

**Universidad de las Ciencias Informáticas
Facultad 2**



Título: Redes de Voz sobre IP mediante PBX basado en Software.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores: Adilenys Suárez Pérez

Darvis Dorvigny Dorvigny.

Tutora: Ing. Maria Luisa Herrera Corbelle.

Junio 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2008.

Adilenys Suarez Pérez

Darvis Dorvigny Dorvigny

Maria Luisa Herrera Corbelle

Firma del Autor

Firma del Autor

Firma del Tutor

AGRADECIMIENTOS

Adilenys Suárez Pérez

Primeramente quisiera agradecer a toda mi familia, por haber estado a mi lado brindándome apoyo, por haber hecho de mí una mujer de bien al inculcarme en todo momento, responsabilidad, respeto, sencillez y sobre todo amor hacia todo lo que hago. Agradezco a mi mami por estar siempre a mi lado, por esos consejos tan llenos de amor y sabiduría en momentos tan necesarios. Agradezco a todos mis compañeros de aula y sobre todo a mis hermanitas del 1439 donde pasé el mejor curso de la universidad. A todas aquellas personas que siempre estuvieron a mi lado dándome apoyo y amor en momentos bien difíciles para mí. A mi amigo Darien, a Made, Iraliya y muchos otros que dejaron en mí profundas huellas; a mi tutora, por todo el tiempo que nos dedicó y soportó, por su paciencia y entrega; a mi compañero de tesis, por esos momentos de difícil comprensión, de arduo trabajo, de alegría y desesperaciones...; a nuestra Revolución cubana por haber hecho posible este Proyecto Futuro y en general a todas aquellas personas que de una manera u otra han hecho posible que haya logrado este hermoso sueño de convertirme en una **Ingeniera de Ciencias Informáticas**.

Darvis Dorvigny Dorvigny

Este ejercicio de culminación de estudios no tendría sentido si no existiera este maravilloso Proyecto Futuro, por tanto mi primer agradecimiento es a la Revolución. Agradezco la preocupación constante de toda mi familia, en especial de mis padres; de los amigos y vecinos, aquellos que siempre esperan más de mí, y me comprometen a seguir adelante. Agradezco a mis hermanos entrañables de la universidad, Yordan, Alain, Amauri, Javier, por el aliento, y la ayuda incondicional; a nuestra tutora, por guiarnos durante todo el proceso de investigación, por sus consejos siempre oportunos, por sus ideas, y por el tiempo que dedicó a compartir sus conocimientos y experiencias con nosotros; a mi compañera de tesis por el aliento constante, por mantenerme fuera del estado inercial que me caracteriza, y por soportarme. Gracias a todos quienes esperan siempre de mí cosas buenas.

DEDICATORIA

Adilenys Suárez Pérez

A mis padres por confiar en mí, en todo momento y por ser lo primero en mi vida.

A todas las personas que estuvieron a mi lado durante mi carrera.

A mis amigos por su incondicionalidad en todo momento.

Darvis Dorvigny Dorvigny

A la Revolución, sin ella no podría aspirar a formarme como Ingeniero en Ciencias Informáticas.

A mis padres, por estar conmigo siempre, por sus consejos, y por su apoyo incondicional.

A mi hermanita, por ser una razón para vivir.

A mi novia, por esperarme.

A mis UCI-Hermanos.

RESUMEN

La VoIP viene evolucionando como una tecnología robusta, con muchas perspectivas. Su arquitectura es sólida, y está basada en conceptos que hacen converger las redes de telefonía y las redes de datos, unificando los servicios y aumentando las prestaciones. En esta investigación se abordan temas referentes a la tecnología VoIP, sus características y conceptos fundamentales. Se muestran varias soluciones que implementan la VoIP a partir del concepto de PBX basado en software. En particular se detallan las características y funciones del SoftPBX Asterisk, así como consideraciones acerca de la viabilidad de su implantación en la UCI.

Finalmente se realiza una propuesta para la implantación de Asterisk en la Universidad de las Ciencias Informáticas, como forma de introducir los conceptos de la Voz sobre IP. Esta propuesta está inspirada en la necesidad de ampliar las capacidades de comunicación telefónica que brinda la universidad, a partir de la incapacidad real de la Central Telefónica instalada de brindar servicios a los nuevos abonados que surgen en la ampliación inmobiliaria de la universidad.

PALABRAS CLAVE: VoIP, SoftPBX, códec, protocolos de señalización, Asterisk

TABLA DE CONTENIDOS

AGRADECIMIENTOS	I
DEDICATORIA	II
RESUMEN	III
INTRODUCCIÓN.....	1
CAPÍTULO 1: TECNOLOGÍA DE VOZ SOBRE EL PROTOCOLO IP	3
1.1 Descripción del funcionamiento de la tecnología VoIP	3
1.2 Codificadores de audio utilizados en VoIP	4
1.3 Protocolos de tiempo real sobre IP	6
1.3.1 Protocolo de Tiempo Real (RTP).....	6
1.3.2 Protocolo de Control en Tiempo Real (RTCP).....	6
1.4 Protocolos de señalización.....	7
1.4.1 H.323	7
1.4.2 SIP	7
1.4.3 MGCP	8
1.4.4 IAX	9
1.5 Ventajas de la VoIP respecto a la telefonía tradicional.....	9
1.6 Problemas en la VoIP.....	10
1.7 Calidad de Servicio en la VoIP	10
1.8 Centralitas PBX.....	11
1.8.1 Intranet PBX (IPBX)	12
1.8.2 SoftPBX.....	13
1.9 Conclusiones del capítulo	19
CAPÍTULO 2: ASTERISK	21
2.1 Introducción a Asterisk	21
2.2 Características y servicios soportados por Asterisk.....	21
2.3 Arquitectura de Asterisk	24
2.3.1 API de canal, (Channel API).....	25

2.3.2 API para traducción de código.....	26
2.3.3 API para formato de ficheros	26
2.3.4 API de aplicaciones, (Application API).....	26
2.4 El Plan de Discado de Asterisk.	27
2.4.1 Extensiones.....	27
2.4.2 Contextos	28
2.4.3 Prioridades	29
2.4.4 Aplicaciones	29
2.4.5 Un pequeño plan de discado	30
2.5 Estructura de conexión lógica	30
Estructura Árbol Jerárquico	31
Estructura <i>Peer To Peer</i> (P2P).....	31
2.6 Consideraciones sobre rendimiento de Asterisk.....	32
2.7 Conclusiones del capítulo	34
CAPÍTULO 3: PROPUESTA DE IMPLANTACION DE ASTERISK EN LA UCI	35
3.1 Requerimientos de hardware y software	35
3.1.1 Hardware para los servidores.....	35
3.1.2 Sistema Operativo.....	36
3.1.3 Versión de Asterisk	36
3.2 Ubicación física de los servidores	36
3.3 Ubicación de los servidores en la red.....	37
3.3 Interconexión de los servidores.....	40
3.4 Plan de numeración	42
3.5 Protocolos de señalización.....	42
3.6 Códecs.....	43
3.8 Pruebas de señalización	44
3.9 Conclusiones del capítulo	46
CONCLUSIONES	47
RECOMENDACIONES	48
BIBLIOGRAFÍA.....	49

ANEXOS.....	51
A. Instalación de Asterisk	51
B. Organización del sistema de archivos de Asterisk	52
C. Configuración de la música en espera.....	53
D. Variables de canal Predefinidas.	54
E. Interconexión de los servidores mediante IAX2	55
GLOSARIO DE TÉRMINOS	59

INTRODUCCIÓN

Las crecientes necesidades de comunicación en la actual sociedad de la información han propiciado el vertiginoso avance de las telecomunicaciones en las últimas décadas. Los adelantos en esta rama del conocimiento han tenido gran impacto en las pequeñas y grandes empresas, en instituciones científicas, académicas y de salud, que cada vez más requieren de modernos servicios de telecomunicaciones para intercomunicar a las personas, acortando las distancias y optimizando indirectamente la productividad de su gestión. Durante mucho tiempo en la telefonía tradicional se han encontrado soluciones de mucha utilidad para estos fines.

En la telefonía convencional se dispone de las Centrales Telefónicas Privadas, llamadas PBXs (acrónimo de *Private Branch Exchange*), para brindar servicios de telefonía a pequeña escala. Las posibilidades de servicios que son capaces de ofrecer las modernas PBX comerciales disponibles en el mercado son innumerables. Sin embargo, estas soluciones cada vez son más costosas y difíciles de adquirir. Las grandes compañías exigen por sus productos sumas millonarias, y mantienen a sus compradores y clientes sujetos a rigurosas y estrictas licencias. Esto se hace sentir en mayor medida cuando aparecen dificultades intrínsecas a la tecnología, que tardan en ser superadas por las limitaciones económicas.

Las miradas se dirigen nuevamente hacia el mundo de las ciencias informáticas. Con la aparición de la posibilidad de transmitir información de voz sobre la red de datos, surge la alternativa que hace posible reducir los costos de desarrollo e implantación, incrementándose las posibilidades de agregar nuevos servicios de telefonía, sin hacer grandes inversiones en materia de tecnología. Cuba, en su afán de lograr soberanía tecnológica, y de abaratar los costos en la gestión de las telecomunicaciones, está realizando esfuerzos por introducir estos avances tecnológicos.

En la Universidad de las Ciencias Informáticas (UCI) existe una amplia red de telefonía, que brinda servicios a toda la comunidad universitaria, desde la intercomunicación entre los distintos departamentos y oficinas, hasta cubrir el área de la residencia de manera exhaustiva. Por medio de la PBX instalada, además de la comunicación interna entre los abonados telefónicos, se accede a la comunicación con la PSTN (acrónimo de *Public Switched Telephone Network*, Red Conmutada de Telefonía Pública).

Debido al constante crecimiento de la universidad en materia de inmuebles que precisarán inevitablemente de servicios telefónicos, se hace difícil disponer de nuevas capacidades de conexión para nuevos abonados de la telefonía. La causa principal radica en que la PBX instalada ya está al máximo de su capacidad de conexión, y su funcionamiento se vuelve errático en horarios de mayor

Introducción

densidad de llamadas telefónicas. Extender las capacidades de esta pizarra se traduce en grandes inversiones financieras.

Otro problema es la ocurrencia de alguna interrupción en el servicio telefónico dentro de la universidad, debido a desperfectos ocurridos por accidentes de variada naturaleza. En dependencia de la gravedad, la poca disponibilidad de recursos puede hacer que varios usuarios afectados permanezcan incomunicados por un tiempo prolongado. La adquisición del reemplazo para los componentes dañados se traduce en grandes gastos financieros.

Con estas problemáticas cabe preguntarse cómo ampliar los servicios de telefonía en la Universidad de las Ciencias Informáticas.

De acuerdo a las características de la investigación se planteó como objeto de estudio las redes de Voz sobre IP, y se definió como campo de acción los PBXs basados en software como alternativa a los servicios de telefonía.

Se identificó como objetivo general elaborar una propuesta para la implantación de servicios de VoIP (acrónimo de *Voice over Internet Protocol*, Voz sobre Protocolo de Internet) mediante PBX basado en software en la UCI.

Con el objetivo de profundizar en el tema se definieron como tareas de investigación:

- Realización de un estudio sobre los servicios que brindan las redes de Voz sobre IP.
- Realización de un estudio de experiencias en el empleo de PBXs basados en software.
- Realización de un estudio sobre las posibles desventajas de la implementación de un SoftPBX sobre la red de datos.
- Realización de un levantamiento de las características de la red de datos en la UCI.
- Realización de una estimación de las capacidades de conexión del SoftPBX que se elijará para la propuesta.

CAPÍTULO 1: TECNOLOGÍA DE VOZ SOBRE EL PROTOCOLO IP

La tecnología VoIP es una nueva forma de transmitir la voz, utilizando una conexión de Internet en lugar de una línea telefónica común. VoIP hace que una llamada telefónica se pueda realizar a través de la red de datos IP hasta llegar al teléfono de la persona que se está llamando.

En este capítulo se abordarán las características y funcionalidades más importantes de esta tecnología.

1.1 Descripción del funcionamiento de la tecnología VoIP

Para realizar una llamada usando esta tecnología, los dos comunicantes se conectan al servidor VoIP a través de sus teléfonos. Posterior a esto el equipo emisor pregunta por el equipo receptor con un protocolo de señalización determinado. El servidor IP devuelve los datos de contacto al emisor, por ejemplo la dirección de red. Concluida esta operación, los teléfonos establecen conexión y acuerdan un código (*G.711*, *G.729*, *GSM*, entre otros). Los datos se comprimen y se envían en forma de paquetes IP mediante un canal de datos en tiempo real, el receptor recibe los paquetes y los decodifica, y por último se escucha la voz.

La diferencia principal entre esta solución y la telefonía tradicional es que usando esta tecnología no se reservan recursos en la red. Cada paquete con voz digitalizada sabe cual es su destino, y la red se encarga de que llegue sin necesidad de que vaya a través del mismo camino que el resto. Cuando uno de los extremos no habla, simplemente no se envía nada, y la red no se satura con canales que no se utilizan. Además la voz puede ser comprimida, lo cual hace que disminuya el consumo de ancho de banda. Otro argumento a favor de esta forma de comunicación de voz es la existencia de numerosas redes IP funcionando en los entornos corporativos de datos.

Para la transmisión de voz sobre una red IP, el estándar VoIP define tres elementos fundamentales en su estructura:

- Terminales: Son los puntos finales de la comunicación y pueden ser implementados como:
 - *Hardware*: Un teléfono IP es un terminal que tiene soporte nativo con VoIP, y puede conectarse directamente a una red IP. Estos dispositivos cuentan con nuevas características no disponibles en un teléfono convencional. A partir de su complejidad funcional son más costosos.
También es posible conectar un teléfono convencional a la red de datos IP para comunicaciones VoIP por medio de adaptadores que conviertan señales analógicas en paquetes IP, y viceversa.

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

- **Software:** Un *softphone* es un programa de computadora, cuya función es emular el funcionamiento de un teléfono IP. El usuario puede interactuar con el software por medio de un micrófono y un altavoz. Son muy asequibles y muy funcionales. Se encuentran versiones propietarias, y de código abierto indistintamente.
- **Servidor:** Provee el manejo y funciones administrativas para soportar el enrutamiento de llamadas a través de la red. El servidor es un elemento opcional, normalmente implementado en software. En caso de existir, todas las comunicaciones pasarán por él.
- **Pasarelas (*Gateway*):** Es quien establece el enlace de la red VoIP con la PSTN. Se encarga de adaptar las señales de la red VoIP para ser manejadas por la PSTN, y viceversa. Actúa de forma totalmente transparente para el usuario.
- **Red IP:** provee conectividad entre todos los terminales, ésta puede ser de tipo privada, una intranet, o Internet.

Los distintos elementos pueden residir en plataformas físicas separadas o bien pueden convivir en la misma plataforma. Es posible encontrar juntos servidor y *gateway* en un mismo dispositivo físico.

1.2 Codificadores de audio utilizados en VoIP

La señal de audio debe de ser digitalizada, comprimida y codificada antes de ser transmitida por la red IP. Para ello se utilizan algoritmos matemáticos implementados en software llamados códecs (acrónimo de codificador/decodificador). En VoIP se utilizan diferentes códec de audio. En dependencia del algoritmo escogido en la transmisión, variará la calidad de la voz, el ancho de banda necesario para la transmisión, y la carga computacional. El objetivo principal de esta tecnología es encontrar un equilibrio entre eficiencia y calidad de la voz.

Aunque el sistema auditivo humano es capaz de captar las frecuencias comprendidas entre 20 Hz y 20 kHz, la gran mayoría de códecs procesan aquella información dentro de la banda de 400 Hz – 3,5 kHz. Aún en este rango de frecuencias cuando se reconstruye la señal, esta sigue siendo comprensible.

A continuación se describen los códecs más utilizados en VoIP:

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

Nombre	Estándar	Descripción	Ancho de Banda (Kb/s)	Frecuencia (kHz)	Observaciones
G.711	ITU-T	Utiliza PCM para comprimir, descomprimir, codificar y decodificar.	64	8	Tiene dos versiones u-law y a-law para muestrear la señal.
G.726	ITU-T	Utiliza ADPCM	16/24/32/40	8	Disminución de ancho de banda y la carga computacional.
G.723.1	ITU-T	Es muy utilizado en la transmisión de multimedia en 5,3 y 6,3 kbit / s.	5.6/6.3	8	Hay que pagar licencia para utilizarlo.
G.729A	ITU-T	Es una simplificación de G729 y es compatible con él.	8	8	Es un algoritmo de elevada carga computacional.
GSM	ETSI	Consta de una carga de CPU aceptable	13	8	No requiere el pago de una licencia.
iLBC	ETSI	Es un algoritmo complejo desarrollado por Global IP Sound (GIPS).	13.3/15.2	8	Ofrece una buena relación ancho de banda – calidad de voz, a cambio de una mayor carga computacional.
Speex	Xiph.Org Foundation	Implementa un algoritmo capaz de variar la velocidad de transmisión.	2.15 / 22.4	2.15-24.6 (NB) 4-44.2 (WB)	Depende de las condiciones de la red.
MP3	ISO	Está optimizado para música y no para telefonía.			Es utilizado por los teléfonos IP principalmente para ofrecer servicios de música en espera.

Tabla 1 Códecs más usados en la VoIP.

1.3 Protocolos de tiempo real sobre IP

Los protocolos de tiempo real para la transmisión de audio y vídeo por Internet se definen dentro de la RFC 1889. Esta norma incluye dos protocolos que constituyen el estándar: RTP y RTCP. El primero, protocolo RTP (acrónimo de *Real Time Protocol*, Protocolo de Tiempo Real), regula el intercambio de información en diferentes formatos (audio y vídeo). El RTCP (acrónimo de *Real Time Control Protocol*, Protocolo de Control de Tiempo Real) regula la comunicación de control que se establece entre los extremos, en paralelo con la transmisión de información.

1.3.1 Protocolo de Tiempo Real (RTP)

El protocolo RTP es el estándar para el transporte de tráfico (audio o vídeo) en tiempo real sobre Internet. En este protocolo se asume la existencia de pérdidas y retardos en la red y la posibilidad de variación de las características de la red durante la comunicación.

1.3.2 Protocolo de Control en Tiempo Real (RTCP)

El RTCP regula el intercambio de mensajes de control entre los participantes en una sesión multimedia. Esta información se refiere, fundamentalmente, a la calidad de servicio (QoS, acrónimo de *Quality of Service*) con que se está desarrollando la comunicación: retardo (*jitter*), tasa de paquetes recibidos y perdidos. Es una comunicación paralela a la transmisión de información, que se establece entre los extremos de forma opcional. Aunque es opcional, es recomendable porque proporciona el estado actual de la comunicación, determina si la calidad de la transmisión es suficiente y permite tomar las medidas oportunas durante la transmisión de información, por ejemplo, utilizar un códec con menor tasa, o adaptar el tamaño del buffer en recepción.

Es un protocolo para informar de la calidad del servicio pero no puede mejorar las prestaciones de la red, es decir no proporciona mecanismos de calidad de servicio. La utilización de RTCP consume un ancho de banda añadido al RTP. Supone entre 1 y 5% del ancho de banda de RTP. Aunque no es un tráfico excesivo, en sesiones con múltiples participantes es necesario controlarlo para evitar avalanchas de datos.

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

1.4 Protocolos de señalización

La señalización en VoIP tiene un papel muy importante en la red, ya que es la encargada de establecer, mantener, administrar y finalizar una conversación entre dos puntos. Además de ofrecer funciones de supervisión, marcado, llamada y retorno de tonos de progreso; también se encarga de proveer QoS, en cada canal de transmisión. En los siguientes apartados se describe alguno de los protocolos más importantes utilizados en VoIP.

1.4.1 H.323

H.323 es una familia de estándares desarrollado por la ITU (*International Telecommunication Union*, Unión Internacional de Telecomunicación) en 1996 con el objetivo de ofrecer un mecanismo de transporte para servicios multimedia sobre redes que no garantizan calidad de servicio, aunque su uso se ha extendido sobretodo al uso sobre redes IP. Pese a que inicialmente fue definido como un protocolo de videoconferencia, rápidamente ha evolucionado para cubrir todas las necesidades de la VoIP. Se trata de una recomendación bastante cerrada donde se define los códecs a utilizar, tanto en audio como en video, y los protocolos de transporte de la información. Fue el primer estándar en adoptar como medio de transporte el protocolo RTP, siendo capaz de aplicar algoritmos de encriptación de la información, evitando de esta manera añadir elementos de seguridad adicionales a los requeridos para la conexión a Internet.

Pese a que técnicamente es un protocolo potente y maduro, el interés por parte de los usuarios del protocolo y empresas actualmente ha disminuido debido principalmente a su complejidad y a ciertas ineficiencias detectadas.

1.4.2 SIP

SIP (acrónimo de *Session Initial Protocol*, Protocolo de Inicio de Sesión) es un protocolo desarrollado por el IETF (*Internet Engineering Task Force*, Grupo de Tareas de Ingeniería de Internet) en 1999 para el control de llamadas multimedia, la implementación de servicios telefónicos avanzados y la modificación y terminación de sesiones de comunicación multimedia entre usuarios. SIP está basado en HTTP (acrónimo de *HyperText Transfer Protocol*, Protocolo de Transferencia de Hipertexto) adoptando las características más importantes de este estándar como son la sencillez de su sintaxis y una estructura cliente/servidor basada en un modelo petición/respuesta. Otra de las ventajas de SIP es su sistema de direccionamiento. Las direcciones SIP tienen una estructura parecida a la de un correo electrónico dotando a sus clientes de una alta movilidad facilitando una posible integración en

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

comunicaciones móviles. Cabe destacar que aunque originalmente SIP tenía como objetivo la simplicidad, en su estado actual se ha vuelto tan complejo como H.323.

Pero el gran potencial de SIP reside en su flexibilidad ya que ofrece la posibilidad de programar nuevos servicios no definidos por la propia recomendación. Entornos de programación como CGI (acrónimo de *Common Gateway Interface*, Interfaz Común de Pasarela) o sencillos lenguajes de programación como CPL (acrónimo *Call Processing Language*, Lenguaje de Procesamiento de Llamadas), son algunas de las herramientas para la implementación de servicios sin que conlleve a un peligro para la integridad del sistema. Esta es la característica principal por la que SIP actualmente goza de un mayor éxito que H.323.

Los clientes SIP llamados *peers* o *user agents*, usan el puerto 5060 en TCP (acrónimo de *Transmission Control Protocol*, Protocolo de Control de Transmisión) y UDP (acrónimo de *User Datagram Protocol*, Protocolo de Datagrama de Usuario) para conectar con los servidores SIP. Es usado simplemente para iniciar y terminar llamadas de voz y video. Todas las comunicaciones de voz/video van sobre RTP.

El protocolo SIP consta de capacidades tales como:

- Localización del usuario;
- Disponibilidad del usuario: Determinación de la voluntad del receptor de la llamada de participar en las comunicaciones;
- Capacidad del usuario: Determinación del medio y de sus parámetros;
- Gestión de la sesión: Transferencia, terminación de sesiones, modificación de los parámetros de la sesión desde el propio *user agent*.

1.4.3 MGCP

Media Gateway Control Protocol (MGCP, Protocolo de Control de Pasarela y Multimedia) es otro estándar de señalización para VoIP desarrollado por la IETF. MGCP está basado en un modelo maestro/esclavo donde el *Call Agent* (servidor) es el encargado de controlar al *gateway*. De esta forma se consigue separar la señalización de la transmisión de la información. Esta importante ventaja propició la colaboración conjunta entre el IETF y la ITU para el desarrollo de una nueva especificación basada en MGCP que fuera complementaria a SIP y H.323. El resultado fue MEGACO aunque la ITU se refiere a este protocolo como H.248.

MEGACO es óptimo para los grandes operadores de telefonía.

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

1.4.4 IAX

Inter-Asterisk eXchange protocol (IAX) fue desarrollado por la empresa *Digium* para la comunicación entre centralitas *Asterisk* aunque actualmente se ha implementado clientes que también soportan este protocolo. El principal objetivo de IAX es minimizar el ancho de banda utilizado en la transmisión de voz y vídeo a través de la red IP y proveer un soporte nativo para ser transparente a los NATs (acrónimo de *Network Address Translation*, Traducción de Dirección de Red). La estructura básica de IAX se fundamenta en la multiplexación de la señalización y del flujo de datos sobre un simple puerto UDP, generalmente el 4569.

El protocolo original ha quedado obsoleto en favor de su segunda versión conocida como IAX2. Permite manejar una gran cantidad de *códecs* y transportar cualquier tipo de datos.

1.5 Ventajas de la VoIP respecto a la telefonía tradicional

Al utilizar VoIP sobre una LAN, es posible ahorrarse el cableado necesario para el teléfono. En los edificios empresariales, suelen ser necesarios dos tipos de cableados, uno para la red de datos a la que se conectan las computadoras, y otro para la red telefónica, a la que se conectan los teléfonos. Utilizando VoIP podrían unirse ambas redes en una sola: la red de datos. Sobre ésta, las computadoras pueden seguir enviando su información normalmente y además es posible aprovechar el ancho de banda no utilizado para enviar voz, de tal forma que las comunicaciones internas se realizarían mediante VoIP, usando el software adecuado en las computadoras, o mediante teléfonos IP implementados directamente en hardware que ya están disponibles, y que se conectarían a la red como cualquier otro dispositivo. VoIP puede facilitar tareas que serían más difíciles de realizar usando las redes telefónicas comunes de la PSTN, como integrarse fácilmente a audioconferencias y videoconferencias.

Ya no se seguirían reservando costosos canales para las comunicaciones como se hace en la conmutación de circuitos, por lo que las llamadas se abaratan considerablemente. Las llamadas de VoIP a VoIP entre cualquier proveedor son generalmente gratis. En contraste las llamadas de VoIP a PSTN representan costos para el usuario de VoIP. Los usuarios de VoIP siempre que estén conectados a Internet, pueden viajar a cualquier lugar en el mundo y seguir haciendo y recibiendo llamadas.

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

1.6 Problemas en la VoIP

A pesar de las grandes ventajas que proporciona la VoIP, todavía no es capaz de proveer la misma calidad en la transmisión de la voz que la telefonía tradicional. Como la comunicación es en tiempo real, someter la voz a procesamientos digitales atenta contra la calidad del sonido original.

Las redes VoIP heredan directamente todos los problemas de seguridad informática que normalmente afectan a la red de datos. Los paquetes de voz pueden ser interceptados y manipulados a voluntad de cualquier atacante. Los servidores pueden ser atacados provocando denegación de servicios. Estos son solo dos ejemplos.

Los mayores problemas de la VoIP radican en la calidad de servicio.

1.7 Calidad de Servicio en la VoIP

La comunicación mediante VoIP se puede considerar exitosa cuando la calidad de las señales que se transmiten satisface a quienes se comuniquen. En este punto juega un papel fundamental la calidad de servicio.

Dentro de los factores que inciden en la calidad de servicio se encuentra el ancho de banda. Cuanto mayor ancho de banda pueda ser utilizado para una conversación, mayor podrá ser la calidad del sonido y del vídeo transmitido.

Un factor que se vuelve más importante en VoIP es la latencia, es decir, el retardo existente desde que un extremo dice algo hasta que el otro lo recibe. En la telefonía tradicional, la latencia es despreciable, a no ser que la llamada sea de muy larga distancia, pues éste es el único factor del que depende, una vez que el canal de comunicación está reservado. En VoIP es más problemático. En primer lugar, el origen debe codificar la voz, empaquetarla en datagramas IP con porciones de la conversación comprimidas. Es necesario obtener una porción de la conversación antes de ser tratada, después debe ser comprimida, y por último enviada. Cada paquete UDP lleva, habitualmente, 30 milisegundos de conversación, por lo que hay que esperar ese tiempo antes de comenzar a procesarlo. Además el proceso de codificado y compresión requiere un tiempo, por lo que desde que el hablante dice algo hasta que comienza a ser enviado pueden transcurrir algo más de 30 milisegundos, debido a la carga computacional que imponga el códec de audio/vídeo que se utilice.

Se requiere además un tiempo hasta que el paquete llega al destino, que está en función de la velocidad de la red subyacente, de su ancho de banda disponible, de la distancia entre ambos extremos, y de la cantidad de elementos de red entre ellos. Por último, cuando el paquete llega al destino, debe ser descomprimido y decodificado, lo que puede suponer otros milisegundos extra. Todo

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

esto puede llevar entre 0.1 y 0.5 segundos. Aunque las opiniones varían, se admite que una latencia de 0.3 segundos es suficiente para que sea difícil mantener una conversación exitosa.

Otro problema que afecta indirectamente a la latencia y que tampoco aparecía en la telefonía tradicional son las variaciones en los tiempos de llegada (*jitters*). Durante una conversación, el origen enviará un paquete UDP cada 30 milisegundos. Cada uno de ellos puede ir por un camino diferente. Incluso aunque vayan por el mismo, el tiempo que tardan en recorrerlo puede variar. Eso se debe a que, al fin y al cabo, las conversaciones de VoIP se transmiten a través de una red a través de la cual también se envían otros datos. Este tipo de tráfico es impredecible, y en cualquier momento llegan ráfagas de datos que deben ser enviadas, consumiendo un ancho de banda que hace que otros paquetes se retrasen. El efecto producido es que los paquetes son enviados cada 30 milisegundos, pero no necesariamente llegan en este tiempo. El destino podría recibir un paquete en desorden.

Para que el usuario no perciba esos problemas, el destino tendrá que construir un *buffer* donde ir colocando los paquetes que va decodificando, y que se irán reproduciendo pasado un período de tiempo suficiente como para amortiguar el efecto del retardo. La necesidad de ese buffer soluciona el problema, pero aumenta aún más la latencia.

El eco puede darse en VoIP debido al acople entre el micrófono y el altavoz del otro extremo. Las palabras dichas por un extremo se reproducen en los altavoces del destino, y la voz es recogida por el micrófono y vuelta a enviar al origen, produciéndose el eco.

El último problema importante en la calidad del servicio es la pérdida de paquetes. Al estar montada sobre una red de datos no preparada inicialmente para la transmisión en tiempo real, los paquetes con porciones de voz pueden ser descartados y perdidos. Si esto ocurre durante una transmisión de datos habitual, la solución será solicitar su reenvío, lo que retrasará ligeramente la transmisión, pero no hará que la información recibida sea inválida. Sin embargo cuando ocurre esto en una transmisión en tiempo real, solicitar el reenvío no es una buena idea. Cuando pueda llegar el paquete perdido, habrá pasado el tiempo suficiente para que ya no tenga sentido reproducirlo. La pérdida de paquetes es algo admisible, pero solo relativamente. Es razonable soportar un cierto porcentaje de paquetes perdidos, siempre que estos no ocurran de forma consecutiva.

1.8 Centralitas PBX

Una central de telefonía privada (PBX) es un dispositivo que permite a las empresas conectar sus terminales telefónicos de forma independiente a la PSTN. Su objetivo fundamental es establecer y mantener la comunicación entre dos puntos finales durante todo el tiempo requerido por los usuarios, y

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

opcionalmente proveer información para contabilizar y facturar las llamadas. Indistintamente muchos fabricantes procuran implementar a sus PBXs la mayor cantidad posible de funcionalidades, de modo que sus productos sean más atractivos para los compradores o clientes, entre otras, señalamos los siguientes:

- Marcado rápido: Es posible marcar, por ejemplo, números de centros de emergencia, previamente definidos y configurados.
- Contestador automático: Mensaje de voz para cuando el usuario no esté disponible.
- Servicio de directorio automatizado: Los usuarios pueden ser enrutados a la extensión deseada tecleando o diciendo verbalmente las iniciales o el nombre del usuario deseado.
- Desvío de llamadas: Es posible que una extensión reciba las llamadas de otra, previamente definida.
- Transferencia de llamadas.
- Llamada en espera
- Aviso mediante timbre cuando una línea externa o extensión está libre: Es una funcionalidad que se puede acoplar también al servicio de llamada en espera.
- Conferencia entre 3 o más usuarios.
- Mensaje de Bienvenida.
- Marcado de una extensión desde el exterior del sistema: Una extensión puede recibir llamadas desde el exterior de la PBX, así como realizar llamadas al exterior.
- Sígame (Follow-me): Es posible programar el desvío de llamadas desde cierta extensión desde una distinta.
- Música en espera.
- Contestador automático de buzón de voz: Tras el mensaje del contestador automático, es posible guardar mensajes de voz.

1.8.1 Intranet PBX (IPBX)

El término IPBX (Intranet PBX) se refiere a aquellas centralitas telefónicas capaces de transmitir la voz sobre redes IP basándose en VoIP. La tendencia actual de los fabricantes de PBXs es incorporar a sus centralitas la posibilidad de transmitir la voz sobre redes de datos. Esta solución ofrece reducción de

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

costos por la gestión de una única infraestructura de red. La interacción simplifica y amplía las posibilidades de servicios de valor añadido.

Para la conexión a la Red de Área Local (LAN, acrónimo de *Local Area Network*) hace uso de tarjetas *ethernet*, y al igual que el resto de PBXs, también posee complejos mecanismos software que adaptan la señal de voz durante la comunicación a cada uno de los diferentes estándares.

1.8.2 SoftPBX

En los últimos años se han realizado esfuerzos, gracias al desarrollo de los sistemas operativos de computadoras, por construir *softwares* capaces de emular el funcionamiento de una IPBX.

Un SoftPBX es una IPBX diseñada completamente en software. Es capaz de proveer todas las funcionalidades de una centralita IP, con la posibilidad de extenderlas, ya que no hay dependencia directa con el hardware de telefonía disponible.

Existiendo tantos desarrollos de hardware en IPBX, ¿por qué elegir una solución basada en software?

Existen varias razones que aconsejan la adopción de un SoftPBX:

- Libertad y Gratuidad: En los SoftPBX de código abierto, esta concepción permite la adquisición, instalación y puesta en marcha de sistemas telefónicos sin dependencia a las grandes compañías de la telefonía.
- Adaptabilidad y Mejora continua: Se trata de software, de códigos informáticos, escritos en lenguajes de programación, por lo que la posibilidad de extender las funcionalidades es mucho mayor frente a las soluciones de hardware. Para la mayor adaptabilidad solo se requiere de personal con conocimientos de telefonía y programación, en lugar de una gran infraestructura de desarrollo de hardware.
- Disminución de costos contra plataformas privativas: En las variantes de los SoftPBXs de código abierto, en la mayoría de los casos se adquieren los programas en internet de manera gratuita, quedando las mayores inversiones para las plataformas de hardware necesarias, y los dispositivos adicionales.

A continuación se describen algunas de estas aplicaciones.

FreeSWITCH

FreeSWITCH es una aplicación de telefonía de código abierto, construida desde cero y diseñado para aprovechar las ventajas de la mayoría de los SoftPBX existentes. Dicha aplicación se puede utilizar

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

como un simple cambio de motor, una puerta de los medios de comunicación o un servidor de medios para acoger aplicaciones que utilizan scripts simples o XML para controlar el flujo de llamadas. Es una pasarela telefónica que une varias tecnologías y plataformas como SIP, H.323, IAX2, LDAP, Zeroconf, XMPP/Jingle, OpenPBX, Bayonne, YATE y Asterisk. [15]

Al mismo tiempo es un software multi-plataformas que se ejecuta en Windows y Linux y es posible ejecutarlo en otros como UNIX, Max OS X, BSD, Solaris 32 y las dos plataformas de 64 bits. Este software apoya tanto el ancho de banda estrecha como los códec ya que es una solución ideal para los dispositivos de puente.

A través de los años, FreeSWITCH se ha ido desarrollando y ha proporcionado toda una amplia gama de servicios especializados capaces de cumplir con funciones más específicas. Algunas de estas nuevas características son:

- Soporte_SRTP: Esta funcionalidad posibilita hacer llamadas con seguro RTP.
- modo_marcado_asterisk: Un módulo de marcado similar a la sintaxis de marcado Asterisk.
- modo_tono_stream: Reproducir los tonos de un archivo en cualquier lugar.

Bayonne

Bayonne ha entrado recientemente en el Proyecto GNU. Es un servidor de telefonía, pero éste es un término bastante general ya que incluye a casi todos los programas que se comunican por medio del habla sobre líneas de voz.

Este software permite funcionalidades como buzones de voz porque es completamente programable por scripts. También puede ser utilizado para crear sistemas de mensajes. Actualmente se encuentran en pleno desarrollo el soporte para bus TDM (acrónimo *Time-Division Multiplexing*, Multiplexación por División de Tiempo), el intercambio de llamadas y los servicios de puerta de salida VoIP.

Este servidor funciona en Linux, xBSD y Microsoft Windows, para la construcción de sistemas de respuestas por voz, y utiliza *plugins* de telefonía para la configuración del tiempo de ejecución del mismo. Ofrece características para permitir aplicaciones basadas en Perl.

Dicho servidor se puede utilizar para construir sistemas de administración basadas en telefonía, asistentes automatizados y sistemas de mensajería de voz. Además ofrece código abierto, escalable e independiente del medio de software en que se desarrolle.

Los usos de este servidor son múltiples, hay aplicaciones controladas por voz y sistemas de correo por voz. También sería posible crear herramientas de administración remotas basadas en el teléfono, sistemas de taquilla automáticos, votación automatizada o envío de anuncios y avisos por teléfono.

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

Bayonne se ha implementado tomando como base un sistema multihilo orientado a eventos que garantiza una muy buena escalabilidad. Esto lo hace utilizable en aplicaciones con alto rendimiento o muchas conexiones. La limitación principal radica en que no es capaz de intercomunicar con distintos protocolos de VoIP

OpenSER

Este proyecto se basa en el protocolo SIP y sus orígenes datan del proyecto SER (SIP Express Router) del Instituto de Investigación FhG FOKUS de Berlín, Alemania, del cual surgió la versión de código abierto OpenSER, en el año 2005.

La tarea de OpenSER es la de recibir y procesar mensajes SIP. Puede ser configurado como Proxy SIP para establecer sesiones de telefonía IP, video llamadas y mensajería instantánea; además, soporta el manejo de sistemas de presencia y puede funcionar como gateway hacia servidores de aplicaciones avanzadas tales como SMS (acrónimo de Short Message Service, Servicios de Mensajes Cortos) o hacia la PSTN .[16]

Dos componentes esenciales en la arquitectura de OpenSER que lo fortalecen son: el núcleo (core) y los módulos, en tanto permiten añadir funcionalidad al núcleo por medio de los módulos y no requieren modificar el núcleo para ello. De esta manera, es posible ajustar OpenSER a determinadas necesidades, con sólo agregar o modificar algunos de sus módulos.

OpenSER está programado totalmente en Lenguaje C para Unix y soporta direccionamiento IPv4 e IPv6, así como métodos de autenticación, autorización y *accounting* (registro detallado) también denominado AAA; además cuenta con interfaces de programación para PERL y Java, y entre otras características soporta ENUM (Electronic Numbering) , protocolo desarrollado por la IETF. Asimismo, dicho software se encuentra disponible para plataformas UNIX/Linux desde su sitio oficial en Internet; y es posible instalarlo sobre *sistemas Debian, Ubuntu, RedHat, CentOS, Fedora*, entre otros, por medio de su gestor de paquetes de forma automatizada.

Solo maneja el protocolo SIP.

Yate

Es el motor de nueva generación de la telefonía, que actualmente se centra en VoIP y PSTN, su poder reside en que es fácil de extender. Es capaz de unificar la voz, vídeo, datos y mensajería instantánea, ya que este motor de enrutamiento es muy flexible.

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

Yate, es un software libre de código abierto que usa Open H.323. El canal H.323 en Yate es considerado la mejor implementación libre basada en Open H.323. Además trabaja como Proxy de señalización SIP-H.323 para las compañías que tienen redes internas SIP y proveedores H.323.

El software está escrito en C++ y soporta scripting en diversos lenguajes de programación (PHP, Perl y Python) e incluso cualquier *Shell (interfaz de comandos)* de Unix. Las bibliotecas de Yate se han desarrollado con el fin de facilitar el desarrollo de las funciones externas ya que es un software listo para la producción. [11]

Yate se puede utilizar como:

- servidor VoIP;
- cliente VoIP ;
- pasarela VoIP para PSTN;
- gatekeeper para H.323 ;
- servidor para conferencias H.323;
- proxy H.323/SIP;
- servidor de registro SIP;
- servidor y cliente IAX;
- servidor para centro de Llamadas;
- motor de respuesta de voz interactiva.

Su desventaja principal es que carece de una comunidad de desarrollo fuerte que le de soporte.

Elastix

Elastix es una distribución basada en Asterisk (otra PBX basada en software, más adelante se detalla con detenimiento), promocionada por la empresa *Soluciones Palosanto*, éste es un software aplicativo que integra herramientas disponibles para PBXs basados en Asterisk en una interfaz simple y fácil de usar. Además añade su propio conjunto de utilidades y permite la creación de módulos de terceros para hacer de éste un paquete de software disponible para la telefonía de código abierto. Sus metas son, la confiabilidad, modularidad y fácil uso, éstas añadidas a la robustez para reportar, hacen de él, una gran opción para implementar un PBX basado en Asterisk. [21]

Las características provistas por Elastix son muchas y variadas, ya que integran varios paquetes de software, cada uno incluye su propio conjunto de características y añade nuevas interfaces para el

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

control y reportes de si mismo, lo que lo hace un paquete completo. Algunas de las características provistas por Elastix son:

- Soporte para video: Se pueden realizar video-llamadas.
- Soporte para Virtualización: Es posible correr múltiples máquinas virtuales de Elastix sobre la misma caja.
- Interfaz Web para el usuario, realmente amigable.
- "Fax a email" para faxes entrante: También se puede enviar algún documento digital a un número de fax a través de una impresora virtual.
- Interfaz para tarifas.
- Configuración gráfica de parámetros de red.
- Reportes de uso de recursos.
- Opciones para reiniciar/apagar remotamente.
- Reportes de llamadas entrantes/salientes y uso de canales.

SipXecs

SipXecs es un servidor modular basado en soluciones que se ejecuta en Linux y no requiere ningún hardware adicional, ya que interopera con cualquier pasarela compatible con SIP, teléfono o aplicación. SipXecs está diseñado como un sistema totalmente distribuido con una configuración centralizada. En este sentido todos los elementos que se pueden ejecutar en un único servidor hacen muy rentable y potente el sistema para las más pequeñas oficinas.

La configuración SipXecs Server proporciona fácil integración con portales web. Además, permite importar datos de configuración de un "*cut-sheet*", como una hoja de cálculo Excel. Todos los datos de configuración se almacenan en una base de datos y presta mecanismos automatizados para crear copia de seguridad y restauración.

SipXecs presenta características clave de un sistema IP PBX, éstas incluyen la gestión de voz, ajustes para la mensajería unificada, la configuración de un auto-asistente personal, configuración de idioma para el usuario del buzón de voz, marcación rápida y directorio de configuración, gestión de presencia, así como un historial de llamadas personales.

Su principal desventaja es que solo maneja el protocolo de señalización SIP.

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

Asterisk

Asterisk es una aplicación con licencia GPL (código abierto) capaz de realizar las funciones de una IPBX. Este proyecto fue desarrollado por el ingeniero Mark Spencer, miembro fundador de la compañía *Digium* y principal desarrollador.

Originalmente fue implementado para cualquiera de las diferentes distribuciones Linux existentes. Actualmente se intenta portar a otros sistemas operativos como Mac, Solaris o Microsoft Windows. Además hace voz sobre IP en cuatro protocolos (H.323, SIP, MGCP, IAX) y puede interoperar con casi todos los basados en estándares de telefonía utilizando el equipo de hardware relativamente barato. [20]

Asterisk pese a ser una aplicación software ofrece las mismas características y servicios que los caros sistemas propietarios PBX como puede ser el buzón de voz, salas de conferencia o música en espera entre otros. Son muchas las ventajas que ofrece respecto a las centralitas hardware algunos de ellos son los siguientes:

- **Funcionalidad:** Asterisk dispone de todas las funcionalidades de las grandes centralitas propietarias (*Cisco, Avaya, Alcatel, Siemens*, etc). Desde las más básicas (desvíos, capturas, transferencias, multi-conferencias), hasta las más avanzadas (Buzones de voz, IVR (*Interactive Voice Response*, Respuesta de Voz Interactiva), CTI (*Computer telephony integration, Integración Teléfono –Computadora*), ACD (*Automatic Call Distributor, Distribuidor Automático de Llamadas*).
- **Escalabilidad:** El sistema puede dar servicio desde 10 usuarios en una sede de una pequeña empresa, hasta 10.000 de una multinacional repartidos en múltiples sedes.
- **Competitividad en coste:** No solo por ser un sistema de código abierto sino gracias a su arquitectura hardware, utiliza plataforma servidor estándar (de propósito no específico) y tarjetas PCI para los interfaces de telefonía, que por la competencia del mercado se han ido abaratando progresivamente.
- **Interoperabilidad y Flexibilidad:** Asterisk ha incorporado la mayoría de estándares de telefonía del mercado, tanto los tradicionales (TDM) con el soporte de puertos de interfaz analógicos (FXS y FXO) y RDSI (básicos y primarios), como los de telefonía IP (SIP, H.323, MGCP, SCCP/Skinny). Eso le permite conectarse a las redes públicas de telefonía tradicional e integrarse fácilmente con centralitas tradicionales (no IP) y otras centralitas IP.

Adicionalmente provee todas las ventajas de tener un IPBX como:

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

- Configurar extensiones de su PBX principal en sitios distantes conectados a través de la red Internet y ahorrar costos en llamadas de larga distancia.
- Conexión con proveedores de Voz sobre IP para realizar llamadas internacionales a bajo costo o asociar un número telefónico de otro país a su sistema de PBX IP.
- Cualquier cosa que permita hacer otra planta telefónica de *Avaya/Cisco/Panasonic* es posible con el sistema de PBX de Asterisk y a una fracción de su precio.

1.9 Conclusiones del capítulo

La VoIP se presenta ante el mundo como una gran alternativa a las soluciones tradicionales de la telefonía. Posee una arquitectura robusta que se basa en nuevos conceptos que hacen converger las redes de telefonía y las redes de datos, unificando los servicios y aumentando las prestaciones.

La gran variedad de códecs y protocolos de VoIP que existen, han propiciado la aparición de tecnologías que cambian el modo tradicional de comunicarse las personas, así como nuevas y menos costosas soluciones para los principales servicios. Así ocurre con la aparición y evolución de los PBXs basados en software.

Aún conociendo las potencialidades de las soluciones de PBXs en software descritas, se concluye que la mejor alternativa para implantar la VoIP en la UCI es el SoftPBX Asterisk. A continuación se listan varias razones para consolidar esta propuesta:

- **Alta reducción de costo debido al empleo de software libre combinada con el bajo precio del hardware para telefonía:** Con la unión de ambas características se obtiene una PBX a un menor costo que cualquier solución tradicional, además de proveer un nivel de funcionalidad equivalente a los modernos sistemas disponibles en el mercado. El diseño de Asterisk permite implementar VoIP aún sin disponer del hardware necesario para interoperar con la PSTN.
- **Control:** Asterisk le permite al usuario tomar el control del sistema de telefonía. Una vez que la llamada está dentro del servidor GNU/Linux, cualquier manipulación puede ser llevada a cabo sobre la misma, o de manera remota.
- **Rápida extensión y desarrollo:** Su poderosa interfaz de comando, así como los ficheros y texto de configuración, le permiten una rápida configuración y diagnóstico en tiempo real.

Capítulo 1: Tecnología de Voz sobre el Protocolo IP

Es programable en varios niveles, y posee una gran comunidad de desarrollo que lo soporta.

- **Adaptación y flexibilidad:** Asterisk puede ser acoplada a cualquier sistema de telefonía existente, debido a la flexibilidad ofrecida por sus ficheros de configuración y código fuente. Esto lo diferencia del resto de los softPBXs.
- **Plan de discado extremadamente flexible:** Esta característica permite la integración de funcionalidades IVR y PBX. Muchas de las características existentes de Asterisk, así como otras a diseñar, pueden ser implementadas usando programación por lógica de extensiones.

CAPÍTULO 2: ASTERISK

Asterisk ha venido emergiendo como una gran revolución dentro de la telefonía. Se ha convertido en una alternativa muy atractiva para los servicios de telefonía, sobre todo por ser capaz de proveer todas las funciones de un PBX moderno de cualquiera de las compañías más prestigiosas del mundo, siendo solo un software corriendo sobre una computadora.

En este capítulo se abordará todo lo referente a la arquitectura, características y funcionalidades de Asterisk.

2.1 Introducción a Asterisk

Originalmente Asterisk se diseñó para trabajar en ambiente GNU/Linux. En la actualidad existen versiones que funcionan sobre otros sistemas operativos como Unix, MacOS X, Solaris, y Microsoft Windows, pero el mayor soporte lo brinda la comunidad de GNU/Linux.

Asterisk trabaja con Voz sobre IP. Entre sus potencialidades, hay que destacar que soporta los diferentes tipos de protocolos de señalización tales como SIP, MGCP, H.323, y su protocolo nativo IAX. Puede convertirse además en proveedor de servicios de predicción de discado, IVR (acrónimo de *Interactive Voice Response*, Respuesta Interactiva de Voz) de configuración personal, oficina PBX remota o centralizada, y servidor de conferencias.

La idea central del proyecto Asterisk fue que cualquier persona en cualquier lugar pudiera adquirir una computadora, instalarle alguna distribución del sistema operativo GNU/Linux, y adicionarle una tarjeta de expansión para conexiones con la PSTN. A toda esta unión añadirle el software Asterisk y al final tener una PBX con todas sus características y servicios con un desempeño aceptable, a un costo mucho menor que cualquier solución tradicional de PBX. [7]

2.2 Características y servicios soportados por Asterisk

Asterisk se presenta como una PBX con todas sus funcionalidades actuales, soporta todas las características de llamadas convencionales en una interfaz de estación, tales como identificador de llamadas, redireccionamiento de llamadas para líneas ocupadas, redireccionamiento de llamadas para llamadas no contestadas, servidor de conferencias, redireccionamiento de llamadas variables, transferencias supervisadas, transferencias no supervisadas, incrementos ADSI (acrónimo de *Active Directory Services Interfaces*, Interfaces de Servicios de Directorio Activo), correo de voz con directorio,

enrutamiento a costo mínimo, enrutamiento de VoIP, aplicaciones de prepago, servicio de llamadas en espera, reconocimiento de voz, discado predictivo, colas de llamadas, respuesta interactiva de voz, entre otras.[12]

Asterisk es programable en diferentes niveles tales como:

- **Lógica de extensiones:** Mediante esta y usando procesamientos lógicos de programación se pueden lograr, tomando como basamento ejemplos ya creados, aplicaciones simples y servicios.
- **Interfaz de Pasarelas Asterisk (AGI):** Es utilizada en la implementación de tareas más complejas. Se puede crear *scripts* con lenguajes tales como PHP y PERL.
- **Nivel de programación C:** Este es el nivel más bajo de programación, es utilizado para implementar aplicaciones, crear controladores (*drivers*), y demás. En general este nivel es necesario para la creación de aplicaciones que requieren de mayor envergadura en su programación.

Algunas funciones en Asterisk son implementadas como aplicaciones o son soportadas por combinación de aplicaciones que se usan en el plan de discado, generalmente soportadas por todos los canales.

A continuación se muestra una lista de algunas funciones PBX soportadas en Asterisk.

- Receptor de Alarmas;
- Autenticación de usuarios;
- Respuesta automatizada;
- Listas negras;
- Transferencias no supervisadas;
- Registros de llamadas detalladas;
- Desvío de llamadas si la extensión está ocupada o no responde;
- Monitoreo de llamadas;
- Estacionamiento de llamadas;
- Colas de llamadas;
- Grabación de llamadas;
- Recuperación de llamadas;
- Encaminamiento de llamadas (DID y ANI);
- Transferencia de llamadas;
- Llamadas en espera;

- Identificación de llamadas;
- Bloqueo de llamadas en origen;
- Identificación de llamadas durante las llamadas en espera;
- Tarjetas de llamadas;
- Conferencias de voz;
- Almacenamiento y recuperación en Base de Datos;
- Integración con Base de Datos;
- Marcación por nombre;
- Tonos distintivos de llamadas;
- Distributed Universal Number Discovery (DUNDi™);
- Función No Molestar;
- Llamadas de emergencia;
- Recepción y transmisión de Fax;
- Lógica de extensiones flexible;
- Presentación interactiva de directorio;
- Respuesta vocal interactiva (IVR);
- Agentes locales y remotos;
- Macros;
- Música en espera, sistema flexible basado en Mp3;
- Reproducción aleatoria o Lineal, control de volumen;
- Marcación predictiva;
- Conversión de protocolos;
- AGI (Asterisk Gateway Interface): Interfaz a *Gateway Asterisk*;
- Gestor gráfico de llamadas;
- Gestión de colas para llamadas salientes;
- Marcación predictiva;
- Interfaz de gestión TCP/IP;
- TDMoE Multiplexación por División de Tiempo sobre tramas *Ethernet*;
- Permite la conexión directa de varias PBX Asterisk;
- Latencia nula;
- Utiliza hardware *Ethernet* normal;
- Permite la integración de instalaciones físicamente separadas;
- Se pueden usar con las conexiones de datos más comunes (mínimo 128kb de subida de datos y máximo 300ms de latencia);

- Permite un plan de discado unificado en todas las oficinas.

2.3 Arquitectura de Asterisk

Asterisk fue diseñado cuidadosamente para obtener la máxima flexibilidad posible. Está conformado por cuatro APIs (acrónimo de Application Programming Interface, Interfaz de Programación de Aplicaciones) específicas definidas alrededor de un núcleo central de PBX. Este núcleo maneja todas las interconexiones internas de la misma abstrayéndose totalmente de protocolos específicos, códecs, e interfaces de hardware provenientes de las aplicaciones de telefonía. Todo esto permite a Asterisk usar el hardware y las tecnologías disponibles hoy para realizar sus funciones esenciales.

Internamente el núcleo maneja estas tareas:

- PBX Switching;
- Lanzador de aplicaciones;
- Traductor de Códecs;
- Programación de tareas y Administración de I/O.

Asterisk tiene un diseño modular. Las diferentes funciones del servidor son implementadas como módulos, los cuales son cargados e inicializados por el Cargador Dinámico de Módulos (CDM) en tiempo de ejecución. Cuando Asterisk es inicializado, el CDM carga e inicializa cada uno de los controladores (drivers), que proporcionan los Controladores de Canal, los Formatos de Ficheros, Códecs, Aplicaciones, etc. Luego los enlaza con la API interna apropiada para su tratamiento.

El punto fundamental de esta arquitectura es que funciona como un *gateway* de media entre protocolos de VoIP, y no solamente como un proxy de señalización. [14]

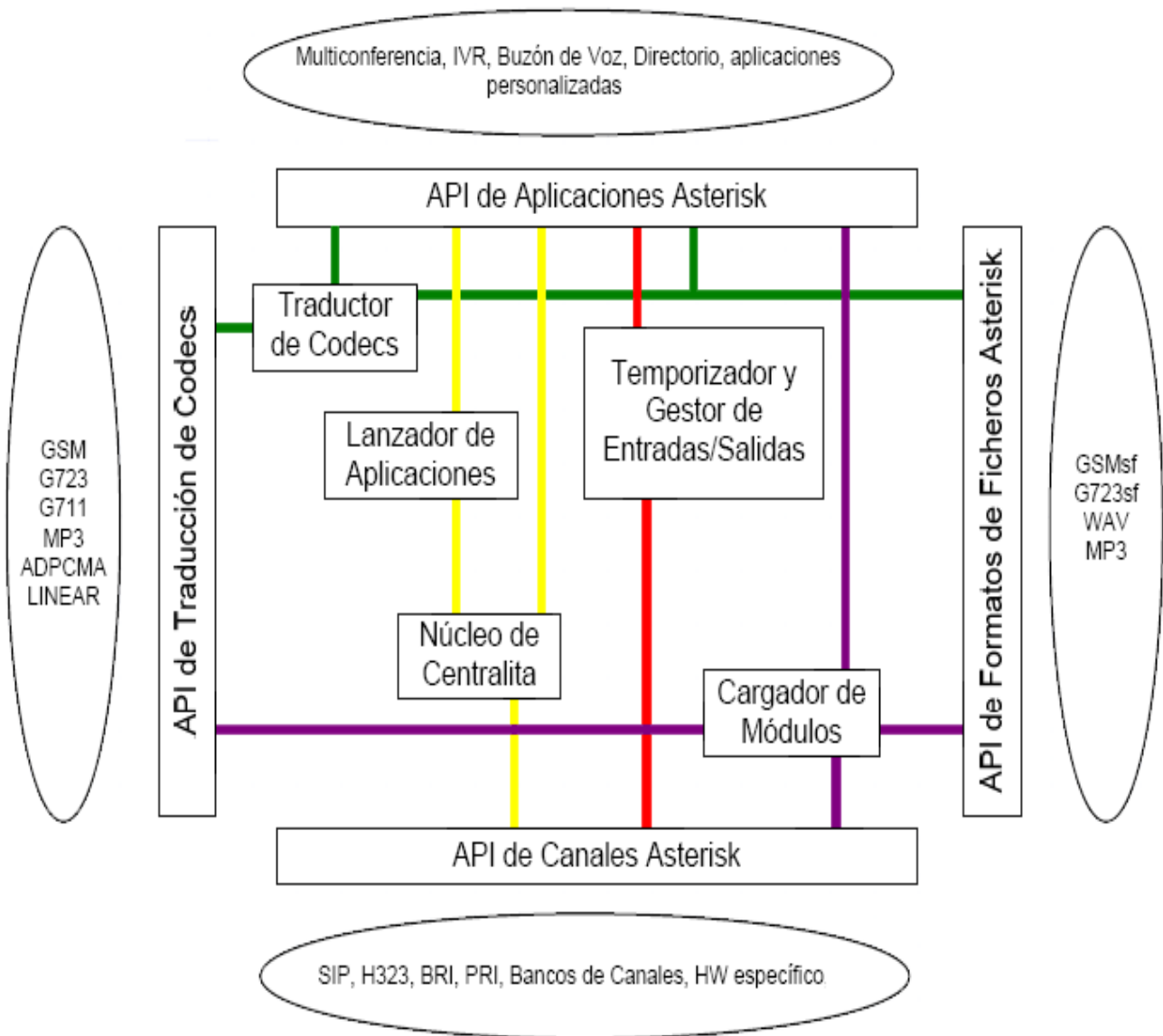


Figura 1 Arquitectura de Asterisk.

2.3.1 API de canal, (Channel API)

Un canal es el equivalente a una línea telefónica en la forma de un circuito de voz digital. Generalmente consiste de una señal analógica de un sistema de telefonía tradicional o alguna combinación de códec y protocolos de señalización. Es una conexión que conduce una llamada entrante o saliente en el sistema Asterisk. La conexión puede venir o salir hacia la telefonía tradicional analógica, digital o IP.

La API de canal le permite al núcleo de conmutación de Asterisk interconectarse con las fuentes TDM (acrónimo de Time División Multiplexión, Multiplexación por División de Tiempo) o las de paquetes de voz. Las fuentes TDM son generalmente dispositivos pseudo-TDM para los cuales el código desempeña funciones DSP (acrónimo de *Digital Signalling Processing*, Procesamiento Digital de Señales). Estas manejan el tipo de conexión a la cual el cliente está llegando, sea una conexión VoIP, ISDN (acrónimo de *Integrated Services Digital Network*, Red Digital de Servicios Integrados), o algún otro tipo de tecnología. Además son usadas para otras fuentes TDM tales como tarjetas de lógica de discado las cuales están soportadas como módulos cargables de Asterisk.

2.3.2 API para traducción de código

Esta API provee de una forma flexible para que el núcleo interactúe con voz codificada, no importando de cuales canales provenga ésta. Además soporta formatos tales como ADPCM (acrónimo de *Adaptive Differential Pulse Code Modulation*, Modulación por Impulsos Codificados Diferencial Adaptivo), G.711 (A-Law y μ -Law), G.723.1, G.726, G.729, GSM, iLBC, PCM Lineal, LPC-10, Speex.

2.3.3 API para formato de ficheros

Esta API le permite a Asterisk la capacidad de leer y reproducir sonidos en diferentes formatos, incluyendo WAV, AU, GSM y MP3. Maneja la lectura y escritura de varios formatos de archivos para el almacenamiento de datos en el sistema de archivos. Además proporciona mayor flexibilidad a las aplicaciones basadas en Asterisk, referente a los tonos de discado, DTMF (acrónimo de *Dual-Tone Multi-Frequency*, Multifrecuencia de Doble Tono), entre otros.

2.3.4 API de aplicaciones, (Application API)

Asterisk ejecuta secuencialmente los comandos asociados a cada extensión. Esos comandos son realmente aplicaciones que controlan el comportamiento de la llamada y del sistema en sí.

Esta API le permite a Asterisk conectar las llamadas de entrada con las llamadas de salida, a través de diversas aplicaciones como por ejemplo, conferencias, el correo de voz, tarjetas de llamada. Todas estas aplicaciones pueden tomar ventaja con esta API, ya que esta hace posible escribir nuevas aplicaciones de telefonía que interactúen directamente con el núcleo PBX.

2.4 El Plan de Discado de Asterisk.

El plan de discado le proporciona a Asterisk gran parte de su flexibilidad, ya que este es el encargado de determinar cómo las llamadas son enrutadas a través del sistema. Indica el itinerario que sigue una llamada desde que entra o sale del sistema hasta que llega a su punto final. De forma general define el comportamiento lógico de la PBX. Está compuesto por cuatro elementos fundamentales: las extensiones, las prioridades, los contextos, y las aplicaciones.

2.4.1 Extensiones

En la telefonía tradicional, las extensiones se asocian con teléfonos, interfaces o menús. En Asterisk, una extensión es una lista de comandos a ejecutar.

- Las extensiones se acceden cuando:
 - se recibe una llamada entrante por un canal dado;
 - el usuario que ha llamado marca la extensión;
 - se ejecuta un salto de extensiones desde el Dialplan de Asterisk.

Las extensiones pueden tener los siguientes formatos:

- Numérico: Se definiría una extensión solo con dígitos del 0 al 9: 8000, 50912.
- Alfanumérico: Se definiría una extensión con cualquier combinación de letras y números: persona1, juan2008, o simplemente juan.
- Numérico con identificador de llamada: Ejemplo 8000/50912
- Nombres predefinidos: normalmente se emplea la letra “s” para referirse la extensión especial de inicio.
- Patrones: Se pueden definir extensiones por medio de patrones, que no son más que caracteres que responden a determinado “patrón” o característica. Por ejemplo: _5XXX. Con este ejemplo se esperan extensiones de 4 dígitos, que comiencen con 5, seguidos de cualesquiera 3 dígitos entre 0 y 9.

Existen otros patrones, por ejemplo, la Z indica cualquier número entre 1 y 9, el carácter “N” significa cualquier dígito entre 2 y 9; el carácter “.” Indica también cualquier dígito, el carácter “_” significa el comienzo de la extensión. Indicar por ejemplo [1456789] significa emplear el 1, el 4, o cualquiera entre 4 y 9, pero nunca el 2 y el 3.

Con este ejemplo de extensión se puede lograr un primer acercamiento a la lógica:

```
exten=>50912, 1, Dial (SIP/50912, 20)
exten=>50912, 2,voicemail (u50912)
exten=>50912,101, voicemail(b50950)
```

La expresión “exten=>” indica cual es el próximo paso para la llamada. Funciona tanto el símbolo “=>” como el “=”. El conjunto 50912 indica el número marcado, el “1” indica la prioridad, es decir, se refiere a que es lo primero que se realizará al marcar el conjunto “50912”, y la función Dial hará que suene el teléfono registrado como 50912, que usa el protocolo SIP, por un tiempo de 20 segundos.

De no contestar, como prioridad 2, se dejará un mensaje de voz al buzón del usuario indicado, o de lo contrario, se envía otro mensaje de voz, pero esta vez a un buzón diferente.

La forma general de definir una extensión es la siguiente:

```
exten=> numero(o nombre usuario), prioridad, aplicación
```

2.4.2 Contextos

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

También son usados para crear menús de voz, que dan al usuario una lista de extensiones para escoger presionando las teclas de un teléfono multi-frecuencial. Esta funcionalidad es conocida como auto-atención de llamadas.

Se nombran definiendo su identificador dentro de corchetes ([]), por ejemplo, si se decidiera la creación de un contexto para la entrada de llamadas, se podría definir como sigue:

```
[entrada].
```

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

El uso creativo de los contextos puede ser usado para implementar diferentes características tales como:

- **Seguridad:** el plan de discado puede ser segmentado para permitirle a cierto número de extensiones la accesibilidad a distintas interfaces o usuarios VoIP.
- **Menús de voz de múltiples niveles:** debido a que las extensiones pueden ser de cualquier tamaño y también se pueden rehusar en cualquier contexto, se puede crear menú de voz.
- **Servicios de Autenticación:** los contextos pueden ser usados para verificar claves o códigos de acceso para la utilización servicios específicos.
- **Retorno de llamada:** para este tipo de servicio pueden ser usadas los contextos de extensiones, scripts externos, y la aplicación `app_qcall`.
- **Centralización del plan de discado:** usando el protocolo IAX se pueden conectar varios planes de discado centralizando en un solo servidor un plan de discado complejo.
- **Modos Día/Noche:** se puede variar el comportamiento de la PBX según el horario.
- **Enrutamiento:** enrutamiento de llamadas basado en extensiones.

2.4.3 Prioridades

Las prioridades son pasos enumerados de ejecución de cada extensión. Cada prioridad llama a una aplicación específica. Normalmente estos números de prioridad comienzan en 1 y aumentan de uno en uno en cada extensión. Estas prioridades son corridas en orden numérica.

2.4.4 Aplicaciones

Las aplicaciones son parte fundamental de Asterisk, ellas tratan al canal de voz, tocando sonidos, aceptando dígitos, o cortando una llamada. Las aplicaciones son llamadas con opciones que afectan a su forma de funcionamiento.

Asterisk ejecuta secuencialmente los comandos asociados a cada extensión. Esos comandos son realmente aplicaciones que controlan el comportamiento de la llamada y del sistema en sí.

Algunos ejemplos:

- **Hangup:** Colgar la llamada.
- **Monitor:** Comenzar la grabación a disco de una llamada.
- **Dial:** Realiza una llamada saliente.
- **Goto:** Salta a otra extensión o contexto.
- **PlayBack:** Reproduce un fichero de sonido.

2.4.5 Un pequeño plan de discado

Fichero: extensions.conf

```
[llamadasentrantes]
exten=>s, 1, Answer()
exten=>s,2,Background(saludos)
exten=>s,3,hangup()

exten=>6060, 1,playback(soporte)
exten=>6060,2,Dial(SIP/6060)
exten=>6060,3,hangup()

exten=>_50N[01234567]X,1,playback(msgFac2)
exten=>_50N[01234567]X, 2, hangup ()
```

Este pequeño plan de numeración, o de discado funciona como sigue:

Cuando alguien se comunica en la interfaz FXO, la comunicación es pasada para la extensión "s" dentro del contexto entrada. La extensión "s" atiende la comunicación y usando el comando Background hace sonar un saludo y aguarda por el discado otro número. Después de discar el 6060 por ejemplo, el sistema toca un mensaje (algo así como "usted acaba de comunicarse con el área..."), y luego llama a la extensión "6060" que se encuentra registrado como un usuario SIP.

En el caso de haber sido discado algún número de cinco dígitos, que comience con 5 seguido de 0, luego por un dígito entre 2 y 9, luego otro entre 0 y 7, y finalmente cualquiera entre 0 y 9; entonces se reproduce un mensaje de voz que podría decir "usted esta llamando a un usuario de la facultad 2", y finalmente se cuelga, o termina la comunicación.

2.5 Estructura de conexión lógica

Se puede definir en el plan de discado quién manejará el flujo RTP, y de qué forma. Asterisk permite dos formas de conexión lógica con sus usuarios, en función de quien establece todo el flujo de media durante la comunicación.

Estructura Árbol Jerárquico

La estructura árbol jerárquico (figura 2) se suele utilizar cuando hay *routers* entre los usuarios, ya que los puertos han de estar disponibles para que cada uno de los usuarios pueda enviar y recibir los *streamings* de audio. Cuando se utiliza este sistema, tanto el tráfico de protocolo de señalización, como de RTP pasan por Asterisk.

De esta manera, no importa qué códec tengan los terminales, ya que Asterisk se encargará de traducir un códec al otro para que se pueda hacer la comunicación (a esto se le conoce como *transcoding*). Tampoco importa dónde se encuentren los terminales, ya sea dentro de la propia red, o fuera de ésta. Asterisk se encargará de redirigir todo el tráfico necesario para que puedan hablar entre sí.

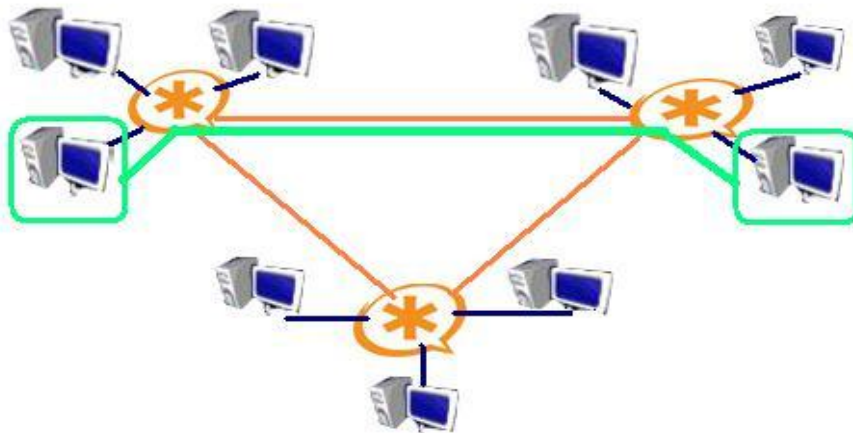


Figura 2 Estructura árbol jerárquico

Estructura Peer To Peer (P2P)

Con esta estructura (figura 3) el servidor Asterisk solo se encarga de realizar la señalización entre los extremos. Todo el tráfico de protocolo de señalización pasa por Asterisk, sin embargo, para el tráfico de RTP, los usuarios gestionan un canal propio, y toda la transmisión de media la controlan ellos. Cuando se utiliza este tipo de estructura, todos deben utilizar el mismo códec de audio, y no hay routers que impidan que el tráfico de voz camine libre en la red.

Este sistema tiene la ventaja de que al no pasar tráfico RTP por el Asterisk, ahorra ancho de banda. El hecho de que todos los usuarios utilizan el mismo códec para la comunicación, ahorra capacidad de procesamiento y memoria en los servidores Asterisk.

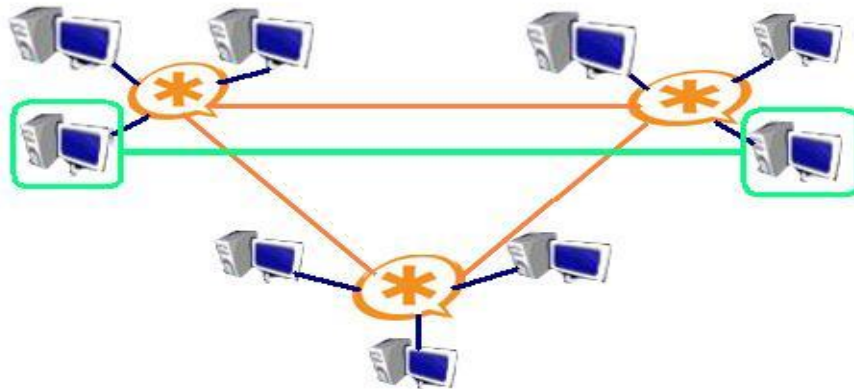


Figura 3 Estructura P2P

2.6 Consideraciones sobre rendimiento de Asterisk.

Asterisk emplean intensivamente la CPU del servidor para procesar los canales de voz, en lugar de tener un dispositivo DSP, dedicado a cada canal. El sistema es muy dependiente del rendimiento de la CPU. Es recomendable preservar al máximo la CPU del servidor donde se instale Asterisk, hasta el punto de solo tenerlo a él corriendo en el servidor.

En términos de requerimientos de recursos, las necesidades de Asterisk son similares a las de un sistema embebido, o una aplicación de tiempo real. Esto se debe en gran parte a su necesidad de contar con acceso prioritario al procesador y a los distintos buses del sistema. Por lo tanto, es muy importante que las funciones que no están directamente relacionadas con las tareas de procesamiento de llamadas de Asterisk, sean ejecutadas con prioridad baja. En los sistemas más pequeños, esto podría no ser un tema de discusión. Sin embargo, en sistemas grandes, los problemas de rendimiento se manifiestan como deficiencias en la calidad de audio para los usuarios, como el eco, el retardo en la voz, etc. Los síntomas se asemejan a los experimentados en un teléfono celular cuando se va fuera de rango. Cuando aumenta la carga de procesamiento, para el sistema será cada vez más difícil mantener las conexiones. Para un PBX, esta situación pudiera llegar a ser grave, por lo que durante la selección de la plataforma, las consideraciones de rendimiento son críticas.

Es interesante destacar que el tamaño de un sistema Asterisk en realidad no viene dado por el número de los usuarios o grupos que maneja, sino más bien por el número de llamadas simultáneas que puede soportar.[13]

Número de canales	Hardware Mínimo Recomendado
No más de 5	400 MHz, x86, 256 MB RAM
5 a 10	1 GHz, x86, 512 MB RAM
Hasta 25	3 GHz, x86, 1 GB RAM
25 o más	Se requiere multiprocesamiento, arquitectura distribuida en varios servidores

Tabla 2 Requerimientos mínimos de hardware para distintos escenarios

En sistemas Asterisk grandes, es muy común desplegar las funcionalidades a través de varios servidores. Uno o más unidades centrales se dedicarán al procesamiento de llamadas; éstos serán complementados por uno o más servidores auxiliares que se encargarán de la manipulación de periféricos (como un sistema de base de datos, un sistema de buzón de voz, un sistema de conferencias, un sistema de gestión, una interfaz web, un firewall, entre otros). Como es el caso en la mayoría de los entornos GNU/Linux, Asterisk se adapta cada vez más en función de las necesidades: utiliza un pequeño sistema capaz de manejar todo el procesamiento de llamadas, y las tareas periféricas pueden distribuirse entre varios servidores cuando el aumento de la demanda es superior a sus capacidades reales. La flexibilidad es la principal razón por la cual Asterisk es extremadamente rentable para una universidad en pleno desarrollo. Desplegar un sistema Asterisk con diseño distribuido es una tarea compleja.

2.7 Conclusiones del capítulo

- Asterisk es un poderoso software para la telefonía, pues es capaz de proveer todas las funciones que realizan las PBX comerciales más modernas.
- Su arquitectura lo hace muy robusto, y programable en distintos niveles.
- A partir de su arquitectura, se define un plan de discado muy potente, programable en línea de comandos, que le brinda mucha flexibilidad al sistema. Se pueden crear complejos y creativos planes de discado para dar mayores facilidades a los usuarios.
- En función de la estructura lógica de conexión, se puede aumentar el número de llamadas que puede gestionar un servidor Asterisk, tomando como consideración el ancho de banda en la red, y la carga computacional en el servidor.
- El rendimiento de los servidores Asterisk es un punto crítico, pues el procesamiento intensivo de los canales de voz se realizan en el microprocesador del servidor. Este factor será limitante para estimar la cantidad de llamadas que puede soportar el sistema.

CAPÍTULO 3: PROPUESTA DE IMPLANTACION DE ASTERISK EN LA UCI

Para realizar el dimensionamiento del sistema Asterisk, hay que tener en cuenta una serie de factores de suma importancia, como el hardware para los servidores, el lugar físico donde se instalarán los servidores, la topología física de la red de la universidad, la forma en que se interconectarán los servidores, los terminales a usar, entre otras consideraciones.

En este capítulo se abordarán los elementos que conforman la propuesta para la implantación de los servicios de VoIP en la UCI, mediante el SoftPBX Asterisk.

3.1 Requerimientos de hardware y software

3.1.1 Hardware para los servidores

Asterisk puede ser intensivo en el uso del procesador, pues él usa el propio procesador de la PC para hacer el procesamiento de los canales de voz. El rendimiento del sistema estará condicionado por la velocidad de procesamiento, la capacidad de almacenamiento masivo y de memoria de acceso aleatorio. Para dimensionar Asterisk en la UCI proponemos adquirir 6 servidores con las siguientes características mínimas:

Procesador	Quad-Core Intel Xeon Processor (2.5 GHz, 85 Watt, 1333 FSB)
Memoria Caché del Procesador	8MB Level 2 (2Mb por núcleo)
Memoria Acceso Aleatorio	8 GB PC2-5300 DIMMs (DDR2-667)
Controlador de red	Embedded NC373i Multifunction Gigabit Server Adapters
Controlador de Almacenamiento	HP Smart Array P400/256MB Controller
Capacidad de almacenamiento	Disco Duro HP 160GB 3GB/S 10000RPM 3.5"
Unidad Óptica	Al menos una unidad lectora DVD, DVD/CD-RW, DVD+R/RW, o CD/CD-RW

Tabla 3 Características de los servidores

Capítulo 3: Propuesta de Implantación de Asterisk en la UCI

Esta propuesta para el hardware está basada en las características de los servidores profesionales disponibles en el mercado, capaces de realizar cálculos intensivos, como los que se necesitarán tras la implantación de Asterisk en la universidad.

3.1.2 Sistema Operativo

Se propone utilizar *Debian GNU/Linux 4.0 ETCH* para instalar Asterisk, por ser la distribución propuesta por *Digium* para implantar el sistema. *Debian* es una distribución estable, tiene gran aceptación internacionalmente en grandes empresas, posee una gran comunidad de desarrolladores que brindan soporte, y está fuertemente ligada a la licencia GPL.

3.1.3 Versión de Asterisk

Se propone instalar la versión 1.14.20 de Asterisk, disponible en el sitio <http://www.asterisk.org/>. Es la última versión estable publicada hasta la fecha de confección de este documento. En esta versión vienen corregidos varios errores detectados en las versiones anteriores. Fundamentalmente se hicieron correcciones en el canal IAX2, con lo que mejoraron sensiblemente su implementación.

3.2 Ubicación física de los servidores

Se propone ubicar los servidores en distintas zonas del *campus* universitario, en atención a la agrupación de los usuarios, y a la existencia de locales con condiciones adecuadas para estos servidores. Estas zonas son:

- Rectorado;
- Área de la Infraestructura Productiva (IP);
- Área de docencia;
- Área de residencia.

Instalar los servidores en estas ubicaciones físicas tiene como ventajas:

- Es posible agrupar los usuarios, que en este caso se definen a nivel de inmueble, dígase laboratorios, aulas, oficinas, apartamentos. El servicio se habilitará para los puestos de trabajo que se encuentren instalados en estos inmuebles y no a usuarios terminales directamente. De modo que se asignará un número telefónico a cada computadora instalada.
- Ya que la instalación sigue un patrón distribuido, al ocurrir un desperfecto técnico en alguno de los servidores, no se interrumpirá todo el servicio, sino solo la parte afectada.

Capítulo 3: Propuesta de Implantación de Asterisk en la UCI

➤ Se aprovechan todas las condiciones físicas disponibles para la instalación de los servidores, donde radican los más importantes equipos de interconexión de redes de la universidad.

Algunas de estas condiciones son:

- Seguridad física.
- Climatización redundante.
- Respaldo eléctrico por cerca de 10 horas.
- Alto nivel en la jerarquía topológica de la red, por consiguiente, mayores niveles de ancho de banda.

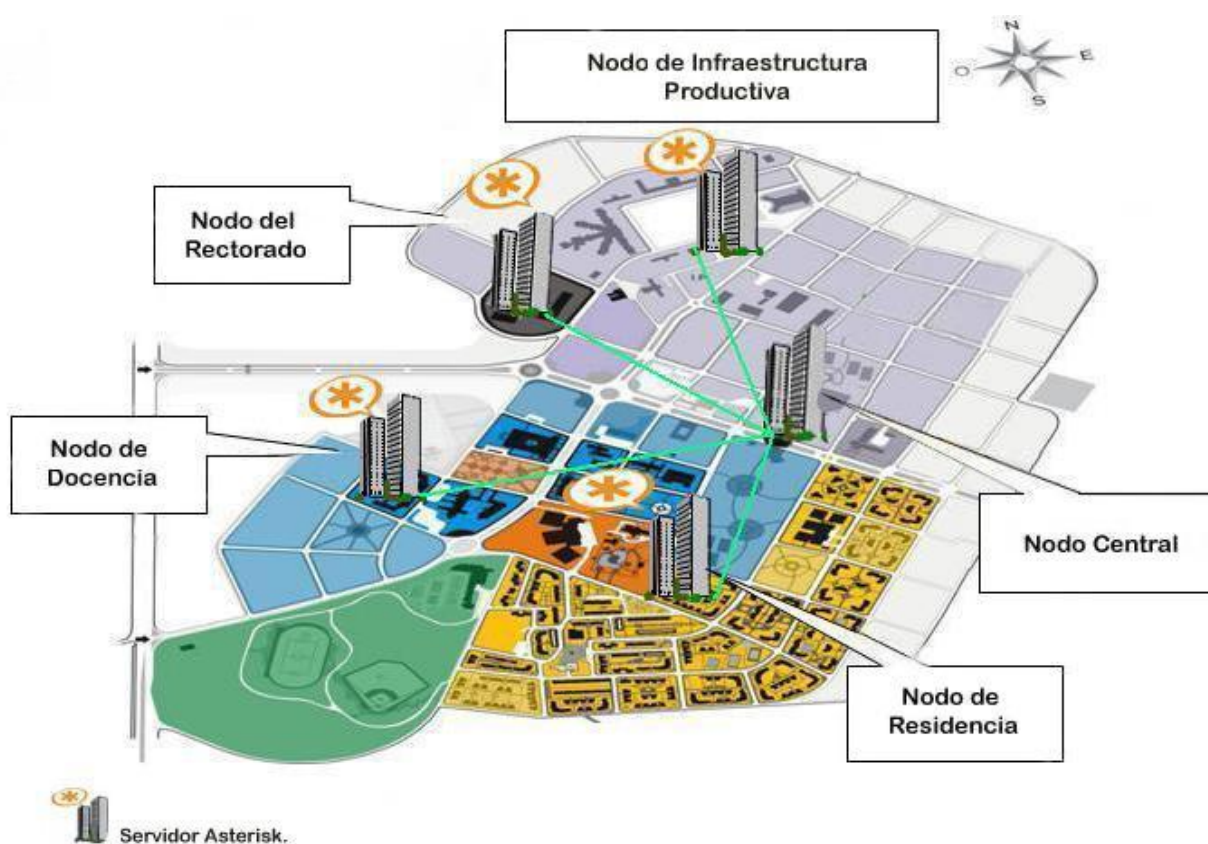


Figura 4: Distribución geográfica de los servidores en el *campus* universitario.

3.3 Ubicación de los servidores en la red

En la figura 5 se observa la ubicación de los servidores en correspondencia a la estructura real de la red de la universidad. Esta muestra una topología de árbol jerárquico, dispuestas en nodos de 3 niveles:

Capítulo 3: Propuesta de Implantación de Asterisk en la UCI

- Nivel 1 o Núcleo, incluyendo el nodo central, donde se encuentran los servidores de los principales servicios telemáticos de la universidad. El enlace en este nivel es de 10 Gb/s.
- Nivel 2 o de Agregación: Constituyen los nodos de las principales áreas de la universidad, donde se ramifica hacia el nivel 3. En enlace es de 1 Gb/s.
- Nivel 3 o de acceso: Es el nivel más bajo en el árbol, es donde se conectan los edificios directamente, y éstos a sus distintos inmuebles y puntos de acceso. En este nivel el enlace es de 100 Mb/s

Los servidores Asterisk quedarán ubicados en los nodos de nivel 1 o núcleo, de la red de la universidad, excepto el área del rectorado, cuyo nodo pertenece al nivel 2 o de agregación. En todos los casos se propone un enlace a los *switch* capa 3, de 1 Gb/s, para garantizar el ancho de banda necesario para las llamadas que se establezcan.

Teniendo en cuenta que en el área de residencia, y en los docentes, se encuentra la mayor cantidad de puntos de acceso a la red, y por consiguiente también la mayor cantidad de computadoras conectadas a la red, se propone el montaje de un clúster Asterisk en cada una de estas dos áreas, conformado por dos servidores con las características planteadas en el epígrafe 3.1, para manejar con mayor eficiencia todo el flujo de llamadas que se pueda generar. Se calculan alrededor de 2500 computadoras conectadas en la residencia, y cerca de 3000 en los docentes.

Se estima que la mayor densidad de llamadas exista en el área de los docentes, pues es el lugar de mayor concentración de computadoras en uso, y donde la comunidad de estudiantes y profesores requieren en el proceso docente-productivo de comunicación.

Cada puesto de trabajo contará con un teléfono basado en software, con el que se registrará en su servidor correspondiente en la red. Podrá realizar llamadas hacia cualquier abonado disponible en su subred, o hacia otro abonado que se encuentre registrado en otro servidor.

Los servidores para esta propuesta solo realizarán el proceso de señalización entre los abonados, de manera que la comunicación se realizará del modo *peer to peer*. Los abonados, luego de realizada la señalización, establecerán un canal de comunicación para el flujo directo de media en tiempo real a través de la red. De esta manera no se realizará procesamiento digital intensivo de señales en el microprocesador de cada servidor, por lo que el éxito de la comunicación entre los abonados recaerá sobre la calidad del canal de media que se establezca entre ellos, y no del rendimiento de los servidores.

Capítulo 3: Propuesta de Implantación de Asterisk en la UCI

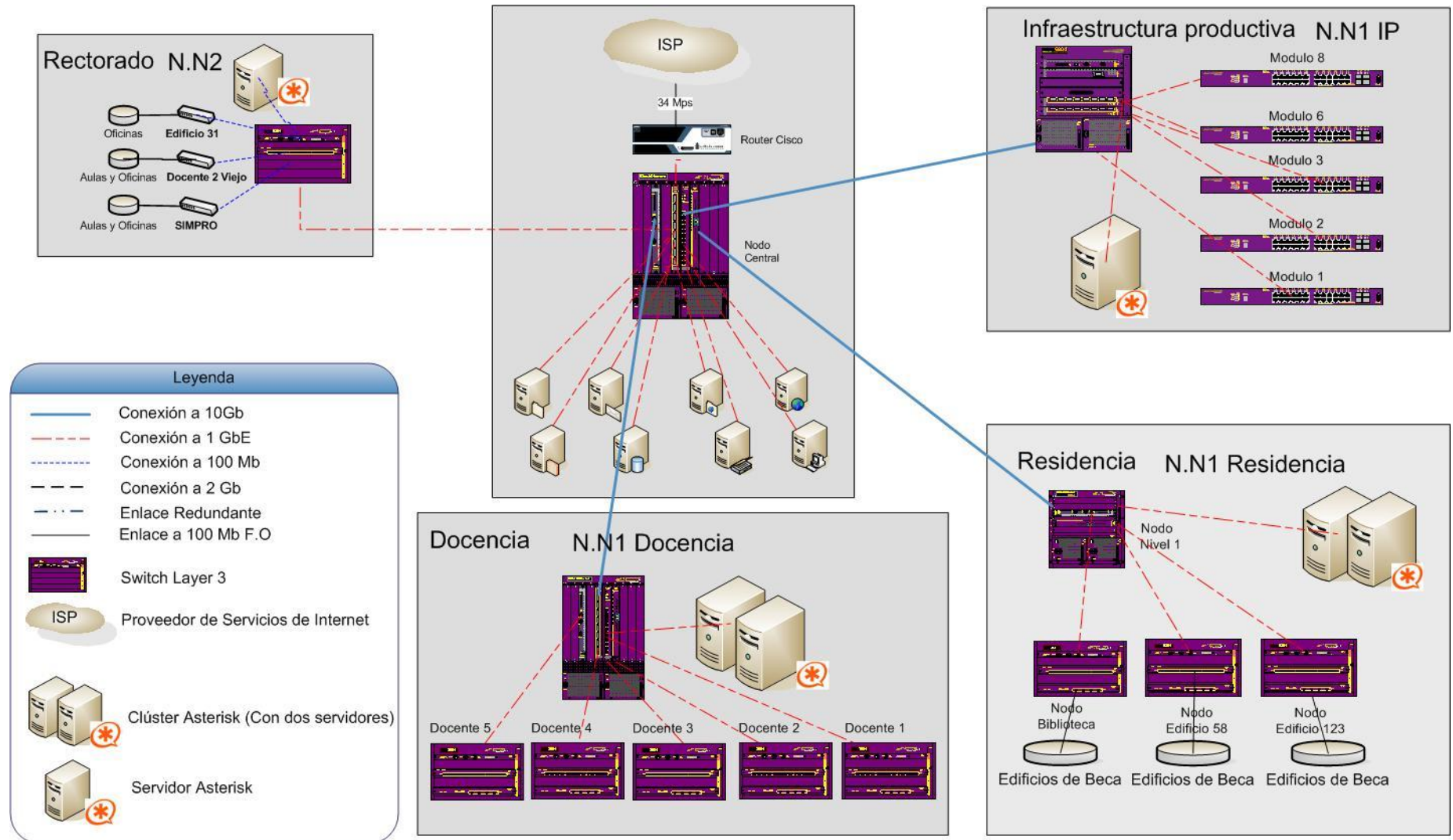


Figura 5 Ubicación de los servidores en la red de datos

3.3 Interconexión de los servidores

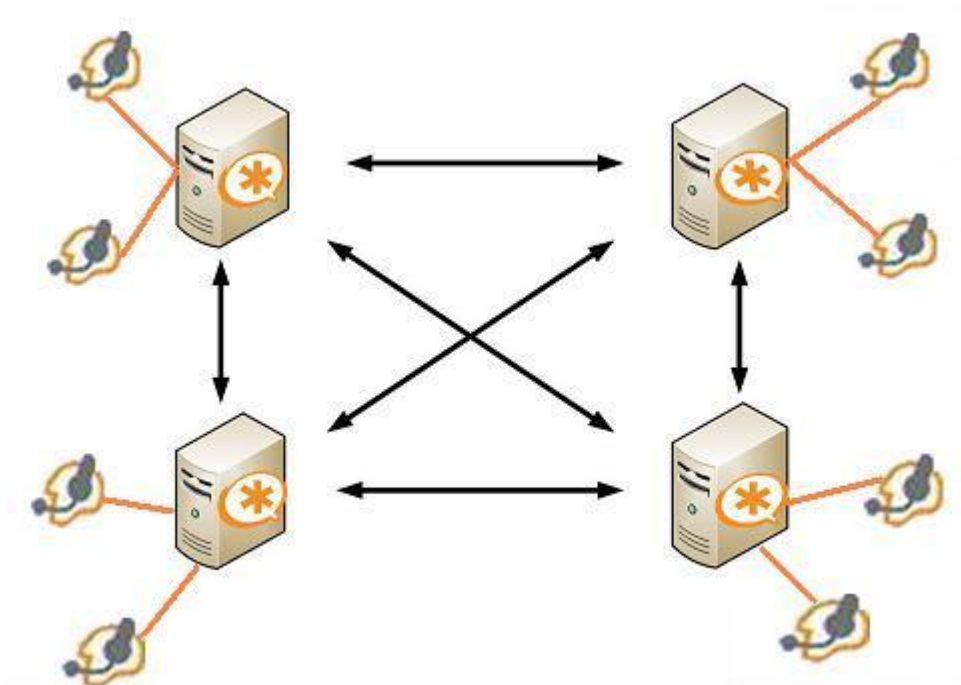


Figura 6 Interconexión de los servidores

Una cuestión importante es la manera en que se realizará la comunicación entre los servidores, a los cuales se conectarán los abonados. La primera observación es que no existe una jerarquía entre estos elementos de red. Todos están al mismo nivel, interconectados de manera que un abonado del servidor de la residencia puede llamar a cualquier abonado del servidor de los docentes, por solo citar un ejemplo.

A continuación se describen los pasos básicos para hacer posible la intercomunicación entre los servidores, por medio del protocolo IAX2:

- Es necesario que todos los servidores tengan IP definidas como estáticas.
- Configurar cada uno de los servidores como *user* de los demás en el fichero *iax.conf*, situado en */etc/asterisk*;
- Configurar cada uno de los servidores como *peer* de los demás en el fichero *iax.conf*, situado en */etc/asterisk*;

Capítulo 3: Propuesta de Implantación de Asterisk en la UCI

- Configurar el plan de discado de cada servidor declarado como *user*, para especificar cuáles llamadas puede realizar a los servidores declarados como *peers*. Esto se hace en el fichero *extensions.conf*, situado en el fichero */etc/asterisk*.

Se muestra como ejemplo cómo deberían quedar los ficheros *iax.conf* y *extensions.conf* en uno de los servidores (ver en los anexos cómo deben quedar configurados los 4 servidores):

En el servidor Asterisk de la residencia:

(fichero *iax.conf*)

```
[general]
register => asteriskrec:<password>@asteriskrec.uci.cu
register => asteriskip:<password>@asteriskip.uci.cu
register => asteriskdoc:<password>@asteriskdoc.uci.cu
```

```
[asteriskrec]
type = friend
user = asteriskrec
host = asteriskrec.uci.cu
```

```
[asteriskip]
type = friend
user = asteriskip
host = asteriskip.uci.cu
```

```
[asteriskdoc]
type = friend
user = asteriskdoc
host = asteriskdoc.uci.cu
```

(fichero *extensions.conf*)

```
exten => _a1XXX,1,Dial(IAX2/asteriskrec/${EXTEN:1},30,r)
exten => _a1XXX,2,Congestion
```

```
exten => _a2XXX,1,Dial(IAX2/asteriskip/${EXTEN:1},30,r)
exten => _a2XXX,2,Congestion
```

```
exten => _a3XXX,1,Dial(IAX2/asteriskdoc/${EXTEN:1},30,r)
exten => _a3XXX,2,Congestion
```

```
exten => _a4XXX,1,Ringin
exten => _a4XXX,2,Dial(SIP/${EXTEN:1},30)
exten => _a4XXX,3,Festival, Extensión no disponible
exten => _a4XXX,4,Hangup()
```

3.4 Plan de numeración

Para que cada abonado pueda registrarse en los servidores correspondientes, estos deben ser nombrados de la siguiente manera:

- asteriskrec.uci.cu: para el servidor ubicado en el nodo del rectorado.
- asteriskip.uci.cu: para el servidor ubicado en el nodo de la Infraestructura Productiva.
- asteriskdoc.uci.cu: para el servidor ubicado en el nodo de docencia.
- asteriskres.uci.cu: para el servidor ubicado en el nodo de residencia.

El plan de numeración se propone como sigue:

- Los abonados del rectorado usarán los números _a1XXXX
- Los abonados de la Infraestructura Productiva usarán los números _a2XXXX
- Los abonados de los docentes usarán los números _a3XXXX
- Los abonados de la residencia usarán los números _a4XXXX

El patrón X significa cualquier dígito entre 0 y 9, incluyéndose estos dígitos también.

3.5 Protocolos de señalización

Para configurar los abonados de cada uno de los servidores se propone utilizar el protocolo de señalización SIP por varias razones:

- Es un protocolo simple, basado en texto. Utiliza primitivas sencillas para el establecimiento de sesiones. No se definen servicios o funciones como en el caso de h.323.
- No es necesario establecer un control centralizado de la comunicación por medio de SIP, pues permite conexiones *Peer to Peer*.
- Asterisk se comporta como un servidor SIP Proxy, por lo que maneja perfectamente las conexiones SIP.
- SIP es soportado por numerosos tipos de terminales tales como: computadoras actuando como un *Softphone*, teléfonos IP, o teléfonos convencionales conectados a un adaptador.
- Una llamada SIP no requiere la existencia inmediata de una conexión en la capa de transporte, aunque la señalización del fin de la llamada debe hacerse explícitamente.
- SIP necesita menos ciclos de CPU para generar mensajes de señalización que los demás protocolos, como H.323. Por lo tanto, teóricamente el servidor puede manejar más transacciones.

Capítulo 3: Propuesta de Implantación de Asterisk en la UCI

Para interconectar los servidores Asterisk se propone utilizar IAX2, por ser el protocolo de señalización nativo para esta tarea, además de las siguientes razones:

- Es un protocolo abierto, no sujeto a restricciones propietarias.
- Es utilizado para manejar conexiones VoIP entre servidores Asterisk, y entre servidores y clientes que también utilizan protocolo IAX.
- Permite manejar una gran cantidad de códecs y un gran número de *streams*, lo que significa que puede ser utilizado para transportar virtualmente cualquier tipo de dato.
- IAX2 utiliza un único puerto UDP, generalmente el 4569, para comunicaciones entre puntos finales (terminales VoIP) para señalización y datos. El tráfico de voz es transmitido *in-band*, lo que hace a IAX2 un protocolo casi transparente a los cortafuegos y realmente eficaz para trabajar dentro de redes internas.
- El protocolo crea sesiones internas y dichas sesiones pueden utilizar cualquier códec que pueda transmitir voz o vídeo. El IAX esencialmente provee control y transmisión de flujos de datos multimedia sobre redes IP. IAX es extremadamente flexible y puede ser utilizado con cualquier tipo de dato incluido vídeo.
- El objetivo principal de IAX es minimizar el ancho de banda utilizado en la transmisión de voz a través de la red IP, con particular atención al control y a las llamadas de voz.

3.6 Códecs

A partir de que se sugiere que los servidores solo se encarguen de realizar la señalización, los abonados deben usar un mismo códec de audio para poder establecer la comunicación de tiempo real por medio de RTP. La elección del códec a utilizar está condicionada no a la carga computacional, puesto que el procesamiento digital se realizará en los ordenadores donde se alojan los abonados, y no será significativo; pero sí al ancho de banda que se utilizará en la transmisión de la voz digitalizada. Por lo antes expuesto se propone el códec *gsm*, ya que es libre, consta de una alta calidad de audio debido a su rigurosa compresión de datos, recupera los paquetes perdidos, y sobre todo requiere de bajo consumo de procesador.

Capítulo 3: Propuesta de Implantación de Asterisk en la UCI

3.7 Configuración de los terminales

Los terminales IP que se proponen, para mantener los costos de implantación en valores aceptables, son los *softphones*, pues solo se requiere adicionalmente disponer de audífonos y micrófonos, y el software correspondiente.

Se puede utilizar cualquier *softphone* que entienda el protocolo SIP, de los disponibles en la red. En especial se sugiere por su versatilidad el *Xlite*, de fácil configuración y puesta en marcha, para los usuarios de Microsoft Windows, y para los usuarios de *GNU/Linux*, el *softphone Ekiga*, disponible en la mayoría de las distribuciones.

3.8 Pruebas de señalización

Para tener una estimación de cómo se comporta un servidor Asterisk, se dispuso la instalación del sistema sobre una plataforma de hardware con las siguientes características:

Microprocesador: Intel Pentium 4, 2.4 GHz, 512 Kb de caché L2

Disco Duro: Seagate, 80 GB, tipo IDE, 7200 RPM

Se instalaron las versiones propuestas del sistema operativo Debian GNU/Linux, y de Asterisk. Se instaló la versión 3.1 del SIPP, un software que genera llamadas virtuales, por medio del protocolo SIP, en otra computadora de iguales características, aunque este detalle no es importante para correr este programa.

Se procedió a seguir las indicaciones de los casos de pruebas provistos por los fabricantes del software SIPP:

En el fichero *sip.conf* del servidor asterisk, situado en */etc/asterisk*, añadir las siguientes líneas:

```
[sipp]
type=friend
context=sipp
host=dynamic
port=6000
user=sipp
canreinvite=no
disallow=all
allow=ulaw
```

En el fichero *extensions.conf* del servidor asterisk, situado en */etc/asterisk*, añadir las siguientes líneas:

```
[sipp]
exten => 2005,1,Answer
exten => 2005,2,SetMusicOnHold(default)
exten => 2005,3,WaitMusicOnHold(20)
exten => 2005,4,Hangup
```

Capítulo 3: Propuesta de Implantación de Asterisk en la UCI

Luego reiniciar el servidor asterisk, y desde la computadora que está corriendo el software de pruebas, ejecutar la línea:

```
sipp -sn uac -d 20000 -s 2005 10.31.18.23 -l 1000
```

El SIPP intenta llamar a la extensión 2005 por tiempo de 20 segundos, mediante el usuario creado recientemente. Se indican como máximo que se realicen 1000 llamadas. El resultado de esta prueba se puede observar en la siguiente imagen:

```
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
10.0(10000 ms)/1.000s 5060 279.04 s 2790 10.31.18.23:5060(UDP)

10 new calls during 1.000 s period 0 ms scheduler resolution
315 calls <limit 500> Peak was 416 calls, after 249 s
0 Running, 713 Paused, 60 Woken up
0 dead call msg <discarded> 1265 out-of-call msg <discarded>
3 open sockets

Messages Retrans Timeout Unexpected-Msg
INVITE -----> 2790 2780 298
 100 <----- 2177 0 0 0
 180 <----- 0 0 0 0
 183 <----- 0 0 0 0
 200 <----- E-RTD1 2177 0 0 0
 ACK -----> 2177 0
Pause [ 10.0s ] 2177
 BYE -----> 2177 900 100 0
 200 <----- 2077 0 0 0

----- [+-!*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
Last Error: Aborting call on UDP retransmission timeout for Call-ID '247...
```

Se interpreta de estos resultados que, teniendo en cuenta que las acciones que llevan a cabo mediante la realización de las llamadas es generar un sonido de música en espera, es posible mantener 416 llamadas simultáneas, es decir, 832 abonados pueden estar comunicándose al mismo tiempo, con un servidor con estas características. En un tiempo de 279 segundos se generaron 2790 llamadas, a una razón de 10 llamadas por segundo.

Generalizar este resultado es algo complejo, pues no fueron llamadas reales las que se realizaron. El software utilizado no es capaz de proveer información acerca de la carga de CPU, del rendimiento de este, la calidad de sonido cuando se arribó al valor límite de llamadas simultáneas, ni tampoco el ancho de banda utilizado. Este método de prueba no es confiable para estresar un servidor, y así conocer el rendimiento real a partir de sus características específicas.

3.9 Conclusiones del capítulo

- La propuesta de implantación de Asterisk se realizó en función de la topología física de la red de datos existente en la universidad.
- Se eligió el patrón distribuido para la instalación de los servidores, aprovechando todas sus ventajas sobre el patrón centralizado.
- Se diseñó un plan de numeración de acuerdo a la distribución propuesta de los servidores.
- Se realizaron pruebas de señalización a pequeña escala por no contar con el *hardware* propuesto.
- No existe un método certificado para medir el rendimiento de un servidor Asterisk en función de sus características de *hardware*.
- Con esta propuesta es posible ampliar las posibilidades de comunicación telefónica en la comunidad universitaria.

CONCLUSIONES

Al término de esta investigación se realizó una propuesta viable para implantar la VoIP en la universidad, dando cumplimiento al objetivo central de ampliar las posibilidades de comunicación telefónica en la UCI.

Como resultados destacables de este trabajo se deben mencionar:

- La VoIP es la mejor alternativa para ampliar las comunicaciones en la universidad.
- Asterisk es el PBX basado en software con mejores características y funcionamiento para implantar la VoIP en la universidad.
- Se debe seguir un patrón distribuido para instalar los servidores Asterisk.
- Los servidores deben encargarse solo de realizar la señalización entre los abonados que se comuniquen.
- El códec de audio que mejor relación ancho de banda – calidad de voz – carga computacional presenta es el gsm.
- Los protocolos de señalización SIP y IAX son los más idóneos para configurar Asterisk en la universidad. IAX para la comunicación entre los servidores, y SIP para la comunicación entre los usuarios y los servidores.

RECOMENDACIONES

Al concluir esta investigación se recomienda:

- Elaborar una aplicación que sea capaz de simular llamadas a un servidor Asterisk, y que brinde información sobre la capacidad de conexiones que soporta, el rendimiento en situaciones de estrés, ancho de banda utilizado para las llamadas, y brindar estadísticas para cada tipo de señalización y códec de audio utilizado;
- Realizar un estudio sobre la interconexión de los servidores Asterisk propuestos con la PBX instalada en la universidad;
- Realizar un estudio sobre la integración del servidor LDAP de la universidad, con el plan de numeración de Asterisk, para poder brindar los servicios de telefonía hasta el nivel de usuario.
- Implementar la propuesta descrita en esta investigación.

BIBLIOGRAFÍA

1. [Online] 2008. [Cited: febrero 10, 2008.] <http://www.asterisk.org>.
2. [Online] 2008. [Cited: enero 26, 2008.] <http://asterisk.org/index.php?menu=hardware>.
3. [Online] IAFX, Solutions (2008). [Cited: marzo 20, 2008.] http://shop.ifax.com/index.php?cPath=32_26_29.
4. [Online] LinuxMall (2005). [Cited: febrero 15, 2008.] <http://www.linuxmall.cl/asterisk.html>.
5. **Manesh, Nasser K.** [Online] marzo 3, 2008. [Cited: febrero 1, 2008.] <http://www.millenigence.com/articles/asterisk-non-technical-review.pdf>.
6. **Powel, Andy.** [Online] febrero 13, 2008. [Cited: marzo 16, 2008.] <http://www.automated.it/guidetoasterisk.htm>.
7. **Spencer, Mark.** [Online] febrero 20, 2007. [Cited: marzo 2, 2008.] http://www.digium.com/downloads/asterisk_whitepaper.pdf.
8. —. [Online] enero 13, 2007. [Cited: febrero 12, 2008.] <http://www.digium.com/handbook-draft.pdf>.
9. [Online] VoiceTronix (2006). [Cited: abril 14, 2008.] <http://www.voicetronix.com/open-source.htm>.
10. [Online] Voip-info. (2005). [Cited: abril 23, 2008.] <http://www.voip-info.org/wiki-Asterisk+introduction+administrators+-+THE+PLACE+TO+START!> .
11. Yate. *Yate*. [Online] febrero 20, 2008. [Cited: marzo 1, 2008.] <http://yate.null.ro/pmwiki/>.
12. **Gonçalves, Flavio E.** *Asterisk PBX, Guía de la Configuración*. Brasil : s.n., 2007.
13. **Jim Van Meggelen, Leif Madsen, Jared Smith.** *The Future of Telephony*. Estados Unidos de América : O'Reilly Media, 2007.
14. **Kris Sheets, Jimmy Terence Estrada, Marcelo Garcia, Dany Saavedra.** *Asterisk en Español, versión 1*. 2005.
15. **Minessale, Anthony.** *freeswitch. freeswitch*. [Online] 2008. [Cited: marzo 1, 2008.] <http://www.freeswitch.org/>.

16. **Ramon, Juan Ignacio.** OpenSer. *OpenSer*. [Online] 1, febrero 2, 2008. [Cited: abril 3, 2008.] <http://www.openser.com/>.
17. **Russomanno, Lic. Donatella.** [Online] abril 8, 2008. [Cited: abril 20, 2008.]
18. **Sagredo, Alberto.** Volp para novatos. *Volp para novatos*. [Online] febrero 15, 2008. [Cited: abril 3, 2008.] <http://www.voipnovatos.es/>.
19. **Víbora, Luisa Bastero de la.** aslan.es. *aslan.es*. [Online] enero 19, 2008. [Cited: febrero 6, 2008.] www.aslan.es.
20. **Villares, Ing. Fernando M.** Asterisk. *Asterisk*. [Online] enero 16, 2008. [Cited: abril 10, 2008.] www.asterisk.es.
21. . Elastix 0.9-alpha [Online] 2008. www.elastix.org.

ANEXOS

A. Instalación de Asterisk

Asterisk tiene dependencias de software con algunos paquetes, los cuales se necesitan instalar. La instrucción de Debian “*apt-get install*” descarga e instala automáticamente los paquetes solicitados y las dependencias necesarias. Los paquetes necesarios son:

- *libstdc++-dev, gcc, g++, make:*

```
apt-get install libstdc++-dev
```

```
apt-get install gcc
```

```
apt-get install g++
```

```
apt-get install make
```

- *Bison y flex:* Necesarios para compilar Asterisk. Bison es un generador de analizadores sintácticos de propósito general que convierte una descripción gramatical para una gramática independiente del contexto y Flex es una utilidad para generar programas que reconocen patrones de texto.

```
apt-get install bison
```

```
apt-get install flex
```

- *Zlib zlib-devel:* El paquete zlib contiene rutinas de compresión y descompresión.

```
apt-get install zlib
```

- *ncurses y ncurses-devel:* El paquete Ncurses proporciona librerías para el manejo de caracteres y terminales, incluidos paneles y menús. Necesario para construir algunas nuevas herramientas de Asterisk.

```
apt-get install ncurses ncurses-devel
```

- *Kernel-sourcecode y kernel-devel:* Para poder compilar Asterisk es necesario tener instaladas las Fuentes del kernel y el paquete de desarrollo del kernel.

```
apt-get install kernel-sourcecode kernel-devel
```

- *Openssl y openssl-devel:* El paquete OpenSSL contiene herramientas de administración y librerías relacionadas con la criptografía. Son útiles para suministrar funciones criptográficas a otros paquetes.

```
apt-get install openssl openssl-devel
```

Ahora se procede a la Instalación de Asterisk:

```
# cd /home/instaladores/asterisk
```

```
# ./configure
```

```
# make clean
```

```
# make install
```

Para compilar los ejemplos de configuración, se sigue la siguiente instrucción. Es muy útil, pues puede servir de referencia para realizar las configuraciones deseadas.

```
# make samples
```

Instalar la documentación de Asterisk. Es necesario tener instalado con anterioridad el paquete doxygen.

```
# apt-get install doxygen
```

```
# make progdocs
```

Para comprobar que se ha instalado correctamente, se ejecuta el comando “*asterisk*”. Cada “v” de los parámetros incrementa el nivel de depuración y la “c” abre una consola de Asterisk:

```
# asterisk -vvvc
```

B. Organización del sistema de archivos de Asterisk

```
/etc/asterisk
```

Contiene los archivos de configuración de Asterisk.

```
/usr/sbin
```

Es el directorio de los binarios del sistema. Contiene los ejecutables y scripts de Asterisk.

```
/usr/lib/asterisk
```

Contiene los objetos binarios específicos de la arquitectura Asterisk.

```
/usr/lib/asterisk/modules
```

Contiene los módulos de tiempo de ejecución para aplicaciones, drivers de canales, códecs, drivers de formato de archivos.

```
/usr/include/asterisk
```

Contiene los archivos de cabecera necesarios para construir las aplicaciones de Asterisk, drivers de canales, y otros módulos cargables.

```
/var/lib/asterisk
```

Contiene los datos variables utilizados por Asterisk en su operación normal.

```
/var/lib/asterisk/agi/bin
```

Contiene los scripts AGI para ser utilizados con las aplicaciones AGI en el plan de numeración.

/var/lib/asterisk/astdb

La base de datos de Asterisk es el equivalente al registro de Windows. Este archivo nunca se usa directamente, pero sus contenidos pueden ser visualizados y modificados en la línea de comandos de Asterisk con el conjunto de funciones “database”.

/var/lib/asterisk/images

Área de almacenamiento para las imágenes referenciadas en el plan de numeración y aplicaciones.

/var/lib/asterisk/keys

Área de almacenamiento para keys públicas y privadas usadas para la autenticación RSA con Asterisk (especialmente IAX).

/var/lib/asterisk/moh/mp3

Directorio de almacenamiento para la música en espera en formato mp3

/var/lib/asterisk/sounds

Directorio de almacenamiento para archivos de audio, usados por las aplicaciones de Asterisk.

/var/run

Contiene los tiempos de ejecución llamados pipes y archivos PID.

/var/run/asterisk/pid

Contiene los identificadores de proceso primarios (PID) de los procesos Asterisk ejecutados.

/var/run/asterisk/ct

Una llamada pipe usada por Asterisk para permitir el modo remoto de operación.

/var/spool/asterisk

Usado por los archivos de encolado de tiempo de ejecución del buzón de voz, llamadas salientes...

/var/spool/asterisk/outgoing

Monitorizado por Asterisk para las llamadas salientes. Cuando un archivo es creado en este directorio, Asterisk analiza el archivo e intenta una llamada saliente que es descargada en la PBX si es respondida.

/var/spool/asterisk/vm

Almacenamiento de los buzones de voz, saludos, entre otras funcionalidades

C. Configuración de la música en espera

Se refiere al fichero *musiconhold.conf*

Definiciones de las clases de música en espera:

[classes]

default => quietmp3:/var/lib/asterisk/mohmp3

loud => mp3:/var/lib/asterisk/mohmp3

random => quietmp3:/var/lib/asterisk/mohmp3,-z

unbuffered => mp3nb:/var/lib/asterisk/mohmp3

quietunbuf => quietmp3nb:/var/lib/asterisk/mohmp3

manual => custom:/var/lib/asterisk/mohmp3,/usr/bin/mpg123 -q -r 8000 -f 8192 -b 2048 --mono -s

D. Variables de canal Predefinidas.

Existen algunas variables de canal predefinidas por Asterisk a las cuales se le puede referenciar en la definición el plan de discado. Las variables predefinidas de Asterisk son sensibles a la diferencia entre mayúsculas y minúsculas. Generalmente estas se escriben en mayúsculas para diferenciarlas.

Ejemplo:

\${CONTEXT}: El nombre del contexto actual.

\${DATETIME}: Fecha y hora actual en el formato: YYYY-MM-DD_HH: MM: SS.

\${TIMESTAMP}: Fecha y hora actual en el formato: YYYYMMDD-HHMMSS.

\${EPOCH}: Época actual estilo UNIX (números de segundos desde 1ero de enero de1970).

\${EXTEN}: Extensión actual.

\${INVALID_EXTEN}: Extensión solicitada al redireccionar a la extensión inválida.

\${LANGUAGE}: Lenguaje actual.

\${PRIORITY}: Prioridad actual.

\${CALLINGPRES}: Variable de presentación para llamadas entrantes.

\${CALLERID}: Identidad del llamante actual (número y nombre).

\${CALLERIDNAME}: El nombre del llamante.

\${CALLERIDNUM}: El número del llamante.

\${CHANNEL}: Nombre de canal actual.

\${RDNIS}: El redireccionador DNIS actual, identificador de llamante para redireccionar la llamada.

\${SIPDOMAIN}: Dominio destino del contexto SIP de las llamadas entrantes.

\${SIP_CODEC}: Usado para habilitar el codificador SIP para una llamada.

\${SIPCALLID}: Encabezado del identificador de llamante SIP.

\${SIPUSERAGENT}: Encabezado del agente de usuario SIP.

\${UNIQUEID}: identificador de llamada SIP exclusivo.

\${HANGUPCAUSE}: Causa de colgado, código de retorno del comando Hangup.

\${DNID}: identificador de número discado.

\${DIALEDPEERNAME}: nombre del par llamado.

`\${DIALEDPEERNUMBER}`: número del par llamado.

`\${DIALEDTIME}`: tiempo de discado del número marcado.

`\${ANSWEREDTIME}`: tiempo de demora de las llamadas respondidas.

`\${DIALSTATUS}`: Estado de la llamada.

`\${ACCOUNTCODE}`: Código de cuenta.

`\${MEETMESECS}`: Cantidad de segundos en que un usuario participa en una conferencia MeetMe.

`\${TXTCIDNAME}`: Resultado de la aplicación TXTCIDName.

E. Interconexión de los servidores mediante IAX2

➤ En el servidor Asterisk de Residencia:

(*fichero iax.conf*)

[general]

register => asteriskrec:<password>@asteriskrec.uci.cu

register => asteriskip:<password>@asteriskip.uci.cu

register => asteriskdoc:<password>@asteriskdoc.uci.cu

[asteriskrec]

type = friend

user = asteriskrec

host = asteriskrec.uci.cu

[asteriskip]

type = friend

user = asteriskip

host = asteriskip.uci.cu

[asteriskdoc]

type = friend

user = asteriskdoc

host = asteriskdoc.uci.cu

(*fichero extensions.conf*)

exten => _a1XXX,1,Dial(IAX2/asteriskrec/\${EXTEN:1},30,r)

exten => _a1XXX,2,Congestion

exten => _a2XXX,1,Dial(IAX2/asteriskip/\${EXTEN:1},30,r)

exten => _a2XXX,2,Congestion

exten => _a3XXX,1,Dial(IAX2/asteriskdoc/\${EXTEN:1},30,r)

exten => _a3XXX,2,Congestion

exten => _a4XXX,1, ringing

```
exten => _a4XXX,2,Dial(SIP/${EXTEN:1},30)
exten => _a4XXX,3,Festival, Extensión no disponible
exten => _a4XXX,4,Hangup()
```

➤ **En el servidor Asterisk de Docencia:**

(*fichero iax.conf*)

```
[general]
register => asteriskrec:<password>@asteriskres.uci.cu
register => asteriskip:<password>@asteriskip.uci.cu
register => asteriskdoc:<password>@asterisrec.uci.cu
```

```
[asteriskrec]
type = friend
user = asteriskrec
host = asteriskrec.uci.cu
```

```
[asteriskip]
type = friend
user = asteriskip
host = asteriskip.uci.cu
```

```
[asteriskres]
type = friend
user = asteriskres
host = asteriskres.uci.cu
```

(*fichero extensions.conf*)

```
exten => _a1XXX,1,Dial(IAX2/asteriskrec/${EXTEN:1},30,r)
exten => _a1XXX,2,Congestión
```

```
exten => _a2XXX,1,Dial(IAX2/asteriskip/${EXTEN:1},30,r)
exten => _a2XXX,2,Congestión
```

```
exten => _a4XXX,1,Dial(IAX2/asteriskres/${EXTEN:1},30,r)
exten => _a4XXX,2,Congestión
```

```
exten => _a3XXX,1,Ringing
exten => _a3XXX,2,Dial(SIP/${EXTEN:1},30)
exten => _a3XXX,3,Festival, Extensión no disponible
exten => _a3XXX,4,Hangup()
```


➤ **En el servidor Asterisk de la IP:**

(*fichero iax.conf*)

[general]

register => asteriskrec:<password>@asteriskrec.uci.cu

register => asteriskres:<password>@asteriskres.uci.cu

register => asteriskdoc:<password>@asteriskdoc.uci.cu

[asteriskrec]

type = friend

user = asteriskrec

host = asteriskrec.uci.cu

[asteriskres]

type = friend

user = asteriskres

host = asteriskres.uci.cu

[asteriskdoc]

type = friend

user = asteriskdoc

host = asteriskdoc.uci.cu

(*fichero extensions.conf*)

exten => _a1XXX,1,Dial(IAX2/asteriskrec/\${EXTEN:1},30,r)

exten => _a1XXX,2,Congestion

exten => _a4XXX,1,Dial(IAX2/asteriskres/\${EXTEN:1},30,r)

exten => _a4XXX,2,Congestion

exten => _a3XXX,1,Dial(IAX2/asteriskdoc/\${EXTEN:1},30,r)

exten => _a3XXX,2,Congestion

exten => _a2XXX,1, Ringing

exten => _a2XXX,2,Dial(SIP/\${EXTEN:1},30)

exten => _a2XXX,3,Festival, Extensión no disponible

exten => _a2XXX,4, Hangup()

➤ **En el servidor Asterisk del Rectorado:**

(*fichero iax.conf*)

[general]

register => asteriskrec:<password>@asteriskres.uci.cu

register => asteriskip:<password>@asteriskip.uci.cu

register => asteriskdoc:<password>@asteriskdoc.uci.cu

[asteriskres]

type = friend

user = asteriskres

host = asteriskres.uci.cu

[asteriskip]

type = friend

user = asteriskip

host = asteriskip.uci.cu

[asteriskdoc]

type = friend

user = asteriskdoc

host = asteriskdoc.uci.cu

(fichero *extensions.conf*)

exten => _a4XXX,1,Dial(IAX2/asteriskrec/\${EXTEN:1},30,r)

exten => _a4XXX,2,Congestión

exten => _a2XXX,1,Dial(IAX2/asteriskip/\${EXTEN:1},30,r)

exten => _a2XXX,2,Congestión

exten => _a3XXX,1,Dial(IAX2/asteriskdoc/\${EXTEN:1},30,r)

exten => _a3XXX,2,Congestión

exten => _a1XXX,1,Ringing

exten => _a1XXX,2,Dial(SIP/\${EXTEN:1},30)

exten => _a1XXX,3,Festival, Extensión no disponible

exten => _a1XXX,4,Hangup()

GLOSARIO DE TÉRMINOS

ADSI: *Active Directory Services Interfaces*, Interfaces de Servicios de Directorio Activo. Es el término utilizado por Microsoft para referirse a su implementación de servicio de directorio en una red distribuida de computadores. Utiliza distintos protocolos (principalmente LDAP, DNS, DHCP, kerberos...).

ADPCM: *Adaptive Differential Pulse Code Modulation*, Modulación por Impulsos Codificados Diferencial Adaptivo. Son códecs de ondas que en vez de cuantificar la señal directamente, como los códecs PCM, cuantifican la diferencia entre la señal y una predicción hecha a partir de la señal, por lo que se trata de una codificación diferencial.

Ancho de banda: En redes telemáticas, el ancho de banda es la cantidad de datos que se puede enviar a través de una conexión de red en un período de tiempo dado. El ancho de banda se indica generalmente en bits por segundo (BPS), o cualquiera de los multiplicadores. El ancho de banda a menudo se utiliza como sinónimo de la tasa de transferencia de datos.

API: *Application Programming Interface*, Interfaz de Programación de Aplicaciones. Representa una interfaz de comunicación entre componentes software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación.

Buffer: Es una ubicación de la memoria en una computadora o en un instrumento digital reservada para el almacenamiento temporal de información digital.

CGI: *Common Gateway Interface*, Interfaz Común de Pasarela. Es una importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web.

CPL: *Call Processing Language*, Lenguaje de Procesamiento de Llamadas, que se utiliza en Asterisk para el tratamiento de las llamadas que se realizan, a partir de la lógica de discado.

Códecs: codificador/decodificador, describe una especificación desarrollada en software, hardware o una combinación de ambos.

Datagramas IP: Es un fragmento de paquete que es enviado con la suficiente información como para que la red pueda simplemente encaminar el fragmento hacia el equipo terminal de datos receptor, de manera independiente a los fragmentos restantes.

Digium: Empresa encargada de proveer de todo el hardware de Asterisk.

DTMF: *Dual-Tone Multi-Frequency*, Multifrecuencia de Doble Tono. Consiste en lo siguiente: Cuando el usuario pulsa en el teclado de su teléfono la tecla correspondiente al dígito que quiere marcar, se envían dos tonos, de distinta frecuencia, que la central descodifica a través de filtros especiales, detectando instantáneamente que dígito se marcó.

DSP: *Digital Signalling Processing*, Procesamiento Digital de Señales. Es un sistema basado en un procesador o microprocesador que posee un juego de instrucciones, un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad

Emisor: Es aquel objeto que codifica el mensaje y lo transmite por medio de un canal o medio hasta un receptor.

Ethernet: Es un estándar de facto de redes de computadoras de área local con acceso al medio por contienda CSMA/CD.

FXS: La interfaz de abonado externo es el puerto que envía la línea analógica al abonado. En otras palabras, es el “enchufe de la pared” que envía tono de marcado, corriente para la batería y tensión de llamada

FXO: Interfaz de central externa es el puerto que recibe la línea analógica. Es un enchufe del teléfono o aparato de fax, o el enchufe de su centralita telefónica analógica. Envía una indicación de colgado/descolgado (cierre de bucle). Como el puerto FXO está adjunto a un dispositivo, tal como un fax o teléfono, el dispositivo a menudo se denomina “dispositivo FXO”.

H.248: Especificación basada en MGCP que fuera complementaria a SIP y H.323, Originalmente este protocolo es el denominado MEGACO.

HTTP: *HyperText Transfer Protocol*, Protocolo de Transferencia de Hipertexto. Es el protocolo usado en cada transacción de la Web (WWW).

IETF: *Internet Engineering Task Force*, Grupo de Tareas de Ingeniería de Internet.

ISDN: *Integrated Services Digital Network*, Red Digital de Servicios Integrados. Es una red que procede por evolución de la Red Digital Integrada (RDI) y que facilita conexiones digitales extremo a extremo para proporcionar una amplia gama de servicios, tanto de voz como de otros tipos, y a la que los usuarios acceden a través de un conjunto de interfaces normalizados.

ITU: *International Telecommunication Union*, Unión Internacional de Telecomunicación.

IVR: *Interactive Voice Response*, Respuesta Interactiva de Voz. Consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz.

LAN: *Local Area Network*, Red de Área Local.

Latencia: La suma de retardos temporales dentro de una red.

NATs: *Network Address Translation*, Traducción de Direcciones de Red.

PBX: *Private Branch Exchanche*, Centralita Telefónica Privadas.

PCM: *Pulse Code Modulation*, Modulación por Codificación de Pulsos. Es un procedimiento de modulación utilizado para transformar una señal analógica en una secuencia de bits.

PSTN: *Public Switched Telephone Network*, Red de Telefonía Pública Conmutada.

QoS: *Quality of Service*, Calidad de Servicio.

Receptor: Es un equipo que recibe una señal, código o mensaje emitido por un transmisor.

RTP: *Real Time Protocol*, Protocolo de Tiempo Real. Es un protocolo de nivel de transporte utilizado para la transmisión de información en tiempo real, como por ejemplo audio y vídeo en una videoconferencia.

Routers: Es un dispositivo de hardware para interconexión de red de computadoras que opera en la capa tres (nivel de red). Este dispositivo permite asegurar el enrutamiento de paquetes entre redes o determinar la ruta que debe tomar el paquete de datos.

SIP: *Session Initial Protocol*, Protocolo de Inicio de Sesión. Es un protocolo estándar para la iniciación, modificación y finalización de sesiones interactivas de usuario donde intervienen elementos multimedia como el video, voz, mensajería instantánea, juegos online y realidad virtual.

Sistema multihilo: En sistemas operativos, es una característica que permite a una aplicación realizar varias tareas concurrentemente.

SIPP: Software usado en la propuesta planteada para realizar pruebas de señalización de un servidor Asterisk.

SoftPBX: PBX basado en Software; esta aplicación realiza las mismas funcionalidades de una PBX convencional.

Softphone: Es un software cuya función es simular el funcionamiento de un teléfono IP.

SMS: *Short Message Service*, Servicio de Mensajes Cortos. Es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos.

TCP: *Transmission Control Protocol*, Protocolo de Control de Transmisión. Es un protocolo de comunicación orientado a conexión y fiable del nivel de transporte.

TDM: *Time-Division Multiplexing*, Multiplexación por División de Tiempo. Es la más utilizada en la actualidad, especialmente en los sistemas de transmisión digitales. En ella, el ancho de banda total del medio de transmisión es asignado a cada canal durante una fracción del tiempo total (intervalo de tiempo).

UDP: *User Datagram Protocol*, Protocolo de Datagrama de Usuario. Es un protocolo del nivel de transporte basado en el intercambio de datagramas. Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión.

VoIP: *Voice over Internet Protocol*, Voz sobre Protocolo de Internet. Es un grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP (Internet Protocol).