos de 300 milisegundos. Si los paquetes se pierden o se retardan, habrá una degradación de calidad en la comunicación.

Los protocolos de transporte que colectivamente definen la "Internet" no fueron originalmente diseñados teniendo en mente la transmisión de flujo de datos en tiempo real. Los puntos de destino deberían, según lo planeado, esperar un tiempo adicional a que se resuelvan los paquetes perdidos, requiriéndolos nuevamente. En una conversación típica, estos mecanismos no sirven. Nuestras conversaciones no se adaptan bien a la pérdida de palabras ni tampoco a un apreciable retardo entre la información transmitida y el emisor.

La red de telefonía pública PSTN fue diseñada específicamente con el propósito de transmisión de voz, y está perfectamente implementada para este trabajo desde un punto de vista técnico. Desde un punto de vista de flexibilidad, sin embargo, sus defectos son obvios incluso hasta a las personas con un limitado entendimiento de tecnología. VoIP tiene la promesa de incorporar comunicaciones de voz a los otros protocolos que llevan las redes, pero debido a las especiales demandas de las conversaciones de voz, se necesitan habilidades especiales para diseñar y mantener esas redes.

El problema con la transmisión de voz basada en paquetes viene del hecho en que la forma en que hablamos es totalmente incompatible con la manera en que la Internet transporta los datos.

Para Asterisk, prácticamente todos los protocolos de voz sobre IP son soportados:

- IAX (versiones 1 y 2)
- SIP
- H.323
- MGCP
- Skinny/SCCP
- UNISTIM
- Voice over Frame Relay (VOFR)
- Jingle Protocol, Jabber – Google Talk

- Interfaz indirecta a Skype

4.5 Puente para Conferencias

Una aplicación para tele-conferencia de voz es de gran utilidad para un PBX. Asterisk tiene la facilidad de integrar muchas características para conferencias de voz con la aplicación MeetMe().

Las funcionalidades que provee Asterisk, al usar este tipo de aplicación son las siguientes:

- Conferencia multiusuario sobre múltiples canales – VoIP y PSTN
- Anuncios: Un hablante y múltiples escuchas
- Administración – Sacar usuarios de las conferencias, setear a canales a mudo y bloquear conferencias.
- Música de fondo mientras esperan otros participantes
- Conferencias estadísticamente configuradas con código de contraseña de entrada.
- Conferencias dinámicas, creadas en demanda (El número de la conferencia es comunicado al usuario cuando entra)

Por ejemplo, La sala de conferencia 101, como está definida en meetme.conf, con password (PIN) 123456:

```
exten => 500,1,MeetMe(101||123456)
```

Sala de conferencia autenticada:

```
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
```

Sala de conferencia dinámica, el usuario debe de ingresar su numero de sala para que la conferencia sea creada:

exten => 9999,1, Wait(1)

exten => 9999,2,MeetMe(IMd) ; NOTA: Si se añade la opción 'e', Asterisk escogerá el número de salón; cambiar la opción 'd' por 'D' si se quiere tener un número pin para la conferencia.

4.6 Interfase de Manejo: Panel de Operador en Flash

La interfase de manejo de Asterisk provee una Interfaz de Programación de Aplicaciones que permite a programas externos la habilidad de crear, monitorear y manejar Asterisk. La interfaz de manejo es un mecanismo poderoso para integrar programas externos de todo tipo a Asterisk.



Figura No. 4.2. Vista web de un Panel de Operador basado en Flash.

Para usar el manejador, se debe definir una cuenta en el archivo `/etc/asterisk/manager.conf`. Este archivo se verá algo así:

[general]

enabled = yes

port = 5038

bindaddr = 0.0.0.0

```
[liliana]
secret = notvery
;deny=0.0.0.0/0.0.0.0
;permit=209.16.236.73/255.255.255.0
read = system,call,log,verbose,command,agent,user
write = system,call,log,verbose,command,agent,user
```

En la sección [general] se tiene que habilitar el servicio seteando el parámetro enabled=yes. El puerto TCP se usará por defecto a 5038.

Para cada usuario, se especificará el IP que se quiere negar o acceder, permisos de lectura/escritura, etc.

El Flash Operator Panel, o Panel de Operador en Flash es el ejemplo más popular del poder de la interfaz de manejo. FOP crea una interfase visual web del sistema y permite control de llamadas.

FOP es mas comúnmente usado para habilitar control en vivo, ver a los usuarios en el sistema y conectar las llamadas entre ellos. Puede ser usado también en un Call Center.

4.7 Asterisk en la industria de las Telecomunicaciones

A pesar de que el inventor del teléfono, Alexander Graham Bell fue una persona académica, la realidad es que en los últimos años de los 1800s, muchas de las personas que estaban tratando de llevar la voz a través de líneas telegráficas tenían la mente orientada a los negocios, tratando siempre de crear un producto que fuera rentable.

La historia de la telefonía tuvo lugar en un entorno muy competitivo, con nuevas compañías creándose alrededor del mundo, frecuentemente con poco o nada de respeto por los patentes que estuvieran violando. Algunos de los monopolios que se formaron tuvieron su inicio en una guerra de patentes.

Es interesante contrastar la historia de la telefonía con la historia de Linux y la Internet. Mientras el teléfono fue creado como un ejercicio comercial, y la industria de la telefonía fue forjada a través de juicios y tomas de posesión de corporaciones, Linux y la Internet se crearon desde la comunidad académica, la cual valora más que se compartan los conocimientos sobre los beneficios económicos.

Las diferencias culturales son entonces obvias, las tecnologías de las telecomunicaciones tienden a ser cerradas, confusas y caras, mientras que las tecnologías de red son generalmente abiertas, documentadas y competitivas.

Si uno compara la cultura de la industria de las telecomunicaciones con la de la Internet, es a veces difícil creer que ambas están relacionadas. La Internet fue creada por entusiastas, donde la contribución al desarrollo de la red PSTN es imposible de contemplar por cualquier individuo. Generalmente la PSTN era un club exclusivo, donde la membresía no estaba abierta al público general.

A pesar de que la Unión de Telecomunicaciones Internacional (ITU - International Telecommunication Union) es el cuerpo designado por las Naciones Unidas como responsable para las telecomunicaciones internacionales, muchos de los protocolos de VoIP (SIP, MGCP, RTP, STUN) no han venido de las salas del ITU, sino del IETF – Internet Engineering Task Force (el cual publica todos sus estándares libres para todos, y permite a cualquiera enviar un borrador de protocolo para la Internet para ser tomado en consideración).

Los protocolos abiertos como SIP pueden tener una ventaja táctica sobre los protocolos del ITU, tales como el H.323 debido a la facilidad con la cual puede ser obtenido. A pesar que H.323 haya sido ampliamente usado por los proveedores de VoIP en sus backbones, es más difícil encontrar puntos finales basados en H.323; ya que los nuevos productos parecen más orientados para soportar SIP.

El camino hacia la apertura del conocimiento es mucho más aceptado por la comunidad. Asterisk abraza el pasado y el futuro de la telefonía. El soporte de H.323 esta disponible, a pesar que la comunidad Internet haya rehusado el uso de H.323 a favor del protocolo SIP de IETF y el nuevo protocolo de telefonía de la comunidad Asterisk IAX.

Una de las cosas más extrañas acerca de los estándares que existen en el mundo de las telecomunicaciones tradicionales es la inhabilidad de las varias empresas de manufactura de implementarlas consistentemente. Cada empresa deseaba un monopolio total, y el concepto de interoperabilidad tendía a quedar rezagada con la opción de ser primero en el mercado con una nueva idea creativa.

Los protocolos de RDSI son un clásico ejemplo de esto. La implementación de las redes RDSI fue (y en muchos casos aún lo es) una propuesta costosa y dolorosa, ya que cada empresa de manufactura decide implementarla de una manera ligeramente diferente. La

RDSI podría haber ayudado a interconectar a los usuarios en una red de datos pública masiva, 10 años antes de Internet. Desafortunadamente, debido a los costos, complejidad y problemas de compatibilidad, la RDSI nunca entregó mucho más que solamente voz, y ocasionalmente una conexión de video o datos para aquellos deseosos de pagar la considerable suma de dinero requerida por tal conexión. La RDSI es un sistema común en Europa y Norte América en implementaciones largas de PBX, pero incluso allí, o en cualquier otro lugar, no están entregando ningún servicio cercano a las capacidades para las que fue creado y diseñado. Mientras la VoIP siga creciendo, la necesidad por la RDSI desaparecerá.

Puede tomar meses, e incluso años para que las grandes empresas de telecomunicaciones puedan admitir una tendencia, y producir un producto compatible con esta tendencia. Para que una nueva tecnología sea implementada, debe de ser analizada durante mucho tiempo, y debe de pasar todas las capas burocráticas antes de entrar al ciclo de desarrollo. Meses y años pueden pasar antes de que un producto pueda ser finalmente lanzado, y frecuentemente están basados en hardware obsoleto. Ellos también tienden a ser costosos y no ofrecen más que un pequeño juego de nuevas características.

Los ciclos de desarrollo lentos no funcionan en los negocios de telecomunicaciones de hoy en día. En la Internet, nuevas ideas toman lugar en unas semanas y se vuelven extremadamente viables en cortos periodos de tiempo. Ya que cada tecnología se debe adaptar a estos cambios, también tienen que hacerlo las telecomunicaciones.

El desarrollo de fuente abierta está inherentemente mejor capacitado para adaptarse al cambio tecnológico rápido, el cual le da una enorme ventaja de competitividad.

La espectacular caída de la industria de las telecomunicaciones puede haber sido causada en gran medida por su inhabilidad al cambio. Tal vez, esta continua inhabilidad es la razón por la que la recuperación ha sido tan lenta. Ahora no hay opción: El cambio o la inanición. La tecnología manejada por la comunidad Internet tiene a Asterisk como mejor competidor en este tema.

Las compañías de telecomunicaciones tradicionales han perdido el toque con sus clientes. Actualmente, las personas tienen ilimitada flexibilidad en otras formas de comunicación, que simplemente no pueden entender por qué las telecomunicaciones no pueden entregar esa flexibilidad como la industria lo había prometido hace algunos años.

El concepto de flexibilidad no es familiar a la industria de telecomunicaciones, y puede no estarlo hasta que los productos de fuente abierta, como Asterisk, empiecen a transformar la naturaleza fundamental de la industria. Esta es una revolución parecida a la que Linux y la Internet comenzaron unos diez años atrás.

En esta era de bases de datos personalizadas y desarrollo de sitios web, las necesidades creativas de los clientes, conjuntamente con las limitaciones de la tecnología, han catapultado un tipo de creatividad nacida de la necesidad: los ingenieros de telecomunicaciones tradicionales están tratando de crear dispositivos funcionales de una pila de componentes disparejos.

El desarrollo de la metodología de un sistema de telefonía propietario dicta que este tendrá un gran número de características, y que el número de características determinará en gran medida el precio del servicio. El hecho de que el usuario necesite solo 5 de 500 características es ignorado, y el deseo del cliente de tener características no disponibles que manejen las necesidades de su negocio es rechazado como "no razonable". Hasta que la flexibilidad se convierta en estándar, las telecomunicaciones permanecerán detenidas en el siglo pasado.

Desde la perspectiva de la industria cerrada, Frederick Brooks (6) opina que "los costos de la complejidad y comunicación de un proyecto aumentan con el cuadrado del número de desarrolladores, mientras que el trabajo hecho solo aumenta linealmente". Sin una metodología de desarrollo basado en comunidad, es muy difícil entregar productos que en el mejor de los casos son un poco más que mejoras incrementales sobre sus predecesores, y en el peor de los casos, una mera colección de parches.

Asterisk ataca el problema directamente, y lo resuelve de una manera que muy pocos sistemas de telefonía pueden. Esta es una tecnología extremadamente demoledora, en parte porque está basada en conceptos que han sido probados una y otra vez: "El mundo de código fuente-cerrado no puede ganar en la carrera tecnológica contra las comunidades de fuente abierta, que pueden poner muchas órdenes de magnitud más de tiempo hábil en un problema" (6).

Uno de los problemas de la industria de telecomunicaciones tradicional ha sido su aparente rechazo a cooperar consigo misma. Los grandes gigantes de telecomunicaciones han estado en el mercado por alrededor de cien años. El concepto de sistemas propietarios cerrados está tan asimilado en su cultura que incluso sus

intentos de ser compatibles con estándares son vencidos por sus deseos de saltar más alto en la competencia, añadiendo nuevas características que nadie más soporta. Por ejemplo, uno solo tiene que mirar a los productos de VoIP ofrecidos por la industria de telecomunicaciones de hoy en día. Mientras que ellos dicen ser compatibles con estándares, lo que uno esperaría es poder conectar realmente un teléfono Cisco con un Switch Nortel, o que un sistema de Casilla de Voz Avaya pueda ser integrado vía IP con un PBX Siemens, pero ése no es el caso en ninguno de ellos.

En la industria de computadoras, las cosas son distintas. Veinte años atrás, si se compraba un servidor IBM, se necesitaba una red IBM y terminales IBM para poder interconectarlos. Ahora los servidores IBM pueden fácilmente interconectarse con terminales Dell a través de una red Cisco (ejecutando Linux, Windows u otros sistemas operativos). Si alguna de esas compañías, hoy en día, sugiriera que solo se pudiera usar sus productos en toda la red descrita anteriormente, sería dejada atrás rápidamente del negocio de computadoras.

CAPITULO V

EVALUACION DE COSTOS

5.1 Costo de Instalación de un PBX Clásico

Un PBX es básicamente una pieza de equipo que conecta llamadas telefónicas. Un sistema PBX telefónico tiene muchas características principales. Establece y mantiene conexiones y también mide las llamadas para procesos de tarificación y contabilidad. Existen muchas razones por las cuales las compañías, negocios e individuos optarían por una solución de sistema telefónico PBX. Pero la razón más obvia es por ahorro.

Los sistemas telefónicos PBX eliminan la necesidad de compra de varias líneas telefónicas separadas para cada empleado individual de una oficina o una casa. Una compañía telefónica actúa como oficina central y todas las llamadas son distribuidas a través de esa oficina. Con un sistema PBX, las llamadas dentro de la compañía son manejadas a través del PBX en lugar de la compañía telefónica. En lugar de comprar y asignar a cada persona un teléfono separado, todo lo que se necesita es una extensión. Para una compañía de 75 empleados, por ejemplo, el sistema PBX elimina la necesidad de contratar 75 líneas individuales con la compañía telefónica. En lugar de ello, la compañía puede comprar 7-10 líneas para el propósito de hacer llamadas fuera de la compañía.

Las llamadas fuera de la compañía son hechas a través del número principal o línea que sirve a la casa u oficina, accediendo a dicha línea marcando el número 9.

Mientras que algunos sistemas de telefonía PBX tienen muchas características, otros que son instalados en casas u oficinas pequeñas no las tienen tantas. Dependiendo de la empresa de manufactura, el sistema telefónico PBX puede tener alguno o muchas de las características siguientes: parqueo de llamadas, transferencia y recepción, llamada en espera, reenvío y contabilidad, conferencia de llamadas, saludos personalizados y

respuesta de llamadas interactivas, casilla de voz, música de espera, servicio nocturno, distribución de llamadas, etc.

Aparte de reducir los costos en los pagos mensuales de teléfono, los sistemas PBX también ayudan a ahorrar dinero reduciendo las llamadas desautorizadas de salida. Los sistemas PBX pueden controlar las llamadas salientes bloqueando números específicos o anexos, llamadas a larga distancia o internacionales.

Los sistemas PBX tales como aquellos usados en casa o pequeños negocios, así como pequeños empresarios pueden variar desde \$100 hasta como \$1,500 o más, dependiendo de sus características. Estos PBX solo interconectan a la PSTN y ofrecen unas muy pobres características.

Compañías gigantes que van desde 100 a 20 000 empleados pueden pagar por una solución PBX entre \$10,000 a \$100,000 por un sistema de PBX muy elaborado. Entre los vendedores estrella de los sistemas de telefonía clásica tenemos: NEC (www.nec.com), Siemens (www.siemens.com), Panasonic (www.panasonic.com) y Nortel (www.nortel.com).

5.1.1 Precios de implementación de Teléfonos de Oficina

La compra de teléfonos de oficina basados únicamente en precios puede fácilmente ocasionar muchos problemas luego: sistemas más confiables y más extensibles tienden a costar más por usuario, pero los ahorros que se ven en el largo plazo pueden agregarle más valor, dependiendo de los requerimientos. Además de poder añadir mas usuarios y líneas, mientras la empresa crece, se debe de evaluar lo fácil que los teléfonos de oficina pueden ser re-actualizados en las nuevas características y tecnologías.

Estimar los costos para un sistema telefónico completo es muy difícil sin conocer las características y requerimientos de una empresa, además de su tamaño y creciente expansión. Los costos pueden fácilmente saltar los miles de dólares. Sistemas encriptados e híbridos (Telefonía clásica + Voz sobre IP) pueden variar entre los \$350 a \$1,000 por usuario, dependiendo de las características seleccionadas (9).

Para implementaciones largas de PBX, los precios pueden comenzar desde los \$800 por usuario, pero usualmente se elevan al rango de los \$1,000 por usuario. De acuerdo a las muchas escalas, las pequeñas oficinas encuentran difícil de encontrar precios debajo de

los \$1,000 por usuario en cualquier sistema, y compañías como 100 o más empleados ahorran considerablemente.

Sistemas sin cable pueden generalmente costar entre \$130 y \$225 por teléfono. Mientras que los ahorros pueden resultar atractivos, hay que tener en mente que es realmente útil para empresas de menos de 10 personas con poca o nada utilización o personalización de aplicaciones basadas en telefonía. Si un sistema telefónico sin cable falla en cumplir los requerimientos, los ahorros que parecerían haberse obtenido se desvanecerán. Y lo que es peor, parte de los recursos de la empresa tendrán que emplearse en mantener el sistema.

Los precios de los sistemas telefónicos pueden variar en base a estos 5 factores:

- **El sistema base:** El sistema de base central, o gabinete controla todo el sistema telefónico. El precio difiere entre los sistemas y se eleva mientras se añadan más tarjetas y accesorios. Una pequeña unidad central puede costar tan poco como \$1,000 con el precio incrementándose considerablemente a los más de diez mil dólares para sistemas largos. El sistema base será el principal factor limitante para el sistema telefónico en términos de características y expansibilidad.
- **Los teléfonos actuales:** La mayoría de los sistemas pueden ser equipados con muchos diferentes tipos de teléfonos de oficina. Los sets menos caros pueden costar menos de \$100, pero es muy difícil de obtener características accesorias o proveen menos de la óptima calidad de sonido. La mayoría de handsets se venden por \$200 a \$300 por unidad. Por otra parte, algunos "teléfonos ejecutivos" se venden por mucho al precio estándar. Las estaciones de recepción son también más caras, pero éstas brindan importantes características para la persona en el centro del sistema telefónico. La mayoría de los negocios tienen una mezcla de ambos modelos.
- **Características extras de los sistemas telefónicos:** También sería necesaria la compra de accesorios de sistemas telefónicos como correo de llamadas, música en espera, y aplicaciones de integración con telefonía por computadora, si el sistema no viene con esas características por defecto. Estas características pueden escalar a los miles de dólares en la compra. Muchas empresas de manufactura que venden equipos telefónicos de oficina agrupan algunas de esas características para que sea menos costoso. Esto generalmente incluye una unidad de control, varios teléfonos y un correo de llamadas. Estos se venden en varios tamaños dependiendo del negocio.

- **Cableado e Instalación:** Los cables de instalación de un edificio no terminado puede ser poco costoso. Sin embargo, la instalación de cableado a través de paredes ya terminadas puede añadir un poco más al costo total. Si la oficina está en un sitio donde existe cableado, es preciso rehusarse para ahorrar costos.
- **Otras razones:** Esto incluye entrenamiento, programación, servicio y futuras modificaciones. El precio esta basado usualmente en el tiempo que estas tareas requerirán. Es mejor comparar las horas que se usarán en completar las tareas de programación, servicios, etc.

5.1.2 Precios de sistemas telefónicos muy pequeños

Compradores con menos de 10 empleados reportan precios entre \$2,000 a \$5,000 (9). Esto también incluye correo de voz y características de auto-respuesta:

- DS2000 NEC Telephone System, con 5 teléfonos e instalación: \$2,000.
- Nortel Norstar CICS T7316E con Call Pilot, 100 correos de voz, identificación de llamadas y 5 teléfonos: \$1,915
- Merling Legend, 15 teléfonos, 8 líneas, correo de voz, auto respuesta y música de espera: \$4,500
- Switch, correo de voz y 8 teléfonos: \$2,399

5.1.3 Precios de sistemas telefónicos pequeños

Los negocios entre 10 a 20 empleados que necesitan solo necesidades telefónicas básicas tales como marcado directo, y música en espera pueden usualmente encontrar sistemas adecuados entre \$3,000 a \$10,000 (9). Todos esos compradores reportan obtener correos de voz en sus sistemas telefónicos y la mayoría incluyen auto-respuesta y música en espera:

- 9 teléfonos, correo de llamadas y mensajes automatizados: \$3,950.
- Sistema telefónico de 16 teléfonos: \$6,000
- Sistema de correo de llamadas básico, conferencia, 10 teléfonos de 6 botones, 14 de 18 botones, 1 de 34 botones para recepción, batería de backup, instalación, etc. \$8,200.
- Hardware de sistema telefónico completo, contrato de servicio y seteo de línea: \$5,000.

5.1.4 Precios de sistemas telefónicos de mediano tamaño

Los compra de sistemas telefónicos mas costosos contienen largos sistemas de integración con computadora (telefonía por computadora). CTI es la tecnología detrás de aplicaciones como "muestra automática de llamadas" y otras.

Estos largos sistemas incluyen correo de voz, mensajes en espera y auto-respuesta. Estos cuestan decenas de miles de dólares, frecuentemente alrededor de \$1,000 por extensión (9).

- 25 teléfonos, 3 unidades sin cable, 6 sistemas de parlantes, 20 extensiones CTI. \$18,000.
- Sistema telefónico con correo de llamadas, PoE switches, cableado, entrenamiento, 3 años de soporte (50-69 extensiones CTI). \$41,000.
- Medical Data Provider, Teléfonos, hardware de soporte, software, 20-29 extensiones CTI. \$32,000

5.2. Costo de Instalación del Sistema Propuesto

El sistema propuesto tiene una cuota de mercado que puede ser atractiva para medianas y grandes empresas, que necesiten una serie de extensiones, y que desean tener disponibles varias aplicaciones como conferencias, correo de voz, sistemas de auto-respuesta, etc.

El sistema es completamente flexible y se acomoda a las necesidades de empresas de diversos tamaños, sin embargo es más usado en las medianas y grandes empresas. Al ser un sistema de fuente abierta, es completamente gratuito. El costo estaría en el hardware que debería ser usado para instalar la aplicación.

Por lo tanto el costo es el de una computadora personal, que depende de las necesidades de los negocios, pero una apropiadamente buena estaría en el orden de los \$500 y una tarjeta Digium apropiada para manejar un par de líneas analógicas esta en el orden de los \$300.

Estos costos como se ven, representa todo lo necesario para construir el sistema, que puede incluso correr sobre softphone en lugar de uso de terminales de teléfono. Los teléfonos y otros periféricos pueden usarse dependiendo de las necesidades de cada negocio.

En general, el costo no es muy alto, y es menor aún si se tiene en consideración que ya se posee una computadora con la potencia necesaria. El valor agregado es que tendrá un servidor capaz de conectarse a todas las tecnologías empleadas en telecomunicaciones, con aplicaciones de las más variadas y complejas.

La instalación del software, aunque requiere de la intervención de un Administrador Linux, se está volviendo cada vez más fácil, especialmente luego de que el proyecto Trixbox (<http://www.trixbox.org>) hizo una compilación de las mejores aplicaciones y herramientas de Asterisk y se creó un CD que hace la instalación muy simple y sencilla de ejecutar. El "bundle" Trixbox contiene todas las herramientas necesarias para convertir a una empresa en un proveedor de Voz sobre IP, con software de llamadas, medición de tarifas, log de llamadas y otras funciones. Vienen además incluidas aplicaciones para ERP/CRM que cambian la manera de enfocar los negocios en las empresas de hoy en día. Entre ellas SugarCRM, que es el líder en productos de Manejo de Relaciones de Cliente, completamente integrado a Asterisk.

Todas estas aplicaciones le dan mucho valor añadido a esos \$800 dólares que podría costar esta solución, haciéndola sumamente atractiva y al alcance de las medianas y grandes empresas.

Mientras más grande sea la empresa, más las necesidades de conexión, integración con otros servicios y más las aplicaciones que se usarían de Asterisk. Esta es una opción que muy pocos se atreverían a despreciar, si es que la conocen.

No todo es color de rosa en el mundo de la fuente abierta, pero si es que se desea implementar esta solución en la empresa es necesario invertir en recursos humanos, en el entrenamiento y el aprendizaje del uso de estas herramientas, como también, si es posible, el desarrollo, como contribución al software actual. El hecho de tener personal calificado, más que una desventaja, podría considerarse como una ventaja en el sentido de que la tecnología es transferida a la empresa. Por tanto, tener personal calificado en el desarrollo es casi tanto como poseer la tecnología para seguir adelante sin depender más que de la comunidad de fuente abierta y los conocimientos obtenidos de la experiencia, más no de alguna empresa de manufactura de productos de telecomunicaciones. Las actualizaciones y subsecuentes beneficios estarían al alcance de la mano, con un completo soporte por parte de la comunidad Asterisk.

CONCLUSIONES

Los estándares están evolucionando a tan rápida velocidad, que para mantenerse con ellos se requiere una habilidad de respuesta rápida a las tecnologías emergentes. Asterisk, cuya virtud es la de ser de fuente abierta, con esfuerzo de desarrollo manejado por la comunidad, encaja muy bien con el desarrollo rápido que la compatibilidad de estándares requiere.

Asterisk no se focaliza en análisis de costo-beneficio o investigación de mercado, sino que evoluciona en respuesta a lo que la comunidad Internet considere necesario.

Por lo expuesto anteriormente, se pueden considerar las siguientes conclusiones del presente trabajo:

1. Los sistemas propietarios que proveen soluciones como centrales telefónicas PBX son muchos más caros que las soluciones basadas en tecnologías de fuente abierta. Estas últimas están a su vez disponibles para ser usados en cualquier tipo de configuración que pueda ser posible encontrar.
2. Los sistemas Asterisk convierten al PBX en un servidor que tiene detrás toda la potencia una plataforma UNIX como base para efectuar todo tipo de aplicaciones.
3. Cualquier funcionalidad que esté disponible en otros sistemas propietarios PBX, está también disponible en Asterisk, y con muchas más opciones y facilidad para su configuración y/o personalización. Esto incluye aplicaciones como correos de voz, sistemas de respuesta interactivos, text-to-speech, panel de operación, música en espera, etc.

4. Es posible usar la misma infraestructura telefónica con un servidor Asterisk sin necesidad de cambiarla por redes IP. Para esto es necesario contar con un banco de canales para multiplexar las señales de las distintas extensiones.
5. Los teléfonos IP de Cisco, Nortel, Avaya y Polycom, por mencionar unos cuantos, son conectados fácil y satisfactoriamente a un sistema Asterisk, no existiendo ningún otro PBX que pueda tener similar funcionalidad.
6. En el Perú, la adopción de sistemas Asterisk como centrales telefónicas manejadas por computadora propondría la creación de un nuevo mercado de soporte para este tipo de tecnología, que esta en constante desarrollo y en sincronía con las nuevas tecnologías.
7. La próxima puesta en marcha de una red Wi-Max por AmericaITel en Lima hará posible la conexión a Internet directamente desde muchos dispositivos de mano, incluyendo móviles con soporte SIP, que llegarían a reemplazar los actuales celulares. Estos dispositivos podrían estar conectados a un sistema asterisk para obtener una oficina móvil al alcance de la mano.
8. Deja el camino listo para poder integrarse a nuevas y recientes tecnologías como Google Talk, DID o ENUM.
9. Puede ser configurado en una maquina Linux, donde la configuración es fácilmente realizable a través de una interfaz web.
10. Como todo sistema de fuente abierta, es soportado y actualizado constantemente por la comunidad de Internet y se prevé que los desarrollos sobre este proyecto sean mucho mayores, rápidos y a la vez mucho mas rico en características que cualquiera de las otras propuestas comerciales.
11. Es lo más recomendable en lo que a soluciones de PBX existen actualmente en el mercado, con perspectivas a soportar futuras tecnologías emergentes y además con mucho menor costo de implementación mientras mas grande sea la solución.

Mientras la Red de Telefonía Tradicional Básica siga funcionando en sus últimos años de vida hasta dejar de existir y se convierta en un protocolo más que se transmite por Internet, las tecnologías de fuente abierta, como Asterisk, dominarán esta transformación.

ANEXO A

Archivo zaptel.conf

Valid devices are:

"e&m" : Channel(s) are signalled using E&M signalling (specific

implementation, such as Immediate, Wink, or Feature Group D

are handled by the userspace library).

"fxsls" : Channel(s) are signalled using FXS Loopstart protocol.

"fxsgs" : Channel(s) are signalled using FXS Groundstart protocol.

"fxsks" : Channel(s) are signalled using FXS Koolstart protocol.

"fxols" : Channel(s) are signalled using FXO Loopstart protocol.

"fxogs" : Channel(s) are signalled using FXO Groundstart protocol.

"fxoks" : Channel(s) are signalled using FXO Koolstart protocol.

"sf" : Channel(s) are signalled using in-band single freq tone.

defaultzone=us

fxoks = 1

fxsks = 2

fxoks = 3

fxsks = 4

ANEXO B

Archivo zapata.conf

[channels]

language=en

context=internal

signalling=fxo_ks

usecallerid=yes

echocancel=yes

echocancelwhenbridged=yes

relaxdtmf=yes

callerid="FAX" <333>

threewaycalling=yes

transfer=yes

callgroup=1

pickupgroup=1

channel => 1

language=en

progzone=eg

;callprogress=yes

busycount=5

hanguponpolarityswitch=yes

busydetect=yes

callerid=asreceived

context=pstn

faxdetect=incoming

signalling=fxs_ks

usecallerid=yes

echocancel=yes

echocancelwhenbridged=yes

relaxdtmf=yes

callgroup=1

pickupgroup=1

channel => 2

language=en

context=internal

signalling=fxo_ks

usecallerid=yes

echocancel=yes

echocancelwhenbridged=yes

relaxdtmf=yes

mailbox=100@internal

adsi=yes

callwaiting=yes

callwaitingcallerid=yes

callreturn=yes

threewaycalling=yes

transfer=yes

callerid="Reception" <100>

callgroup=1

pickupgroup=1

channel => 3

language=en

progzone=eg

;callprogress=yes

busycount=5

hanguponpolarityswitch=yes

busydetect=yes

callerid=asreceived

context=fax

faxdetect=incoming

signalling=fxs_ks

usecallerid=yes

echocancel=yes

echocancelwhenbridged=yes

relaxdtmf=yes

callgroup=1

pickupgroup=1

channel => 4

group=1

channel => 2,4

ANEXO C

Parte de la configuración del **archivo sip.conf**

```
;***** Protocol definitions *****
```

```
[general]
```

```
;----- general setup
```

```
port = 5060
```

```
bindaddr = 212.103.170.132
```

```
tos = 0x14
```

```
;----- codecs setup
```

```
disallow = all
```

```
allow=gsm
```

```
;allow=g729
```

```
allow=ulaw
```

```
;----- other options
```

```
;context = internal
```

```
;maxexpirey=180
```

```
;defaultexpirey=160
```

```
;----- register to peers
```

```
context=voicetrading
```

```
nat=yes
```

```
;promiscdir = yes
```

```
canreinvite=yes
```


externip = 212.103.170.132

localnet = 192.168.1.0/24

register=655342370:gtr6KTD1@213.61.187.154

qualify=yes

[voicetrading]

type=peer

allow=gsm

secret=go!gfTm1

username=655349870

fromuser=655349870

host=213.61.187.154

;realm=213.61.187.154

nat=yes

.***** Users *****
,

[101]

username = 101

accountcode = 101

host = dynamic

nat=yes

canreinvite=yes

mailbox = 101@internal

secret = 1234

context = internal

fromuser = 101

login = 101

type = friend

pickupgroup=1

callgroup=1

[102]

username = 102

accountcode = 102

host = dynamic

nat=yes

canreinvite=yes

mailbox = 102@internal

secret = 1234

context = internal

fromuser = 102

login = 102

type = friend

pickupgroup=1

callgroup=1

[111]

username = 111

accountcode = 111

host = dynamic

nat = yes

context = internal

fromuser = 111

login = 111

type = friend

mailbox = 111@internal

secret = 1234

canreinvite=yes

pickupgroup=1

callgroup=1

[103]

username = 103

accountcode = 103

host = dynamic

nat = yes

context = internal

fromuser = 103

login = 103

type = friend

mailbox = 103@internal

secret = 1234

canreinvite=yes

pickupgroup=1

callgroup=1

[104]

username = 104

accountcode = 104

host = dynamic

nat = yes

context = internal

fromuser = 104

login = 104

type = friend

mailbox = 104@internal

secret = 1234

canreinvite=yes

pickupgroup=1

callgroup=1

[105]

username = 105

accountcode = 105

host = dynamic

nat = yes

context = internal

fromuser = 105

login = 105

type = friend

mailbox = 105@internal

secret = 1234

canreinvite=yes

[106]

username = 106

accountcode = 106

host = dynamic

nat = yes

context = internal

fromuser = 106

login = 106

type = friend

mailbox = 106@internal

secret = 1234

canreinvite=no

pickupgroup=1

callgroup=1

ANEXO D

Archivo extensions.conf

```
***** General options *****
```

```
[general]
```

```
static=yes
```

```
writeprotect=yes
```

```
***** Globals values *****
```

```
[globals]
```

```
FWDUSERID=581649
```

```
FWDUSERNAME=581649
```

```
PHONE1=100
```

```
PHONE1VM=100
```

```
FWDEXTEN=100
```

```
FWDGW=IAX2/581649@fwd-gw
```

```
VTUSERID=test
```

```
VTUSERNAME=test
```

```
VTGW=IAX2/test:test@iax.voicetrading.com
```

```
VTSUSERID=655349870
```

```
VTSUSERNAME=gu6RTm1
```

```
VTSGW=IAX2/625449870:guFGTm1@iax.voicetrading.com
```

```
VTSGWSIP=655349870:gu6RTm1@213.61.187.154
```

```
***** DIAL PLAN *****
```


[macro-oneline]

exten => s,1,Dial(\${ARG2},20) ;dial extension number

exten => s,2,Voicemail(u\${ARG1}) ;if unavailable play voicemail un

exten => s,3,hangup

exten => s,102,Voicemail(b\${ARG1}) ;if busy play vm b

exten => s,103,Hangup

;-----

[fwd-out]

exten => _3XX.,1,SetCIDNum(\${FWDUSERID})

exten => _3XX.,2,SetCIDName(\${FWDUSERNAME})

exten => _3XX.,3,Dial(\${FWDGW}/\${EXTEN:1},60,r)

exten => _3XX.,4,Playback(please-try-again)

exten => _3XX.,5,Hangup

[fwd-incoming]

exten => 581649,1,Goto(pstn,s,2)

[iax.voicetrading.com]

;exten => _8.,1,Authenticate(1234)

exten => _8.,2,SetCIDNum(\${VTUSERID})

exten => _8.,3,SetCIDName(\${VTUSERNAME})

exten => _8.,4,Dial(\${VTGW}/\${EXTEN:1})

exten => _8.,5,Playback(invalid,please-try-again)

exten => _8.,6,Hangup

[iax.voicetrading.com.]

exten => _5.,1,Authenticate(0710)

exten => _5.,2,SetCIDNum(\${VTSUSERID})

exten => _5.,3,SetCIDName(\${VTSUSERNAME})

exten => _5.,4,Dial(\${VTSGW}/\${EXTEN:1})

exten => _5.,5,Playback(invalid)

exten => _5.,6,Hangup

[iax.voicetrading.com.sip]

exten => _6.,1,Authenticate(0710)

exten => _6.,2,SetCIDNum(\${VTSUSERID})

exten => _6.,3,SetCIDName(\${VTSUSERNAME})

exten => _6.,4,Dial(SIP/\${EXTEN:1}@voicetrading)

exten => _6.,5,Playback(invalid)

exten => _6.,6,Hangup

[iaxtel]

exten => _1700NXXXXXX,1,Dial(IAX2/itsyn:letmein@iaxtel.com/\${EXTEN}@iaxtel)

exten => _1888NXXXXXX,1,Dial(IAX2/itsyn:letmein@iaxtel.com/\${EXTEN}@iaxtel)

exten => _1877NXXXXXX,1,Dial(IAX2/itsyn:letmein@iaxtel.com/\${EXTEN}@iaxtel)

exten => _1866NXXXXXX,1,Dial(IAX2/itsyn:letmein@iaxtel.com/\${EXTEN}@iaxtel)

exten => _1800NXXXXXX,1,Dial(IAX2/itsyn:letmein@iaxtel.com/\${EXTEN}@iaxtel)

[pstn]

;ignorepat => 9

exten => _90X.,1,Authenticate(5656)

exten => _90X.,2,AbsoluteTimeout(0)

exten => _90X.,3,Dial(zap/g1/\${EXTEN:1})

exten => _9XX.,1,AbsoluteTimeout(0)

exten => _9XX.,2,Dial(zap/g1/\${EXTEN:1})

exten => _9XX.,3,hangup

exten => _9XX.,103,playback(please-try-again)

exten => _9XX.,104,playtones(busy)

exten => _9XX.,105,hangup

exten => _90X.,104,playback(please-try-again)

exten => _90X.,105,playtones(busy)

exten => s,1,answer

exten => s,2,NoOp,\${CALLERID}

exten => s,3,setcallerid(\${CALLERID})

;exten => s,4,background(en/A1)

exten => s,4,wait(1)

exten => s,5,background(welcome)

;exten => s,5,background(diana/A1)

;exten => s,6,gotoif(\${night}=1)?after,s,1:s,7)

```
;exten => s,7,gotoiftime(17:00-9:00|sun-thu|*|*?after,s,1) ;afterhours

;exten => s,8,gotoiftime(*|fri-sat|*|*?after,s,1)

;exten => s,9,playback(giggle1)

exten => *,1,directory(internal)

exten => s,6,WaitMusicOnHold(2)

exten => s,7,goto(en,s,8)

exten => t,1,goto(en,s,8)

exten => i,1,playback(invalid)

exten => i,2,goto(s,2)

;exten => 1,1,goto(en,s,1)

;exten => 2,1,goto(ar,s,1)

;exten => 3,1,dial(zap/97)

exten => 77*,1,authenticate(7777)

exten => 77*,2,setglobalvar(night=1)

exten => 66*,1,authenticate(7777)

exten => 66*,2,setglobalvar(night=0)

;exten => 0,1,dial(zap/3,30) ;commented to add the voicemail macro

exten => 0,1,gotoiftime(19:00-9:00|sun-thu|*|*?after,s,1) ;afterhours

exten => 0,2,gotoiftime(*|fri-sat|*|*?after,s,1)

exten => 0,3,Macro(online,101,sip/101)

exten => fax,1,dial(zap/1,30)
```

exten => fax,2,hangup

include => internal

[fax]

ignorepat => 9

exten => s,1,dial(zap/1,30)

exten => s,2,hangup

exten => _90X.,1,Authenticate(5656)

exten => _90X.,2,AbsoluteTimeout(0)

exten => _90X.,3,Dial(zap/G1/\${EXTEN:1})

exten => _9XX.,1,AbsoluteTimeout(0)

exten => _9XX.,2,Dial(zap/G1/\${EXTEN:1})

exten => _9XX.,3,hangup

exten => _9XX.,103,playback(please-try-again)

exten => _9XX.,104,playtones(busy)

exten => _9XX.,105,hangup

exten => _90X.,104,playback(please-try-again)

exten => _90X.,105,playtones(busy)

[after]

exten => s,1,dial(sip/102,10)

exten => s,2,macro(online,111,sip/111)

exten => s,3,hangup

[ar]

exten => s,1,wait(1)

exten => s,2,background(diana/C1)

exten => s,3,wait(1)

exten => s,4,background(diana/C1)

exten => s,5,wait(1)

exten => s,6,background(diana/C1)

exten => 0,1,Macro(online,101,sip/101)

exten => 0,2,hangup

exten => 1,1,goto(sales,s,1)

exten => 2,1,goto(tech,s,1)

exten => 3,1,authenticate(4569)

exten => 3,2,goto(menu,s,1)

exten => s,7,background(diana/D1)

exten => s,8,Macro(online,101,sip/101)

exten => s,9,hangup

exten => t,1,hangup

exten => h,1,hangup

exten => 9,1,goto(pstn,s,2)

[sales]

exten=>s,1,background(diana/D2)

exten=>1,1,background(diana/M0)

exten=>1,2,goto(sales,s,1)

exten=>2,1,background(diana/m1)

exten=>2,2,goto(sales,s,1)

exten=>9,1,goto(ar,s,4)

exten=>s,2,background(giha/D2)

exten=>s,3,Macro(online,101,sip/101)

[tech]

exten=>s,1,background(diana/E1)

exten=>s,2,authenticate(1234)

exten=>s,3,background(diana/SUP1)

exten=>00,1,background(diana/Fn)

exten=>9,1,goto(ar,s,4)

exten=>s,103,Macro(online,101,sip/101)

[en]

exten => s,1,wait(1)

exten => s,2,background(en/C1)

exten => s,3,wait(1)

exten => s,4,background(en/C1)

exten => s,5,wait(1)

exten => s,6,background(en/C1)

exten => 0,1,Macro(online,101,sip/101)

exten => 0,2,hangup

;exten => 1,1,Goto(enmenu,s,1)

exten => 1,1,goto(ensales,s,1)

exten => 2,1,goto(entech,s,1)

exten => 3,1,authenticate(4569)

exten => 3,2,goto(enmenu,s,1)

exten => s,7,background(en/D1)

exten => s,8,Macro(online,102,sip/102)

exten => s,9,hangup

exten => t,1,hangup

exten => h,1,hangup

exten => 9,1,goto(pstn,s,2)

[ensales]

exten=>s,1,background(en/D2)

exten=>1,1,background(en/M0)

exten=>2,1,background(en/M1)

exten=>s,2,background(en/D2)

exten=>s,3,Macro(online,101,sip/101)

exten=>9,1,goto(en,s,4)

[entech]

exten=>s,1,background(en/E1)

exten=>s,2,authenticate(1234)

exten=>s,3,background(en/SUP1)

exten=>00,1,background(en/Fn)

exten=>9,1,goto(en,s,4)

exten=>s,103,Macro(online,101,sip/101)

[menu]

exten => s,1,background(diana/news.wav)

exten => s,2,background(diana/sys.wav)

exten => s,3,background(diana/dial_0_return_main_menu.wav)

exten => 1,1,background(news.WAV)

exten => 1,2,goto(s,1)

exten => 2,1,background(cpu-util.WAV)

exten => 2,2,goto(s,1)

exten => i,1,goto(s,1)

exten => s,4,wait(2)

exten => s,5,background(diana/news.wav)

exten => s,6,background(diana/sys.wav)

exten => s,7,background(diana/dial_0_return_main_menu.wav)

exten => s,8,hangup

exten => 0,1,goto(ar,s,2)

```
;exten => t,1,hangup
```

```
exten => h,1,hangup
```

```
[enmenu]
```

```
exten => s,1,background(en/news-serverstat.WAV)
```

```
exten => s,2,background(en/returntoprev.WAV)
```

```
exten => 1,1,background(news.WAV)
```

```
exten => 1,2,goto(s,1)
```

```
exten => 2,1,background(cpu-util.WAV)
```

```
exten => 2,2,goto(s,1)
```

```
exten => i,1,goto(s,1)
```

```
exten => s,3,wait(2)
```

```
exten => s,4,background(en/news-serverstat.WAV)
```

```
exten => s,5,background(en/returntoprev.WAV)
```

```
exten => s,6,hangup
```

```
exten => 0,1,goto(en,s,2)
```

```
exten => h,1,hangup
```

```
[mmm]
```

```
exten => s,1,Background(menu)
```

```
exten => 1,1,Playback(weather-station)
```

```
exten => 2,1,Dial(${EXTEN:1},20) ;dial extension number
```

```
exten => 3,1,Playback(all-your-base)
```

[internal]

exten => 999,1,answer

exten => 999,2,musiconhold()

exten => 122,1,meetme(122||)

exten => 444,1,authenticate(1616)

exten => 444,2,read(gsm,vm-enter-num-to-call,10)

exten => 444,3,system(/usr/bin/gsmsendsms -d /dev/ttyS0 -b 19200 \${gsm} "This is an SMS test from Evolve.we will have to figure out how to easily change the text.")

exten => 444,4,playback(privacy-thankyou)

exten => 444,5,hangup

;exten => 5656,1,DTMFToText(||)

exten => 11,1,playtones(dial) ;for testing call progress tones

exten => 12,1,playtones(busy) ;for testing call progress tones

exten => 13,1,playtones(ring) ;for testing call progress tones

exten => 14,1,playtones(congestion) ;for testing call progress tones

exten => 15,1,playtones(waiting) ;for testing call progress tones

exten => 7777,1,Goto(temp,s,1)

exten => 7778,1,Goto(motion,s,1)

exten => 9966,1,Goto(mmm,s,1)

exten => 101,1,Macro(online,101,SIP/101)

exten => 102,1,Macro(online,102,SIP/102)

```
exten => 103,1,Macro(online,103,SIP/103)

exten => 104,1,Macro(online,104,SIP/104)

exten => 105,1,Macro(online,105,SIP/105)

exten => 106,1,Macro(online,106,SIP/106)

exten => 107,1,Macro(online,107,SIP/107)

exten => 108,1,Macro(online,108,SIP/108)

exten => 109,1,Macro(online,109,SIP/109)

exten => 333,1,Macro(online,333,ZAP/1)

exten => 111,1,Macro(online,111,SIP/111)

exten => 100,1,Macro(online,100,zap/3)

exten => 00,1,Macro(online,00,zap/1)

exten => 666,1,Dial(IAX2/server2:1234@server2/104)

exten => 677,1,goto(pstn,s,1)

exten => 201,1,Macro(online,201,IAX2/201)

exten => 202,1,Macro(online,202,IAX2/202)

exten => 203,1,Macro(online,203,IAX2/203)

exten => 204,1,Macro(online,204,IAX2/204)

exten => 205,1,Macro(online,205,IAX2/205)

exten => 206,1,Macro(online,206,IAX2/206)

exten => 207,1,Macro(online,207,IAX2/207)

exten => 208,1,Macro(online,208,IAX2/208)
```

```
exten => 209,1,Macro(online,205,IAX2/209)

exten => 5000,1,Macro(online,5000,IAX2/5000)

exten => 555,1,VoicemailMain(${CALLERIDNUM})

exten => 500,1,Playback(demo-abouttotry)

exten => 500,2,Dial(IAX2/guest@misery.digium.com/s@default)

exten => 500,3,Playback(demo-nogo)

exten => 500,4,Goto(s,6)

exten => 600,1,Playback(demo-echotest)

exten => 600,2,Echo

exten => 600,3,Playback(demo-echodone)

include => pstn

;exten => 9999,1,AGI(fastsms|20101636337|My maintainer is mollesting me, please help
me out of this misery...|na6oosh)

;exten => ${FWDEXTEN},1,Dial(${PHONE1},30)

;exten => ${FWDEXTEN},2,Voicemail(u${PHONE1VM})

;exten => ${FWDEXTEN},3,Hangup

;exten => ${FWDEXTEN},102,Voicemail(b${PHONE1VM})

;exten => ${FWDEXTEN},103,Hangup

;include => record-outboundmsgs

; at this point we could do something like reschedule the call to try again later

; or send an email saying the msg was not received,
```

; or ...

[outboundmsg2]

exten => s,1,DigitTimeout,5 ; Set Digit Timeout to 5 seconds

exten => s,2,ResponseTimeout,10 ; Set Response Timeout to 10 seconds

exten => s,3,Answer

exten => s,4,Wait(1)

exten => s,5,Background(weather) ; "play outbound msg"

exten => s,6,Background(weather-station) ; "Press 1 to replay or 2 to acknowledge receiving this message"

exten => 1,1,Goto(s,5) ; replay message

exten => 2,1,Goto(msgack,s,1) ; acknowledge message

exten => t,1,Playback(vm-goodbye)

exten => t,2,Hangup

; at this point we could do something like reschedule the call to try again later

; or send an email saying the msg was not received,

[msgack]

exten => s,1,Playback(outboundmsgs/thankyou)

exten => s,2,Playback(vm-goodbye)

exten => 2,1,Goto(msgack,s,1) ; acknowledge message

exten => t,1,Playback(vm-goodbye)

exten => t,2,Hangup

; at this point we could do something like reschedule the call to try again later

; or send an email saying the msg was not received,

[msgack]

exten => s,1,Playback(outboundmsgs/thankyou)

exten => s,2,Playback(vm-goodbye)

exten => 2,1,Goto(msgack,s,1) ; acknowledge message

exten => t,1,Playback(vm-goodbye)

exten => t,2,Hangup

; at this point we could do something like reschedule the call to try again later

; or send an email saying the msg was not received,

; or ...

[msgack]

exten => s,1,Playback(outboundmsgs/thankyou)

exten => s,2,Playback(vm-goodbye)

exten => s,3,Hangup

; at this point we might want to log the message acknowledgement somewhere

; and perhaps trigger some additional processing

[record-outboundmsgs]

; Record voice files

; Before using this the first time

mkdir /var/lib/asterisk/sounds/outboundmsgs

```
chown asterisk_user:asterisk_user /var/lib/asterisk/sounds/outboundmsgs
```

(Where asterisk_user = the user that asterisk runs under: = root for many installations)

```
; In a context for incoming calls put something like
```

```
; include => record-outboundmsgs
```

```
; Then call
```

```
2051 to Record a new outbound msg1
```

```
2052 to Record a new outbound msg2
```

```
2061 to Record the msg played when the recipient acks the message
```

```
2062 to Record the "How to ACK message"
```

```
; After dialing one of the extensions above:
```

```
=; Wait for the record start tone
```

```
Record your message
```

```
Press # to stop recording
```

```
Listen to an automatic playback of your new message
```

```
; outbound msg1
```

```
exten => 2051,1,Wait(2)
```

```
exten => 2051,2,Record(outboundmsgs/msg1:gsm)
```

```
exten => 2051,3,Wait(2)
```

```
exten => 2051,5,Playback(weather)
```

```
exten => 2051,4,Playback(outboundmsgs/msg1)
```

```
exten => 2051,6,wait(2)
```


exten => 2051,7,Hangup

; outbound msg2

exten => 2052,1,Playback(weather)

exten => 2052,2,Record(outboundmsgs/msg2:gsm)

exten => 2052,3,Wait(2)

exten => 2052,4,Playback(outboundmsgs/msg2)

exten => 2052,5,wait(2)

exten => 2052,6,Hangup

; Msg played when msg is acked

exten => 2061,1,Wait(2)

exten => 2061,2,Record(outboundmsgs/thankyou:gsm)

exten => 2061,3,Wait(2)

exten => 2061,4,Playback(outboundmsgs/thankyou)

exten => 2061,5,wait(2)

exten => 2061,6,Hangup

; Msg played after outbound msg: "Press 1 to replay or 2 to acknowledge receiving this message"

***** END OF FILE *****

BIBLIOGRAFIA

- [1] Jane Laino, "The Telecom Handbook: Understanding Telephone Systems and Services", R&D - UK, Nov. 2002 - ISBN: 1578200717
- [2] Harry Nyquist, "Certain Factors Affecting Telegraph Speed", Bell System Technical Journal (1924)
- [3] Claude Shannon, "Communication in the Presence of Noise", Institute of Radio Engineers – 1949 - ASIN: B0007EOIMA.
- [4] Jonathan Davidson, James Peters and Brian Grace, "Voice over IP Fundamentals", Cisco Press, March 27, 2000 - ISBN-10: 1-57870-168-6
- [5] Frederick Brooks, "The Mythical Man-Month: Essays on Software Engineering", Addison-Wesley Professional, August 1995 - ISBN: 0201835959
- [6] Eric S. Raymond, "The Cathedral and the Bazaar", The Linux Kongress, May 1997 - <http://catb.org/esr/writings/cathedral-bazaar/>
- [7] Mark Spencer, Mack Allison and Christopher Rhodes, "The Asterisk HandBook", Digium Inc. - March 2003
- [8] David Comillion and Barrie Dempster, "Building Telephony Systems with Asterisk" – Packt Publishing - September 2005
- [9] Buyerzone – "Guía de Compradores y cotizaciones de Telefonía de Oficina" - http://www.buyerzone.com/telecom_equipment/phone_systems/bps-phone-system-prices.html

[10] Wikipedia – <http://www.wikipedia.org>